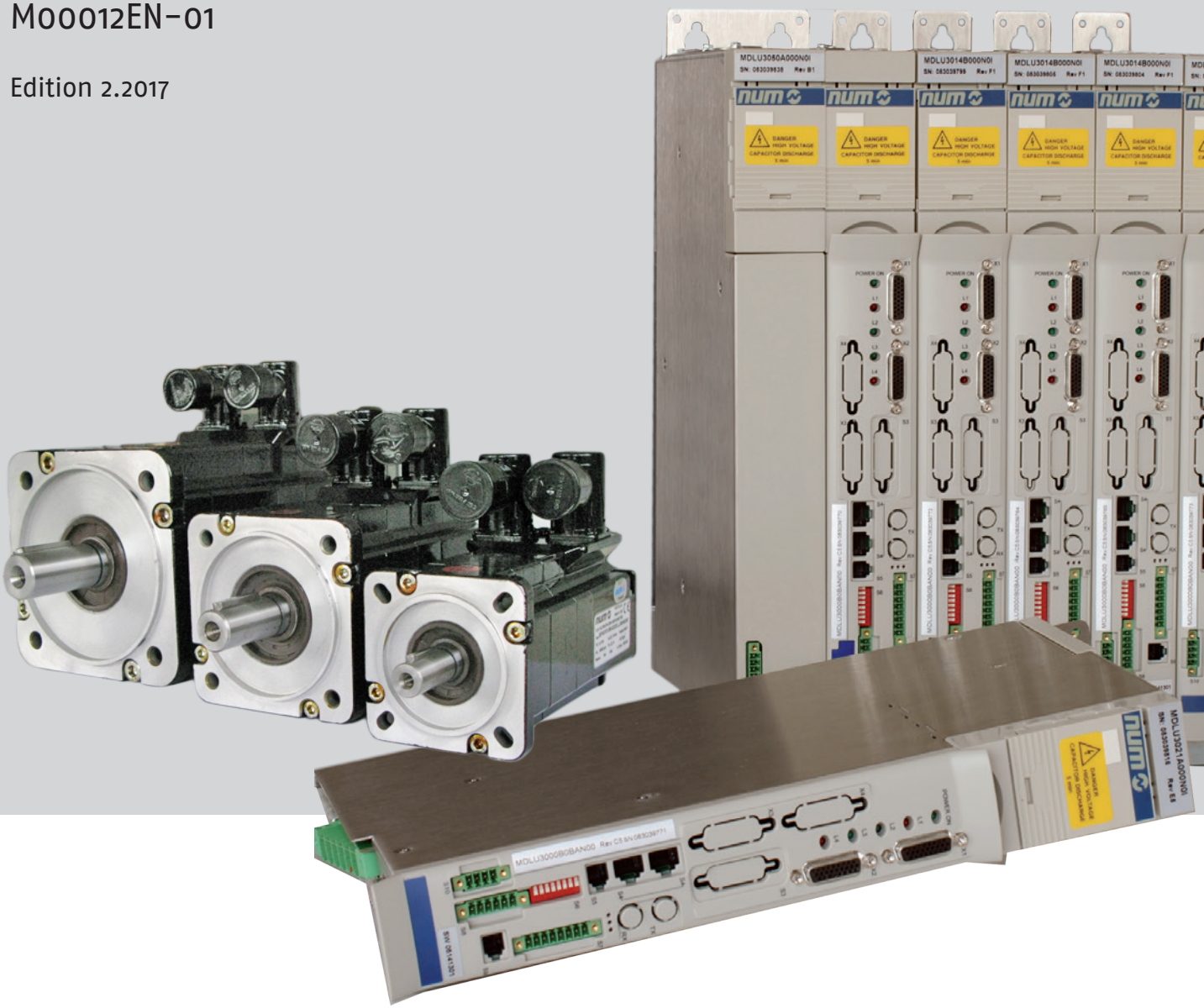
Flexium CANopen Axes

Programming and parameters manual

(for software release 3.5x and 4.5x)

M00012EN-01

Edition 2.2017



www.num.com

num
CNC HighEnd Applications

Despite the care taken in the preparation of this document, NUM cannot guarantee the accuracy of the information it contains and cannot be held responsible for any errors therein, nor for any damage which might result from the use or application of the document.

The physical, technical and functional characteristics of the hardware and software products and the services described in this document are subject to modification and cannot under any circumstances be regarded as contractual.

The programming examples described in this manual are intended for guidance only. They must be specially adapted before they can be used in programs with an industrial application, according to the automated system used and the safety levels required.

Summary

1 Foreword 17

1.1 Products certification and normatives 17
1.2 Machinery directive 17
1.3 Use of the manual 18
1.4 Flexium CANopen Axes Software User's information 18
1.5 General drives installation warnings instructions 19
1.6 Hazard and safety use instructions - Symbology use in the manual 19
1.7 View of the hazardous areas around Flexium CANopen Axes units 20
1.8 How to read the manual 21
1.9 How to read the drive parameters list with WPM 22
1.9.1 Reading a general parameters list table 22
1.9.2 How to read a parameters description - V50 example 23
1.10 How to read the drive measure point (MP) list with WPM 24
1.10.1 How to read a general measure point (MP) list table 24
1.11 Document revisions 24
1.12 Glossaries - Abbreviations 25
1.13 NUM Agency 27

2 CANopen and system overview 31

2.1 System overview 31
2.2 CANopen interface 32
2.3 CANopen communications layers 33
2.3.1 CAN Physical Layer 33
2.3.2 CAN Data Link Layer 33
2.3.3 CANopen Application Layer 33
2.4 CANopen objects 34
2.4.1 Object directory 34
2.4.2 Object index 34
2.4.3 Object group data types and profiles groups 34
2.5 CANopen profiles 35
2.5.1 Standardised profiles 35
2.5.2 DS 301 communications profile 35
2.5.3 DSP 402 device profile 36
2.5.4 Manufacturer-specific profiles 36
2.6 Field bus devices in CAN bus 37
2.7 Operating modes and functions in field bus operation 37
2.7.1 Flexium CANopen library 37
2.8 Communications profile 38
2.8.1 Object directory 38
2.8.2 Object dictionary structure 38
2.8.3 Object description 39
2.8.4 Standardised objects 39
2.9 Communications objects 40
2.9.1 CANopen message (COD-ID and COB IDs) 40
2.10 Service data communication 43
2.10.1 Overview 43

2.10.2 SDO data exchange....43
2.10.3 SDO message....44
2.10.4 Reading and writing data....45
2.11 Process data communication....49
2.11.1 Overview....49
2.11.2 PDO data exchange....49
2.11.3 PDO message....50
2.11.4 PDO addresses....53
2.11.5 PDO settings....54
2.11.6 PDO mapping....56
2.11.7 PDO transmission type....57
2.12 Emergency service....58
2.12.1 Error evaluation and handling....59
2.13 Network management services....62
2.13.1 NMT services for device control....63
2.13.2 Services for connection monitoring....64
2.13.3 Node/life guarding....65
2.13.4 Heartbeat....67
2.14 Node address....68
2.14.1 Setting the NMT node address by dip-switches....68
2.15 Transfert rate by V01 parameter....68

3 Flexium PLC library description....73

3.1 Integrating a Flexium CANopen Axes device in FXTools....73
3.1.1 Installing a CANopen device in FlexiumTools....73
3.2 Installing the “NUMDriveCANInterface” library....75
3.3 Adding a CANBus to the Flexium_RTS....77
3.3.1 CANBus Settings....79
3.4 Adding a CANopen_Manager to the CANBus....80
3.4.1 CANopen_Manager Settings....80
3.5 Adding a NUMDrive C CANopen Mono-Axis device to the CANopen_Manager....81
3.5.1 Mono-Axis device settings....82
3.5.1.1 Register: CANopen Remote Device (Heartbeat producer - consumer)....82
3.5.1.2 Register: PDO Mapping (PDO selection for Velocity or Positioning Mode)....84
3.5.1.3 Register: PDO Mapping (PDO Properties - Transmission Type settings)....85
3.5.1.4 Register: Receive PDO Mapping (Change RPDO Mapping)....86
3.5.1.5 Register: Send PDO Mapping (Change TPDO Mapping)....87
3.5.1.6 Register: Service Data Object (Add or Remove SDO’s & Change SDO values)....88
3.5.1.7 Register: CANopen I/O Mapping (Setting up the I/O Addresses)....90
3.6 Adding a NUMDrive C CANopen Bi-Axes device to the CANopen_Manager....92
3.6.1 NUMDrive C CANopen Bi-Axes device settings....94
3.6.1.1 Register: CANopen Remote Device (Heartbeat producer - consumer)....94
3.6.1.2 Register: PDO Mapping (PDO selection for Velocity or Positioning Mode)....96
3.6.1.3 Register: PDO Mapping (PDO Properties - Transmission Type settings)....97
3.6.1.4 Register: Receive PDO Mapping (Change RPDO Mapping)....100
3.6.1.5 Register: Send PDO Mapping (Change TPDO Mapping)....101
3.6.1.6 Register: Service Data Object (Add or Remove SDO’s & Change SDO values)....102
3.6.1.7 Register: CANopen I/O Mapping (Setting up the I/O Addresses)....104
3.7 NUMDriveCAN Interface library – data types and structure....106
3.7.1 Important data types....106
3.7.1.1 “DutInpNUMDriveCAN” & “DutOutNUMDriveCAN”....106
3.7.1.2 “EnumNumDriveCan_Mode”....106
3.7.1.3 “EnumNumDriveCan_Select”....106
3.7.1.4 “EnumNumDriveCan_HomingMethod”....107
3.7.1.5 “EnumNumDriveCan_Param”....107
3.8 Library structure....108
3.8.1 Overview Library Function Blocks....108

3.8.2 NUMDriveCAN_VelocityMode Structure 108
3.8.3 Parameters / Variables Description Velocity Mode 109
3.8.4 NUMDriveCAN_PositioningMode Structure 110
3.8.5 Parameters / Variables Description Positioning Mode 111
3.9 “Velocity Mode” in NUMDriveCANInterface library 113
3.9.1 Instancing the “Velocity Mode” function block 113
3.9.2 “VAR Config” for “NUMDriveCANInterface” instances 114
3.9.3 Using an instance in a POU 115
3.9.4 IntelliSense 116
3.9.5 General Settings 117
3.9.6 Mono-Axis and Bi-Axes distinction 118
3.9.7 Drive Control 119
3.9.7.1 “TorqueEnable” 119
3.9.7.2 “Operation Mode Setting” 119
3.9.7.3 “ManualControl” 120
3.9.7.4 “AutoControl” 120
3.9.8 Instance invoke 121
3.9.9 “DriveStatus” 121
3.9.10 “Alarms” 122
3.9.11 “AlarmReset” 122
3.9.12 “SDOAccess” 123
3.9.12.1 “ReadParameters” 123
3.9.12.2 “SetParameters” 124
3.9.12.3 Read / Set - Several Parameters 125
3.10 “Positioning Mode” in “NUMDriveCANInterface” library 126
3.10.1 “VAR Config” for “NUMDriveCANInterface” instances 127
3.10.2 Using an instance in a POU 128
3.10.3 IntelliSense 129
3.10.4 “General Settings” 130
3.10.4.1 Homing Enable 130
3.10.4.2 Software Position Limits 130
3.10.5 Mono-Axis and Bi-Axes distinction 131
3.10.6 “Drive Control” 132
3.10.6.1 “TorqueEnable” and “ReferenceEnable” 132
3.10.6.2 “Operation Mode Setting” 132
3.10.6.3 “ManualControl” 133
3.10.6.4 AutoControl” 133
3.10.6.5 Relative Movement 134
3.10.6.6 Change Positioning Immediately 134
3.10.7 “Homing” 135
3.10.8 Instance invoke 137
3.10.9 “DriveStatus” 138
3.10.10 “Alarms” 139
3.10.11 “AlarmReset” 139
3.10.12 “SDOAccess” 140
3.10.12.1 “ReadParameters” 140
3.10.12.2 “SetParameters” 141
3.10.12.3 Read / Set - Several Parameters 142
3.11 Modulo function 143
3.12 Extended SDO Access 144
3.13 Synchronous Start in Positioning Mode 146
3.14 Example for extended axis status information in Positioning Mode 151

4 Flexium CANopen Axes state machine and operating modes ...155

4.1 NMT management services: the Communication state machine 155
4.2 Error control mechanisms 157
4.3 Device Control 158
4.3.1 State machine 158
4.3.1.1 State device 159
4.3.1.2 State transitions 160

4.3.2 Conditions preventing state machine transitions 164
4.4 CANopen Device Profile 165
4.4.1 Standardized operating modes 165
4.4.2 Profile position operating mode 166
4.4.2.1 Profile position objects description 168
4.4.3 Profile velocity operating mode 169
4.4.3.1 Profile velocity objects description 170
4.4.4 Homing operating mode 170
4.4.4.1 Homing operating objects description 173
4.4.5 Interpolation operating mode 174
4.4.5.1 Interpolation operating objects description 176
4.4.6 Hardware limit switches and HOME switches 177
4.4.6.1 Configuration for the NUMDrive C CANopen Mono-Axis 177
4.4.6.2 Configuration for the NUMDrive C CANopen Bi-Axes 177
4.4.6.3 Limit switch status display with WPM 178

5 NUMDrive C with CANopen interface 189

5.1 NUMDrive C configuration with WPM 189
5.1.1 Motor / drive association and sensor configuration with WPM 189
5.1.1.1 Motor / drive selection 189
5.1.1.2 Sensor configuration (X1, X2, X3, X4) 190
5.1.2 NUM sensor User's configuration 191
5.1.2.1 NUM sensor table example 192
5.1.3 Drive parameters display 193
5.1.3.1 Parameters selection and modification 194
5.1.3.2 Bit parameters selection and modification 195
5.1.3.3 Drive filter selection and modification 196
5.2 NUMDrive C Mono-Axis objects description 197
5.2.1 NC Units 197
5.2.2 Objects description 197
Object 1000h: Device Type 197
Object 1001h: Error Register 198
Object 1003h: Pre-defined Error Field 199
Object 1005h: COB-ID SYNC 200
Object 1006h: Communication Cycle Period 200
Object 1012h: COB-ID Time Stamp 201
Object 100Ch: Guard Time 201
Object 100Dh: Life Time Factor 202
Object 1014h: COB-ID EMCY 203
Object 1015h: Inhibit Time Emergency 203
Object 1016h: Heartbeat Consumer Entries 204
Object 1017h: Producer Heartbeat Time 205
Object 1018h: Identity Object 206
Object 1020h: Verify Configuration 208
Object 1200h: Server SDO Parameter 1 209
Object 1201h: Server SDO Parameter 2 210
Object 1400h: Receive PDO Communication Parameter 1 212
Object 1401h: Receive PDO Communication Parameter 2 214
Object 1402h: Receive PDO Communication Parameter 3 216
Object 1403h: Receive PDO Communication Parameter 4 218
Object 1600h: Receive PDO Mapping Parameter 1 220
Object 1601h: Receive PDO Mapping Parameter 2 221
Object 1602h: Receive PDO Mapping Parameter 3 222
Object 1603h: Receive PDO Mapping Parameter 4 223
Object 1800h: Transmit PDO Communication Parameter 1 225
Object 1801h: Transmit PDO Communication Parameter 2 227
Object 1802h: Transmit PDO Communication Parameter 3 229
Object 1803h: Transmit PDO Communication Parameter 4 231
Object 1A00h: Transmit PDO Mapping Parameter 1 233
Object 1A01h: Transmit PDO Mapping Parameter 2 234
Object 1A02h: Transmit PDO Mapping Parameter 3 236

Object 1A03h: Transmit PDO Mapping Parameter 4 237
Object 3000h: Drive internal status flags 239
Object 3001h: TPDO trigger transmission window 240
Object 3002h: RPDO1 shadow before SYNC validation 241
Object 6007h: Abort Connection Option Code 242
Object 603Fh: Error Code 243
Object 6040h: Controlword 243
Object 6041h: Statusword 244
Object 605Ah: Quick Stop Option Code 245
Object 6060h: Modes of Operation 246
Object 6061h: Modes of Operation Display 247
Object 6064h: Position Actual Value in User Unit 247
Object 6065h: Following Error Window 248
Object 6066h: Following Error Time Out 248
Object 6067h: Position Window 249
Object 606Ch: Velocity Actual Value 249
Object 606Dh: Velocity Window 250
Object 607Ah: Target Position 251
Object 607Bh: Position Range Limit 252
Object 607Ch: Home Offset 253
Object 607Dh: Software Position Limit 254
Object 607Fh: Max Profile Velocity (associated to V98) 256
Object 6081h: Profile Velocity in pp-mode 257
Object 6083h: Profile Acceleration 258
Object 6084h: Profile Deceleration 258
Object 6085h: Quick Stop Deceleration 259
Object 6086h: Motion Profile Type 259
Object 6098h: Homing Method 260
Object 6099h: Homing Speeds 261
Object 609Ah: Homing Acceleration 262
Object 60C0h: Interpolation Sub Mode Select 262
Object 60C1h: Interpolation Data Record 263
Object 60C2h: Interpolation Time Period 264
Object 60F2h: Positioning Option Code 265
Object 60FDh: Digital Inputs 266
Object 60FFh: Target Velocity 267
Object 6502h: Supported Drive Modes 268
5.2.3 Object dictionary access rights 269
5.3 NUMDrive C Bi-Axes objects description 271
5.3.1 NC Units 271
5.3.2 Objects description 271
Object 1000h: Device Type 271
Object 1001h: Error Register 272
Object 1003h: Pre-defined Error Field 273
Object 1005h: COB-ID SYNC 274
Object 1006h: Communication Cycle Period 274
Object 100Ch: Guard Time 275
Object 100Dh: Life Time Factor 276
Object 1012h: COB-ID Time Stamp 277
Object 1014h: COB-ID EMCY 278
Object 1015h: Inhibit Time Emergency 278
Object 1016h: Heartbeat Consumer Entries 279
Object 1017h: Producer Heartbeat Time 280
Object 1018h: Identity Object 281
Object 1020h: Verify Configuration 283
Object 1200h: Server SDO Parameter 1 284
Object 1201h: Server SDO Parameter 2 285
Object 1400h: Receive PDO Communication Parameter 1 [Axis$_n$] 287
Object 1401h: Receive PDO Communication Parameter 2 [Axis$_n$] 289
Object 1402h: Receive PDO Communication Parameter 3 [Axis$_n$] 291
Object 1403h: Receive PDO Communication Parameter 4 [Axis$_n$] 293
Object 1440h: Receive PDO Communication Parameter 65 [Axis$_{n+1}$] 295

Object 1441h: Receive PDO Communication Parameter 66 [Axis$_{n+1}$]....297
Object 1442h: Receive PDO Communication Parameter 67 [Axis$_{n+1}$]....299
Object 1443h: Receive PDO Communication Parameter 68 [Axis$_{n+1}$]....301
Object 1600h: Receive PDO Mapping Parameter 1 [Axis$_n$]....303
Object 1601h: Receive PDO Mapping Parameter 2 [Axis$_n$]....304
Object 1602h: Receive PDO Mapping Parameter 3 [Axis$_n$]....305
Object 1603h: Receive PDO Mapping Parameter 4 [Axis$_n$]....306
Object 1640h: Receive PDO Mapping Parameter 65 [Axis$_{n+1}$]....308
Object 1641h: Receive PDO Mapping Parameter 66 [Axis$_{n+1}$]....309
Object 1642h: Receive PDO Mapping Parameter 67 [Axis$_{n+1}$]....310
Object 1643h: Receive PDO Mapping Parameter 68 [Axis$_{n+1}$]....311
Object 1800h: Transmit PDO Communication Parameter 1 [Axis$_n$]....313
Object 1801h: Transmit PDO Communication Parameter 2 [Axis$_n$]....315
Object 1802h: Transmit PDO Communication Parameter 3 [Axis$_n$]....317
Object 1803h: Transmit PDO Communication Parameter 4 [Axis$_n$]....319
Object 1840h: Transmit PDO Communication Parameter 65 [Axis$_{n+1}$]....321
Object 1841h: Transmit PDO Communication Parameter 66 [Axis$_{n+1}$]....323
Object 1842h: Transmit PDO Communication Parameter 67 [Axis$_{n+1}$]....325
Object 1843h: Transmit PDO Communication Parameter 68 [Axis$_{n+1}$]....327
Object 1A00h: Transmit PDO Mapping Parameter 1 [Axis$_n$]....329
Object 1A01h: Transmit PDO Mapping Parameter 2 [Axis$_n$]....330
Object 1A02h: Transmit PDO Mapping Parameter 3 [Axis$_n$]....332
Object 1A03h: Transmit PDO Mapping Parameter 4 [Axis$_n$]....333
Object 1A40h: Transmit PDO Mapping Parameter 65 [Axis$_{n+1}$]....335
Object 1A41h: Transmit PDO Mapping Parameter 66 [Axis$_{n+1}$]....336
Object 1A42h: Transmit PDO Mapping Parameter 67 [Axis$_n$]....337
Object 1A43h: Transmit PDO Mapping Parameter 68 [Axis$_n$]....338
Object 3000h: Drive internal status flags [Axis$_n$]....340
Object 3001h: TPDO trigger transmission window [Axis$_n$]....341
Object 3002h: RPDO1 shadow before SYNC validation [Axis n]....342
Object 3100h: Drive internal status flags [Axis$_{n+1}$]....343
Object 3101h: TPDO trigger transmission window [Axis$_{n+1}$]....343
Object 3102h: RPDO65 shadow before SYNC validation [Axis n+1]....344
Object 6007h: Abort Connection Option Code [Axis$_n$]....345
Object 603Fh: Error Code [Axis$_n$]....346
Object 6040h: Controlword [Axis$_n$]....346
Object 6041h: Statusword [Axis$_n$]....347
Object 605Ah: Quick Stop Option Code [Axis$_n$]....348
Object 6060h: Modes of Operation [Axis$_n$]....349
Object 6061h: Modes of Operation Display [Axis$_n$]....350
Object 6064h: Position Actual Value in User Unit [Axis$_n$]....350
Object 6065h: Following Error Window [Axis$_n$]....351
Object 6066h: Following Error Time Out....351
Object 6067h: Position Window [Axis$_n$]....352
Object 606Ch: Velocity Actual Value (Axes$_n$)....352
Object 606Dh: Velocity Window [Axis$_n$]....353
Object 607Ah: Target Position [Axes$_n$]....354
Object 607Bh: Position Range Limit [Axes$_n$]....355
Object 607Ch: Home Offset [Axes$_n$]....356
Object 607Dh: Software Position Limit [Axes$_n$]....357
Object 607Fh: Max Profile Velocity [Axes$_n$]....359
Object 6081h: Profile Velocity in pp-mode [Axis$_n$]....360
Object 6083h: Profile Acceleration [Axis$_n$]....361
Object 6084h: Profile Deceleration [Axis$_n$]....361
Object 6085h: Quick Stop Deceleration [Axis$_n$]....362
Object 6086h: Motion Profile Type [Axis$_n$]....362
Object 6098h: Homing Method [Axis$_n$]....363
Object 6099h: Homing Speeds [Axis$_n$]....364
Object 609Ah: Homing Acceleration [Axis$_n$]....365
Object 60C0h: Interpolation Sub Mode Select [Axis$_n$]....365
Object 60C1h: Interpolation Data Record [Axis$_n$]....366
Object 60C2h: Interpolation Time Period [Axis$_n$]....367
Object 60F2h: Positioning Option Code....368

Object 60FDh: Digital Inputs [Axis$_n$] 369
Object 60FFh: Target Velocity [Axis$_n$] 370
Object 6502h: Supported Drive Modes [Axis$_n$] 371
Object 67FFh: Single Device Type [Axis$_n$] 372
Object 6807h: Abort Connection Option Code [Axis$_{n+1}$] 372
Object 683Fh: Error Code [Axis$_{n+1}$] 373
Object 6840h: Controlword [Axis n+1] 373
Object 6841h: Statusword [Axis$_{n+1}$] 374
Object 685Ah: Quick Stop Option Code [Axis$_{n+1}$] 374
Object 6860h: Modes of Operation [Axis$_{n+1}$] 375
Object 6861h: Modes of Operation Display [Axis$_{n+1}$] 375
Object 6864h: Position Actual Value in User Unit [Axis$_{n+1}$] 376
Object 6865h: Following Error Window [Axis$_{n+1}$] 376
Object 6866h: Following Error Time Out 377
Object 6867h: Position Window [Axis$_{n+1}$] 377
Object 686Ch: Velocity Actual Value [Axis$_{n+1}$] 378
Object 686Dh: Velocity Window [Axis$_{n+1}$] 378
Object 687Ah: Target Position [Axis$_{n+1}$] 379
Object 687Bh: Position Range Limit [Axis$_{n+1}$] 380
Object 687Ch: Home Offset [Axis$_{n+1}$] 381
Object 687Dh: Software Position Limit [Axis$_{n+1}$] 382
Object 687Fh: Max Profile Velocity [Axis$_{n+1}$] 383
Object 6881h: Profile Velocity in pp-mode [Axis$_{n+1}$] 383
Object 6883h: Profile Acceleration [Axis$_{n+1}$] 384
Object 6884h: Profile Deceleration [Axis$_{n+1}$] 384
Object 6885h: Quick Stop Deceleration [Axis$_{n+1}$] 385
Object 6886h: Motion Profile Type [Axis$_{n+1}$] 385
Object 6898h: Homing Method [Axis$_{n+1}$] 386
Object 6899h: Homing Speeds [Axis$_{n+1}$] 387
Object 689Ah: Homing Acceleration [Axis$_{n+1}$] 388
Object 68C0h: Interpolation Sub Mode Select [Axis$_{n+1}$] 388
Object 68C1h: Interpolation Data Record [Axis$_{n+1}$] 389
Object 68C2h: Interpolation Time Period [Axis$_{n+1}$] 390
Object 68F2h: Positioning Option Code 391
Object 68FDh: Digital Inputs [Axis$_{n+1}$] 392
Object 68FFh: Target Velocity [Axis$_{n+1}$] 392
Object 6D02h: Supported Drive Modes [Axis$_{n+1}$] 393
Object 6FFFh: Single Device Type [Axis$_{n+1}$] 393
5.3.3 Object dictionary access rights 394
5.4 NUMDrive C general parameters list 397
5.5 NUMDrive C parameters description 409
5.5.1 Mono-Axis and Bi-Axes parameters description 409
V01 CAN configuration 409
V20 Sensor connectors X3, X4 configuration (Bi-Axes configuration only) 410
V21 TP1/2 output configuration (Bi-Axes configuration only) 411
V22 PROGO1/2 output configuration (Bi-Axes configuration only) 412
V23 Do not use 412
V30 General Drive Configuration 413
V30 Use of the bits 413
V32 Configuration Braking following alarms 414
V32 Use of configuration bit 1 - Braking modalities 415
V33 Alarm Reset Configuration 416
V33 Use of the bits 417
V34 Configuration DF1 / DF2 relay [1=break] 418
V34 Use of the bits 418
V35 SW function 419
V41 Motor zero speed threshold, enable, brake (Linear motor) 420
V42 Motor zero speed threshold, enable, brake 420
V43 Braking Timeout on configuration V30 - V32 parameters 420
V44 Braking Current on configuration V30 - V32 parameters 421
V50 Torque current limit [0] 422

V51 Proportional speed loop gain [0] 422
V52 Integral speed loop gain [0] 423
V53 Do not use 423
V57 Torque current limit - high range [0] 423
V58 Proportional speed loop gain - high range [0] 424
V59 Integral speed loop gain - high range [0] 424
V60 Proportional speed loop gain [0] 424
V75 Do not use 424
V80 Set Point Weight speed loop gain 425
V81 Anti-windup speed loop gain 425
V94 Speed reference Limit [% maximum speed] [0] 426
V95 Maximum motor speed [0] 426
V96 Threshold for overspeed alarm [0] 427
V97 Speed reference limit [% max. speed] [0] 427
V98 Maximum motor speed [0] 428
V99 Overspeed alarm threshold [0] 429
V108 Rated motor current [I^2t] 430
V108 (V115) Thermal image management 431
V109 Vertical load compensation current 432
V115 Rated motor current [I^2t] 432
V125 Current percentage V108 for EPSM 433
V128 Locked rotor current I^2t (% V108) 433
V129 Locked rotor current I^2t (% V115) 434
V130 Associate MP with TP1 (Analog / Digital) 434
V131 Associate MP with TP2 (Analog / Digital) 435
V132 Associate MP with TP3 (Digital) 435
V133 Associate MP with TP4 (Digital) 436
V134 Multiplication factor for TP1 436
V135 Multiplication factor for TP2 437
V136 Offset for analog TP1 test points 438
V137 Offset for analog TP2 test points 439
V138 Configuration of analog TP 439
V139 Configuration of Programmable Relays 440
V139 Use of the configuration bits 441
Bit 0 = Brake management [1=active] 441
Bit 1 = Brake management output on [0 = PROGO1, 1 = PROGO2] 441
Bit 2 = Absolute value for programmable relay 1 441
Bit 3 = Absolute value for programmable relay 2 441
Bit 4 or 5 = Programmable relay 1 / 2 mode - 442
Bit 6 or 7 = Logic of prog. relay 1 [1=positive, 0=negative] 442
V139 Use of the Programmable relays 443
PROGO1 use in modality brake management 443
PROGO1 use in modality Positive logic, Thresholds, Relative value 443
PROGO1 use in modality Negative logic, Thresholds, Relative value 443
PROGO1 use in modality Positive logic, Bands, Absolute value 444
PROGO1 use in modality Negative logic, Bands, Absolute value 444
PROGO1 use in modality Positive logic, Thresholds, Absolute value 445
PROGO1 use in modality Negative logic, Thresholds, Absolute value 445
PROGO1 use in modality Positive logic, Bands, Relative value 446
PROGO1 use in modality Negative logic, Bands, Relative value 446
PROGO1 use in modality Positive logic, Bands, Absolute value 447
PROGO1 use in modality Negative logic, Bands, Absolute value 447
V140 Associate MP with programmable relay 1 448
V141 Programmable relay 1 threshold 1 449
V142 Programmable relay 1 threshold 2 449
V143 Programmable relay 1, MP scale factor 450
V144 Programmable relay 1 tripping time constant 451
V150 Associate MP with programmable relay 2 452
V151 Programmable relay 2, threshold 1 452
V152 Programmable relay 2, threshold 2 452
V153 Programmable relay 2, MP scale factor 452
V154 Programmable relay 2 tripping time constant 453
V160 Do not use 453

V165 Do not use 453
V166 Do not use 453
V169 Do not use 453
V170 Do not use..... 454
V171 Do not use..... 454
V172 Do not use 454
V173 Do not use 454
V175 Do not use..... 454
V176 Do not use 454
V180 Current percentage (V108) for EPS 455
V182 U/V Phase relative weight for EPS..... 455
V183 Phases V/W relative weight for EPS 456
V190 Do not use 456
V191 Do not use 456
V205 Do not use 456
V206 Do not use 456
V207 Do not use 457
V208 Do not use 457
V209 Do not use 457
V210 Do not use 457
V218 Do not use 457
V220 Reference configuration..... 458
V220 Use of the configuration bits.....459
V221 Enable configurations..... 460
V225 Digital input configuration [GP_IN_A or B].....461
V258 Motor sensor type 462
V260 Speed regulator Parameters exchange 463
V261 Position regulator Parameters exchange 464
V266 Do not use 465
V268 Electrical phase offset Hall effect sensor.....465
V269 Position offset for electrical phasing 465
V270 Motor PTC-NTC value..... 466
V272 Motor thermal sensor configuration 467
V273 Electrical phase offset management.....467
V301 Proportional position gain [0] 468
V306 S1 sensor MULTI coefficient [0]..... 469
V307 S1 sensor DIVI coefficient [0]..... 469
V310 S1 sensor offset in Controller physical unit [0] 470
V310 Application example S1 sensor type P (High resolution Multiturn encoder) 471
V310 Application example S1 sensor type Q (High resolution Singleturn encoder) .472
V311 S2 sensor MULTI coefficient [0].....473
V312 S2 sensor DIVI coefficient [0].....474
V313 S2 sensor offset in Controller physical unit [0] 474
V313 Application example S1 sensor type P (High resolution Multiturn encoder) .475
V314 Sinusoidals signal range of S1 sensor 476
V315 Sinusoidals signal range of S2 sensor 477
V317 S1 sensor power supply voltage.....478
V318 S2 sensor power supply voltage.....478
V340 Step of speed calculation S1.....479
V341 Step of speed calculation S2..... 480
V342 Speed filter by S1 sensor 481
V343 Speed filter by S2 sensor 481
V344 S1 Sensor code ID 482
V345 S2 Sensor code ID 482
V346 S1 Sensor number 483
V347 S2 Sensor number..... 483
V348 S1 Sensor configuration..... 484
V348 Use of configuration bits from bit 0 to bit 5 485
V348 Use of configuration bits from bit 6 to bit 20 486
V349 S2 Sensor configuration..... 488
V349 Use of configuration bits from bit 0 to bit 5 489
V349 Use of configuration bits from bit 6 to bit 20 490
V350 Position loop sensor configuration 492

V351 Speed loop sensor configuration....492
V352 Current loop sensor configuration....492
V353 Period of position sensor/motor turn [0] 493
V353 Application example with 1Vpp sinusoidal encoder 494
V353 Application example with 1Vpp sinusoidal optical scale.... 494
V355 AP02: Mechanical ratio load turns 495
V356 AP02: Mechanical ratio sensor turns 495
V360 Period by turn sensor S1.... 496
V361 Absolute resolution by turn sensor S1.... 496
V365 Number of absolute sensor turns S1....497
V368 Intervals between coded references S1....497
V370 Number of absolute sensor turns S2....498
V371 Absolute resolution by turn sensor S2....498
V375 Number of absolute sensor turns S2.... 499
V378 Intervals between coded references S2.... 499
V380 Period of linear sensor S1.... 500
V381 Absolute resolution of linear sensor S1.... 500
V390 Period of linear sensor S2....501
V391 Absolute resolution of linear sensor S2....501
V500 Filter configuration....502
V500 Use of bit 7 - Filters UPDATE on line....503
V500xx - V501 ÷ V549 Filter on torque and speed reference 504
V500xx - FILTER 1, 2 for torque current reference.... 505
V500xx - FILTER 3, 4 for torque current reference.... 506
V500xx - FILTER 1 for speed reference and FILTER notch for AP13....507
V1050 Torque current limit [1].... 508
V1051 Proportional speed loop gain [1].... 509
V1052 Integral speed loop gain [1] 509
V1053 Do not use 510
V1057 Torque current limit - High range [1] 510
V1058 Proportional speed loop gain - High range [1].... 511
V1059 Integral speed loop gain - High range [1] 511
V1060 Proportional speed loop gain [1].... 512
V1075 Do not use 512
V1094 Speed reference Limit [% maximum speed] [1].... 512
V1095 Maximum motor speed [1] 512
V1096 Threshold for overspeed alarm [1].... 513
V1097 Speed reference limit [% max. speed] [1].... 513
V1098 Maximum motor speed [1] 514
V1099 Overspeed alarm threshold [1] 515
V1301 Proportional position gain [1].... 516
V1306 S1 sensor MULTI coefficient [1].... 517
V1307 S1 sensor DIVI coefficient [1] 518
V1310 S1 sensor offset in Controller physical unit [1].... 518
V1311 S2 sensor MULTI coefficient [1] 519
V1312 S2 sensor DIVI coefficient [1].... 519
V1313 S2 sensor offset in Controller physical unit [1].... 520
V1353 Period of position sensor/motor turn [1].... 520
5.6 NUMDrive C test points with WPM.... 521
5.6.1 Drive test point.... 521
5.6.2 Test Points (TP) and Measure Points (MP) display 522
5.6.3 Measure points general list (MP).... 523
5.6.4 Use of measure points: notes ( ).... 527
5.7 NUMDrive C application - User example.... 528
5.7.1 Speed loop parameters optimization V51 / V52 528
5.7.1.1 Speed loop parameters optimization procedure.... 528
5.7.1.2 Speed loop parameters optimization procedure A 529
5.7.1.3 Speed loop parameters optimization procedure B 531
5.7.2 Programmable relais 1 or 2 application examples.... 533
5.7.2.1 Control of an axis in motion (using relay no. 1 - Output on PROGO1) 533
5.7.3 Motor brake management.... 534
5.7.3.1 Management of motor holding brake 534

5.7.3.2 Use of the Programmable relays in modality brake management 535
5.7.3.3 Management of motor holding brake in case of alarms 536
5.7.4 MULTI /DIVI parameters utilization V306, V307 537
5.7.4.1 MULTI/DIVI with S1 sensor type P, Q 537
5.7.4.2 MULTI/DIVI with type Q motor sensor for spindle motor with C axis application 538
5.7.5 Sensors configuration and applications 539
5.7.5.1 Enter new sensor configuration 539
5.7.5.2 Sinusoidal rotary encoder 541
5.7.5.3 Sinusoidal linear scale encoder 543
5.7.5.4 Linear sinusoidal encoder for linear motor 545
5.7.5.5 Rotary sinusoidal encoder for direct torque motor 547
5.7.6 Homing mode and homing setting 549
5.7.6.1 Homing mode information and functional description 549
5.7.6.2 General definitions and Homing Methods 550
5.7.7 Index Pulse identification (Pulse Mark) 554
5.8 Diagnostic signalling and software alarms 555
5.8.1 Alarm signalling 555
5.8.2 Alarm signalling 556
5.8.2.1 Led POWER ON status 556
5.8.3 Mono-Axis alarm signalling status 557
5.8.3.1 Mono-Axis alarm possible causes/remedy 557
5.8.4 Bi-Axes alarm signalling status 558
5.8.4.1 Bi-Axes alarm possible causes/remedy 558
5.8.5 Alarms configuration 559
5.8.5.1 General drive alarms / securities configuration 560
5.8.6 Alarms reset procedure 561
5.8.7 Alarms signalling 562
5.8.8 Alarms extension signalling 566
5.8.8.1 Extension for 02, 03, 07, 08 alarms 566
5.8.8.2 Extension for 09 alarm Stegmann encoder 567
5.8.8.3 Extension for 09 alarms EnDat encoder 567
5.8.8.4 Alarm 09 common extensions from 200 to 261 568
5.8.8.5 Extension for alarms from 10 to 15 570
5.8.8.6 Extension for alarm 16 571
5.8.8.7 Extension for alarm 18, 37, 38, 49 574
5.8.8.8 Extension for alarm 52 575
5.8.8.9 Extension for alarm 53 576
5.8.8.10 Extension for alarm 54, 55, 56, 90, 91 577
5.8.9 Meaning of the Word 0x1AF2 by WPM (PLC Status word) 578
5.8.10 Meaning of the Word 0x1A72 by WPM (Warning register) 581
5.8.11 Meaning of the Word 0x1AF1 by WPM (NC Status word) 582

Page deliberately empty.

Summary

1 Foreword 17
1.1 Products certification and normatives 17
1.2 Machinery directive 17
1.3 Use of the manual 18
1.4 Flexium CANopen Axes Software User's information 18
1.5 General drives installation warnings instructions 19
1.6 Hazard and safety use instructions - Symbology use in the manual 19
1.7 View of the hazardous areas around Flexium CANopen Axes units 20
1.8 How to read the manual 21
1.9 How to read the drive parameters list with WPM 22
1.9.1 Reading a general parameters list table 22
1.9.2 How to read a parameters description - V50 example 23
1.10 How to read the drive measure point (MP) list with WPM 24
1.10.1 How to read a general measure point (MP) list table 24
1.11 Document revisions 24
1.12 Glossaries - Abbreviations 25
1.13 NUM Agency 27

Page deliberately empty.

1 Foreword

1.1 Products certification and normatives

CE

NUMDrive C used in conformity with the relevant manuals AMOMAN012 and AMOMAN007 fullfills the basic requirements of Electromagnetic Compatibility and of Electrical Safety as prescribed by the Directives:

- **2004/108/EC dated 15th December 2004**
- **2006/95/EC dated 19th December 2006**

The NUMDrive C conformity to the EMC directive 2004/108/EC can only be assessed if the drive is installed inside a machine or cabinet. The installation instruction inside of this manual must be followed.

For more EMC detailed informations please refer to the M00027 “Cabinet layout and EMC Guide - Installation and Wiring Guide manual”.

1.2 Machinery directive

- **2006/42/EC**

The designed products are intended to be incorporated into machinery or assembled with other machinery to constitute products covered by the council directive. Commissioning is forbidden until the products have been integrated in a machine that conforms with the directive 2006/42/EC.

roducts have been integrated in a machine that conforms with the directive 2006/42/EC.

1.3 Use of the manual

This manual is for use by qualified service personnel only. To reduce the risk due to an improprery use of the product, do not perform any servicing other than that contained in this manual unless you are qualified to do so.

To ensure proper use of these products, please always read this User's manual carefully and retain for future reference. Should the unit require maintenance, please contact a Num authorized service center only.

1.4 Flexium CANopen Axes Software User's information

Before connecting, operating or adjusting these products, please read these following User's software information:

- Flexium CANopen Axes Mono-Axis must be associated with software release version is 3.5x.
- Flexium CANopen Axes Bi-Axes must be associated with software release version is 4.5x.
- All alarms extensions indicated and described in this chapter are referred to the drive software release till 3.5x and 4.5.x version only.

Therefore, to programming and adjust in the right mode the appropriate drive parameters, please consult carefully the chapter 1.8 "How to read the manual".

Num declines all responsibility for injury or damage to persons or objects due to incorrect use of this product.

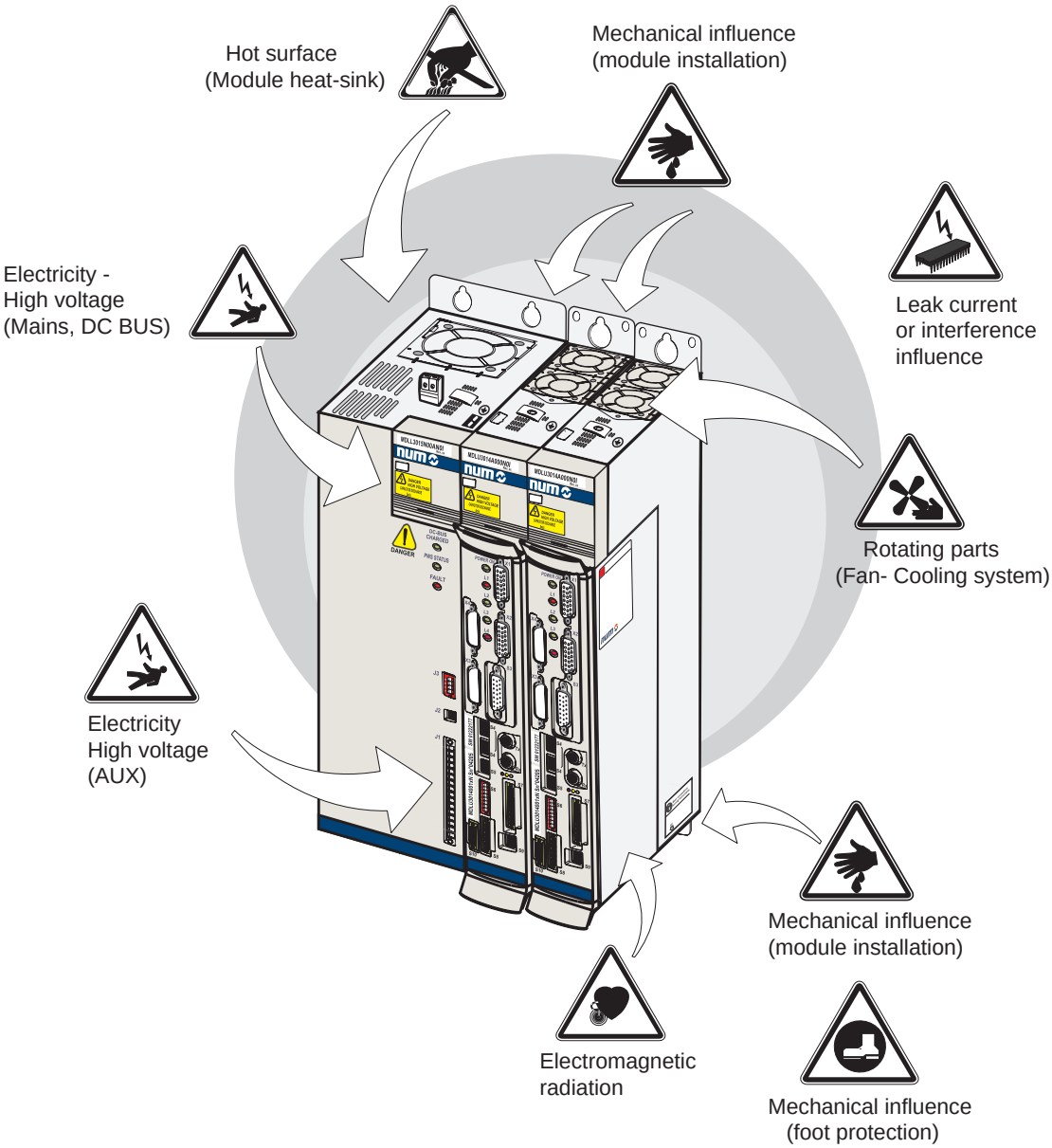
1.5 General drives installation warnings instructions

- All parameters described at modification level 1 and 2, are parameters which may be adjusted during a standard drive setup. All the international, state, regional and local standards must be complied during the installation and during the use of this product.
- Only properly qualified personnel having extensive knowledge in servomotor technology should install, commission and/or maintain the product. All parameters classified at level superior than 2 are reserved parameters, and they never must be adjusted. Please refer to the chapter 4.
- To comply with current safety regulations, all wiring and connections must be made in accordance with the electrical diagrams shown in the NUMDrive C installation manuals code AMOMAN012. NUM declines all responsibility for injury or damage to persons or objects due to incorrect use of this product.
- Use only original NUM cables. The original NUM cables have special characteristics (capacitance value...). Not to respect these characteristics can cause serious damage to motors.
- During the servomotor installation, always make sure that the connection cable used is the one established in the NUMDrive C installation manual code AMOMAN012.
  The use of an incorrect cable leads to the violation of the international standards.

1.6 Hazard and safety use instructions - Symbology use in the manual

Symbol	Meaning of the symbol
	The information " **i** " icone indicates important notes on product use.
	The STOP icone indicates a wrong operation which may damage the product.
	The TRIANGLE icone indicates hazard for human beings and this information is referring to personal damage! Whenever one of these icons appear, please beware, use and apply adequate precautions to avoid hazards or damage situations during the drive installation and commissioning operations! Please beware, consult chapter "Hazardous areas around drives / motors". Please beware! Prevent and safety instruction against accidental contact with electrical componenets.

1.7 View of the hazardous areas around Flexium CANopen Axes units



NUM declines all responsibility for injury or damage to persons or objects due to incorrect use of this product.

1.8 How to read the manual

This manual is structured and divided into twelve main chapters as indicated here-below.
We suggest to follow these read instructions in such away to find easily your information.

Chapter	Description
1	**Foreword** This chapter introduces all warnings, hazardous areas and all safety infomation on product use Please take into account these instructions before to switch on the drive.
2	**CANopen and system overview** This chapter introduces: - CANopen overview, - CAN bus and CANopen interface, - CANopen communication layers, Objects (directory, index, groups and type), - Profiles, Filed bus, Communication objects, - Service data communication (SDO data exachange, SDO message), - Process data communication (PDO addresses, PDO settings, Enable PDO, PDO mapping...), ...
3	**Flexium PLC library description** This chapter introduces: - PLC libraries integration, how to programming and make the integration / installation of a new device, - How to make a Mono-Axis, Bi-Axes settings of PDO mapping, SDO mapping, Registers, - Read / Set parameters and variables, - Setting I/O addresses, - Homing setting, - Drive status / alarms reset and setting,... - How to use of “Velocity Mode” and “Positioning Mode” libraries.
4	**Flexium CANopen Axes state machine and operating modes** This chapter introduces: - Drive state machine and the operating mode - Describes NMT management service, - Error control and Device Control, - State of Machine, - CANopen Device Profile...
5	**NUMDrive C with CANopen interface** This chapter introduces: - Main drive configuration, - How to make a motor / drive association, sensor configuration, parameter selection, - How to make a parameter display and modification, drive filter selection and modification. - How to use the Mono-Axis and Bi-Axes objects and their description, - General NUMDrive C parameters list and the detailed drive parameters description (included adjustement range, defualt value, motor use, HALT / Modification Level requirement and typically User’s example) - Drive test points (MP), - Typical User example, - Drive diagnostic signalling and software alarms. (with included the alarms “Possible cause / Possible remedy” by troubleshoot information tables and the meaning of the Word 0x1AF1, Word 0x1AF2, Word 0x1AF72.

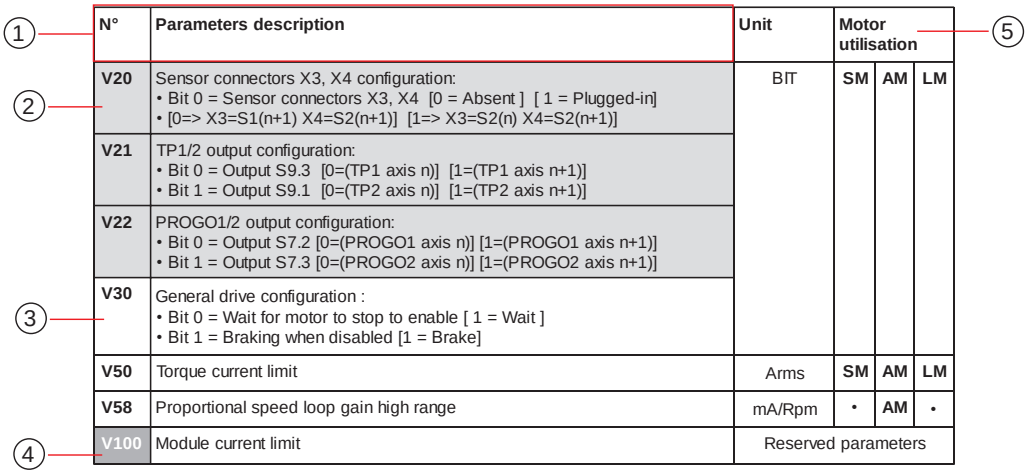
1.9 How to read the drive parameters list with WPM

All the parameters indicated and described in this manual are parameters classified at modification level 1 and 2. In the chapter 5 of the manual a quickly general parameters list informations is described. Please also refer to the chapter 5 of the this manual to give the full parameters description.
(See How to read a parameter description on next page)

These parameters are typically parameters which have to be adjusted during a standard drive setup and commissioning.

All parameters classified at level superior than 2 are reserved parameters, and they never must be adjusted. To ensure a proper use of the parameters, please always read the here-below legend carefully to reduce the risk due to an impropery use of them.

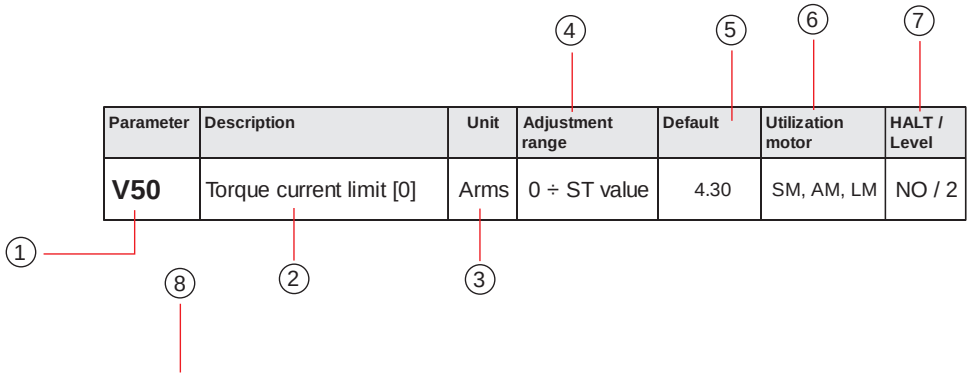
1.9.1 Reading a general parameters list table



N°	Parameters description	Unit	Motor utilisation		
V20	Sensor connectors X3, X4 configuration: • Bit 0 = Sensor connectors X3, X4 [0 = Absent] [1 = Plugged-in] • [0=> X3=S1(n+1) X4=S2(n+1)] [1=> X3=S2(n) X4=S2(n+1)]	BIT	SM	AM	LM
V21	TP1/2 output configuration: • Bit 0 = Output S9.3 [0=(TP1 axis n)] [1=(TP1 axis n+1)] • Bit 1 = Output S9.1 [0=(TP2 axis n)] [1=(TP2 axis n+1)]				
V22	PROGO1/2 output configuration: • Bit 0 = Output S7.2 [0=(PROGO1 axis n)] [1=(PROGO1 axis n+1)] • Bit 1 = Output S7.3 [0=(PROGO2 axis n)] [1=(PROGO2 axis n+1)]				
V30	General drive configuration : • Bit 0 = Wait for motor to stop to enable [1 = Wait] • Bit 1 = Braking when disabled [1 = Brake]				
V50	Torque current limit	Arms	SM	AM	LM
V58	Proportional speed loop gain high range	mA/Rpm	•	AM	•
V100	Module current limit	Reserved parameters			

1	Parameter number, description and unit. ([ ] = internal unit)
2	Parameters in light area are dedicated for NUMDrive C Bi-Axes only. These parameters are taking into account in the Axis(n) configuration only.
3	Parameter in white area are used for all drive User's application at level 1 and 2.
4	Parameter in dark-grey area are "Reserved parameter" classified at level > 2. and they never be adjusted.
5	Motor application: AM = Asynchronous motor SM = Synchronous motor LM = Linear motor, • = Not used in this motor configuration.
[0] [1]	Set parameters exchange 0 Set parameters exchange 1

1.9.2 How to read a parameters description - V50 example



Through the V50 parameter it is possible to reduce the maximum drive current in order to limit the maximum torque delivered by the motor. The value is expressed in Arms.

- Enter the Arms value of the desired torque current into parameter V50.
- Example of V50 adjustment for MDLU3021A... (15 Arms. max)
- By entering the value 10,00 in parameter V50 the effective current delivered by the system will be 10 Arms max.....(follow.....).

1	Parameter number.
2	Parameter description with parameter set [0] or [1].
3	Parameter Unit.
4	Minimum and Maximum allowable values. Where ST value is present means that such values are automatically loading by the system.
5	Default value of the parameter.
6	Utilization motor: Please see “Motor application” description on page before.
7	- HALT / Level: - HALT = where YES is required press HALT icon as indicated on chapter 2.1.2. - Level = Display the associated parameter level (1 or 2).
8	Parameter utilization.

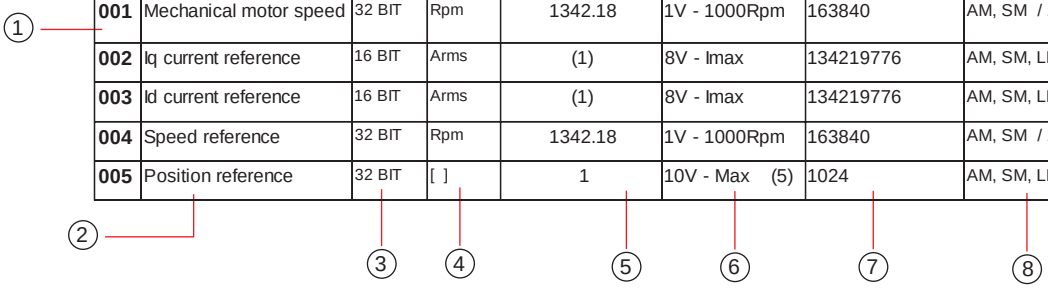
1.10 How to read the drive measure point (MP) list with WPM

All the measure points (MP) indicated and described in this manual are classified at modification level 1 and 2. Please refer to chapter 5 of the this manual to give the full measure point description and application. These measure points are typically measure points which have to be adjusted during a standard drive setup and commissioning.

All measure points classified at level superior than 2 are reserved, and they never must be adjusted. To ensure a proper use, please always read the here-below legend carefully to reduce the risk due to an impropery use of them.

1.10.1 How to read a general measure point (MP) list table

MP	Measure point description	Internal unit [IU]	Physical Unit [PU]	V143, V153 digital scale factor (DSF) [PU] = [ ] / DSF	Analog scale	V134, V135 analog scale factor (ASF)	Motor utilisation / Level
001	Mechanical motor speed	32 BIT	Rpm	1342.18	1V - 1000Rpm	163840	AM, SM / 1
002	Iq current reference	16 BIT	Arms	(1)	8V - Imax	134219776	AM, SM, LM / 1
003	Id current reference	16 BIT	Arms	(1)	8V - Imax	134219776	AM, SM, LM / 1
004	Speed reference	32 BIT	Rpm	1342.18	1V - 1000Rpm	163840	AM, SM / 1
005	Position reference	32 BIT	[ ]	1	10V - Max (5)	1024	AM, SM, LM / 1



1	MP number used for MP selection.
2	MP description
3	MP internal unit 16, 32 bits
4	MP physical unit ([ ] = internal unit)
5	V143, V153 digital scale factor for drive programmable relay management. (See V143, V153 parametes on chapter 9 of the manual)
6	Analog scale conversion output. (.) See note xx.
7	V134, V135 analog scale for multiplication factor for TP1, TP2. (See V134, V135 parametes on chapter 9 of the manual)
8	Motor application means: - AM = Asynchronous motor - SM = Synchronous motor - LM = Linear motor. - 1 = MP level of selection.

1.11 Document revisions

Code	Edition	Evolution Index	Software revision compliance	Meaning of the major documentation evolution points	See chapter
M00012EN	3.2010	00	3.2x, 4.2x	First version of the Flexium CANopen Axes Programming and parameters manual.	-
M00012EN	2.2017	01	3.5x, 4.5x	Second version of the Flexium CANopen Axes Programming and parameters manual.	-

1.12 Glossaries - Abbreviations

Abbreviations	Meaning
AM	Asynchronous motor
AMOMAN012	NUMDrive C Installation and Maintenance manual
AMS100, 132, 160	NUM Drive spindle motor range
APxxx	Special application
AUX	Auxiliary power supply
Bi-Axes	Drive for two motors control
BHX, BPH, BPG, BPL, BHL260	NUM axis motor range
BP	Drive Basic Performance version
Broadcast	Type of data transmission in the network. One device sends a message to all devices on the network
C	Configurable alarm
CAN	Controller Area Network standardized open Fieldbus
CANopen	Device and manufacturer-independent description language for communication in the CAN bus
ccd	Command code, part of a SDO message
CiA	CAN in Automation, CAN interest group, sets standards for CAN and CANopen
CNC / Controller	Control unit
COB	Communication OBject, transport unit in a CAN network
COB-ID	(Communication OBject-Identifier) uniquely identifies every communi-cations object in a CAN network
Control Unit	NUMDrive C control unit module
Crimping tool	Tool for connector assembly
CS	Command Specifier: Identifier for identification of LSS services
DAC	Digital - Analog Converter
Default value	Factory settings
DISC NT	Digital bus communication type
DSP	CiA Draft Standard Proposal (DSP)
DS 301	Standardises the CANopen communications profile
DSP 402	Standardises the CANopen device profile for drives and positioning controls
EMC	Elettromagnetic Compatibility
EMCY object	Emergency object: Object for rapid transmission of error messages in the network
EnDat	Type of encoder
Error class	Classification of operational faults into groups corresponding to the error responses
EU	European Union
IEC65	Signal level standard
Fieldbus	A bus optimised for data transmission between field devices. A Fieldbus is “open”, meaning that it is not proprietary (not supported by only one manufacturer).
Flexium CANopen Axes	MDLU3 ... NUMDrive C Mono-Axis and Bi-Axes digital drives module (Power + Control Units) with CANopen interface
Flexium Tools (FX Tools)	Commissioning tool for Flexium systems
Heartbeat	Used for unconfirmed connection message from network devices
High-End cables	Optional NUM power / sensor cables
HOMING	CNC procedure
HP	Drive High Performance version
LED	Light Emitting Diode
LM	Linear motor
Low-End cables	Optional NUM power cables
Limit switch	Switch that signals an overrun of the permissible travel range
LSS	Layer Setting Services

M	Motor
MASTER	NUMDrive C module for TANDEM application
MDLC1010N00	Capacitor module
MDLL3, MDLL2	Power Supply module
MDLU3...N	NUM digital drive
MDLQ3001N00	Auxiliary 250W power supply
Mono-Axis	Drive for one motor control
MP	Measure Point (... associated to TP1 or TP2)
MSA, MSS	Num Drive Motorspindle range
NC	Not Configurable alarm
NMT	Network management service
NUMDrive C	MDLU3 ... Mono-Axis and Bi-Axes digital drives module (Power + Control Units) with CANopen or DISC NT digital bus
NUM-STO	Safe Torque Off certified SIL 2 module
SAM-Num	NUM-Safety certified SIL 2 module
SM	Synchronous motor
PC	Personal computer
PDO	Process Data Object
PDS	Power Drive System
PLC	Programmable Logic Controller
PTC-NTC	Motor thermal probe
Power Unit	NUMDrive C power supply module
RTR	Remote Transmission Request
RJ11/ RJ45	Service Data Object
SYNC object	Synchronisation object, object for synchronisation of users in the network
SDO	Service Data Object
size x	Module width (50mm, 100mm, 200mm...)
S1, S2	Sensor connectors for NUM HP drive (equal to X1, X2, X3 or X4 for NUMDrive C)
STOP function Category x	See standard EN60204-1
SLAVE	NUMDrive C module for TANDEM application
SuB HD	Multipolar connector
TANDEM	Anti-backlash and torque duplication applications
TN, TT	Distribution systems
T_PDO	Transmit PDO
TP1, TP2	Analog test point
X1, X2, X3, X4	NUMDrive C sensor connectors
WPM	Commissioning tool for NUM Service only

1.13 NUM Agency

The list of NUM agencies is given click here http://www.num.com

Page deliberately empty.

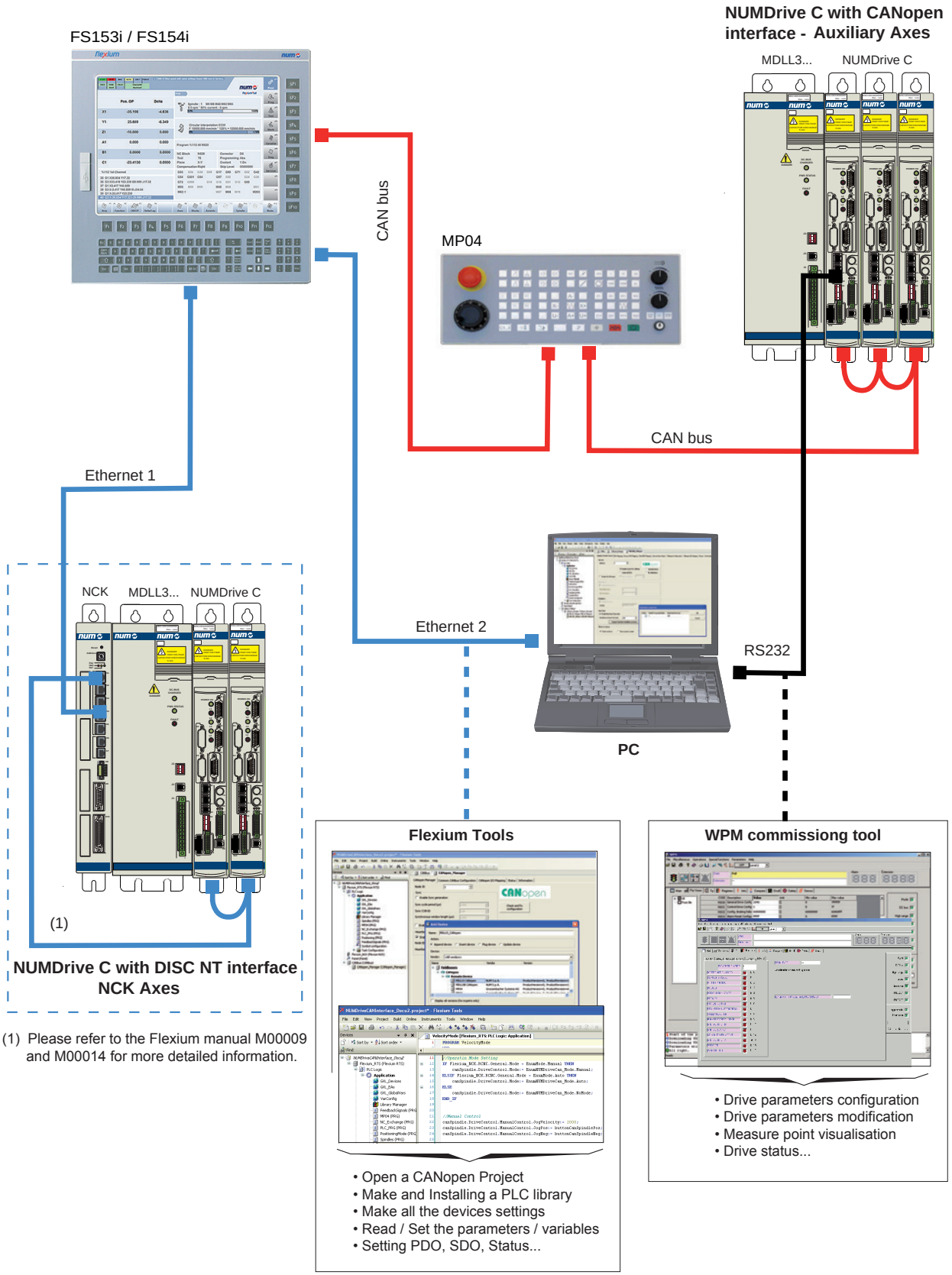
Summary

2 CANopen and system overview 31
2.1 System overview 31
2.2 CANopen interface 32
2.3 CANopen communications layers 33
2.3.1 CAN Physical Layer 33
2.3.2 CAN Data Link Layer 33
2.3.3 CANopen Application Layer 33
2.4 CANopen objects 34
2.4.1 Object directory 34
2.4.2 Object index 34
2.4.3 Object group data types and profiles groups 34
2.5 CANopen profiles 35
2.5.1 Standardised profiles 35
2.5.2 DS 301 communications profile 35
2.5.3 DSP 402 device profile 36
2.5.4 Manufacturer-specific profiles 36
2.6 Field bus devices in CAN bus 37
2.7 Operating modes and functions in field bus operation 37
2.7.1 Flexium CANopen library 37
2.8 Communications profile 38
2.8.1 Object directory 38
2.8.2 Object dictionary structure 38
2.8.3 Object description 39
2.8.4 Standardised objects 39
2.9 Communications objects 40
2.9.1 CANopen message (COD-ID and COB IDs) 40
2.10 Service data communication 43
2.10.1 Overview 43
2.10.2 SDO data exchange 43
2.10.3 SDO message 44
2.10.4 Reading and writing data 45
2.11 Process data communication 49
2.11.1 Overview 49
2.11.2 PDO data exchange 49
2.11.3 PDO message 50
2.11.4 PDO addresses 53
2.11.5 PDO settings 54
2.11.6 PDO mapping 56
2.11.7 PDO transmission type 57
2.12 Emergency service 58
2.12.1 Error evaluation and handling 59
2.13 Network management services 62
2.13.1 NMT services for device control 63
2.13.2 Services for connection monitoring 64
2.13.3 Node/life guarding 65
2.13.4 Heartbeat 67
2.14 Node address 68
2.14.1 Setting the NMT node address by dip-switches 68
2.15 Transfert rate by V01 parameter 68

Page deliberately empty.

2 CANopen and system overview

2.1 System overview



(1) Please refer to the Flexium manual M00009 and M00014 for more detailed information.

2.2 CANopen interface

CANopen interface implements the CAN bus (CAN:Controller Area Network) technology as for economical data transmission.
In fact the CAN bus is also used in industrial automation technology and has been further developed for communication at fieldbus level.

This interface is a standardised open fieldbus, through which is possible to associate devices coming from different manufacturers and communicate with each other.

CANopen is a device “manufacturer-independent” description language for communication on the CAN bus. CANopen offers a unified base for exchanging commands and data between CAN bus devices.

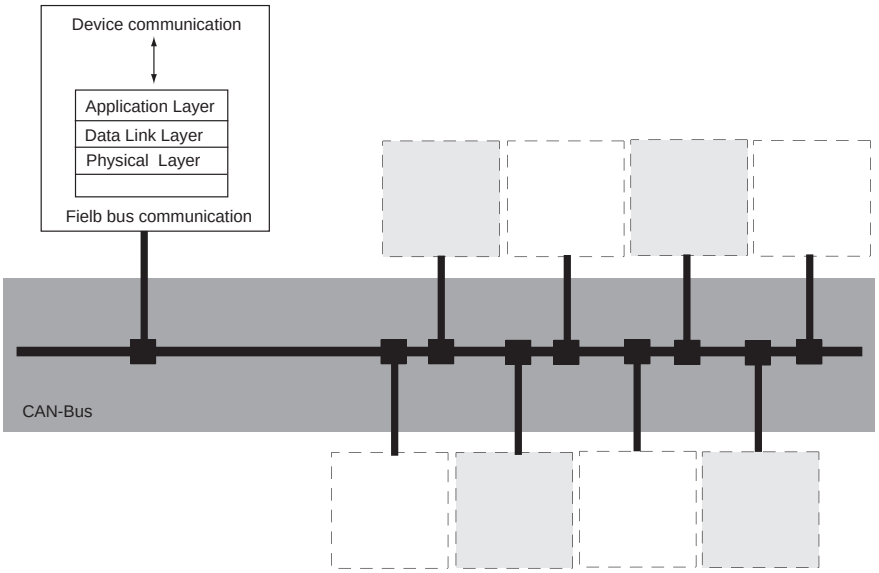
Mainly the features of the CANopen interface are:

- Multimaster capacity.
- Each device connected in the fieldbus can send and receive data independently without being assigned to an “ordering” master function.
- Devices can be linked into an existing network without requiring reconfiguration of the entire system. The address of a new device does not need to be specified on the network.
- Messages with higher priority are sent first for time-critical applications.
- Various backup processes in the network reduce the probability of an undetected, faulty data transfer.
- The transmission technology with multiple devices connections via a bus cable.
  Every network device can send and receive messages. Data between network devices are transmitted serially.

2.3 CANopen communications layers

CANopen uses the CAN bus technology for data communications.
CANopen is based on the ISO-OSI layer model on the data communications basic network service.
The communication layers secure data communications in the CAN bus are as follow:

- CAN Physical Layer
- CAN Data Link Layer
- CANopen Application Layer.



2.3.1 CAN Physical Layer

The physical layer defines the electrical properties of the CAN bus such as plug connectors, cable length and cable properties such as bit-coding and bit-timing.

2.3.2 CAN Data Link Layer

The data link layer connects the network devices. It sets the priorities of individual data packets and monitors and corrects errors.

2.3.3 CANopen Application Layer

The application layer uses communications objects (COB) to exchange data between the various devices. Communication objects are elementary components for creating a CANopen application.

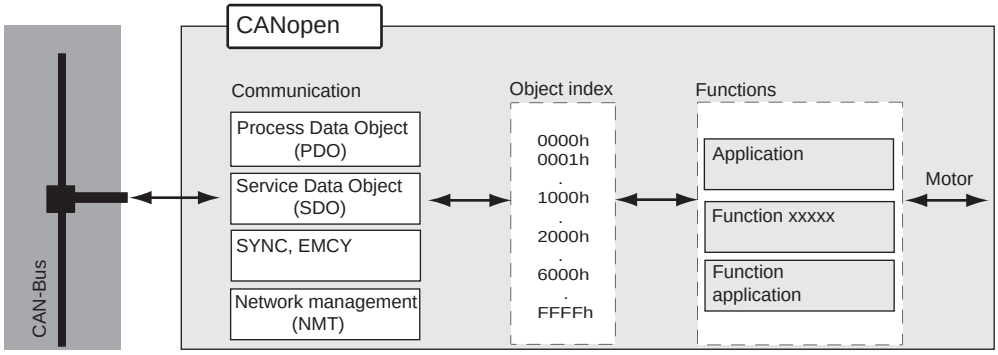
2.4 CANopen objects

All processes under CANopen are executed via objects.
The objects carry out different tasks; they act as communications objects for data transport to the fieldbus, control the process of establishing a connection or monitor the network devices.
If objects are directly connected to the device (device-specific objects), the device functions can be used and changed with device-specific objects:

2.4.1 Object directory

The central controller connection for all objects is the "**object directory"** of every network device. Other devices find all objects listed here with which they can establish a connection with the device. Please refer to the chapters 3, 4 for for the list of all objects.

Objects for describing the data types and executing the communications tasks and device functions under CANopen are registered.



2.4.2 Object index

The objects are addressed over a 16-bit index, which are displayed as a four character hexadecimal number. The objects are pooled together in groups.

INDEX	**Object groups**
0000h	Reserved
0001h -009Fh	Static and complex data types (see CiA DS301 for definitions)
00A0h-0FFFh	Communication profile, standardised in DS 301
1000h-1FFFh	Optional
2000h-5FFFh	Manufacturer-specific device profiles
6000h-9FFFh	Standardised device profiles, e.g. in DSP 402
A000h-FFFFh	Reserved

Please refer to the chapter 3 and 4 for the list of all objects.

2.4.3 Object group data types and profiles groups

The messages that move over the network as bit streams have the same meaning for sender and receiver with the data types. They are agreed via the objects of the data types.
CANopen objects carry out various tasks in field bus operation.
Profiles combine the objects in accordance with their prescribed tasks.

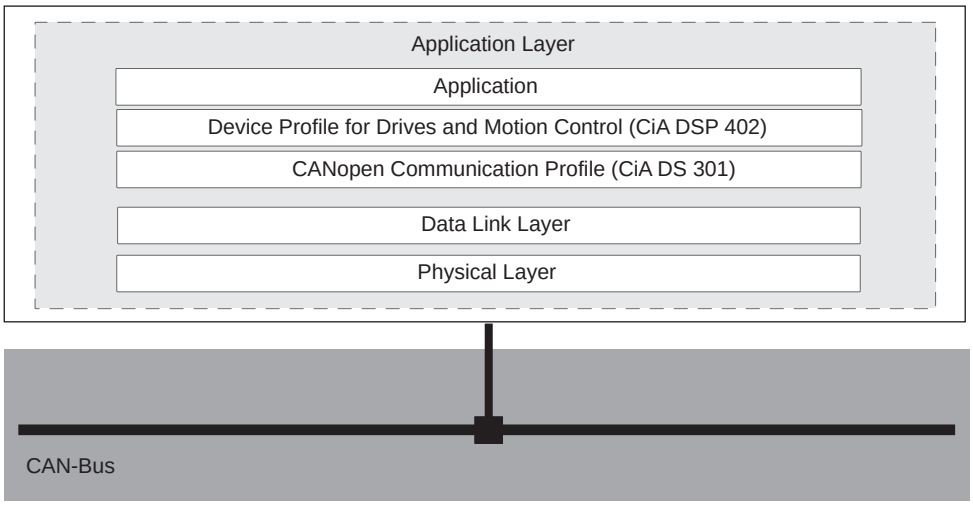
2.5 CANopen profiles

2.5.1 Standardised profiles

Standardised profiles describe objects that can be applied to various devices without additional configuration.

The association for CAN in Automation e. V. (CiA) has standardised different profiles which include the communications profile DS 301 and the device profile DSP 402.

These profiles are part of the user layer in the ISO-OSI layers model.
Additional services are available from a manufacturer-specific device profiles and applications.



2.5.2 DS 301 communications profile

The DS 301 communications profile forms the interface between device profiles and CAN bus. It was specified in 1995 under the name DS 301 and defines unified standards for common data exchange between different device types under CANopen.

The communications profile objects in the device carry out the tasks of data and parameter exchange with other network devices and initialise, control and monitor the device in the network.

The principal communications profile objects are:

- Process data object PDO
- Service data object SDO
- Objects with special functions for synchronisation SYNC and for error reporting and response EMCY
- Objects of network management NMT for initialisation, error monitoring and device status monitoring.

2.5.3 DSP 402 device profile

The DSP 402 device profile describes standardised objects for positioning, monitoring and settings of drives. The tasks of the objects are:

- Device control and status monitoring (Device Control)
- Standardised parameter setting
- Switching, verification and execution of operating modes.

2.5.4 Manufacturer-specific profiles

The basic functions of a device can be used with device profiles standardised with objects. Only manufacturer-specific device profiles offer the complete range of functions.

2.6 Field bus devices in CAN bus

Many and different field bus devices can be operated in the same field bus.
CANopen offers a unified base for exchanging commands and data between CAN bus devices.

2.7 Operating modes and functions in field bus operation

The drive operates in field bus mode with the following operating modes and functions:

- Position drive control with relative and absolute positioning
- Speed drive control
- Homing

The operating functions include
- Ramp functions
- Emergency Stop function
- Monitoring functions.

The following can be achieved over the fieldbus:
- Parameter settings for the positioning controller can be called up and changed,
- Inputs and outputs of the signal interface can be monitored
- Diagnostics and fault monitoring functions can be activated.

2.7.1 Flexium CANopen library

Please refer to the chapter 3.

2.8 Communications profile

CANopen manages communications between the network devices with object directories and objects. A network device can use process data objects (PDO) and service data objects (SDO) to request the object data from the object directory of another device and, if permissible, write back modified values.

The following can be done with the access to the objects of the network devices:
- exchange parameter values
- start movement functions of individual CAN bus devices
- query status information.

2.8.1 Object directory

Every CANopen device administers an object directory, in which all objects for communications are listed. The objects are addressed in the object directory with a 16-bit long index.

One or more 8-bit-long subindex entries to every object point to individual data fields in the object. Index and subindex are shown in hexadecimal characters, recognisable by the attached “h”.

Example
The following table shows index and subindex entries with the example of the object software position limit (607Dh) for identifying the position of the software limit switch.

Index	Subindex	Name	Meaning
607Dh	00h	-	Number of data fields
607Dh	01h	Min. position limit	Bottom limit value switch
607Dh	02h	Max. position limit	Top limit value switch

2.8.2 Object dictionary structure

Index range

Index range (hex)	Object groups
0000h	Reserved
0001h - 001Fh	Static data types
0020h -003Fh	Complex data types
0040h -005Fh	Manufacturer-specific data types
0060h -007Fh	Status data types for the device profiles
0080h -009Fh	Complex data types for the device profiles
00A0h-0FFFh	Reserved
1000h-1FFFh	Communications profile
2000h-5FFFh	Manufacturer-specific objects (1)
6000h-9FFFh	Standardised device profiles (1)
A000h-FFFFh	Reserved

2.8.3 Object description

The objects of the following object groups are described differently for CANopen programming with a positioning controller:

- 1xxxh objects: Communications objects on following pages.
- 6xxxh objects: Standardised objects of the device profile in chapter 5.

Example object description - Object 1000h: Device Type

The device shows the implemented device profile and the device type.
The lower 16 bit contain the device profile number and the upper 16 bit an additional information.

Object description	
INDEX	1000
Name	Device Type
Object Code	Variable
Data Type	UNSIGNED32
Category	Mandatory

Entry description	
Access	CONST
PDO Mapping	No
Default Value	0x0FFF0192
Lower Limit	0x00000000
Upper Limit	0xFFFFFFFF
Unit	-

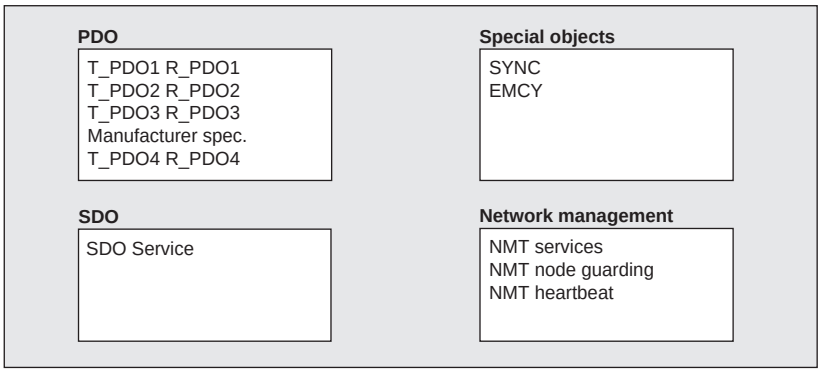
2.8.4 Standardised objects

Standardised objects form the basis of applying the same applications for the various network devices of a device type. This requires the devices to list the objects in their directory.

Standardised objects are defined in the DS 301 communications profile and the DSP 402 device profile.

2.9 Communications objects

The communications objects are standardised with the DS 301 CANopen communications profile. The objects can be classified into four groups according to their tasks.



- PDO (process data object) for realtime transmission of process data
- SDO (service data object) for read and write access to the object directory
- Objects for controlling CAN messages:
    - SYNC object (synchronisation object) for synchronisation of network devices
    - EMCY object (emergency object) for error display of a device or its peripheral.

- Network management services:
    - NMT services for initialisation and network control (NMT: network management)
    - NMT guarding objects for monitoring the network user.

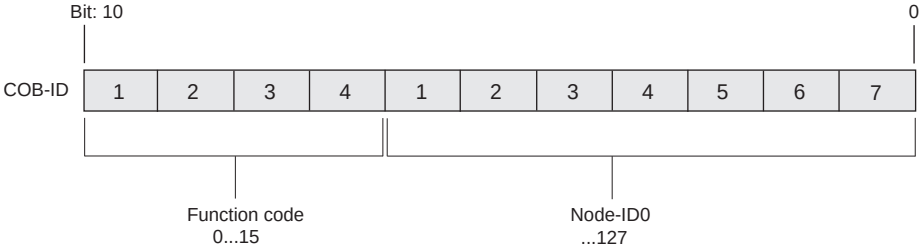
2.9.1 CANopen message (COD-ID and COB IDs)

The CAN message can be displayed in simplified form for work with CANopen objects and for data exchange, because most of the bits are used to ensure error-free data transmission.

These bits are automatically removed from the received message by the data security layer, the data link layer of the OSI layer model, and added to a message before transmission. The two bit fields “identifier” and “data” form the simplified CANopen message. The “identifier” corresponds to the “COB Id” and the “data” field corresponds to the maximum 8-byte data frame of a CANopen message.

The COB Id (Communication OBject Identifier) has two tasks in the control of communications objects:
- Bus arbitration: specification of transmission priorities
- Identification of communications objects.

An 11-bit COB identifier as per the CAN 2.0 specification is defined for CAN communications.
It comprises two parts:
- Function code, 4 bit size
- Node-ID, 7 bit size.



Function code

The function code classifies the communications objects. Because the bits of the function code in the COB Id are significantly higher, the function code simultaneously controls the transmission priorities: Objects with a small function code are sent at high priority.
For example, with simultaneous bus access an object with the function code “1” is sent before an object with the function code “3”.-

Node address

Every network device is configured before network operation. It is given a unique, 7-bit-long node address (node-Id) between 1 (01h) and 127 (7Fh). The device address “0” is reserved for “broadcast” transmissions, which are used to send the messages to all devices simultaneously.

COB IDs of the communications objects for Mono-Axis and Bi-Axes

The following table shows the COB IDs of all communications objects in the factory setting.
The column “Index of object parameters” shows the index of special objects with which the settings of the communications objects can be read or modified by SDO.
The default values for sub-index 1 of the PDO parameter (COB ID for T_PDO7, T_PDO33 und T_PDO34) are all 0x000h + node ID per default.

Communications object for Mono-Axis version

Communications object	COB ID decimal (hexadecimal)	Configuration parameters
NMT Start/Stop Service	0h (0d)	-
SYNC object	80h (128d)	1005h....1007h
EMCY object	80h (128d) + node-ID	1014h
T_PDO1	180h (384d) + node-ID (1)	1800h
R_PDO1	200h (512d) + node ID (1)	1400h
T_PDO2	280h (640d) + node ID (1)	1801h
R_PDO2	300h (768d) + node-ID (1)	1401h
T_PDO3	380h (896d) + node-ID (1)	1802h
R_PDO3	400h (1024d) + node-ID (1)	1402h
T_PDO4	480h (1152d) + node-ID (1)	1803h
R_PDO4	500h (1280d) + node-ID (1)	1403h
SDO answer	580h (1408d) + node-ID	1200h
SDO request	600h (1536d) + node ID	1200h
NMT error control	700h (1792d) + node-ID	100Ch..100Eh
(1) The COB-ID of the PDO is configurable by the user.		

Example: Selection of a COB-ID

For a device with the node address 5, the COB-ID of the communications object T PDO1 is:
180h+node-ID = 180h + 5 = 185h.

Communications object for Bi-Axes version

Communications object	COB ID decimal (hexadecimal)	Configuration parameters
NMT Start/Stop Service	0h (0d)	-
SYNC object	80h (128d)	1005h....1007h
EMCY object	80h (128d) + node-ID	1014h
T_PDO1	180h (384d) + node-ID (1)	1800h
R_PDO1	200h (512d) + node ID (1)	1400h
T_PDO2	280h (640d) + node ID (1)	1801h
R_PDO2	300h (768d) + node-ID (1)	1401h
T_PDO3	380h (896d) + node-ID (1)	1802h
R_PDO3	400h (1024d) + node-ID (1)	1402h
T_PDO4	480h (1152d) + node-ID (1)	1803h
R_PDO4	500h (1280d) + node-ID (1)	1403h
T_PDO65	1C0h (448d) + node-ID (1)	1840h
R_PDO65	240h (576d) + node ID (1)	1440h
T_PDO66	2C0h (704d) + node ID (1)	1841h
R_PDO66	340h (832d) + node-ID (1)	1441h
T_PDO67	3C0h (960d) + node-ID (1)	1842h
R_PDO67	440h (1088d) + node-ID (1)	1442h
T_PDO68	4C0h (1216d) + node-ID (1)	1843h
R_PDO68	540h (1344d) + node-ID (1)	1443h
SDO answer	580h (1408d) + node-ID	1200h
SDO request	600h (1536d) + node ID	1200h
NMT error control	700h (1792d) + node-ID	100Ch..100Eh
DBT Services (1)	2023 (7E7h), 2024 (7F8h)	
NMT Services (1)	2025 (7E9h), 2026 (7EAh)	
(1) The COB-ID of the PDO is configurable by the user.		

Data frame

The data frame of the CANopen message can hold up to 8 bytes of data. In addition to the data frame for SDOs and PDOs, special frame types are specified in the CANopen profile:

- Error data frame
- Remote data frame for requesting a message.

The data frames are described with the relevant communications objects.

2.10 Service data communication

2.10.1 Overview

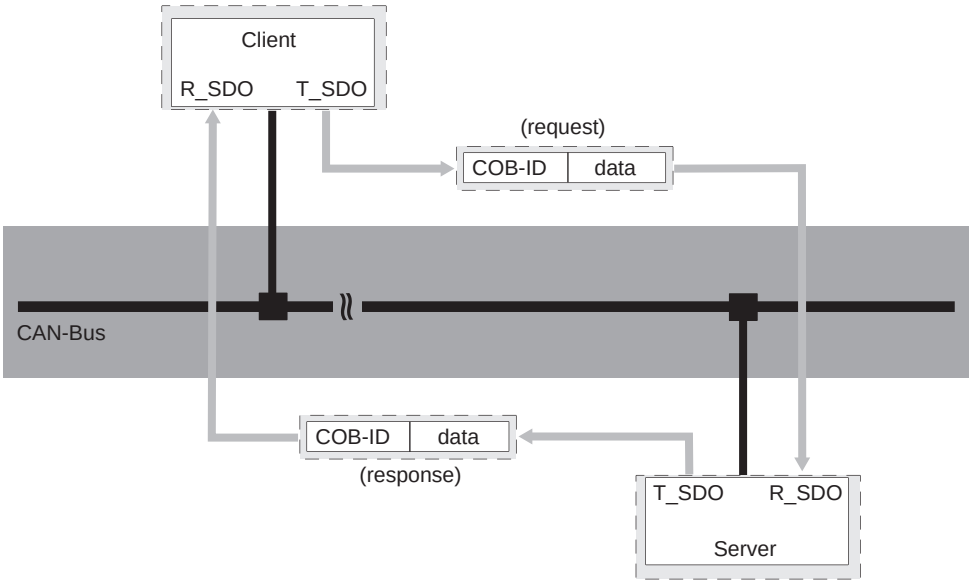
Service data objects (SDO: Service Data Object) can be used to access the entries of an object directory via index and subindex.
The values of the objects can be read and, if permissible, also be changed.

Every network device has at least one server SDO to be able to respond to read and write requests from a different device. A client SDO is only required to request SDO messages from the object directory of a different device or to change them there.

The SDO request of a SDO client can be used to send the request for data exchange, to be received with the SDO answer. The data frame of a SDO is always 8 bytes.
SDOs have a higher COB-Id than PDOs and therefore are sent over the CAN bus at a lower priority. SDO communication is for general inquiries and generate messages with more overhead than PDO, and is therefore more appropriate for the configuration stage, or to write parameters in non time-critical moment.

2.10.2 SDO data exchange

A service data object (SDO) sends parameter data between two devices.
The data exchange conforms to the client-server relationship. The server is the device to whose object directory a SDO message refers.
In our case, the SDO client is typically the motion controller (which for example is trying to configure a drive) and the SDO server is the drive itself.

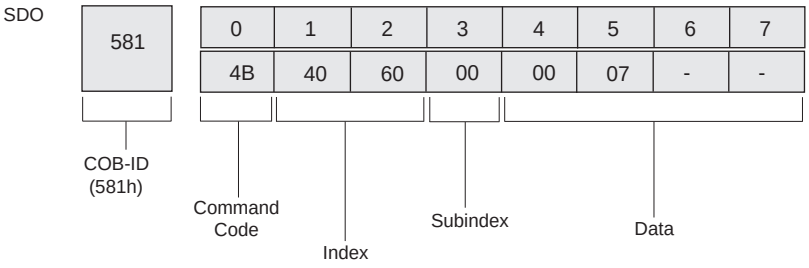


Message types
Client-server communications are triggered by the client to send parameter values to the server or to obtain them from the server. In both cases the client starts the communication with a request and receives a response from the server.

2.10.3 SDO message

A SDO message in simplified form consists of the COB-Id and the SDO data frame, in which up to four bytes of data can be sent. Longer data strings are distributed over multiple SDO messages with a special protocol.The device sends SDOs of up to 4 bytes data length (data). Larger quantities of data such as 8.byte values of the “Visible String 8” data type can be distributed over multiple SDOs and are sent successively in 7-byte blocks.

The following example diagram shows an example of a SDO message.

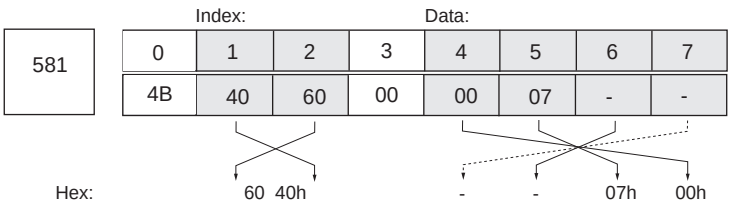


COB ID
R_SDO and T_SDO have different COB IDs, compare this to table on chapter 2.9.1.
Settings for the service data object (SDO) can be read via the objects Server SDO Parameter (1200ha and 1201h).

Data frame
The data frame of an SDO messages consists of:

- Command code in which the SDO message type and the data length of the transmitted value are encrypted
- Index and sub-index, which point to the object whose data are transported with the SDO message. In the case of an error the faulty SDO is specified itself with index and sub-index.
- Data that comprise up to 4 bytes.

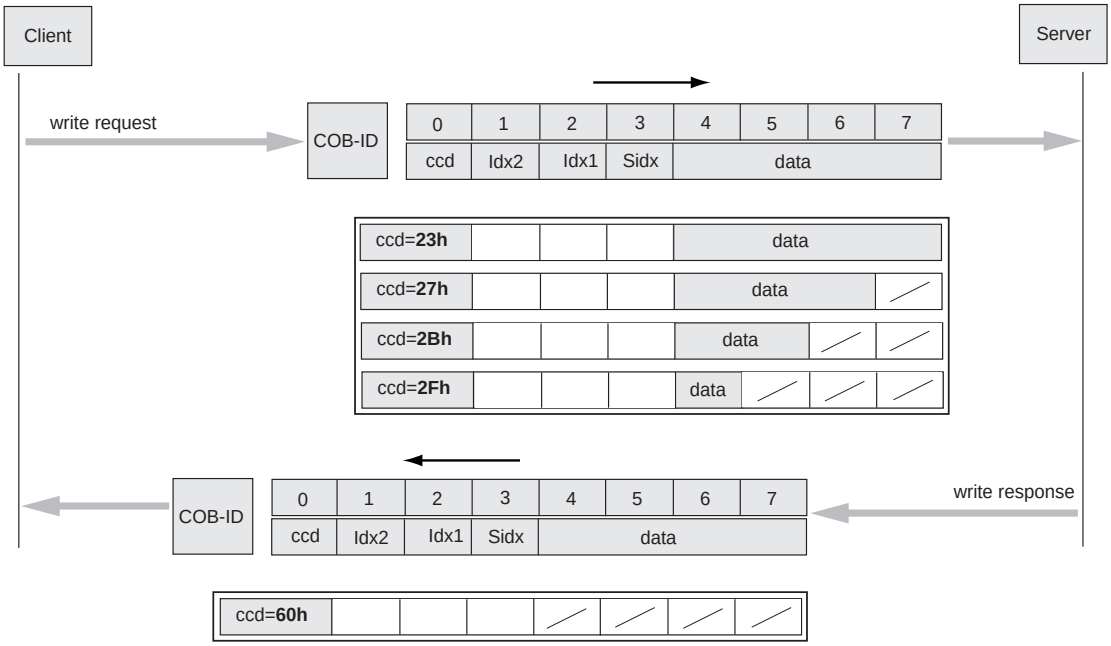
Evaluation of numeric values
Index and data are transmitted left-aligned in Intel format. If the SDO contains numerical values over 1 byte in length, the data must be converted bit-by-bit before and after a transmission.



2.10.4 Reading and writing data

Write data

The client starts a write request by sending index, subindex, data length and value. The server sends a response indicating whether the data were correctly processed.
The response contains the same index and subindex, but no data.



Unused bytes in the data field are shown with a slash in the graphic. The content is not defined.

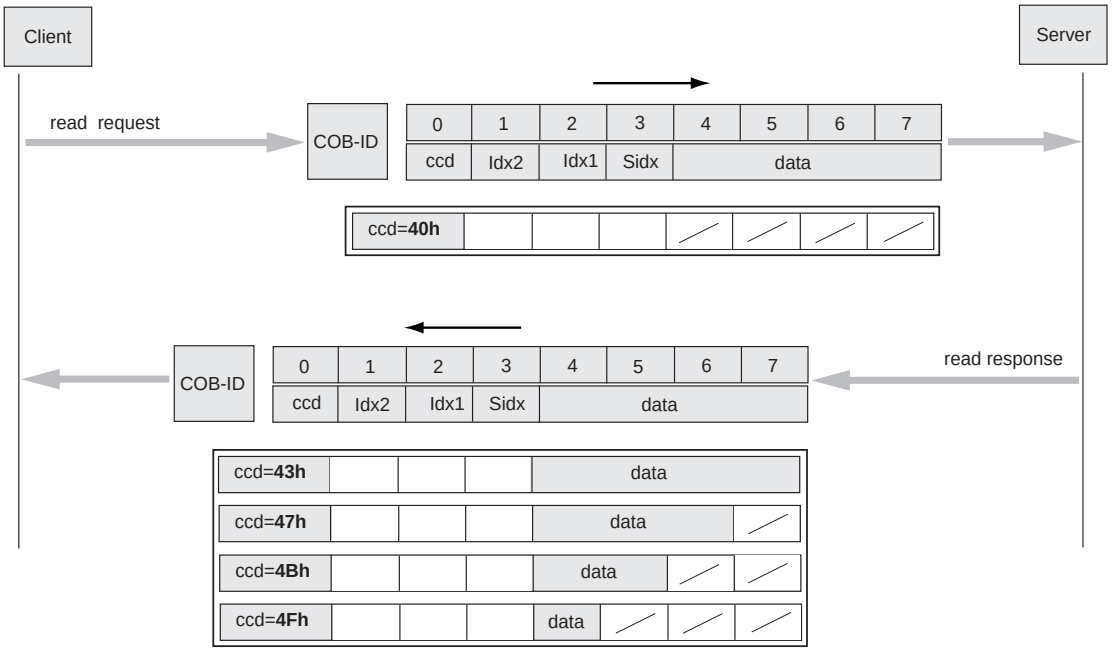
ccd-coding

The table below shows the command code for writing parameter values. It depends on the message type and the transmitted data length.

Message type	Data length used				
	4 byte	3 byte	2 byte	1 byte	
write request	23h	27h	2Bh	2Fh	Send parameters
write response	60h	60h	60h	60h	Response
error response	80h	80h	80h	80h	Error

Read data

The client starts a read request by sending index and subindex that point to the object or the object value whose value it wants to read out. The server responds to the query with the desired data.
The SDO response contains the same index and subindex. The length of the response data is specified in the command code “ccd”.



Unused bytes in the data field are shown with a slash in the graphic. The content is not defined.

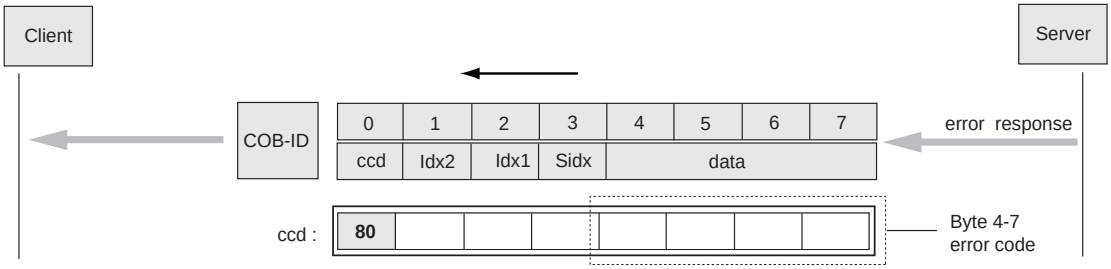
ccd-coding

The table below shows the command code for sending a read value. It depends on the message type and the transmitted data length.

Message type	**Data length used**				
	4 byte	**3 byte**	**2 byte**	**1 byte**	
read request	40h	40h	40h	40h	Request read value
read response	43h	47h	4Bh	4Fh	Return read value
error response	80h	80h	80h	80h	Error

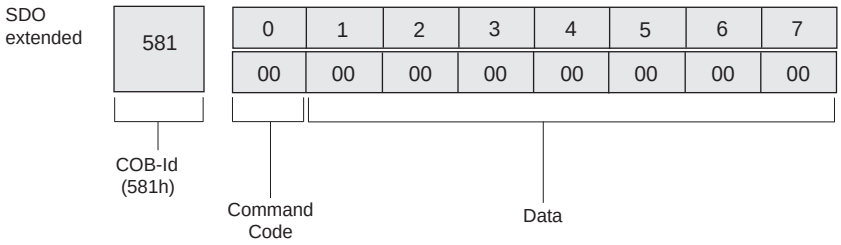
Error response

If a message could not be evaluated without errors, the server sends an error message. For details on the evaluation of the error message please refer to the table below “SDO error message “.



Abort code	Description
0503 0000h	Toggle bit not alternated
0504 0000h	SDO protocol timed out
0504 0001h	Client /Server command specifier not valid or unknown
0504 0002h	Invalid block size (block mode only)
0504 0003h	Invalid sequence number (block mode only)
0504 0004h	CRC error (block mode only)
0504 0005h	Out of memory
0601 0000h	Unsupported access to an object
0601 0001h	Attempt to read a write only object
0601 0002h	Attempt to write a read only object
0602 0000h	Object does not exsist in the object dictionary
0604 0041h	Object cannot be mapped to the PDO
0604 0042h	The number and length of the objects to be mapped would exceed PDO length
0604 0043h	General parameter incompatibility reason
0604 0047h	General internal incompatibility in the device
0606 0000h	Access failed due to AN hardware error
0607 0010h	Date type does not match, length of service parameter does not match
0607 0012h	Date type does not match, length of service parameter too high
0607 0013h	Date type does not match, length of service parameter too low
0609 0011h	Sub-index does not exist
0609 0030h	Invalid value for parameter (download only)
0609 0031h	Value of parameter written too high (download only)
0609 0032h	Value of parameter written too low (download only)
0609 0036h	Maximum value is less than minimum value
060A 0023h	Resource not available: SDO connection
0800 0000h	General error
0800 0020h	Data cannot be transferred or stored to the application
0800 0021h	Data cannot be transferred or stored to the application because of local control
0800 0022h	Data cannot be transferred or stored to the application because of the present device state
0800 0023h	Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of AN file error)
0800 0024h	No data available

Data lengths > 4 bytes
SDO messages with data lengths of more than 4 bytes, e.g. 8 bytes values of the data type “visible string 8”, are sub-divided over a number of telegrammes. The first telegramme it is constructed in exactly the same way as a SDO message > 4 byte.
On further messages the index and sub-index are not transmitted so there is a larger range available for data transmission.



The first 4 bytes are transmitted in the first SDO message. The remaining bytes are transmitted with further messages in parts of up to a maximum of 7 bytes in length.

When transmitting data > 4 bytes, there is no byte by byte conversion of the data, it is simply transmitted continuously in ASCII format.

Data frame extended SDO message
The data frame of an extended SDO messages consists of:

- Command code command code), in which the SDO message type, a so-called “toggle bit,” the data lengths of the transmitted values and a bit which indicates the end of the message, are encrypted.

  The value of the toggle bits for every telegramme of an extended SDO message must toggle between 0 and 1; the toggle bit of the first telegramme is 0, that of the second telegramme is 1, that of the third telegramme is again 0, etc.
  The bit which shows the end of the complete SDO message always has the value 0 and just for the last telegramme of the SDO message has the value 1.

- Data that comprise up to 7 bytes.

2.11 Process data communication

2.11.1 Overview

Process Data Objects (PDO) Process Data Object) are used for real-time data exchange of process data such as actual and setpoint position or operating status of the device.

The transmission can be executed very fast, because it is sent without additional administration data and does not require a response from the recipient. The flexible data length of a PDO message also increases the data throughput. A PDO message can send up to 8 bytes of data. If only 2 bytes are occupied, only 2 data bytes are sent.

The length of a PDO message and the allocation of the data fields are specified by PDO mapping. For more information see chapter 2.11.6 “PDO mapping“. PDO messages can be exchanged between devices that generate or use process data.

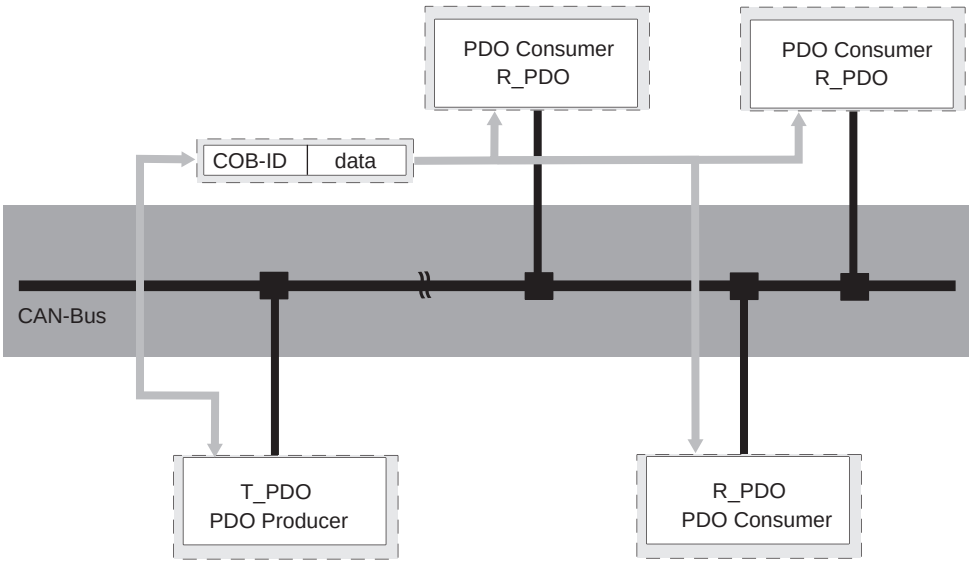
2.11.2 PDO data exchange

Data exchange with PDOs conforms to the producer-consumer relation-ship and can be triggered by three methods:

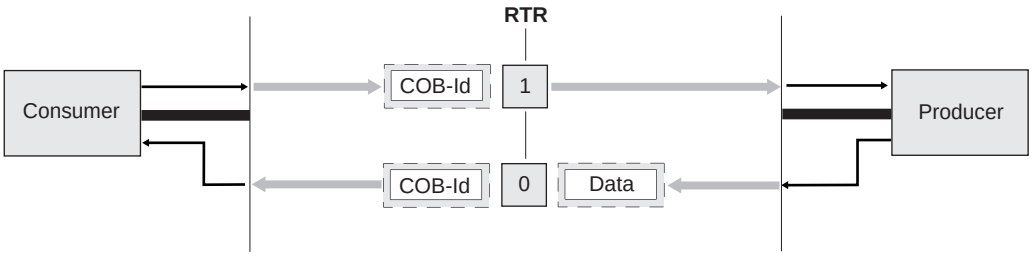
- synchronised
- event-driven, asynchronous
- by request of a consumer, asynchronous.

The synchronised data processing is controlled by the SYNC object. Synchronous PDO messages are sent immediately like the standard PDO messages, but are only evaluated on the next SYNC. For example, multiple drives can be started simultaneously by synchronised data exchange.

The device evaluates PDO messages that are called on request or are event-controlled immediately. The transmission type can be specified separately for every PDO with subindex 02h (transmission type) of the PDO communications parameter.
The objects are shown in table on chapter 2.11.5 “PDO settings”.



Requesting messages
One or more network devices with a consumer function can request PDO messages from a producer. The producer is identified by the COB ID of the request and responds with the requested PDO.



The RTR bit of a CAN message is used to detect a request (RTR:Re-mote Transmission Request). The COB ID remains the same for both messages: RTR = 0: Transmission of dataRTR = 1: Requesting data.

2.11.3 PDO message

A PDO always is available for sending and receiving a PDO message:

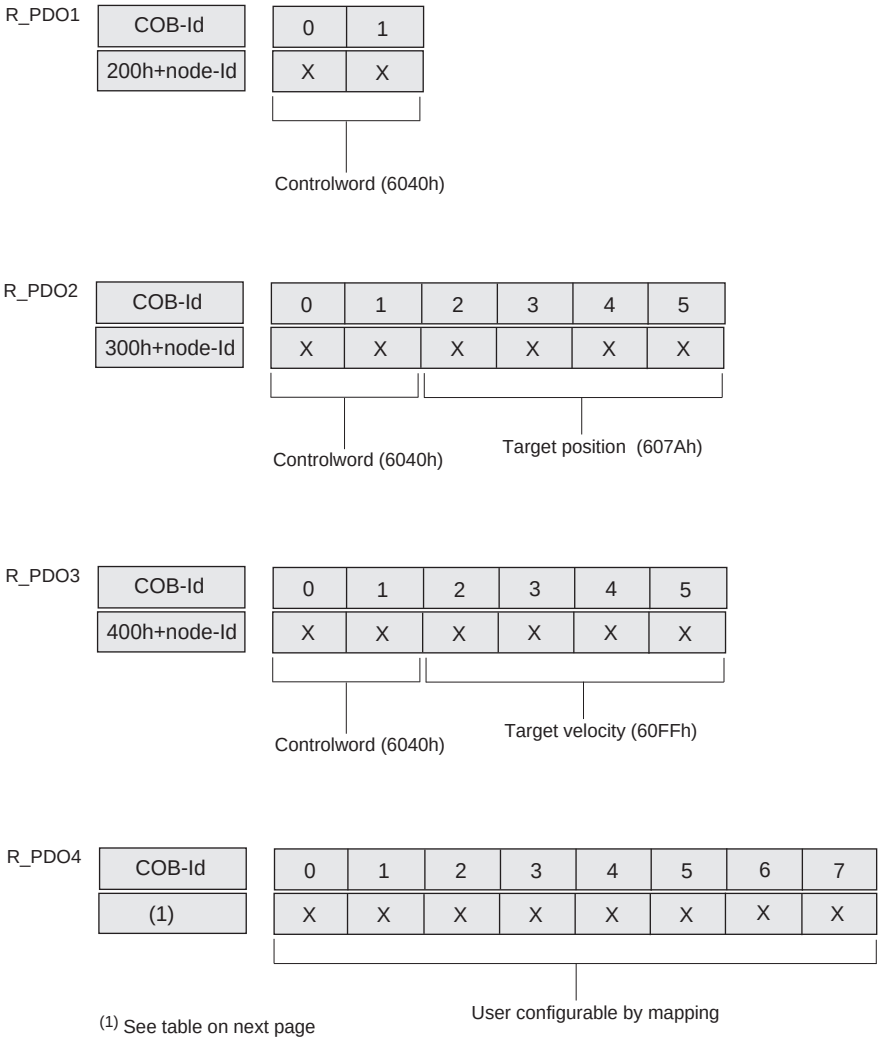
- The T_PDO for sending PDO messages (T: Transmit),
- The R_PDO for receiving PDO messages (R: Receive).

The following settings for PDOs correspond to the standard defaults for the device, unless otherwise specified. They can be read and set via objects of the communications profile.

The positioning controller uses 4 receive PDOs and 4 transmit PDOs.

Static (not changeable) object mapping applies for T/RPDO1,2,3; T/RPDO4 is user-mappable.
All PDOs are disabled by default and should be selected and enabled depending on the required application.

Receive PDOs



R_PDO1

In the first receive PDO the control word, object controlword (6040h), of the status machine is represented, which can be used to set the operating status of the device.
The communication properties for R_PDO1 are configurable. The PDO mapping is static. The default setting of R_PDO1 is asynchronous, i.e. is event-controlled. Information about the control word of the status machine can be found in chapter 4 “Drive state machine“.

R_PDO2

With the second PDO the control word and the target position of a movement command, object Target position (607Ah), is received for point-to-point positioning in the “profile position mode”.
The communication properties for R_PDO2 are configurable. The PDO mapping is static. The default setting of R_PDO2 is asynchronous, i.e. is event-controlled.

R_PDO3

In the third receive PDO are mapped the control word and the setpoint speed, object Target velocity (60FFh). This is useful for speed mode in the “profile velocity mode”. The default setting for R_PDO3 is asynchronous. The communication properties for R_PDO3 are configurable. The PDO mapping is static.

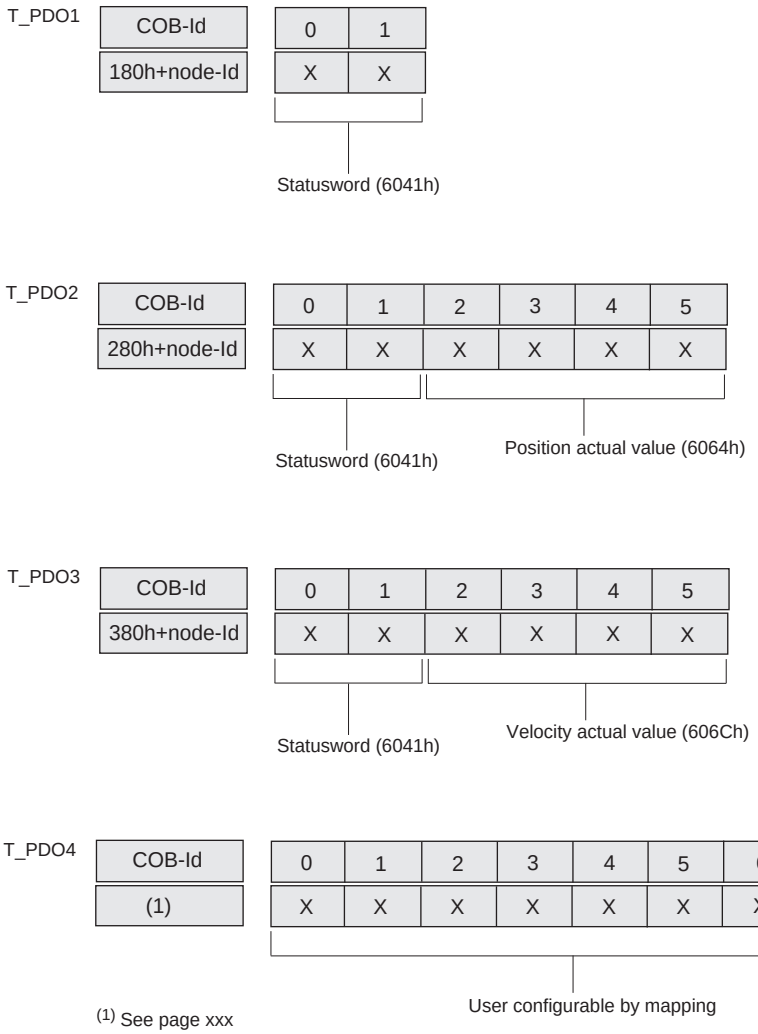
R_PDO4

The RPDO4 mapping is free for the user. Please refer to the Object dictionary description chapter 5 to understand which objects are available for mapping.

All R_PDO can be configured as synchronous or asynchronous, cyclic or acyclic.

Send PDOs

The objects for T_PDO1, T_PDO2, T_PDO3 are permanently set (static mapping). The T_PDO4 is user-mappable.



T_PDO1

In the first send PDO the status word, object statusword (6041h), of the status machine is mapped. The communication properties for T_PDO1 are configurable. The PDO mapping is static. Information about the status word of the status machine can be found in the chapter 5.

T_PDO2

In the second send PDO the status word and the current position of the motor, object Position actual value (6064h), is mapped to monitor point-to-point positioning in the profile position mode.
The communication properties for T_PDO2 are configurable. The PDO mapping is static.

T_PDO3

In the third send PDO the status word and the current speed, object Velocity actual value (606Ch), is mapped for monitoring the speed mode in profile velocity mode."
The communication properties for T_PDO3 are configurable. The PDO mapping is static.

T_PDO4

The TPDO4 mapping is free for the user. Please refer to the Object dictionary description chapter 8 and chapter 9 to understand which objects are available for mapping.

All T_PDO can be configured as synchronous or asynchronous, cyclic or acyclic.

2.11.4 PDO addresses

Every PDO has a specific COB-Id, assigned during the PDO configuration phase in the “PDO communication parameter, subindex 1” . The address can be freely selected in the available address areas, see this table:

The CiA specification suggest addresses for TPDO1..3 and RPDO1..3, and these values are reflected in default settings for NUMDC-CAN drives. Besides, we have T/RPDO4 (and T/RPDO65..68 for the biaxis). Here is a suggestion for PDO mapping:

Mono-Axis or Axis$_n$	
TPDO1	180 + addr.
TPDO2	280 + addr.
TPDO3	380 + addr.
TPDO4	480 + addr.
RPDO1	200 + addr.
RPDO2	300 + addr.
RPDO3	400 + addr.
RPDO4	500 + addr.

Axis$_{n+1}$	
TPDO65	1C0 + addr.
TPDO66	2C0 + addr.
TPDO67	3C0 + addr.
TPDO68	4C0 + addr.
RPDO65	240 + addr.
RPDO66	340 + addr.
RPDO67	440 + addr.
RPDO68	540 + addr.

This table leave room for 64 axis addresses between distinct PDOs, and can be used up to this number of axis. If the machine is composed by more than 64 modules, a careful selection of PDO addresses is required to avoid overlapping.
Anyway, in the typical case, the user enable only the PDO that’s useful for that particular application; and the base address for the required PDOs can be modified to fit the available space.

For example, if the machine is composed by 120 bi-axis modules commanded in position mode, when only T/RPDO2 and T/RPDO66 are useful, the following mapping can be used:

TPDO2	180 + addr.	(used address space: from 180h to 1F8h)
RPDO2	300 + addr.	(used address space: from 300h to 378h)

TPDO66	200 + addr.	(used address space: from 200h to 278h)
RPDO66	380 + addr.	(used address space: from 380h to 3F8h)

2.11.5 PDO settings

The settings for PDOs can be read and changed with 8+8 communication objects; the first table indicate the objects to be used for the communivìcation setting, the latter table indicate the objects used to configure the mapping (where applicable).

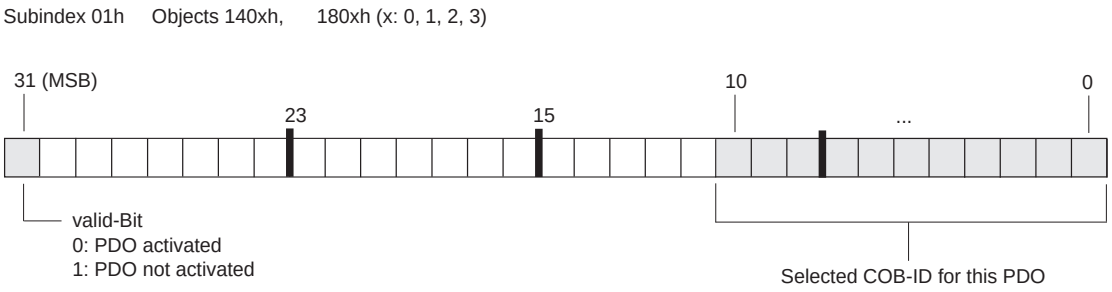
Communication parameters table

Object	Meaning
Receive PDO1 communication parameter (1400h)	R_PDO1 communication settings
Receive PDO2 communication parameter (1401h)	R_PDO2 communication settings
Receive PDO3 communication parameter (1402h)	R_PDO3 communication settings
Receive PDO4 communication parameter (1403h)	R_PDO4 communication settings
Transmit PDO1 communication parameter (1800h)	T_PDO1 communication settings
Transmit PDO2 communication parameter (1801h)	T_PDO2 communication settings
Transmit PDO3 communication parameter (1802h)	T_PDO3 communication settings
Transmit PDO4 communication parameter (1803h)	T_PDO4 communication settings
Receive PDO65 communication parameter (1440h)	R_PDO65 communication settings
Receive PDO66 communication parameter (1441h)	R_PDO66 communication settings
Receive PDO67 communication parameter (1442h)	R_PDO67 communication settings
Receive PDO68 communication parameter (1443h)	R_PDO68 communication settings
Transmit PDO65 communication parameter (1840h)	T_PDO65 communication settings
Transmit PDO66 communication parameter (1841h)	T_PDO66 communication settings
Transmit PDO67 communication parameter (1842h)	T_PDO67 communication settings
Transmit PDO68 communication parameter (1843h)	T_PDO68 communication settings
Note: The gray sections are used for Bi-Axes only.	

Mapping parameters table

Object	Meaning
Receive PDO1 mapping parameter (1600h)	R_PDO1 mapping settings
Receive PDO2 mapping parameter (1601h)	R_PDO2 mapping settings
Receive PDO3 mapping parameter (1602h)	R_PDO3 mapping settings
Receive PDO4 mapping parameter (1603h)	R_PDO4 mapping settings
Transmit PDO1 mapping parameter (1A00h)	T_PDO1 mapping settings
Transmit PDO2 mapping parameter (1A01h)	T_PDO2 mapping settings
Transmit PDO3 mapping parameter (1A02h)	T_PDO3 mapping settings
Transmit PDO4 mapping parameter (1A03h)	T_PDO4 mapping settings
Receive PDO65 mapping parameter (1640h)	R_PDO65 mapping settings
Receive PDO66 mapping parameter (1641h)	R_PDO66 mapping settings
Receive PDO67 mapping parameter (1642h)	R_PDO67 mapping settings
Receive PDO68 mapping parameter (1643h)	R_PDO68 mapping settings
Transmit PDO65 mapping parameter (1A40h)	T_PDO65 mapping settings
Transmit PDO66 mapping parameter (1A41h)	T_PDO66 mapping settings
Transmit PDO67 mapping parameter (1A42h)	T_PDO67 mapping settings
Transmit PDO68 mapping parameter (1A43h)	T_PDO68 mapping settings
Note: The gray sections are used for Bi-Axes only.	

Enable PDO
In the default setting all PDOs are disabled. The PDOs must be explicitly enabled as needed.
A PDO is enabled with bit 31 (valid bit) in subindex 01h of that communications object:



Example
Setting for R_PDO3 in object 1402h

- Subindex 01h = 8000 04xxh: R_PDO3 not enabled
- Subindex 01h = 0000 04xxh: R_PDO3 enabled.

Values for “x” in the example depend on the setting of the COB-Id.

PDO time intervals
The time intervals “inhibit time” and “event timer” can be set for every transmit PDO.

- The time interval “inhibit time” can be used to reduce the load on the CAN bus, which can be the result of continuous transmission of T_PDOs. If an interval time that is not equal to zero is entered, a sent PDO will be sent again no more frequently than this interval time.
  The time is set with subindex 03h.

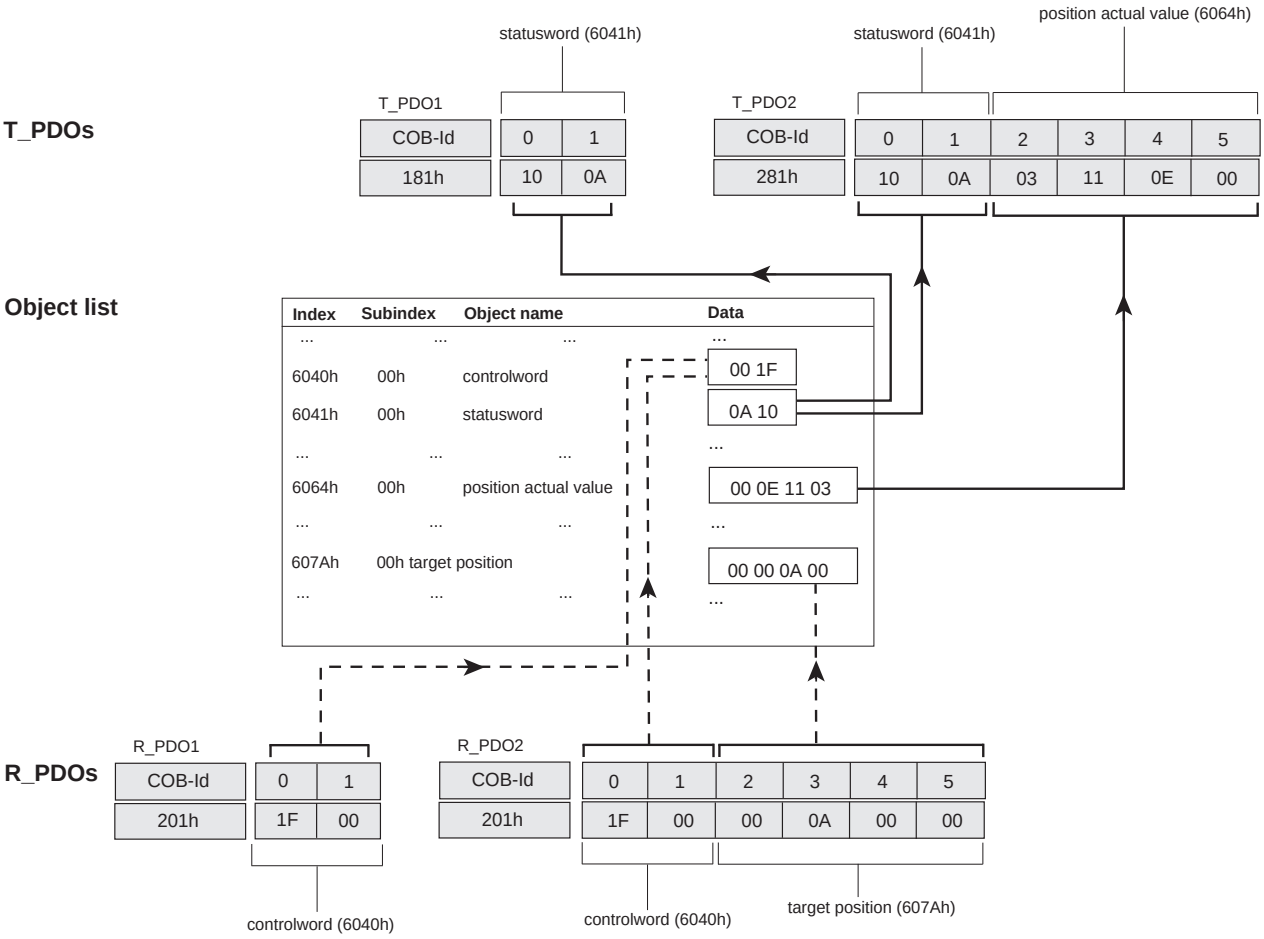
- The time interval “event timer” triggers an event message periodically. After the interval time has expired the device transmits the event-controlled T_PDO. The time is set with subindex 05h.

- The “inhibit time” is the last and most important filter before the TPDO transmission.
  In any case the PDO will be transmitted more frequently than “inhibit time” even if the “event timer” is shorter.

2.11.6 PDO mapping

Up to 8 bytes of data from different areas of the object directory can be sent with a PDO message. The mapping of data in a PDO message is referred to as PDO mapping.

Figure below shows the data exchange between PDOs and object directory with two examples of objects in T_PDO1 and R_PDO1 of the PDOs.



Static PDO mapping

The N065 positioning controller supports static PDO mapping, that is all objects are mapped in the respective PDO according to an established, unchangeable setting.
The settings for PDO mapping are defined in an assigned communications object for every PDO.

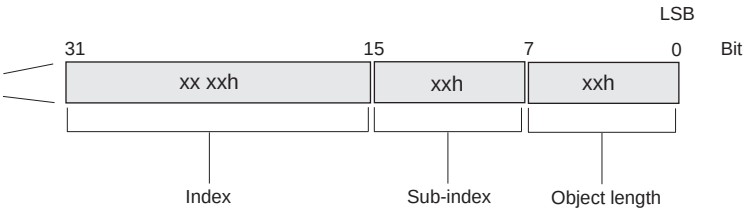
Object	Description
1st receive PDO parameter (1600h)	PDO mapping for R_PDO1
2nd receive PDO parameter (1601h)	PDO mapping for R_PDO2
3rd receive PDO parameter (1602h)	PDO mapping for R_PDO3
1st transmit PDO mapping (1A00h)	PDO mapping for T_PDO1
2nd transmit PDO mapping (1A01h)	PDO mapping for T_PDO2
3rd transmit PDO mapping (1A02h)	PDO mapping for T_PDO3
7th transmit PDO mapping (1A03h)	PDO mapping for T_PDO7
33rd transmit PDO mapping (1A20h)	PDO mapping for T_PDO33
34th transmit PDO mapping (1A21h)	PDO mapping for T_PDO34

Structure of entries

Up to 8 8-byte values of 8 different objects can be mapped in a PDO. Every communications object for setting the PDO mapping provides 4 sub-index entries. A sub-index entry contains 3 pieces of information on the object: the index, the sub-index and the number of bits that the object occupies in the PDO.

PDO-Mapping-Object

00h	2		
01h	6041h	00h	10h
02h	606Ch	00h	20h
03h	...	...	...
...			
40h	...	...	...



The number of valid sub-index entries is contained in sub-index 00h of the communications object.

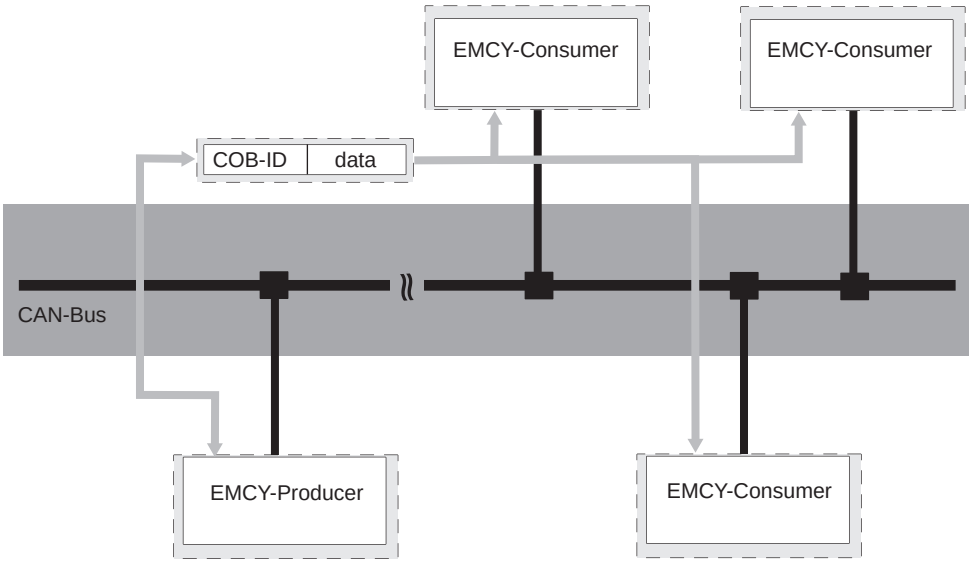
2.11.7 PDO transmission type

For “TPDO communication parameter” object, the subindex 02 define the “transmission type”. The following transmission types are foreseen by CiA specifications and supported by NUMDC-CAN:

0	synchronous, acyclic
1h..F0h	synchronous, every 1..F0h SYNC message
F1h..FBh	reserved
FCh	RTR-only, synchronous
FDh	RTR-only, asynchronous
FEh	event-driven (the firing event is event timer)
FFh	event-driven (the firing event is event timer OR transported value change)

2.12 Emergency service

The emergency service reports internal device error over the CAN bus.
The error message is sent to all devices with an EMCY object in accordance with the consumer-producer relationship.



Boot Up message
The communications profile DS 301, version 3.0, defines an additional task for the EMCY object: sending a boot-up message. A boot-up message informs all network devices that the device that sent the message is ready for operation in the CAN network.

A boot-up message consists of the COB ID of the EMCY object and is sent without data.
Standard setting of the COB ID is 128 (80h).

2.12.1 Error evaluation and handling

The occurrence of an internal alarm reflects in many actions, as defined by CANopen specifications. In fact, the same information is duplicated in many points.
The elements involved are:

- "Error Register" object 1001h
- "Pre-defined error field" object 1003h
- "Error code" object 603Fh
- EMCY message transmitted by the drive
- "FAULT" flag in the statusword.

Error Register object 1001h
Object 1001h is a bitfield (1-byte wide). NUMDC_CAN only support bit 0 (generic error), which is set when the drive is in FAULT state and cleared otherwise.

Error Register structure

Bit	M / O	Meaning
0	M	Generic error
1	O	Current
2	O	Voltage
3	O	Temperature
4	O	Communication error (overrun, error state)
5	O	Device profile specific
6	O	Reserved (always 0h)
7	O	Manufacturer-specific
M = Mandatory. O = Optional.		

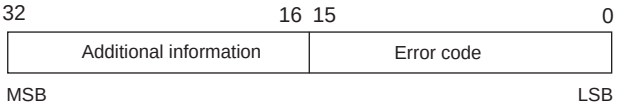
Pre-defined error field object 1003h
Object 1003h holds the number of alarms and the alarm codes.

Subindex 1003h: 0 is the number of alarms for the whole device (not for the single axis); this means that it can be 1 on a faulty Mono-Axis, or on a Bi-Axes when only one axis is in alarm, but it's 2 on a Bi-Axes with both drives in fault.

Subindex 1003h: 1..2 holds the alarm codes, the 1003h:1 being the most recent. When the 2nd alarm occur, the value present in 1003h:1 is moved to 1003h:2 and the most recent is written to 1003h:1.

This is what CANopen's specs write about 1003h:1..n:
The error numbers are of type UNSIGNED32 and are composed of a 16-biterror code and a 16-bit additional error information field, which is manufacturer-specific.
The error code shall be contained in the lower 2 bytes (LSB) and the additional information shall be included in the upper 2 bytes (MSB). If this object is supported it shall consist of two object entries at least. The length entry on sub-index 00h and at least one error entry at sub-index 01h.

Pre-defined error field structure



The contents of 1003h1..n is a 32-bit word.
The "error code" field (the LSW) is the same value sent in bytes 0..1 of the EMCY message and saved in 603Fh/683Fh.
Most values for "error code" are predefined by CANopen specification and linked to typical drives alarms (overvoltage, overcurrent, ...).

There is a “free” section in the range 0xFF00 till to 0xFFFF, and NUMDC_CAN drive use this range to code internal alarms.
“Additional information” is manufacturer-specific. We use this field to code the alarm extension as well as the axis (N, N+1) indication.
It should be clear that object 1003h is at “drive-level”, not “axis-level”, so NUMDC_CAN embed the axis information in the manufacturer-specific “Additional information” field.

To summarize, this is the way an alarm is coded in 1003h:1..n:
- (NUM alarm code) OR (0xFF00) is saved in “Error code”;
- (NUM “alarm extension” << 4) OR (1 if axis=N+1) is saved in “Additional information”.

EMCY message
The emergency EMCY service signals internal fatal device errors. There are predefined error types, described in the communication and device profiles. NUMDC_CAN only use manufacturer-specific errorcodes. The EMCY is transmitted with highest priority and consist of 8 bytes (bytes 3..7 are manufacturer-specific):

- Byte 0, 1: Emergency error code; this value is also saved in object Error code 603F h and in the LSW of 1003h:1..n
- Byte 2: Error register; this value is also saved in the object Error register 1001 h
- Byte 3, 4: Manufacturer specific error code extension;
- Byte 5, 6: unused (transmitted as 0).
- Byte 7: unused (transmitted as 0).

It should be clear that EMCY is a message at “drive-level”, not “axis-level”, so NUMDC_CAN embed the axis information in the manufacturer-specific part of the message.
The EMCY is transmitted at every alarm. When, after an alarm reset, all alarms for the drive are cleared, an EMCY message with all 0’s is emitted.

Device profile specific “Error code” object 603Fh/683Fh.
The Error code 603F h captures the code of the last error that occurred in the drive; it’s the same value stored in the LSW of 1003h:1..n.
NUMDC_CAN use (manufacturer-specific) values in the range FF00h..FFFFh.
Being in the 6xxxh section there is one of these objects for every axis.

Statusword
In the Statusword 6041 h there is the FAULT bit (bit) that becomes 1 when an alarm occur.

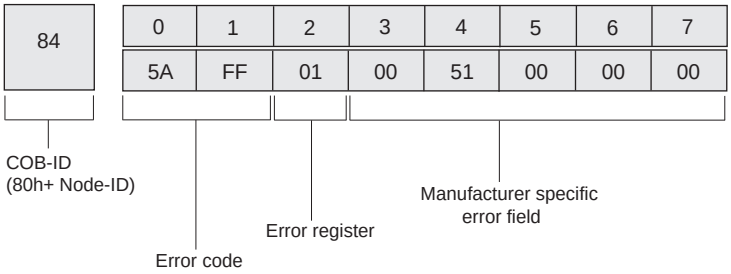
NUMDC_CAN specific implementation of error signalling
Let’s suppose that alarm A03.000 happen on axis N+1, when both axis were running.
This is what happen:
- The “generic alarm” bit will be set in 1001h:0
- 1003h:0 is incremented to 1
- 1003h:1 become 0x0001’FF03
- 683Fh become 0xFF03
- This EMCY is emitted: 03 FF 01 00 01 00 00 00
- The FAULT bit in Statusword 6841h is set.

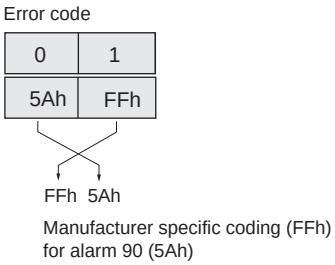
Then the alarm A09.254 happen on axis N:
- 1003h:0 is incremented to 2
- 1003h:1 become 0x0FE0’FF09
- 1003h:2 become 0x0001’FF03 (this is the old value for 1003h:1 shifted down)
- 603Fh become 0xFF09
- This EMCY is emitted: 09 FF 01 0F E0 00 00 00
- The FAULT bit in Statusword 6041h is set.

The alarm code, extension can be retrieved by the EMCY message this way:

Alarm code	Byte 0
Alarm extension	((byte3 << 8) OR (byte4)) >> 4
Axis involved (N or N+1)	(byte4 AND 0x01)

Example: Alarm A90.5 for $Axis_{n+1}$ at address 4





- Byte 0, 1 - error code: Error code, value is also stored in the object Error code (603Fh)
- Byte 2 - error register: Error register, value is also stored in the object Error register(1001h)
- Byte 3-7, manufacturer-specific error field.

COB-ID for EMCY

The COB-Id is calculated from the node address for every device in the network that supports an EMCY object:

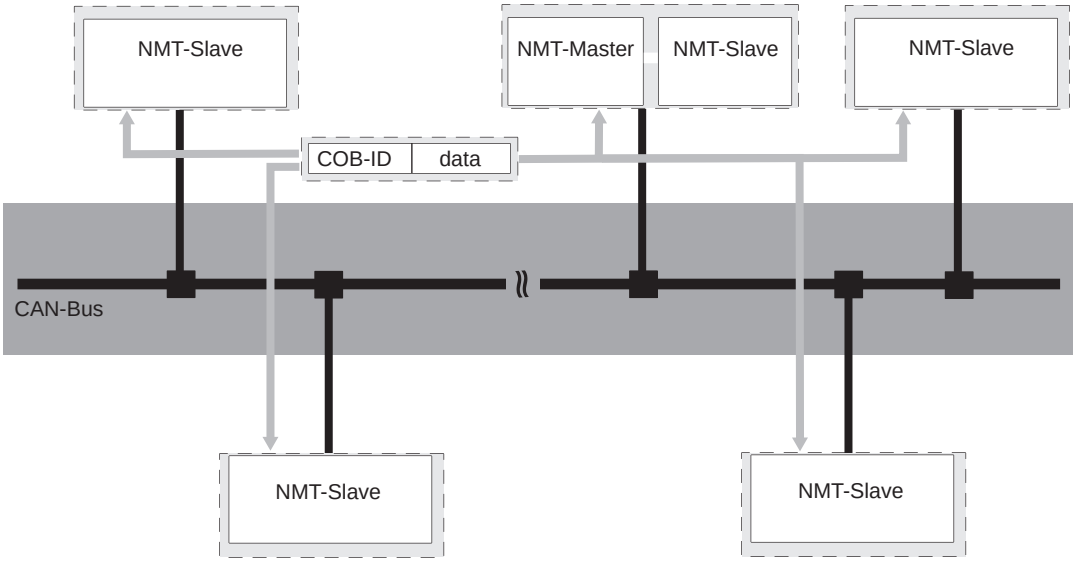
- COB-Id = function code EMCY object (80h) + node-Id.

The function code of the COB-Id can be changed with the object COB-ID emergency(1014h).

2.13 Network management services

Network management (NMT) is a component of the CANopen communications profile and is used to initialise the network and start, stop and monitor the network devices in network mode. NMT services are executed in a master-slave relationship.

The NMT master addresses individual NMT slaves through their node address. A message with node address “0” is directed to all NMT slaves simultaneously.



The device can only take on the function of a NMT slave.

NMT services
NMT services can be divided into two groups:

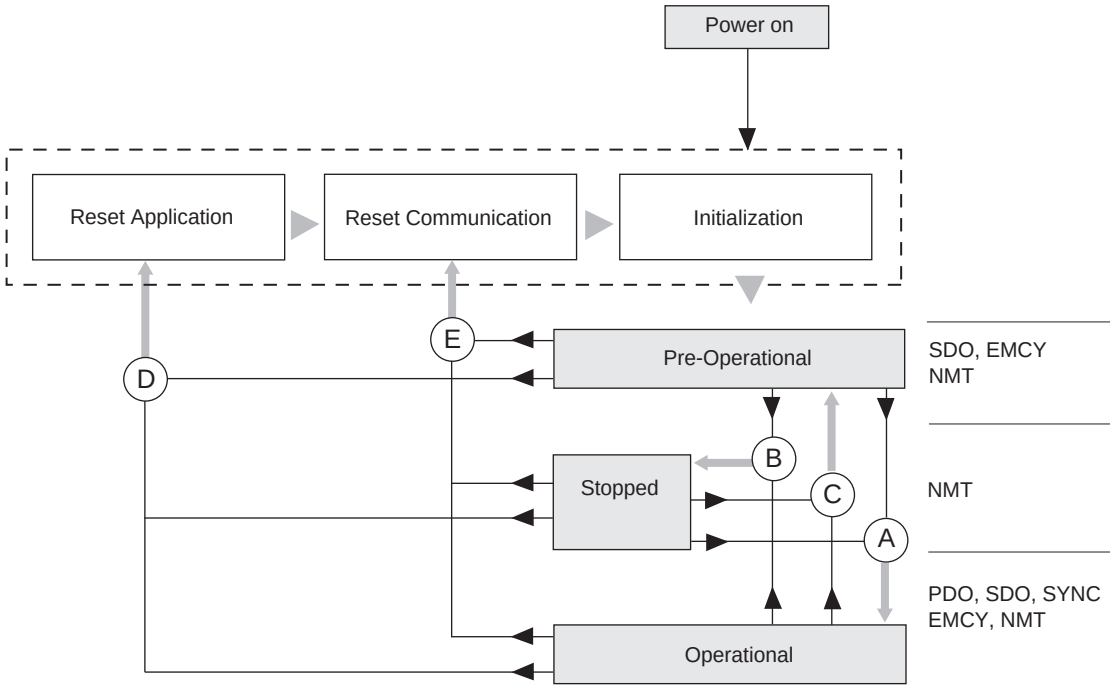
- Services for device control, to initialise devices for CANopen communications and to control the behaviour of devices in network operation

- Services for connection monitoring, to ensure error-free network operation

2.13.1 NMT services for device control

NMT status machine

The NMT status machine describes the initialising and status of an NMT slave in mains operation.



The graphic shows on the right side all communications objects that can be used in the specific network status.

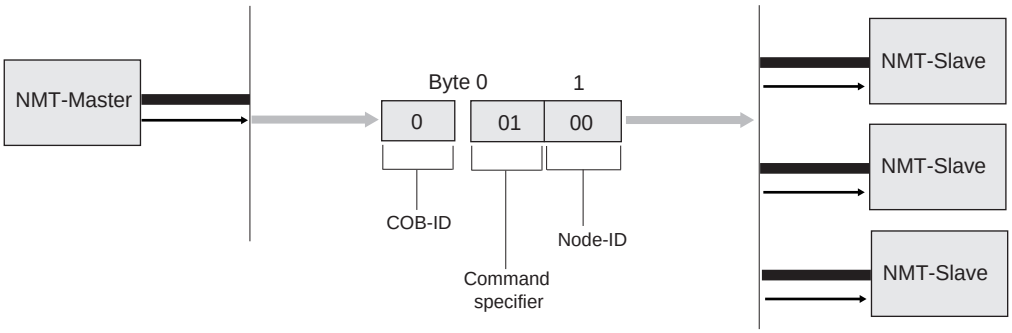
Initialisation

An NMT slave automatically runs through an initialisation phase after the supply voltage is switched on (power on) to prepare it for CAN bus operation.

After completing initialisation, the slave switches over into the “pre-operational” status and sends the EMCY object a boot-up message without data. Now a NMT master can control the operational behaviour of an NMT slave in the network with 5 NMT services, shown in the above graph with the letters A to E.

NMT service	Transition	Description
Start remote node (Start remote node)	A	Switches over into “operational status” Start normal mains operation to all users.
Stop remote node (Stop remote node)	B	Switches over into the “status stopped” Stops communication of the users in the network. If connection monitoring is activated, it remains switched on.
Enter Pre-Operational (change to the status “pre-operational”)	C	Switches to “pre-operational status”, all communications objects except for PDOs can be used. The pre-operational status can be used for configuration by SDOs: - PDO mapping - start of synchronisation - start of connection monitoring.
Reset node (reset nodes)	D	Switch to “”reset application” status “ Load saved data about the device profi-les and switch automatically to “pre-ope-rational” via the “”reset communication” status”.
Reset communication (reset communications data)	E	Switch to “”reset communication” status”. “Load stored communication profile data and switch automatically to “pre-operational” status.”

NMT message
The NMT services for device control are sent as unconfirmed message with the COB-Id = 0. By default they receive top priority on the CAN bus.
The data frame of the NMT device service consists of 2 bytes.



The first byte, the “command specifier” identifies the NMT service in use.

Command Specifier	NMT service	Transition
1 (01h)	Start remote node	A
2 (02h)	Stop remote node	B
128 (80h)	Enter Pre-Operational	C
129 (81h)	Reset node	D
130 (82h)	Reset communication	E

The second byte addresses the receiver of a NMT message with a node address between 1 and 127 (7Fh). A message with the node address “0” is directed to all NMT slaves.

2.13.2 Services for connection monitoring

Connection monitoring monitors the communications status of network devices, so a response to the failure of a device or an interruption in the network is possible. Three NMT services for connection monitoring are available:

- ”Node guarding” for monitoring the connection of a NMT slave
- ”Life guarding” for monitoring the connection of a NMT master
- ”Heartbeat” for the unconfirmed connection message from network devices.

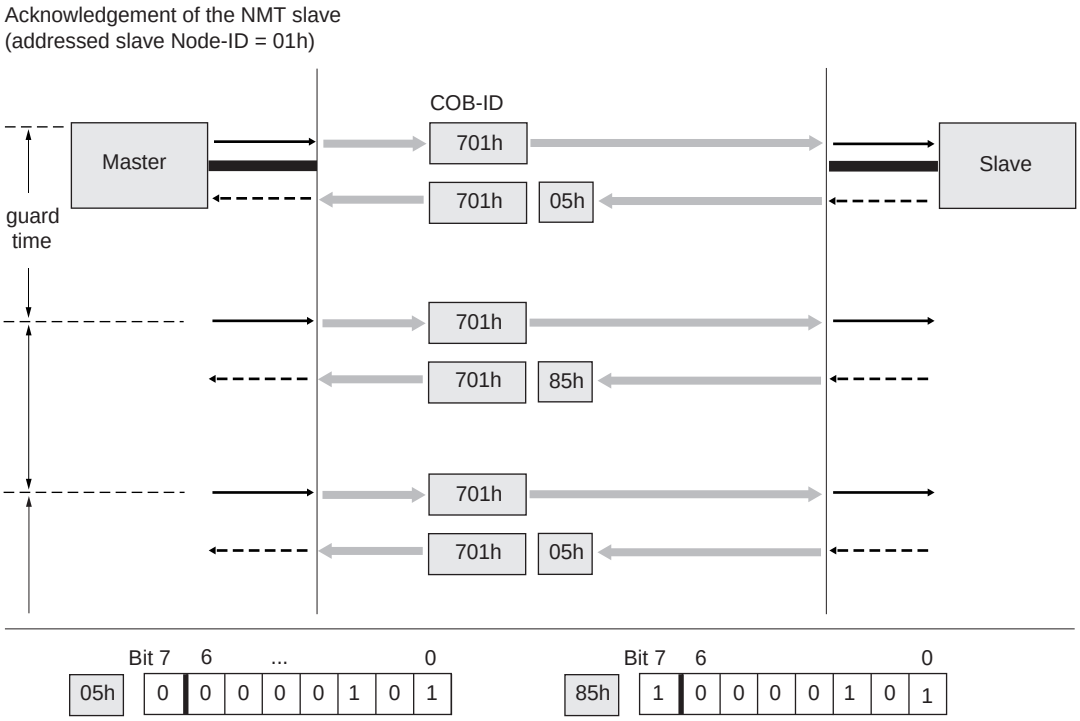
2.13.3 Node/life guarding

COB ID

Connection monitoring is executed with the communications object NMT error control (700h+node-Id). The COB ID for every NMT slave is calculated from the node address:
COB ID = function code NMTerror control (700h) + node-Id.

Structure of the NMT message

On request of the NMT master the NMT slave responds with one data byte.



Bit 0 to 6 identify the NMT status of the slave:

- 4 (04h): “Stopped”
- 5 (05h): “Operational”
- 127 (7Fh) : “Pre-Operational”.

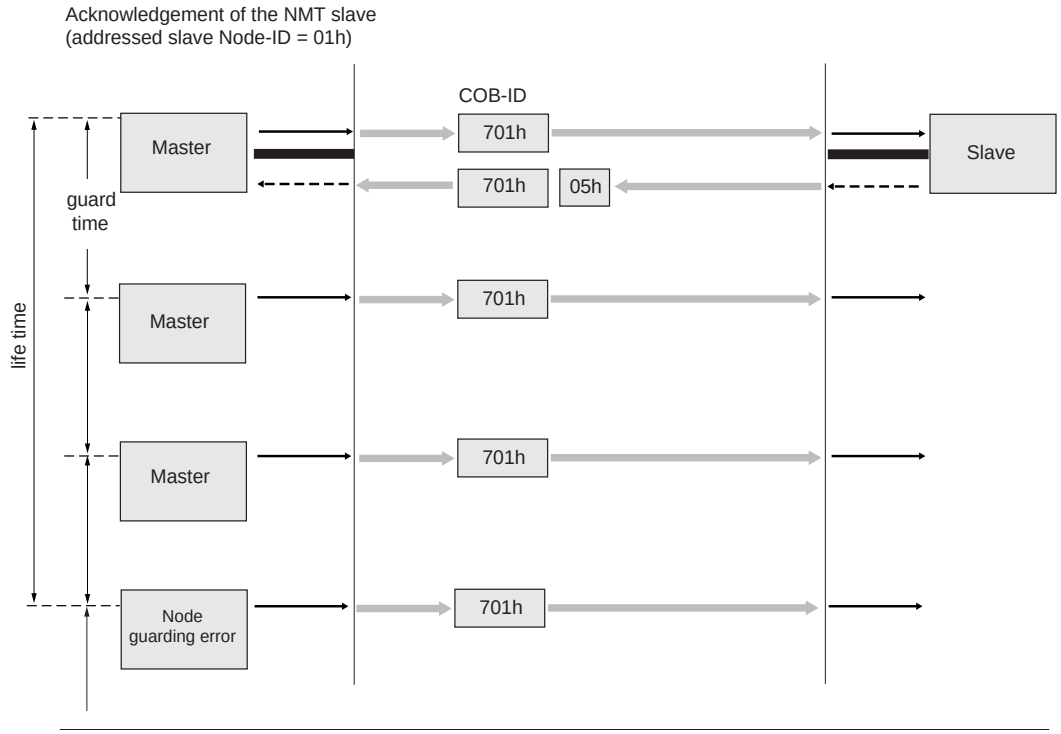
After every guard-time interval bit 7 switches its status between “0” and “1”, so the NMT master can detect and ignore a second acknowledgement within the guard-time internal. The first request when starting connection monitoring begins with bit 7 = 0.
Connection monitoring must not be enabled during the initialisation phase of a device. The status of bit 7 is reset as soon as the device runs though the NMT status “Reset communication”.
Connection monitoring continues in the NMT status “Stopped”.

Connection error

The NMT master reports a connection error to the higher level master program if:

- the slave does not acknowledge within the “guard time” period
- the NMT status of the slave has changed without the initiation of the NMT master.

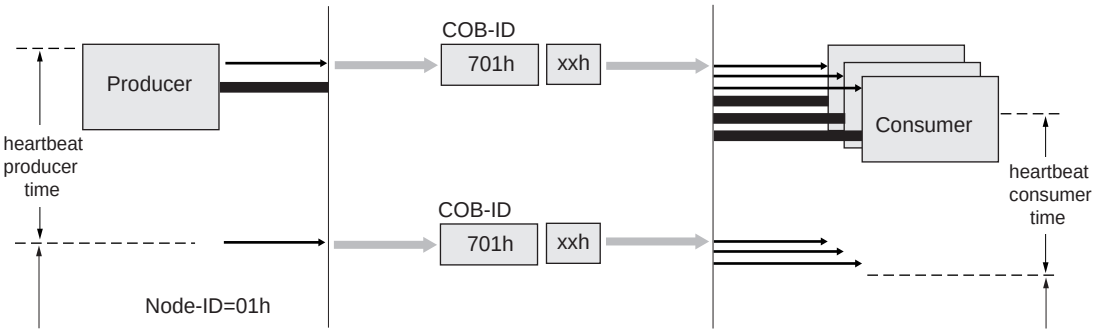
Figure below shows an error message after the end of the third cycle because of a missing answer of a NMT slave.



2.13.4 Heartbeat

The optional heartbeat protocol replaces the node/life guarding protocol. It is recommended for new device versions. A heartbeat producer transmits a heartbeat message cyclically at the frequency defined in the object .
Producer heartbeat time (1017h). One or more consumers can receive this message.
Producer heartbeat time (1016h) = 0 disables heartbeat monitoring.

The relationship between producer and consumer can be configured with objects. If a consumer does not receive a signal within the time period set in the object.
Consumer heartbeat time (1016h) then it generates an error message (heartbeat event).
Consumer heartbeat time (1016h) = 0 deactivates monitoring by a consumer.



Data byte for NMT status evaluation of the heartbeat producer:
- 0 (00h): “Boot Up”
- 4 (04h): “Stopped”
- 5 (05h): “Operational”
- 127 (7Fh): “Pre-operational”.

Time intervals
The time intervals are set in 1-ms steps and must not be set smaller for the consumer than for the producer. Whenever the heartbeat message is received the time interval of the producer is restarted.

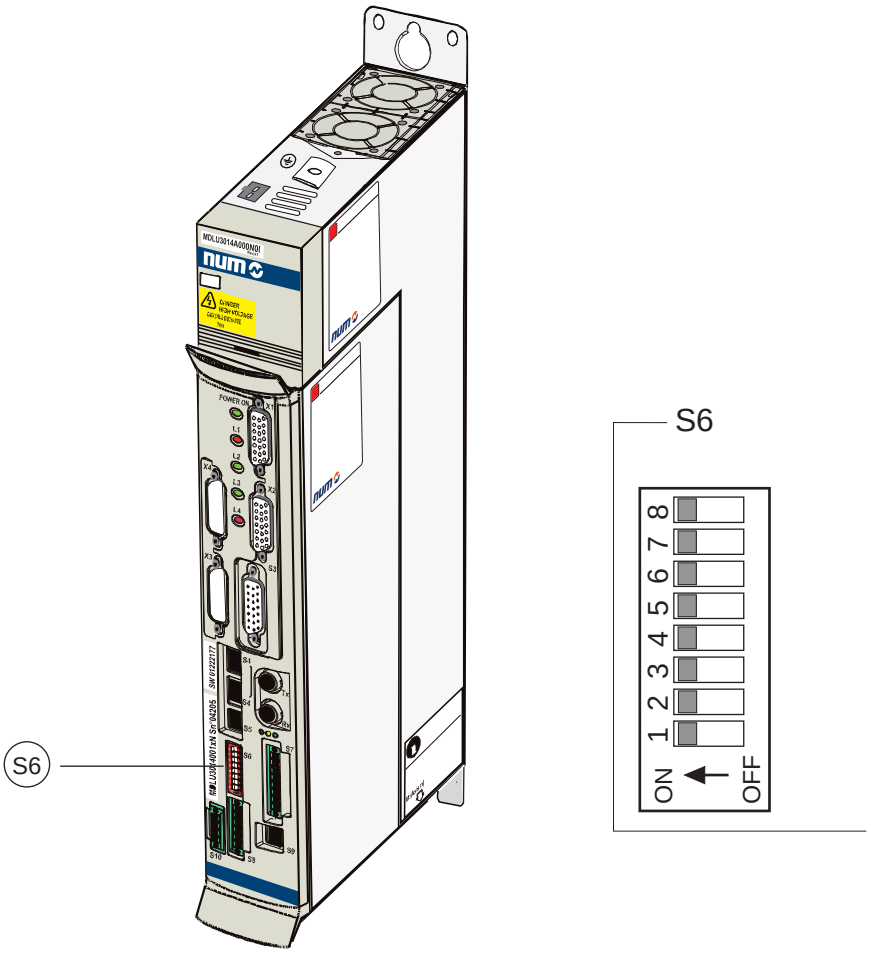
Start of monitoring
“”Heartbeat monitoring starts as soon as the time interval of the producer is greater than zero.

“”If heartbeat monitoring is active during the NMT status change to “pre-operational” then heartbeat monitoring starts by sending the boot up message.

”””””””” Information about “the boot-up message” can be found in chapter 2.12 “Emergency service“.
“Devices can monitor each other by means of the heartbeat message.”
They adopt a consumer and producer function simultaneously.

2.14 Node address

2.14.1 Setting the NMT node address by dip-switches



Please refer to NUMDrive C Installation manual AMOMAN012 chapter 9 for more information about the dip switches setting.

2.15 Transfert rate by V01 parameter

Please refer to the chapter 5 for V01 parameter description.

Page deliberately empty.

Page deliberately empty.

Summary

3 Flexium PLC library description 73

3.1 Integrating a Flexium CANopen Axes device in FXTools 73
3.1.1 Installing a CANopen device in FlexiumTools 73
3.2 Installing the “NUMDriveCANInterface” library 75
3.3 Adding a CANBus to the Flexium_RTS 77
3.3.1 CANBus Settings 79
3.4 Adding a CANopen_Manager to the CANBus 80
3.4.1 CANopen_Manager Settings 80
3.5 Adding a NUMDrive C CANopen Mono-Axis device to the CANopen_Manager 81
3.5.1 Mono-Axis device settings 82
3.5.1.1 Register: CANopen Remote Device (Heartbeat producer - consumer) 82
3.5.1.2 Register: PDO Mapping (PDO selection for Velocity or Positioning Mode) 84
3.5.1.3 Register: PDO Mapping (PDO Properties - Transmission Type settings) 85
3.5.1.4 Register: Receive PDO Mapping (Change RPDO Mapping) 86
3.5.1.5 Register: Send PDO Mapping (Change TPDO Mapping) 87
3.5.1.6 Register: Service Data Object (Add or Remove SDO’s & Change SDO values) 88
3.5.1.7 Register: CANopen I/O Mapping (Setting up the I/O Addresses) 90
3.6 Adding a NUMDrive C CANopen Bi-Axes device to the CANopen_Manager 92
3.6.1 NUMDrive C CANopen Bi-Axes device settings 94
3.6.1.1 Register: CANopen Remote Device (Heartbeat producer - consumer) 94
3.6.1.2 Register: PDO Mapping (PDO selection for Velocity or Positioning Mode) 96
3.6.1.3 Register: PDO Mapping (PDO Properties - Transmission Type settings) 97
3.6.1.4 Register: Receive PDO Mapping (Change RPDO Mapping) 100
3.6.1.5 Register: Send PDO Mapping (Change TPDO Mapping) 101
3.6.1.6 Register: Service Data Object (Add or Remove SDO’s & Change SDO values) 102
3.6.1.7 Register: CANopen I/O Mapping (Setting up the I/O Addresses) 104
3.7 NUMDriveCAN Interface library – data types and structure 106
3.7.1 Important data types 106
3.7.1.1 “DutInpNUMDriveCAN” & “DutOutNUMDriveCAN” 106
3.7.1.2 “EnumNumDriveCan_Mode” 106
3.7.1.3 “EnumNumDriveCan_Select” 106
3.7.1.4 “EnumNumDriveCan_HomingMethod” 107
3.7.1.5 “EnumNumDriveCan_Param” 107
3.8 Library structure 108
3.8.1 Overview Library Function Blocks 108
3.8.2 NUMDriveCAN_VelocityMode Structure 108
3.8.3 Parameters / Variables Description Velocity Mode 109
3.8.4 NUMDriveCAN_PositioningMode Structure 110
3.8.5 Parameters / Variables Description Positioning Mode 111
3.9 “Velocity Mode” in NUMDriveCANInterface library 113
3.9.1 Instancing the “Velocity Mode” function block 113
3.9.2 “VAR Config” for “NUMDriveCANInterface” instances 114
3.9.3 Using an instance in a POU 115
3.9.4 IntelliSense 116
3.9.5 General Settings 117
3.9.6 Mono-Axis and Bi-Axes distinction 118
3.9.7 Drive Control 119
3.9.7.1 “TorqueEnable” 119
3.9.7.2 “Operation Mode Setting” 119
3.9.7.3 “ManualControl” 120
3.9.7.4 “AutoControl” 120

3.9.8 Instance invoke 121
3.9.9 “DriveStatus” 121
3.9.10 “Alarms” 122
3.9.11 “AlarmReset” 122
3.9.12 “SDOAccess” 123
3.9.12.1 “ReadParameters” 123
3.9.12.2 “SetParameters” 124
3.9.12.3 Read / Set - Several Parameters 125
3.10 “Positioning Mode” in “NUMDriveCANInterface” library 126
3.10.1 “VAR Config” for “NUMDriveCANInterface” instances 127
3.10.2 Using an instance in a POU 128
3.10.3 IntelliSense 129
3.10.4 “General Settings” 130
3.10.4.1 Homing Enable 130
3.10.4.2 Software Position Limits 130
3.10.5 Mono-Axis and Bi-Axes distinction 131
3.10.6 “Drive Control” 132
3.10.6.1 “TorqueEnable” and “ReferenceEnable” 132
3.10.6.2 “Operation Mode Setting” 132
3.10.6.3 “ManualControl” 133
3.10.6.4 “AutoControl” 133
3.10.6.5 Relative Movement 134
3.10.6.6 Change Positioning Immediately 134
3.10.7 “Homing” 135
3.10.8 Instance invoke 137
3.10.9 “DriveStatus” 138
3.10.10 “Alarms” 139
3.10.11 “AlarmReset” 139
3.10.12 “SDOAccess” 140
3.10.12.1 “ReadParameters” 140
3.10.12.2 “SetParameters” 141
3.10.12.3 Read / Set - Several Parameters 142
3.11 Modulo function 143
3.12 Extended SDO Access 144
3.13 Synchronous Start in Positioning Mode 146
3.14 Example for extended axis status information in Positioning Mode 151

3 Flexium PLC library description

3.1 Integrating a Flexium CANopen Axes device in FXTools

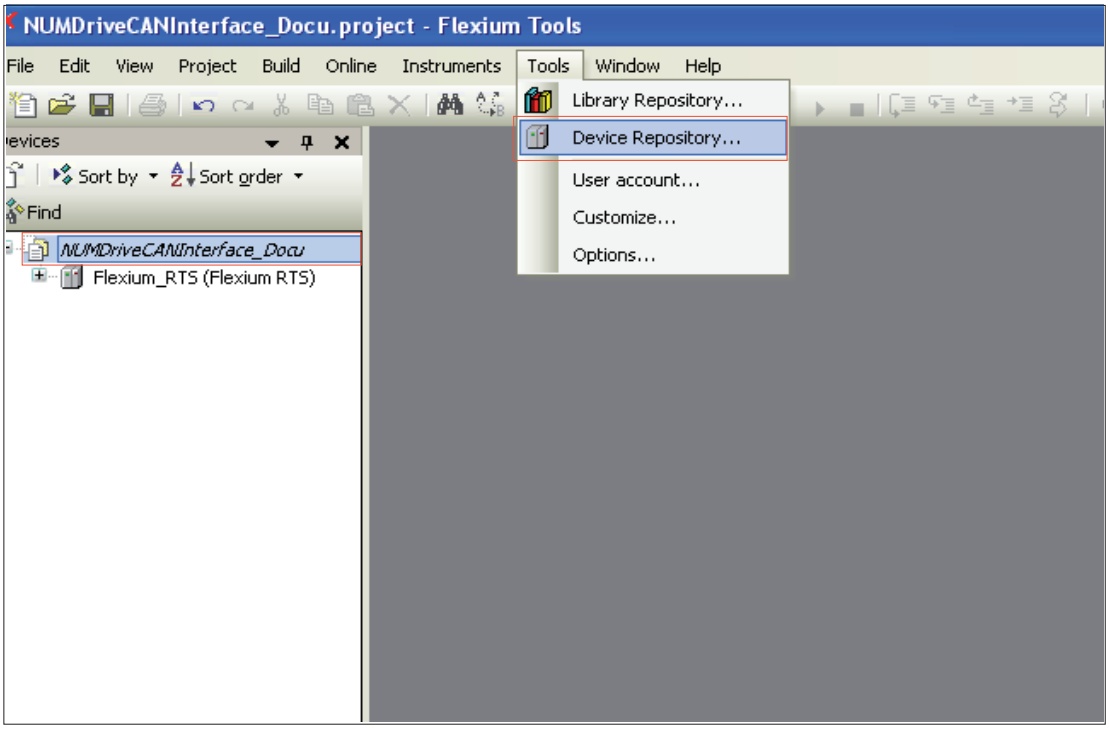
3.1.1 Installing a CANopen device in FlexiumTools

This installation is only necessary if the NUMDrive C CANopen Device is not available after the standard installation of FXTools. (Flexium Tools)

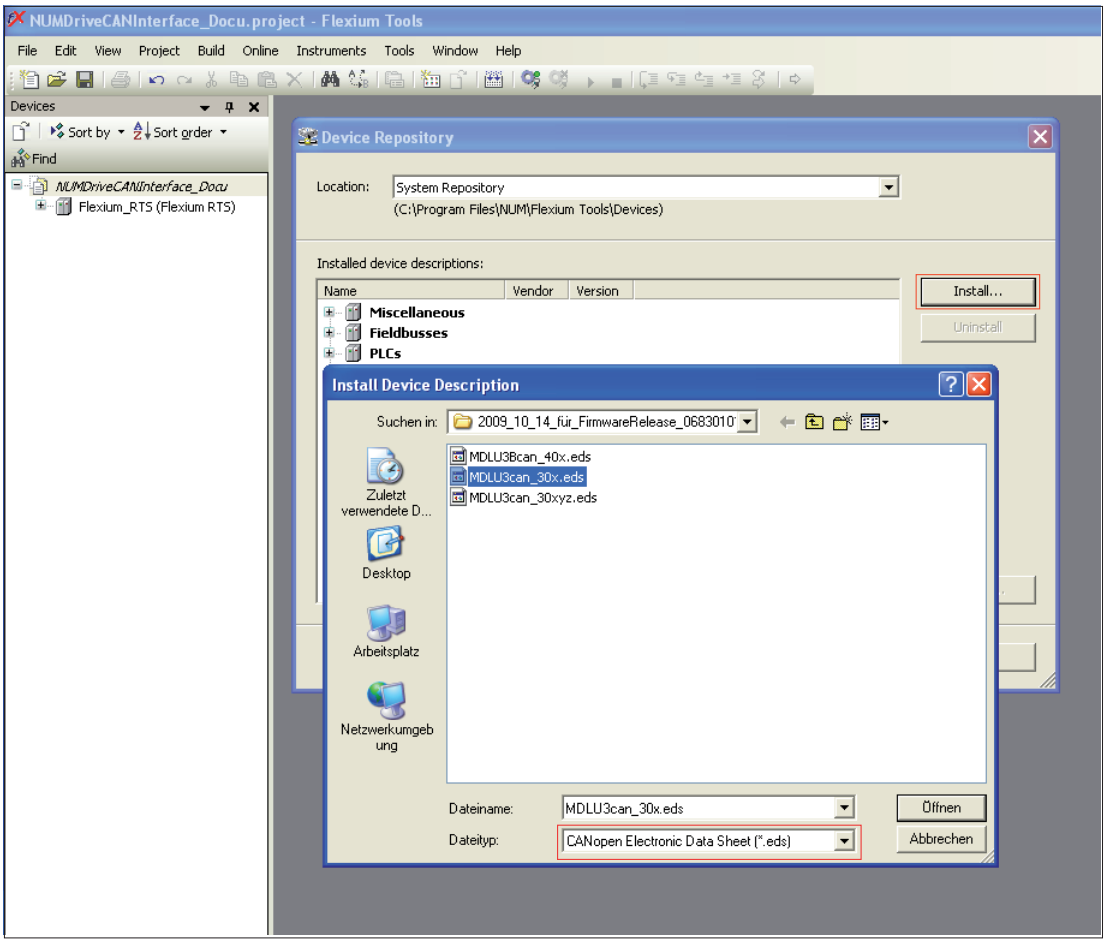
To use a NUMDrive C CANopen Device in your project it is necessary to install the device to the Device Repository of your FlexiumTools first.

To install a CANopen device you need the according "*.eds" file (electronic data sheet). Use the device repository to install CANopen devices in FlexiumTools, like described in the following steps:

Open the Device Repository Window: ☞Tools > Device Repository



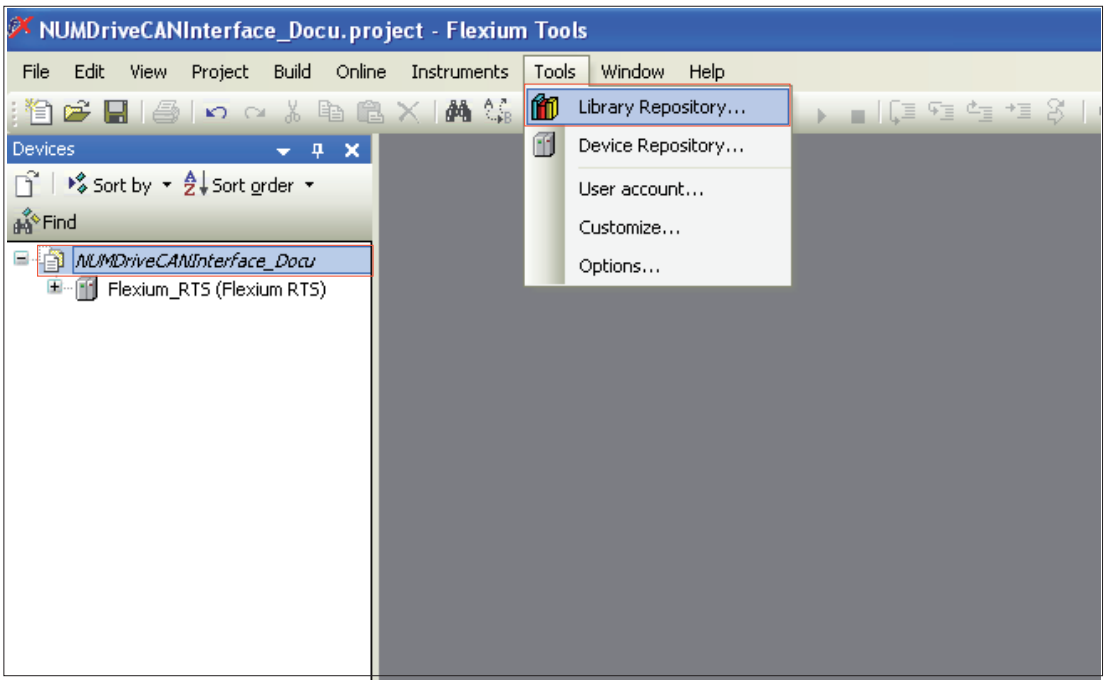
Click ☞“Install” and select the according “*.eds” file
Make sure that “CANopen Electronic Data Sheet” is selected



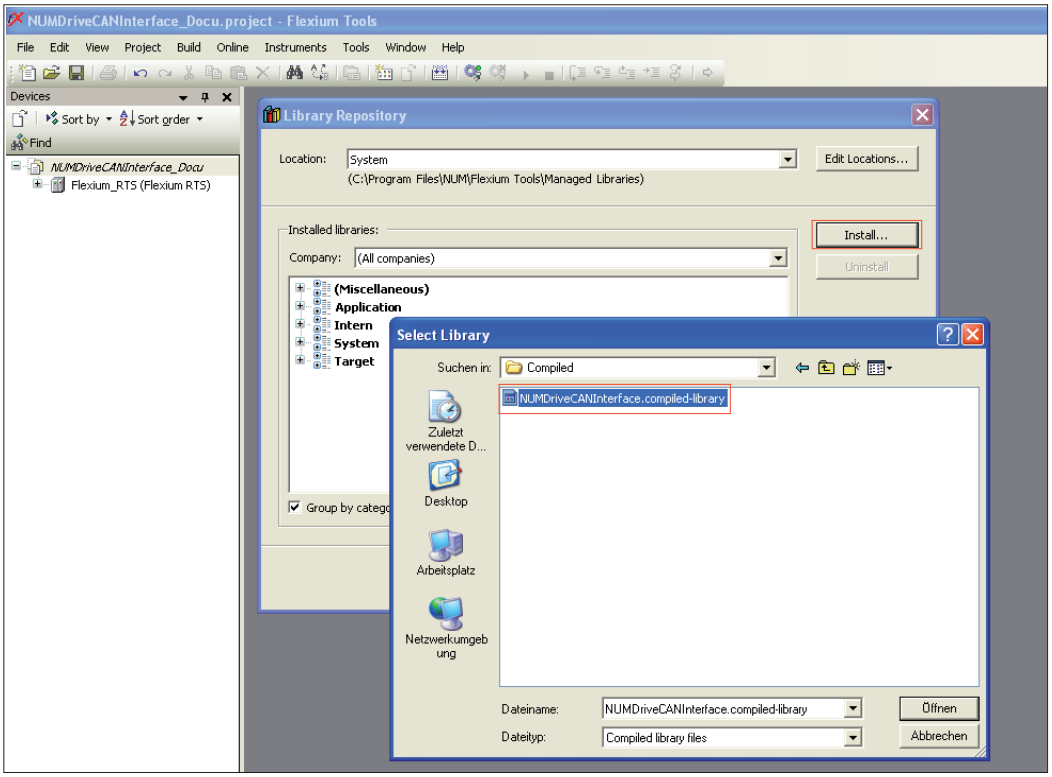
3.2 Installing the “NUMDriveCANInterface” library

This installation is only necessary if the NUMDriveCANInterface Library not available after the standard installation of FXTools. To use the NUMDriveCANInterface Library in your project it is necessary to install the library to the Library Repository of your first FlexiumTools.
To install the NUMDriveCANInterface Library you need the **“*.compiled-library” file.** Use the device repository to install CANopen devices in FlexiumTools, like described in the following steps:

Open the Device Repository Window: ☞Tools > Library Repository

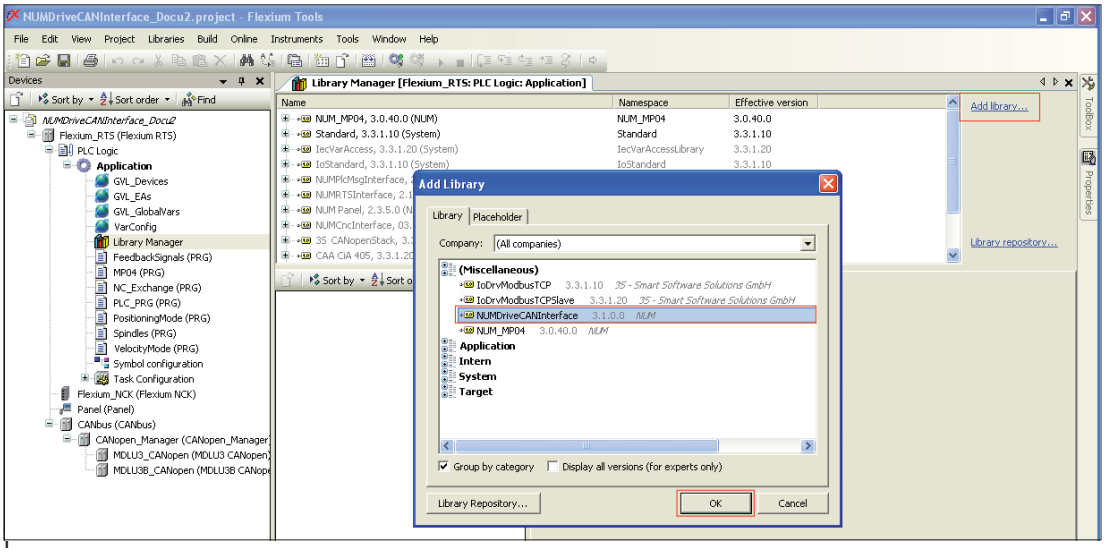


Click to ☞ “Install” and select the according “*.compiled-library” file.



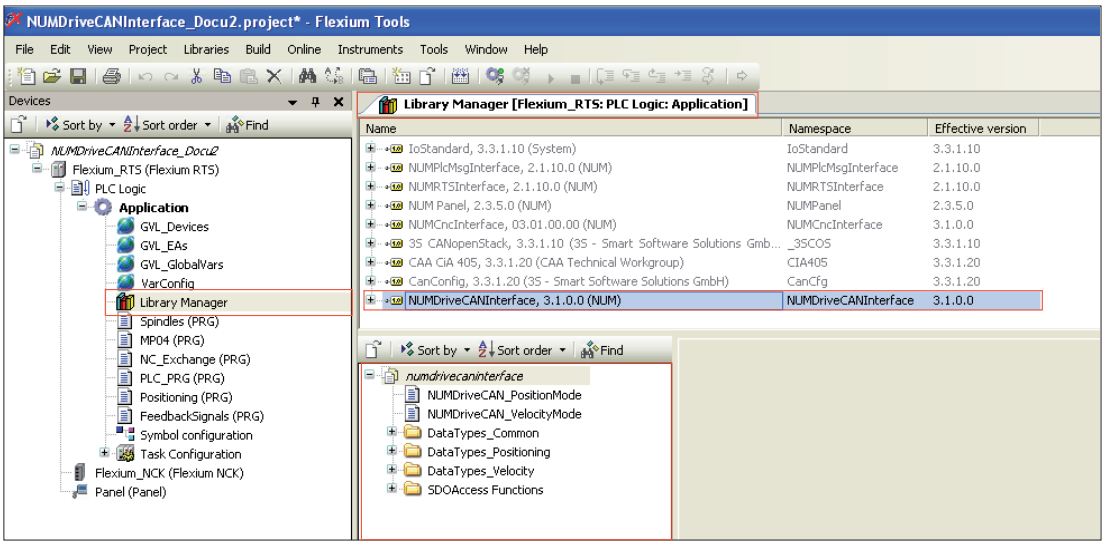
After the successful installation the NUMDriveCANInterface is available in FXTools to add it to the Library Manager.

Click to ☞ “Add Library” the selection window will appear select the library “NUMDriveCANInterface” than press ☞ OK



After you made this selection, the “NUMDriveCANInterface” library is imported to your project. You can see it in the “Library Manager” window.

When you click ☞on the library in this window you will see a list of all data types and function blocks in the library.



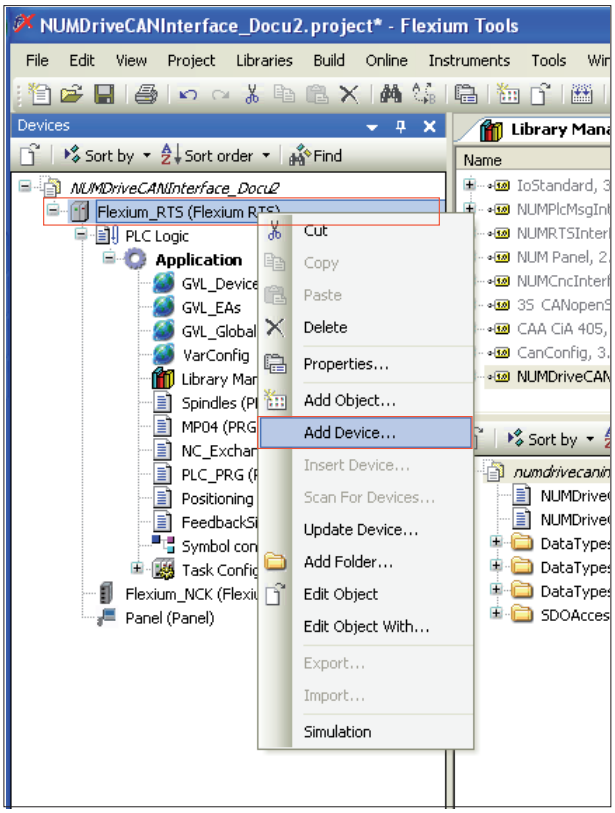
3.3 Adding a CANBus to the Flexium_RTS

It is recommended to set up the whole CANbus after the EtherCAT bus in the device tree and using a high address area which is over all the others before. Otherwise it is possible that a modification of the Safe I/Os or EtherCAT I/Os can change your CANbus address areas.

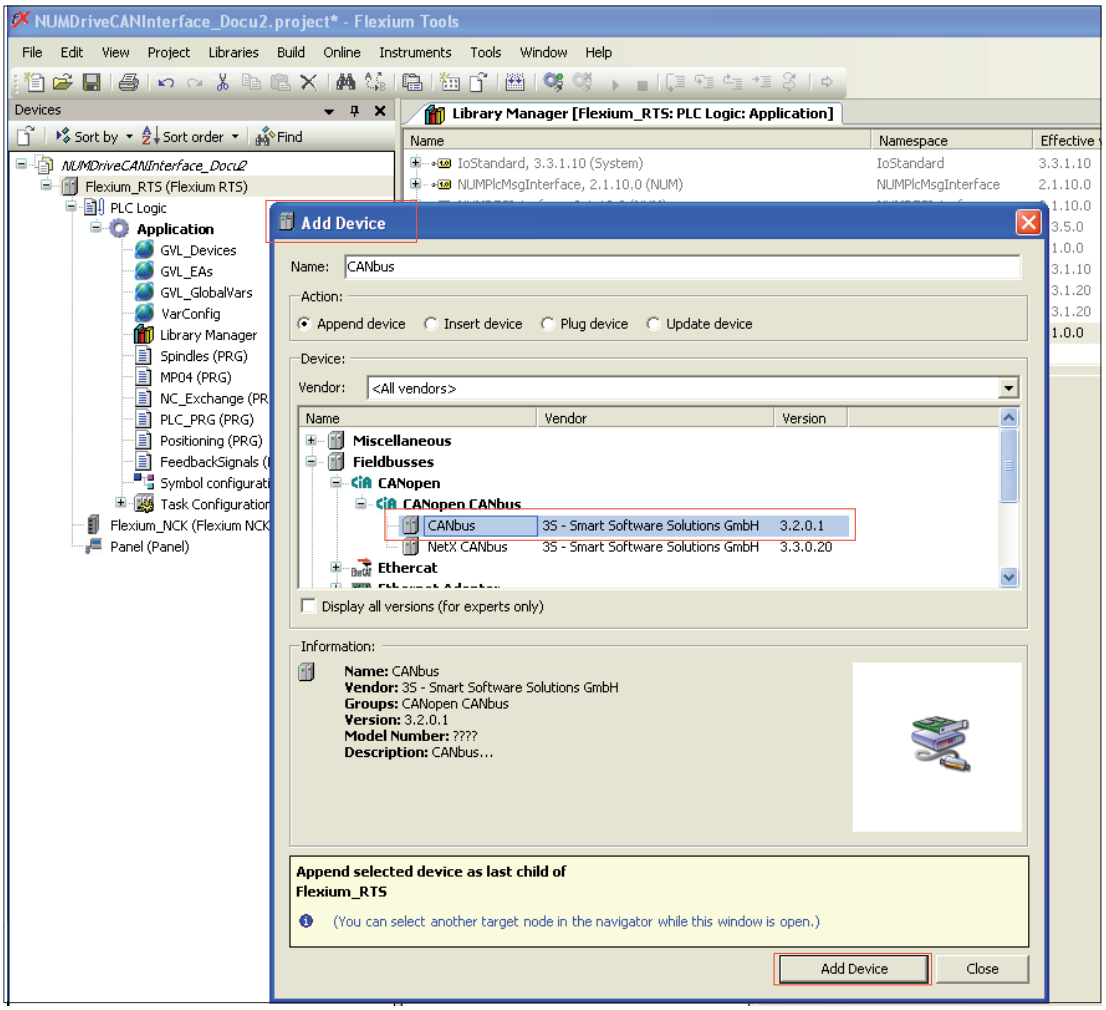
If you want to use CANopen devices in your application it is mandatory to have a CANBus. CoDeSys requires a hardware configuration for this CANBus.

This hardware configuration must contain several settings to let the PLC know how to communicate with the CANBus. We will take a detailed look to that settings later, but first we have to add a CANBus to the Flexium_RTS.

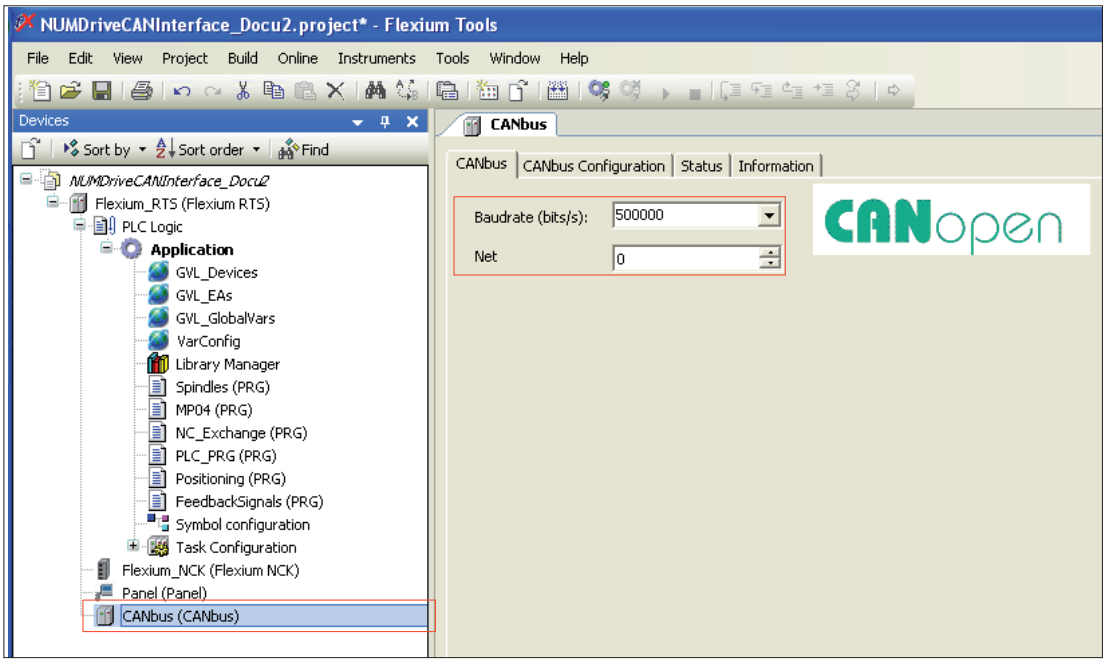
Make a right mouse-click on ☞ “Flexium_RTS” in the “Devices” window and select “Add Device”.



The “Add Device” window appears; here you can select the ☞ “CANbus”.



After you clicked ☞ the Button “Add Device” the CANBus will be append to the Flexium_RTS in the “Devices” window.



The CANBus appears in the “Devices” window. With a double click on ☞ “CANBus” the “CANbus” parameter register appears. For these settings please refer to the next page.

3.3.1 CANBus Settings

For the CANBus settings make the following operation:

- Baudrate configuration (CANopen Bit rate and cable length):
  Please refer to the V01 parameter on chapter 5.

- Net:
  Net defines the interface on the backside of the FS152i.
  The relation between the hardware interface on the FS152i and this Net setting in FlexiumTools is as described in the table below:

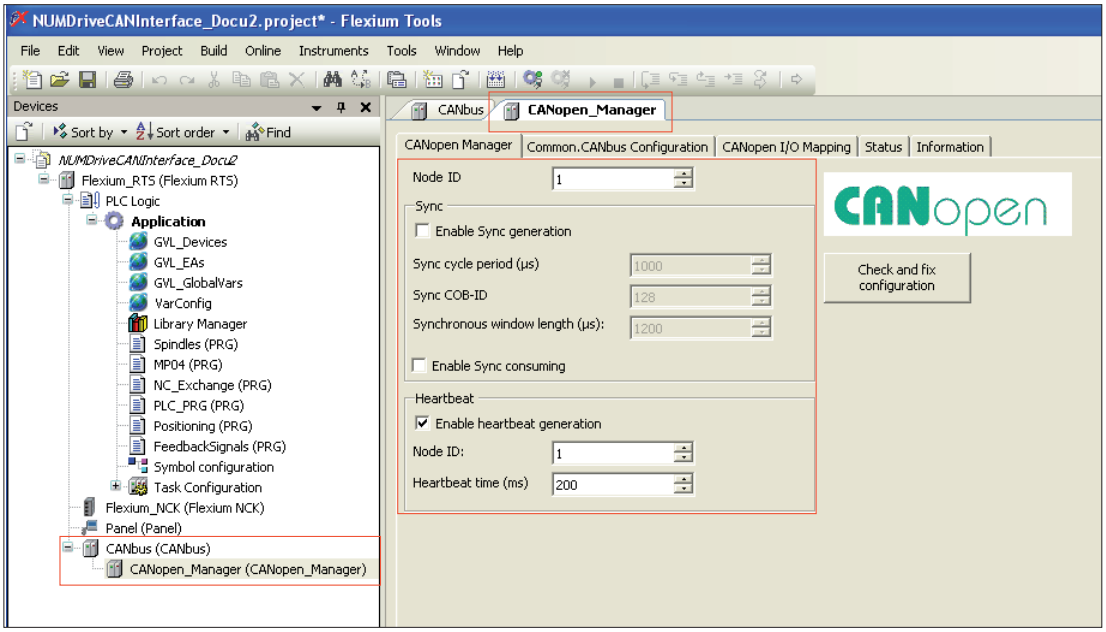
Interface	Net
CAN1	0
CAN2	1

3.4 Adding a CANopen_Manager to the CANBus

The next step to set up a working CANBus configuration is to add a CANopen_Manager to the CANBus. A CANopen_Manager represents the Master on CANBus in the Flexium System.
There are some settings that must be set up in the CANopen_Manager parameters like NodeID, Heartbeat time and Heartbeat NodeID.

You can add a CANopen_Manager to the CANBus in the same way like you add the CANBus to the Flexium_RTS.

First make a right mouse-click on ☞“CANbus” in the “Devices” window and select “Add Device”.
After the “Add Device” window appears select ☞“CANopen_Manager” and the Manager will be append to the “CANbus”
Double click the ☞“CANopen_Manager” in the “Devices” window and the “CANopen_Manager” parameter register appears on the right side.



3.4.1 CANopen_Manager Settings

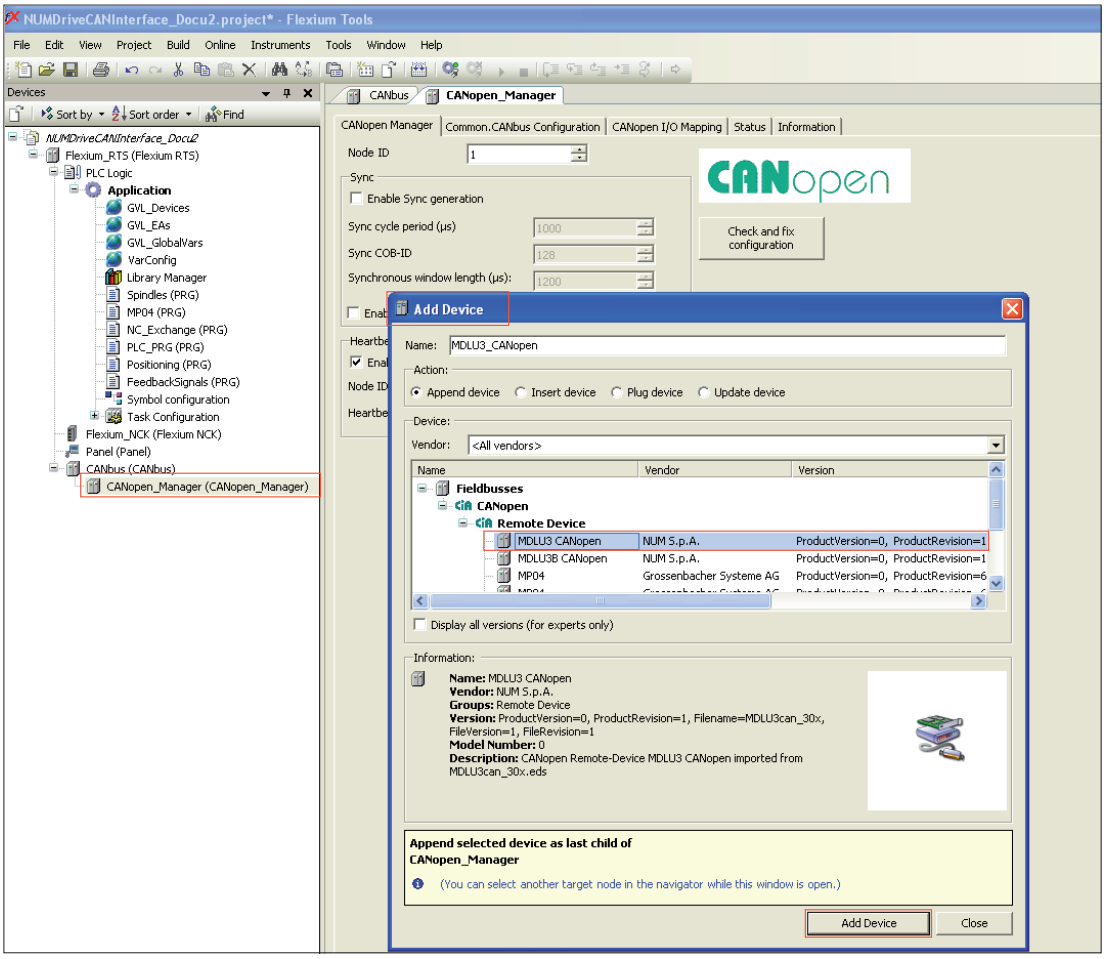
For the CANopen_Manager settings make the following operation:

- NodeID: NodeID = 1 for the Master is recommended.
- Enable heartbeat generation: only if needed (if heartbeat generation is enabled CANopen devices can control if the Master (PLC) is still on the bus and “alive”.
- The “Heartbeat NodeID”: it should be the same as the CANopen_Manager (Master) NodeID.
- The heartbeat time: it depends on specific application properties like cable length, number of devices, busload…it should be chosen in respect to the demanded security level.
- Enable Sync generation: only if needed.

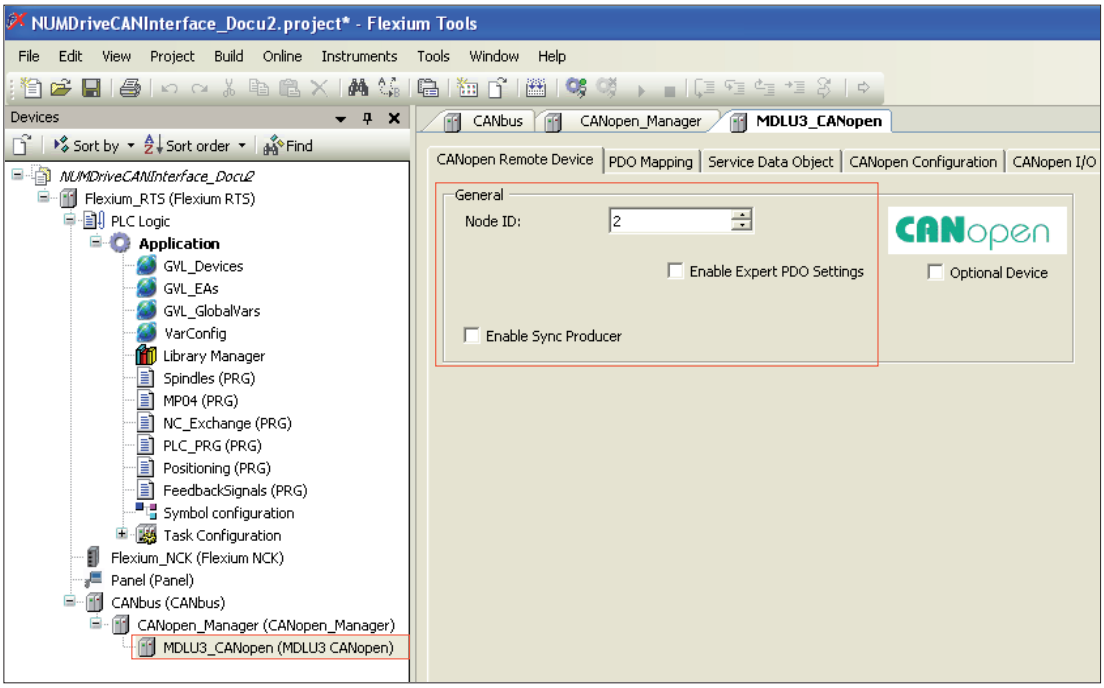
3.5 Adding a NUMDrive C CANopen Mono-Axis device to the CANopen_Manager

You can add a NUMDrive C CANopen device to the CANopen_Manager in the same way like you add the CANopen_Manager to the CANBus.

First make a right mouse-click on ☞“CANopen_Manager” in the “Devices” window and select “Add Device”.
After the “Add Device” window appears select ☞“MDLU3 CANopen” (for Mono-Axis) or “MDLU3B CANopen” (for Bi-Axes) dependent of your application and the NUMDrive C CANopen device will be append to the “CANopen_Manager”.



Double click the ☞“MDLU3_CANopen” in the “Devices” window and the “MDLU3_CANopen” parameter register appears on the right side.



3.5.1 Mono-Axis device settings

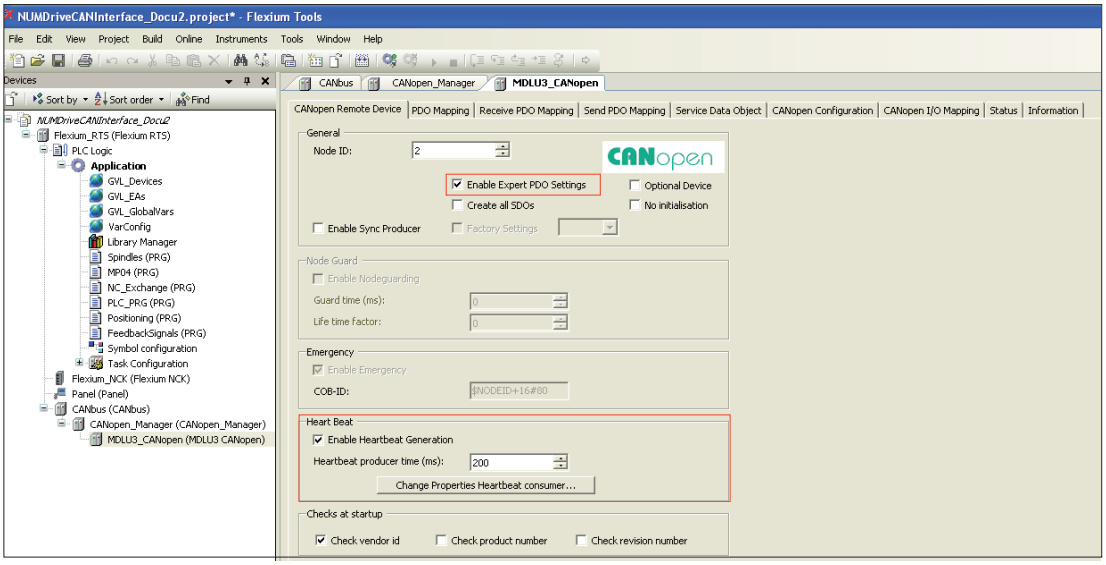
3.5.1.1 Register: CANopen Remote Device (Heartbeat producer - consumer)

For this setting make the followiing operations:

- NodeID: it is the hardware address of the device (the address that is set up with the dip switch S6 of the drive. Please refer to NUMDrive C Installation Manual AMOMAN012.

- Enable Expert PDO Settings: click here if you want to extend to more specific parameters as indicated in the picture upon next page.

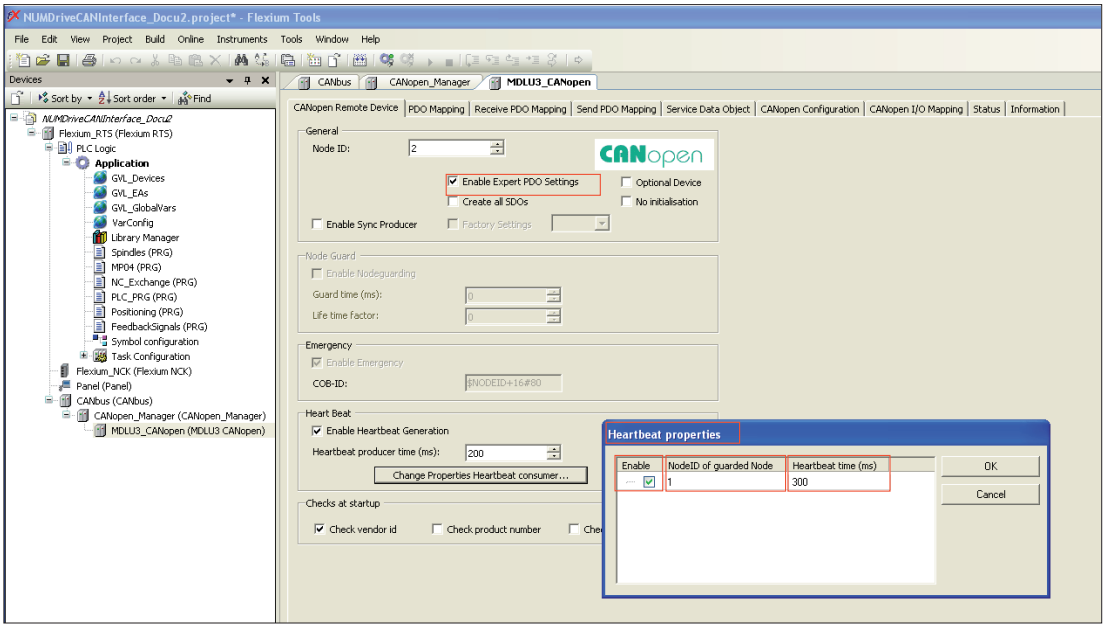
Extended view if “Expert PDO Settings” is enabled



Extended settings

- Enable Heartbeat Generation : (if heartbeat generation is enabled the Master and other CANopen devices can control if the drive is still on the bus and “alive”.
- Heartbeat producer time: defines the time period for the heartbeat produced by the drive.
- Heartbeat consumer: defines that this drive controls (consumes) the Master heartbeat or a heartbeat that is produced by another device.
  To make this definitions click ☞“Change Properties Heartbeat Consumer”

Heartbeat Consumer Settings



- Enable : Enables heartbeat consumer properties.
- NodeID: NodeID of the device which should be controlled.
- Heartbeat time : During this time period the heartbeat consumer expects a heartbeat (heartbeat producer time of the device that should be controlled x 1.5)

Explanation for this example

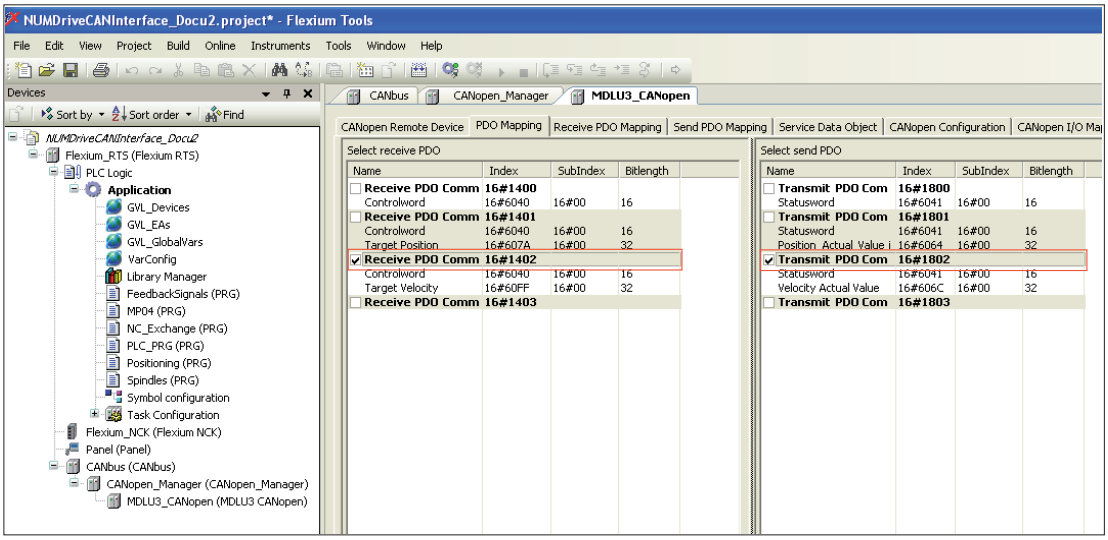
- The NUMDrive C (Addr.2) should control the Master (Addr.1) Heartbeat.
- The Master Heartbeat is set up with a producer time of 200ms.
- Consumer time should be 1.5 x producer time → 1.5 x 200ms = 300ms.

Reaction

- If the NUMDrive C doesn’t get a Master Heartbeat within 300ms, (in this example) the drive changes to error state.

3.5.1.2 Register: PDO Mapping (PDO selection for Velocity or Positioning Mode)

NUMDrive C CANopen provide two different function modes, velocity mode or positioning mode. Of course there are some differences for setting up and programming these two modes.
The first choice you have to do is to enable the correct PDO dependent on the mode you want to use.



- For Positioning Mode: Enable Receive PDO2 and Transmit PDO2
- For Velocity Mode: Enable Receive PDO3 and Transmit PDO3
- Disable all other PDO´s.

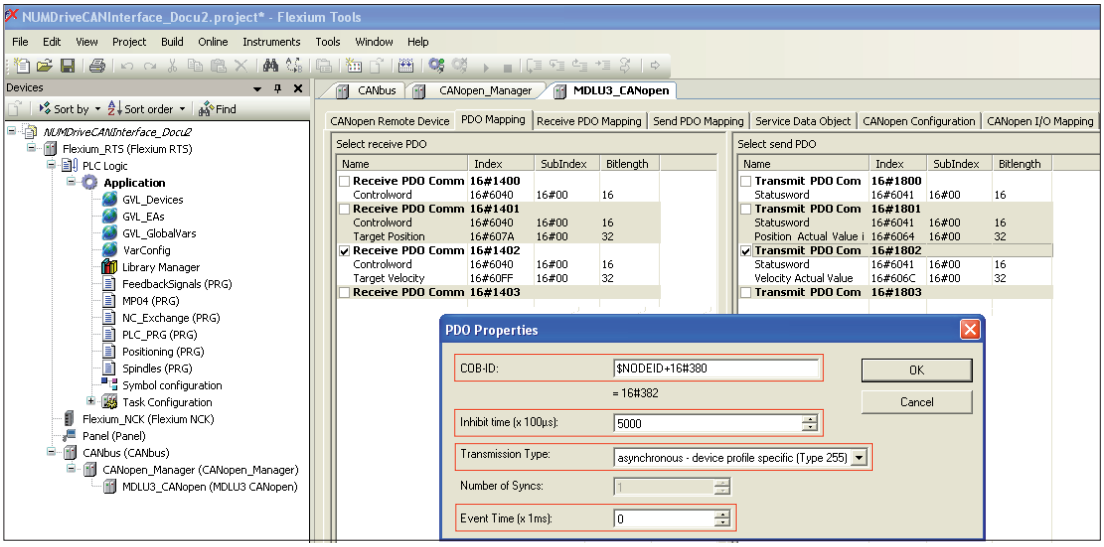
The RPDO1 and TPDO1 include only the controlword that means if you use these PDO´s you will get no actual position or actual velocity information!
In this example there is RPDO3 and TPDO3 selected (Velocity Mode).

3.5.1.3 Register: PDO Mapping (PDO Properties - Transmission Type settings)

On a CANopen bus there is the possibility to define when and how the enabled PDO´s should be sent. For communication with a NUMDrive C CANopen the Transmission Type 255 (asynchronous device profile specific) is selected as default.

To reach the parameter window for transmission types just double click ☞ the PDO you want change.

Transmission Type behaviour on Transmit PDO's



Transmission Type 255 is asynchronous that means the PDO gets send when there is a change in the PDO value. For a Transmit PDO Type 255 you can change following behaviour:

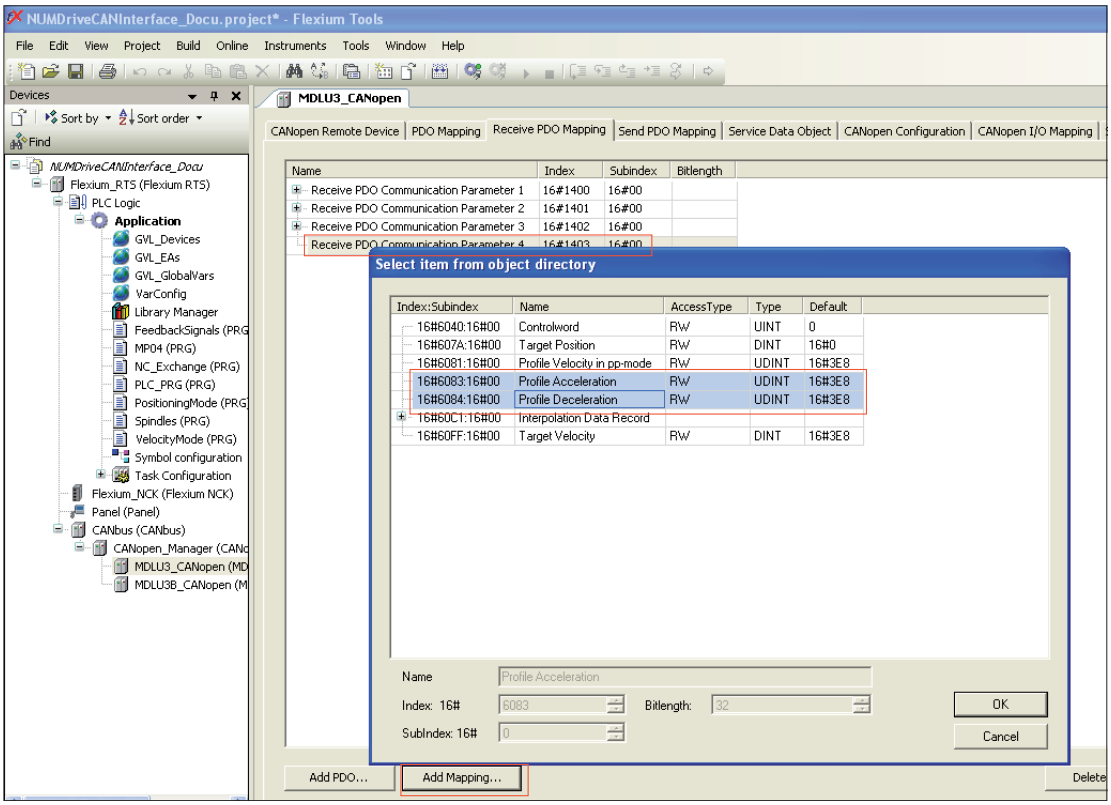
- COB-ID : do not change
- Inhibit Time : inhibits the sending of the next PDO until the inserted time period is reached (only if there was a PDO value change before).
- Event Time: is the opposite of Inhibit Time, the PDO is send after this time also if there was no PDO value change before.

It is possible to make different Transmission Type settings for each enabled PDO! For receive PDO´s there is no inhibit or event time!

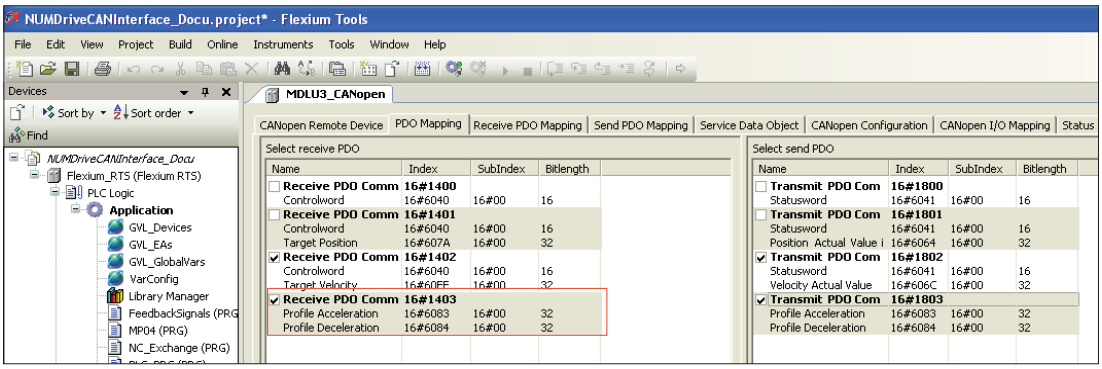
3.5.1.4 Register: Receive PDO Mapping (Change RPDO Mapping)

It is also possible to change the content of a PDO. For example if you want to change the acceleration and deceleration a lot of times you can use PDO4 (that is empty by default) to map the acceleration and deceleration object into it.

For this select PDO 4 and click to the ☞“Add Mapping” button.



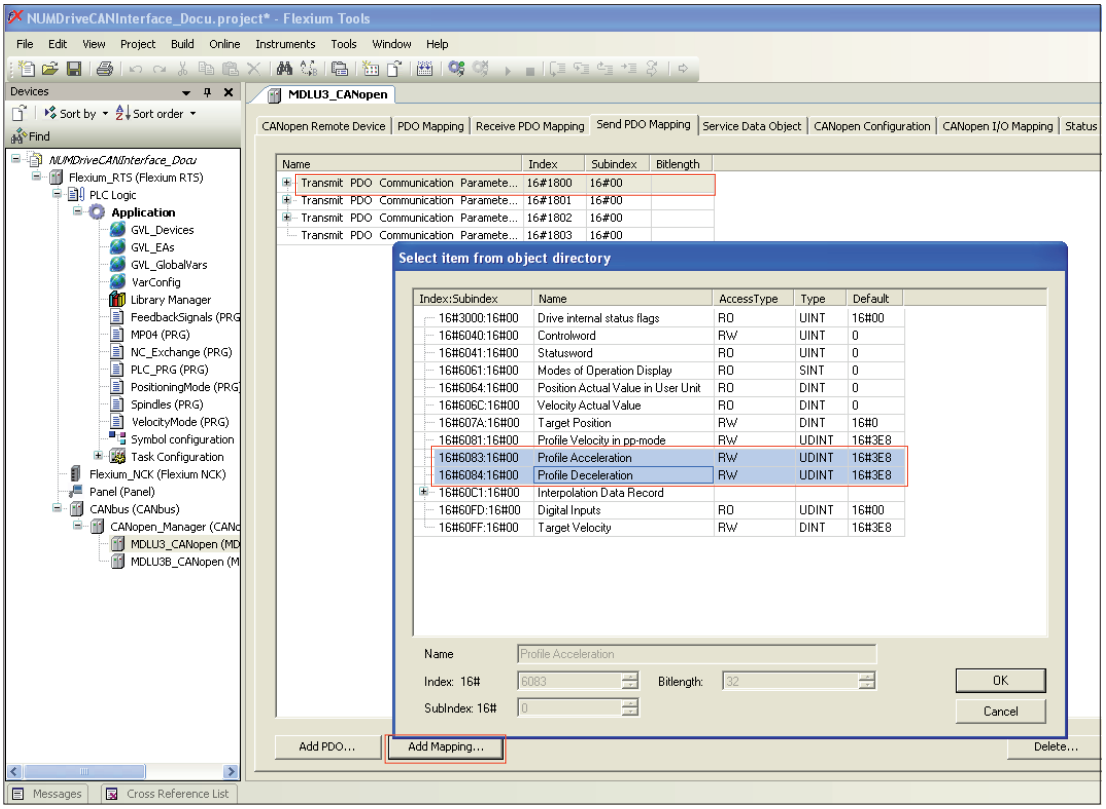
- In the appearing window you select ☞the objects you want to send (and remember, send PDO´s are sent by the CANopen device) and click ☞OK.
- Now change the register to ”PDO Mapping” and you can see that PDO4 has the content you have selected.
- Now you only must enable PDO4 to use it. Don’t forget to make a I/O Mapping also for PDO4 like for the other enabled PDO’s. Please refer to the chapter 3.5.1.7. Register: CANopen I/O Mapping (Setting up the I/O Addresses)



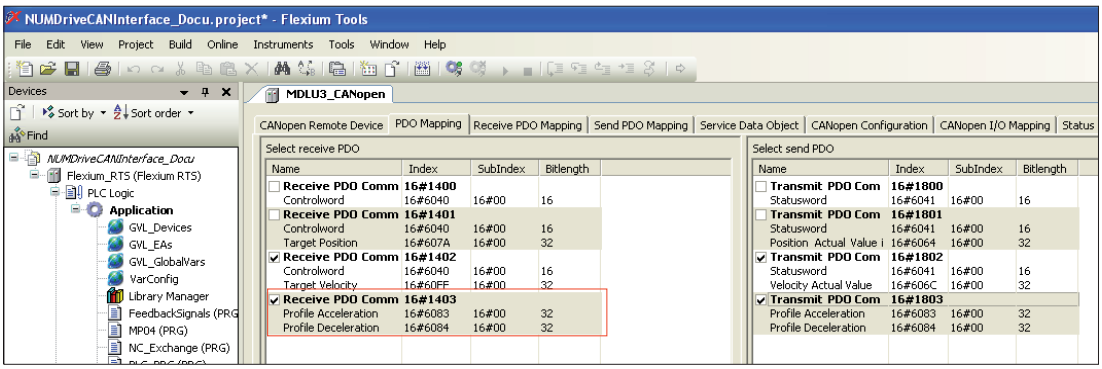
3.5.1.5 Register: Send PDO Mapping (Change TPDO Mapping)

Like for receive PDO's you can also change the content of transmit PDO's.
In this example we want to see if our change of acceleration or deceleration is reflected by the drive.

For this select PDO 4 and click to the ☞"Add Mapping" button.



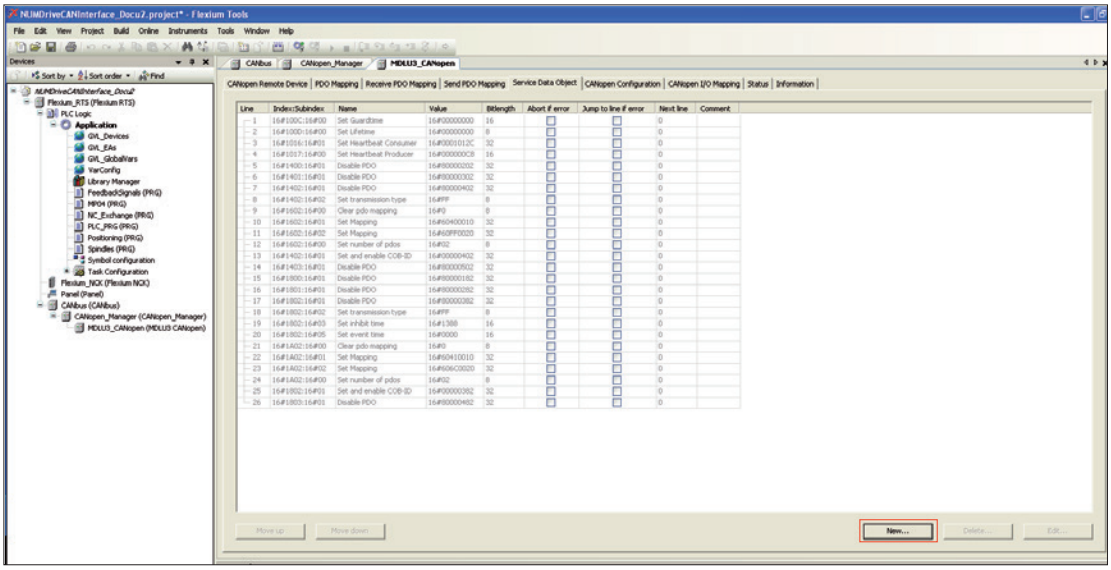
- In the appearing window you select ☞the objects you want to send (and remember, send PDO´s are sent by the CANopen device) and click ☞OK.
- Now change the register to "PDO Mapping" and you can see that PDO4 has the content you have selected.
- Now you only must enable PDO4 to use it. Don't forget to make a I/O Mapping also for PDO4 like for the other enabled PDO's. Please refer to the chapter 3.5.1.7. Register: CANopen I/O Mapping (Setting up the I/O Addresses)



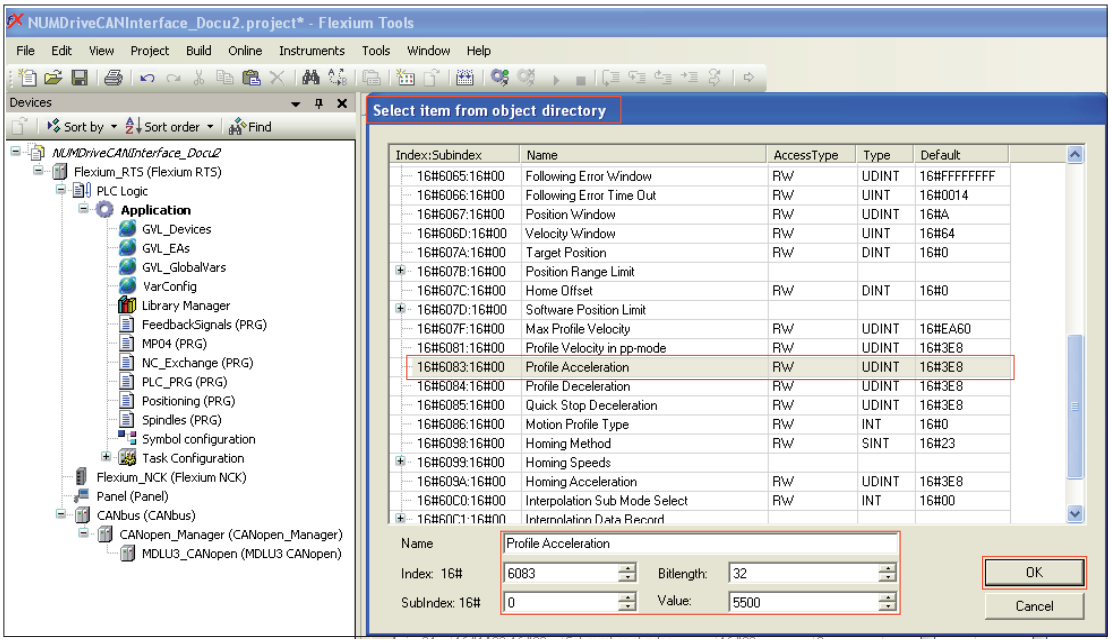
3.5.1.6 Register: Service Data Object (Add or Remove SDO's & Change SDO values)

Some NUMDrive C CANopen device parameter must be changeable by SDO according to CAN specification. There are several parameters which were removed from WPM access and transferred to SDO communication.

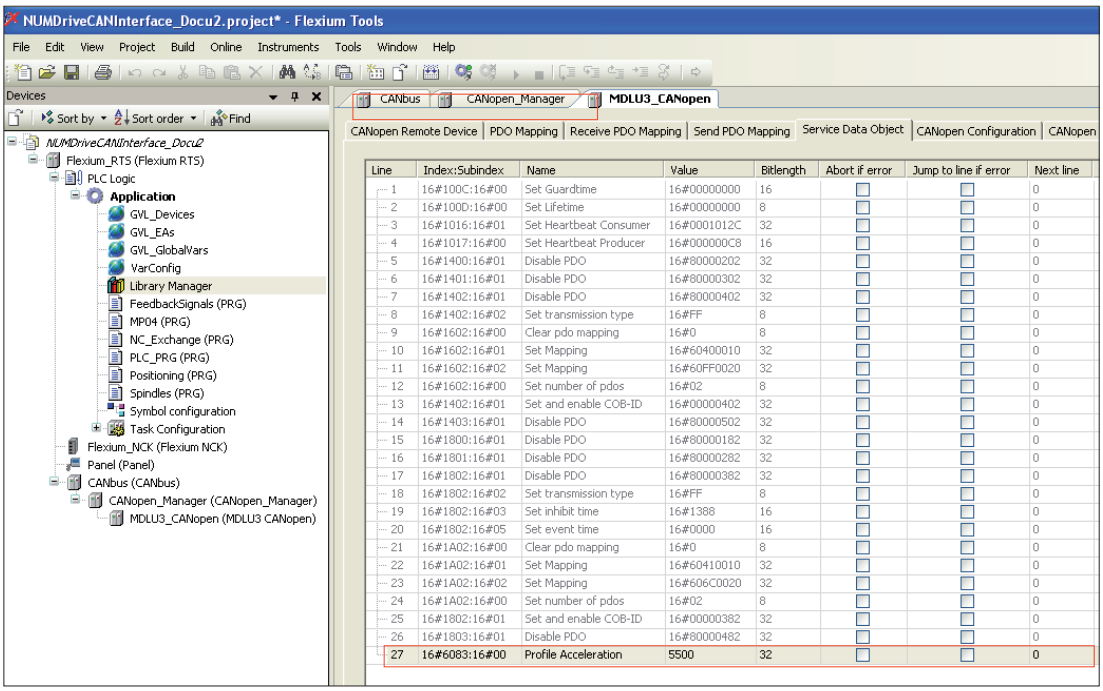
The procedure to add a specific SDO that's needed and change the SDO value with FXTools is always the same. Below the default (not removable) SDO´s which are declared in the appropriate **"*.eds"** file.
Expanding this default SDO´s, which are transmitted during the boot up of the PLC, is possible by clicking ☞the "New" button as indicated below.



The "Select item from object dictionary" window appears.
Here you can select a SDO object from the object dictionary on your demand.



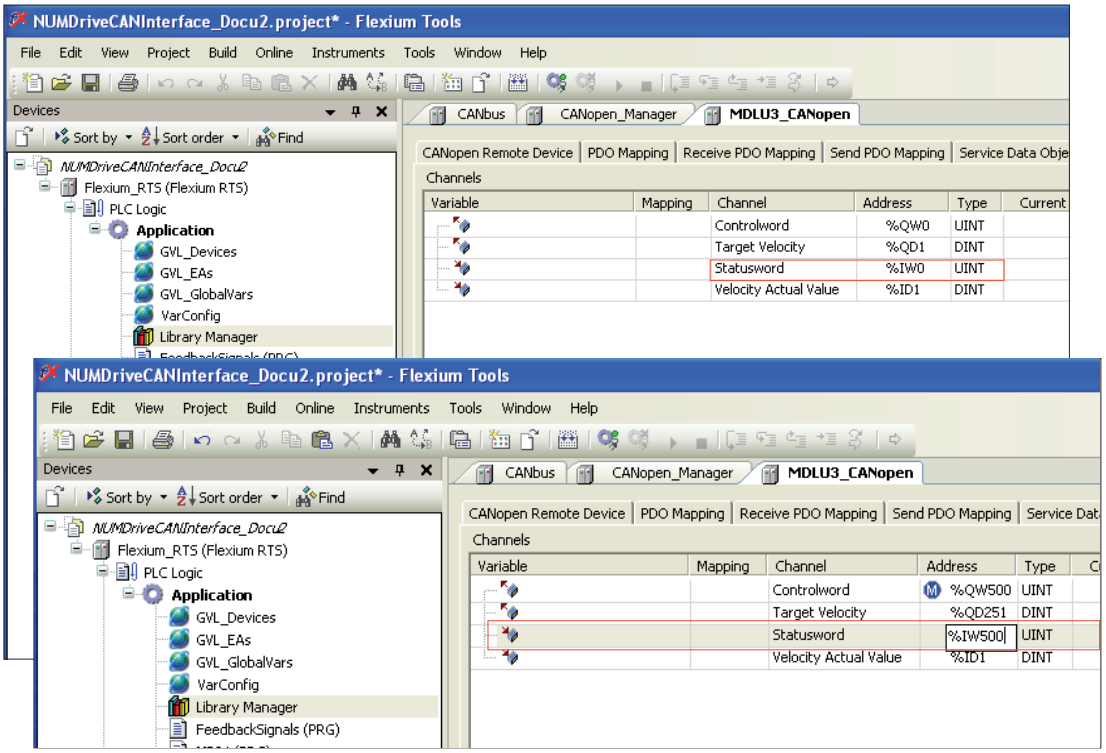
To change a value, just edit the value textbox and confirm.
As result your chosen SDO object will be inserted to the SDO object overview window.



Now, also this SDO object is transmitted during the boot up of the PLC but not immediately after adding!

3.5.1.7 Register: CANopen I/O Mapping (Setting up the I/O Addresses)

To establish a communication between PLC and NUMDrive C CANopen you must define I/O address areas. Dependent on the SDO selection you made before (see: Chapter 3.5.1.2 Register: PDO Mapping) you must assign the objects, which are declared to the selected PDO´s, to unique PLC addresses.
As default, FXTools take the next free address and assigns it to the enabled object.
Often this behaviour is not favored. To change the addresses to your choice, double click directly into the ☞ “Address” cell and it will get editable.

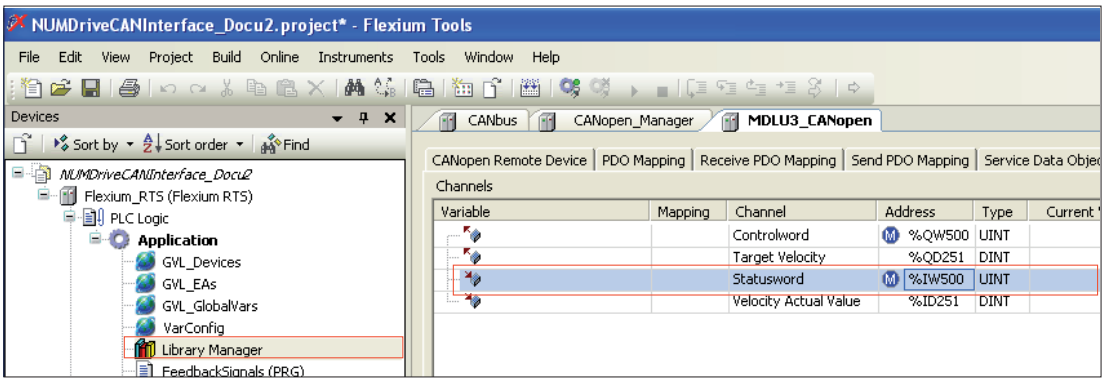


Important note
Remember, one TPDO for velocity or positioning includes one controlword (uint) and one target value (dint) and one RPDO for velocity or positioning includes one statusword (uint) and one actual value (dint)!

If you change the respectively first of these two objects (which gets marked with the blue “M”) the address for the other object is getting actualized automatically to the next possible address.
You are not allowed to change the respectively second of these two objects because the library handles one PDO as one package with one address. So never change the second entry!

The picture below shows a correct configuration example.

Each time the first object address of a PDO is changed manually and marked with a **blue “M”** and the second object address is calculated automatically without a blue “M” mark.



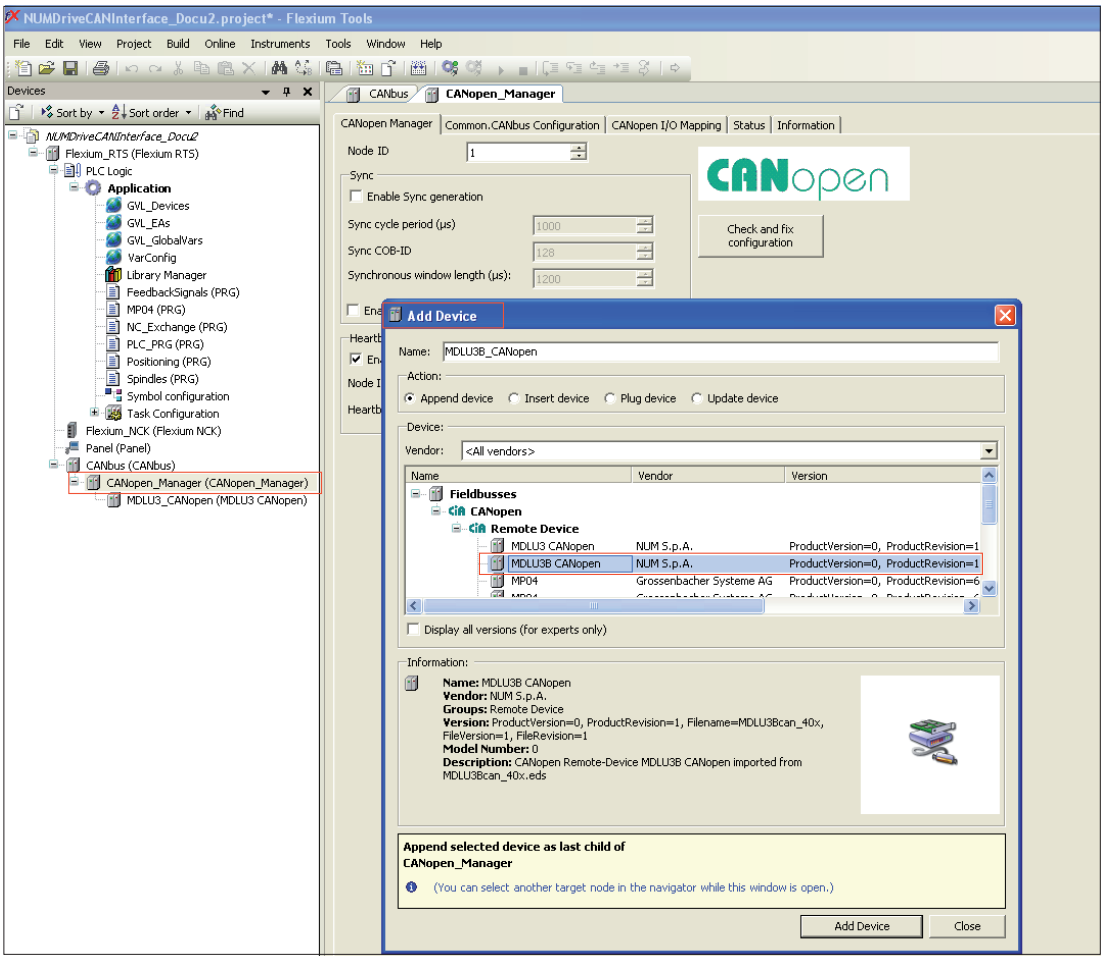
Independent from different busses or different devices all I/O addresses in FXTools must be unique.

3.6 Adding a NUMDrive C CANopen Bi-Axes device to the CANopen_Manager

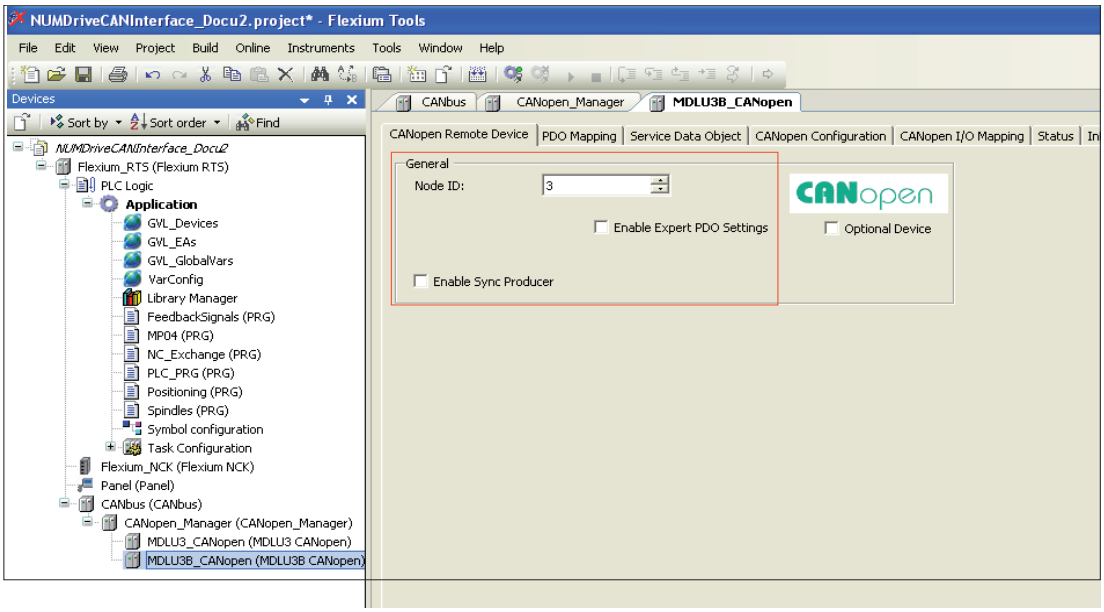
You can add a NUMDrive C CANopen device to the CANopen_Manager in the same way like you add the CANopen_Manager to the CANBus.

First make a right mouse-click on ☞ “CANopen_Manager” in the “Devices” window and select ☞“Add Device”.

After the “Add Device” window appears select ☞ “MDLU3 CANopen” (for Mono-Axis) or “MDLU3B CANopen” (for Bi-Axes) dependent of your application and the NUMDrive C CANopen device will be append to the “CANopen_Manager”.



Double click the ☞"MDLU3B_CANopen" in the "Devices" window and the "MDLU3B_CANopen" parameter register appears on the right side.



3.6.1 NUMDrive C CANopen Bi-Axes device settings

3.6.1.1 Register: CANopen Remote Device (Heartbeat producer - consumer)

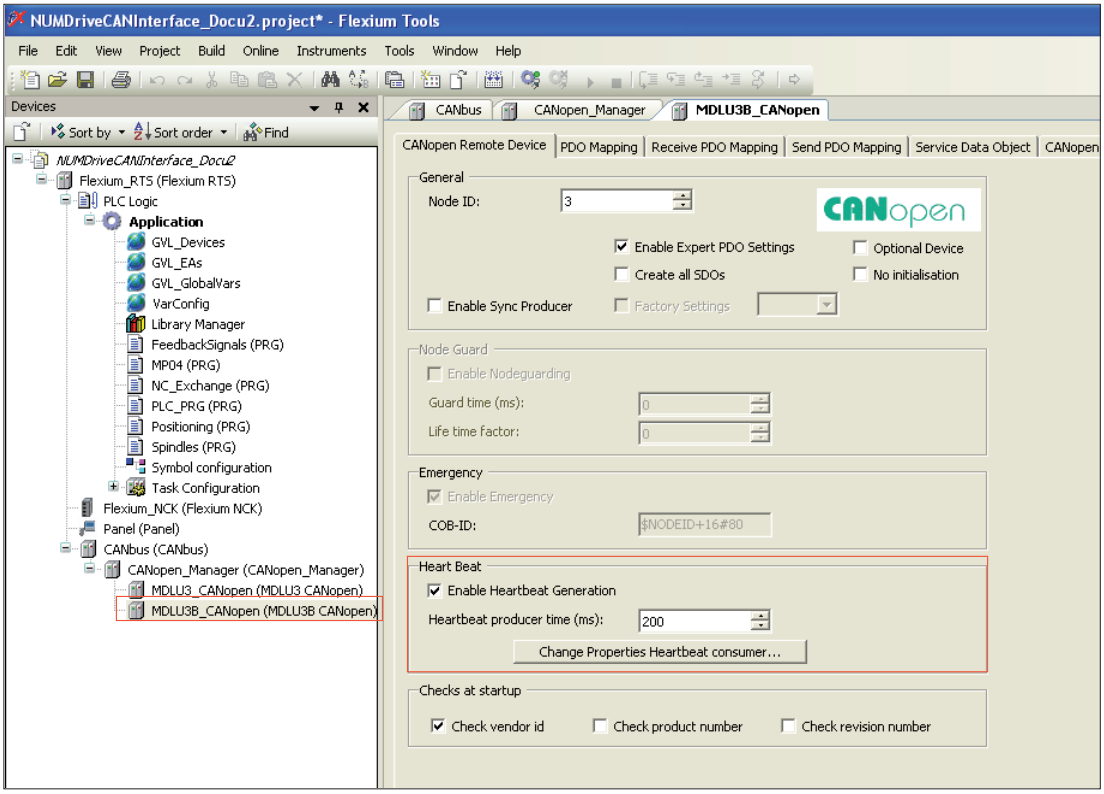
For this Register make following settings:

- NodeID: It is the hardware address of the device (the address that is set up with the dip switch S6 at the front of the drive. Please refer to the NUMDrive C Installation manual AMOMAN012.

- Enable Expert: Click here if you want to extend to more specific parameters.
  PDO Settings

Important note
NUMDrive C CANopen “MDLU3 Bi-Axes” Drive has only one hardware address and therefore only one NodeID.

Extended view if “Expert PDO Settings” is enabled



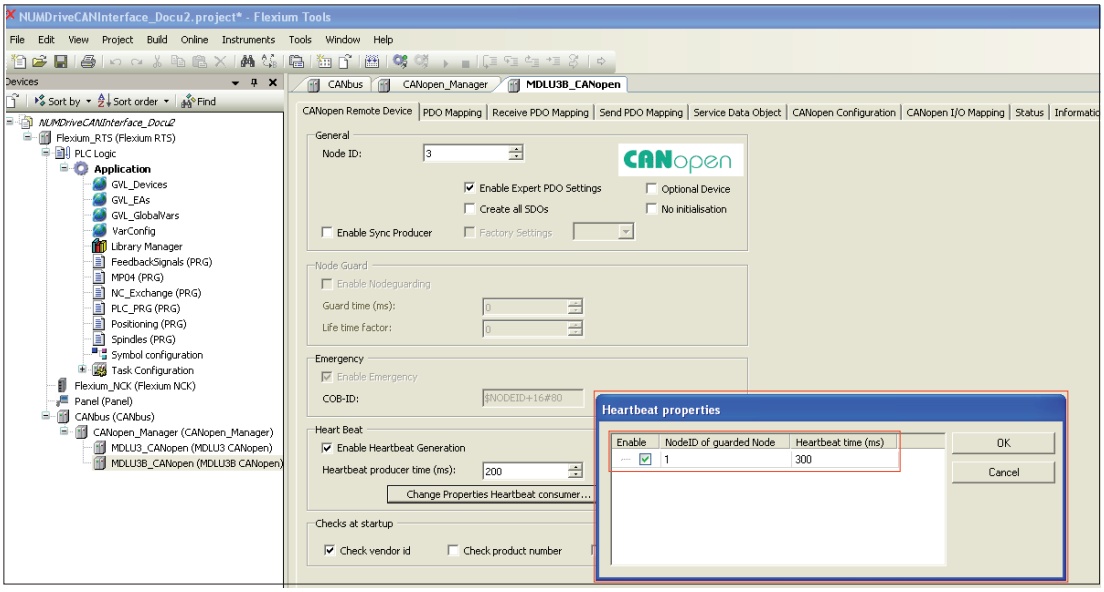
Extended settings

- Enable Heartbeat Generation : (if heartbeat generation is enabled the Master and other CANopen devices can control if the drive is still on the bus and “alive”.

- Heartbeat producer time: defines the time period for the heartbeat produced by the drive.

- Heartbeat consumer: defines that this drive controls (consumes) the Master heartbeat or a heartbeat that is produced by another device.
  To make this definitions click ☞“Change Properties Heartbeat Consumer”. See next page.

Heartbeat Consumer Settings



- Enable: Enables heartbeat consumer properties.
- NodeID: NodeID of the device which should be controlled.
- Heartbeat time: during this time period the heartbeat consumer expects a heartbeat (heartbeat producer time of the device that should be controlled x 1.5).

Explanation for this example

- The NUMDrive C CANopen (Addr.3) should control the Master (Addr.1) Heartbeat.
- The Master Heartbeat is set up with a producer time of 200ms.
- Consumer time should be 1.5 x producer time → 1.5 x 200ms = 300ms.

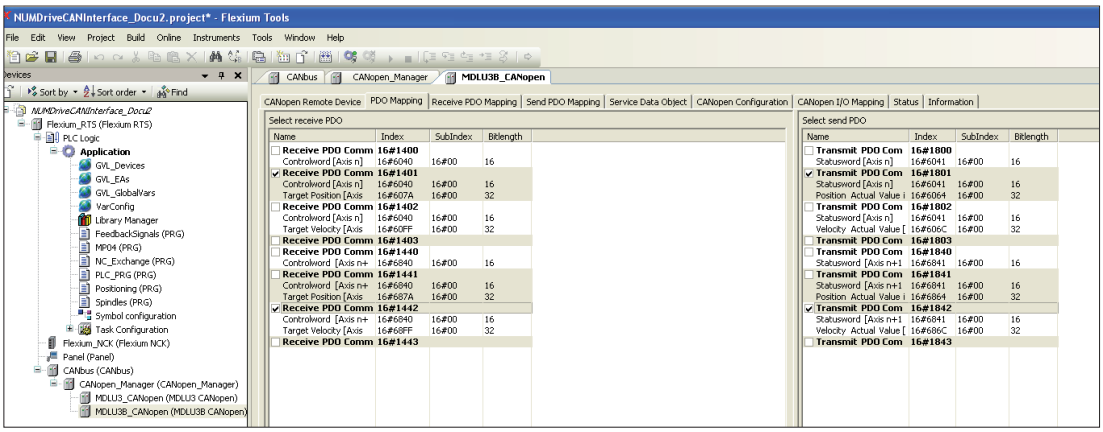
Reaction

If the NUMDrive C doesn't get a Master Heartbeat within 300ms, (in this example) the drive changes to error state.

3.6.1.2 Register: PDO Mapping (PDO selection for Velocity or Positioning Mode)

NUMDrive C CANopen drives provide two different function modes, velocity mode or positioning mode. Of course there are some differences for setting up and programming these two modes. The first choice you have to do is to enable the correct PDO dependent on the mode you want to use. With a Bi-Axes drive it is possible to select different modes for each internal axis.

- PDO 1-4 belong to Bi-Axes$_n$
- PDO 65-68 belong to Bi-Axes$_{n+1}$



Bi-Axes$_n$:

- For Positioning Mode: Enable Receive PDO2 and Transmit PDO2
- For Velocity Mode: Enable Receive PDO3 and Transmit PDO3

Bi-Axes$_{n+1}$:

- For Positioning Mode: Enable Receive PDO66 and Transmit PDO66
- For Velocity Mode: Enable Receive PDO67 and Transmit PDO67
- Disable all other PDO´s.

Important note
RPDO1 and TPDO1 for Bi-Axes$_n$, or RPDO65 and TPDO65 for Bi-Axes$_{n+1}$ include only the controlword that means if you use these PDO´s you will get no actual position or actual velocity information!

In this example is selected:

- Bi-Axes$_n$: RPDO2 and TPDO2 (Positioning Mode).
- Bi-Axes$_{n+1}$: RPDO67 and TPDO67 (Velocity Mode).

3.6.1.3 Register: PDO Mapping (PDO Properties - Transmission Type settings)

On a CANopen bus there is the possibility to define when and how the enabled PDO´s should be sent. For communication with a NUMDrive C CANopen the Transmission Type 255 (asynchronous device profile specific) is selected as default. To reach the parameter window for transmission types just double click ☞the PDO you want change.

It is also possible to change the COB-ID of the selected PDO in this window.
On a Mono-Axis it is not necessary to make a change of the COB-ID because it is foreseen that a CANopen device has 4 PDO´s (as standard). For this 4 PDO´s there are default COB-ID's available.

On a Bi-Axes drive there are 8 PDO´s available (of course there are never all 8 PDO´s enabled) and so we have to set up a COB-ID for the PDO´s > 4. It is recommended to use following COB-ID's:

PDO	**COB-ID**
Default COB-ID PDO 1-4	
TPDO1	16# 180 + NODE ID
TPDO2	16# 280 + NODE ID
TPDO3	16# 380 + NODE ID
TPDO4	16# 480 + NODE ID
RPDO1	16# 200 + NODE ID
RPDO2	16# 300 + NODE ID
RPDO3	16# 400 + NODE ID
RPDO4	16# 500 + NODE ID
Specific COB-ID PDO 65-68	
TPDO65	16# 1C0 + NODE ID
TPDO66	16# 2C0 + NODE ID
TPDO67	16# 3C0 + NODE ID
TPDO68	16# 4C0 + NODE ID
RPDO65	16# 240 + NODE ID
RPDO66	16# 340 + NODE ID
RPDO67	16# 440 + NODE ID
RPDO68	16# 540 + NODE ID

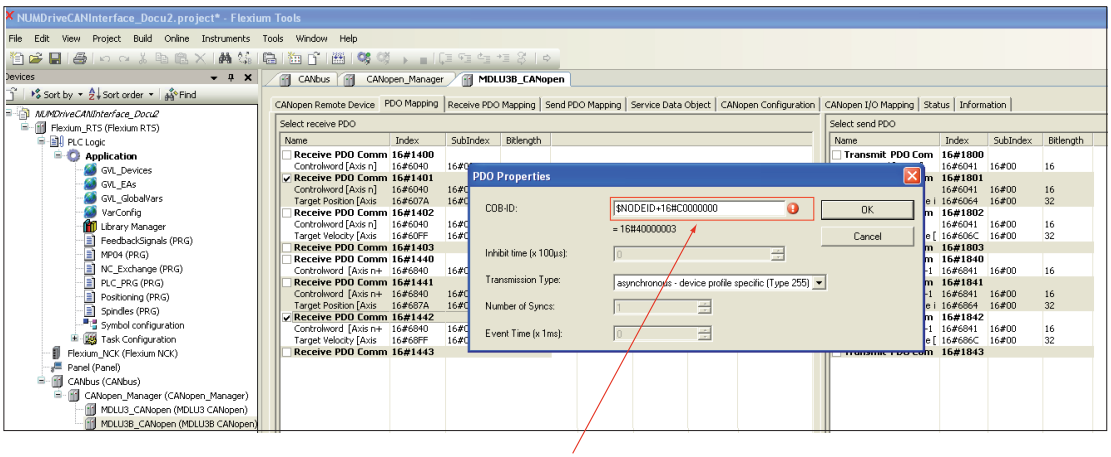
Important note
If there is a device with a NODE ID > 64 which uses PDO 1-4, there may be an overlapping COB-ID when following these recommended addresses. Please choose other addresses for your specific application in such a case, with respect to the CIA CANopen specifications.

Restricted CAN-IDs

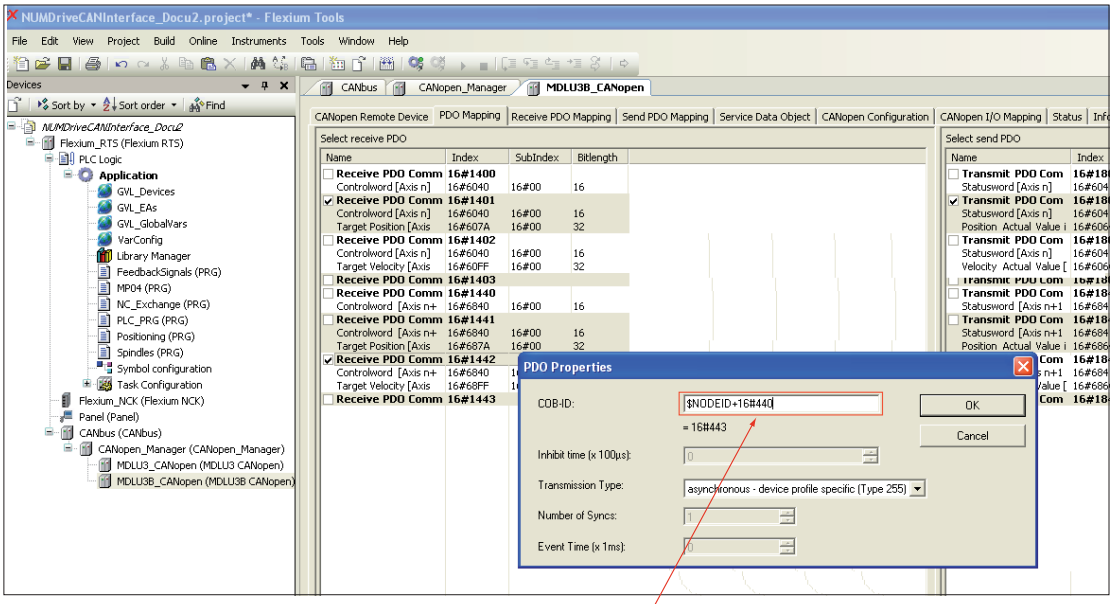
CAN-ID	**used by COB**
0 (000h)	NMT
1 (001h) - 127 (07Fh)	reserved
257 (101h) - 384 (180h)	reserved
1409 (581h) - 1535 (5FFh)	default SDO (tx)
1537 (601h) - 1663 (67Fh)	default SDO (rx)
1760 (6E0h) - 1791 (6FFh)	reserved
1793 (701h) - 1919 (77Fh)	NMT Error Control
2020 (780h) - 2047 (7FFh)	reserved

Change a COB-ID

To change COB-ID's double click ☞the PDO you want to modify and insert a correct address.

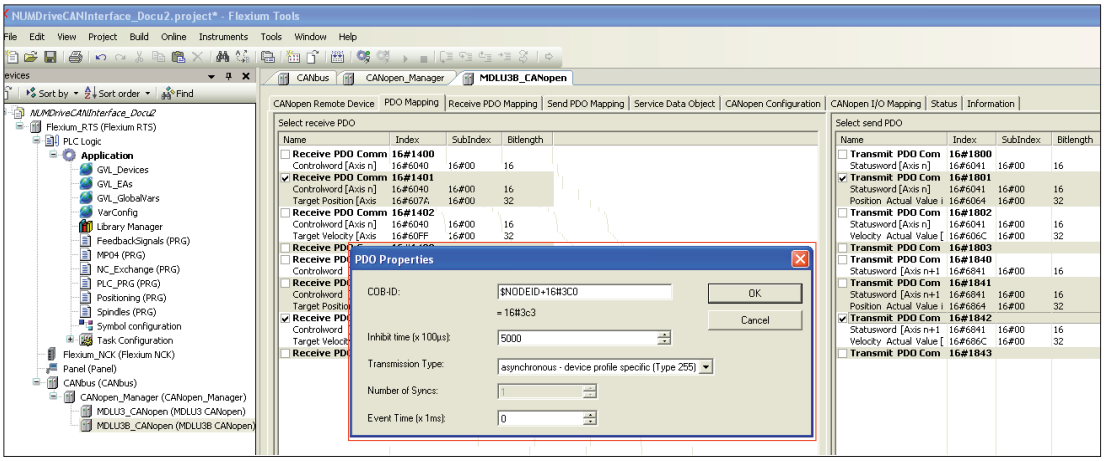


If you see a red exclamation mark the COB-ID is invalid



Change to a valid address
In this example: NODE ID + 16# 440 for RPDO 67

Transmission Type behaviour on Transmit PDO's



Transmission Type 255 is asynchronous that means the PDO gets send when there is a change in the PDO value. For a Transmit PDO Type 255 you can change following behaviour:

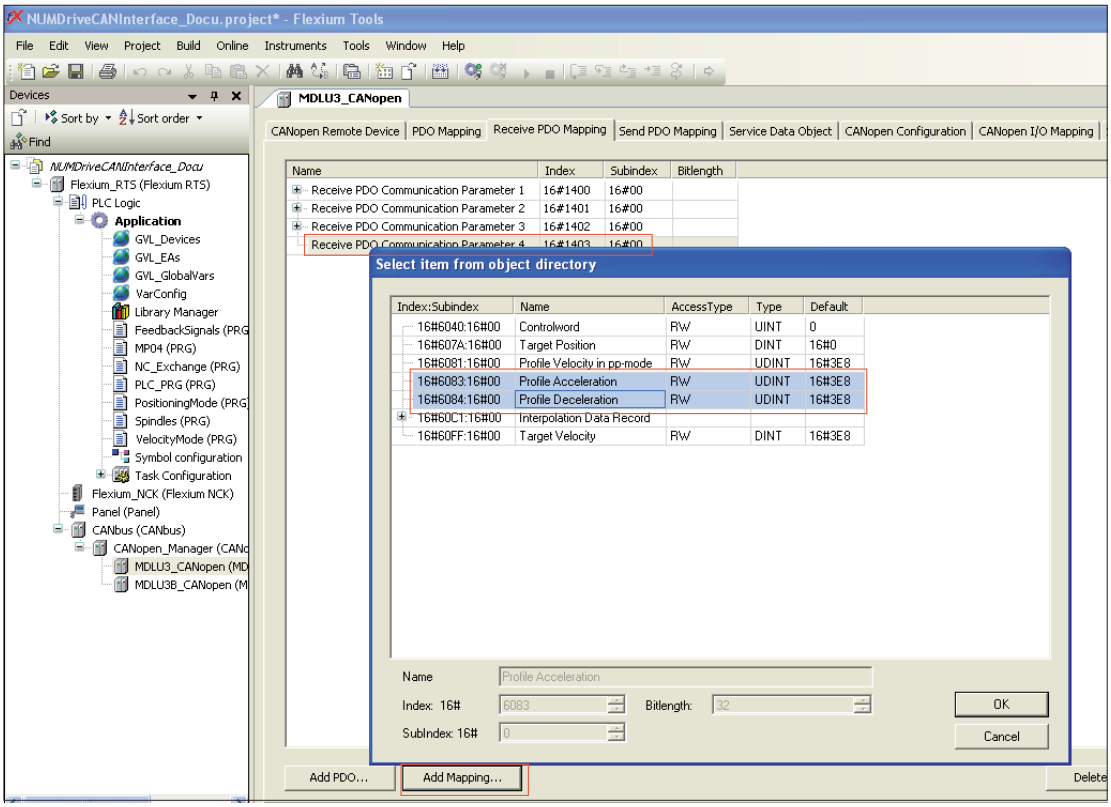
- COB-ID: modify according to the table above if necessary.
- Inhibit Time: inhibits the sending of the next PDO until the inserted time period is reached (only if there was a PDO value change before).
- Event Time: is the opposite of Inhibit Time, the PDO is send after this time also if there was no PDO value change before.

It is possible to make different Transmission Type settings for each enabled PDO! For receive PDO´s there is no inhibit or event time!.

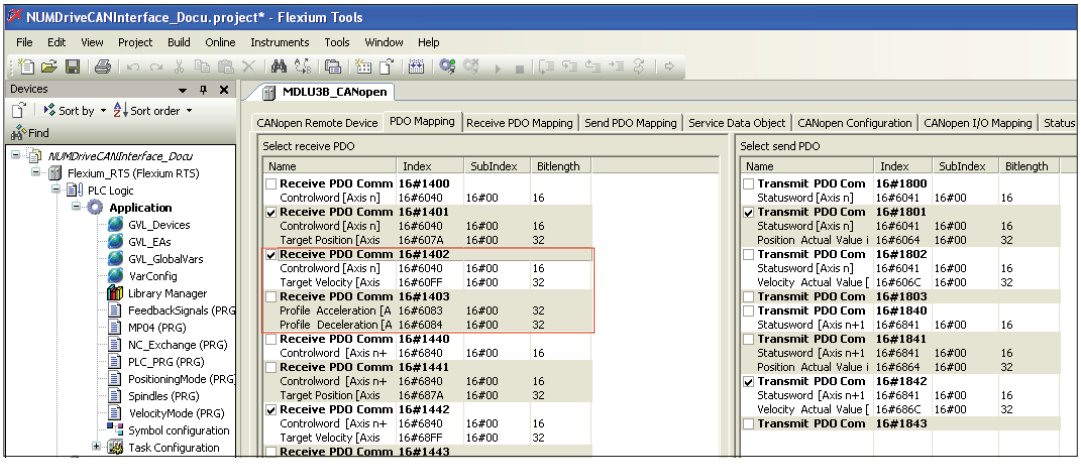
3.6.1.4 Register: Receive PDO Mapping (Change RPDO Mapping)

It is also possible to change the content of a PDO. For example if you want to change the acceleration and deceleration a lot of times you can use PDO4 (or 68) (which are empty by default) to map the acceleration and deceleration object into it.

For this example select PDO 4 and click to the ☞ “Add Mapping” button.



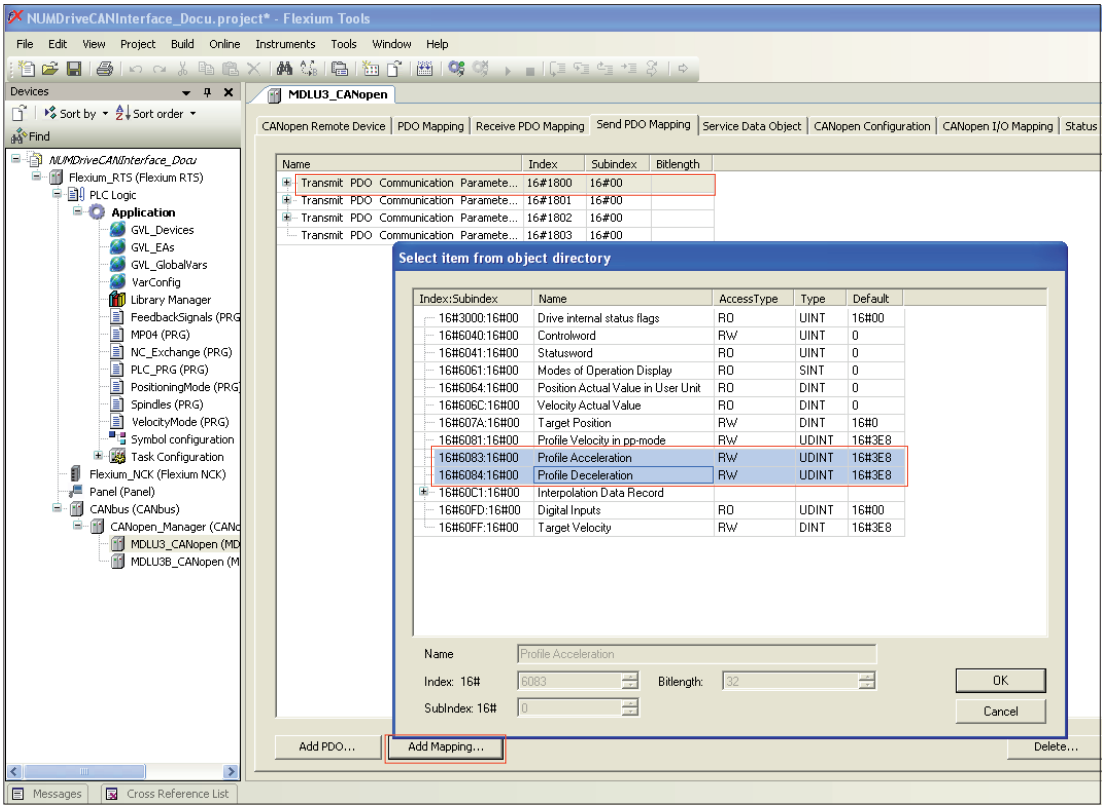
- In the appearing window you select the objects you want to receive (and remember receive PDO´s are received by the CANopen device) and click ☞ OK.
- Now change the register to ”PDO Mapping” and you can see that PDO4 has the content you have selected.
- Now you only must enable PDO4 to use it.
  Don’t forget to make a I/O Mapping also for PDO4 like for the other enabled PDO’s.
  Please refer to the chapter: 3.5.1.7. Register: CANopen I/O Mapping (Setting up the I/O Addresses).



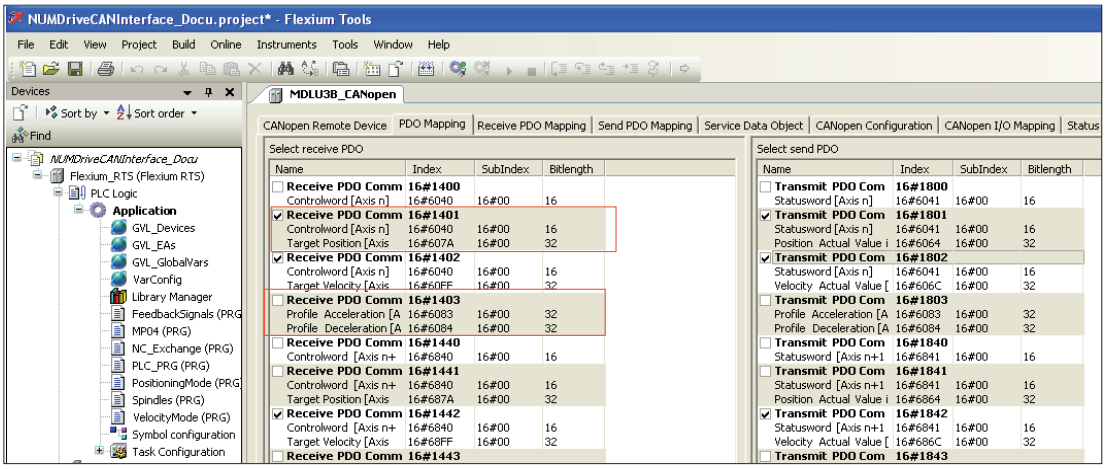
3.6.1.5 Register: Send PDO Mapping (Change TPDO Mapping)

Like for receive PDO's you can also change the content of transmit PDO's.
In this example we want to see if our change of acceleration or deceleration is reflected by the drive.

For this select PDO 4 and click to the ☞ "Add Mapping" button.



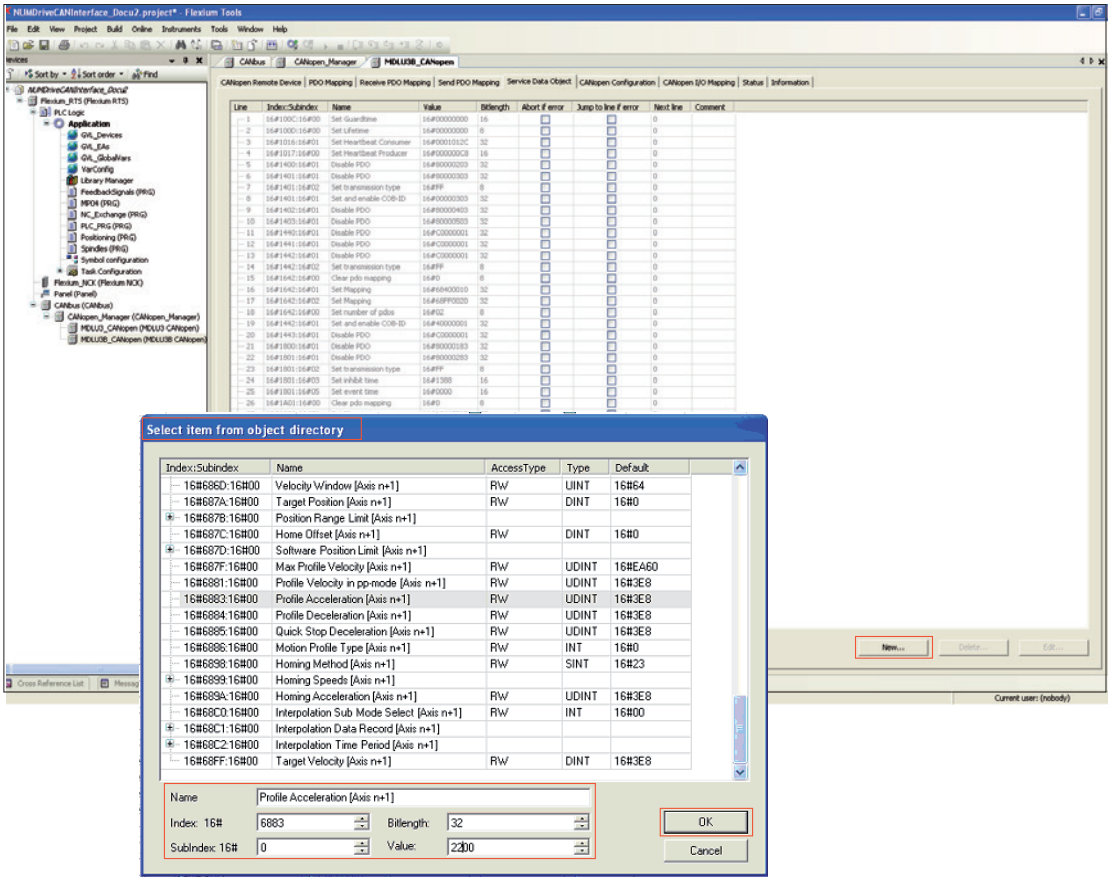
- In the appearing window you select ☞the objects you want to send (and remember, send PDO´s are sent by the CANopen device) and click ☞OK.
- Now change the register to "PDO Mapping" and you can see that PDO4 has the content you have selected.
- Now you only must enable PDO4 to use it. Don't forget to make a I/O Mapping also for PDO4 like for the other enabled PDO's. Please refer to the chapter 3.5.1.7. Register: CANopen I/O Mapping (Setting up the I/O Addresses)



3.6.1.6 Register: Service Data Object (Add or Remove SDO's & Change SDO values)

Some NUMDrive C CANopen device parameter must be changeable by SDO according to CAN specification. There are several parameters which were removed from WPM access and transferred to SDO communication.
The procedure to add a specific SDO that's needed and change the SDO value with FXTools is always the same. Below the default (not removable) SDO's which are declared in the appropriate "***.eds**" file.

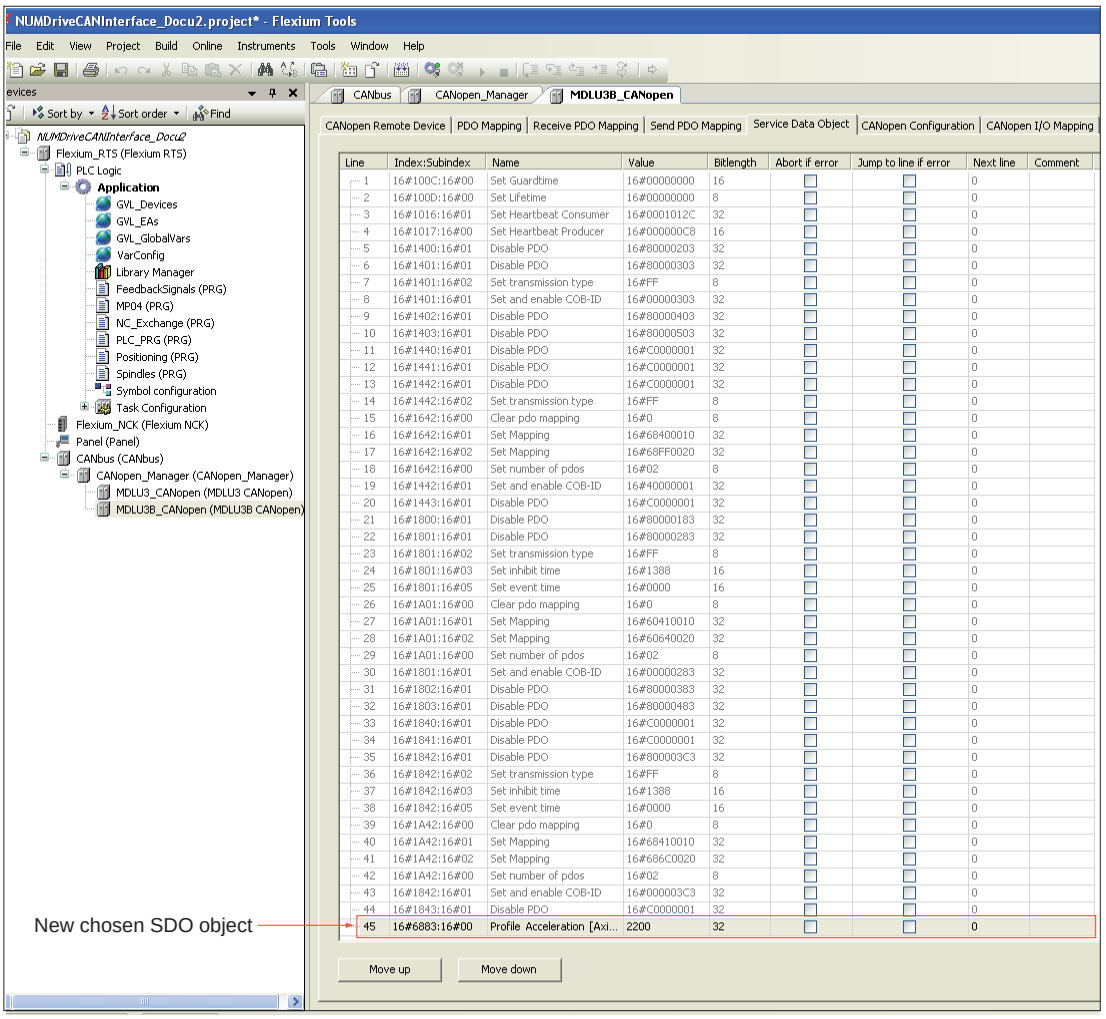
Expanding this default SDO's, which are transmitted during the boot up of the PLC, is possible by clicking the ☞ "New" button. The "Select item from object dictionary" window appears.
Here you can select a SDO object from the object dictionary on your demand.



To change a value, just ☞ edit the value textbox and confirm.
As result your chosen SDO object will be inserted to the SDO object overview window as indicated in the next page.

SDO's with addresses 16#60xx belong to Bi-Axes$_n$
SDO's with addresses 16#68xx belong to Bi-Axes$_{n+1}$.

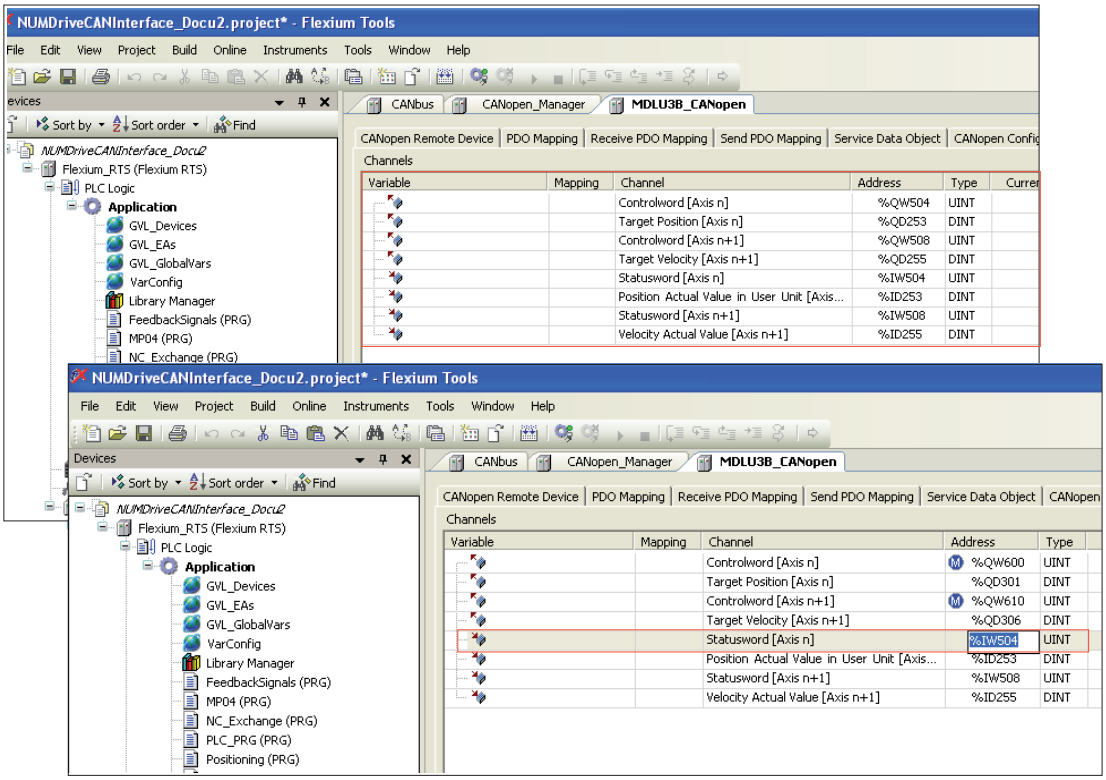
Now, also this SDO object is transmitted during the boot up of the PLC but not immediately after adding!



3.6.1.7 Register: CANopen I/O Mapping (Setting up the I/O Addresses)

To establish a communication between PLC and NUMDrive C CANopen you must define I/O address areas. Dependent on the SDO selection you made before (see: Chapter 3.6.1.2 Register: PDO Mapping) you must assign the objects, which are declared to the selected PDO´s, to unique PLC addresses. As default, FXTools take the next free address and assigns it to the enabled object.

Often this behaviour is not favored. To change the addresses to your choice, double click directly into the ☞ “Address” cell and it will get editable.

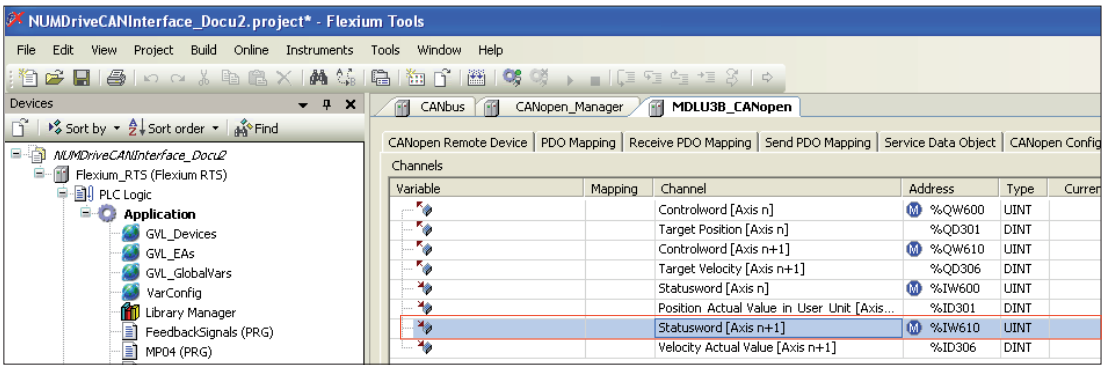


Important note
Remember one TPDO for velocity or positioning includes one controlword (uint) and one target value (dint) and one RPDO for velocity or positioning includes one statusword (uint) and one actual value (dint)!

If you change the respectively first of these two objects (which gets marked with the blue “M”) the address for the other object is getting actualized automatically to the next possible address.
You are not allowed to change the respectively second of these two objects because the library handles one PDO as one package with one address. So never change the second entry!

The picture below shows a correct configuration example.

Each time the first object address of a PDO is changed manually and marked with a blue “M” and the second object address is calculated automatically without a blue “M” mark.



Important note
Independent from different busses or different devices all I/O addresses in FXTools must be unique!

3.7 NUMDriveCAN Interface library – data types and structure

3.7.1 Important data types

Before we will use the “NUMDriveCANInterface” Library to control a “NUMDrive C CANopen” device we should take a closer look at the data types that are used to make several setting or realize parameter exchange.

3.7.1.1 “DutInpNUMDriveCAN” & “DutOutNUMDriveCAN”

In the previous chapter we made the I/O mapping for the drive controlword, statusword etc.
When we will start programming our application in the next chapter, the first thing we will have to do is to declare the addresses we mapped to the in- and outputs to the function block instance which we need to control the drive.

We will use a functionality that is called “VAR_Config” to make this operation.
The “VAR_Config” functionality maps an address to an internal variable in the function block instance.

The data type which is declared at the “VAR_Config” functionality must fit to the data type in the function block instance.
The data type for the input address area is “DutInpNUMDriveCAN” and the data type for the output address area is “DutOutNUMDriveCAN”.

3.7.1.2 “EnumNumDriveCan_Mode”

With the enumeration “EnumNumDriveCan_Mode” you can change the operating mode of the drive. There are 4 entries available.

Important note
The function “Homing” is reserved for “Positioning Mode” only!

VAR_GLOBAL EnumNUMDriveCan_Mode

Name	Type	Inherited from	Address	Initial	Comment
NoMode	ENUMNUMDRIVECAN_MODE			0	
Manual	ENUMNUMDRIVECAN_MODE			1	
Auto	ENUMNUMDRIVECAN_MODE			2	
Homing	ENUMNUMDRIVECAN_MODE			3	

3.7.1.3 “EnumNumDriveCan_Select”

With the enumeration “EnumNumDriveCan_Select” you can choose the type of the drive you want to control. There are 3 entries available.

VAR_GLOBAL EnumNUMDriveCan_Select

Name	Type	Inherited from	Address	Initial	Comment
MonoAxis	ENUMNUMDRIVECAN_SELECT			1	
BiAxis_n	ENUMNUMDRIVECAN_SELECT			2	
BiAxis_n1	ENUMNUMDRIVECAN_SELECT			3	

3.7.1.4 "EnumNumDriveCan_HomingMethod"

With the enumeration "EnumNumDriveCan_HomingMethod" you can choose the method of the homing functionality in "Positioning Mode" if demanded. There are 3 entries available.

VAR_GLOBAL EnumNUMDriveCAN_HomingMethod

Name	Type	Inherited from	Address	Initial	Comment
PosRefDir	ENUMNUMDRIVECAN_HOMINGMETH...			1	
NegRefDir	ENUMNUMDRIVECAN_HOMINGMETH...			2	
EqualsExtendedSdoSettings	ENUMNUMDRIVECAN_HOMINGMETH...			3	

Entry number 3 means that the homing method is set with a SDO parameter transmission which includes an extended list of homing methods.

3.7.1.5 "EnumNumDriveCan_Param"

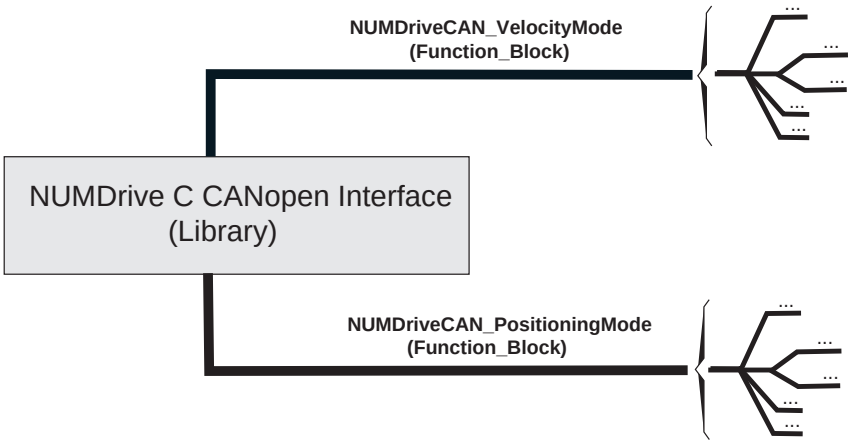
With the enumeration "EnumNumDriveCan_Param" you can choose parameter you want to transmit to the drive when using the "SDO Access Functions" which are integrated in the "NUMDriveCANInterface" library. There are 21 entries (parameters) available. Some are only for internal use of the library and are not further documented in this manual.

ENUM EnumNUMDriveCan_Param

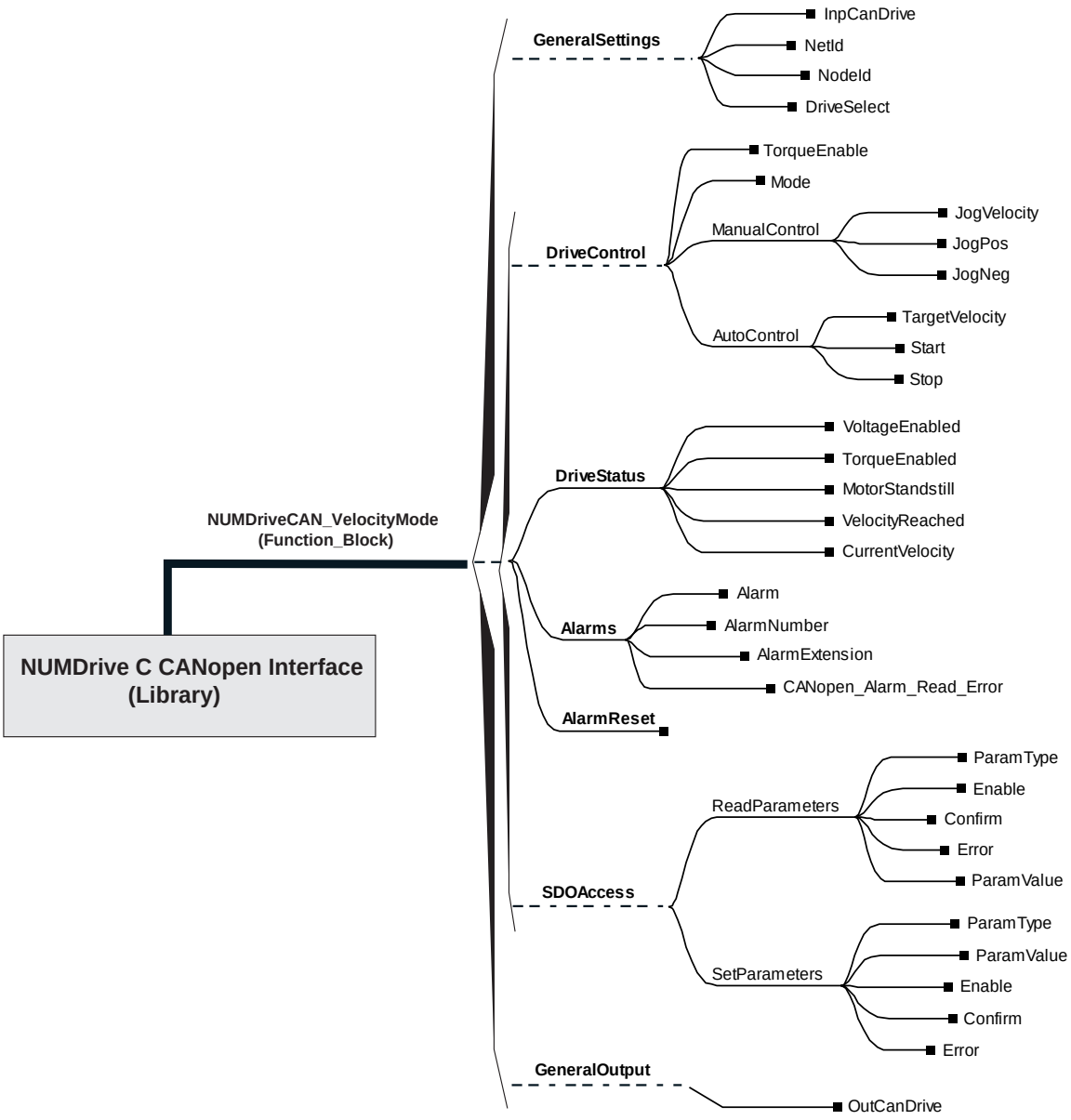
Name	Datentyp	Geerbt von	Adresse	Initialwert	Kommentar
FollowingErrorWindow	INT			1	
PositionWindow	INT			2	
PosRangeLimitMin	INT			3	
PosRangeLimitMax	INT			4	
SoftPosLimitMin	INT			5	
SoftPosLimitMax	INT			6	
Velocity	INT			7	
Acceleration	INT			8	
Deceleration	INT			9	
MaxVelocity	INT			10	
HomeOffset	INT			11	
HomingSpeedFast	INT			12	
HomingSpeedSlow	INT			13	
HomingAcDeceleration	INT			14	
HomingMethod	INT			15	
InternalFlags	INT			16	
ModeInternal	INT			17	
ModeInternal2	INT			18	
RelativMoveFlag	INT			19	
ShadowControlWordSynch	INT			20	
TargetPosition	INT			21	

3.8 Library structure

3.8.1 Overview Library Function Blocks



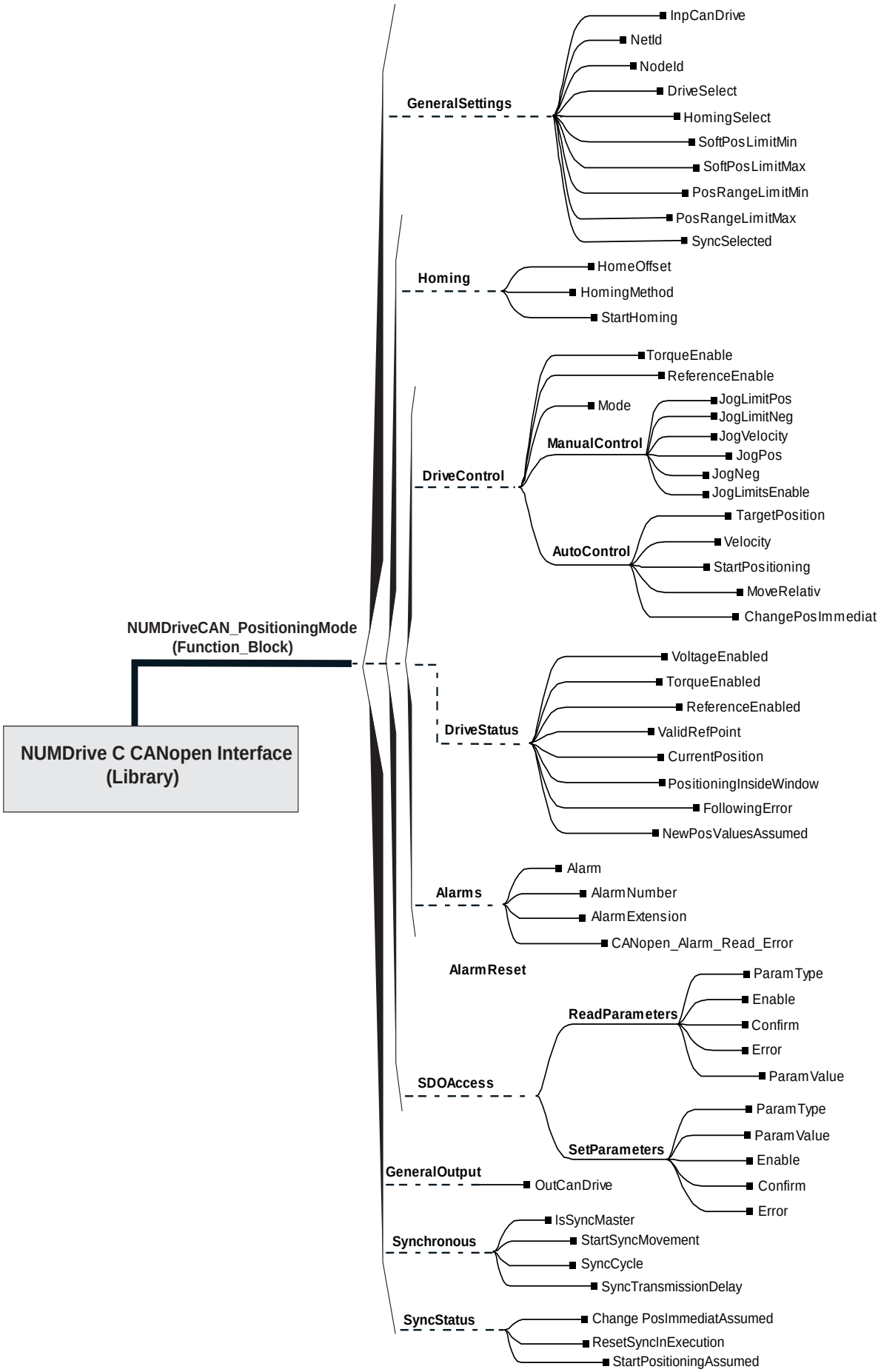
3.8.2 NUMDriveCAN_VelocityMode Structure



3.8.3 Parameters / Variables Description Velocity Mode

NUMDriveCAN_VelocityMode		
Name	**Data type**	**Description**
InpCanDrive	STRUCT (DutInpNUMDriveCAN)	VAR_Config input for input address assignment
NetID	USINT	CANopen interface used on FS1xi / FS192i or Box PC: - CAN1 = NetID 0 - CAN2 = NetID 1
NodeId	USINT	Device hardware address
DriveSelect	ENUM (EnumNUMDriveCan_Select)	Drive Type (Mono Axis, Bi-Axes part n, Bi-Axes part n1)
TorqueEnable	BOOL	Drive torque enable
Mode	ENUM (EnumNUMDriveCan_Mode)	Mode selection (Manual, Auto, NoMode, Homing only for internal use)
JogVelocity	UDINT	Velocity for Jog mode
JogPos	BOOL	Starts the Jog movement in positive direction
JogNeg	BOOL	Starts the Jog movement in negative direction
TargetVelocity	REAL	Velocity in Automatic mode
Start	BOOL	Starts movement in Automatic mode
Stop	BOOL	Stops movement in Automatic mode
VoltageEnabled	BOOL	Status signal from drive – voltage enabled
TorqueEnabled	BOOL	Status signal from drive- torque enabled
MotorStandstill	BOOL	Status signal from drive – motor at standstill
VelocityReached	BOOL	Status signal from drive – target velocity reached
CurrentVelocity	REAL	Status signal from drive – actual velocity
Alarm	BOOL	An alarm has occurred on the drive
AlarmNumber	WORD	Alarm number from drive
AlarmExtension	WORD	Alarm extension number from drive
CANopenAlarm_Read_Error	CANOPEN_KERNEL_ERROR	Read Error from CANopen stack
AlarmReset	BOOL	Resets alarm message
ParamType	ENUM (EnumNUMDriveCan_Param)	Parameter type definition for SDO transmission
Enable	BOOL	Enables SDO transmission
Confirm	BOOL	SDO transmission confirm
Error	BOOL	SDO transmission error
ParamValue	DINT	Value for read/write access
OutCanDrive	STRUCT(DutOutNUMDriveCAN)	VAR_Config output for output address assignment

3.8.4 NUMDriveCAN_PositioningMode Structure



3.8.5 Parameters / Variables Description Positioning Mode

NUMDriveCAN_PositioningMode		
Name	**Data type**	**Description**
InpCanDrive	STRUCT (DutInpNUMDriveCAN)	VAR_Config input for input address assignment
NetID	USINT	CANopen interface used on FS1xi / FS192i or Box PC: - CAN1 = NetID 0 - CAN2 = NetID 1
NodeId	USINT	Device hardware address
DriveSelect	ENUM (EnumNUMDriveCan_Select)	Drive Type (Mono Axis, Bi-Axes part n, Bi-Axes part n1)
HomingSelect	BOOL	Enables the internal homing function
SoftPosLimitMax	REAL	Maximum software position limit
SoftPosLimitMin	REAL	Minimum software position limit
PosRangeLimitMax	REAL	Maximum position Range (for Modulo)
PosRangeLimitMin	REAL	Minimum position Range (for Modulo)
SyncSelected	BOOL	Enable Synchronous Transmission
HomeOffset	REAL	Position after homing point is reached
HomingMethod	ENUM (EnumNUMDriveCan_ HomingMethod)	Defines the way how the homing function is executed
StartHoming	BOOL	Starts the homing procedure (motion)
TorqueEnable	BOOL	Drive torque enable
ReferenceEnable	BOOL	Drive reference enable
Mode	ENUM (EnumNUMDriveCan_Mode)	Mode selection (Manual, Auto, NoMode, Homing only for internal use
JogLimitPos	REAL	Lower Software Pos Limit in JOG
JogLimitNeg	REAL	Upper Software Pos Limit in JOG
JogVelocity	UDINT	Velocity for Jog mode
JogPos	BOOL	Starts the Jog movement in positive direction
JogNeg	BOOL	Starts the Jog movement in negative direction
JogLimitsEnable	BOOL	Enable Jog Pos Limits
TargetPosition	REAL	The target position after the next start
Velocity	UDINT	The velocity for the next positioning task
StartPositioning	BOOL	Starts movement in Automatic mode
MoveRelativ	BOOL	Enables the relative motion
ChangePosImmediat	BOOL	The current position and velocity values are executed immediately
VoltageEnabled	BOOL	Status signal from drive - voltage enabled
TorqueEnabled	BOOL	Status signal from drive- torque enabled
ReferenceEnabled	BOOL	Status signal from drive - reference enabled
ValidRefPoint	BOOL	Drive has valid reference point
CurrentPosition	REAL	Status signal from drive - actual position
PositioningInside Window	BOOL	Current Position is inside defined Range (Window) of the last target Position
FollowingError	BOOL	Status signal from drive – Drive detected a following error
NewPosValuesAssumed	BOOL	Drive has assumed the new positioning values
Alarm	BOOL	An alarm has occurred on the drive
AlarmNumber	WORD	Alarm number from drive
AlarmExtension	WORD	Alarm extension number from drive

CANopenAlarm_Read_Error	CANOPEN_KERNEL_ERROR	Read Error from CANopen stack
AlarmReset	BOOL	Resets alarm message
ParamType	ENUM (EnumNUMDriveCan_Param)	Parameter type definition for SDO transmission
Enable	BOOL	Enables SDO transmission
Confirm	BOOL	SDO transmission confirm
Error	BOOL	SDO transmission error
ParamValue	DINT	Value for read/write access
OutCanDrive	STRUCT (DutOutNUMDriveCAN)	VAR_Config output for output address assignment
IsSyncMaster	BOOL	Enable this device as Master for thes synch signal in the synch network
StartSyncMovement	BOOL	Starts the synchronous movement of the commanded drive in the synch network
SyncCycle	UINT	Synch Cycle periode in ms
SyncTransmittionDelay	TIME	Internal time constant - default 200ms - Enlarge this value only if there are many axes and the feedback signals “StartPositioningAssumed” or “ChangePosImmediatAssumed” of all axes are not present as expected.
Change PosImmediat Assumed	BOOL	Feedback signal - “ChangePosImmediat“ command was sent to drive. Drive is ready to StartSyncMovement with the next “StartSyncMovement“ command.
ResetSyncInExecution	BOOL	After a Reset in Sync Mode, the drive needs some time to run through the reset. During the internal Reset procedure is active it is not allwed to set any other commands. This bit is TRUE during the internal Reset procedure is running.
StartPositioning Assumed	BOOL	Feedback signal - “StartPositioning“ command was sent to drive. Drive is ready to StartSyncMovement with the next “StartSyncMovement“ command.

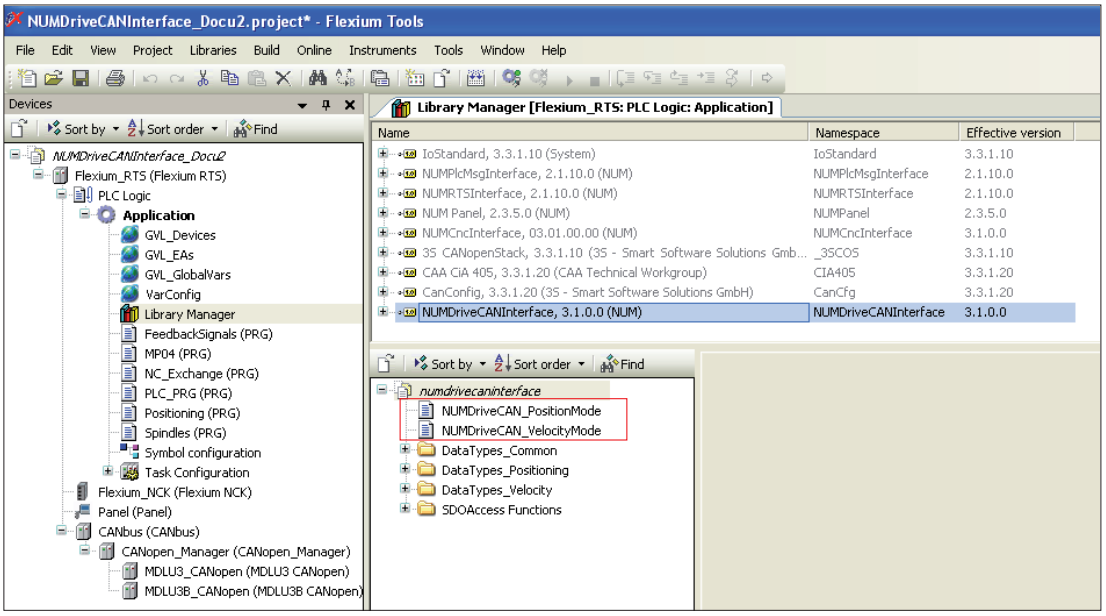
3.9 “Velocity Mode” in NUMDriveCANInterface library

3.9.1 Instancing the “Velocity Mode” function block

By double clicking the “Library Manager” an overview window appears on the right side which displays all installed libraries. If a library gets selected a list with all integrated function blocks, data types, methods etc. as shown below.

Selecting the “NUMDriveCANInterface” library shows two function blocks and all data types that are used internally. The data type listing is useful if the programmer wants to know what kind of data type must be declared for in or output parameter to communicate with the library.D. The function block listings gives information about the available function blocks which can be instanced.

In the “NUMDriveCANInterface” library there are two function blocks available. With instancing one of this function blocks you declare which mode (Velocity or Positioning) you want to use.

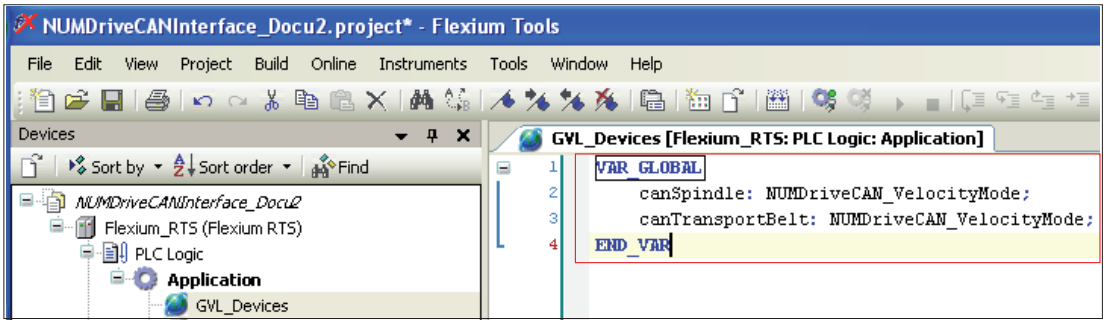


To realize an interaction with the drive in your whole project, we recommend to create the instance in a GVL. In this state of your project (making an instance of the mode function block) you don’t need to differentiate between Mono-Axis and Bi-Axes.
How to tell the instance which type of drive it shall drive will be explained later in this chapter.

In this example:

- canSpindle → is the Mono-Axis Drive (Velocity Mode)
- canTransportBelt → is the Bi-Axes Drive$_{n+1}$ (Velocity Mode)

(As declared in Chapter 3.5.1.2 (Mono-Axis) and Chapter 3.6.1.2 (Bi-Axes))
Later in this documentation we will program the Bi-Axes Drive$_{n}$ as Positioning Mode Axis.



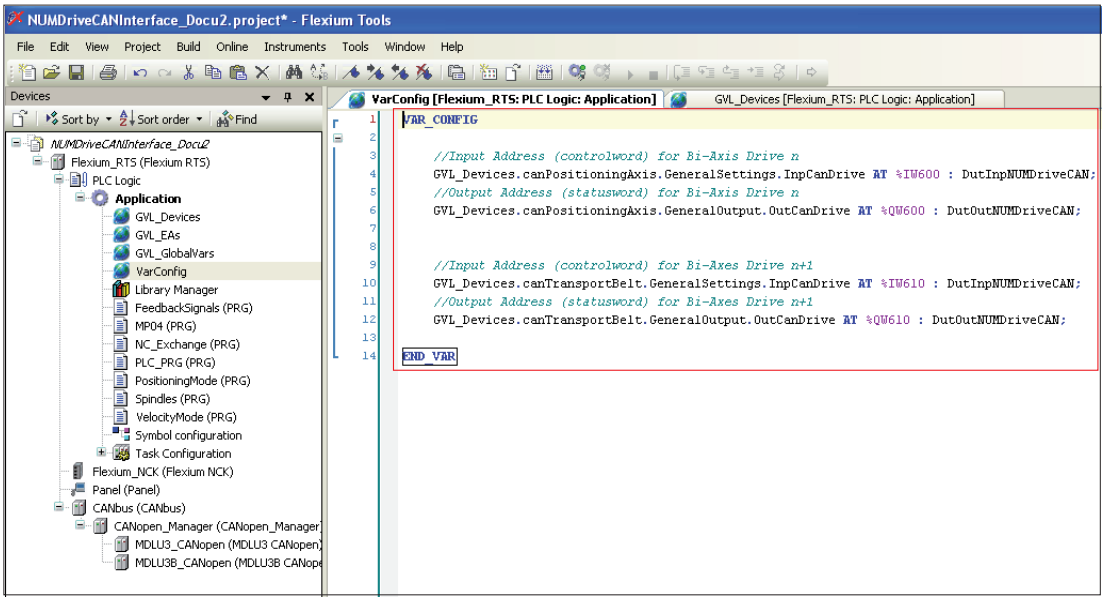
The instances “canSpindle” and “canTransportBelt” are now available in your whole project and include all functions of the “NUMDriveCAN_VelocityMode” function block to control a “NUMDrive C CAN” in Velocity Mode.

3.9.2 “VAR Config” for “NUMDriveCANInterface” instances

To make it possible for the library instance to communicate with a drive, it is necessary to declare the status word and control word to the library instance first.

In the chapter 3.5.1.7 (Mono-Axis) or chapter 3.6.1.7 (Bi-Axes) we have already declared I/O addresses to the status and control words of the different drives. The preferred way to map this declared addresses to the library instance is to use the “VAR Config” functionality.

“VAR Config” maps an address area (beginning with the starting address number) directly into a library instance to the foreseen variable structure.
To make a “VAR Config” you must create a GVL and change the GVL Keywords to “VAR_Config”.
Then you have to map the I/O addresses to the foreseen variables in the library instance like displayed in the picture below.

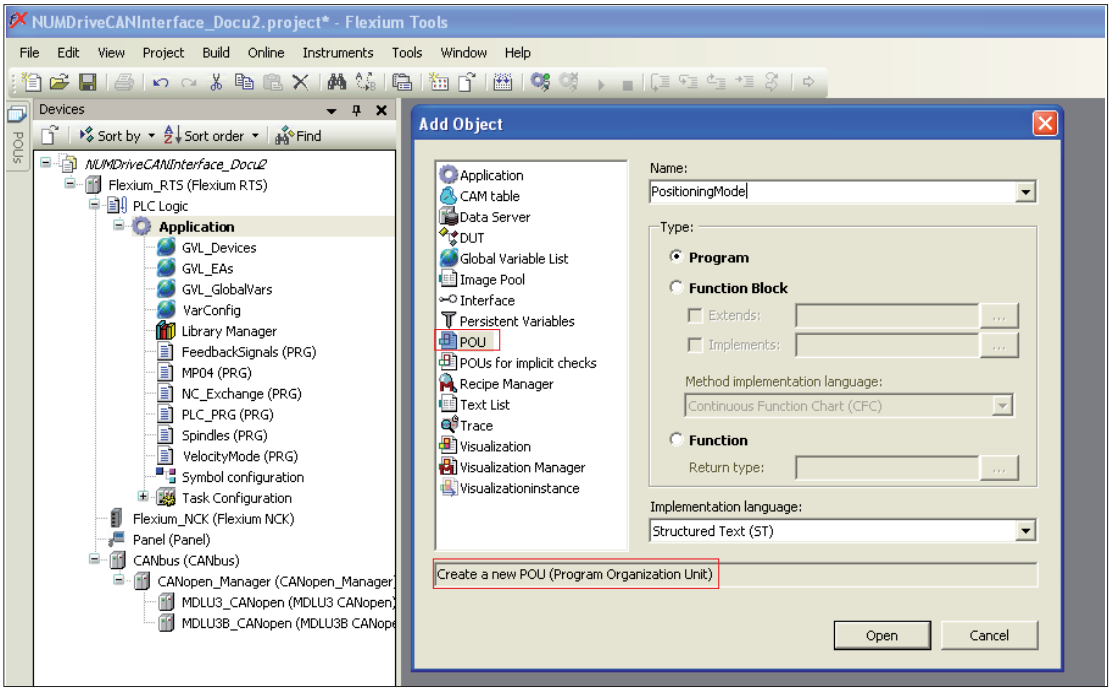


In this state of your project (making the “VAR_Config”) you don’t need to differentiate between Mono-Axis and Bi-Axes. How to tell the instance which type of drive it shall drive will be explained later in this chapter.

3.9.3 Using an instance in a POU

The fact that we made global instances of the “VelocityMode” function block makes it very comfortable to use these instances (canSpindle and canTransportBelt) in all POU’s (Program Organization Unit) of our project.

The only thing we have to do, is to add a new POU, define it as “Program” in “Structured Text” and give it a meaningful name.

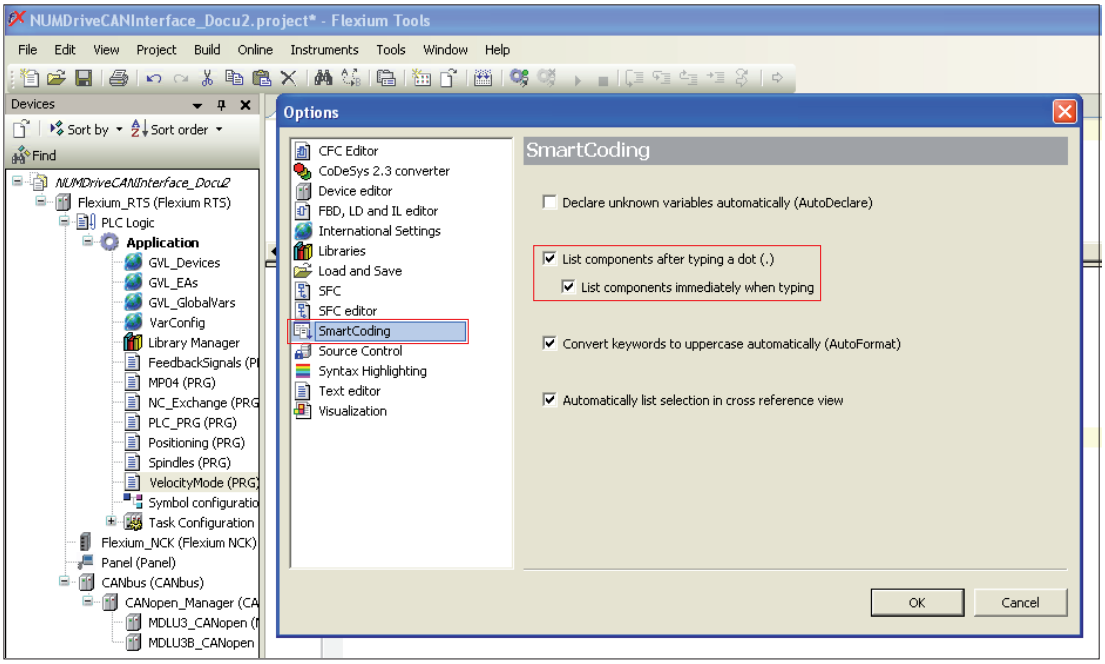


After this step we can start to program the interactions between our PLC and the drives by using the instances of the “VelocityMode” function block (canSpindle and canTransportBelt).

3.9.4 IntelliSense

For the further work we recommend to enable the CoDeSys IntelliSense functionality to see the library structure after typing a dot (.)

To enable IntelliSense click on Tools → Options → Smart Coding.

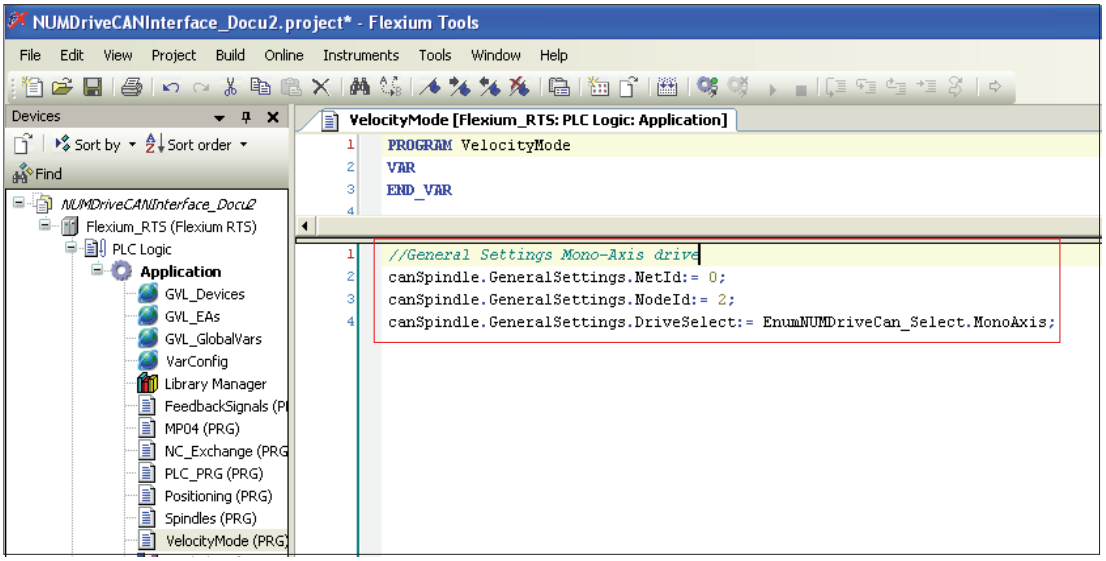


Also check other options to personalize your Flexium Tools surface and behaviour on your demands

3.9.5 General Settings

Some more additional important information must be written to the instances by parameters. These parameters are needed to make it clear for the instance, to which “Net” (interface number) it shall send their commands, the NodeID (hardware address) of the target drive and the drive type (Mono-Axis or Bi-Axes) as well as the internal drive part in case of a Bi-Axes (n or n+1).

For these important settings we use the parameters that are nested under “GeneralSettings”.

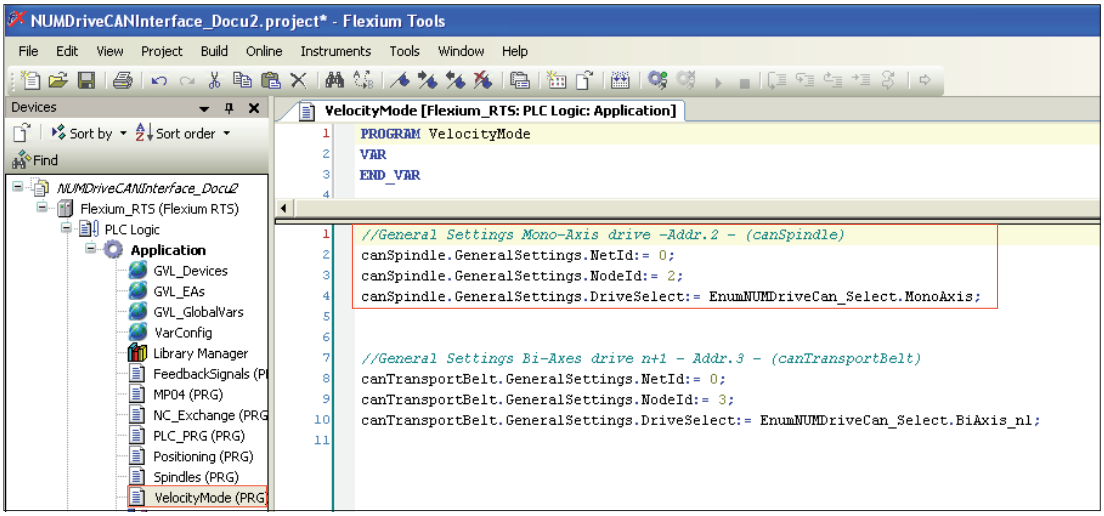


In this example the “NUMDrive C CANopen” which drives the “CAN Spindle” is connected to Net “0” (like the setting we made in Chapter 3.3 Adding a CANBus to the Flexium_RTS)
It has the hardware address “2” and the “NUMDrive C CANopen” is a Mono-Axis drive.

3.9.6 Mono-Axis and Bi-Axes distinction

In this example we declared also a Bi-Axes drive on Net “0” with NodeID (hardware address) “3”. The first internal part of the Bi-Axes drive (n) we declared for “Positioning Mode” (in PDO mapping) and the second internal drive part (n+1) we declared for “Velocity Mode” (in PDO mapping).

The instance for the internal drive part (n+1) we have already made (canTransportBelt) as well as the “VAR_Config”. The next step is to make the “General Settings” for this prepared instance.



As we can see in the picture above, the only difference between Mono-Axis and Bi-Axes drives, is to set the important parameter “DriveSelect” to the correct enumeration value with respect to the target drive type (Mono, Bi-n, Bi-n1) we want to communicate to.

All following settings, functions, and program parts are absolutely the same for Mono-Axis and Bi-Axes drives.

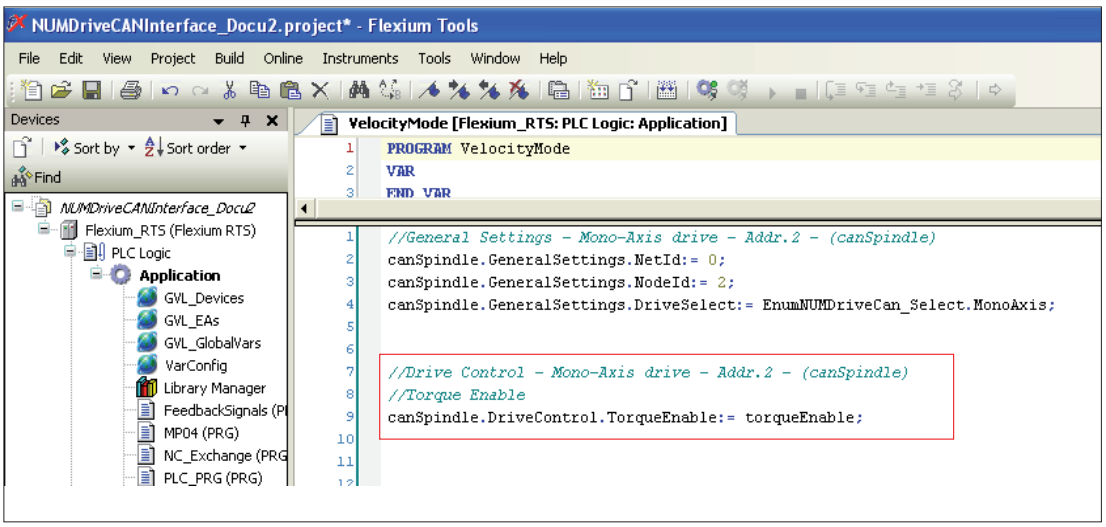
Important note
The only thing in programming to differentiate between a Mono-Axis and a Bi-Axes drive is this one and only enumeration parameter “myInstance.GeneralSettings.DriveSelect”!

This point is valid for programming Bi-Axes drives in “Velocity Mode” as well as in “PositioningMode”!

3.9.7 Drive Control

3.9.7.1 “TorqueEnable”

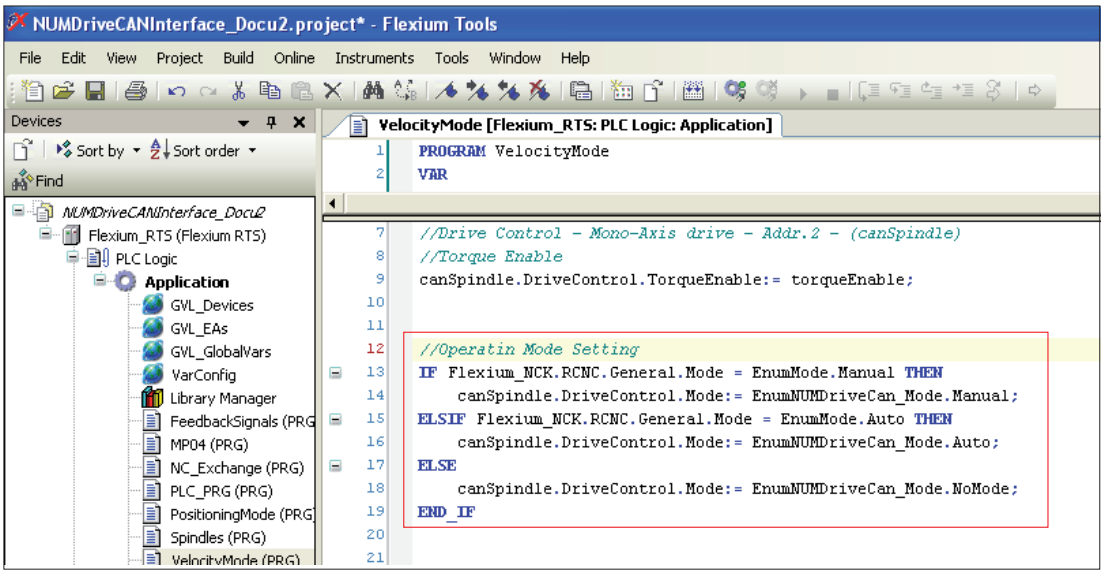
To set or reset the “Torque” on your drive use this Boolean parameter:



3.9.7.2 “Operation Mode Setting”

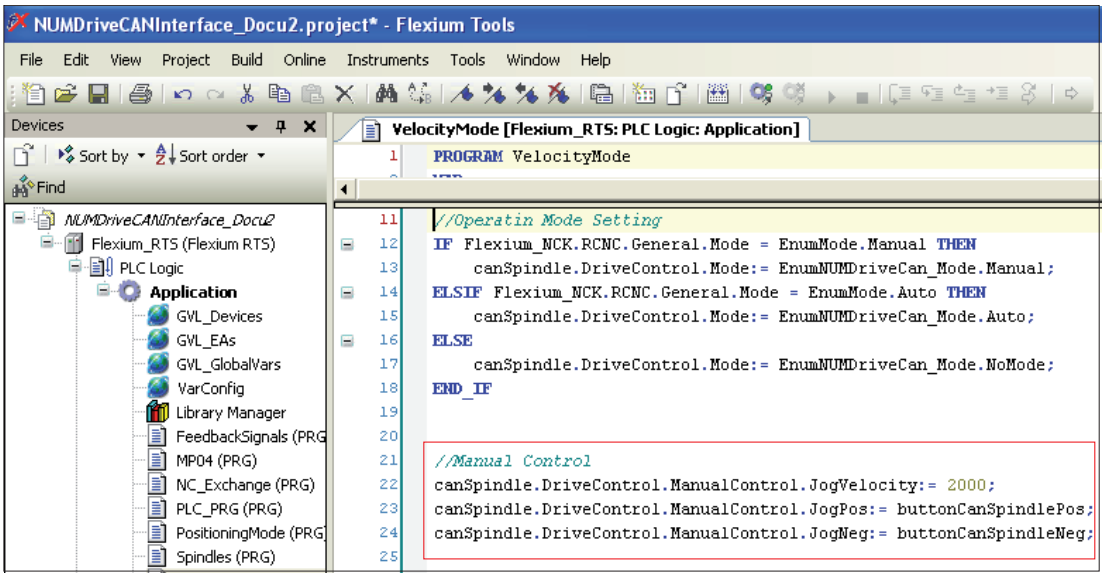
To change the operation mode on the drive you have to set the enumeration parameter “Mode” to the demanded enumeration value.

If you want to make the drive mode dependant from the “Flexium_NCK Mode”, which is changeable by HMI Buttons, you can use this little code snippet.



3.9.7.3 “ManualControl”

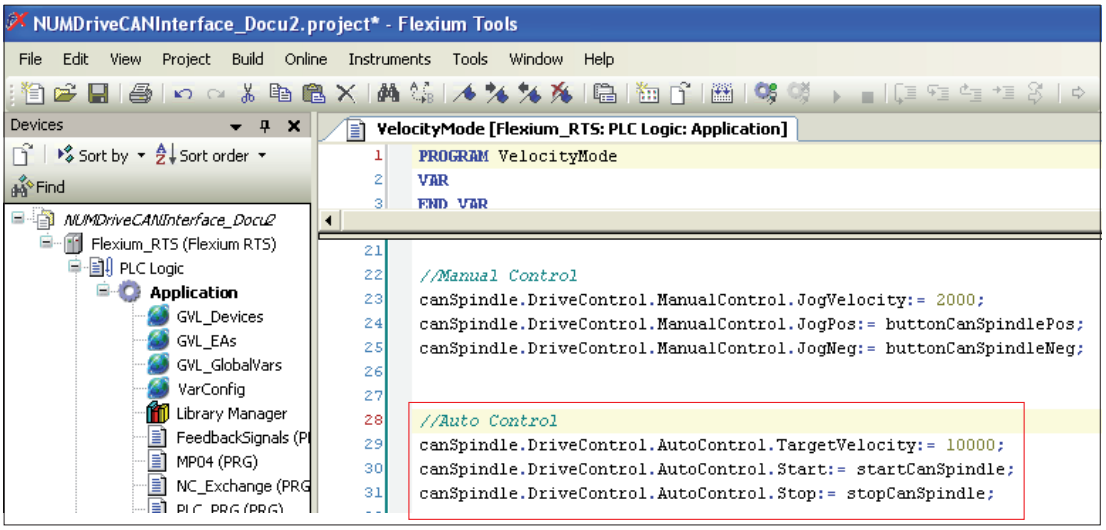
If you want to use the drive (also) in manual mode e.g. by pushing two different buttons for positive and negative direction motion, just use the parameters that are nested under “ManualControl”. Also you can set a separate velocity for the manual operation mode.



The “JogVelocity” is changeable during the motor is moving!

3.9.7.4 “AutoControl”

To change the “TargetVelocity” in “Automatic Mode” or to give a “Start” and a “Stop” use the following parameters.

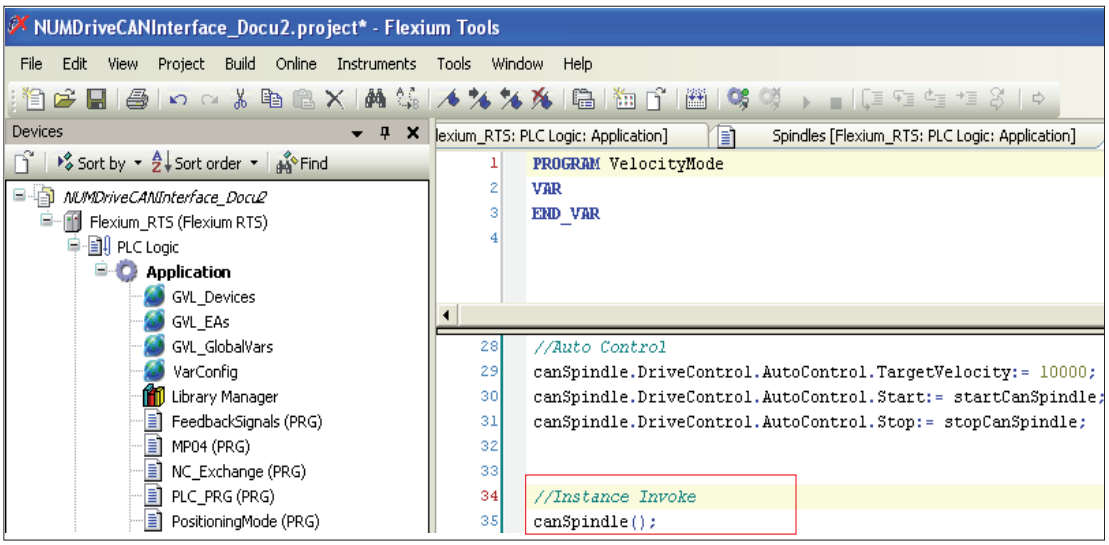


The “TargetVelocity” is changeable during the motor is moving!

.

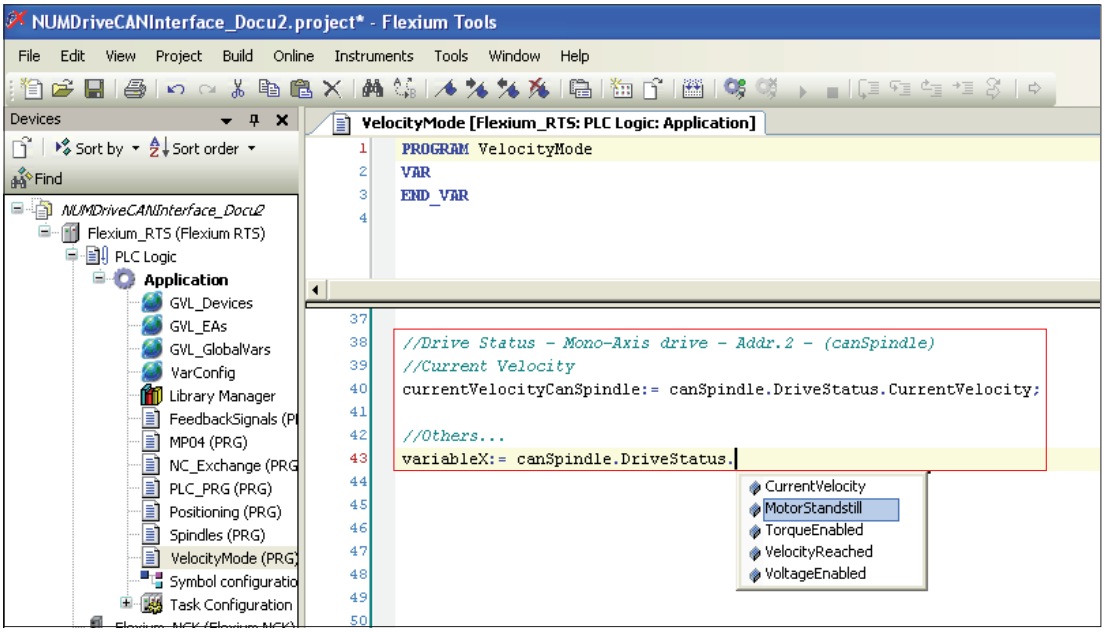
3.9.8 Instance invoke

What we have done until now is to give our instance “canSpindle” all the input parameter it needs to communicate with the drive and give several commands dependant on the variables we have programmed to the “DriveControl” functions.
To start the real interaction between the drive and the PLC instance, we must invoke our instance to make it work.



3.9.9 “DriveStatus”

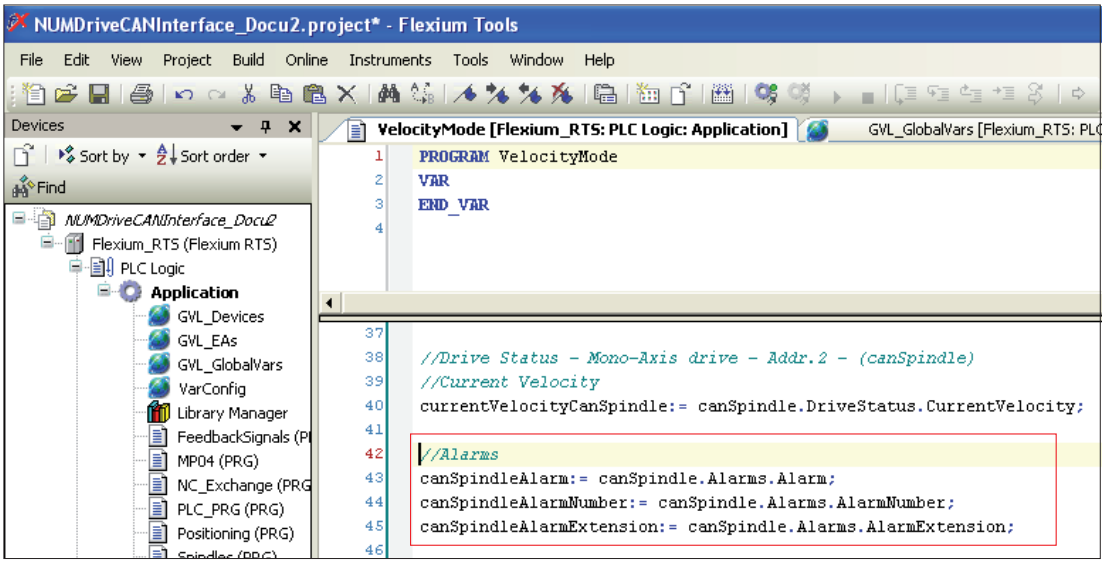
After we called up the instance, the communication between our PLC “canSpindle” instance and the “NUMDrive C CANopen” is already running and in fact, the drive is ready to move now.
But no control without status information and so we should take a look at the “DriveStatus”.
These are feedback information’s from the drive that can (and should) be used in the PLC.



The “CurrentVelocity” is a “REAL” data type parameter which mirrors the current velocity of the drive. All other self-explanatory parameters from “DriveStatus” are Boolean variables that can be used on your demands.

3.9.10 “Alarms”

The “NUMDriveCANInterface” library provides several alarm parameters to get the alarm state and alarm numbers from the target drive, in the case an alarm occurs on the drive.

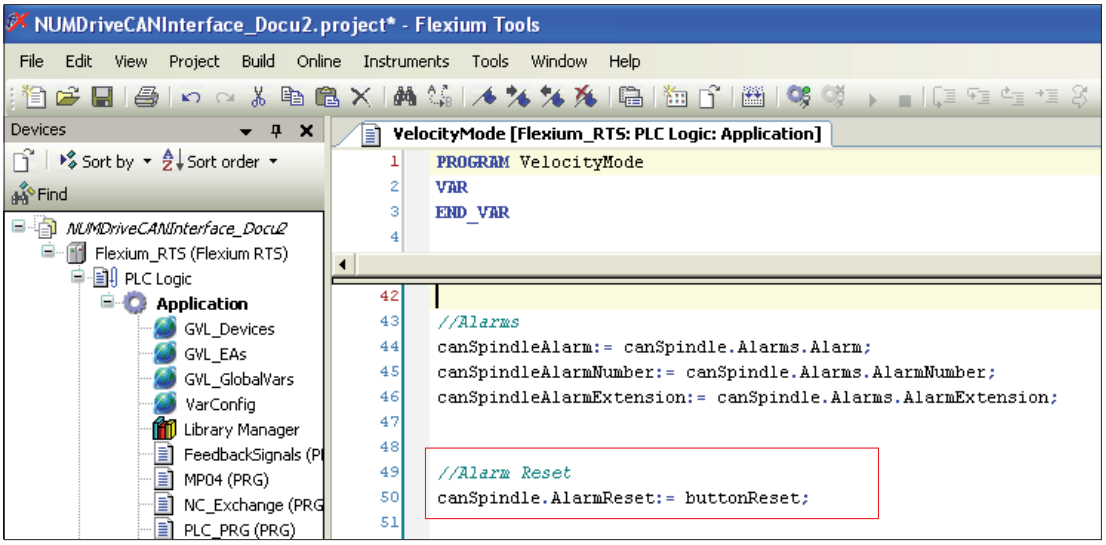


Important to know is that the parameter “Alarm” is a Boolean data type but “AlarmNumber” and “AlarmExtension” are Word data types. The coding of the “AlarmNumbers” and “AlarmExtensions” are equal to “NUMDrive C” with “DiscNT” interface.

There is one other parameter that can give useful information in the “Alarms” area.
The parameter is available under: “myInstance.Alarms.CANOPEN_Alarm_Read_Error” and the data type is “CANOPEN_KERNEL_ERROR”. This parameter gives information about a problem in CANopen communication during an internal “Alarm Read” request.

3.9.11 “AlarmReset”

When an “Alarm” occurs on a drive and the reason for this alarm is found and corrected, the Alarm must be reset. To send a reset to your drive, use following function:
- A reset is only transmitted to the drive, if there was an alarm detected before by the library.



3.9.12 “SDOAccess”

3.9.12.1 “ReadParameters”

The “NUMDriveCANInterface” libraries provide a function which you can use to read drive parameters via SDO in an easy way. You have to declare this function the parameter you want to read.
For this you should use the enumeration “EnumNUMDriveCAN_Param”.

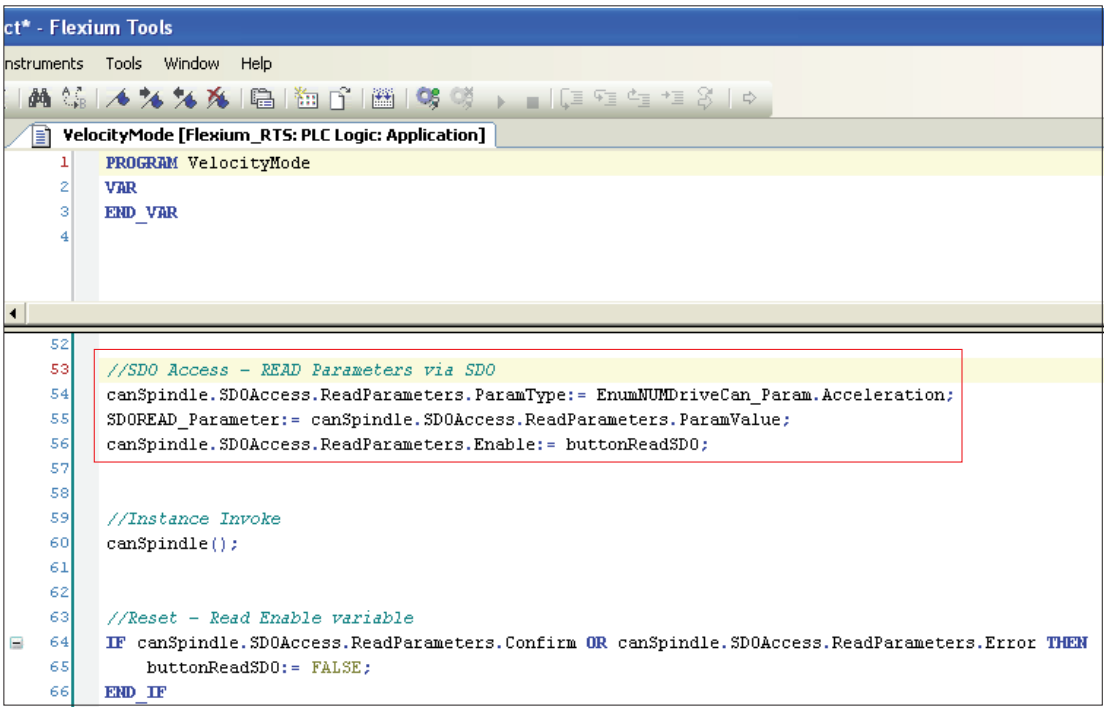
The “ParamValue” variable gives you back the value of the parameter you want to read.
Data type for this variable is “DINT”.

The “Enable” variable starts the reading transmission in the function.
The variable, that enables the reading process for the function (e.g. readSDORequest), must be set one time per read request (by any condition) and is set back when the function fires a confirm (or error) feedback signal, if you program the reset condition like shown below.

Consider that it is not allowed to “Enable” the function for more than one read request per read cycle. That means, you can only start a new read request (with the same or modified parameters) after the “Confirm” or “Error” feedback signal has reset the enable variable.

If you want to use this function in an optimal way you should place the code in the way shown in the picture below. The parametrize step should be done before the instance invoke and the “If condition” for reset the enable variable should be programmed after invoking the instance.
This sequence should be respected to decrease the CANopen busload.

Consider, that it’s allowed to invoke one and the same instance only one time in your whole application!



3.9.12.2 “SetParameters”

The “NUMDriveCANInterface” libraries provide also a function which you can use to Write (or Set) drive parameters via SDO in an easy way.

You have to tell this function the parameter you want to write.
For this you should use the enumeration “EnumNUMDriveCAN_Param”.

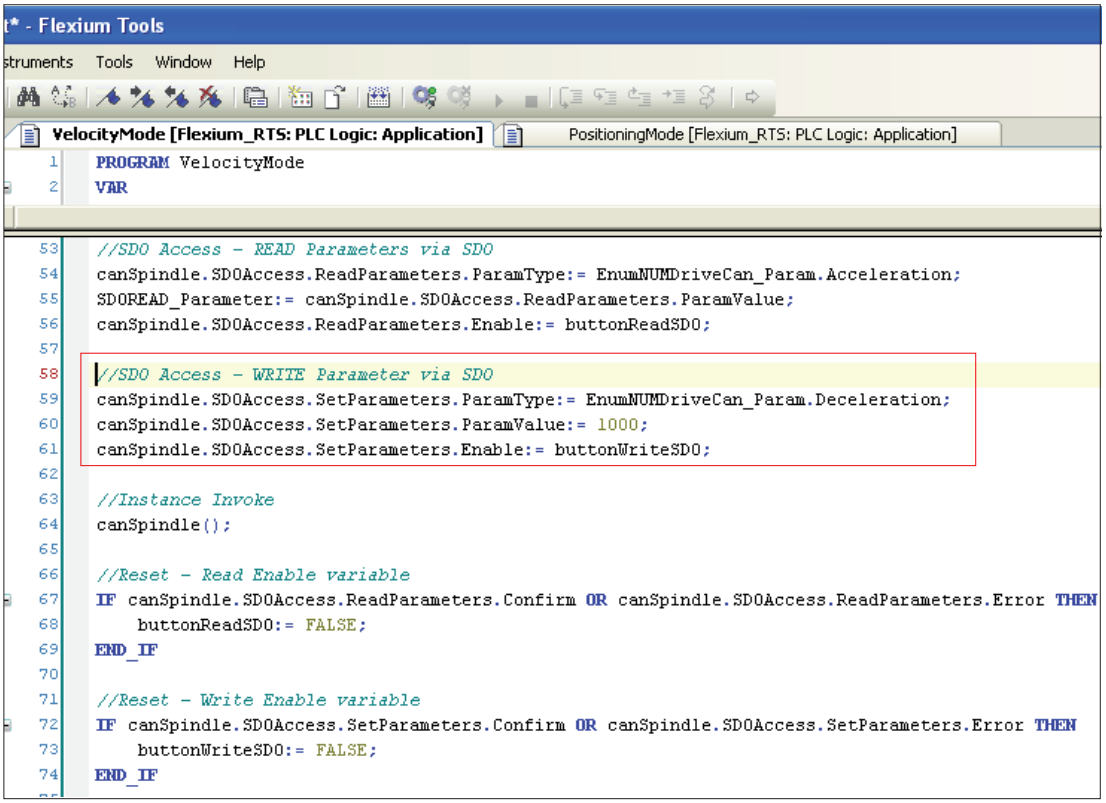
You must write the value you want to transmit to the “ParamValue” variable.
Data type for this variable is “DINT”.

The “Enable” variable starts the writing transmission in the function.
This variable, that enables the writing process for the function (e.g. writeSDORequest), must be set one time per write request (by any condition) and is set back when the function fires a confirm (or error) feedback signal, if you program the reset condition like shown below.

Consider that it is not allowed to “Enable” the function for more than one write request per write cycle. That means, you can only start a new write request (with the same or modified parameters) after the “Confirm” or “Error” feedback signal has reset the enable variable.

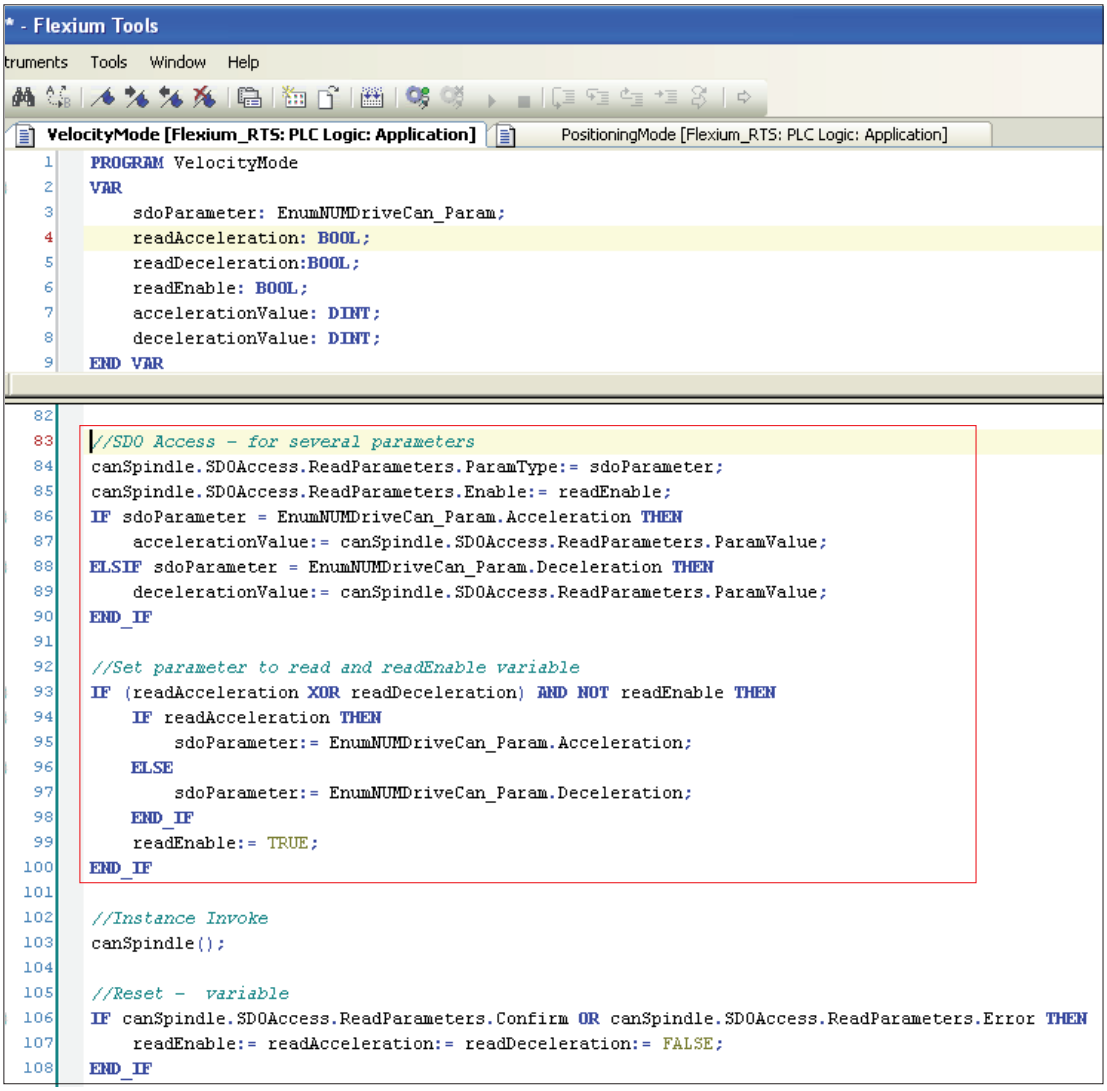
If you want to use this function in an optimal way you should place the code in the way shown in the picture below.
The parametrize step should be done before the instance invoke and the “If condition” for reset the enable variable should be programmed after invoking the instance.
This sequence should be respected to decrease the CANopen busload.

Consider, that it’s allowed to invoke one and the same instance only one time in your whole application!



3.9.12.3 Read / Set - Several Parameters

If you want to read or write several parameters in you application you have to consider that it is not allowed to “Enable” the function for more than one read/write request per read/write cycle.
That means, you can only start a new read request (with the same or modified parameters) after the “Confirm” or “Error” feedback signal has reset the enable variable.



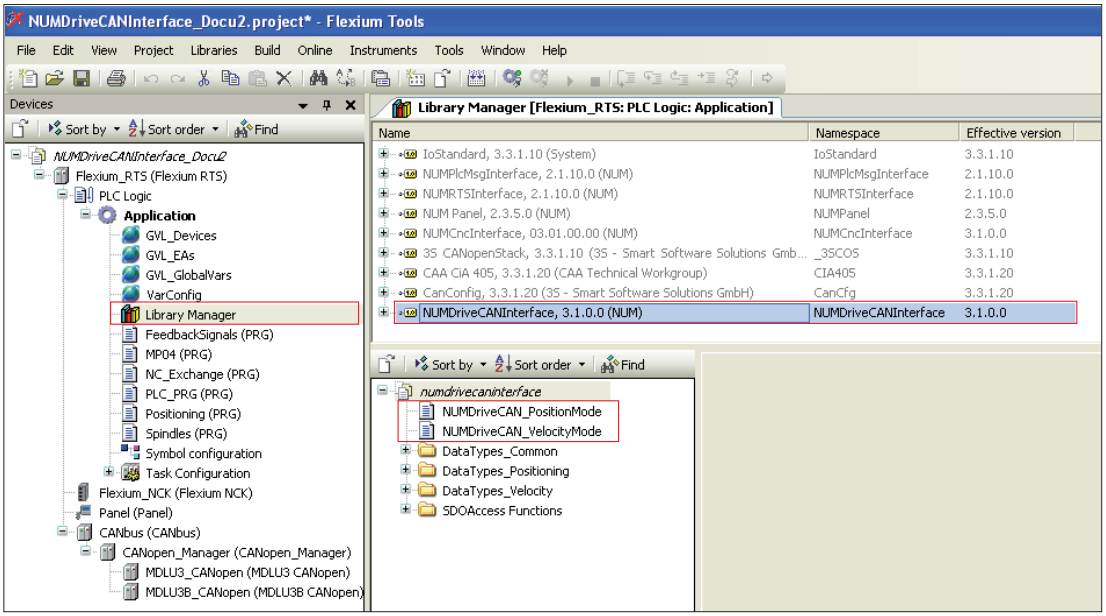
If your application needs many read and write transmissions to SDO objects there exists another possibility to realize a SDO transmission to a CANopen device.
Please refer to the Chapter 3.12 Extended SDO Access.

3.10 “Positioning Mode” in “NUMDriveCANInterface” library

By double clicking the “Library Manager” an overview window appears on the right side which displays all installed libraries. If a library gets selected a list with all integrated function blocks, data types, methods etc. as shown below.

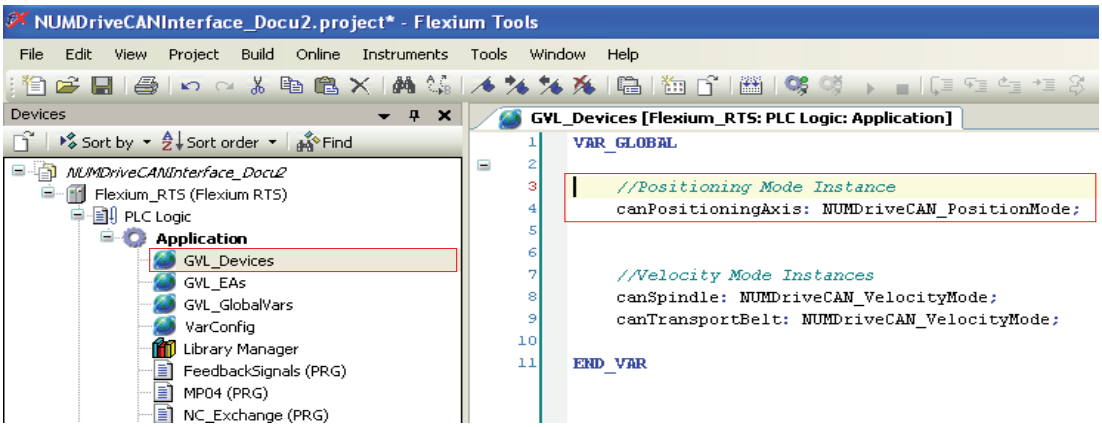
Selecting the “NUMDriveCANInterface” library shows two function blocks and all data types that are used internally. The data type listing is useful if the programmer wants to know what kind of data type must be declared for in or output parameter to communicate with the library.

The function block listings gives information about the available function blocks which can be instanced. In the “NUMDriveCANInterface” library there are two function blocks available.
With instancing one of this function blocks you declare which mode (Velocity or Positioning) you want to use.



To realize an interaction with the drive in your whole project, we recommend to create the instance in a GVL. In this state of your project (making an instance of the mode function block) you don’t need to differentiate between Mono- and Bi-Axes.
How to tell the instance which type of drive it shall drive will be explained later in this chapter.

In this example:
canPositioningAxis is the Bi-Axes Drive n (Positioning Mode)
(As declared in Chapter 3.6.1.2 (Bi-Axes)



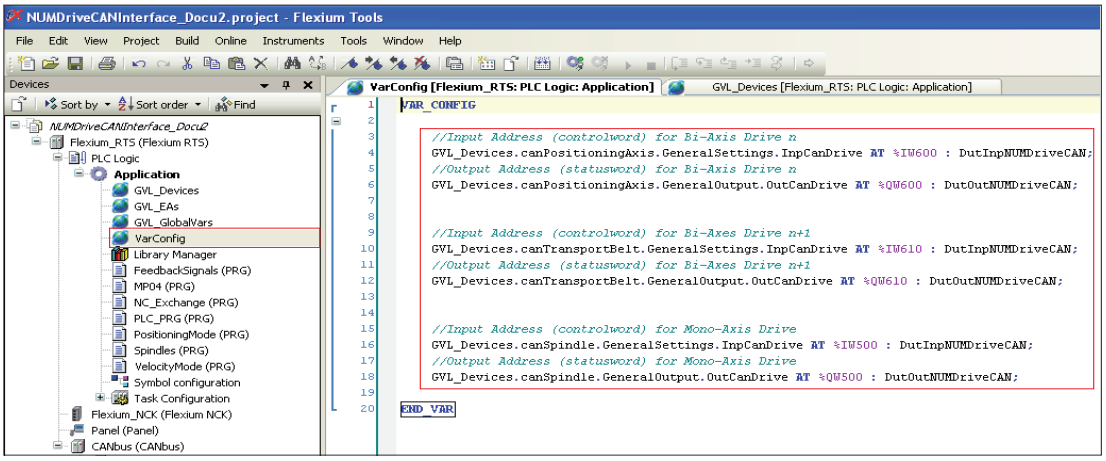
The instance “canPositioningAxis” is now available in your whole project and include all functions of the “NUMDriveCAN_PositioningMode” function block to control a “NUMDrive C CAN” in Positioning Mode.

3.10.1 “VAR Config” for “NUMDriveCANInterface” instances

To make it possible for the library instance to communicate with a drive, it is necessary to declare the status word and control word to the library instance first.
In Chapter 3.6.1.7 (Bi-Axes) we have already declared I/O addresses to the status and control words of the different drives.

The preferred way to map this declared addresses to the library instance is to use the “VAR Config” functionality. “VAR Config” maps an address area (beginning with the starting address number) directly into a library instance to the foreseen variable structure. To make a “VAR Config” you must create a GVL and change the GVL Keywords to “VAR_Config”.

Then you have to map the I/O addresses to the foreseen variables in the library instance like displayed in the picture below.

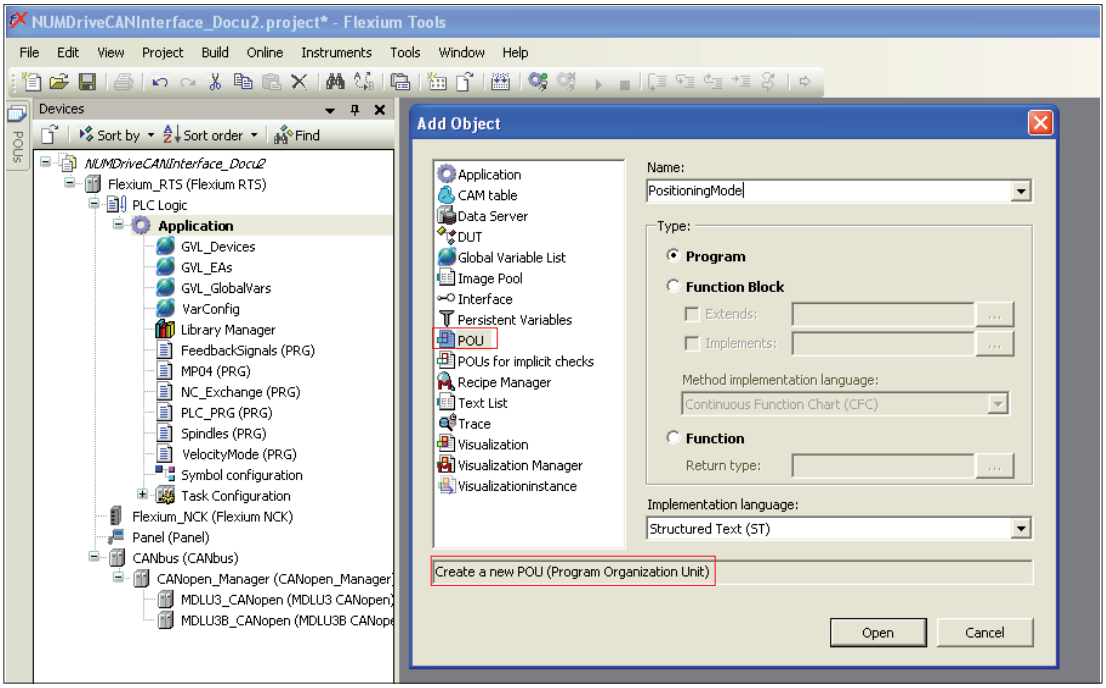


In this state of your project (making the “VAR_Config”) you don’t need to differentiate between Mono- and Bi-Axes. How to tell the instance which type of drive it shall drive will be explained later in this chapter.

3.10.2 Using an instance in a POU

The fact that we made a global instance of the “PositioningMode” function block makes it very comfortable to use these instance (canPositioningAxis) in all POU’s (Program Organization Unit) of our project.

The only thing we have to do is to add a new POU, define it as “Program” in “Structured Text” and give it a meaningful name

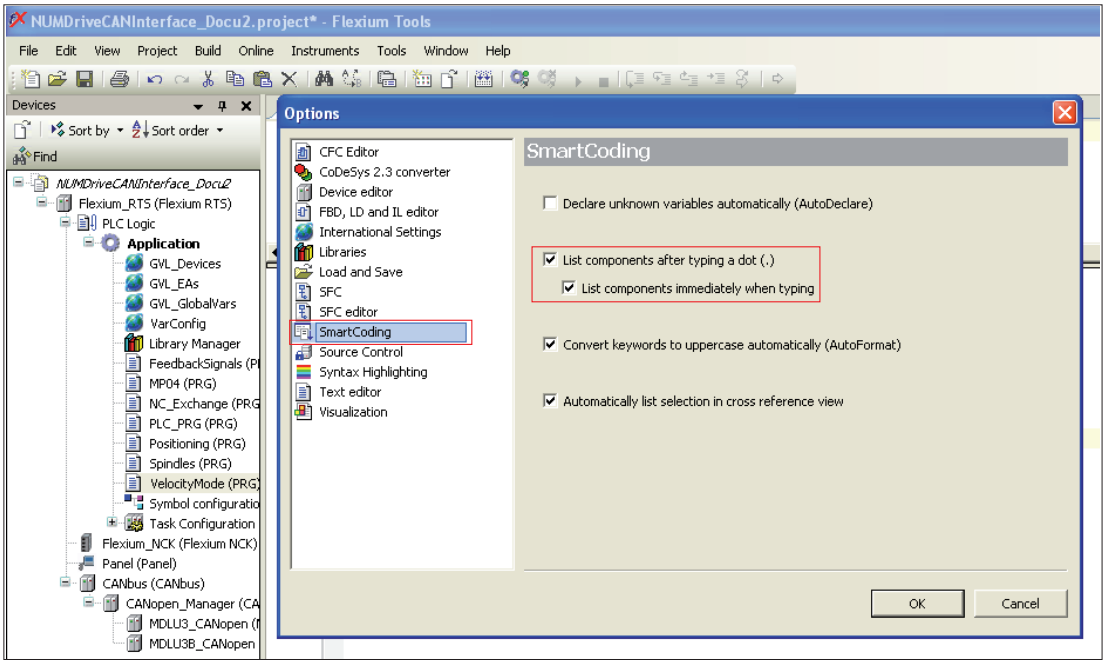


After this step we can start to program the interactions between our PLC and the drives by using the instance of the “PositioningMode” function block (canPositioningAxis).

3.10.3 IntelliSense

For the further work we recommend to enable the CoDeSys IntelliSense functionality to see the library structure after typing a dot (.)

To enable IntelliSense click on Tools → Options → Smart Coding.

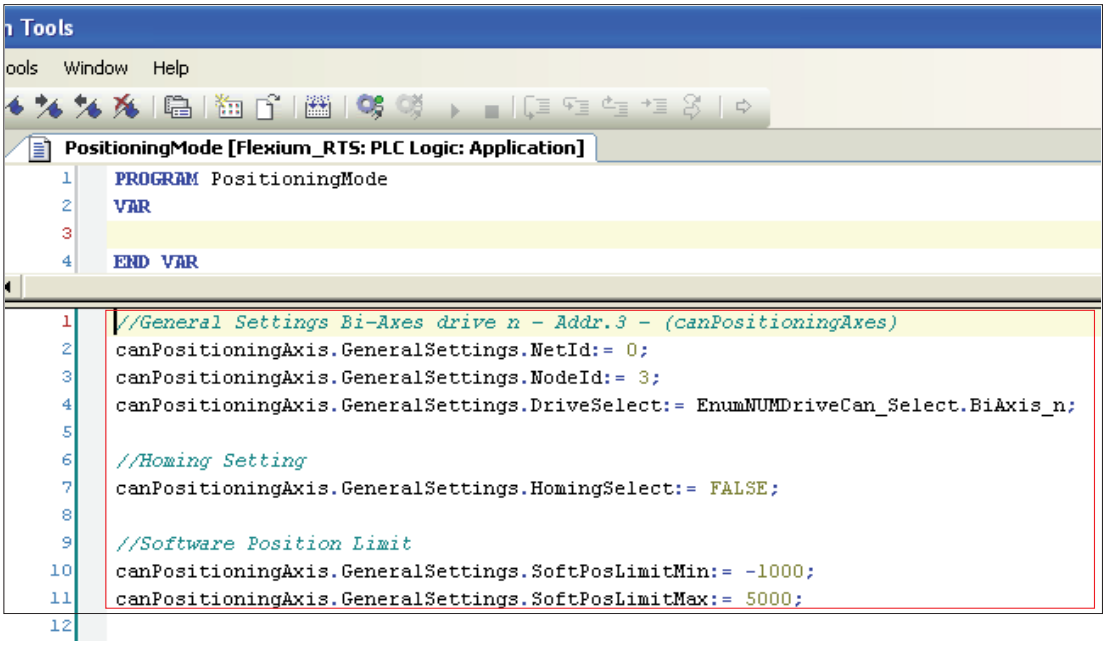


Also check other options to personalize your FlexiumTools surface and behaviour on your demands.

3.10.4 “General Settings”

Some more additional important information must be written to the instances by parameters.
These parameters are needed to make it clear for the instance, to which “Net” (interface number) it shall send their commands, the NodeID (hardware address) of the target drive and the drive type (Mono-Axis or Bi-Axes) as well as the internal drive part in case of a Bi-Axes (n or n+1).

For these important settings we use the parameters that are nested under “GeneralSettings”.



In this example the “NUMDrive C CANopen” which drives the “CAN Positioning Axes” is connected to Net “0” (like the setting we made in Chapter 3.3 Adding a CANBus to the Flexium_RTS)
It has the hardware address “3”, the “NUMDrive C CANopen” is a Bi-Axes drive and the internal Axis part is n.

3.10.4.1 Homing Enable

With the parameter “HomingSelect” you can declare if you have a sensor with absolute encoder (without homing) or you need a homing functionality because of your sensor or your demands.

3.10.4.2 Software Position Limits

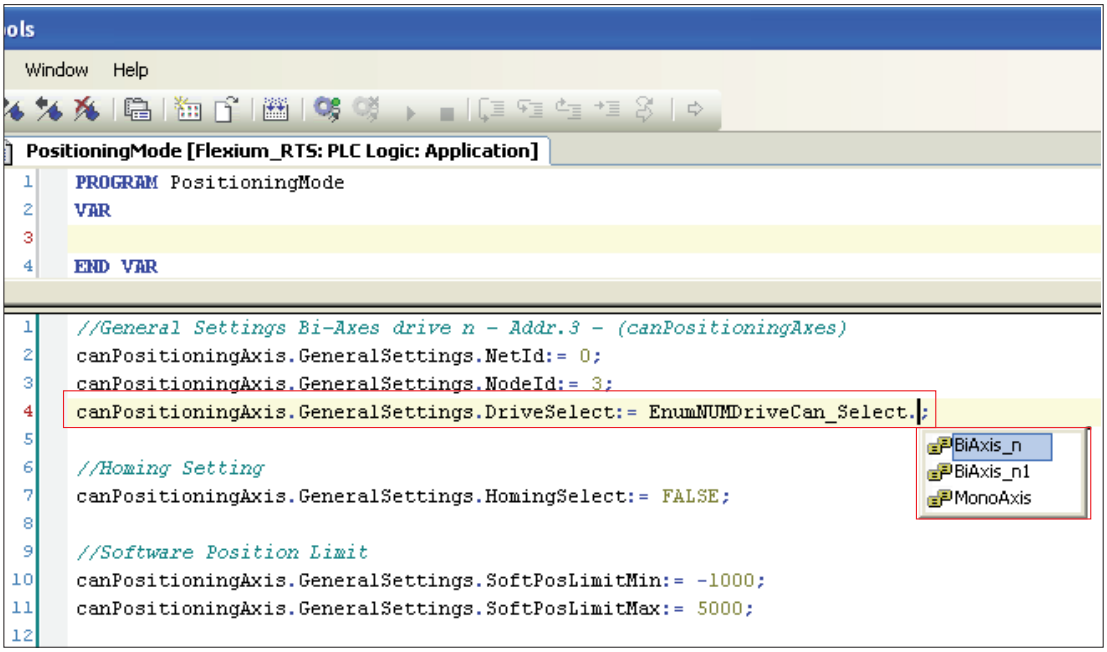
To declare the traverse path of your axis you can set up the software position limits minimum and maximum with the parameters “SoftPosLimitMin” and “SoftPosLimitMax”.

Important note
Consider that the software position limits are only transmitted during the boot up of the system, so it is not possible to change the software position limits during the control unit is running.

3.10.5 Mono-Axis and Bi-Axes distinction

As you can see in the picture below, the only difference between Mono-Axis and Bi-Axes drives, is to set the important parameter “DriveSelect” to the correct enumeration value with respect to the target drive type (Mono, Bi-n, Bi-n1) you want to communicate to.

All following settings, functions, and program parts are absolutely the same for Mono-Axis and Bi-Axes drives.

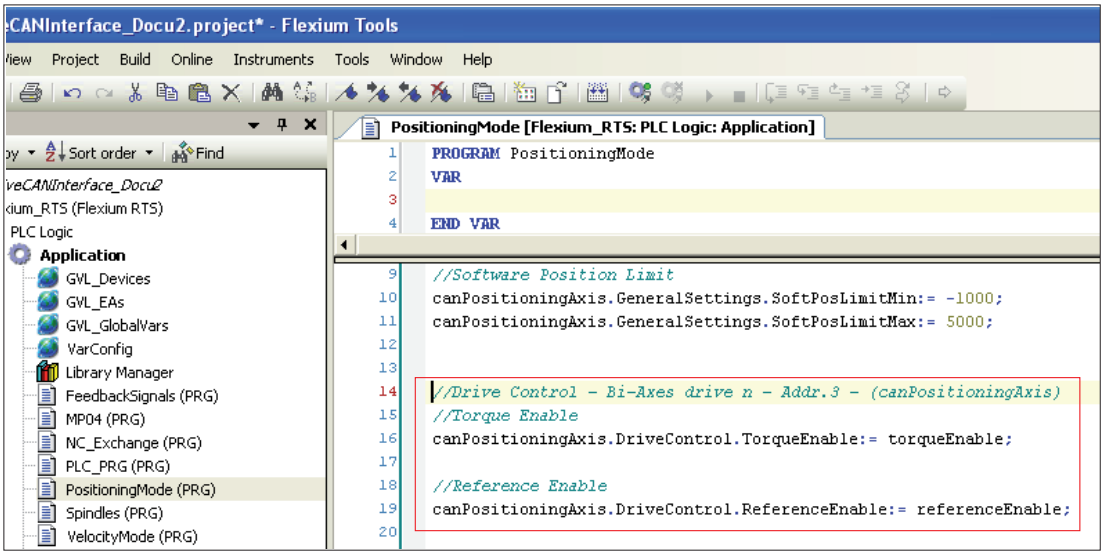


Important note
The only thing in programming to differentiate between a Mono-Axis and a Bi-Axes drive is this one and only enumeration parameter “myInstance.GeneralSettings.DriveSelect”! This point is valid for programming Bi-Axes drives in “Velocity Mode” as well as in “PositioningMode”!

3.10.6 “Drive Control”

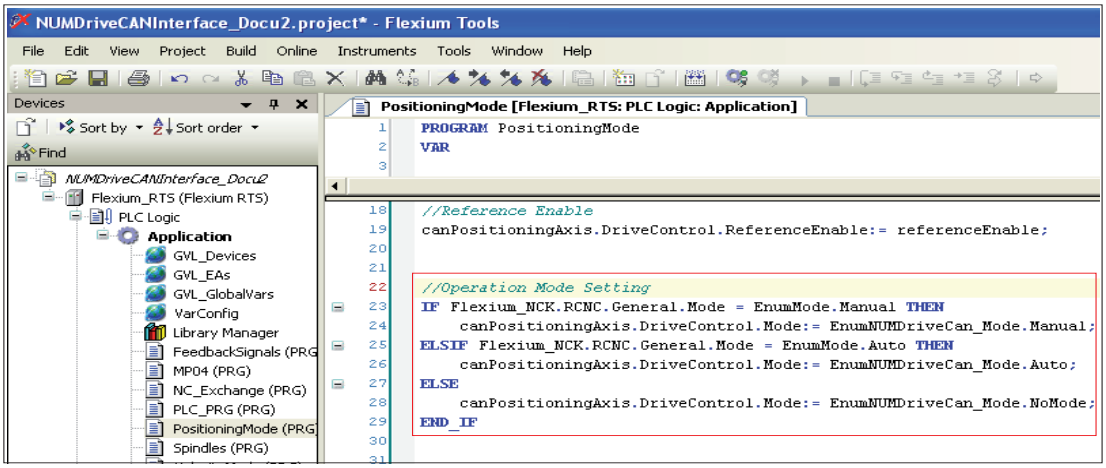
3.10.6.1 “TorqueEnable” and “ReferenceEnable”

To set or reset the “Torque” and “Reference” on your drive use these Boolean parameters:



3.10.6.2 “Operation Mode Setting”

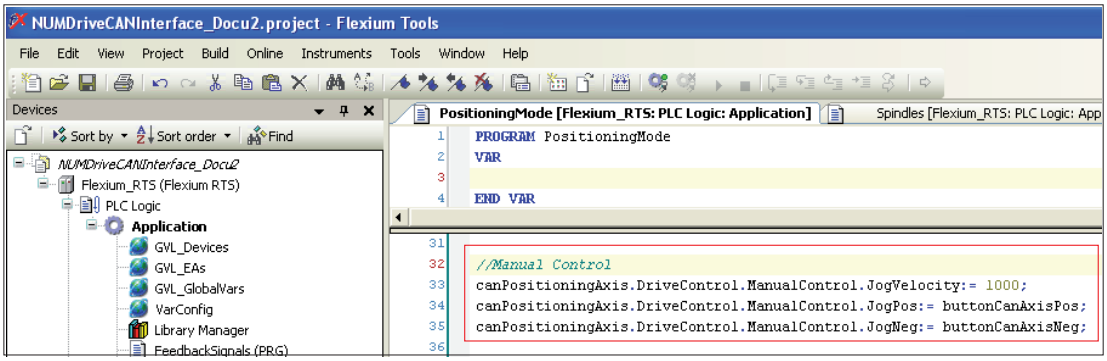
To change the operation mode on the drive you have to set the enumeration parameter “Mode” to the demanded enumeration value.
If you want to make the drive mode dependant from the “Flexium_NCK Mode”, which is changeable by HMI Buttons, you can use this little code snippet.



3.10.6.3 “ManualControl”

If you want to use the drive (also) in manual mode e.g. by pushing two different buttons for positive and negative direction motion, just use the parameters that are nested under “ManualControl”.

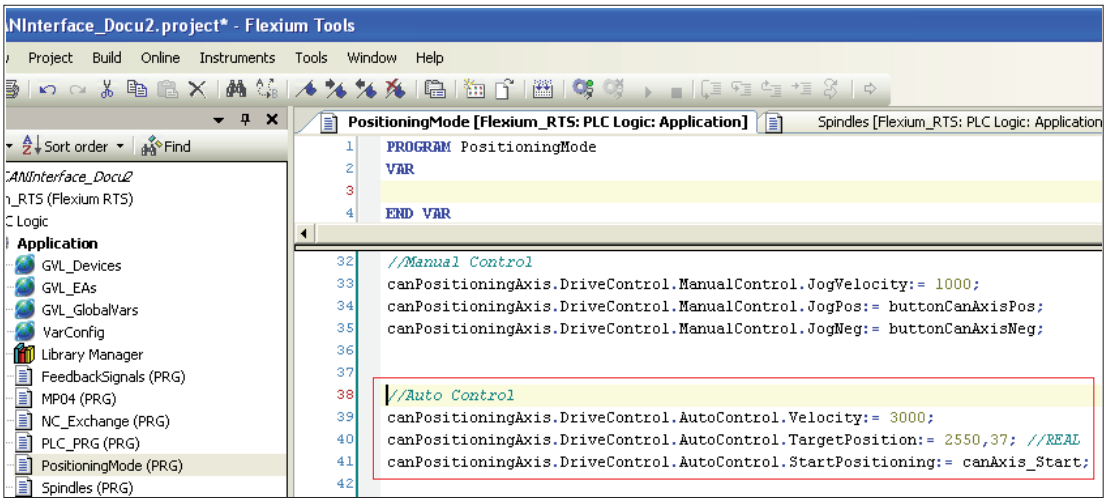
Also you can set a separate velocity for the manual operation mode.



The “JogVelocity” is changeable during the motor is moving!

3.10.6.4 “AutoControl”

To change the “TargetVelocity”, the “TargetPosition” and “StartPositioning” in “Automatic Mode” use the following parameters.

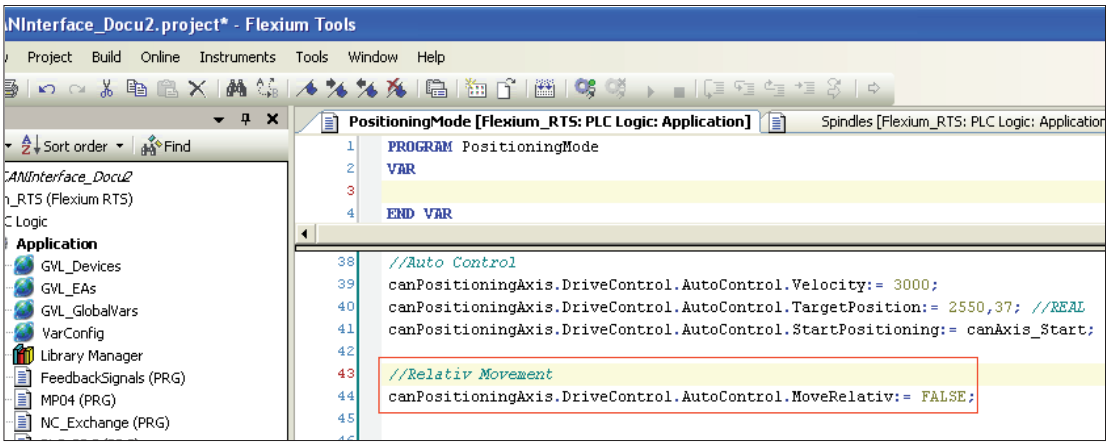


Consider that the “StartPositioning” needs only a “high” signal to start the movement after that the “canAxis_Start” variable can be reset and the movement continues. To stop the motion before reaching the target position you have to reset the “ReferenceEnable” parameter.

The motion starts when the motor is at “Standstill” and the last movement was finished successfully. To start a new movement during the motor is running (with a new target position or another velocity) or after an aborted positioning, take a look at the “Change Positioning Immediately” function.

3.10.6.5 Relative Movement

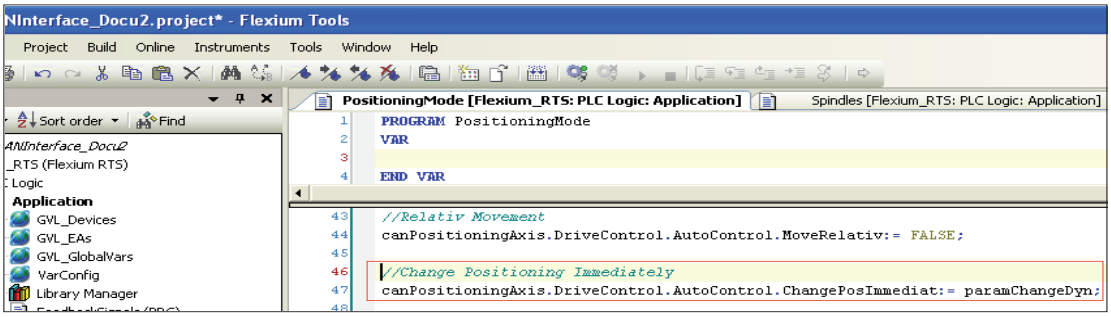
If you don't want to make absolute movement to the target position you have programmed but to add the value of the target position to your current position, just enable the "MoveRelativ" function or disable it for absolute target positions.



3.10.6.6 Change Positioning Immediately

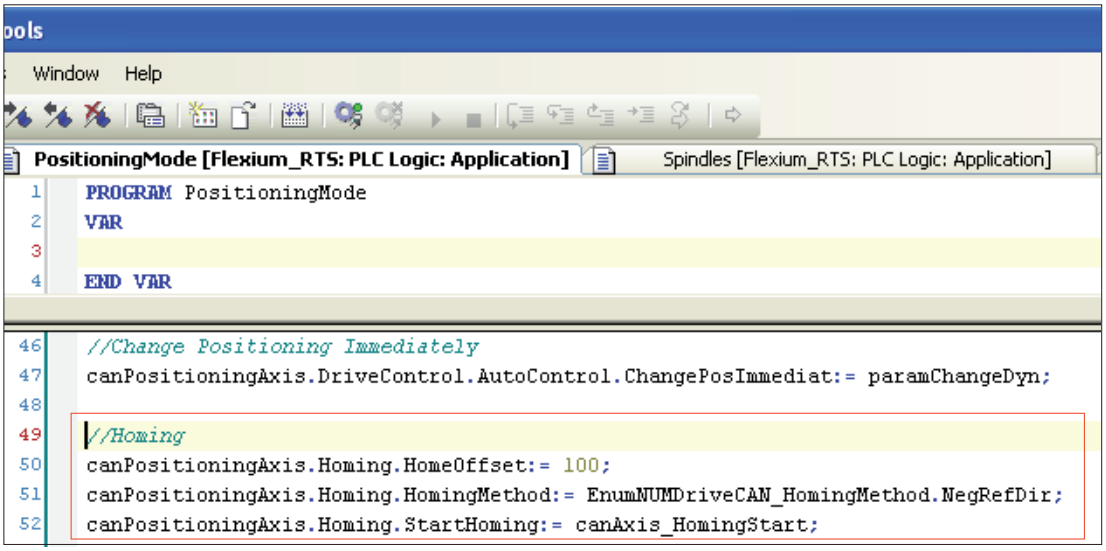
Normally, after a positioning was completed successfully, you can start a new positioning with modified "TargetPosition" and "Velocity" parameters. The motion starts when the motor is at "Standstill" and the last movement was finished successfully.

To start a new movement during the motor is running (with a new target position and/or another velocity) or after an aborted positioning, set the "ChangePosImmediat" parameter and the new motion starts with respect to the new "TargetPosition" and "Velocity" immediately. In case of changing the motion direction by using this function, of course the deceleration and acceleration values are respected.



3.10.7 “Homing”

If you have enabled the “HomingSelect” parameter (“myInstance.GeneralSettings.HomingSelect:= true”), you have to make two further parameter settings and start the homing movement like in this code snippet.



“HomeOffset” is the value that is set as current position after the reference point was reached. “StartHoming” executes the movement.

“HomingMethod” defines the way of the homing procedure that you want to use - (“EqualsExtendedSDOSettings”) means that the homing method is set not with this enumeration but by transmitting a specific homing method number to the according SDO object by using SDO transmission.

The **Homing method 6098h** object determines the method that will be used during homing.
By choosing a method of homing by writing a value to homing method will clearly establish:

- the homing signal (positive limit switch, negative limit switch, home switch)
- the direction of actuation
- the position of the index pulse

Please refer to the table on the next page for detailed homing methods.

Additional to this there are two more parameter for homing mode which are reachable by SDO communication:

- **Homing speeds 6099h**
- **Homing Acceleration 609Ah.**

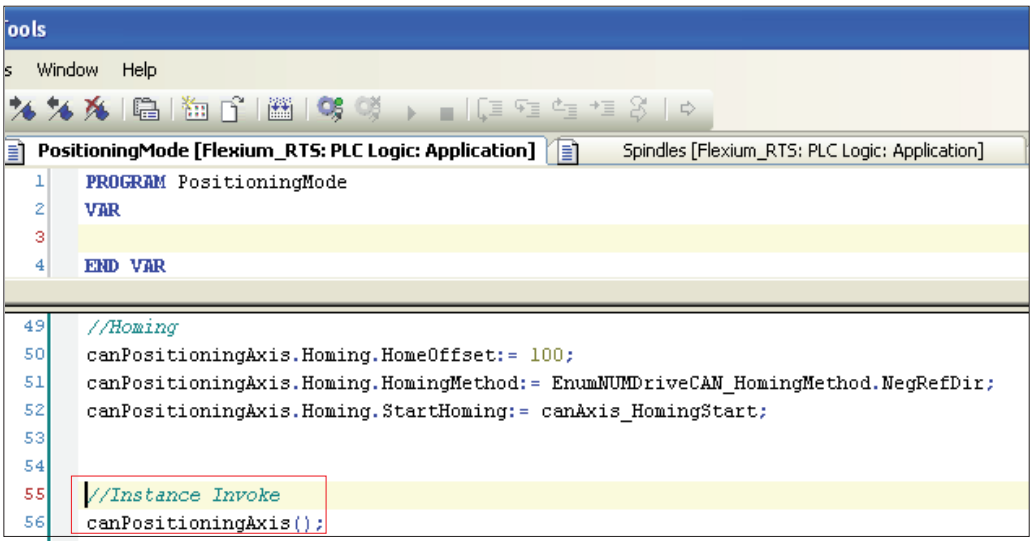
Homing methods

Value	Description
-128 ... -1	Manufacturer specific
0	No homing operation request
1	LIMN with index pulse
2	LIMP with index pulse
7	REF+ with index pulse, inverted, outside
8	REF+ with index pulse, inverted, inside
9	REF+ with index pulse, not inverted, inside
10	REF+ with index pulse, not inverted, outside
11	REF- with index pulse, inverted, outside
12	REF- with index pulse, inverted, inside
13	REF- with index pulse, not inverted, inside
14	REF- with index pulse, not inverted, outside
17	LIMN
18	LIMP
23	REF+, inverted, outside
24	REF+, inverted, inside
25	REF+, not inverted, inside
26	REF+, not inverted, outside
27	REF-, inverted, outside
28	REF-, inverted, inside
29	REF-, not inverted, inside
30	REF-, not inverted, outside
33	Index pulse negative direction
34	Index pulse positive direction
35	Homing on the current position
36 ... 127	Reserved

Legenda.
LIMN: homing on the negative limit switch
LIMP: homing on the positive limit switch
REF+: search home switch with movement in positive direction of rotation
REF-: search home switch with movement in negative direction of rotation
Inv.: invert direction in switch
Not Inv.: not invert direction in switch
Outside: index pulse/distance outside switch
Inside: index pulse/distance inside switch.

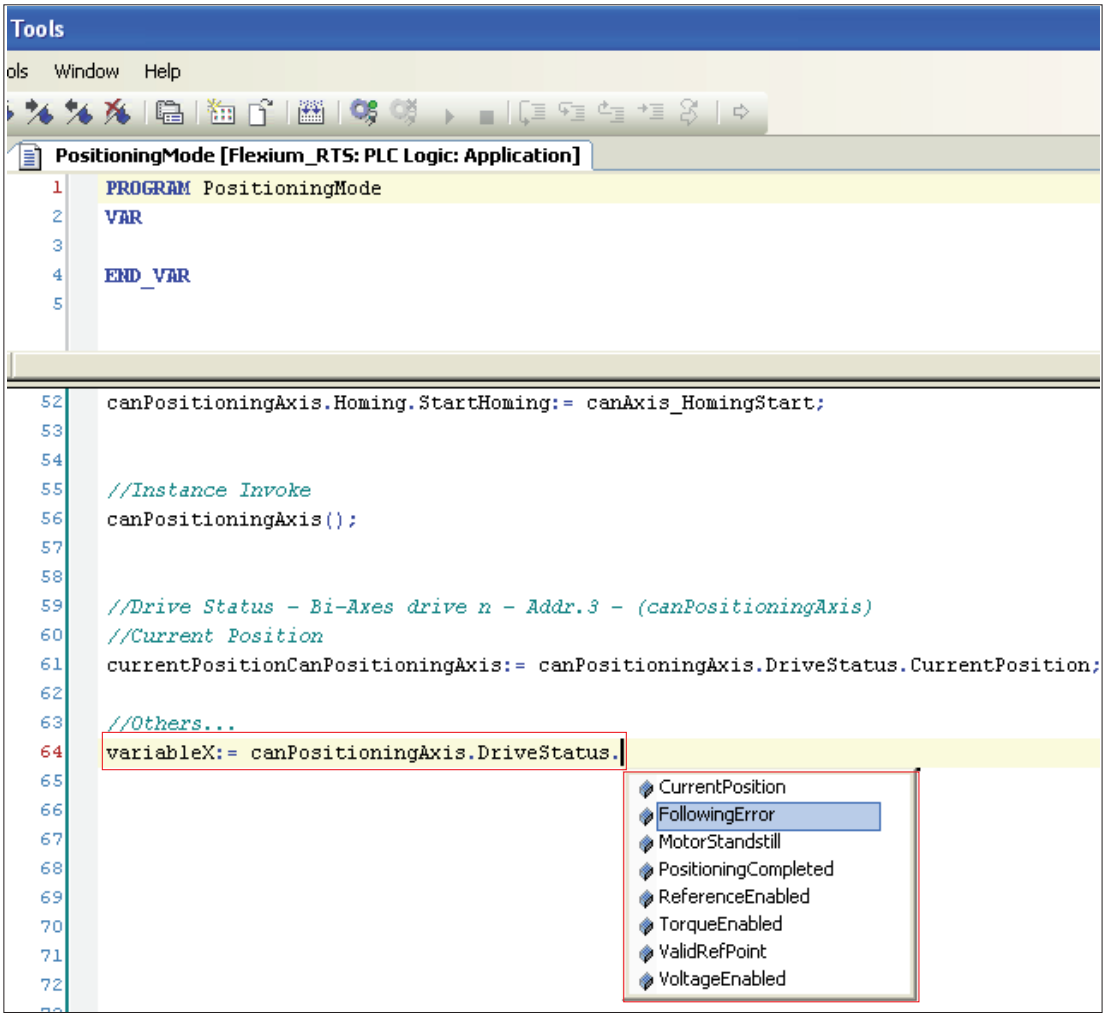
3.10.8 Instance invoke

What we have done until now, is to give our instance “canPositioningAxis” all the input parameter it needs to communicate with the drive and give several commands dependant on the variables we have programmed to the “DriveControl” functions.
To start the real interaction between the drive and the PLC instance, we must invoke our instance to make it work.



3.10.9 “DriveStatus”

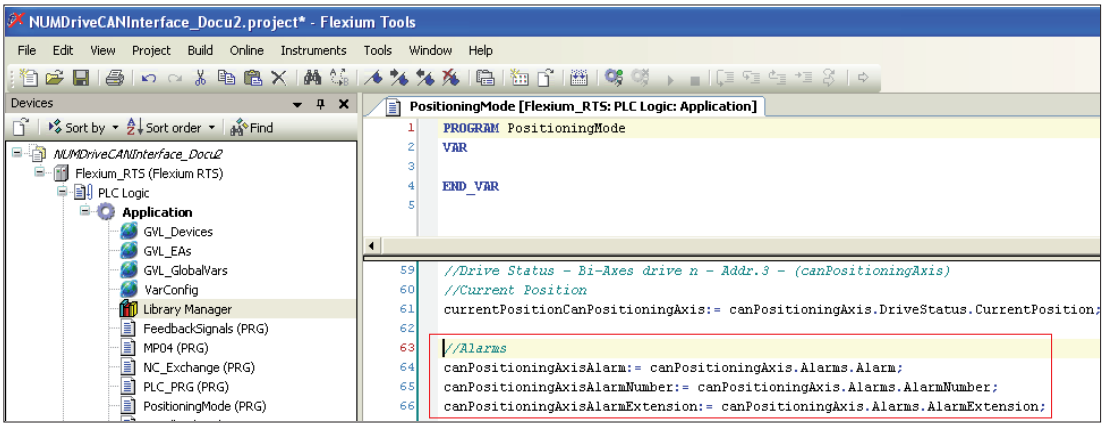
After we invoked the instance, the communication between our PLC “canPositioningAxis” instance and the “NUMDrive C CANopen” is already running and in fact, the drive is ready to move now. But no control without status information and so we should take a look at the “DriveStatus”. This are feedback information’s from the drive that can (and should) be used in the PLC.



The “CurrentPosition” is a “REAL” data type parameter which mirrors the current position of the drive. All other self-explanatory parameters from “DriveStatus” are Boolean variables that can be used on your demands.

3.10.10 “Alarms”

The “NUMDriveCANInterface” library provides several alarm parameters to get the alarm state and alarm numbers from the target drive, in the case an alarm occurs on the drive.



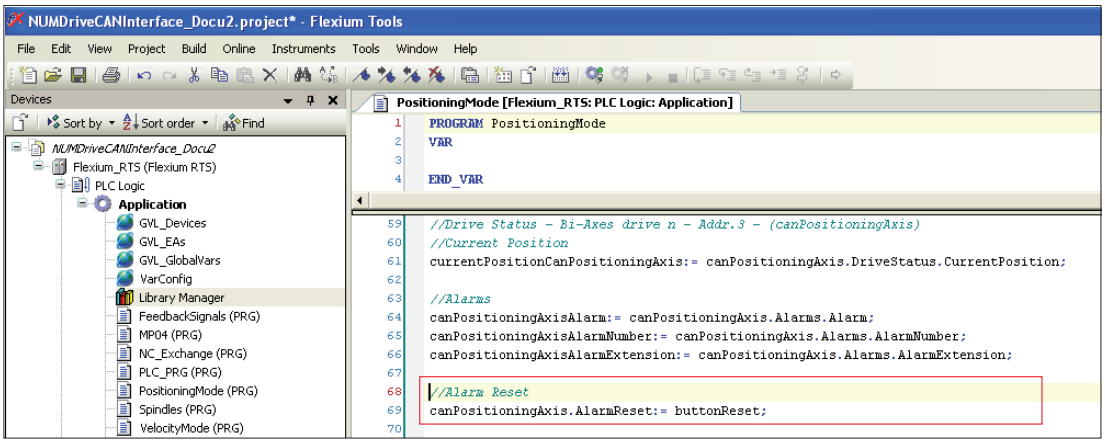
Important to know is that the parameter “Alarm” is a Boolean data type but “AlarmNumber” and “AlarmExtension” are Word data types. The coding of the “AlarmNumbers” and “AlarmExtensions” are equal to “NUMDrive C” with “DiscNT” interface.

There is one other parameter that can give useful information in the “Alarms” area.
The parameter is available under: “myInstance.Alarms.CANOPEN_Alarm_Read_Error” and the data type is “CANOPEN_KERNEL_ERROR”.

This parameter gives information about a problem in CANopen communication during an internal “Alarm Read” request.

3.10.11 “AlarmReset”

When an “Alarm” occurs on a drive and the reason for this alarm is found and corrected, the Alarm must be reset. To send a reset to your drive, use following function:
A reset is only transmitted to the drive, if there was an alarm detected before by the library.



3.10.12 “SDOAccess”

3.10.12.1 “ReadParameters”

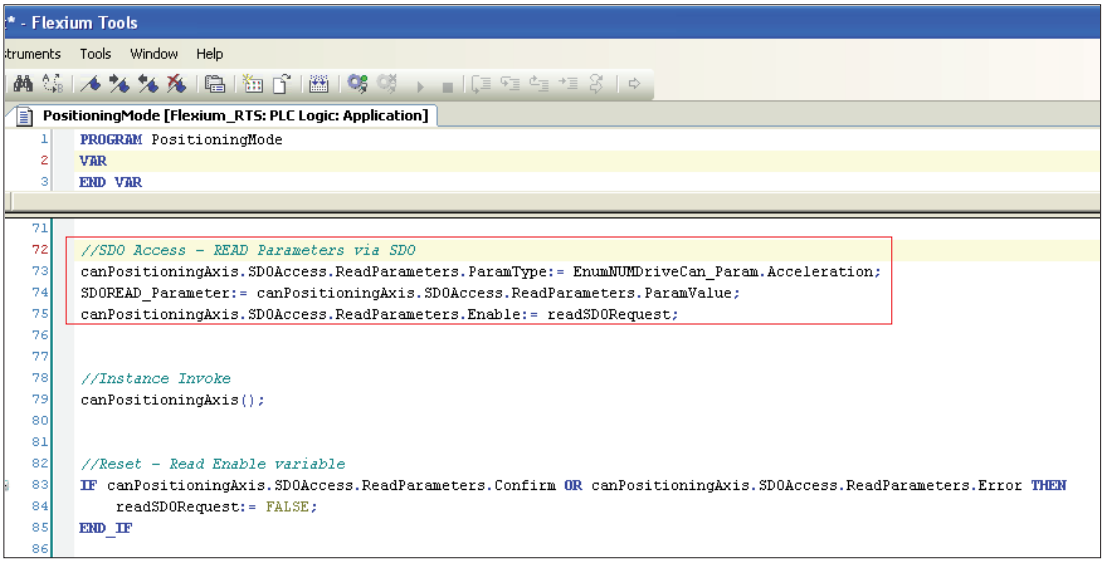
The “NUMDriveCANInterface” libraries provide a function which you can use to read drive parameters via SDO in an easy way. You have to tell this function the parameter you want to read.
For this you should use the enumeration “EnumNUMDriveCAN_Param”.

The “ParamValue” variable gives you back the value of the parameter you want to read.
Data type for this variable is “DINT”.

The “Enable” variable starts the reading transmission in the function.
The variable, that enables the reading process for the function (e.g. readSDORequest), must be set one time per read request (by any condition) and is set back when the function fires a confirm (or error) feedback signal, if you program the reset condition like shown below.
Consider that it is not allowed to “Enable” the function for more than one read request per read cycle. That means, you can only start a new read request (with the same or modified parameters) after the “Confirm” or “Error” feedback signal has reset the enable variable.

If you want to use this function in an optimal way you should place the code in the way shown in the picture below. The parametrize step should be done before the instance invoke and the “If condition” for reset the enable variable should be programmed after invoking the instance.

This sequence should be respected to decrease the CANopen busload.



Consider, that it’s allowed to invoke one and the same instance only one time in your whole application!

3.10.12.2 “SetParameters”

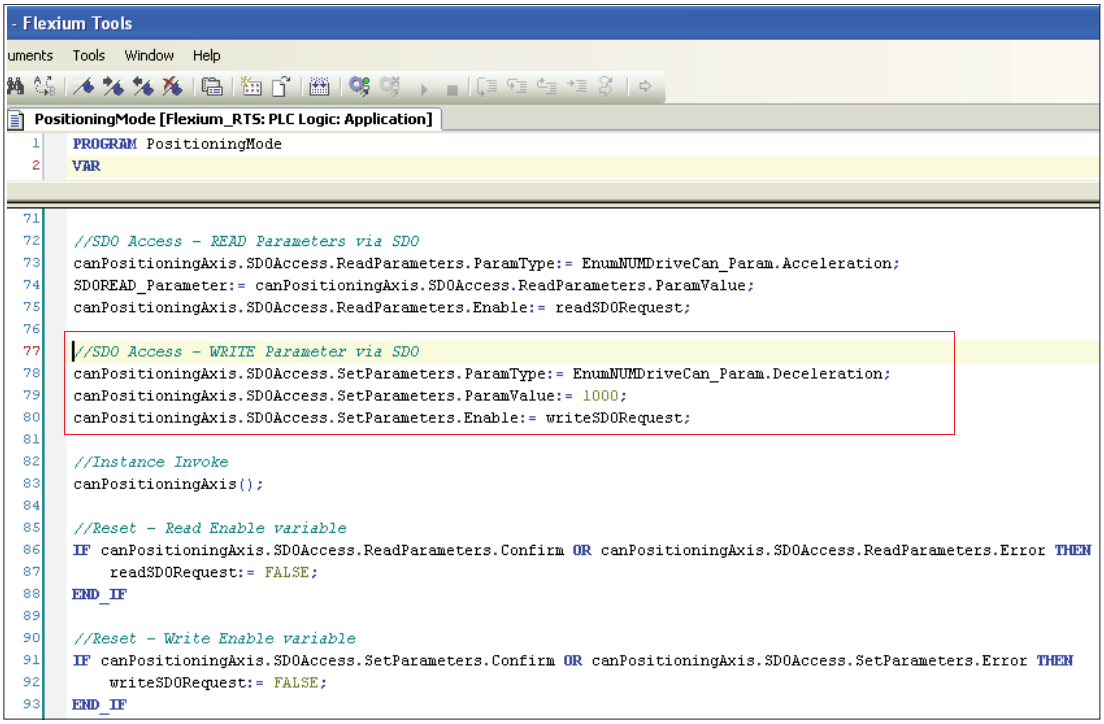
The “NUMDriveCANInterface” libraries provide also a function which you can use to Write (or Set) drive parameters via SDO in an easy way. You have to tell this function the parameter you want to write. For this you should use the enumeration “EnumNUMDriveCAN_Param”.

You must write the value you want to transmit to the “ParamValue” variable.
Data type for this variable is “DINT”.

The “Enable” variable starts the writing transmission in the function.
This variable, that enables the writing process for the function (e.g. writeSDORequest), must be set one time per write request (by any condition) and is set back when the function fires a confirm (or error) feedback signal, if you program the reset condition like shown below.
Consider that it is not allowed to “Enable” the function for more than one write request per write cycle. That means, you can only start a new write request (with the same or modified parameters) after the “Confirm” or “Error” feedback signal has reset the enable variable.

If you want to use this function in an optimal way you should place the code in the way shown in the picture below. The parametrize step should be done before the instance invoke and the “If condition” for reset the enable variable should be programmed after invoking the instance.

This sequence should be respected to decrease the CANopen busload.

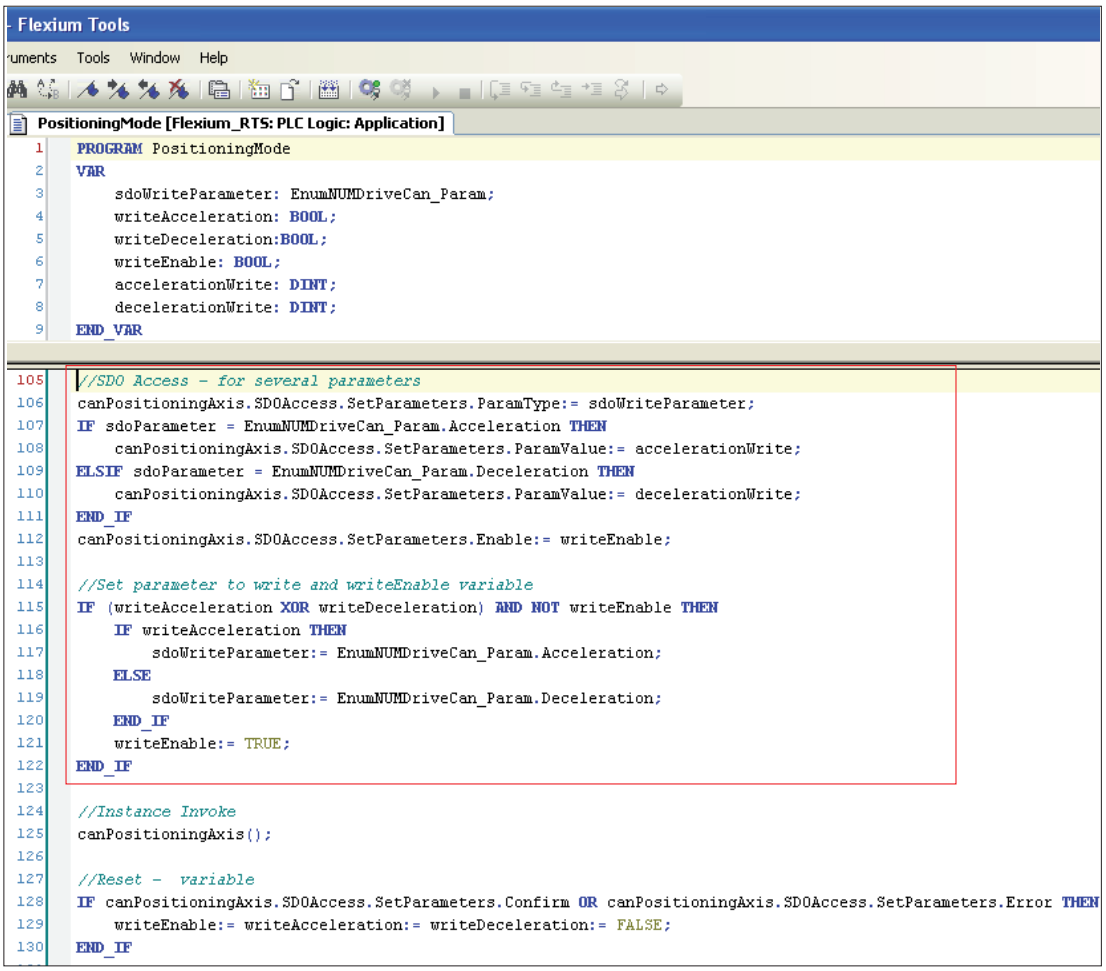


Consider, that it’s allowed to invoke one and the same instance only one time in your whole application!

3.10.12.3 Read / Set - Several Parameters

If you want to read or write several parameters in you application you have to consider that it is not allowed to “Enable” the function for more than one read/write request per read/write cycle.

That means, you can only start a new read request (with the same or modified parameters) after the “Confirm” or “Error” feedback signal has reset the enable variable.



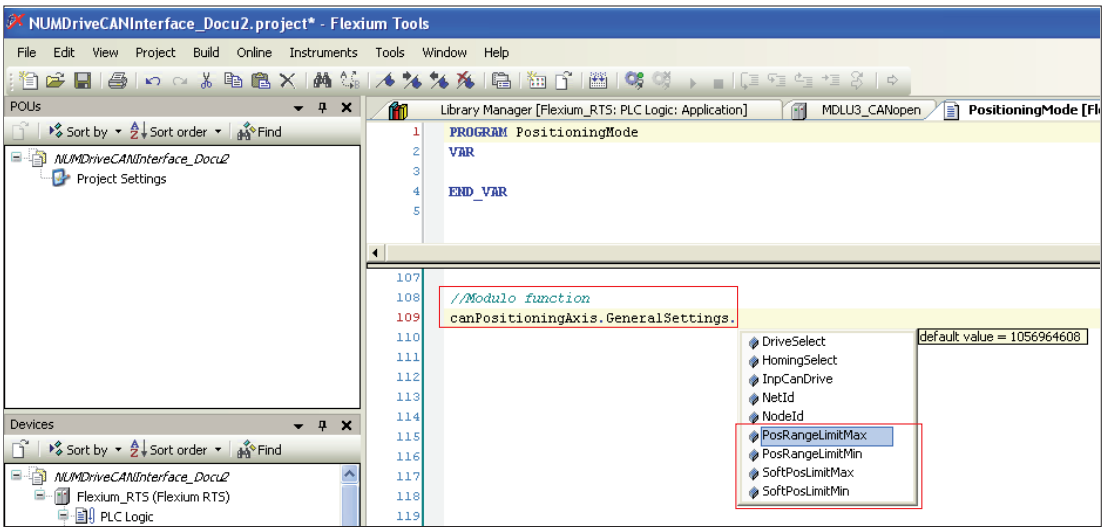

```
1   PROGRAM PositioningMode
2   VAR
3       sdoWriteParameter: EnumNUMDriveCan_Param;
4       writeAcceleration: BOOL;
5       writeDeceleration:BOOL;
6       writeEnable: BOOL;
7       accelerationWrite: DINT;
8       deceleration Write: DINT;
9   END_VAR

105 //SDO Access - for several parameters
106 canPositioningAxis.SDOAccess.SetParameters.ParamType:= sdoWriteParameter;
107 IF sdoParameter = EnumNUMDriveCan_Param.Acceleration THEN
108     canPositioningAxis.SDOAccess.SetParameters.ParamValue:= accelerationWrite;
109 ELSIF sdoParameter = EnumNUMDriveCan_Param.Deceleration THEN
110     canPositioningAxis.SDOAccess.SetParameters.ParamValue:= decelerationWrite;
111 END_IF
112 canPositioningAxis.SDOAccess.SetParameters.Enable:= writeEnable;
113
114 //Set parameter to write and writeEnable variable
115 IF (writeAcceleration XOR writeDeceleration) AND NOT writeEnable THEN
116     IF writeAcceleration THEN
117         sdoWriteParameter:= EnumNUMDriveCan_Param.Acceleration;
118     ELSE
119         sdoWriteParameter:= EnumNUMDriveCan_Param.Deceleration;
120     END_IF
121     writeEnable:= TRUE;
122 END_IF
123
124 //Instance Invoke
125 canPositioningAxis();
126
127 //Reset -  variable
128 IF canPositioningAxis.SDOAccess.SetParameters.Confirm OR canPositioningAxis.SDOAccess.SetParameters.Error THEN
129     writeEnable:= writeAcceleration:= writeDeceleration:= FALSE;
130 END_IF
```

If your application needs many read and write transmissions to SDO objects there exists another possibility to realize a SDO transmission to a CANopen device. (Chapter 3.12 Extended SDO Access).

3.11 Modulo function

The “Modulo” function is handled by 4 parameters, this parameters are nested under “GeneralSettings. The Modulo function is used for rotative axis applications.

In the previous example (Chapter 3.6) we already looked at two of these four parameters, the “Software Position Limits”. The other 2 parameters are called “PositionRangeLimits”.
The “Modulo function” must be initialized by setting these four parameters in a specific way



Setting up different function mode with following combinations:

- a ”**Non-Modulo**” axis when
  “PosRangeLimitMin” < “SoftPosLimitMin” < any allowable position < “SoftPosLimitMax” < “PosRangeLimitMax”

- a ”**Modulo**” axis **with “Smart Direction”** when
  “SoftPosLimitMin” < “PosRangeLimitMin” < any allowable position < “PosRangeLimitMax” < “SoftPosLimitMax”

  “Smart Direction” means the axis will choose the shortest path, for example from 10° to 355° it will turn back by 15°, anticlockwise.
  The “Position Range Limits” define the positions that can be programmed as target positions.

- a **”Modulo”** axis **without “Smart Direction”** when
  “SoftPosLimitMin” = “PosRangeLimitMin” < any allowable position < “PosRangeLimitMax” = “SoftPosLimitMax”

  Without “Smart Direction” the axis always move forward if (setpoint-measure) > 0 and backward otherwise.

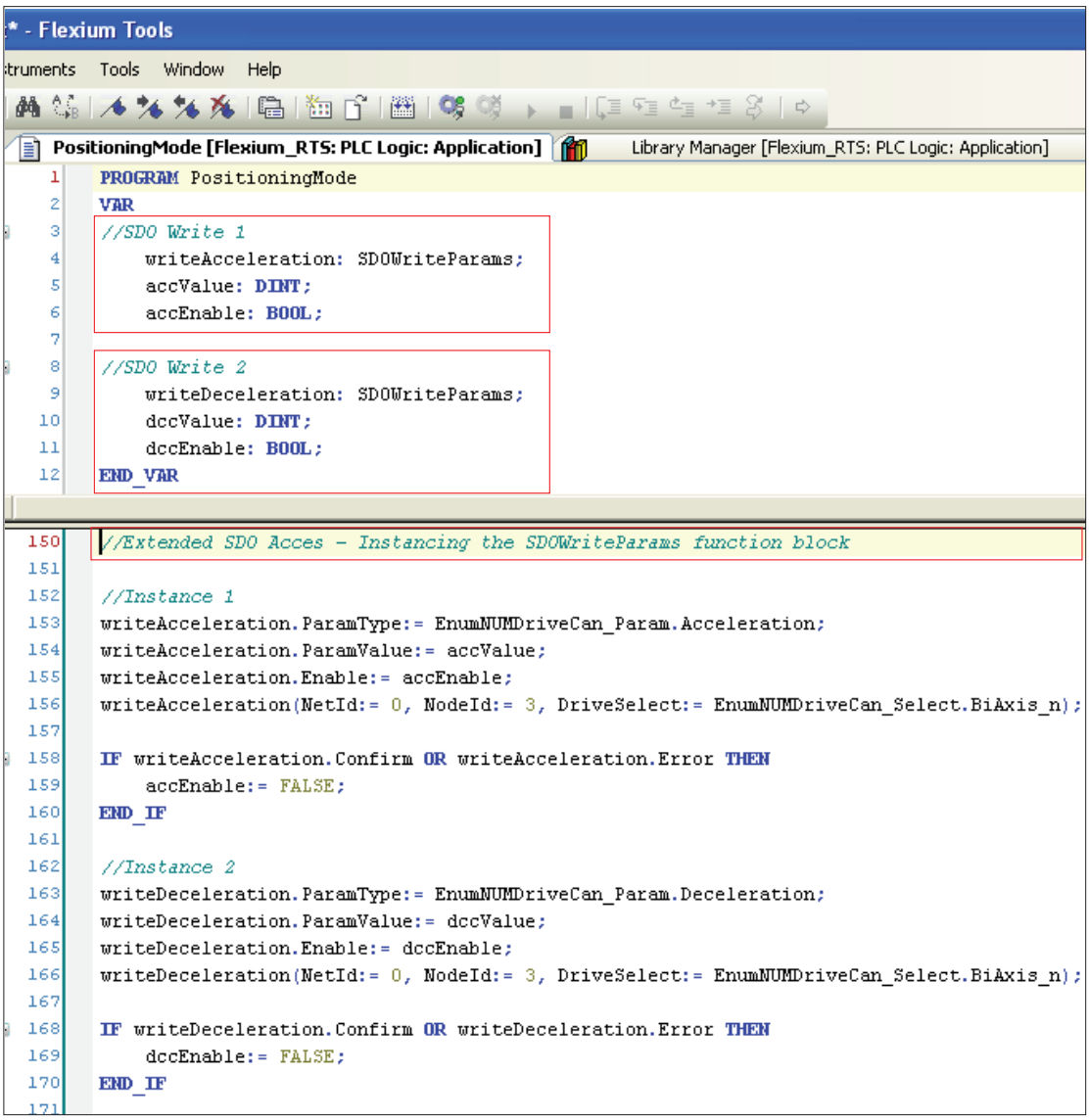
Of course, the “non smart” mode is useful if your mechanics can’t tolerate an endless rotation because the motor cables will twist around the motor. In “non smart” mode the movement never crosses the top point.

3.12 Extended SDO Access

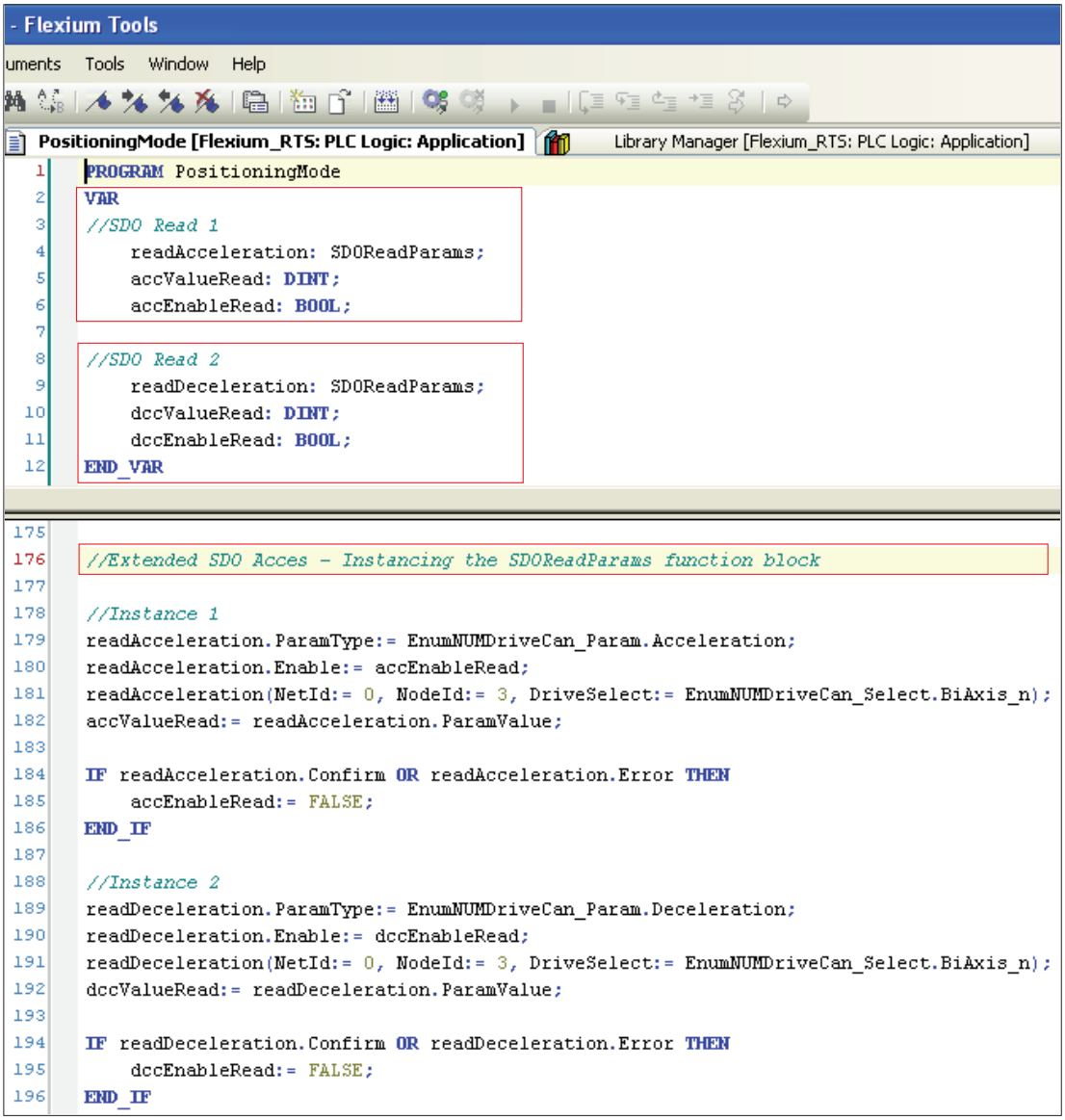
If your application needs many read and write transmissions to SDO objects there exists another possibility to realize a SDO transmission to a CANopen device.
Take a look at the library manager and you will find two function blocks in the folder “SDOAccess Functions” called “SDOReadParams” and “SDOWriteParams”.

This function blocks can be instanced several times and used for transmission to SDO objects without organizing the order of the transmission access like it is mandatory when you use the “myInstance.SDOAccess.SetParameters” function.

Here is the example for two write requests:



Here is the example for two read requests:



3.13 Synchronous Start in Positioning Mode

To enable the synchronous start for 2 or more motors some additional configuration steps are needed for the demanded motor drives.

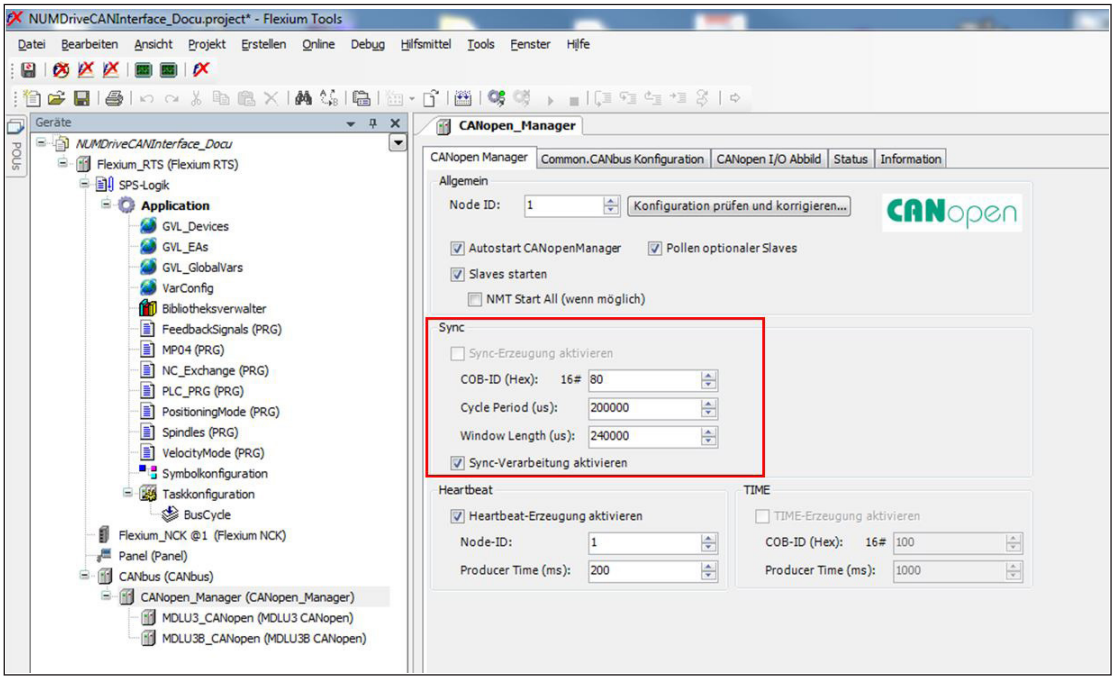
This setting can change your I/O address mapping, if you have already set up this CANopen Drive without Synch Mode.

It is recommended to set up the whole CANbus after the EtherCAT bus in the device tree and using a high address area which is over all the others before. Otherwise it is possible that a modification of the Safe I/Os or EtherCAT I/Os can change your CANbus address areas.

If you make a Reset in Synch Mode, the internal Reset procedure takes some time. To have a control over this time there is a status bit *"myInstance. SyncStatus.ResetSyncInExecution"* which gives you an information while the reset is running in sync mode.
During this time the *"ResetSyncInExecution"* bit is TRUE and it is not allowed to make any other commands during that time.

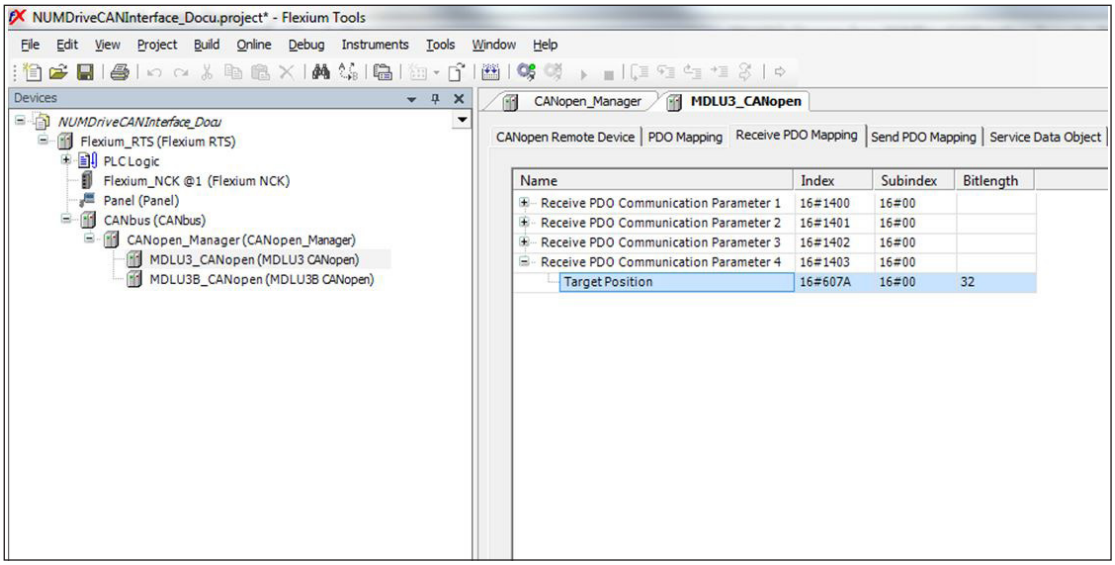
First of all, enable the Sync Consuming of the CANopen Manager and set the parameters in the Sync group box. "COB-ID" must be 80hex (this entry was 128 dec. before FlexiumTools V3.5.0.0!) (to accord to the fixed Sync-ID of 80h) produced by the "NUMDriveCANInterface" library.

"Cycle Period" should be the same value as in the library instance parameter "SyncCycle" which is described later in this chapter.
"Window Length" should be the value of "Cycle Period" multiplied with factor 1, 2.



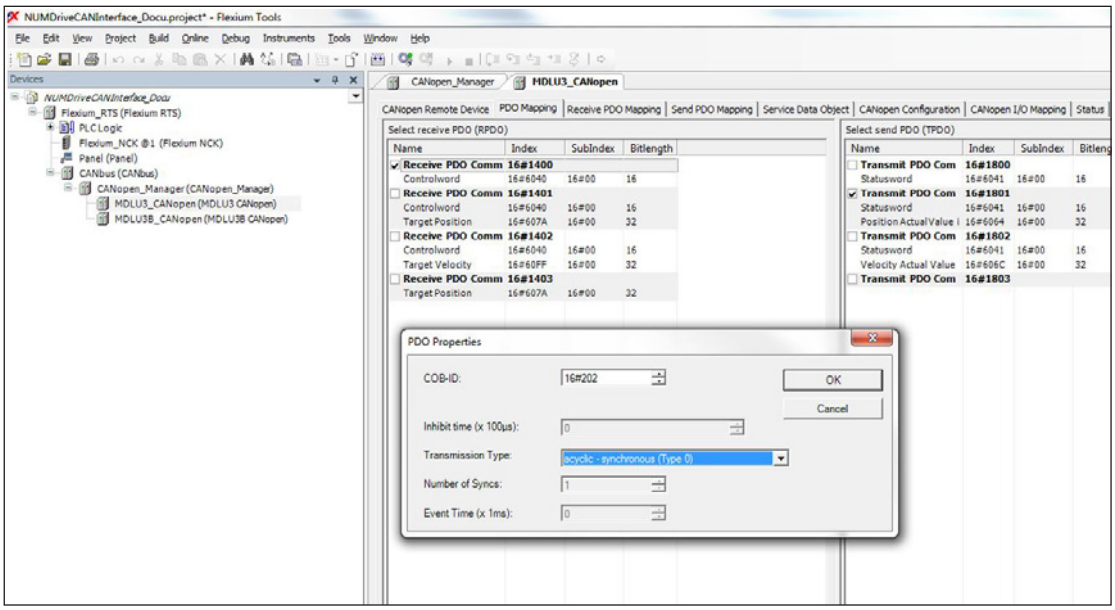
Open the slide “Receive PDO Mapping“.

Map the “*Target Position*“ to Receive PDO 4 (click on “*Receive PDO Communication Parameter 4*” and then click on the button below “*Add Mapping…*”. Select “Target Position” in the appearing dialog box).

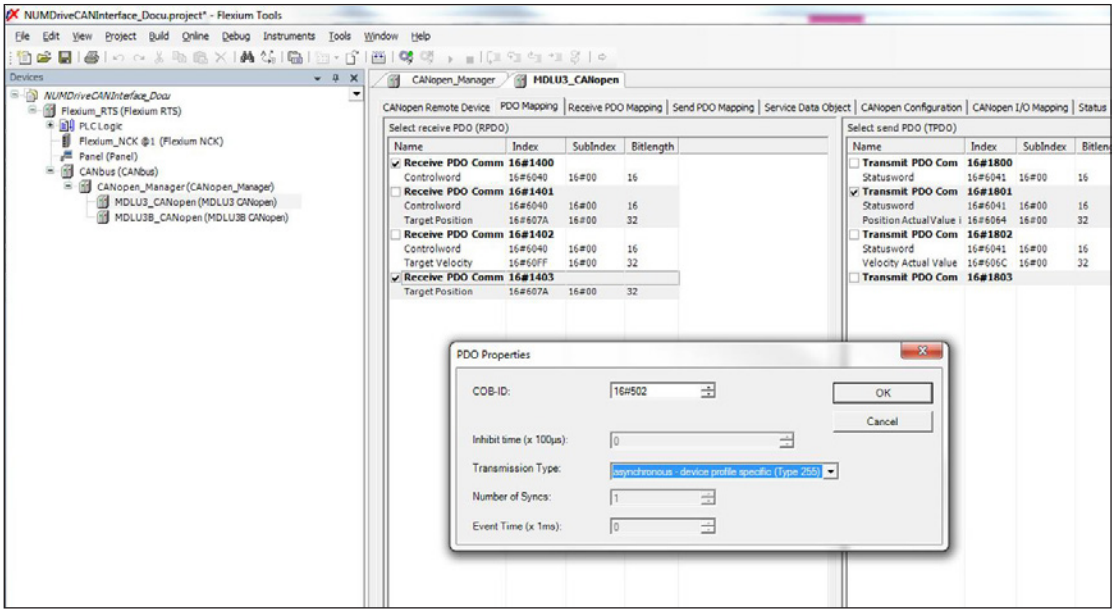


Open the slide “PDO Mapping“.

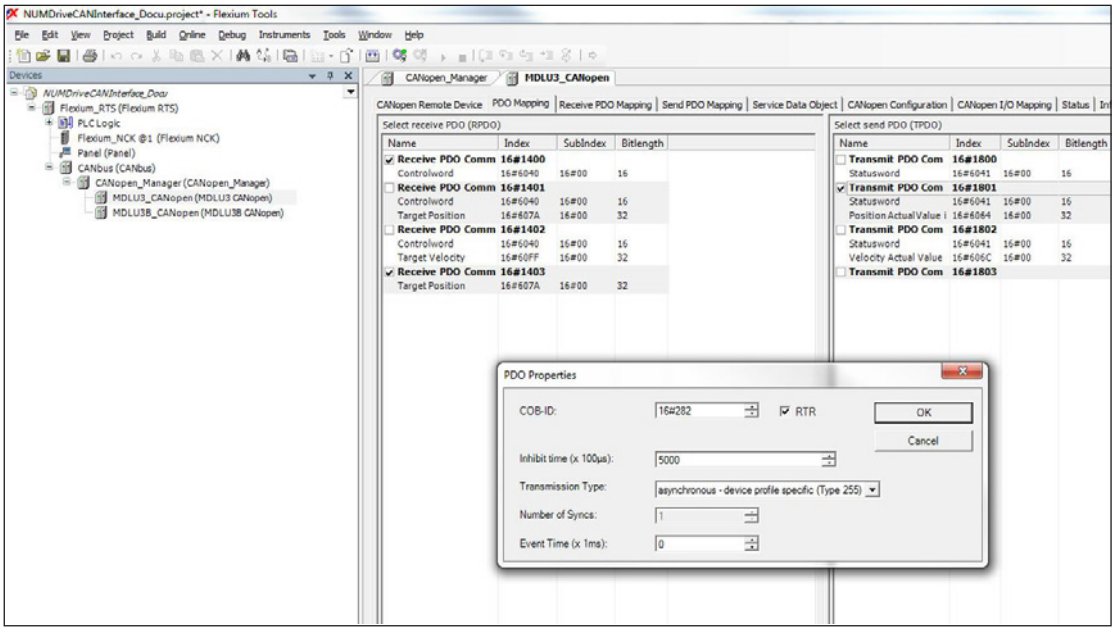
Open the dialog box for “*PDO Properties*” by double clicking “*Receive PDO Comm*”.
Enable Receive PDO 1 and set the Transmission Type to “*acyclic - synchronous (Type 0)*”.



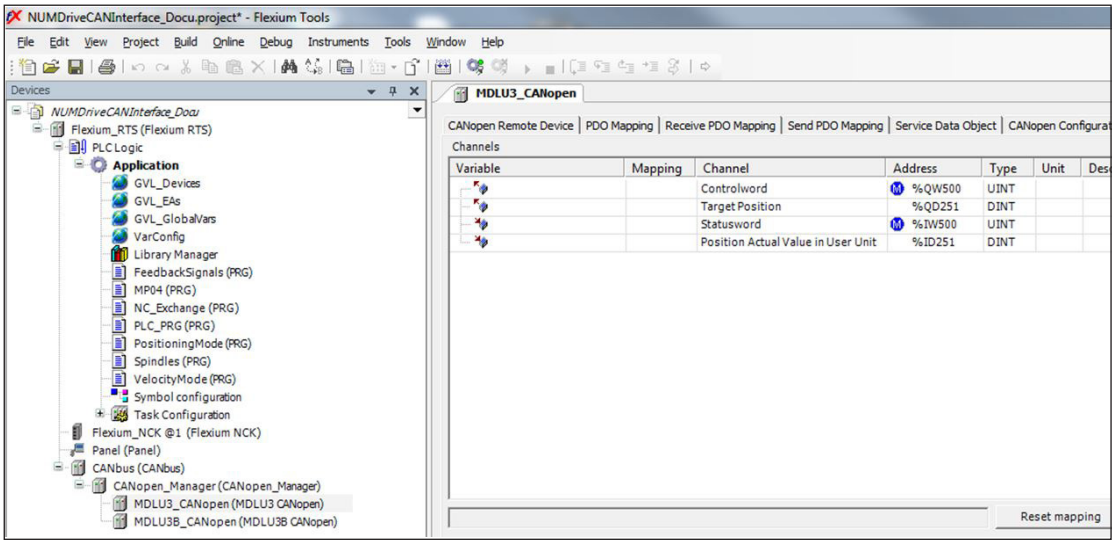
Enable Receive PDO 4 and set the Transmission Type to “*asynchronous - device profile specific (Type 255)*”.



Enable Transmit PDO 2 and set the Transmission Type to “*asynchronous - device profile specific (Type 255)*”.



Please consider the correct I/O mapping after this changes:



For Bi-Axes Drives it is necessary to make all this drive specific and PDO mapping specific settings for both drive parts, Drive_n and Drive_n+1.

The next step is to enable the *Sync function* in your instance of "NUMDriveCAN_PositioningMode" and set the Sync parameter variables to the correct values.
To enable the *Sync Signal* for the Synch Network you have to declare one (only one) Synch Master and set up the needed Master parameters:

IsSyncMaster	Only one Drive in the Network can be the Synch Master which controls the Synch Signal.
SyncCycle	The needed parameter fo this function is the SyncCycle. This is the cycle period of the synch signal.
SyncTransmissionDelay	This parameter is a delay time for waiting the feedback signals ("*StartPositioningAssumed*" and "*ChangePosImmediatAssumed*") from the Drives in the synch network after a "*StartPositioning*" or "*ChangePosImmediat*" command. It is possible to enlarge this delay time (default = 200ms) if there are many axes in the synch network and there are problems with the feedback signals ("*StartPositioningAssumed*" and "*ChangePosImmediatAssumed*") from the drives.

Example for the Synch signal Master drive in Synch mode:

```
//General Settings for Synch Network Master Axes - Bi-Axes drive n - (can PositioningAxes)
canPositioningAxis.GeneralSettings.NetId:= 0;
canPositioningAxis.GeneralSettings.NodeId:= 3;
canPositioningAxis.GeneralSettings.DriveSelect:= NUMDriveCANInterface.EnumNUMDriveCan_Select.BiAxis_n;

canPositioningAxis.GeneralSettings.SyncSelect:= TRUE;              //If this Drive is integrated to the Synch network - else this bitt must be False!
canPositioningAxis.Synchronous.IsSyncMaster:= TRUE;                //Only one Drive can be the Synch Master - all other Drives must be set to FALSE!
canPositioningAxis.Synchronous.SyncCycle:= 200;                    //The Cycle Period of the Synch Signal
canPositioningAxis.Synchronous.SyncTransmissionDelay:= T#300MS;    //Internat Time Constant - Default = 200ms
//- Enlarge this value only if there are many Axes and you don't get the Feedback Signals "StartPositioningAssumed" or "ChangePosImmediatAssumed" of all Axes
```

All other drives in the synch network are not allowed to be declared as "*IsSyncMaster*", but must be enabled for the synch mode with "*SyncSelect:= TRUE*".

Example for the the other (not Master) drives in Synch mode:

```
//General Settings for Synch Network - rest of the Axes - Bi-Axes drive n+1 - (can PositioningAxes)
canPositioningAxis_2.GeneralSettings.NetId:= 0;
canPositioningAxis_2.GeneralSettings.NodeId:= 3;
canPositioningAxis_2.GeneralSettings.DriveSelect:= NUMDriveCANInterface.EnumNUMDriveCan_Select.BiAxis_n1;

canPositioningAxis_2.GeneralSettings.SyncSelect:= TRUE;            //If this Drive is integrated to the Synch network - else this bitt must be False!
canPositioningAxis_2.Synchronous.IsSyncMaster:= FALSE;             //Only one Drive can be the Synch Master - all other Drives must be set to FALSE!

//General Settings for Synch Network - rest of the Axes - Mono-Axes drive x (can PositioningAxes)
canPositioningAxis_x.GeneralSettings.NetId:= 0;
canPositioningAxis_x.GeneralSettings.NodeId:= 4;
canPositioningAxis_x.GeneralSettings.DriveSelect:= NUMDriveCANInterface.EnumNUMDriveCan_Select.MonoAxis

canPositioningAxis_x.GeneralSettings.SyncSelect:= TRUE;            //If this Drive is integrated to the Synch network - else this bitt must be False!
canPositioningAxis_x.Synchronous.IsSyncMaster:= FALSE;             //Only one Drive can be the Synch Master - all other Drives must be set to FALSE!
```

After these settings the "*SynchronousMode*" for starting several drives at the same moment is activated.

Of course, the "*StartPositioning*" or "*ChangePosImmediat*" command for all drives in the synch network, which should start synchronous, must be set in the same PLC cycle.

```
//Prepare the next Positioning Start in Sync Mode
canPositioningAxis.DriveControl.AutoControl.StartPositioning:= btn_Start_Pos;
canPositioningAxis_2.DriveControl.AutoControl.StartPositioning:= btn_Start_Pos;
canPositioningAxis_x.DriveControl.AutoControl.StartPositioning:= btn_Start_Pos;
```

Also the new target position, motor speed etc. must be already set before enabling the "*StartPositioning*" or "*ChangePosImmediat*".

After the command "*StartPositioning*" or "*ChangePosImmediat*" was set, every single drive of the synch network answers with the bit "myInstance.SyncStatus.StartPositioningAssumed" or "myInstance.SyncStatus.ChangePosImmediatAssumed" to show, that the drive is really ready to start the movement.

To enable the movement of all the needed drives you have to collect this feedback bits and if all conditions are as you like command the real movement start with the bit "myInstance.Synchronous.StartSyncMovement" for every drive you want to start the movement.

```
// Collect all Feedback Signals and start the real movement after the feedback conditions are rigth
StartPositioningAssumed_1:= canPositioningAxis.SyncStatus.StartPositioningAssumed;
StartImmediateAssumed_1:= canPositioningAxis.SyncStatus.ChangePosImmediatAssumed;

StartPositioningAssumed_2:= canPositioningAxis_2.SyncStatus.StartPositioningAssumed;
StartImmediateAssumed_2:= canPositioningAxis_2.SyncStatus.ChangePosImmediatAssumed;

StartPositioningAssumed_3:= canPositioningAxis_x.SyncStatus.StartPositioningAssumed;
StartImmediateAssumed_3:= canPositioningAxis_x.SyncStatus.ChangePosImmediatAssumed;

canPositioningAxis.Synchronous.StartSyncMovement:= canPositioningAxis_2.Synchronous.StartSyncMovement:= canPositioningAxis_x.Synchronous.StartSyncMovement:= FALSE;
IF  canPositioningAxis.DriveControl.Mode = EnumNUMDriveCan_Mode.Auto THEN
    IF StartPositioningAssumed_1 AND StartPositioningAssumed_2 AND StartPositioningAssumed_3 THEN
        canPositioningAxis.Synchronous.StartSyncMovement:= canPositioningAxis_2.Synchronous.StartSyncMovement:= canPositioningAxis_x.Synchronous.StartSyncMovement:= TRUE;
    END_IF
    IF StartImmediateAssumed_1 AND StartImmediateAssumed_2 AND StartImmediateAssumed_3 AND btn_Start_Pos_Immediate THEN
        canPositioningAxis.Synchronous.StartSyncMovement:= canPositioningAxis_2.Synchronous.StartSyncMovement:= canPositioningAxis_x.Synchronous.StartSyncMovement:= TRUE;
    END_IF
END_IF
```

When you command the "myInstance.Synchronous.StartSyncMovement" bit, the movement will start immediately for this commanded drive.

3.14 Example for extended axis status information in Positioning Mode

Some additional axis status information which are not available in the standard exchange zone of the library, can be reached by using this code snippet.
The additional status is dependent of the current state of the axis, so only one of this conditions can be valid at the same time. This fact you have to consider in your program code!

The additional information is:

- Axis is decelerating
- Axis has stopped (Reference is disabled)
- Target Position is reached
- Target Position is not reached.

```
IF (UINT_TO_WORD(canPositioningAxis.GeneralSettings.InpCanDrive.statusWord) AND 16#4)  = 4 THEN
    IF (UINT_TO_WORD(canPositioningAxis.GeneralOutput.OutCanDrive.controlWord) AND 16#100)  = 0 THEN  //Halt Bit = 0
        IF (UINT_TO_WORD(canPositioningAxis.GeneralSettings.InpCanDrive.statusWord) AND 16#400)  = 1024 THEN
            AxisCondition:= 1;      //Reference is enabled - Target Position is reached
        ELSE
            AxisCondition:= 0;      //Reference is enabled - Target Position is not reached
        END_IF
    ELSE        //Halt Bit = 1 //Axis is in Halt = Reference disabled
        IF (UINT_TO_WORD(canPositioningAxis.GeneralSettings.InpCanDrive.statusWord) AND 16#400)  = 1024 THEN
            AxisCondition:= 2;      //Reference is disabled - Axis has stopped
        ELSE
            AxisCondition:= 3;      //Reference is disabled - Axis is decelerating
        END_IF
    END_IF
ELSE
    AxisCondition:= 2;      //Reference is disabled - Axis has stopped
END_IF


// AxisCondition description:
// 0 = Target Position is not reached
// 1 = Target Position is reached
// 2 = Reference is disabled - Axis has stopped
// 3 = Axis is decelerating

AxisCondition;  //Data type = INT
```

Page deliberately empty.

Summary

4 Flexium CANopen Axes state machine and operating modes .155
4.1 NMT management services: the Communication state machine 155
4.2 Error control mechanisms 157
4.3 Device Control 158
4.3.1 State machine 158
4.3.1.1 State device 159
4.3.1.2 State transitions 160
4.3.2 Conditions preventing state machine transitions 164
4.4 CANopen Device Profile 165
4.4.1 Standardized operating modes 165
4.4.2 Profile position operating mode 166
4.4.2.1 Profile position objects description 168
4.4.3 Profile velocity operating mode 169
4.4.3.1 Profile velocity objects description 170
4.4.4 Homing operating mode 170
4.4.4.1 Homing operating objects description 173
4.4.5 Interpolation operating mode 174
4.4.5.1 Interpolation operating objects description 176
4.4.6 Hardware limit switches and HOME switches 177
4.4.6.1 Configuration for the NUMDrive C CANopen Mono-Axis 177
4.4.6.2 Configuration for the NUMDrive C CANopen Bi-Axes 177
4.4.6.3 Limit switch status display with WPM 178

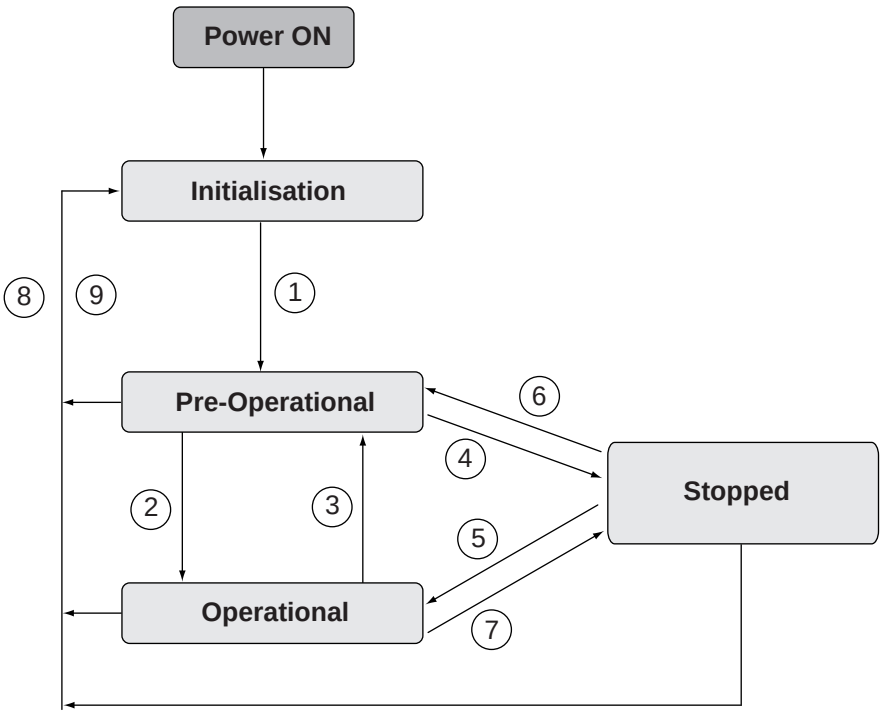
Page deliberately empty.

4 Flexium CANopen Axes state machine and operating modes

The correct operation of a Flexium CANopen Axis require the handling of 2 separate state machines: the Communication state machine and the Device control state machine.
Both should be commanded to evolve to an operational state to have the motion.

4.1 NMT management services: the Communication state machine

NMT commands are used to initialise, start, stop and monitor the devices in the network.
CANopen defines a minimal boot-up procedure for slave devices in the network:



(X) Please refer to the here below detailed description.

The device contains only 3 states PRE-OPERATIONAL, OPERATIONAL and STOPPED.
After Power on the device is going to be initialised and set in the PRE-OPERATIONAL status automatically (1).
In this state reading and writing to its object dictionary via SDO is possible.

The device can now be configured: that means setting of objects or changing default values and so on. Only the master controls the following state transitions of each device in the network.
The device can be switched into the OPERATIONAL state via the NMT command Start Remote Node in order to start PDO communication (2) (5).

PDO communication can be stopped by the network master by simply switching the slave back to PRE-OPERATIONAL state by using the Enter Pre-Operational State NMT command (3) (6).

Via the Stop Remote Node NMT command the master can force the slave to the state STOPPED (4) (7). In this state no besides network and error control mechanisms are available.

The NMT command Reset Communication resets the communication on the node (8).
All communication parameters will be set to their default values.

The application will be reset by Reset Node NMT command (9): this command resets all application parameters and call Reset Communication command (not available in NUMDC_CAN).

The NMT services for device control are sent with the COB-Id = 0 (top priority). The data frame of the NMT service consists of 2 bytes:

- Byte 0: Command specifier
- Byte 1: Node address (a message with the node address 0 is a broadcast message)

Command Specifier	NMT command	Transition
01h	Start Remote Node	2.5
02h	Stop Remote Node	4.7
80h	Enter Pre-Operational State	3.6
81h	Reset Node	9
82h	Reset Communication	8

4.2 Error control mechanisms

For node monitoring two different mechanisms are defined: Node Guarding and Heartbeat. NUMDC_CAN supports both these error control mechanisms.

The Node Guarding is a periodical monitoring of certain network nodes. Each node can be checked from the master with a certain period (Guard time 100Ch). A second parameter Life time factor 100Dh defines a factor when the connection should be applied as lost.
The resolution of the guarding time is 1 ms.

Heartbeat is an error control service without need for remote frames.
The Heartbeat-producer transmits cyclic a heartbeat message (Heartbeat-Consumer Time 1017h). One or more Heartbeat-consumer receives this message and monitors this indication. For monitoring the Heartbeat-consumer has a entry at its object dictionary: the Heartbeat-Consumer Time 1016h.

If a consumer does not receive a signal within the time intervals defined in this object, it generates an error message. The time intervals are set in 1 ms steps and must be set greater for the consumer than for the producer.

4.3 Device Control

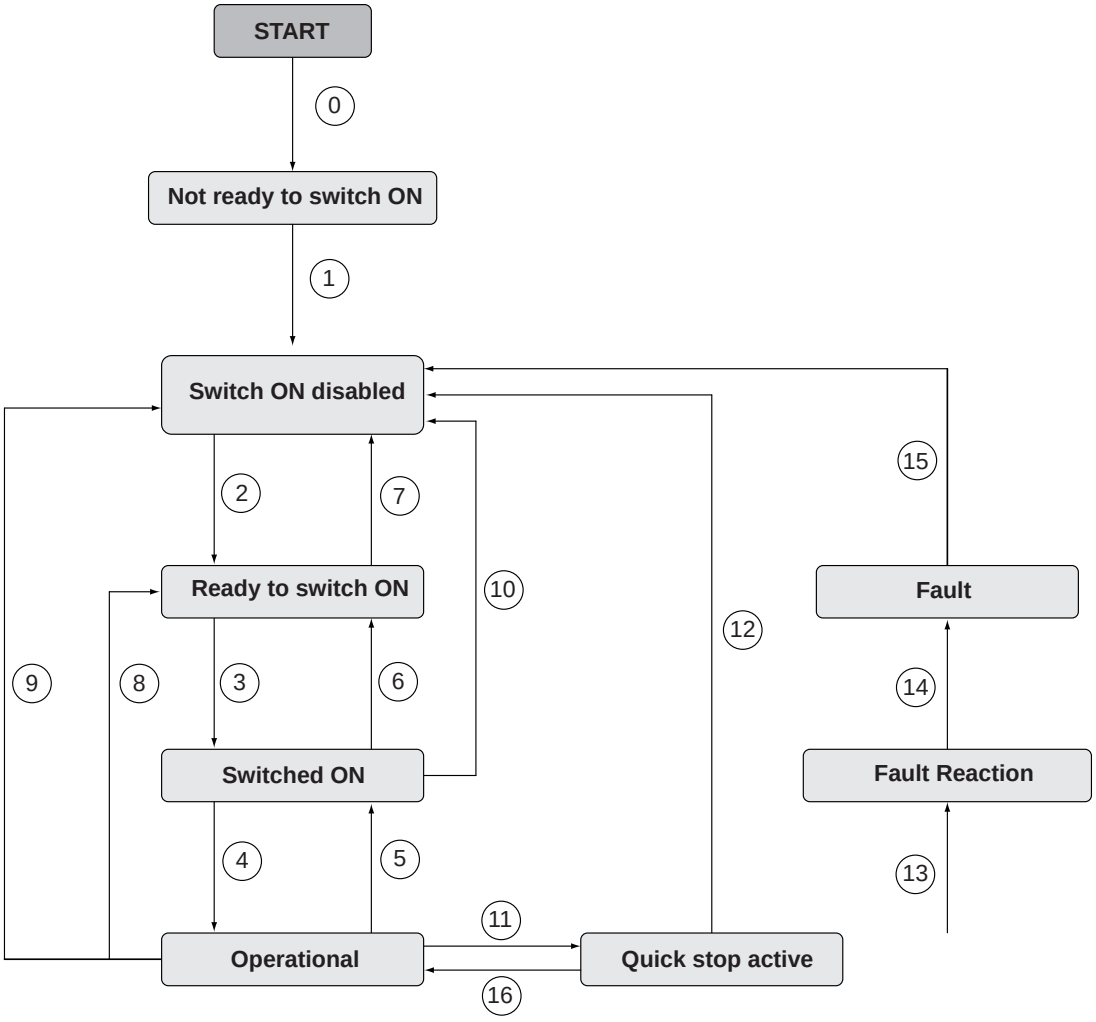
A complete device control with its functionalities and features is made by:

- the controlling of the states
- the controlling of operating modes

The state of the drive can be controlled by the Controlword 6040h and showed by Statusword 6041h. In remote mode the device is controlled directly from the CANopen network by PDO and SDO.

4.3.1 State machine

The state machine describes the device status and the possible control sequences of the drive. A single state represents a special internal or external behavior. The relationship between the operating states and the state transitions is shown in the state machine diagram below:



(X) Please refer to next page “State transition x” for the detailed description.

4.3.1.1 State device

The following states of the device are possible:

State device	Description
Not ready to switch ON	• Low level power has been applied to the drive. • The drive is being initialized or is running self test. • The drive function is disabled.
Switch ON disable	• Drive initialization is complete. • The drive parameters have been set up. • Drive parameters may be changed. • High voltage may not be applied to the drive. • The drive function is disabled.
Ready to switch ON	• Drive parameters may be changed. • High voltage may be applied to the drive. • The drive function is disabled.
Switch ON	• High voltage has been applied to the drive. • The power is ready. • Drive parameters may be changed. • The drive function is disabled.
Operation enable	• The drive function is enabled and power is applied to the motor. • No fault has been detected. • Drive parameters may be changed.
Quick stop active	• The drive function is enabled and power is applied to the motor. • Drive parameters may be changed. • The quick stop function is being executed.
Fault reaction active	• The drive function is enabled and power is applied to the motor. • Drive parameters may be changed. • A fault has occurred in the drive. • The quick stop function is being executed.
Fault	• Drive parameters may be changed. • A fault has occurred in the drive. • High voltage switched off. • The drive function is disabled.

4.3.1.2 State transitions

State transitions are caused by internal events in the drive or by commands from the network via the Controlword. Please also refer to the diagram on the previous page.

State device	Description
State transition 0	Start → Not ready to switch ON
Event	Reset
Action	The drive starts initialization phase

State transition 1	Not ready to switch ON → Switch ON disabled
Event	The drive has initialized and self-tested successfully
Action	Activate communication

State transition 2	Switched ON disabled → Ready to switch ON
Event	'Shutdown' command received from network
Action	During this transition a RUN/HALT/RUN sequence is executed

State transition 3	Ready to switch ON → Switched ON
Event	'Switch ON' command received from network. Note. This transition is executed only when: - The DC bus is present. - The NUM-STO Safety module (if present) is not in allarm state.
Action	The power section is switched ON

State transition 4	Switched ON → Operation enable
Event	'Enable operation' command received from network
Action	The drive functions is enabled

State transition 5	Operation enable → Switched ON
Event	'Disable operation' command received from network
Action	The drive operation will be disabled

State transition 6	Switched ON → Ready to switch ON
Event	'Shutdown' command received from network
Action	The power section is switched OFF

State transition 7	Ready to switch ON → Switch ON disabled
Event	'Quick stop' and 'disable voltage' commands received from network
Action	None

State transition 8	Operation enable → Ready to switch ON
Event	- 'Shutdown' command received from network. - DC bus not more present. - NUM-STO Safety module intervention.
Action	The power section is switched OFF immediately and the motor is free to rotate if unbraked

State transition 9	Operation enable → Switch ON disabled
Event	'Disable voltage' command received from network
Action	The power section is switched OFF immediately and the motor is free to rotate if unbraked

State transition 10	Switch ON → Switch on disabled
Event	'Disable voltage' or 'quick stop' command received from network
Action	The power section is switched OFF immediately and the motor is free to rotate if unbraked

State transition 11	Operation enable → Quick stop active
Event	'Quick stop' command received from network
Action	The quick stop function is executed

State transition 12	Quick stop active → Switch ON disabled
Event	The ‘disable voltage’ command is received from the network
Action	The power section is switched OFF

State transition 13	All states → Fault reaction active
Event	A fault is occurred in the drive
Action	Execute appropriate fault reaction

State transition 14	Fault reaction active → Fault
Event	The fault reaction is completed
Action	The drive function is disabled. The power section may be switched OFF

State transition 15	Fault → Switch ON disabled
Event	‘Fault reset’ command received from network
Action	A reset of the fault condition is carried out if no fault exists currently on the drive. After leaving the state Fault the bit ‘Fault Reset’ of the controlword has to be cleared by the network

State transition 16	Quick stop active → Operation enable
Event	‘Enable operation’ command received from network
Action	The drive functions are re-enabled

Controlword 6040h

The control word 6040h consists of bits for:
- the controlling of the states
- the controlling of operating mode
- manufacturer specific options.

15..11	10..9	8	7	6..4	3	2	1	0
Manufacturer specific	Reserved	Halt	Fault reset	Operation mode specific	Enable operation	Quick stop	Enable voltage	Switch ON

MSB LSB

Bits 0 ÷ 3 and 7 are mandatory and are used like device control commands:

Command	**Controlword bit**					**Transitions**
	Fault reset	**Enable operation**	**Quick stop**	**Enable voltage**	**Switched ON**	
Shutdown	0	X	1	1	0	2, 6, 8
Switch ON	0	0	1	1	1	3
Switch ON	0	1	1	1	1	3
Disable voltage	0	X	X	0	X	7, 9, 10, 12
Quick stop	0	X	0	1	X	7, 10, 11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4
Fault reset	0 →1	X	X	X	X	15

Bit 4-6	These bits are operation mode specific. For more detailed information please refer to chapter 4.4.2, 4.4.3 and 4.4.4.
Bit 8	HALT bit. For more detailed information please refer to chapter 4.4.2, 4.4.3 and 4.4.4
Bit 11-15	These bits are manufacturer specific. For more detailed information please refer to chapter 4.4.2, 4.4.3 and 4.4.4.

Statusword 6041h

The Statusword 6041h indicates the current status of the drive. No bits are latched.
The statusword consists of bits for:

- the current status of the drive
- the operating state of the mode
- manufacturer specific options.

Bit	M / O	Description
0	M	Ready to switch ON
1	M	Switched ON
2	M	Operation enabled
3	M	Fault
4	M	Voltage enable
5	M	Quick stop
6	M	Switched on disabled
7	O	Warning
8	O	Manufacturer specific
9	M	Remote
10	M	Target reached
11	M	Internal limit active
12-13	O	Operation mode specific
14-15	O	Manufacturer specific
M = Mandatory. O = Optional.		

Bits 0 ÷ 3, 5 and 6 indicate the status of the device:

Value (binary)	State
xxxx xxxx x0xx 0000	Not ready to switch ON
xxxx xxxx x1xx 0000	Switch ON disabled
xxxx xxxx x01x 0001	Ready to switch ON
xxxx xxxx x01x 0011	Switched ON
xxxx xxxx x01x 0111	Operation enabled
xxxx xxxx x00x 0111	Quick stop active
xxxx xxxx x0xx 1111	Fault reaction active
xxxx xxxx x0xx 1000	Fault

Bit 4	High voltage is applied to the drive when this bit is set to 1.
Bit 7	A drive warning is present if it is set. This bit is set and reset by the device.
Bit 9	If the bit is set, then parameters may be modified via the CAN network. If the bit is reset, then the drive is in local mode.
Bit 10	If the bit is set, then a set point has been reached. If halt occurred and the drive has halted then this bit is set too.
Bit 11	This bit set by the drive indicates that an internal limitation is active (e.g. position range limit)
Bit 12-13	These bits are operation mode specific. For more detailed information please refer to chapter 4.4.2, 4.4.3 and 4.4.4.
Bit 14-15	These bits are manufacturer specific. For more detailed information please refer to chapter 4.4.2, 4.4.3 and 4.4.4.

4.3.2 Conditions preventing state machine transitions

Some conditions could prevent the required transition to be executed.

- The transition 3 from “Ready to switch ON” to “Switched ON” is not executed if there is the “PWM-OFF” condition (bit 7 of 1AF1, one can always see this communication word using WPM) because of SAM problems.

- The transition 3 from “Ready to switch ON” to “Switched ON” is not executed if there isn’t the DC bus voltage.

- The transition 4 from “Switched ON” to “Operation enabled” is not executed if something prevent the drive to become with “TORQUE REF” and “SPEED REF” enabled.
  The request for “TORQUE REF” and “SPEED REF” are issued by the state machine itself, but there are conditions that prevent the execution of these commands (for example, safety modules not ready, “CNC control excluded” by means of WPM, wrong setting of V221, loss of DC bus).

- If the drive is in “Operation enabled” and the DC bus is lost for some reasons, the state machine jump back to “Ready to switch ON” with the transition 8 (eventually after the braking if the motor was moving).

When we say “the transition from xx to yy is not executed” we don’t mean it will be never executed if that opportunity is lost; because the command word is a “state” command, the transition will be executed as soon as the preconditions become satisfacted.

4.4 CANopen Device Profile

A device profile defines a standard device. For these standard devices a basic functionalities have been specified, which every device within a class has to exhibit.

The CANopen Device Profile ensures a minimum of identical behavior for a kind of devices. Each device has to fulfill the requirements on the behavior i.e. the implemented application state machine. Further it has to support all mandatory objects, parameters and data for the device.

The reference Device Profile for Drive and Motion Control is CiA-402.

4.4.1 Standardized operating modes

NUMDC_CAN operates in 4 standardized operating modes:

- Profile position
- Profile velocity
- Homing (see chapter 5.7.6 and next pages)
- Cyclic reference, interpolating mode.

Standardized operating modes can be started and monitored with the following objects of the CANopen device profile.
The Modes of operation 6060 h (Mandatory) switches the actually chosen operation mode.

Value	Description
-1 … -128	Manufacturer specific modes of operation
0	Reserved
1	Profile Position Mode
2	Velocity Mode (not supported)
3	Profile velocity Mode
4	Torque Profile Mode (not supported)
5	Reserved
6	Homing Mode
7	Interpolated Position Mode
8 … 127	Reserved

The actual mode of the drive is reflected in the object Modes of operation display 6061 h (Mandatory) that shows the current mode of operation. The meaning of the returned value corresponds to that of the modes of operation object (index 6060 h).

The drive supports more then one and several distinct modes of operation. Supported drive modes 6502 h (Optional) gives an overview of the implemented operating modes in the device. This object is read only.

31..6	5	4..3	2	1	0
Reserved/not used	Homing	Reserved/not used	Profile velocity	Not used	Profile position
MSB					LSB

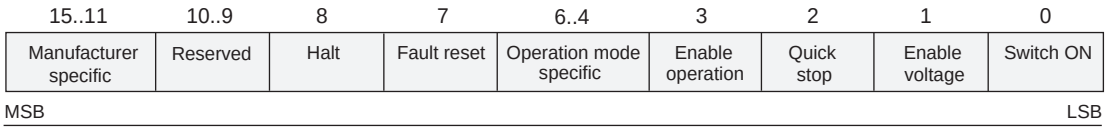
4.4.2 Profile position operating mode

In this operating mode a movement with a programmable travel profile is run from a actual position to a target position. A movement profile can be set with programmable values for acceleration and deceleration ramps and final speed. Here NUMDrive C object dictionary entries related to this operating mode.

The profile position mode is controlled by some bits of Controlword 6040h and Statusword 6041h as indicated in the tables below

The Controlword 6040h consists of bits for:

- the controlling of the states
- the controlling of operating mode
- manufacturer specific options.



Controlword 6040h

Bit	Name	Value	Description
4	New set-point	0	Does not assume target position
		1	Assume target position (start)
5	Change set immediately	0	Finish the actual positioning and then start the next positioning
		1	Interrupt the actual positioning and start the next positioning
6	abs/rel	0	Target position is an absolute value
		1	Target position is a relative value
8	Halt	0	Execute positioning
		1	Stop axis with profile deceleration

The Statusword 6041h consists of bits for:

15..14	13..12	11	10	9	8	7	6	5	4	3	2	1	0
Manufact. specific	Operation mode specific	Internal limit active	Target reached	Remote	Manufact. specific	Warning	Switched ON disabled	Quick Stop	Voltage Enable	Fault	Operation Enabled	Switched ON	Ready to Switch ON

MSB LSB

Statusword 6041h bits

Bit	Name	Value	Description
10	Target reached	0	Halt = 0: Target position not reached Halt = 1: Axis decelerates
		1	Halt = 0: Target position reached Halt = 1: Velocity of axis is 0
12	Set-point acknowledge	0	Trajectory generator has not assumed the positioning values (yet)
		1	Trajectory generator has assumed the positioning values
13	Following error	0	No following error
		1	Following error
14	PP mode target position reached*	0	Current position outside Position window 6x67h
		1	Current position inside Position window 6x67h
15	Reference**	0	Drive has not valid reference point
		1	Drive has valid reference point

NOTE that bit 10 change meaning according to HALT bit state.

* Manufacturer specific: it indicates whether current positioning is inside positioning window; irrespective from the positioning being interrupted by a “Halt” or a fault.
** Manufacturer specific.

4.4.2.1 Profile position objects description

<table>
<tr><th>Object</th><th>Description</th></tr>
<tr><td>Position actual value 6064h</td><td>Represents the actual value of the position measurement device in user-defined units.</td></tr>
<tr><td>Following error window 6065h</td><td>Defines a range of tolerated position values symmetrically to the position demand value. If the position actual value is out of the following error window, a following error occurs. If the value of this object is 2^{32}-1, the following control is switched off.</td></tr>
<tr><td>Following error time out 6066h</td><td>Indicates the configured time for a following error condition.
The value is given in ms. If, for all the configured time, a following error occurs, the bit 13 of the status word will be set to 1. If, after that, the following error condition disappears, the bit 13 of the status word will be reset to 0.</td></tr>
<tr><td>Position window 6067h</td><td>Defines a symmetrical range of accepted positions relatively to the target position. If the actual value of the position encoder is within the position window, this target position is regarded as reached.If the value of this object is 2^{32}-1, the position windows control is switched off.</td></tr>
<tr><td>Target position 607Ah</td><td>It is the position that the drive should move to using the current settings of motion control parameters such as velocity, acceleration, deceleration, etc. It is given in user defined position units. The target position value can be interpreted as either a relative or absolute value depending on bit 6 in Controlword.
In case of relative value the positioning path is specified relative to the previous target position. In case of absolute value the positioning path is specified with reference to the zero point of the axis (defined with the homing operating mode before the first absolute positioning).</td></tr>
<tr><td>Position range limit 607Bh</td><td>Contains the sub-parameters min position range limit (subindex 1) and max position range limit (subindex 2). On reaching or exceeding these limits the input value shall wrap automatically to the other end of the range.</td></tr>
<tr><td>Software position limit 607Dh</td><td>Contains the sub-parameters min position limit (subindex 1) and max position limit (subindex 2). These parameters define the absolute position limits for the position demand value and the position actual value.
Every new target position must be checked against these limits. The limit positions are specified in user defined units (same as target position) and are always relative to the machine home position.</td></tr>
<tr><td>Max profile velocity 607Fh</td><td>It is the maximum allowed speed in either direction during a profiled move. It is given in the same units as profile velocity.</td></tr>
<tr><td>Profile velocity 6081h</td><td>It is the velocity normally attained at the end of the acceleration ramp during a profiled move. It is valid for both directions of motion and it is given in user defined.</td></tr>
<tr><td>Profile acceleration 6083h</td><td>It is given in user defined acceleration units.</td></tr>
<tr><td>Profile deceleration 6084h</td><td>It is given in the same units as profile acceleration.</td></tr>
<tr><td>Motion profile type 6086h</td><td>It is used to select the type of motion profile used to perform a profiled move.

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>-32768 … -1</td><td>Manufacturer specific</td></tr>
<tr><td>0</td><td>Linear ramp (trapezoidal profile)</td></tr>
<tr><td>1</td><td>Sin^2 ramp (not supported)</td></tr>
<tr><td>2</td><td>Jerk-free ramp (not supported)</td></tr>
<tr><td>3</td><td>Jerk-limited ramp (non supported)</td></tr>
<tr><td>4 … 32767</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>Following error actual value 6081h</td><td>It represents the actual value of the following error, it is given in user defined position units.</td></tr>
</table>

4.4.3 Profile velocity operating mode

In this operating mode NUMDC_CAN accelerates to an programmable Set Point speed. A movement profile can be set with programmable values for acceleration and deceleration.
Here NUMDC_CAN object dictionary entries related to this operating mode.

The profile velocity mode is controlled by some bits of Controlword 6040h and Statusword 6041h as indicated in the tables below:

The Controlword 6040h consists of bits for:

- the controlling of the states
- the controlling of operating mode
- manufacturer specific options.

15..11	10..9	8	7	6..4	3	2	1	0
Manufacturer specific	Reserved	Halt	Fault reset	Operation mode specific	Enable operation	Quick stop	Enable voltage	Switch ON

MSB LSB

Controlword 6040h bits

Bit	Name	Value	Description
8	Halt	0	Execute the motion
		1	Stop axis with profile deceleration

The Statusword 6041h consists of bits for:

15..14	13..12	11	10	9	8	7	6	5	4	3	2	1	0
Manufact. specific	Operation mode specific	Internal limit active	Target reached	Remote	Manufact. specific	Warning	Switched ON disabled	Quick Stop	Voltage Enable	Fault	Operation Enabled	Switched ON	Ready to Switch ON

MSB LSB

Statusword 6041h bits

Bit	Name	Value	Description
10	Target reached	0	Halt = 0: Target velocity not (yet) reached
		1	Halt = 1: Axis decelerates
12	Speed	0	Speed is not equal to 0
		1	Speed is equal to 0
14	Operating mode finished*	0	Not finished
		1	Finished
15	Reference**	0	Drive has not valid reference point
		1	Drive has valid reference point

* Manufacturer specific: the operating mode is completed and motor standstill achieved by “Halt”, by an error or after a preset default = 0.
** Manufacturer specific.

The output value of the trajectory generator may be corrected by the output value of the position control function.
It is then provided as a demand value for the velocity controller Velocity demand value 606Bh. It is given in the velocity units.

4.4.3.1 Profile velocity objects description

Object	Description
Velocity actual value 606Ch	(actual speed of the motor) It is also represented in the velocity units.
Velocity window 606Dh	Monitors whether the required process velocity has been achieved after an eventual acceleration or deceleration phase. It is also represented in the velocity units.
Target Velocity 60FFh	It is the input for the trajectory generator (set point speed) and the value is in the velocity units.

4.4.4 Homing operating mode

In homing operating mode an absolute scale reference of the motor position at a defined axis position is established. This chapter describes the method by which a drive seeks the home position also called reference point or zero point.

There are various methods of achieving this using limit switches at the ends of travel or a home switch (zero point switch) in mid-travel, most of the methods also use the index (zero) pulse train from an incremental encoder. The user can specify method of homing and speeds.

There are 2 homing speed; in a typical cycle the faster speed is used to find the home switch and the slower speed is used to find the index pulse. Unlike the other operating modes, a homing movement must be completed before to switch to a new operating mode.

The homing mode is controlled by some bits of of Controlword 6040h and Statusword 6041h as indicated in the tables below:

The Controlword 6040h consists of bits for:

- the controlling of the states
- the controlling of operating mode
- manufacturer specific options.

15..11	10..9	8	7	6..4	3	2	1	0
Manufacturer specific	Reserved	Halt	Fault reset	Operation mode specific	Enable operation	Quick stop	Enable voltage	Switch ON

MSB LSB

Controlword 6040h bits

Bit	Name	Value	Description
4	Homing operation start	0	Homing mode inactive
		0 --> 1	Start homing mode
		1	Homing mode active
		0 --> 1	Interrupt homing mode
8	Halt	0	Execute the instruction of bit 4
		1	Stop axis with homing deceleration

The Statusword 6041h consists of bits for:

15..14	13..12	11	10	9	8	7	6	5	4	3	2	1	0
Manufact. specific	Operation mode specific	Internal limit active	Target reached	Remote	Manufact. specific	Warning	Switched ON disabled	Quick Stop	Voltage Enable	Fault	Operation Enabled	Switched ON	Ready to Switch ON

MSB LSB

Statusword 6041h bits

Bit	Name	Value	Description
10	Target reached	0	Halt = 0: Home position not reached Halt = 1: Axis decelerates
		1	Halt = 0: Home position reached Halt = 1: Velocity of axis is 0
12	Homing attained	0	Homing mode not yet completed
		1	Homing mode carried out successfully
13	Homing error	0	No homing error
		1	Homing mode carried out not successfully
14	Homing completed*	0	Homing in progress
		1	Homing completed, motor at a standstill
15	Reference*	0	Drive has not valid reference point
		1	Drive has valid reference point

* Manufacturer specific.

The Homing method 6098h object determines the method that will be used during homing. By choosing a method of homing by writing a value to homing method will clearly establish:

- the homing signal (positive limit switch, negative limit switch, home switch)
- the direction of actuation
- the position of the index pulse.

Value	Description
-128 … -1	Manufacturer specific
0	No homing operation request
1	LIMN with index pulse
2	LIMP with index pulse
7	REF+ with index pulse, inv., outside
8	REF+ with index pulse, inv., inside
9	REF+ with index pulse, not inv., inside
10	REF+ with index pulse, not inv., outside
11	REF- with index pulse, inv., outside
12	REF- with index pulse, inv., inside
13	REF- with index pulse, not inv., inside
14	REF- with index pulse, not inv., outside
17	LIMN
18	LIMP
23	REF+, inv., outside
24	REF+, inv., inside
25	REF+, not inv., inside
26	REF+, not inv., outside
27	REF-, inv., outside
28	REF-, inv., inside
29	REF-, not inv., inside
30	REF-, not inv., outside
33	Index pulse negative direction
34	Index pulse positive direction
35	Homing on the current position
36 … 127	Reserved

where:

- LIMN : homing on the negative limit switch
- LIMP : homing on the positive limit switch
- REF+ : search home switch with movement in positive direction of rotation
- REF- : search home switch with movement in negative direction of rotation
- inv. : invert direction in switch
- not inv. : not invert direction in switch
- outside : index pulse/distance outside switch
- inside : index pulse/distance inside switch.

4.4.4.1 Homing operating objects description

Object	Description
Homing speed 6099h	The object defines the speeds used during homing and is given velocity units: speed during search for switch (subindex 1) and speed during search for zero (subindex 2)
Home offset 607Ch	The object defines the configured difference between the zero position for the application and the machine home position found during homing. Once the homing is completed, the zero position is offset from the home position by adding the home offset..
Homing acceleration 609Ah	The object indicates the acceleration and deceleration to be used during homing operation.
Please refer to chapter 4.4.6 for the instructions to connect limit and homing switches.	

4.4.5 Interpolation operating mode

In IP mode the drive cyclically receives a new position setpoint slightly advanced with respect to the previous, and continuously moves approaching to the new target.

To have synchronized behaviour between different drives on the network a SYNC message is broadcasted by the master at fixed intervals. All drives will apply the (previously) received command word and set point at the moment of the SYNC received (i.e. the received data are buffered until the SYNC), and all the drives will freeze statusword and measure at the moment of the SYNC, and then transmit them before the next SYNC. Of course the SYNC period must be selected long enough to avoid excessive busload.

The CANopen model uses the SYNC also as time base for interpolation: the period of the SYNC is also the base period for interpolation (the ratio is 1:1 as we don't support 60C3h).
Due to the structure of the CAN network, some jitter on the transmission of the SYNC is unavoidable.
Besides the infinite possibilities foreseen by the CiA specification, we only support linear IP, 1-record buffer length, updated every cycle - the most similar to our CNC working mode. Although it's not compulsory, it's also advisable to configure the TPDO used to get the measure as synchronous, every-SYNC (type 0001h).

Here are the key points for the use of **I**nter**P**olation mode:

- **Synchronized use of setpoint and freeze of measures**
  The IP mode require from the master the transmission of a SYNC message to have a time synchronization signal common to all drives.
  The cyclical sequence of events required is:
  - the master send to every drive (in sequence) the PDOs of the new commandword and the new position setpoint. The 'commandword.bit4' can be 0 (interpolation not active), and the received setpoint will not be used; or it can be 1 (interpolation active) and the setpoint is used. Commandword.bit5 and 6 have no meaning in IP mode and must be 0. The drive take the values but don't use it until a SYNC is received.
  - the master send a SYNC, and all the drives apply the received setpoints and freeze the current measure.
  - after the SYNC, all the drives (in sequence) will send to the master statuswords and measures.

- **No need for handshake on new setpoint**
  The profile position mode require an handshake on start/acknowledge. The master should send the new setpoint in 607A, then the START bit in the commandword, then wait the ACK bit in the status word, then it should clear the START bit.
  Such a sequence is inefficient in the case of a cyclic transmission like interpolated mode, so the handshake is not required in this case. The master should simply send the setpoint (in 60C1h.1) and the commandword with the bit 4 (=interpolation active) at 1 every interpolation cycle.

- **Different meaning of some bits in commandword / statusword**
  - Commandword.4 means "interpolation active"
  - Commandword.5 is reserved and must be written to 0
  - Commandword.6 is reserved and must be written to 0
  - Statusword.10 acts like in PP mode, meaning "target reached" in HALT=0 or "speed 0" if HALT=1
  - Statusword.12 means "interpolation active", and so is basically a copy of Commandword.4
  - Statusword.14 (manufacturer specific) means "target reached" every time.

- **60C1h**
  In IP mode the position setpoint must be sent to 60C1h.1, not to 607Ah.

- **SYNC issues**
  The object 60C2h is the interpolation period. In NUM drives there is always V266 for the same reason; so the drive really use the latter, and the CANopen parameter is present only for conformity. When the drive switch to OPERATION ENABLE and the mode of operation (6060h) is IP, the drive will generate an alarm A16.266 if V266 is different from the value sent by the master in 60C2h.1. Of course this means the master must send an appropriate value to 60C2.1 in the INIT phase, otherwise this object contain the default value, likely different from V266.
  In Bi-Axis drives, an alarm A16.266 is generated also if the 2 drives have V266 different from each other. (Please refer to the chapter 12)

- **T/RPDO suggestion**
  In order to reduce the busload it would be useful to have one RPDO carrying both the commandword and the position setpoint for interpolation (60C1h.1). This RPDO is not standard, so we suggest to use the reconfigurable RPDO4 for this purpose.
  The following code is a suggestion to configure such a mapping:

```
Send 605h 8 2Fh 03h 16h 00h 00h 00h 00h 00h    // deactivate mapping: 0->subindex 0
Send 605h 8 23h 03h 16h 01h 10h 00h 40h 60h    // add mapped object 1: byte0..1, object 6040h, subidx 0, 16bit
Send 605h 8 23h 03h 16h 02h 20h 01h C1h 60h    // add mapped object 2: byte2..5, object 60C1h, subidx 1, 32bit
Send 605h 8 2Fh 03h 16h 00h 02h 00h 00h 00h    // activate mapping: 2 mapped objects->subindex 0
```

  This RPDO must be configured as “synchronous-every SYNC” to be used with SYNC:

```
Send 605h 8 2Fh 03h 14h 02h 01h 00h 00h 00h   // TPDO2: trasmission type (01h = every SYNC)
```

  A good choice would be to configure also TPDO2 (66 for the axis N+1) to get statusword and measure together after every SYNC:

```
Send 605h 8 2Fh 01h 18h 02h 01h 00h 00h 00h   // TPDO2: trasmission type (01h = every SYNC)
```

The interpolation mode is controlled by some bits of Controlword 6040h and Statusword 6041h as indicated in the tables below:

Controlword 6040h bits

Bit	Name	Value	Description
4	Interpolation active	0	Interpolation mode is not active, setpoints will not be used
		1	Interpolation mode is active, setpoint will be used as next target
5	Reserved	0	Write as ‘0’
6	Reserved	0	Write as ‘0’
8	Halt	0	Execute the instruction of bit 4
		1	Stop axis with homing deceleration

Statusword 6041h bits

Bit	Name	Value	Description
10	Target reached	0	Halt = 0: (last) target not reached Halt = 1: Axis decelerates
		1	Halt = 0: (last) target reached Halt = 1: Velocity of axis is 0
12	Interpolation active	0	Homing mode not yet completed
		1	Homing mode carried out successfully
13	Target reached*	0	(last) target not yet reached
		1	(last) target reached
15	Reference*	0	Drive has not valid reference point
		1	Drive has valid reference point
* Manufacturer specific.			

4.4.5.1 Interpolation operating objects description

Object	Description
Mode of operation 6060h	It should be set to 07h to select the interpolation mode
Interpolation sub-mode 60C0h	Select the interpolation mode. Since only linear interpolation is supported, it's value can be only 00h.
Interpolation data record 60C1h	It has 2 sub-index: - Subindex 0 (highest s.index supported) is fixed at 01h; - Subindex 1 (data record 1) should be cyclically written with the position setpoint.
Interpolation time period 60C2h	It has 3 sub-index: - Subindex 0 (highest s.index supported) is fixed at 02h; - Subindex 1 (interpolation time period value) should be written with the same value of V266; - Subindex 2 (interpolation time period index) is fixed at -3 and means: milliseconds.
Supported drive modes 6502h	(6D02 for the Axis N+1 on the Bi-Axes) has the bit 6 = 1 to indicate the availability of interpolation mode.

4.4.6 Hardware limit switches and HOME switches

NUMDrive C CANopen drives will provide the connection for 2 hardware limit switches (proximity) to every axis. Adding these connections to the already present “Home switch” connection, this means a Bi-Axes must have 6 GP-IN terminals.

These connections are arranged as follow:
- 2 inputs from the existing GP-IN_A, GP-IN_B;
- 2 inputs from the “PROGO1/2” terminals, now of bidirectional type;
- 2 inputs from the terminal 4, 6 on S8 connector.

All these signals are active LOW by default (this setting can be changed by means of parameters). This is a safety-related settings (if the related parameter V139 is not modified from the default value and the limit switches are not connected an alarm will be generated).
To remove an unnecessary complexity, these connection are fixed and not configurable on CANopen drives. Please refer to the AMOMAN012 Installation Manual to make all these connections.

4.4.6.1 Configuration for the NUMDrive C CANopen Mono-Axis

- The HOMING switch is always GP-IN B;
- The 2 inputs on S8 (pin 4 and pin 6) are always limit switches. They are active LOW by default (connect to +24V to allow the move), unless the logic is inverted by means of V139 bit 10.

This configuration is selected to allow the availability of “PROGO” on Mono-Axis.
Please refer to the AMOMAN012 Installation Manual to make all these connections.

4.4.6.2 Configuration for the NUMDrive C CANopen Bi-Axes

- The HOMING switch for $Axis_n$ is always GP-IN_A
- The HOMING switch for $Axis_{n+1}$ is always GP-IN_B
- The 2 “new” inputs GP-IN_C and GP-IN_D on PROGO1-2 can be used as limit switches for the $Axis_n$ or as usual outputs for relay (i.e. brake); the selection between input/output is done in V22, bit 2, 3.

- The 2 inputs on S8 (pin 4 and pin 6) are always limit switches for the $Axis_{n+1}$.

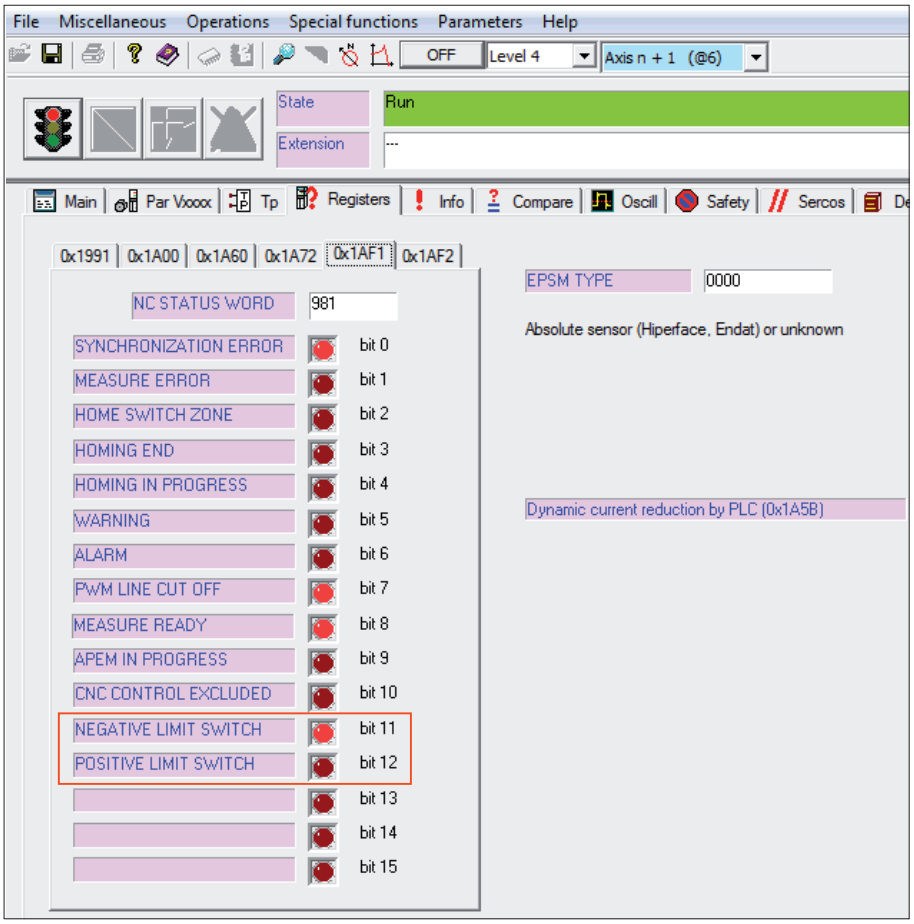
The limit switches are active LOW by default (connect to +24V to allow the move), unless the logic is inverted by means of V139 bit 10. Please refer to chapter 5.5.1.

Please refer to the AMOMAN012 Installation Manual to make all these connections.

4.4.6.3 Limit switch status display with WPM

By WPM it is possible to displaty the “logical” status of the limit switches in 1AF1.11 (negative) and 1AF1.12 (positive). The LED lit when the limit switch is reached.
See bits 11, 12 as indicated in the figure below

The limit switches alarms are active only during movement commanded through the CANopen interface; if you move the axis using the digital console of WPM the limit switches are not taken into account.



Page deliberately empty.

Page deliberately empty.

Summary

5 NUMDrive C with CANopen interface 189
5.1 NUMDrive C configuration with WPM 189
5.1.1 Motor / drive association and sensor configuration with WPM 189
5.1.1.1 Motor / drive selection 189
5.1.1.2 Sensor configuration (X1, X2, X3, X4) 190
5.1.2 NUM sensor User's configuration 191
5.1.2.1 NUM sensor table example 192
5.1.3 Drive parameters display 193
5.1.3.1 Parameters selection and modification 194
5.1.3.2 Bit parameters selection and modification 195
5.1.3.3 Drive filter selection and modification 196
5.2 NUMDrive C Mono-Axis objects description 197
5.2.1 NC Units 197
5.2.2 Objects description 197
Object 1000h: Device Type 197
Object 1001h: Error Register 198
Object 1003h: Pre-defined Error Field 199
Object 1005h: COB-ID SYNC 200
Object 1006h: Communication Cycle Period 200
Object 1012h: COB-ID Time Stamp 201
Object 100Ch: Guard Time 201
Object 100Dh: Life Time Factor 202
Object 1014h: COB-ID EMCY 203
Object 1015h: Inhibit Time Emergency 203
Object 1016h: Heartbeat Consumer Entries 204
Object 1017h: Producer Heartbeat Time 205
Object 1018h: Identity Object 206
Object 1020h: Verify Configuration 208
Object 1200h: Server SDO Parameter 1 209
Object 1201h: Server SDO Parameter 2 210
Object 1400h: Receive PDO Communication Parameter 1 212
Object 1401h: Receive PDO Communication Parameter 2 214
Object 1402h: Receive PDO Communication Parameter 3 216
Object 1403h: Receive PDO Communication Parameter 4 218
Object 1600h: Receive PDO Mapping Parameter 1 220
Object 1601h: Receive PDO Mapping Parameter 2 221
Object 1602h: Receive PDO Mapping Parameter 3 222
Object 1603h: Receive PDO Mapping Parameter 4 223
Object 1800h: Transmit PDO Communication Parameter 1 225
Object 1801h: Transmit PDO Communication Parameter 2 227
Object 1802h: Transmit PDO Communication Parameter 3 229
Object 1803h: Transmit PDO Communication Parameter 4 231
Object 1A00h: Transmit PDO Mapping Parameter 1 233
Object 1A01h: Transmit PDO Mapping Parameter 2 234
Object 1A02h: Transmit PDO Mapping Parameter 3 236
Object 1A03h: Transmit PDO Mapping Parameter 4 237
Object 3000h: Drive internal status flags 239
Object 3001h: TPDO trigger transmission window 240
Object 3002h: RPDO1 shadow before SYNC validation 241
Object 6007h: Abort Connection Option Code 242
Object 603Fh: Error Code 243
Object 6040h: Controlword 243

Object 6041h: Statusword....244
Object 605Ah: Quick Stop Option Code....245
Object 6060h: Modes of Operation....246
Object 6061h: Modes of Operation Display....247
Object 6064h: Position Actual Value in User Unit....247
Object 6065h: Following Error Window....248
Object 6066h: Following Error Time Out....248
Object 6067h: Position Window....249
Object 606Ch: Velocity Actual Value....249
Object 606Dh: Velocity Window....250
Object 607Ah: Target Position....251
Object 607Bh: Position Range Limit....252
Object 607Ch: Home Offset....253
Object 607Dh: Software Position Limit....254
Object 607Fh: Max Profile Velocity (associated to V98)....256
Object 6081h: Profile Velocity in pp-mode....257
Object 6083h: Profile Acceleration....258
Object 6084h: Profile Deceleration....258
Object 6085h: Quick Stop Deceleration....259
Object 6086h: Motion Profile Type....259
Object 6098h: Homing Method....260
Object 6099h: Homing Speeds....261
Object 609Ah: Homing Acceleration....262
Object 60C0h: Interpolation Sub Mode Select....262
Object 60C1h: Interpolation Data Record....263
Object 60C2h: Interpolation Time Period....264
Object 60F2h: Positioning Option Code....265
Object 60FDh: Digital Inputs....266
Object 60FFh: Target Velocity....267
Object 6502h: Supported Drive Modes....268
5.2.3 Object dictionary access rights....269
5.3 NUMDrive C Bi-Axes objects description....271
5.3.1 NC Units....271
5.3.2 Objects description....271
Object 1000h: Device Type....271
Object 1001h: Error Register....272
Object 1003h: Pre-defined Error Field....273
Object 1005h: COB-ID SYNC....274
Object 1006h: Communication Cycle Period....274
Object 100Ch: Guard Time....275
Object 100Dh: Life Time Factor....276
Object 1012h: COB-ID Time Stamp....277
Object 1014h: COB-ID EMCY....278
Object 1015h: Inhibit Time Emergency....278
Object 1016h: Heartbeat Consumer Entries....279
Object 1017h: Producer Heartbeat Time....280
Object 1018h: Identity Object....281
Object 1020h: Verify Configuration....283
Object 1200h: Server SDO Parameter 1....284
Object 1201h: Server SDO Parameter 2....285
Object 1400h: Receive PDO Communication Parameter 1 [$Axis_n$]....287
Object 1401h: Receive PDO Communication Parameter 2 [$Axis_n$]....289
Object 1402h: Receive PDO Communication Parameter 3 [$Axis_n$]....291
Object 1403h: Receive PDO Communication Parameter 4 [$Axis_n$]....293
Object 1440h: Receive PDO Communication Parameter 65 [$Axis_{n+1}$]....295
Object 1441h: Receive PDO Communication Parameter 66 [$Axis_{n+1}$]....297
Object 1442h: Receive PDO Communication Parameter 67 [$Axis_{n+1}$]....299
Object 1443h: Receive PDO Communication Parameter 68 [$Axis_{n+1}$]....301
Object 1600h: Receive PDO Mapping Parameter 1 [$Axis_n$]....303
Object 1601h: Receive PDO Mapping Parameter 2 [$Axis_n$]....304
Object 1602h: Receive PDO Mapping Parameter 3 [$Axis_n$]....305
Object 1603h: Receive PDO Mapping Parameter 4 [$Axis_n$]....306

Object 1640h: Receive PDO Mapping Parameter 65 [Axis$_{n+1}$] 308
Object 1641h: Receive PDO Mapping Parameter 66 [Axis$_{n+1}$] 309
Object 1642h: Receive PDO Mapping Parameter 67 [Axis$_{n+1}$] 310
Object 1643h: Receive PDO Mapping Parameter 68 [Axis$_{n+1}$] 311
Object 1800h: Transmit PDO Communication Parameter 1 [Axis$_n$] 313
Object 1801h: Transmit PDO Communication Parameter 2 [Axis$_n$] 315
Object 1802h: Transmit PDO Communication Parameter 3 [Axis$_n$] 317
Object 1803h: Transmit PDO Communication Parameter 4 [Axis$_n$] 319
Object 1840h: Transmit PDO Communication Parameter 65 [Axis$_{n+1}$] 321
Object 1841h: Transmit PDO Communication Parameter 66 [Axis$_{n+1}$] 323
Object 1842h: Transmit PDO Communication Parameter 67 [Axis$_{n+1}$] 325
Object 1843h: Transmit PDO Communication Parameter 68 [Axis$_{n+1}$] 327
Object 1A00h: Transmit PDO Mapping Parameter 1 [Axis$_n$] 329
Object 1A01h: Transmit PDO Mapping Parameter 2 [Axis$_n$] 330
Object 1A02h: Transmit PDO Mapping Parameter 3 [Axis$_n$] 332
Object 1A03h: Transmit PDO Mapping Parameter 4 [Axis$_n$] 333
Object 1A40h: Transmit PDO Mapping Parameter 65 [Axis$_{n+1}$] 335
Object 1A41h: Transmit PDO Mapping Parameter 66 [Axis$_{n+1}$] 336
Object 1A42h: Transmit PDO Mapping Parameter 67 [Axis$_n$] 337
Object 1A43h: Transmit PDO Mapping Parameter 68 [Axis$_n$] 338
Object 3000h: Drive internal status flags [Axis$_n$] 340
Object 3001h: TPDO trigger transmission window [Axis$_n$] 341
Object 3002h: RPDO1 shadow before SYNC validation [Axis n] 342
Object 3100h: Drive internal status flags [Axis$_{n+1}$] 343
Object 3101h: TPDO trigger transmission window [Axis$_{n+1}$] 343
Object 3102h: RPDO65 shadow before SYNC validation [Axis n+1] 344
Object 6007h: Abort Connection Option Code [Axis$_n$] 345
Object 603Fh: Error Code [Axis$_n$] 346
Object 6040h: Controlword [Axis$_n$] 346
Object 6041h: Statusword [Axis$_n$] 347
Object 605Ah: Quick Stop Option Code [Axis$_n$] 348
Object 6060h: Modes of Operation [Axis$_n$] 349
Object 6061h: Modes of Operation Display [Axis$_n$] 350
Object 6064h: Position Actual Value in User Unit [Axis$_n$] 350
Object 6065h: Following Error Window [Axis$_n$] 351
Object 6066h: Following Error Time Out 351
Object 6067h: Position Window [Axis$_n$] 352
Object 606Ch: Velocity Actual Value (Axes$_n$) 352
Object 606Dh: Velocity Window [Axis$_n$] 353
Object 607Ah: Target Position [Axes$_n$] 354
Object 607Bh: Position Range Limit [Axes$_n$] 355
Object 607Ch: Home Offset [Axes$_n$] 356
Object 607Dh: Software Position Limit [Axes$_n$] 357
Object 607Fh: Max Profile Velocity [Axes$_n$] 359
Object 6081h: Profile Velocity in pp-mode [Axis$_n$] 360
Object 6083h: Profile Acceleration [Axis$_n$] 361
Object 6084h: Profile Deceleration [Axis$_n$] 361
Object 6085h: Quick Stop Deceleration [Axis$_n$] 362
Object 6086h: Motion Profile Type [Axis$_n$] 362
Object 6098h: Homing Method [Axis$_n$] 363
Object 6099h: Homing Speeds [Axis$_n$] 364
Object 609Ah: Homing Acceleration [Axis$_n$] 365
Object 60C0h: Interpolation Sub Mode Select [Axis$_n$] 365
Object 60C1h: Interpolation Data Record [Axis$_n$] 366
Object 60C2h: Interpolation Time Period [Axis$_n$] 367
Object 60F2h: Positioning Option Code 368
Object 60FDh: Digital Inputs [Axis$_n$] 369
Object 60FFh: Target Velocity [Axis$_n$] 370
Object 6502h: Supported Drive Modes [Axis$_n$] 371
Object 67FFh: Single Device Type [Axis$_n$] 372
Object 6807h: Abort Connection Option Code [Axis$_{n+1}$] 372
Object 683Fh: Error Code [Axis$_{n+1}$] 373
Object 6840h: Controlword [Axis n+1] 373

Object 6841h: Statusword [Axis$_{n+1}$] 374
Object 685Ah: Quick Stop Option Code [Axis$_{n+1}$] 374
Object 6860h: Modes of Operation [Axis$_{n+1}$] 375
Object 6861h: Modes of Operation Display [Axis$_{n+1}$] 375
Object 6864h: Position Actual Value in User Unit [Axis$_{n+1}$] 376
Object 6865h: Following Error Window [Axis$_{n+1}$] 376
Object 6866h: Following Error Time Out 377
Object 6867h: Position Window [Axis$_{n+1}$] 377
Object 686Ch: Velocity Actual Value [Axis$_{n+1}$] 378
Object 686Dh: Velocity Window [Axis$_{n+1}$] 378
Object 687Ah: Target Position [Axis$_{n+1}$] 379
Object 687Bh: Position Range Limit [Axis$_{n+1}$] 380
Object 687Ch: Home Offset [Axis$_{n+1}$] 381
Object 687Dh: Software Position Limit [Axis$_{n+1}$] 382
Object 687Fh: Max Profile Velocity [Axis$_{n+1}$] 383
Object 6881h: Profile Velocity in pp-mode [Axis$_{n+1}$] 383
Object 6883h: Profile Acceleration [Axis$_{n+1}$] 384
Object 6884h: Profile Deceleration [Axis$_{n+1}$] 384
Object 6885h: Quick Stop Deceleration [Axis$_{n+1}$] 385
Object 6886h: Motion Profile Type [Axis$_{n+1}$] 385
Object 6898h: Homing Method [Axis$_{n+1}$] 386
Object 6899h: Homing Speeds [Axis$_{n+1}$] 387
Object 689Ah: Homing Acceleration [Axis$_{n+1}$] 388
Object 68C0h: Interpolation Sub Mode Select [Axis$_{n+1}$] 388
Object 68C1h: Interpolation Data Record [Axis$_{n+1}$] 389
Object 68C2h: Interpolation Time Period [Axis$_{n+1}$] 390
Object 68F2h: Positioning Option Code 391
Object 68FDh: Digital Inputs [Axis$_{n+1}$] 392
Object 68FFh: Target Velocity [Axis$_{n+1}$] 392
Object 6D02h: Supported Drive Modes [Axis$_{n+1}$] 393
Object 6FFFh: Single Device Type [Axis$_{n+1}$] 393
5.3.3 Object dictionary access rights 394
5.4 NUMDrive C general parameters list 397
Do not use 402
Do not use 408
Do not use 408
5.5 NUMDrive C parameters description 409
5.5.1 Mono-Axis and Bi-Axes parameters description 409
V01 CAN configuration 409
V20 Sensor connectors X3, X4 configuration (Bi-Axes configuration only) 410
V21 TP1/2 output configuration (Bi-Axes configuration only) 411
V22 PROGO1/2 output configuration (Bi-Axes configuration only) 412
V23 Do not use 412
V30 General Drive Configuration 413
V30 Use of the bits 413
V32 Configuration Braking following alarms 414
V32 Use of configuration bit 1 - Braking modalities 415
V33 Alarm Reset Configuration 416
V33 Use of the bits 417
V34 Configuration DF1 / DF2 relay [1=break] 418
V34 Use of the bits 418
V35 SW function 419
V41 Motor zero speed threshold, enable, brake (Linear motor) 420
V42 Motor zero speed threshold, enable, brake 420
V43 Braking Timeout on configuration V30 - V32 parameters 420
V44 Braking Current on configuration V30 - V32 parameters 421
V50 Torque current limit [0] 422
V51 Proportional speed loop gain [0] 422
V52 Integral speed loop gain [0] 423
V53 Do not use 423
V57 Torque current limit - high range [0] 423
V58 Proportional speed loop gain - high range [0] 424

V59 Integral speed loop gain - high range [0] 424
V60 Proportional speed loop gain [0] 424
V75 Do not use 424
V80 Set Point Weight speed loop gain 425
V81 Anti-windup speed loop gain 425
V94 Speed reference Limit [% maximum speed] [0] 426
V95 Maximum motor speed [0] 426
V96 Threshold for overspeed alarm [0] 427
V97 Speed reference limit [% max. speed] [0] 427
V98 Maximum motor speed [0] 428
V99 Overspeed alarm threshold [0] 429
V108 Rated motor current [I^2t] 430
V108 (V115) Thermal image management 431
V109 Vertical load compensation current 432
V115 Rated motor current [I^2t] 432
V125 Current percentage V108 for EPSM 433
V128 Locked rotor current I^2t (% V108) 433
V129 Locked rotor current I^2t (% V115) 434
V130 Associate MP with TP1 (Analog / Digital) 434
V131 Associate MP with TP2 (Analog / Digital) 435
V132 Associate MP with TP3 (Digital) 435
V133 Associate MP with TP4 (Digital) 436
V134 Multiplication factor for TP1 436
V135 Multiplication factor for TP2 437
V136 Offset for analog TP1 test points 438
V137 Offset for analog TP2 test points 439
V138 Configuration of analog TP 439
V139 Configuration of Programmable Relays 440
V139 Use of the configuration bits 441
V139 Use of the Programmable relays 443
V140 Associate MP with programmable relay 1 448
V141 Programmable relay 1 threshold 1 449
V142 Programmable relay 1 threshold 2 449
V143 Programmable relay 1, MP scale factor 450
V144 Programmable relay 1 tripping time constant 451
V150 Associate MP with programmable relay 2 452
V151 Programmable relay 2, threshold 1 452
V152 Programmable relay 2, threshold 2 452
V153 Programmable relay 2, MP scale factor 452
V154 Programmable relay 2 tripping time constant 453
V160 Do not use 453
V165 Do not use 453
V166 Do not use 453
V169 Do not use 453
V170 Do not use 454
V171 Do not use 454
V172 Do not use 454
V173 Do not use 454
V175 Do not use 454
V176 Do not use 454
V180 Current percentage (V108) for EPS 455
V182 U/V Phase relative weight for EPS 455
V183 Phases V/W relative weight for EPS 456
V190 Do not use 456
V191 Do not use 456
V205 Do not use 456
V206 Do not use 456
V207 Do not use 457
V208 Do not use 457
V209 Do not use 457
V210 Do not use 457
V218 Do not use 457
V220 Reference configuration 458

V220 Use of the configuration bits....459
V221 Enable configurations....460
V225 Digital input configuration [GP_IN_A or B]....461
V258 Motor sensor type....462
V260 Speed regulator Parameters exchange....463
V261 Position regulator Parameters exchange....464
V266 Do not use....465
V268 Electrical phase offset Hall effect sensor....465
V269 Position offset for electrical phasing....465
V270 Motor PTC-NTC value....466
V272 Motor thermal sensor configuration....467
V273 Electrical phase offset management....467
V301 Proportional position gain [0]....468
V306 S1 sensor MULTI coefficient [0]....469
V307 S1 sensor DIVI coefficient [0]....469
V310 S1 sensor offset in Controller physical unit [0]....470
V311 S2 sensor MULTI coefficient [0]....473
V312 S2 sensor DIVI coefficient [0]....474
V313 S2 sensor offset in Controller physical unit [0]....474
V314 Sinusoidals signal range of S1 sensor....476
V315 Sinusoidals signal range of S2 sensor....477
V317 S1 sensor power supply voltage....478
V318 S2 sensor power supply voltage....478
V340 Step of speed calculation S1....479
V341 Step of speed calculation S2....480
V342 Speed filter by S1 sensor....481
V343 Speed filter by S2 sensor....481
V344 S1 Sensor code ID....482
V345 S2 Sensor code ID....482
V346 S1 Sensor number....483
V347 S2 Sensor number....483
V348 S1 Sensor configuration....484
V349 S2 Sensor configuration....488
V350 Position loop sensor configuration....492
V351 Speed loop sensor configuration....492
V352 Current loop sensor configuration....492
V353 Period of position sensor/motor turn [0]....493
V355 AP02: Mechanical ratio load turns....495
V356 AP02: Mechanical ratio sensor turns....495
V360 Period by turn sensor S1....496
V361 Absolute resolution by turn sensor S1....496
V365 Number of absolute sensor turns S1....497
V368 Intervals between coded references S1....497
V370 Number of absolute sensor turns S2....498
V371 Absolute resolution by turn sensor S2....498
V375 Number of absolute sensor turns S2....499
V378 Intervals between coded references S2....499
V380 Period of linear sensor S1....500
V381 Absolute resolution of linear sensor S1....500
V390 Period of linear sensor S2....501
V391 Absolute resolution of linear sensor S2....501
V500 Filter configuration....502
V500xx - V501 ÷ V549 Filter on torque and speed reference....504
V1050 Torque current limit [1]....508
V1051 Proportional speed loop gain [1]....509
V1052 Integral speed loop gain [1]....509
V1053 Do not use....510
V1057 Torque current limit - High range [1]....510
V1058 Proportional speed loop gain - High range [1]....511
V1059 Integral speed loop gain - High range [1]....511
V1060 Proportional speed loop gain [1]....512
V1075 Do not use....512
V1094 Speed reference Limit [% maximum speed] [1]....512

V1095 Maximum motor speed [1] 512
V1096 Threshold for overspeed alarm [1] 513
V1097 Speed reference limit [% max. speed] [1] 513
V1098 Maximum motor speed [1] 514
V1099 Overspeed alarm threshold [1] 515
V1301 Proportional position gain [1] 516
V1306 S1 sensor MULTI coefficient [1] 517
V1307 S1 sensor DIVI coefficient [1] 518
V1310 S1 sensor offset in Controller physical unit [1] 518
V1311 S2 sensor MULTI coefficient [1] 519
V1312 S2 sensor DIVI coefficient [1] 519
V1313 S2 sensor offset in Controller physical unit [1] 520
V1353 Period of position sensor/motor turn [1] 520

5.6 NUMDrive C test points with WPM 521
5.6.1 Drive test point 521
5.6.2 Test Points (TP) and Measure Points (MP) display 522
5.6.3 Measure points general list (MP) 523
5.6.4 Use of measure points: notes ( ) 527

5.7 NUMDrive C application - User example 528
5.7.1 Speed loop parameters optimization V51 / V52 528
5.7.1.1 Speed loop parameters optimization procedure 528
5.7.1.2 Speed loop parameters optimization procedure A 529
5.7.1.3 Speed loop parameters optimization procedure B 531
5.7.2 Programmable relais 1 or 2 application examples 533
5.7.2.1 Control of an axis in motion (using relay no. 1 - Output on PROGO1) 533
5.7.3 Motor brake management 534
5.7.3.1 Management of motor holding brake 534
5.7.3.2 Use of the Programmable relays in modality brake management 535
5.7.3.3 Management of motor holding brake in case of alarms 536
5.7.4 MULTI /DIVI parameters utilization V306, V307 537
5.7.4.1 MULTI/DIVI with S1 sensor type P, Q 537
5.7.4.2 MULTI/DIVI with type Q motor sensor for spindle motor with C axis application 538
5.7.5 Sensors configuration and applications 539
5.7.5.1 Enter new sensor configuration 539
5.7.5.2 Sinusoidal rotary encoder 541
5.7.5.3 Sinusoidal linear scale encoder 543
5.7.5.4 Linear sinusoidal encoder for linear motor 545
5.7.5.5 Rotary sinusoidal encoder for direct torque motor 547
5.7.6 Homing mode and homing setting 549
5.7.6.1 Homing mode information and functional description 549
5.7.6.2 General definitions and Homing Methods 550
5.7.7 Index Pulse identification (Pulse Mark) 554

5.8 Diagnostic signalling and software alarms 555
5.8.1 Alarm signalling 555
5.8.2 Alarm signalling 556
5.8.2.1 Led POWER ON status 556
5.8.3 Mono-Axis alarm signalling status 557
5.8.3.1 Mono-Axis alarm possible causes/remedy 557
5.8.4 Bi-Axes alarm signalling status 558
5.8.4.1 Bi-Axes alarm possible causes/remedy 558
5.8.5 Alarms configuration 559
5.8.5.1 General drive alarms / securities configuration 560
5.8.6 Alarms reset procedure 561
5.8.7 Alarms signalling 562
5.8.8 Alarms extension signalling 566
5.8.8.1 Extension for 02, 03, 07, 08 alarms 566
5.8.8.2 Extension for 09 alarm Stegmann encoder 567
5.8.8.3 Extension for 09 alarms EnDat encoder 567
5.8.8.4 Alarm 09 common extensions from 200 to 261 568
5.8.8.5 Extension for alarms from 10 to 15 570
5.8.8.6 Extension for alarm 16 571
5.8.8.7 Extension for alarm 18, 37, 38, 49 574

5.8.8.8 Extension for alarm 52.....575
5.8.8.9 Extension for alarm 53.....576
5.8.8.10 Extension for alarm 54, 55, 56, 90, 91 577
5.8.9 Meaning of the Word 0x1AF2 by WPM (PLC Status word) 578
5.8.10 Meaning of the Word 0x1A72 by WPM (Warning register) 581
5.8.11 Meaning of the Word 0x1AF1 by WPM (NC Status word) 582

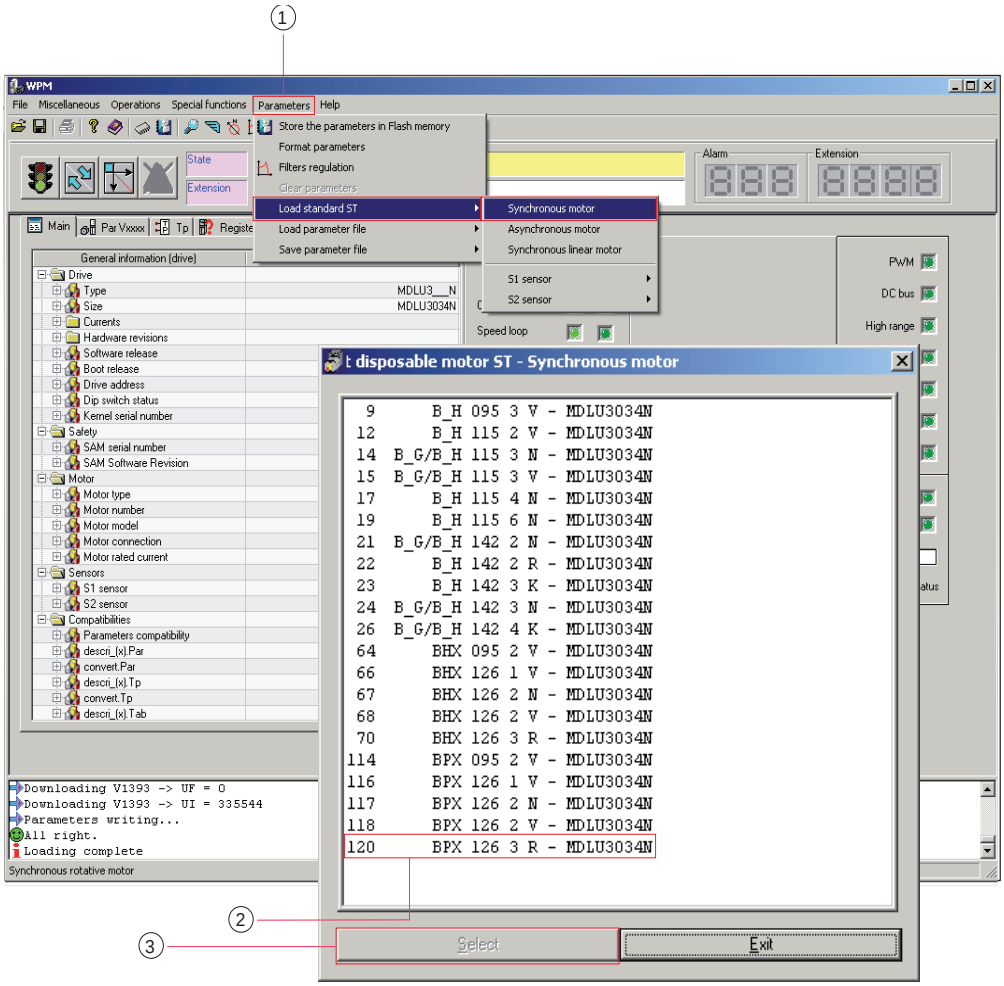
5 NUMDrive C with CANopen interface

5.1 NUMDrive C configuration with WPM

5.1.1 Motor / drive association and sensor configuration with WPM

5.1.1.1 Motor / drive selection

Open WPM tool and follow the here under instruction:

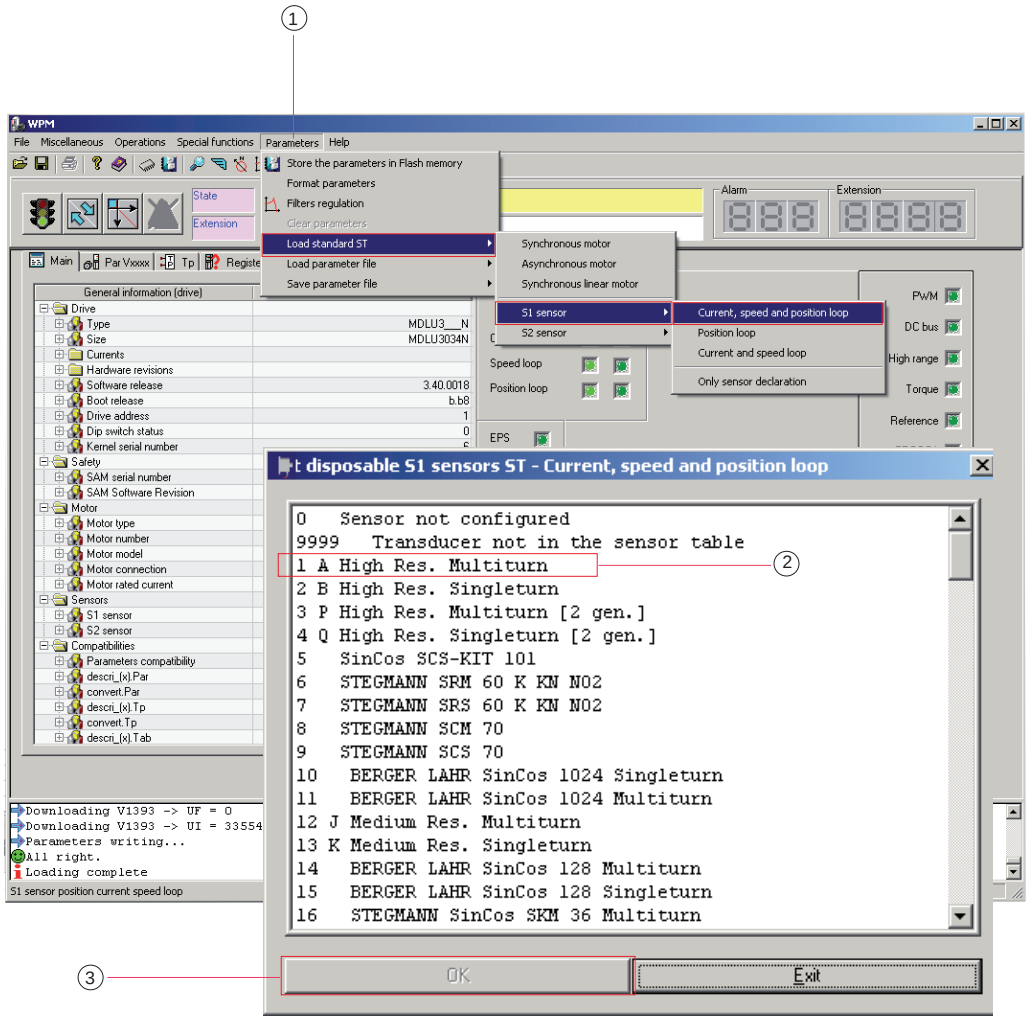


1	Select: → Parameters → Load standard ST → Synchronuos motor or → Asynchronous motor → Synchronuos linear motor
2	Select the motor / MDLU3 drive association
3	Than press → Select to enter the association

In case of motor / drive association not present on the general configuration table, contact NUM Technical Service, as an incorrect configuration may seriously damage the motor or drive.

For other detailed information about NUMDrive C configuration, i.e. for installing or setting a CANopen Mono-Axis or Bi-Axes device, please refer to the chapter 3 “Flexium PLC library”.

5.1.1.2 Sensor configuration (X1, X2, X3, X4)



1	Select: → Parameters → Load standard ST → S1 or S2 sensor → Select the loop declaration (Current, speed and position or Position loop or ...
2	Select the S1 sensor (A)
3	Than press → OK to enter the configuration

For more detailed information about the use of WPM tool, please refer to the WPM Instruction manual upon “Help” windows.

5.1.2 NUM sensor User's configuration

Due to a different NUMDrive C applications, the sensor configuration must always be carried out considering the different drive sensor typology. (See also V20 parameter)

Therefore, please give the appropriate drive / sensor configuration as indicated in the table below and give the NUM sensor selection as indicated in the next page.

S1, S2 configuration	NUMDrive C		
	Mono-Axis	Bi-Axes (B)	Bi-Axes (C)
Encoder S1	X1	X1, X2	X1, X3
Encoder S2	X2	-	X2, X4

5.1.2.1 NUM sensor table example

WPM gives automatically the sensor S1 or S2 configuration with NUM sensor selecting “Sensor configuration and application on chapter 5.7.5. Here below are indicated some NUM sensor.
Otherwise, if the sensor is not shown in the table (i.e. new one), see chapter 5.7.5.1 “ Enter new sensor configuration”.

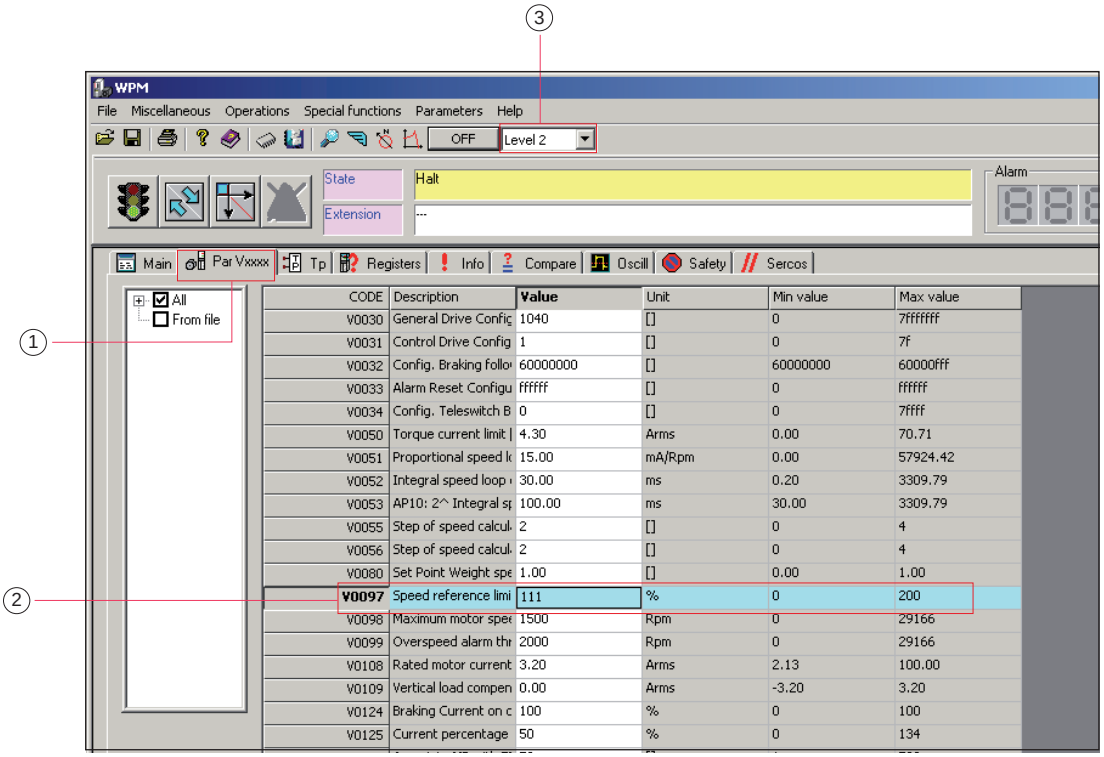
If the sensor is direct measure sensor, to avoid motor racing, may be necessary to modify the bit9 of V349 (Reversal measure for position loop).

Sensor N°	V258 value	S1, S2 (X1, X2, X3, X4) NUM sensor description
0		Not configurated sensor
9999		Not codified sensor
1 (A)	**A**	High resolution Multiturn encoder (absolute)
2 (B)	**B**	High resolution Singleturn encoder (incremental)
3 (P)	**P**	High resolution Multiturn encoder (absolute 2 gen.)
4 (Q)	**Q**	High resolution Singleturn encoder (incrementa 2 gen.)
5		SinCos SCS-KIT 101encoder
6		STEGMAN SRM 60 K KN N02 encoder
7		STEGMAN SRS 60 K KN N02 encoder
12	**J**	SinCos SKM 36 Multiturn encoder
13	**K**	SinCos SKS 36 Singleturn encoder
204		Heidenhain LS186C
205		Heidenhain ROD 280C-18000
300		Heidenhain LC181 ENDAT
301		Heidenhain ROQ 425-2048 ENDAT
302		Heidenhain ROQ 413-512 ENDAT
303		Heidenhain LC481 ENDAT
304		Heidenhain RNC 723 ENDAT
305		Heidenhain ENC 113 ENDAT
306		Heidenhain RNC 220 ENDAT
307		Heidenhain ROC 415 ENDAT
308		Heidenhain ENC 413-512 ENDAT
309		Heidenhain ENC 1313-512 ENDAT
310		Heidenhain ENC 413-2048 ENDAT
311		Heidenhain ENC 1313-2048 ENDAT

5.1.3 Drive parameters display

The parameter editor gives a rapid overview of the drive parameters.
By means the editor is possible:

- Setting the MDLU3 drive parameters
- Display and modify all drive parameters as for LEVEL 1, 2 of modification only.
  Parameters gives on superior levels are reserved to the service NUM and are indicated in the tables on chapeter 5.4 as “Reserved Parameters”.
- Click on “Par Vxxxx “ icon to gain the parameters pages.

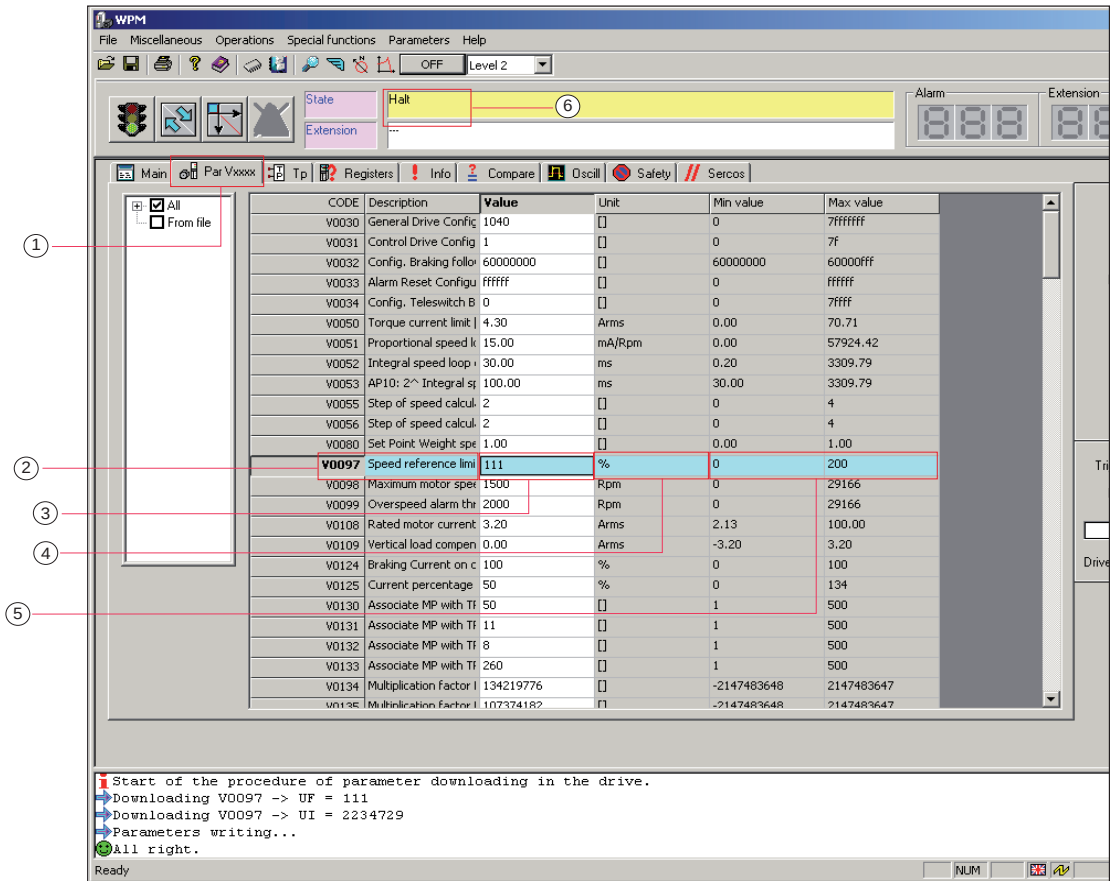


1	Parameters display icon
2	Drive parameter display
3	Parameters level selection

The drive parameters illustrated on next pages, are typically parameters that must be adjusted during the standard drive installation.
The highlight parameters are superior level parameters, such parameters are reserved to the SERVICE NUM. Therefore, they not require a standard adjustment. Please refer to the chapter 1 “How to read a parameters list and parameter description”.

For more detailed information about the use of WPM tool, please refer to the WPM Instruction manual upon “Help” windows.

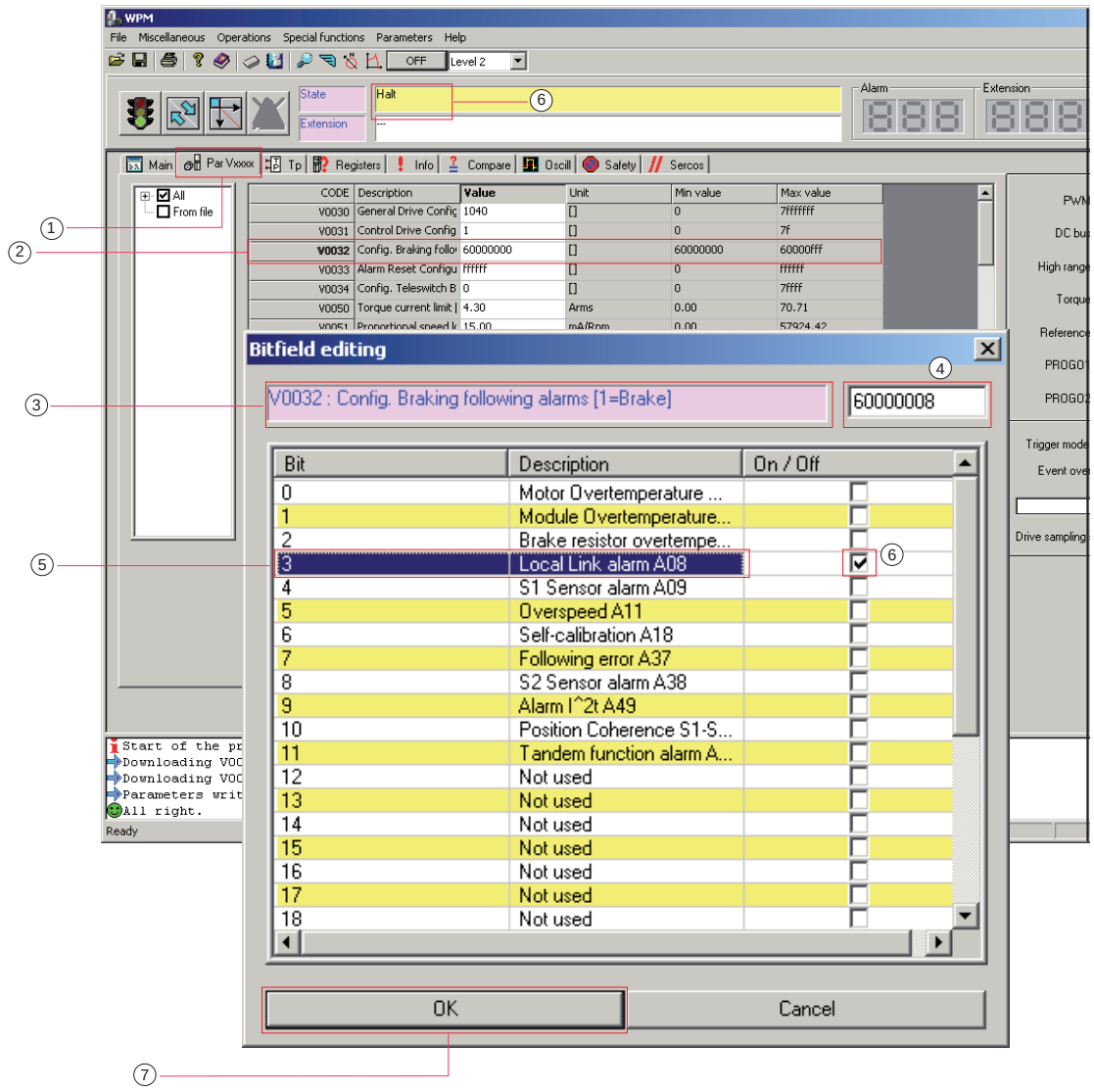
5.1.3.1 Parameters selection and modification



1	Parameters display icon
2	Drive parameter display
3	Parameters value. Enter the New modification value than press ENTER.
4	Parameters unit
5	Parameter adjusting scale
6	Drive state: Halt or Run

For more detailed information about the use of WPM tool, please refer to the WPM Instruction manual upon “Help” windows.

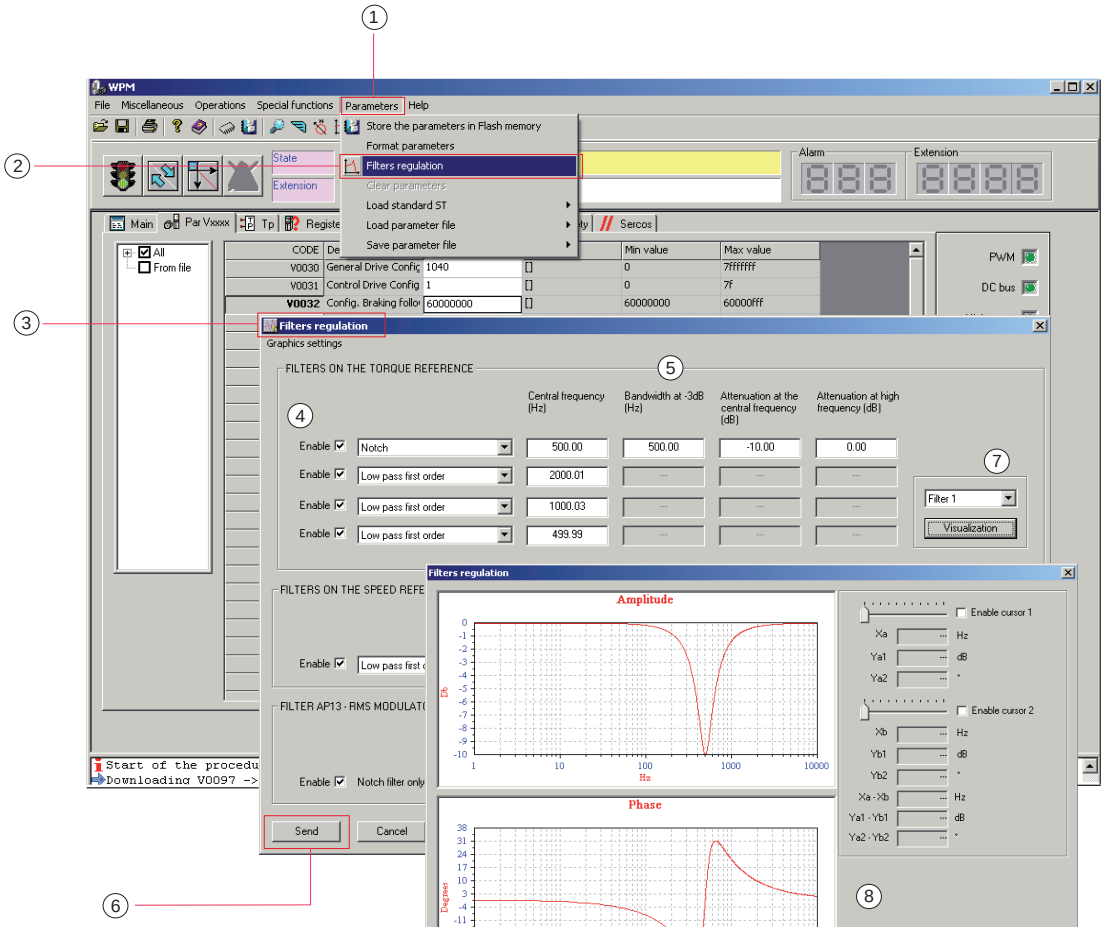
5.1.3.2 Bit parameters selection and modification



1	Parameters display icon
2	Double click on “Value” area to open the Bitfiled editing page
3	Parameter description
4	Parameter value
5	Bit modification value
6	Bit modification “On” state
7	Than press → OK to enter the modification

For more detailed information about the use of WPM tool, please refer to the WPM Instruction manual upon “Help” windows.

5.1.3.3 Drive filter selection and modification



1	Click on the icon to open the “Filters regulation” page.
2	Click on the icon “Filters” to open the filter setupchapter
3	Filters regulation” page
4	Filter selection
5	Filter type and values modification
6	Click on Send to enter the filters setup
7	Click on “Visualization” to make to filter regulation visualization
8	“Filters regulation” page

For more detailed information about the use of WPM tool, please refer to the WPM Instruction manual upon “Help” windows.

5.2 NUMDrive C Mono-Axis objects description

5.2.1 NC Units

NUMDrive C CANopen Mono-Axes uses “NC units” to measure position, speed, acceleration.
One “NC unit” is the minimum increment the user want to appreciate (beware, it is not an indication of the maximum resolution achievable with the system: is an expression of the desired resolution).

The “Multi/Divi” selection is done to have all the measure-related objects expressed in this unit.

With an example the things are simpler:
- the axis is a linear one, with a rotative motor and a screw;
- the measure should be expressed in micrometers: this means, for example, object 6064h will be directly “micron”;
- the sensor is a 1024 pulse encoder;
- there is a 1:1 gearbox;
- the screw is 10mm/turn;
- this means the measure will increment by 10.000 microns every turn;
- the suggested Multi/Divi is 10000/4096
  (please refer to the chapter 5.5.1 Parameters description).

5.2.2 Objects description

Object 1000h: Device Type

The device shows the implemented device profile and the device type.
The lower 16 bit contain the device profile number and the upper 16 bit an additional information.

Object description	
INDEX	1000
Name	Device Type
Object Code	Variable
Data Type	UNSIGNED32
Category	Mandatory

Entry description	
Access	CONST
PDO Mapping	No
Default Value	0x0FFF0192
Lower Limit	0x00000000
Upper Limit	0xFFFFFFFF
Unit	-

Object 1001h: Error Register

The object shows the error status of the device.
The error register is an 8-bit field, and the format is fixed by the CANopen specification.

The detailed description of the fault can be found with the object predefined error field (1003h) and with the object error code (603Fh).
Errors are already signalled by an EMCY message as soon as they occur.

Bit Meaning

- 0 generic error
- 1 reserved
- 2 reserved
- 3 reserved
- 4 communication error (overrun, error state)
- 5 device profile specific
- 6 reserved
- 7 manufacturer specific.

Object description	
INDEX	1001
Name	Error Register
Object Code	Variable
Data Type	UNSIGNED8
Category	Mandatory

Entry description	
Access	RO
PDO Mapping	No
Default Value	0x00
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Object 1003h: Pre-defined Error Field

The object saves the latest error messages that were shown as EMCY messages.

- The entry under subindex 00h contains the number of saved error messages.
- The current error message is stored under subindex 01h, older messages are moved to high subindex entries.
- Writing 0 to subindex 00h resets the error list.

Object description	
INDEX	1003
Name	Pre-defined Error Field
Object Code	Array
Data Type	UNSIGNED32
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Errors
Entry Category	Mandatory
Access	RW
PDO Mapping	No
Default Value	0x1
Lower Limit	0x00000000
Upper Limit	0x000000FE
Unit	-

Sub-Index	001
Description	Standard Error Field
Entry Category	Mandatory
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x00000000
Upper Limit	0xFFFFFFFF
Unit	-

Please refer to the chapter 2.12.1 for more detailed informations.

Object 1005h: COB-ID SYNC

The object shows the COB-ID of the SYNC object and specifies whether a device sends or receives SYNC messages.

NUMDC-CAN drive can only receive SYNC messages.
For synchronisation a device in the network must send SYNC objects.

The device generate a SYNC message if bit 30 is set.
The COB-ID can be changed in the NMT “Pre-Operational” status only.

Object description	
INDEX	1005
Name	COB-ID SYNC
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x80000080
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Object 1006h: Communication Cycle Period

This entry defines the communication cycle period in us.
It is 0, if not used.

Object description	
INDEX	1006
Name	Communication Cycle Period
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x00000000
Upper Limit	0xFFFFFFFF
Unit	us

Object 1012h: COB-ID Time Stamp

COB-ID of the Time Stamp Object (TIME).
If bit 31 is set the device consumes the Time Stamp message and if bit 30 is set the device produces the Time Stamp message.
The meaning of the other bits is equal to the other communication objects.

Object description	
INDEX	1012
Name	COB-ID Time Stamp
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x80000100
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Object 100Ch: Guard Time

This entry contains the guard time in milli-seconds.
It is 0, if not used.
It shows the time span for connection monitoring (node guarding) of an NMT slave.

The time span for the connection monitoring of a NMT master is derived from the time span “guard time” multiplied by the “life time factor”, object Life time factor(100Dh) .

The time span can be changed in the NMT “Pre-Operational” status only.

Object description	
INDEX	100C
Name	Guard Time
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	ms

Object 100Dh: Life Time Factor

The object shows the factor that together with the time span “guard time” gives the time interval for the connection monitoring of a NMT master.

Within this period the NMT slave device waits for a monitoring request by node guarding by the NMT master device. Life time = (guard time * life time factor).

The value “0” disables the monitoring of the NMT master.
If the connection monitoring by the NMT master remains disabled during the time interval “life time”, the device reports an error and switches to error status.

The time factor can be changed in the NMT “Pre-Operational” status.The time span “guard time” is set with the object Guard time (100Ch)

Object description	
INDEX	100D
Name	Life Time Factor
Object Code	Variable
Data Type	UNSIGNED8
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFF
Unit	-

Object 1014h: COB-ID EMCY

COB-ID used for emergency message (Emergency Server).

Object description	
INDEX	1014
Name	COB-ID EMCY
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RO
PDO Mapping	No
Default Value	0x00000080
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Object 1015h: Inhibit Time Emergency

The object specifies the waiting (inhibit) period for the repeated transmission of EMCY messages as a multiple of 100µs.

Object description	
INDEX	1015
Name	Inhibit Time Emergency
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	-

Object 1016h: Heartbeat Consumer Entries

The object saves the settings of the “heartbeat” consumer for NMT monitoring by “heartbeat” connection message. The consumer heartbeat time defines the expected heartbeat cycle time and thus has to be higher than the corresponding producer heartbeat time configured on the device producing this heartbeat.

The node number (Node-Id) and the monitoring time (Heartbeat time) of network users are established by means of the object which is being mo-nitored by the drive control over the Heartbeat protocol.

- The bits 31 - 24 of the sub-index 01 should be 0.
- The bits 23 - 16 contain the node-id.
- The lower 16 bits contain the heartbeat time.

Sub-index 01h
HeartbeatMonitoring

A device sends its status cyclically by means of theHeartbeat protocol; see object Producer heartbeat time (1017h). The “heart beat” of selected devices can be monitored by the drive control. If the heart beat is not received then an event is triggered (Abort connection).
The reaction of the drive to such an event can be defined with the object Abort connection option code (6007h/687Fh).

Object description	
INDEX	1016
Name	Heartbeat Consumer Entries
Object Code	Array
Data Type	UNSIGNED32
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x01
Lower Limit	0x1
Upper Limit	0x7F
Unit	-

Sub-Index	001
Description	Consumer Heartbeat Time 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0x02FFFFFF
Unit	-

Object 1017h: Producer Heartbeat Time

This object is the companion of 1016h.
The producer heartbeat time defines the cycle time of the heartbeat.
If the time is 0 it is not used.
The time has to be a multiple of 1 msec.

Object description	
INDEX	1017
Name	Producer Heartbeat Time
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	-

Object 1018h: Identity Object

This object contains general information about the device.

- Sub-Index 1 contains a unique value allocated each manufacturer.
- Sub-Index 2 identifies the manufacturer specific product code (device version).
- Sub-Index 3 contains the revision number.
- Bit 31-16 is the major revision number and Bit 15-0 the minor revision number.
- Sub-Index 4 identifies a manufacturer specific serial number.

Object description	
INDEX	1018
Name	Identity Object
Object Code	Record
Data Type	IDENTITY
Category	Mandatory

Entry description	
Sub-Index	000
Description	number of entries
Entry Category	Mandatory
Access	RO
PDO Mapping	No
Default Value	0x4
Lower Limit	0x1
Upper Limit	0x4
Unit	-

Sub-Index	001
Description	Vendor Id
Entry Category	Mandatory
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Product Code
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	003
Description	Revision number
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO

PDO Mapping	No
Default Value	0x01
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	004
Description	Serial number
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Object 1020h: Verify Configuration

This object contains date and time of the last configuration.

- Sub index 001 shall contain the number of days after January 1, 1984.
- Sub index 002 shall contain the number of seconds after midnight.

Object description	
INDEX	1020
Name	Verify Configuration
Object Code	Array
Data Type	UNSIGNED32
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x2
Lower Limit	0x02
Upper Limit	0x02
Unit	-

Sub-Index	001
Description	Configuration date
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Configuration time
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Object 1200h: Server SDO Parameter 1

The object contains the parameters for the SDOs for which the device is the server.
It is here for information only: the default settings shouldn't be changed.

Object description	
INDEX	1200
Name	Server SDO Parameter 1
Object Code	Record
Data Type	SDO_PARAMETER
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x02
Lower Limit	0x02
Upper Limit	0x02
Unit	-

Sub-Index	001
Description	COB-ID Client -> Server
Entry Category	Mandatory
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x00000600
Lower Limit	0x00000600
Upper Limit	0xBFFFFFFF
Unit	-

Sub-Index	002
Description	COB-ID Server -> Client
Entry Category	Mandatory
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x00000580
Lower Limit	0x00000580
Upper Limit	0xBFFFFFFF
Unit	-

Object 1201h: Server SDO Parameter 2

The object contains the parameters for the SDOs for which the device is the server.
It is here for information only: the default settings shouldn't be changed.

Object description	
INDEX	1201
Name	Server SDO Parameter 2
Object Code	Record
Data Type	SDO_PARAMETER
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x03
Lower Limit	0x02
Upper Limit	0x03
Unit	-

Sub-Index	001
Description	COB-ID Client -> Server
Entry Category	Mandatory
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x800006E0
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	COB-ID Server -> Client
Entry Category	Mandatory
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x800006E0
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	003
Description	Node ID of the SDO Client
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x00
Upper Limit	0x7F
Unit	-

Object 1400h: Receive PDO Communication Parameter 1

The object saves the settings for the first receive PDO (RPDO1). It is composed of subindex 0..3.

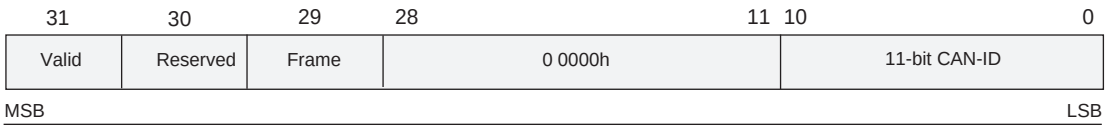
Object description	
INDEX	1400
Name	Receive PDO Communication Parameter 1
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

- **Sub index 0: number of entries, ReadOnly**

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x03
Lower Limit	0x02
Upper Limit	0x05
Unit	-

- **Sub index 1: enable, COB-ID setting**

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000200
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-



Structure of RPDO COB-ID

Bit 31: RPDO is enabled if bit 31="0".
The COB-ID of the object can be changed in the NMT "Pre-Operational" status only.

- **Sub index 2: transmission type**

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFF
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

The transmission type define when the received value is evaluated.

- Transmission type = 0 (acyclic transmission): the evaluation is linked to the SYNC object, but not the transmission of the PDO. A received PDO message is evaluated with the following SYNC.
- Transmission type = 1..252 (cyclic transmission): a value between 1 and 240 shows the number of SYNC cycles after which a received PDO is evaluated.
- Transmission types = 241..251 are reserved
- Transmission type = 252 is RTR-only, synchronous mode.
- Transmission type = 253 is RTR-only, asynchronous mode.
- Transmission type = 254..255 is asynchronous, the received value is used immediately when received.

- **Sub index 3: inhibit time**

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

The inhibit time is the last filter before RPDO evaluationt. It is specified in 100-uS units

Object 1401h: Receive PDO Communication Parameter 2

The object saves the settings for RPDO2.
Please refer to the description of the object 1400h for more detailed information.

Object description	
INDEX	1401
Name	Receive PDO Communication Parameter 2
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x03
Lower Limit	0x02
Upper Limit	0x05
Unit	-

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x80000300
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFF
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

Object 1402h: Receive PDO Communication Parameter 3

The object saves the settings for RPDO3.
Please refer to the description of the object 1400h for more detailed information.

Object description	
INDEX	1402
Name	Receive PDO Communication Parameter 3
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x03
Lower Limit	0x02
Upper Limit	0x05
Unit	-

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x80000400
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFF
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

Object 1403h: Receive PDO Communication Parameter 4

The object saves the settings for RPDO4.
Please refer to the description of the object 1400h for more detailed information.

Object description	
INDEX	1403
Name	Receive PDO Communication Parameter 4
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x03
Lower Limit	0x02
Upper Limit	0x05
Unit	-

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x80000500
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFE
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

Object 1600h: Receive PDO Mapping Parameter 1

The object saves the settings for the mapping of the 1st receive PDO (RPDO1), and shows which objects are mapped in RPDO1 and received with the PDO itself.
The PDO mapping for RPDO1 cannot be modified.

It is composed of subindex 0..1.

Object description	
INDEX	1600
Name	Receive PDO Mapping Parameter 1
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

- **Sub index 0: number of entries, Read-only**

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RO
PDO Mapping	No
Default Value	0x01
Lower Limit	0x0
Upper Limit	0x40
Unit	-

- **Sub index 1: 1st mapping entry (fixed)**

Entry description	
Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x60400010
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

PDO mapping of the object “controlword” (6040h).

Object 1601h: Receive PDO Mapping Parameter 2

The object saves the settings for the mapping of the 2nd receive PDO (RPDO2), and shows which objects are mapped in RPDO2 and received with the PDO itself.

The PDO mapping for RPDO2 cannot be modified. The default assignment is preset and useful for the “profile position mode”.
It is composed of subindex 0..2.

Object description	
INDEX	1601
Name	Receive PDO Mapping Parameter 2
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

- **Sub index 0: number of entries, Read-only**

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RO
PDO Mapping	No
Default Value	0x02
Lower Limit	0x0
Upper Limit	0xFF
Unit	-

- **Sub index 1: 1st mapping entry (fixed)**

Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x60400010
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

PDO mapping of the object “controlword” (6040h).

- **Sub index 2: 2nd mapping entry (fixed)**

Sub-Index	002
Description	Mapping Entry 2
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x607A0020
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

PDO mapping of the object “target position” (607Ah).

Object 1602h: Receive PDO Mapping Parameter 3

The object saves the settings for the mapping of the 3rd receive PDO (RPDO3), and shows which objects are mapped in RPDO3 and received with the PDO itself.

The PDO mapping for RPDO3 cannot be modified. The default assignment is preset for speed mode in the “profile velocity mode”. It is composed of subindex 0..2.

Object description	
INDEX	1602
Name	Receive PDO Mapping Parameter 3
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

- **Sub index 0: number of entries, Read-only**

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RO
PDO Mapping	No
Default Value	0x02
Lower Limit	0x0
Upper Limit	0xFF
Unit	-

- **Sub index 1: 1st mapping entry (fixed)**

Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x60400010
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

PDO mapping of the object “controlword” (6040h).

- **Sub index 2: 2nd mapping entry (fixed)**

Sub-Index	002
Description	Mapping Entry 2
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x60FF0020
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

PDO mapping of the object “target velocity” (60FFh).

Object 1603h: Receive PDO Mapping Parameter 4

The object saves the settings for the mapping of the 4th receive PDO (RPDO4), and shows how objects can be mapped in RPDO4 and received with the PDO itself.

The PDO mapping for RPDO4 can be modified (and should be written before the use).
RPDO4 is for expert users only. It is composed of subindex 0..2.

Object description	
INDEX	1603
Name	Receive PDO Mapping Parameter 4
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

- **Sub index 0: number of entries, Read-only**

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFF
Unit	-

- **Sub index 1..4: mapping entries (fixed)**

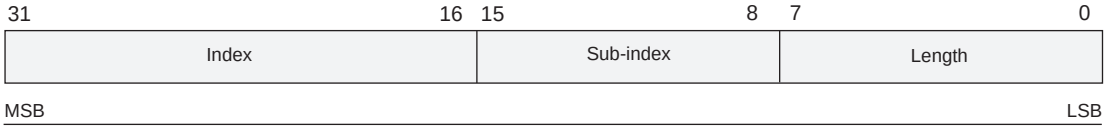
Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFFFFF
Unit	-

Sub-Index	002
Description	Mapping Entry 2
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFFFFF
Unit	-

Sub-Index	003
Description	Mapping Entry 3
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFFFFF
Unit	-

Sub-Index	004
Description	Mapping Entry 4
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFFFFF
Unit	-

Here should be written the selection of objects to map, and the corresponding length.
The format is…



(Where “length” is the bit size of the mapped object: for example, 0x10 for a word like “statusword 6041h”)

The sequence to configure a new mapping for RPDO4 is:

- Deactivate the mapping by writing 0 to subindex 0;
- Select the desired objects and size writing to subindex 1 to 4 (stop when ready);
- Activate the mapping by writing to subindex 0 the number of selected objects.

Object 1800h: Transmit PDO Communication Parameter 1

The object saves the settings for the first transmit PDO (TPDO1). It is composed of subindex 0..5.

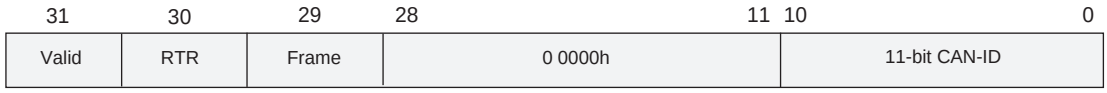
Object description	
INDEX	1800
Name	Transmit PDO Communication Parameter 1
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

- **Sub index 0: number of entries, Read-only**

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x05
Lower Limit	0x02
Upper Limit	0x06
Unit	-

- **Sub index 1: enable, COB-ID setting**

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000180
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-



Structure of TPDO COB-ID

- Bit 31: TPDO is enabled if bit 31=”0”.
- Bit 30: RTR bit. If a device supports TPDOs with RTR (remote transmission request), it can request a PDO from a PDO producer with RTR = “0” in accordance with the producer consumer relationship.

Settings

TPDO1 is sent asynchronously and/or event-controlled at every change of the PDO data.
The byte assignment of the TPDO1 is specified via PDO mapping with the object 1st transmit PDO mapping (1A00h).

The following assignment is the default:
- Bytes 0..1: statusword (6041h).

The COB-ID of the object can be changed in the NMT “Pre-Operational” status.

- **Sub index 2: transmission type**

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFF
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

The transmission type define when the received value is evaluated.

- Transmission type = 0 (acyclic transmission): the evaluation is linked to the SYNC object, but not the transmission of the PDO. A received PDO message is evaluated with the following SYNC.
- Transmission type = 1..252 (cyclic transmission): a value between 1 and 240 shows the number of SYNC cycles after which a received PDO is evaluated.
- Transmission types = 241..251 are reserved
- Transmission type = 252 is RTR-only, synchronous mode.
- Transmission type = 253 is RTR-only, asynchronous mode.
- Transmission type = 254..255 is asynchronous, the received value is used immediately when received.

- **Sub index 3: inhibit time**

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

The inhibit time is the last filter before TPDO transmission, defining the maximum rate for message generation. It is specified in 100-uS units.

- **Sub index 4: reserved**
  Do not modify.

- **Sub index 5: event timer**

Sub-Index	005
Description	Event Timer
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	ms

If the value is different from 0, a transmission is triggered every “event timer” ms. This trigger source add to other sources (if, for example, the transmission type is 255).

Object 1801h: Transmit PDO Communication Parameter 2

The object saves the settings for TPDO2.
Please refer to the description of the object 1800h for more detailed information.

Object description	
INDEX	1801
Name	Transmit PDO Communication Parameter 2
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x05
Lower Limit	0x02
Upper Limit	0x06
Unit	-

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000280
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFF
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

Sub-Index	004
Description	Compatibility Entry
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	005
Description	Event Timer
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	ms

Object 1802h: Transmit PDO Communication Parameter 3

The object saves the settings for TPDO3.
Please refer to the description of the object 1800h for more detailed information.

Object description	
INDEX	1802
Name	Transmit PDO Communication Parameter 3
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x05
Lower Limit	0x02
Upper Limit	0x06
Unit	-

Entry description	
Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000380
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Entry description	
Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFF
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Entry description	
Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

Entry description	
Sub-Index	004
Description	Compatibility Entry
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Entry description	
Sub-Index	005
Description	Event Timer
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	ms

Object 1803h: Transmit PDO Communication Parameter 4

The object saves the settings for TPDO4.
Please refer to the description of the object 1800h for more detailed information.

Object description	
INDEX	1803
Name	Transmit PDO Communication Parameter 4
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x05
Lower Limit	0x02
Upper Limit	0x06
Unit	-

Entry description	
Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000480
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFE
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

Sub-Index	004
Description	Compatibility Entry
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	005
Description	Event Timer
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	ms

Object 1A00h: Transmit PDO Mapping Parameter 1

The object saves the settings for the mapping of the 1st transmit PDO (TPDO1), and shows which objects are mapped in TPDO1 and transmitted with the PDO itself.
The PDO mapping for TPDO1 cannot be modified.

It is composed of subindex 0..1.

Object description	
INDEX	1A00
Name	Transmit PDO Mapping Parameter 1
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

- **Sub index 0: number of entries, Read-only**

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RO
PDO Mapping	No
Default Value	0x01
Lower Limit	0x0
Upper Limit	0xFF
Unit	-

- **Sub index 1: 1st mapping entry (fixed)**

Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x60410010
Lower Limit	0x00000000
Upper Limit	0xFFFFFFFF
Unit	-

PDO mapping of the object “statusword” (6041h).

Object 1A01h: Transmit PDO Mapping Parameter 2

The object saves the settings for the mapping of the 2nd transmit PDO (TPDO2), and shows which objects are mapped in TPDO2 and transmitted with the PDO itself.
The PDO mapping for TPDO2 cannot be modified. The default assignment is preset and useful for the “profile position mode”.

It is composed of subindex 0..2.

Object description	
INDEX	1A01
Name	Transmit PDO Mapping Parameter 2
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

- **Sub index 0: number of entries, Read-only**

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RO
PDO Mapping	No
Default Value	0x02
Lower Limit	0x0
Upper Limit	0xFF
Unit	-

- **Sub index 1: 1st mapping entry (fixed)**

Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x60410010
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

PDO mapping of the object “statusword” (6041h).

- **Sub index 2: 2nd mapping entry (fixed)**

Sub-Index	002
Description	Mapping Entry 2
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x60640020
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

PDO mapping of the object “position actual value in user units” (6064h).

Object 1A02h: Transmit PDO Mapping Parameter 3

The object saves the settings for the mapping of the 3rd transmit PDO (TPDO3), and shows which objects are mapped in TPDO3 and transmitted with the PDO itself.
The PDO mapping for TPDO3 cannot be modified. The default assignment is preset for speed mode in the “profile velocity mode”. It is composed of subindex 0..2.

Object description	
INDEX	1A02
Name	Transmit PDO Mapping Parameter 3
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

- **Sub index 0: number of entries, Read-only**

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RO
PDO Mapping	No
Default Value	0x02
Lower Limit	0x0
Upper Limit	0xFF
Unit	-
Unit	-

- **Sub index 1: 1st mapping entry (fixed)**

Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x60410010
Lower Limit	0x0
Upper Limit	0xFFFFFFF
Unit	-

PDO mapping of the object “statusword” (6041h).

- **Sub index 2: 2nd mapping entry (fixed)**

Sub-Index	002
Description	Mapping Entry 2
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x606C0020
Lower Limit	0x0
Upper Limit	0xFFFFFFF
Unit	-

PDO mapping of the object “velocity actual value” (606Ch).

Object 1A03h: Transmit PDO Mapping Parameter 4

The object saves the settings for the mapping of the 4th transmit PDO (TPDO4), and shows how objects can be mapped in TPDO4 and transmitted with the PDO itself.
The PDO mapping for TPDO4 can be modified (and should be written before the use). TPDO4 is for expert users only. It is composed of subindex 0..2

Object description	
INDEX	1A03
Name	Transmit PDO Mapping Parameter 4
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

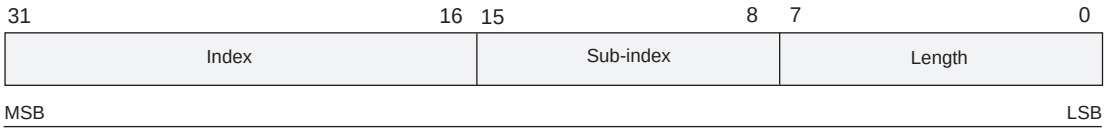
- **Sub index 0: number of entries, Read-only**

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFF
Unit	-

- **Sub index 1..4: mapping entries (fixed)**

Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x00
Upper Limit	0xFFFFFFFF
Unit	-

Here should be written the selection of objects to map, and the corresponding length.
The format is…



(Where “length” is the bit size of the mapped object: for example, 0x10 for a word like “controlword 6040h”)

The sequence to configure a new mapping for TPDO4 is:
- Deactivate the mapping by writing 0 to subindex 0;
- Select the desired objects and size writing to subindex 1 to 4 (stop when ready);

Activate the mapping by writing to subindex 0 the number of selected objects.

Sub-Index	002
Description	Mapping Entry 2
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	003
Description	Mapping Entry 3
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	004
Description	Mapping Entry 4
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Object 3000h: Drive internal status flags

This manufacturer-specific object can be used to read some internal device flags, used to monitor the drive operation.

Bit code object

Bit	Description		Word by WPM
0	MOTOR MOVING	1 = Motor in motion	0x1AF2
		0 = Motor in motion	
1	MEASURE READY	1 = Ready	0x1AF1
		0 = Not ready	
2	CONSOL WPM ACTIVE	1 = Escluded	0x1AF2
		0 = Active	
3	PWM CUT OFF	1 = PWM enable	0x1AF2
		0 = PWM disable	
4	PROGO1		0x1AF2
5	PROGO2		0x1AF2
6	ELECTRICAL GEAR	1 = High gear	0x1AF2
		0 = Low gear	
7	I^2 t INTERVENTION		0x1A72
8	ERROR UGA SENSOR 1		0x1A72
9	ERROR UGA SENSOR 2		0x1A72
10	FOLLOWING ERROR IU OVERFLOW		0x1A72
11	S1 ENDAT 2.2 CRC ERROR		0x1A72
12	S1 ENDAT 2.2 CRC ERROR		0x1A72
13-15	Do not used.		

Object description	
INDEX	3000
Name	Drive internal status flags
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RO
PDO Mapping	Yes
Default Value	0x00
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	-

Object 3001h: TPDO trigger transmission window

This is an absolute value (not related to any physical unit) defining a threshold window to transmit the TPDO object, when programmed with transmission type = 255.

This behaviour apply to TPDO2-66 (composed by statusword+position) and TPDO3-67 (composed by statusword+speed); the window is applied to the physical value (position or speed, respective) and, of course, not to statusword changes.

Using this object (and the corresponding 3101h for the 2nd axis of the biaxis module) it is possible to define an acceptable “ripple” on the selected measure (position for TPDO2 and speed for TPDO3) that will not fire the PDO transmission: thus having a silent behaviour when the axis is not moving.

The rule used to generate the TPDO transmission (type 255) is:

- abs (New value - Old value) >= TPDO trigger transmission window : Transmission O.K
- abs (New value - Old value) < TPDO trigger transmission window : NO Transmission.

Object description	
INDEX	3001
Name	TPDO trigger transmission window
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x00000000
Upper Limit	0xFFFFFFFF
Unit	-

Object 3002h: RPDO1 shadow before SYNC validation

If the RPDO1 is declared as “type 0” (= acyclic, synchronized) this object can be used to read back the “future” value of the command-word, sent but not yet updated by the SYNC.

If the RPDO1 is not of “synchronized” types, i.e. 0xFE or 0xFF, this is simply a copy of the commandword.

Object description	
INDEX	3002
Name	RPDO1 shadow before SYNC validation
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RO
PDO Mapping	No
Default Value	0x00
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	-

Object 6007h: Abort Connection Option Code

The object Abort connection option code is used to set the reaction of the drive to a loss of connection to NMT-Master (Life-Guarding Event or Heartbeat error).

The NUMDrive C CANopen drive use a fixed behaviour to the NMT error, generating the A91 alarm. Please refer to the chapter 5.8 for more detailed information upon such alarm.

The reaction to this alarm is configurable using parameters V32, V33, V34.
Please refer to the chapter 5.5.1 for more detailed information upon such parameters.

Object description	
INDEX	6007
Name	Abort Connection Option Code
Object Code	Variable
Data Type	INTEGER16
Category	Optional

Entry description	
Access	RO
PDO Mapping	No
Default Value	0x1
Lower Limit	0x1
Upper Limit	0x1
Unit	-

Object 603Fh: Error Code

This object captures the code of the last error that occured in the drive.
The value repre-sents the lower 16 bit in object Pre-defined error field (1003h).

Object description	
INDEX	603F
Name	Error Code
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RO
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	-

Object 6040h: Controlword

The drive will to control by this object. This object operates the state machine and the modes.
Please refer to the chapter 5.8 for more detailed description upon the operation-specific behaviour of controlword.

- Bit 0 = Switch on
- Bit 1 = Enable voltage
- Bit 2 = Quick stop
- Bit 3 = Enable operation
- Bit 4-6 = Operation mode specific. (1)
- Bit 7 = Fault reset
- Bit 8 = Halt
- Bit 9, 10 = Reserved
- Bit 11-15 = Manufacturer specific. (1)

(1) For more detailed information please refer to chapter 4.4.2, 4.4.3 and 4.4.4.
.

Object description	
INDEX	6040
Name	Controlword
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RW
PDO Mapping	Yes
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	-

Object 6041h: Statusword

This object indicates the current state of the drive.
Please refer to the chapter 4.3 for more detailed description upon the operation-specific behaviour of statusword:

- Bit 0 = Ready to switch on
- Bit 1 = Switched on
- Bit 2 = Operation enabled
- Bit 3 = Fault
- Bit 4 = Voltage enabled
- Bit 5 = Quick stop
- Bit 6 = Switch on disabled
- Bit 7 = Warning
- Bit 8 = Manufacturer specific
- Bit 9 = Remote
- Bit 10 = Target reached
- Bit 11 = Internal limit active
- Bit 12-13 = Operation mode specific. (1)
- Bit 14-15 = Manufacturer specific. (1)

(1) For more detailed information please refer to chapter 4.4.2, 4.4.3 and 4.4.4.

Object description	
INDEX	6041
Name	Statusword
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RO
PDO Mapping	Yes
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	-

Object 605Ah: Quick Stop Option Code

The object is used to set the reaction of the drive to a “Quick Stop” command (interruption of movement).
The state machine of the drive control branches due to this command into the status “Quick Stop active”, see object Controlword (6040h).

NUMDrive C CANopen only foreseen the reaction “Slow down on quick stop ramp and stay in Quick-Stop”; the motor is brought to the rotational speed 0 based on the rapid stop ramp.

The drive controller can then be brought back into the status “Operation Enabled” by the comma nd “Enable Operation”, object Statusword (6041h).

Object description	
INDEX	605A
Name	Quick Stop Option Code
Object Code	Variable
Data Type	INTEGER16
Category	Optional

Entry description	
Access	RO
PDO Mapping	No
Default Value	0x6
Lower Limit	0x6
Upper Limit	0x6
Unit	-

Object 6060h: Modes of Operation

The operating mode of the drive control is set by the object. The following operating modes are supported:

- Profile Position mode (Positioning according to CiA DSP-402)
- Profile Velocity mode (Speed mode according to CiA DSP-402)
- Homing Mode (Homing of the drive according to CiA DSP-402)
- Interpolation mode with 1-sample deep queue, SYNC and cyclic reference communication.

Profile Position mode
In the operating mode Profile Position mode (value 1) the positioning commands are processed over the fieldbus. The reference value of a positioning command is the target position of the drive movement: Object Target position (607Ah). The target position is mapped together with the control word of the status machine (controlword) in the Receive-PDO R_PDO2; see object 2nd receive PDO mapping parameter (1601h).

Profile Velocity mode
In the operating mode Profile Velocity mode (value 3) the movement commands are processed over the fieldbus. The reference value of a movement command is the set speed of the drive movement: Object Target velocity (60FFh). The set speed is mapped together with the control word of the status machine (Controlword) in the Receive-PDO R_PDO3; see object 3rd receive PDO mapping parameter (1602h).

Homing Mode
In the operating mode homing (Homing Mode, value 6) the position memory of the positioning system is initialised. Homing takes place either through value assignment or through a reference movement.

There are some limitation when changing operating mode:
- Only accepted if position RANGE and position LIMITs are well defined = the 2 interval don't cross each other;
- Homing mode is accepted only if its homing offset is within position range AND within position limits;
- Homing, PP, IP mode are accepted only if NOT sensorless;
- The mode change is performed only if the preceding operation has completed, otherwise the request is refused and the error flag "statusword.13" would be set (but the SDO request will NOT return an error in this case – because the change request is checked when the SDO protocol has already been confirmed).

Object description	
INDEX	6060
Name	Modes of Operation
Object Code	Variable
Data Type	INTEGER8
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0xFF
Lower Limit	0xFF
Upper Limit	0x7
Unit	-

Object 6061h: Modes of Operation Display

The actual operational mode (selected by means of 6060h) can be read using this object.

Object description	
INDEX	6061
Name	Modes of Operation Display
Object Code	Variable
Data Type	INTEGER8
Category	Optional

Entry description	
Access	RO
PDO Mapping	Yes
Default Value	0xFF
Lower Limit	0xFF
Upper Limit	0x6
Unit	-

Object 6064h: Position Actual Value in User Unit

The object indicates the current position of the drive in user defined units. See chapter XXX for description of “NC units”.

PDO-Mapping
The object 6064h is available for PDO mapping. It is also mapped by default configuration to bytes 2..5 of the TPDO2; see object 2nd transmit PDO mapping parameter (1A01h).

Object description	
INDEX	6064
Name	Position Actual Value in User Unit
Object Code	Variable
Data Type	INTEGER32
Category	Optional

Entry description	
Access	RO
PDO Mapping	Yes
Default Value	0x0
Lower Limit	0x80000000
Upper Limit	0x7FFFFFFF
Unit	NC unit (see also chapter 5.2.1)

Object 6065h: Following Error Window

The following error window defines the range of tolerated position values symmetrical to the target position. If the current position is out of range a following error occurs.

Object description	
INDEX	6065
Name	Following Error Window
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0xFFFFFFFF
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	NC unit (see also chapter 5.2.1)

Object 6066h: Following Error Time Out

This object indicates the time for a following error condition, after that the bit 13 of the statusword shall be set to 1 in the profile position mode and in the cyclic synchronous position mode. The following error condition can generate the alarm A37 (if enabled).

Object description	
INDEX	6066
Name	Following Error Time Out
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x0014
Lower Limit	0x0002
Upper Limit	0x00C8
Unit	ms

Object 6067h: Position Window

The object (tolerance window for the target position) describes a symmetrical range in which the setpoint, object Target position (607Ah) is taken as reached and the bit Target reached in the status word, object Statusword (6041h), bit 10) is set.

The position regulator seeks independently of the size of the tolerance window to exactly approach the setpoint.

Object description	
INDEX	6067
Name	Position Window
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0xA
Lower Limit	0x0
Upper Limit	0x7FFFFFFF
Unit	NC unit (see also chapter 5.2.1)

Object 606Ch: Velocity Actual Value

The object indicates the current speed of the drive. If the motor is an asynchronous spindle used in V/f mode, the unit of this object is directly RPM; in all other cases, the unit is 1000 NC units/min.

PDO-Mapping
The object 6064 is available for PDO mapping. It is also mapped by default configuration in to bytes 2..5 of the TPDO3; see object 3rd transmit PDO mapping parameter (1A02h).

Object description	
INDEX	606C
Name	Velocity Actual Value
Object Code	Variable
Data Type	INTEGER32
Category	Optional

Entry description	
Access	RO
PDO Mapping	Yes
Default Value	0x0
Lower Limit	0x80000000
Upper Limit	0x7FFFFFFF
Unit	non V/f mode: 1000 NC unit/min V/f mode: RPM (see also chapter 5.2.1)

Object 606Dh: Velocity Window

The object describes the symmetrical range around the specified speed of rotation in transient response status in which the specified speed of rotation is considered as reached.

The specified speed of rotation is set in the operating mode “Profile position mode” by the object Profile velocity (6081h) and in the operating mode “Profile velocity mode” by the object Target velocity (60FFh).

Object description	
INDEX	606D
Name	Velocity Window
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x64
Lower Limit	0x0
Upper Limit	0x7FFF
Unit	non V/f mode: 1000 NC unit/min V/f mode: RPM (see also chapter 5.2.1)

Object 607Ah: Target Position

The object is used to command a new position value which will be approached. The position value is set for the positioning mode according to DSP 402.

Target position
Positioning is started by bit New setpoint in the control wort of the status machine:

- Object Controlword (6040h), bit 4.

The position value is interpreted dependent on bit 6 of the control word as an absolute or relative position. Positioning can only be executed from in the status “Operation Enabled” if the positioning mode is activated.

Plausibility checking
With transfer over of the start command for positioning, the drive checks whether the given target position is within the reference operating range. If the “START POSITIONING” bit is given when the object 607A is programmed with an invalid value, the alarm A90.1 is generated.

PDO-Mapping
The object 607Ah is available for PDO mapping. It is also mapped by default configuration to bytes 2..5 of the RPDO2; see object 2nd receive PDO mapping parameter (1601h).

Object description	
INDEX	607A
Name	Target Position
Object Code	Variable
Data Type	INTEGER32
Category	Optional

Entry description	
Access	RW
PDO Mapping	Yes
Default Value	0x0
Lower Limit	0xC1000000
Upper Limit	0x3F000000
Unit	NC unit (see also chapter 5.2.1)

Object 607Bh: Position Range Limit

This object describes the axis range. It determines the kind of axis in the context with “software position limit”. Please also refer to the description of object 607Dh “Software position limits”.

Object description	
INDEX	607B
Name	Position Range Limit
Object Code	Array
Data Type	INTEGER32
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x2
Lower Limit	0x2
Upper Limit	0x2
Unit	-

Sub-Index	001
Description	Min Position Range Limit
Entry Category	Optional
Data Type	INTEGER32
Access	RW
PDO Mapping	No
Default Value	0xC1000000
Lower Limit	0xC1000000
Upper Limit	0x3F000000
Unit	NC unit (see also chapter 5.2.1)

Sub-Index	002
Description	Max Position Range Limit
Entry Category	Optional
Data Type	INTEGER32
Access	RW
PDO Mapping	No
Default Value	0x3F000000
Lower Limit	0xC1000000
Upper Limit	0x3F000000
Unit	NC unit (see also chapter 5.2.1)

Object 607Ch: Home Offset

This object is the position difference between the home position and the zero position.

Object description	
INDEX	607C
Name	Home Offset
Object Code	Variable
Data Type	INTEGER32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0xC1000000
Upper Limit	0x3F000000
Unit	NC unit (see also chapter 5.2.1)

Object 607Dh: Software Position Limit

The object indicates the absolute software limit positions for the operating range. The drive can be operated between these limits.The software limit positions refers to the point which was established by homing as a zero point.

Relations between “Software position limits” and “Position range limits”
The mutual position of object 607Bh and 607Dh define the behavior of the drive’s measure scale. There are 3 interesting cases:

- linear axis, or rotative axis where the movement is less than one turn (for example, from 10° to 350°): in this case, the software position limit range 607Dh should be strictly internal to the position range limit 607Bh.
- Rotative axis, where the axis is allowed to move by more than one turn: in this case, the position range limit 607Bh should be strictly internal to the software position limit range 607Dh (in this way, the software position limit cannot be reached); the movement will use the shortest path (for example, incrementing by 2° when moving from 359° to 1°) and the measure will be incremented “modulo” 607Bh.
- Rotative axis, where the axis is NOT allowed to move by more than one turn and the measure should be “modularized”: in this case, the position range limit 607Bh should be EQUAL to the software position limit range 607Dh (in this way, the software position limit cannot be crossed and, for example, the movement from 359° to 1° is done by turning back by 358°); and the measure will be incremented “modulo” 607Bh.

In any case, object 607Bh and 607Dh should not overlap each other.

Object description	
INDEX	607D
Name	Software Position Limit
Object Code	Array
Data Type	INTEGER32
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x2
Lower Limit	0x2
Upper Limit	0x2
Unit	-

Sub-Index	001
Description	Min Software Position Limit
Entry Category	Optional
Data Type	INTEGER32
Access	RW
PDO Mapping	No
Default Value	0xC1000000
Lower Limit	0xC1000000
Upper Limit	0x3F000000
Unit	NC unit (see also chapter 5.2.1)

Sub-Index	002

Description	Max Software Position Limit
Entry Category	Optional
Data Type	INTEGER32
Access	RW
PDO Mapping	No
Default Value	0x3F000000
Lower Limit	0xC1000000
Upper Limit	0x3F000000
Unit	NC unit (see also chapter 5.2.1)

Object 607Fh: Max Profile Velocity (associated to V98)

This object specifies the maximal allowed velocity used during positioning moves at the end of the acceleration ramp.

Object description	
INDEX	607F
Name	Max Profile Velocity
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0xEA60
Lower Limit	0x1
Upper Limit	0x36EE80
Unit	non V/f mode: 1000 NC unit/min V/f mode: RPM (see also chapter 5.2.1)

In NUMDrive C CANopen drives there are 2 separate objects expressing a similar value (the maximum motor speed), the ST parameter V98 and the CANopen object 607Fh (687Fh for the $Axis_{n+1}$).
They should not be confused, because their purpose is different.

The V98 is a mechanical limit for the motor, and is loaded with the motor ST; while the CANopen object 607Fh is a limit used by the CANopen interface to check against speed requests coming from CANopen commands. In no case should be 607Fh > V98 (and the typical setting is 607Fh = V98).

The 2 values are expressed in different measure units, and this is the formula to match each other:

Formula

The formula to have the speed expressed in NCunits/min is:
NC units/min = rpm * MULTI/DIVI * (4*V360)

So, the limits are the same if:
607Fh = V98 * MULTI/DIVI * (4*V360) / 1000 while the last “1000” is because the scale of 607Fh is expressed as 1000th NCunits/min.

NOTE: the real speed limit is V98 * (V97/100).

Object 6081h: Profile Velocity in pp-mode

This object is the velocity in the profile position mode.
The object indicates the speed of travel of the drive after completing the acceleration phase.
The value should not exceed the object “Max profile velocity (607Fh)”

Object description	
INDEX	6081
Name	Profile Velocity in pp-mode
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	Yes
Default Value	0x3E8
Lower Limit	0x0
Upper Limit	0x36EE80
Unit	non V/f mode: 1000 NC unit/min V/f mode: RPM (see also chapter 5.2.1)

Object 6083h: Profile Acceleration

This object is the linear acceleration in the profile velocity mode and the profile position mode.
If the motor is an asynchronous spindle used in V/f mode, the unit of this object is RP/s^2 (that means turns/s^2); in all other cases, the unit is 1000 NC units/s^2.

Object description	
INDEX	6083
Name	Profile Acceleration
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	Yes
Default Value	0x3E8
Lower Limit	0xA
Upper Limit	0x186A0
Unit	non V/f mode: 1000 NC unit/s^2 V/f mode: TURN/s^2 (see also chapter 5.2.1)

Object 6084h: Profile Deceleration

This object is the linear deceleration.
Profile deceleration is given in the same units as profile acceleration. Please refer to the description of object 6083h for a description about the units used for this parameter.

Object description	
INDEX	6084
Name	Profile Deceleration
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	Yes
Default Value	0x3E8
Lower Limit	0xA
Upper Limit	0x186A0
Unit	non V/f mode: 1000 NC unit/s^2 V/f mode: TURN/s^2 (see also chapter 5.2.1)

Object 6085h: Quick Stop Deceleration

The object prescribes the delay for a rapid stop (Quick stop).

Sub-index 00h
Specification of the rapid stop deceleration is in acceleration units; see object “Acceleration dimension index” (608Eh). For a compact drive the specification represents an increase in motor incrementas per second squared.

Default value
The default value for the rapid stop deceleration represents the minimum deceleration ramp of the compact drive, see object “Max deceleration” (60C6h). The rapid stop deceleration cannot be adjusted.

Object description	
INDEX	6085
Name	Quick Stop Deceleration
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x3E8
Lower Limit	0xA
Upper Limit	0x186A0
Unit	non V/f mode: 1000 NC unit/s^2 V/f mode: TURN/s^2 (see also chapter 5.2.1)

Object 6086h: Motion Profile Type

This object is used to select the type of the motion profile. The NUMDrive C CANopen drive only supports trapezoidal movement profiles (linear ramp, value = ”0”).

Object description	
INDEX	6086
Name	Motion Profile Type
Object Code	Variable
Data Type	INTEGER16
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0x0
Unit	-

Object 6098h: Homing Method

This object determines the method that will be used during the homing mode.

- -x = manufacturer specific
- 0 = No homing operation required
- 1-35 = Methods 1 to 35 (see note below)
- 36-127 = reserved.

Object description	
INDEX	6098
Name	Homing Method
Object Code	Variable
Data Type	INTEGER8
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x23
Lower Limit	0x1
Upper Limit	0x23
Unit	-

Please refer to the chapter 5.7.6 “Homing mode and homing setting” and DSP-402 for more detailed information upon Homing Method.

Object 6099h: Homing Speeds

This object defines a slow and a fast velocity used during the homing mode.

Object description	
INDEX	6099
Name	Homing Speeds
Object Code	Array
Data Type	UNSIGNED32
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x2
Lower Limit	0x2
Upper Limit	0x2
Unit	-

Sub-Index	001
Sub-Index	001
Description	Fast Homing Speed
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x64
Lower Limit	0x1
Upper Limit	0x57E40
Unit	non V/f mode: 1000 NC unit/min V/f mode: RPM (see also chapter 5.2.1)

Sub-Index	002
Description	Slow Homing Speed
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x32
Lower Limit	0x1
Upper Limit	0x8CA0
Unit	non V/f mode: 1000 NC unit/min V/f mode: RPM (see also chapter 5.2.1)

Object 609Ah: Homing Acceleration

This object is the acceleration used in the homing mode.
Acceleration to be used for all accelerations and decelerations with the standard homing modes, in acceleration units.

Object description	
INDEX	609A
Name	Homing Acceleration
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x3E8
Lower Limit	0xA
Upper Limit	0x186A0
Unit	non V/f mode: 1000 NC unit/s^2 V/f mode: TURN/s^2 (see also chapter 5.2.1)

Object 60C0h: Interpolation Sub Mode Select

This read-only object shows the interpolation modes availables.
Currently NUMDrice C CANopen only supports linear interpolation.

Object description	
INDEX	60C0
Name	Interpolation Sub Mode Select
Object Code	Variable
Data Type	INTEGER16
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x00
Upper Limit	0x00
Unit	-

Object 60C1h: Interpolation Data Record

This object indicates data words, which are necessary to perform the interpolation algorithm. The number N of data words in the record is defined by interpolation data configuration.

The interpretation of the data words in interpolation data record may vary with the different possible interpolation modes as set by the interpolation_sub_mode_select.
For the linear interpolation mode each interpolation data record simply is regarded as a new position set-point.

Object description	
INDEX	60C1
Name	Interpolation Data Record
Object Code	Array
Data Type	INTEGER32
Category	Optional

Entry description	
Sub-Index	000
Description	Highest sub-index supported
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x1
Lower Limit	0x00000001
Upper Limit	0x00000001
Unit	-

Sub-Index	001
Description	Data Record 1
Entry Category	Optional
Data Type	INTEGER32
Access	RW
PDO Mapping	Yes
Default Value	0x0
Lower Limit	0x80000000
Upper Limit	0x7FFFFFFF
Unit	-

Object 60C2h: Interpolation Time Period

This object indicates the interpolation cycle time.
This object includes the following sub-indices:

- sub-index 1: value of the time
- sub-index 2: dimension index to the time value in sub-index 1.

Object description	
INDEX	60C2
Name	Interpolation Time Period
Object Code	Record
Data Type	P402_IP_PERIOD_T
Category	Optional

Entry description	
Sub-Index	000
Description	Highest sub-index supported
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x02
Lower Limit	0x02
Upper Limit	0x0
Unit	-

Sub-Index	001
Description	Interpolation time period value
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0x2
Lower Limit	0x2
Upper Limit	0x14
Unit	(10^interpolation_time_index) s

Sub-Index	002
Description	Interpolation time index
Entry Category	Optional
Data Type	INTEGER8
Access	RW
PDO Mapping	No
Default Value	0xFD
Lower Limit	0xFD
Upper Limit	0xFD
Unit	-

Object 60F2h: Positioning Option Code

This object indicates the positioning behaviour as described by the profile position mode or the interpolated position mode. This object is organized bit-wise.

The bits have the following meaning:
- bit 0,1: relative option
- bit 2-15 not available.

Object description	
INDEX	60F2
Name	Positioning Option Code
Object Code	Variable
Data Type	INTEGER16
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x0000
Upper Limit	0x0002
Unit	-

Object 60FDh: Digital Inputs

This object reflects the current state of the digital input signals. The user may apply any signals to these inputs for special purposes like limit or reference switches.

Bit	Description
0	Negative limit switch
1	Positive limit switch
2	HOME position switch

Object description	
INDEX	60FD
Name	Digital Inputs
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RO
PDO Mapping	Yes
Default Value	0x00
Lower Limit	0x00000000
Upper Limit	0xFFFFFFFF
Unit	-

Object 60FFh: Target Velocity

The object is used to enter the required speed setpoint for a drive movement in the operating mode “Profile velocity mode”.

Setpoint speed
A drive movement in Profile Velocity mode is started by deleting the bit Halt (value 1 -> 0) in the control word of the status machine: Object Controlword (6040h), bit 8.
Drive movements can only be executed from the status “Operation Enabled” if the speed mode (Profile Velocity mode) is activated.

PDO-Mapping
The object 60FFh is available for PDO mapping. It is also mapped by default configuration to bytes 2..5 of the RPDO3; see object 3rd receive PDO mapping parameter (1602h).

Object description	
INDEX	60FF
Name	Target Velocity
Object Code	Variable
Data Type	INTEGER32
Category	Optional

Entry description	
Access	RW
PDO Mapping	Yes
Default Value	0x3E8
Lower Limit	0xFFC91180
Upper Limit	0x36EE80
Unit	non V/f mode: 1000 NC unit/min V/f mode: RPM (see also chapter 5.2.1)

Object 6502h: Supported Drive Modes

This object documents all operational modes which are supported by the drive.

- Bit 31-16 = manufacturer specific
- Bit 15-07 = reserved
- Bit 6 = ip
- Bit 5 = hm
- Bit 4 = reserved
- Bit 3 = tq
- Bit 2 = pv
- Bit 1 = vl
- Bit 0 = pp.

Object description	
INDEX	6502
Name	Supported Drive Modes
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RO
PDO Mapping	No
Default Value	0x65
Lower Limit	0x65
Upper Limit	0x65
Unit	-

5.2.3 Object dictionary access rights

Not all objects can be written at any time. Please refer to the summary as indicated on the below.

Index	Name	Access notes
1000	Device Type	Read Only
1001	Error Register	Read Only
1003	Pre-defined Error Field	Read Only
1005	COB ID SYNC	PRE-OP and SwitchOnDisabled
1006	Communication Cycle Period	PRE-OP and SwitchOnDisabled
100C	Guard Time	Read / Write
100D	Life Time Factor	Read / Write
1012	COB ID Time Stamp	Read / Write
1014	COB ID EMCY	Read / Write
1015	Inhibit Time Emergency	Read / Write
1016	Heartbeat Consumer Entries	PRE-OP, before the first heartbeat
1017	Producer Heartbeat Time	PRE-OP, before the first heartbeat
1018	Identity Object	Read / Write
1020	Verify Configuration	Read / Write
1200	Server SDO Parameter 1	Read / Write
1201	Server SDO Parameter 2	Read / Write
1400	Receive PDO Communication Parameter 1	Only when comm. FSM is in PRE-OPERATIONAL
1401	Receive PDO Communication Parameter 2	Only when comm. FSM is in PRE-OPERATIONAL
1402	Receive PDO Communication Parameter 3	Only when comm. FSM is in PRE-OPERATIONAL
1403	Receive PDO Communication Parameter 4	Only when comm. FSM is in PRE-OPERATIONAL
1600	Receive PDO Mapping Parameter 1	Only when comm. FSM is in PRE-OPERATIONAL
1601	Receive PDO Mapping Parameter 2	Only when comm. FSM is in PRE-OPERATIONAL
1602	Receive PDO Mapping Parameter 3	Only when comm. FSM is in PRE-OPERATIONAL
1603	Receive PDO Mapping Parameter 4	Only when comm. FSM is in PRE-OPERATIONAL
1800	Transmit PDO Communication Parameter 1	Only when comm. FSM is in PRE-OPERATIONAL
1801	Transmit PDO Communication Parameter 2	Only when comm. FSM is in PRE-OPERATIONAL
1802	Transmit PDO Communication Parameter 3	Only when comm. FSM is in PRE-OPERATIONAL
1803	Transmit PDO Communication Parameter 4	Only when comm. FSM is in PRE-OPERATIONAL
3002	RPDO1 shadow before SYNC validation	Only when comm. FSM is in PRE-OPERATIONAL
1A00	Transmit PDO Mapping Parameter 1	Only when comm. FSM is in PRE-OPERATIONAL
1A01	Transmit PDO Mapping Parameter 2	Only when comm. FSM is in PRE-OPERATIONAL
1A02	Transmit PDO Mapping Parameter 3	Only when comm. FSM is in PRE-OPERATIONAL
1A03	Transmit PDO Mapping Parameter 4	Only when comm. FSM is in PRE-OPERATIONAL
6007	Abort Connection Option Code [Axis^{n}]	Read Only[1]
603F	Error Code [Axis$_{n}$]	Read Only
6040	Controlword [Axis$_{n}$]	Read / Write
6041	Statusword [Axis$_{n}$]	Read Only
605A	Quick Stop Option Code [Axis$_{n}$]	Read Only[1]

6060	Modes of Operation [Axis$_n$]	• Only accepted if position RANGE and position LIMITs are well defined = the 2 interval do not cross each other. • Homing mode is accepted only if its homing offset is within position range AND within position limits. • Homing, PP, IP mode are accepted only if NOT sensorless. • The mode change is performed only if the preceding operation has completed, otherwise the request is refused and the error flag “statusword.13” would be set (but the SDO request will NOT return an error in this case – because the change request is checked when the SDO protocol has already been confirmed).
6061	Modes of Operation Display [Axis$_n$]	Read Only
6064	Position Actual Value in User Unit [Axis$_n$]	Read Only
6065	Following Error Window [Axis$_n$]	Read / Write
6066	Following Error time out [Axis$_n$]	Read / Write
6067	Position Window [Axis$_n$]	Read / Write
606C	Velocity Actual Value [Axis$_n$]	Read Only
606D	Velocity Window [Axis$_n$]	Read / Write
607A	Target Position [Axis$_n$]	Everytime, used on START bit edge
607B	Position Range Limit [Axis$_n$]	• Only changeable if in SWITCH-ON DISABLED or FAULT state. • Obviously it accepts only INF < SUP.
607C	Home Offset [Axis$_n$]	• Only changeable if in SWITCH-ON DISABLED or FAULT state. • Accepted only if it is within position range AND within position limits.
607D	Software Position Limit [Axis$_n$]	• Only changeable if in SWITCH-ON DISABLED or FAULT state. • Obviously it accepts only INF < SUP.
607F	Max Profile Velocity [Axis$_n$]	Read / Write
6081	Profile Velocity in pp mode [Axis$_n$]	Read / Write
6083	Profile Acceleration [Axis$_n$]	Read / Write
6084	Profile Deceleration [Axis$_n$]	Read / Write
6085	Quick Stop Deceleration [Axis$_n$]	Read / Write
6086	Motion Profile Type [Axis$_n$]	Read / Write
6098	Homing Method [Axis$_n$]	• Can be changed only if “another” homing operation is not running. Otherwise the SDO request will be refused.
6099	Homing Speeds [Axis$_n$]	Read / Write
609A	Homing Acceleration [Axis$_n$]	Read / Write
60C0	Interpolation Sub Mode Select	Read / Write
60C1	Interpolation Data Record	Read / Write
60C2	Interpolation Time Period	• Can be changed only when IP mode is not selected. When the IP mode is selected and the drive is in OPERATION ENABLED it must match V266 otherwise an alarm A16.266 will be generated.
60FD	Digital Inputs [Axis$_n$]	Read Only
60F2	Positioning Option Code [Axis$_n$]	Read Only
60FF	Target Velocity [Axis$_n$]	Read / Write
6502	Supported Drive Modes [Axis$_n$]	Read Only

5.3 NUMDrive C Bi-Axes objects description

5.3.1 NC Units

NUMDrive C CANopen Bi-Axes uses “NC units” to measure position, speed, acceleration.
One “NC unit” is the minimum increment the user want to appreciate (beware, it is not an indication of the maximum resolution achievable with the system: is an expression of the desired resolution).

The “Multi/Divi” selection is done to have all the measure-related objects expressed in this unit.

With an example the things are simpler:
- the axis is a linear one, with a rotative motor and a screw;
- the measure should be expressed in micrometers: this means, for example, object 6064h will be directly “micron”;
- the sensor is a 1024 pulse encoder;
- there is a 1:1 gearbox;
- the screw is 10mm/turn;
- this means the measure will increment by 10.000 microns every turn;
- the suggested Multi/Divi is 10000/4096
  (please refer to the chapter 5.5.1 Parameters description)

5.3.2 Objects description

Object 1000h: Device Type

The device shows the implemented device profile and the device type.
The lower 16 bit contain the device profile number and the upper 16 bit an additional information.

Object description	
INDEX	1000
Name	Device Type
Object Code	Variable
Data Type	UNSIGNED32
Category	Mandatory

Entry description	
Access	CONST
PDO Mapping	No
Default Value	0x0FFF0192
Lower Limit	0x00000000
Upper Limit	0xFFFFFFFF
Unit	-

Object 1001h: Error Register

The object shows the error status of the device.
The error register is an 8-bit field, and the format is fixed by the CANopen specification.

The detailed description of the fault can be found with the object predefined error field (1003h) and with the object error code (603Fh).

Errors are already signalled by an EMCY message as soon as they occur.

Bit Meaning

- 0 generic error
- 1 reserved
- 2 reserved
- 3 reserved
- 4 communication error (overrun, error state)
- 5 device profile specific
- 6 reserved
- 7 manufacturer specific.

Object description	
INDEX	1001
Name	Error Register
Object Code	Variable
Data Type	UNSIGNED8
Category	Mandatory

Entry description	
Access	RO
PDO Mapping	No
Default Value	0x00
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Object 1003h: Pre-defined Error Field

The object saves the latest error messages that were shown as EMCY messages.

- The entry under subindex 00h contains the number of saved error messages.
- The current error message is stored under subindex 01h, older messages are moved to high subindex entries.
- Writing 0 to subindex 00h resets the error list.

Object description	
INDEX	1003
Name	Pre-defined Error Field
Object Code	Array
Data Type	UNSIGNED32
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Errors
Entry Category	Mandatory
Access	RW
PDO Mapping	No
Default Value	0x1
Lower Limit	0x00000000
Upper Limit	0x000000FE
Unit	-

Sub-Index	001
Description	Standard Error Field
Entry Category	Mandatory
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x00000000
Upper Limit	0xFFFFFFFF
Unit	-

Please refer to the chapter 2.12.1 for more detailed informations.

Object 1005h: COB-ID SYNC

The object shows the COB-ID of the SYNC object and specifies whether a device sends or receives SYNC messages.

NUMDC-CAN drive can only receive SYNC messages.
For synchronisation a device in the network must send SYNC objects.

The device generate a SYNC message if bit 30 is set.
The COB-ID can be changed in the NMT “Pre-Operational” status only..

Object description	
INDEX	1005
Name	COB-ID SYNC
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x80000080
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Object 1006h: Communication Cycle Period

This entry defines the communication cycle period in us.
It is 0, if not used.

Object description	
INDEX	1006
Name	Communication Cycle Period
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x00000000
Upper Limit	0xFFFFFFFF
Unit	us

Object 100Ch: Guard Time

This entry contains the guard time in milli-seconds.
It is 0, if not used.
It shows the time span for connection monitoring (node guarding) of an NMT slave.

The time span for the connection monitoring of a NMT master is derived from the time span “guard time” multiplied by the “life time factor”, object Life time factor(100Dh) .

The time span can be changed in the NMT “Pre-Operational” status only.

Object description	
INDEX	100C
Name	Guard Time
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	ms

Object 100Dh: Life Time Factor

The object shows the factor that together with the time span “guard time” gives the time interval for the connection monitoring of a NMT master.

Within this period the NMT slave device waits for a monitoring request by node guarding by the NMT master device. Life time = (guard time * life time factor).

The value “0” disables the monitoring of the NMT master.
If the connection monitoring by the NMT master remains disabled during the time interval “life time”, the device reports an error and switches to error status.

The time factor can be changed in the NMT “Pre-Operational” status.The time span “guard time” is set with the object Guard time (100Ch)

Object description	
INDEX	100D
Name	Life Time Factor
Object Code	Variable
Data Type	UNSIGNED8
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFF
Unit	-

Object 1012h: COB-ID Time Stamp

COB-ID of the Time Stamp Object (TIME).
If bit 31 is set the device consumes the Time Stamp message and if bit 30 is set the device produces the Time Stamp message.
The meaning of the other bits is equal to the other communication objects.

Object description	
INDEX	1012
Name	COB-ID Time Stamp
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x80000100
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Object 1014h: COB-ID EMCY

COB-ID used for emergency message (Emergency Server).

Object description	
INDEX	1014
Name	COB-ID EMCY
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RO
PDO Mapping	No
Default Value	0x00000080
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Object 1015h: Inhibit Time Emergency

The object specifies the waiting (inhibit) period for the repeated transmission of EMCY messages as a multiple of 100µs.

Object description	
INDEX	1015
Name	Inhibit Time Emergency
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	-

Object 1016h: Heartbeat Consumer Entries

The object saves the settings of the “heartbeat” consumer for NMT monitoring by “heartbeat” connection message. The consumer heartbeat time defines the expected heartbeat cycle time and thus has to be higher than the corresponding producer heartbeat time configured on the device producing this heartbeat.

The node number (Node-Id) and the monitoring time (Heartbeat time) of network users are established by means of the object which is being mo-nitored by the drive control over the Heartbeat protocol.

- The bits 31 - 24 of the sub-index 01 should be 0.
- The bits 23 - 16 contain the node-id.
- The lower 16 bits contain the heartbeat time.

Sub-index 01h
HeartbeatMonitoring

A device sends its status cyclically by means of theHeartbeat protocol; see object Producer heartbeat time (1017h). The “heart beat” of selected devices can be monitored by the drive control. If the heart beat is not received then an event is triggered (Abort connection).
The reaction of the drive to such an event can be defined with the object Abort connection option code (6007h/687Fh).

Object description	
INDEX	1016
Name	Heartbeat Consumer Entries
Object Code	Array
Data Type	UNSIGNED32
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x01
Lower Limit	0x1
Upper Limit	0x7F
Unit	-

Sub-Index	001
Description	Consumer Heartbeat Time 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0x02FFFFFF
Unit	-

Object 1017h: Producer Heartbeat Time

This object is the companion of 1016h.
The producer heartbeat time defines the cycle time of the heartbeat.
If the time is 0 it is not used.
The time has to be a multiple of 1 msec.

Object description	
INDEX	1017
Name	Producer Heartbeat Time
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	-

Object 1018h: Identity Object

This object contains general information about the device.

- Sub-Index 1 contains a unique value allocated each manufacturer.
- Sub-Index 2 identifies the manufacturer specific product code (device version).
- Sub-Index 3 contains the revision number.
- Bit 31-16 is the major revision number and Bit 15-0 the minor revision number.
- Sub-Index 4 identifies a manufacturer specific serial number.

Object description	
INDEX	1018
Name	Identity Object
Object Code	Record
Data Type	IDENTITY
Category	Mandatory

Entry description	
Sub-Index	000
Description	number of entries
Entry Category	Mandatory
Access	RO
PDO Mapping	No
Default Value	0x4
Lower Limit	0x1
Upper Limit	0x4
Unit	-

Sub-Index	001
Description	Vendor Id
Entry Category	Mandatory
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Product Code
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	003
Description	Revision number
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x01
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	004
Description	Serial number
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Object 1020h: Verify Configuration

This object contains date and time of the last configuration.

- Sub index 001 shall contain the number of days after January 1, 1984.
- Sub index 002 shall contain the number of seconds after midnight.

Object description	
INDEX	1020
Name	Verify Configuration
Object Code	Array
Data Type	UNSIGNED32
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x2
Lower Limit	0x02
Upper Limit	0x02
Unit	-

Sub-Index	001
Description	Configuration date
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Configuration time
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Object 1200h: Server SDO Parameter 1

The object contains the parameters for the SDOs for which the device is the server. Itis here for information only: the default settings shouldn't be changed.

Object description	
INDEX	1200
Name	Server SDO Parameter 1
Object Code	Record
Data Type	SDO_PARAMETER
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x02
Lower Limit	0x02
Upper Limit	0x02
Unit	-

Sub-Index	001
Description	COB-ID Client -> Server
Entry Category	Mandatory
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x00000600
Lower Limit	0x00000600
Upper Limit	0xBFFFFFFF
Unit	-

Sub-Index	002
Description	COB-ID Server -> Client
Entry Category	Mandatory
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x00000580
Lower Limit	0x00000580
Upper Limit	0xBFFFFFFF
Unit	-

Object 1201h: Server SDO Parameter 2

The object contains the parameters for the SDOs for which the device is the server.
It is here for information only: the default settings shouldn't be changed.

Object description	
INDEX	1201
Name	Server SDO Parameter 2
Object Code	Record
Data Type	SDO_PARAMETER
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x03
Lower Limit	0x02
Upper Limit	0x03
Unit	-

Sub-Index	001
Description	COB-ID Client -> Server
Entry Category	Mandatory
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x800006E0
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	COB-ID Server -> Client
Entry Category	Mandatory
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x800006E0
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	003
Description	Node ID of the SDO Client
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x00
Upper Limit	0x7F
Unit	-

Object 1400h: Receive PDO Communication Parameter 1 [Axis$_n$]

The object saves the settings for the first receive PDO (RPDO1). It is composed of subindex 0..3.

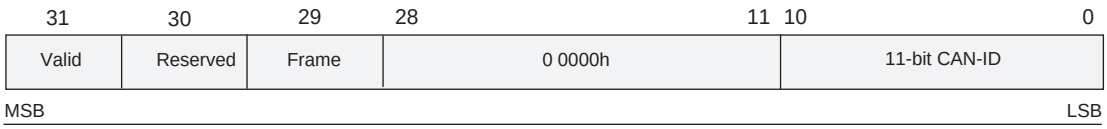
Object description	
INDEX	1400
Name	Receive PDO Communication Parameter 1
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

- **Sub index 0: number of entries, ReadOnly**

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x03
Lower Limit	0x02
Upper Limit	0x05
Unit	-

- **Sub index 1: enable, COB-ID setting**

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000200
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-



Structure of RPDO COB-ID

Bit 31: RPDO is enabled if bit 31=”0”.
The COB-ID of the object can be changed in the NMT “Pre-Operational” status only.

- **Sub index 2: transmission type**

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFF
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

The transmission type define when the received value is evaluated.

- Transmission type = 0 (acyclic transmission): the evaluation is linked to the SYNC object, but not the transmission of the PDO. A received PDO message is evaluated with the following SYNC.
- Transmission type = 1..252 (cyclic transmission): a value between 1 and 240 shows the number of SYNC cycles after which a received PDO is evaluated.
- Transmission types = 241..251 are reserved
- Transmission type = 252 is RTR-only, synchronous mode.
- Transmission type = 253 is RTR-only, asynchronous mode.
- Transmission type = 254..255 is asynchronous, the received value is used immediately when received.

- **Sub index 3: inhibit time**

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

The inhibit time is the last filter before RPDO evaluationt. It is specified in 100-uS units.

Object 1401h: Receive PDO Communication Parameter 2 [Axis$_n$]

The object saves the settings for RPDO2.
Please refer to the description of the object 1400h for more detailed information.

Object description	
INDEX	1401
Name	Receive PDO Communication Parameter 2
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x03
Lower Limit	0x02
Upper Limit	0x05
Unit	-

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x80000300
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFF
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

Object 1402h: Receive PDO Communication Parameter 3 [Axis$_n$]

The object saves the settings for RPDO3.
Please refer to the description of the object 1400h for more detailed information.

Object description	
INDEX	1402
Name	Receive PDO Communication Parameter 3
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x03
Lower Limit	0x02
Upper Limit	0x05
Unit	-

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x80000400
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFF
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

Object 1403h: Receive PDO Communication Parameter 4 [Axis$_n$]

The object saves the settings for RPDO4.
Please refer to the description of the object 1400h for more detailed information.

Object description	
INDEX	1403
Name	Receive PDO Communication Parameter 4
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x03
Lower Limit	0x02
Upper Limit	0x05
Unit	-

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x80000500
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFE
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

Object 1440h: Receive PDO Communication Parameter 65 [Axis$_{n+1}$]

Please refer to Object 1400h for the detalied description.

Object description	
INDEX	1440
Name	Receive PDO Communication Parameter 65
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x03
Lower Limit	0x02
Upper Limit	0x05
Unit	-

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0xC0000000
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFF
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

Object 1441h: Receive PDO Communication Parameter 66 [Axis$_{n+1}$]

Please refer to Object 1401h for the detalied description.

Object description	
INDEX	1441
Name	Receive PDO Communication Parameter 66
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x03
Lower Limit	0x02
Upper Limit	0x05
Unit	-

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0xC0000000
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFF
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

Object 1442h: Receive PDO Communication Parameter 67 [Axis$_{n+1}$]

Please refer to Object 1402h for the detalied description.

Object description	
INDEX	1442
Name	Receive PDO Communication Parameter 67
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x03
Lower Limit	0x02
Upper Limit	0x05
Unit	-

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0xC0000000
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFF
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

Object 1443h: Receive PDO Communication Parameter 68 [Axis$_{n+1}$]

Please refer to Object 1403h for the detalied description.

Object description	
INDEX	1443
Name	Receive PDO Communication Parameter 68
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x03
Lower Limit	0x02
Upper Limit	0x05
Unit	-

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0xC0000000
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFE
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

Object 1600h: Receive PDO Mapping Parameter 1 [Axis$_n$]

The object saves the settings for the mapping of the 1st receive PDO (RPDO1), and shows which objects are mapped in RPDO1 and received with the PDO itself.
The PDO mapping for RPDO1 cannot be modified.

It is composed of subindex 0..1.

Object description	
INDEX	1600
Name	Receive PDO Mapping Parameter 1
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RO
PDO Mapping	No
Default Value	0x01
Lower Limit	0x0
Upper Limit	0x40
Unit	-

Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x60400010
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Object 1601h: Receive PDO Mapping Parameter 2 [Axis$_n$]

The object saves the settings for the mapping of the 2nd receive PDO (RPDO2), and shows which objects are mapped in RPDO2 and received with the PDO itself.

The PDO mapping for RPDO2 cannot be modified. The default assignment is preset and useful for the “profile position mode”. It is composed of subindex 0..2.

Object description	
INDEX	1601
Name	Receive PDO Mapping Parameter 2
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

- **Sub index 0: number of entries, Read-only**

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RO
PDO Mapping	No
Default Value	0x02
Lower Limit	0x0
Upper Limit	0xFF
Unit	-

- **Sub index 1: 1st mapping entry (fixed)**

Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x60400010
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

PDO mapping of the object “controlword” (6040h).

- **Sub index 2: 2nd mapping entry (fixed)**

Sub-Index	002
Description	Mapping Entry 2
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x607A0020
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

PDO mapping of the object “target position” (607Ah).

Object 1602h: Receive PDO Mapping Parameter 3 [Axis$_n$]

The object saves the settings for the mapping of the 3rd receive PDO (RPDO3), and shows which objects are mapped in RPDO3 and received with the PDO itself.

The PDO mapping for RPDO3 cannot be modified. The default assignment is preset for speed mode in the “profile velocity mode”. It is composed of subindex 0..2.

Object description	
INDEX	1602
Name	Receive PDO Mapping Parameter 3
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

- **Sub index 0: number of entries, Read-only**

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RO
PDO Mapping	No
Default Value	0x02
Lower Limit	0x0
Upper Limit	0xFF
Unit	-

- **Sub index 1: 1st mapping entry (fixed)**

Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x60400010
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

PDO mapping of the object “controlword” (6040h).

- **Sub index 2: 2nd mapping entry (fixed)**

Sub-Index	002
Description	Mapping Entry 2
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x60FF0020
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

PDO mapping of the object “target velocity” (60FFh)

Object 1603h: Receive PDO Mapping Parameter 4 [Axis$_n$]

The object saves the settings for the mapping of the 4th receive PDO (RPDO4), and shows how objects can be mapped in RPDO4 and received with the PDO itself.

The PDO mapping for RPDO4 can be modified (and should be written before the use).
RPDO4 is for expert users only. It is composed of subindex 0..2.

Object description	
INDEX	1603
Name	Receive PDO Mapping Parameter 4
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

- **Sub index 0: number of entries, Read-only**

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFF
Unit	-

- **Sub index 1..4: mapping entries (fixed)**

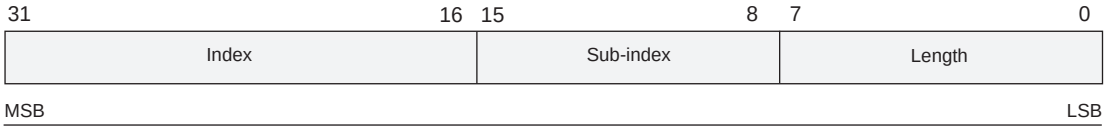
Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFFFFF
Unit	-

Sub-Index	002
Description	Mapping Entry 2
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFFFFF
Unit	-

Sub-Index	003
Description	Mapping Entry 3
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFFFFF
Unit	-

Sub-Index	004
Description	Mapping Entry 4
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFFFFF
Unit	-

Here should be written the selection of objects to map, and the corresponding length.
The format is…



(Where “length” is the bit size of the mapped object: for example, 0x10 for a word like “statusword 6041h”)

The sequence to configure a new mapping for RPDO4 is:
- Deactivate the mapping by writing 0 to subindex 0;
- Select the desired objects and size writing to subindex 1 to 4 (stop when ready);
- Activate the mapping by writing to subindex 0 the number of selected objects.

Object 1640h: Receive PDO Mapping Parameter 65 [Axis$_{n+1}$]

Please refer to Object 1600h for the detalied description.

Object description	
INDEX	1640
Name	Receive PDO Mapping Parameter 65
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RW
PDO Mapping	No
Default Value	0x01
Lower Limit	0x0
Upper Limit	0x40
Unit	-

Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x68400010
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Object 1641h: Receive PDO Mapping Parameter 66 [Axis$_{n+1}$]

Please refer to Object 1601h for the detalied description.

Object description	
INDEX	1641
Name	Receive PDO Mapping Parameter 66
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RW
PDO Mapping	No
Default Value	0x02
Lower Limit	0x0
Upper Limit	0xFF
Unit	-

Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x68400010
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Mapping Entry 2
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x687A0020
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Object 1642h: Receive PDO Mapping Parameter 67 [Axis$_{n+1}$]

Please refer to Object 1602h for the detalied description.

Object description	
INDEX	1642
Name	Receive PDO Mapping Parameter 67
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RW
PDO Mapping	No
Default Value	0x02
Lower Limit	0x0
Upper Limit	0xFF
Unit	-

Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x68400010
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Mapping Entry 2
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x68FF0020
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Object 1643h: Receive PDO Mapping Parameter 68 [Axis$_{n+1}$]

Please refer to Object 1603h for the detalied description.

Object description	
INDEX	1643
Name	Receive PDO Mapping Parameter 68
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x0
Upper Limit	0xFF
Unit	-

Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFFFFF
Unit	-

Sub-Index	002
Description	Mapping Entry 2
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFFFFF
Unit	-

Sub-Index	003
Description	Mapping Entry 3
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFFFFF
Unit	-

Sub-Index	004
Description	Mapping Entry 4
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Object 1800h: Transmit PDO Communication Parameter 1 [Axis$_n$]

The object saves the settings for the first transmit PDO (TPDO1). It is composed of subindex 0..5.

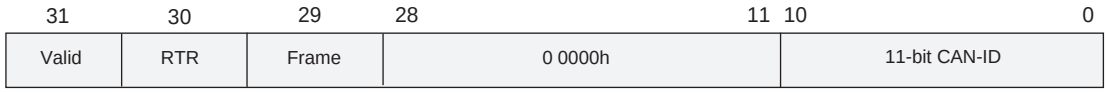
Object description	
INDEX	1800
Name	Transmit PDO Communication Parameter 1
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

- **Sub index 0: number of entries, Read-only**

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x05
Lower Limit	0x02
Upper Limit	0x06
Unit	-

- **Sub index 1: enable, COB-ID setting**

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000180
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-



Structure of TPDO COB-ID

- Bit 31: TPDO is enabled if bit 31=”0”.
- Bit 30: RTR bit. If a device supports TPDOs with RTR (remote transmission request), it can request a PDO from a PDO producer with RTR = “0” in accordance with the producer consumer relationship.

Settings

TPDO1 is sent asynchronously and/or event-controlled at every change of the PDO data.
The byte assignment of the TPDO1 is specified via PDO mapping with the object 1st transmit PDO mapping (1A00h).

The following assignment is the default:

- Bytes 0..1: statusword (6041h).

The COB-ID of the object can be changed in the NMT “Pre-Operational” status.

- **Sub index 2: transmission type**

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFF
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

The transmission type define when the received value is evaluated.

- Transmission type = 0 (acyclic transmission): the evaluation is linked to the SYNC object, but not the transmission of the PDO. A received PDO message is evaluated with the following SYNC.
- Transmission type = 1..252 (cyclic transmission): a value between 1 and 240 shows the number of SYNC cycles after which a received PDO is evaluated.
- Transmission types = 241..251 are reserved
- Transmission type = 252 is RTR-only, synchronous mode.
- Transmission type = 253 is RTR-only, asynchronous mode.
- Transmission type = 254..255 is asynchronous, the received value is used immediately when received.

- **Sub index 3: inhibit time**

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

The inhibit time is the last filter before TPDO transmission, defining the maximum rate for message generation. It is specified in 100-uS units.

- **Sub index 4: reserved**
  Do not modify.

- **Sub index 5: event timer**

Sub-Index	005
Description	Event Timer
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	ms

If the value is different from 0, a transmission is triggered every “event timer” ms. This trigger source add to other sources (if, for example, the transmission type is 255).

Object 1801h: Transmit PDO Communication Parameter 2 [Axis$_n$]

The object saves the settings for TPDO2.
Please refer to the description of the object 1800h for more detailed information.

Object description	
INDEX	1801
Name	Transmit PDO Communication Parameter 2
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x05
Lower Limit	0x02
Upper Limit	0x06
Unit	-

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000280
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFF
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

Sub-Index	004
Description	Compatibility Entry
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	005
Description	Event Timer
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	ms

Object 1802h: Transmit PDO Communication Parameter 3 [Axis$_n$]

The object saves the settings for TPDO3.
Please refer to the description of the object 1800h for more detailed information.

Object description	
INDEX	1802
Name	Transmit PDO Communication Parameter 3
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x05
Lower Limit	0x02
Upper Limit	0x06
Unit	-

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000380
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFF
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

Sub-Index	004
Description	Compatibility Entry
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	005
Description	Event Timer
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	ms

Object 1803h: Transmit PDO Communication Parameter 4 [Axis$_n$]

The object saves the settings for TPDO4.
Please refer to the description of the object 1800h for more detailed information.

Object description	
INDEX	1803
Name	Transmit PDO Communication Parameter 4
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x05
Lower Limit	0x02
Upper Limit	0x06
Unit	-

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000480
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFE
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

Sub-Index	004
Description	Compatibility Entry
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	005
Description	Event Timer
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	ms

Object 1840h: Transmit PDO Communication Parameter 65 [Axis$_{n+1}$]

Please refer to Object 1800h for the detalied description.

Object description	
INDEX	1840
Name	Transmit PDO Communication Parameter 65
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x05
Lower Limit	0x02
Upper Limit	0x06
Unit	-

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0xC0000000
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFF
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

Sub-Index	004
Description	Compatibility Entry
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	005
Description	Event Timer
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	ms

Object 1841h: Transmit PDO Communication Parameter 66 [Axis$_{n+1}$]

Please refer to Object 1801h for the detalied description.

Object description	
INDEX	1841
Name	Transmit PDO Communication Parameter 66
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x05
Lower Limit	0x02
Upper Limit	0x06
Unit	-

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0xC0000000
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFF
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

Sub-Index	004
Description	Compatibility Entry
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	005
Description	Event Timer
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	ms

Object 1842h: Transmit PDO Communication Parameter 67 [Axis$_{n+1}$]

Please refer to Object 1802h for the detalied description.

Object description	
INDEX	1842
Name	Transmit PDO Communication Parameter 67
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x05
Lower Limit	0x02
Upper Limit	0x06
Unit	-

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0xC0000000
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFF
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

Sub-Index	004
Description	Compatibility Entry
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	005
Description	Event Timer
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	ms

Object 1843h: Transmit PDO Communication Parameter 68 [Axis$_{n+1}$]

Please refer to Object 1803h for the detalied description.

Object description	
INDEX	1843
Name	Transmit PDO Communication Parameter 68
Object Code	Record
Data Type	PDO_COMM_PAR
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x05
Lower Limit	0x02
Upper Limit	0x06
Unit	-

Sub-Index	001
Description	COB-ID
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0xC0000000
Lower Limit	0x00000001
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Transmission Type
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0xFE
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	003
Description	Inhibit Time
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x1388
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	100 us

Sub-Index	004
Description	Compatibility Entry
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Sub-Index	005
Description	Event Timer
Entry Category	Optional
Data Type	UNSIGNED16
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	ms

Object 1A00h: Transmit PDO Mapping Parameter 1 [Axis$_n$]

The object saves the settings for the mapping of the 1st transmit PDO (TPDO1), and shows which objects are mapped in TPDO1 and transmitted with the PDO itself.
The PDO mapping for TPDO1 cannot be modified.

It is composed of subindex 0..1.

Object description	
INDEX	1A00
Name	Transmit PDO Mapping Parameter 1
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

- **Sub index 0: number of entries, Read-only**

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RO
PDO Mapping	No
Default Value	0x01
Lower Limit	0x0
Upper Limit	0xFF
Unit	-

- **Sub index 1: 1st mapping entry (fixed)**

Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x60410010
Lower Limit	0x00000000
Upper Limit	0xFFFFFFFF
Unit	-

PDO mapping of the object “statusword” (6041h)

Object 1A01h: Transmit PDO Mapping Parameter 2 [Axis$_n$]

The object saves the settings for the mapping of the 2nd transmit PDO (TPDO2), and shows which objects are mapped in TPDO2 and transmitted with the PDO itself.
The PDO mapping for TPDO2 cannot be modified. The default assignment is preset and useful for the “profile position mode”.

It is composed of subindex 0..2.

Object description	
INDEX	1A01
Name	Transmit PDO Mapping Parameter 2
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

- **Sub index 0: number of entries, Read-only**

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RO
PDO Mapping	No
Default Value	0x02
Lower Limit	0x0
Upper Limit	0xFF
Unit	-

- **Sub index 1: 1st mapping entry (fixed)**

Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x60410010
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

PDO mapping of the object “statusword” (6041h)

- **Sub index 2: 2nd mapping entry (fixed)**

Sub-Index	002
Description	Mapping Entry 2
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x60640020
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

PDO mapping of the object “position actual value in user units” (6064h).

Object 1A02h: Transmit PDO Mapping Parameter 3 [Axis$_n$]

The object saves the settings for the mapping of the 3rd transmit PDO (TPDO3), and shows which objects are mapped in TPDO3 and transmitted with the PDO itself.
The PDO mapping for TPDO3 cannot be modified. The default assignment is preset for speed mode in the “profile velocity mode”. It is composed of subindex 0..2.

Object description	
INDEX	1A02
Name	Transmit PDO Mapping Parameter 3
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

- **Sub index 0: number of entries, Read-only**

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RO
PDO Mapping	No
Default Value	0x02
Lower Limit	0x0
Upper Limit	0xFF
Unit	-
Unit	-

- **Sub index 1: 1st mapping entry (fixed)**

Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x60410010
Lower Limit	0x0
Upper Limit	0xFFFFFFF
Unit	-

PDO mapping of the object “statusword” (6041h).

- **Sub index 2: 2nd mapping entry (fixed)**

Sub-Index	002
Description	Mapping Entry 2
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x606C0020
Lower Limit	0x0
Upper Limit	0xFFFFFFF
Unit	-

PDO mapping of the object “velocity actual value” (606Ch).

Object 1A03h: Transmit PDO Mapping Parameter 4 [Axis$_n$]

The object saves the settings for the mapping of the 4th transmit PDO (TPDO4), and shows how objects can be mapped in TPDO4 and transmitted with the PDO itself.
The PDO mapping for TPDO4 can be modified (and should be written before the use). TPDO4 is for expert users only.

It is composed of subindex 0..2

Object description	
INDEX	1A03
Name	Transmit PDO Mapping Parameter 4
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

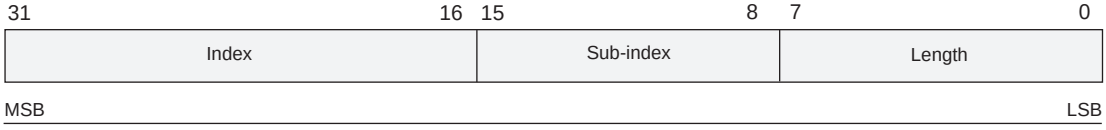
- **Sub index 0: number of entries, Read-only**

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFF
Unit	-

- **Sub index 1..4: mapping entries (fixed)**

Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x00
Upper Limit	0xFFFFFFFF
Unit	-

Here should be written the selection of objects to map, and the corresponding length.
The format is…



(Where “length” is the bit size of the mapped object: for example, 0x10 for a word like “controlword 6040h”)

The sequence to configure a new mapping for TPDO4 is:
- Deactivate the mapping by writing 0 to subindex 0;
- Select the desired objects and size writing to subindex 1 to 4 (stop when ready);

Activate the mapping by writing to subindex 0 the number of selected objects

Sub-Index	002
Description	Mapping Entry 2
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	003
Description	Mapping Entry 3
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	004
Description	Mapping Entry 4
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Object 1A40h: Transmit PDO Mapping Parameter 65 [Axis$_{n+1}$]

Please refer to Object 1A00h for the detalied description.

Object description	
INDEX	1A40
Name	Transmit PDO Mapping Parameter 65
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RO
PDO Mapping	No
Default Value	0x01
Lower Limit	0x0
Upper Limit	0xFF
Unit	-

Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x68410010
Lower Limit	0x00000000
Upper Limit	0xFFFFFFFF
Unit	-

Object 1A41h: Transmit PDO Mapping Parameter 66 [Axis$_{n+1}$]

Please refer to Object 1A01h for the detalied description.

Object description	
INDEX	1A41
Name	Transmit PDO Mapping Parameter 66
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RW
PDO Mapping	No
Default Value	0x02
Lower Limit	0x0
Upper Limit	0xFF
Unit	-

Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x68410010
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Mapping Entry 2
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x68640020
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Object 1A42h: Transmit PDO Mapping Parameter 67 [Axis$_n$]

Please refer to Object 1A02h for the detalied description..

Object description	
INDEX	1A42
Name	Transmit PDO Mapping Parameter 67
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RW
PDO Mapping	No
Default Value	0x02
Lower Limit	0x0
Upper Limit	0xFF
Unit	-

Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x68410010
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Mapping Entry 2
Entry Category	Optional
Data Type	UNSIGNED32
Access	RO
PDO Mapping	No
Default Value	0x686C0020
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Object 1A43h: Transmit PDO Mapping Parameter 68 [Axis$_n$]

Please refer to Object 1A03h for the detalied description.

Object description	
INDEX	1A43
Name	Transmit PDO Mapping Parameter 68
Object Code	Record
Data Type	PDO_MAPPING
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Mandatory
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x0
Upper Limit	0xFF
Unit	-

Sub-Index	001
Description	Mapping Entry 1
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x00
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	002
Description	Mapping Entry 2
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	003
Description	Mapping Entry 3
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Sub-Index	004
Description	Mapping Entry 4
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x00000000
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Object 3000h: Drive internal status flags [Axis$_n$]

This manufacturer-specific object can be used to read some internal device flags, used to monitor the drive operation.

Bit code object

Bit	Description		Word by WPM
0	MOTOR MOVING	1 = Motor in motion	0x1AF2
		0 = Motor in motion	
1	MEASURE READY	1 = Ready	0x1AF1
		0 = Not ready	
2	CONSOL WPM ACTIVE	1 = Escluded	0x1AF2
		0 = Active	
3	PWM CUT OFF	1 = PWM enable	0x1AF2
		0 = PWM disable	
4	PROGO1		0x1AF2
5	PROGO2		0x1AF2
6	ELECTRICAL GEAR	1 = High gear	0x1AF2
		0 = Low gear	
7	I^2 t INTERVENTION		0x1A72
8	ERROR UGA SENSOR 1		0x1A72
9	ERROR UGA SENSOR 2		0x1A72
10	FOLLOWING ERROR IU OVERFLOW		0x1A72
11	S1 ENDAT 2.2 CRC ERROR		0x1A72
12	S1 ENDAT 2.2 CRC ERROR		0x1A72
13-15	Do not used.		

Object description	
INDEX	3000
Name	Drive internal status flags [Axis n]
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RO
PDO Mapping	Yes
Default Value	0x00
Lower Limit	0x0000
Upper Limit	0xFFFF
Unit	-

Object 3001h: TPDO trigger transmission window [Axis$_n$]

This is an absolute value (not related to any physical unit) defining a threshold window to transmit the TPDO object, when programmed with transmission type = 255.

This behaviour apply to TPDO2-66 (composed by statusword+position) and TPDO3-67 (composed by statusword+speed); the window is applied to the physical value (position or speed, respective) and, of course, not to statusword changes.

Using this object (and the corresponding 3101h for the 2nd axis of the biaxis module) it is possible to define an acceptable “ripple” on the selected measure (position for TPDO2 and speed for TPDO3) that will not fire the PDO transmission: thus having a silent behaviour when the axis is not moving.

The rule used to generate the TPDO transmission (type 255) is:

- abs (New value - Old value) >= TPDO trigger transmission window : Transmission O.K
- abs (New value - Old value) < TPDO trigger transmission window : NO Transmission.

Object description	
INDEX	3001
Name	TPDO trigger transmission window
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x00000000
Upper Limit	0xFFFFFFFF
Unit	-

Object 3002h: RPDO1 shadow before SYNC validation [Axis n]

If the RPDO1 [RPDO65] are declared as “type 0” (= acyclic, synchronized) this object can be used to read back the “future” value of the command-word, sent but not yet updated by the SYNC.

If the RPDO1 [RPDO65] are not of “synchronized” types, i.e. 0xFE or 0xFF, this is simply a copy of the commandword.

Object description	
INDEX	3002
Name	RPDO1 shadow before SYNC validation [Axis n]
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RO
PDO Mapping	No
Default Value	0x00
Lower Limit	-
Upper Limit	-
Unit	-

Object 3100h: Drive internal status flags [Axis$_{n+1}$]

Please refer to Object 3000h for the detalied description.

Object description	
INDEX	3100
Name	Drive internal status flags [Axis n+1]
Object Code	Variable
Data Type	UNSIGNED8
Category	Optional

Entry description	
Access	RO
PDO Mapping	Yes
Default Value	0x00
Lower Limit	0x00
Upper Limit	0xFF
Unit	-

Object 3101h: TPDO trigger transmission window [Axis$_{n+1}$]

Please refer to Object 3001h for the detalied description.

Object description	
INDEX	3101
Name	TPDO trigger transmission window [Axis n+1]
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x00000000
Upper Limit	0xFFFFFFFF
Unit	-

Object 3102h: RPDO65 shadow before SYNC validation [Axis n+1]

If the RPDO1 [RPDO65] are declared as “type 0” (= acyclic, synchronized) this object can be used to read back the “future” value of the command-word, sent but not yet updated by the SYNC.

If the RPDO1 [RPDO65] are not of “synchronized” types, i.e. 0xFE or 0xFF, this is simply a copy of the commandword.

Object description	
INDEX	3102
Name	RPDO65 shadow before SYNC validation [Axis n+1]
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RO
PDO Mapping	No
Default Value	0x00
Lower Limit	-
Upper Limit	-
Unit	-

Object 6007h: Abort Connection Option Code [Axis$_n$]

The object Abort connection option code is used to set the reaction of the drive to a loss of connection to NMT-Master (Life-Guarding Event or Heartbeat error).

The NUMDrive C CANopen drive use a fixed behaviour to the NMT error, generating the A91 alarm. Please refer to the chapter 12 for more detailed information upon such alarm.

The reaction to this alarm is configurable using parameters V32, V33, V34.
Please refer to the chapter 5.5 for more detailed information upon such parameters.

Object description	
INDEX	6007
Name	Abort Connection Option Code [Axis n]
Object Code	Variable
Data Type	INTEGER16
Category	Optional

Entry description	
Access	RO
PDO Mapping	No
Default Value	0x1
Lower Limit	0x1
Upper Limit	0x1
Unit	-

Object 603Fh: Error Code [Axis$_n$]

This object captures the code of the last error that occured in the drive.
The value repre-sents the lower 16 bit in object Pre-defined error field (1003h).

Object description	
INDEX	603F
Name	Error Code [Axis n]
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RO
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	-

Object 6040h: Controlword [Axis$_n$]

The drive will to control by this object. This object operates the state machine and the modes.
Please refer to the chapter 5.8 for more detailed description upon the operation-specific behaviour of controlword.

- Bit 0 = Switch on
- Bit 1 = Enable voltage
- Bit 2 = Quick stop
- Bit 3 = Enable operation
- Bit 4-6 = Operation mode specific
- Bit 7 = Fault reset
- Bit 8 = Halt
- Bit 9, 10 = Reserved
- Bit 11-15 = Manufacturer specific.

Object description	
INDEX	6040
Name	Controlword [Axis n]
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RW
PDO Mapping	Yes
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	-

Object 6041h: Statusword [Axis$_n$]

This object indicates the current state of the drive.
Please refer to chapter 4.3 for a detailed description of operation-specific behaviour of statusword:

- Bit 0 = Ready to switch on
- Bit 1 = Switched on
- Bit 2 = Operation enabled
- Bit 3 = Fault
- Bit 4 = Voltage enabled
- Bit 5 = Quick stop
- Bit 6 = Switch on disabled
- Bit 7 = Warning
- Bit 8 = Manufacturer specific
- Bit 9 = Remote
- Bit 10 = Target reached
- Bit 11 = Internal limit active
- Bit 12-13 = Operation mode specific
- Bit 14-15 = Manufacturer specific.

Object description	
INDEX	6041
Name	Statusword [Axis n]
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RO
PDO Mapping	Yes
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	-

Object 605Ah: Quick Stop Option Code [Axis$_n$]

The object is used to set the reaction of the drive to a “Quick Stop” command (interruption of movement).
The state machine of the drive control branches due to this command into the status “Quick Stop active”, see object Controlword (6040h).

NUMDrive C CANopen only foreseen the reaction “Slow down on quick stop ramp and stay in Quick-Stop”; the motor is brought to the rotational speed 0 based on the rapid stop ramp, object Quick stop deceleration (6085h).
The drive controller can then be brought back into the status “Operation Enabled” by the comma nd “Enable Operation”, object Statusword (6041h).

Object description	
INDEX	605A
Name	Quick Stop Option Code [Axis n]
Object Code	Variable
Data Type	INTEGER16
Category	Optional

Entry description	
Access	RO
PDO Mapping	No
Default Value	0x6
Lower Limit	0x6
Upper Limit	0x6
Unit	-

Object 6060h: Modes of Operation [Axis$_n$]

The operating mode of the drive control is set by the object. The following operating modes are supported:

- Profile Position mode (Positioning according to CiA DSP-402)
- Profile Velocity mode (Speed mode according to CiA DSP-402)
- Homing Mode (Homing of the drive according to CiA DSP-402)
- Interpolation mode with 1-sample deep queue, SYNC and cyclic reference communication.

Profile Position mode
In the operating mode Profile Position mode (value 1) the positioning commands are processed over the fieldbus. The reference value of a positioning command is the target position of the drive movement: Object Target position (607Ah). The target position is mapped together with the control word of the status machine (controlword) in the Receive-PDO R_PDO2; see object 2nd receive PDO mapping parameter (1601h).

Profile Velocity mode
In the operating mode Profile Velocity mode (value 3) the movement commands are processed over the fieldbus. The reference value of a movement command is the set speed of the drive movement: Object Target velocity (60FFh). The set speed is mapped together with the control word of the status machine (Controlword) in the Receive-PDO R_PDO3; see object 3rd receive PDO mapping parameter (1602h).

Homing Mode
In the operating mode homing (Homing Mode, value 6) the position memory of the positioning system is initialised. Homing takes place either through value assignment or through a reference movement.

There are some limitation when changing operating mode:
- Only accepted if position RANGE and position LIMITs are well defined = the 2 interval don't cross each other;
- Homing mode is accepted only if its homing offset is within position range AND within position limits;
- Homing, PP, IP mode are accepted only if NOT sensorless;
- The mode change is performed only if the preceding operation has completed, otherwise the request is refused and the error flag "statusword.13" would be set (but the SDO request will NOT return an error in this case – because the change request is checked when the SDO protocol has already been confirmed).

Object description	
INDEX	6060
Name	Modes of Operation
Object Code	Variable
Data Type	INTEGER8
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0xFF
Lower Limit	0xFF
Upper Limit	0x7
Unit	-

Object 6061h: Modes of Operation Display [Axis$_n$]

The actual operational mode (selected by means of 6060h) can be read using this object.

Object description	
INDEX	6061
Name	Modes of Operation Display [Axis n]
Object Code	Variable
Data Type	INTEGER8
Category	Optional

Entry description	
Access	RO
PDO Mapping	Yes
Default Value	0xFF
Lower Limit	0xFF
Upper Limit	0x6
Unit	-

Object 6064h: Position Actual Value in User Unit [Axis$_n$]

TThe object indicates the current position of the drive in user defined units. See chapter 5.3.1 for description of “NC units”.

PDO-Mapping
The object 6064h is available for PDO mapping. It is also mapped by default configuration to bytes 2..5 of the TPDO2; see object 2nd transmit PDO mapping parameter (1A01h).

Object description	
INDEX	6064
Name	Position Actual Value in User Unit [Axis n]
Object Code	Variable
Data Type	INTEGER32
Category	Optional

Entry description	
Access	RO
PDO Mapping	Yes
Default Value	0x0
Lower Limit	0x80000000
Upper Limit	0x7FFFFFFF
Unit	NC unit (see also chapter 5.3.1)

Object 6065h: Following Error Window [Axis$_n$]

The following error window defines the range of tolerated position values symmetrical to the target position. If the current position is out of range a following error occurs.

Object description	
INDEX	6065
Name	Following Error Window [Axis n]
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0xFFFFFFFF
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	NC unit (see also chapter 5.3.1)

Object 6066h: Following Error Time Out

This object indicates the time for a following error condition, after that the bit 13 of the statusword shall be set to 1 in the profile position mode and in the cyclic synchronous position mode. The following error condition can generate the alarm A37 (if enabled).

Object description	
INDEX	6066
Name	Following Error Time Out
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x0014
Lower Limit	0x0002
Upper Limit	0x00C8
Unit	ms

Object 6067h: Position Window [Axis$_n$]

setpoint, object Target position (607Ah) is taken as reached and the bit Target reached in the status word, object Statusword (6041h), bit 10) is set.

The position regulator seeks independently of the size of the tolerance window to exactly approach the setpoint.

Object description	
INDEX	6067
Name	Position Window [Axis n]
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0xA
Lower Limit	0x0
Upper Limit	0x7FFFFFFF
Unit	NC unit (see also chapter 5.3.1)

Object 606Ch: Velocity Actual Value (Axes$_n$)

The object indicates the current speed of the drive. If the motor is an asynchronous spindle used in V/f mode, the unit of this object is directly RPM; in all other cases, the unit is 1000 NC units/min.

PDO-Mapping
The object 6064 is available for PDO mapping. It is also mapped by default configuration in to bytes 2..5 of the TPDO3; see object 3rd transmit PDO mapping parameter (1A02h).

Object description	
INDEX	606C
Name	Velocity Actual Value [Axis n]
Object Code	Variable
Data Type	INTEGER32
Category	Optional

Entry description	
Access	RO
PDO Mapping	Yes
Default Value	0x0
Lower Limit	0x80000000
Upper Limit	0x7FFFFFFF
Unit	non V/f mode: 1000 NC unit/min V/f mode: RPM (see also chapter 5.3.1)

Object 606Dh: Velocity Window [Axis$_n$]

The object describes the symmetrical range around the specified speed of rotation in transient response status in which the specified speed of rotation is considered as reached.

The specified speed of rotation is set in the operating mode “Profile position mode” by the object Profile velocity (6081h) and in the operating mode “Profile velocity mode” by the object Target velocity (60FFh).

Object description	
INDEX	606D
Name	Velocity Window [Axis n]
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x64
Lower Limit	0x0
Upper Limit	0x7FFF
Unit	non V/f mode: 1000 NC unit/min V/f mode: RPM (see also chapter 5.3.1)

Object 607Ah: Target Position [Axes$_n$]

The object is used to command a new position value which will be approached. The position value is set for the positioning mode according to DSP 402.

Target position
Positioning is started by bit New setpoint in the control wort of the status machine:

- Object Controlword (6040h), bit 4.

The position value is interpreted dependent on bit 6 of the control word as an absolute or relative position. Positioning can only be executed from in the status “Operation Enabled” if the positioning mode is activated.

Plausibility checking
With transfer over of the start command for positioning, the drive checks whether the given target position is within the reference operating range. If the “START POSITIONING” bit is given when the object 607A is programmed with an invalid value, the alarm A90.1 is generated.

PDO-Mapping
The object 607Ah is available for PDO mapping. It is also mapped by default configuration to bytes 2..5 of the RPDO2; see object 2nd receive PDO mapping parameter (1601h).

Object description	
INDEX	607A
Name	Target Position [Axis n]
Object Code	Variable
Data Type	INTEGER32
Category	Optional

Entry description	
Access	RW
PDO Mapping	Yes
Default Value	0x0
Lower Limit	0xC1000000
Upper Limit	0x3F000000
Unit	NC unit (see also chapter 5.3.1)

Object 607Bh: Position Range Limit [Axes$_n$]

This object describes the axis range. It determines the kind of axis in the context with “software position limit”. Please also refer to the description of object 607Dh “Software position limits”.

Object description	
INDEX	607B
Name	Position Range Limit [Axis n]
Object Code	Array
Data Type	INTEGER32
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries [Axis n]
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x2
Lower Limit	0x2
Upper Limit	0x2
Unit	-

Sub-Index	001
Description	Min Position Range Limit [Axis n]
Entry Category	Optional
Data Type	INTEGER32
Access	RW
PDO Mapping	No
Default Value	0xC1000000
Lower Limit	0xC1000000
Upper Limit	0x3F000000
Unit	NC unit (see also chapter 5.3.1)

Sub-Index	002
Description	Max Position Range Limit [Axis n]
Entry Category	Optional
Data Type	INTEGER32
Access	RW
PDO Mapping	No
Default Value	0x3F000000
Lower Limit	0xC1000000
Upper Limit	0x3F000000
Unit	NC unit (see also chapter 5.3.1)

Object 607Ch: Home Offset [Axes$_n$]

This object is the position difference between the home position and the zero position.

Object description	
INDEX	607C
Name	Home Offset [Axis n]
Object Code	Variable
Data Type	INTEGER32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0xC1000000
Upper Limit	0x3F000000
Unit	NC unit (see also chapter 5.3.1)

Object 607Dh: Software Position Limit [Axes$_n$]

The object indicates the absolute software limit positions for the operating range. The drive can be operated between these limits.The software limit positions refers to the point which was established by homing as a zero point.

Relations between "Software position limits" and "Position range limits"
The mutual position of object 607Bh and 607Dh define the behavior of the drive's measure scale. There are 3 interesting cases:

- linear axis, or rotative axis where the movement is less than one turn (for example, from 10° to 350°): in this case, the software position limit range 607Dh should be strictly internal to the position range limit 607Bh.
- Rotative axis, where the axis is allowed to move by more than one turn: in this case, the position range limit 607Bh should be strictly internal to the software position limit range 607Dh (in this way, the software position limit cannot be reached); the movement will use the shortest path (for example, incrementing by 2° when moving from 359° to 1°) and the measure will be incremented "modulo" 607Bh.
- Rotative axis, where the axis is NOT allowed to move by more than one turn and the measure should be "modularized": in this case, the position range limit 607Bh should be EQUAL to the software position limit range 607Dh (in this way, the software position limit cannot be crossed and, for example, the movement from 359° to 1° is done by turning back by 358°); and the measure will be incremented "modulo" 607Bh.

In any case, object 607Bh and 607Dh should not overlap each other.

Object description	
INDEX	607D
Name	Software Position Limit
Object Code	Array
Data Type	INTEGER32
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x2
Lower Limit	0x2
Upper Limit	0x2
Unit	-

Sub-Index	001
Description	Min Software Position Limit
Entry Category	Optional
Data Type	INTEGER32
Access	RW
PDO Mapping	No
Default Value	0xC1000000
Lower Limit	0xC1000000
Upper Limit	0x3F000000
Unit	NC unit (see also chapter 5.3.1)

Sub-Index	002

Description	Max Software Position Limit
Entry Category	Optional
Data Type	INTEGER32
Access	RW
PDO Mapping	No
Default Value	0x3F000000
Lower Limit	0xC1000000
Upper Limit	0x3F000000
Unit	NC unit (see also chapter 5.3.1)

Object 607Fh: Max Profile Velocity [Axes$_n$]

This object specifies the maximal allowed velocity used during positioning moves at the end of the acceleration ramp.

Object description	
INDEX	607F
Name	Max Profile Velocity
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0xEA60
Lower Limit	0x1
Upper Limit	0x36EE80
Unit	non V/f mode: 1000 NC unit/min V/f mode: RPM (see also chapter 5.3.1)

In NUMDrive C CANopen drives there are 2 separate objects expressing a similar value (the maximum motor speed), the ST parameter V98 and the CANopen object 607Fh (687Fh for the Axis$_{n+1}$).
They should not be confused, because their purpose is different.

The V98 is a mechanical limit for the motor, and is loaded with the motor ST; while the CANopen object 607Fh is a limit used by the CANopen interface to check against speed requests coming from CANopen commands.
In no case should be 607Fh > V98 (and the typical setting is 607Fh = V98).

The 2 values are expressed in different measure units, and this is the formula to match each other:

The formula to have the speed expressed in NCunits/min is:
NC units/min = rpm * MULTI/DIVI * (4*V360)

So, the limits are the same if:
607Fh = V98 * (V97/100) * MULTI/DIVI * (4*V360) / 1000 while the last “1000” is because the scale of 607Fh is expressed as 1000th NCunits/min.

NOTE: the real speed limit is V98 * (V97/100).

Object 6081h: Profile Velocity in pp-mode [Axis$_n$]

This object is the velocity in the profile position mode.
The object indicates the speed of travel of the drive after completing the acceleration phase.
The value should not exceed the object “Max profile velocity (607Fh)”

Object description	
INDEX	6081
Name	Profile Velocity in pp-mode
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	Yes
Default Value	0x3E8
Lower Limit	0x0
Upper Limit	0x36EE80
Unit	non V/f mode: 1000 NC unit/min V/f mode: RPM (see also chapter 5.3.1)

Object 6083h: Profile Acceleration [Axis$_n$]

This object is the linear acceleration in the profile velocity mode and the profile position mode.
If the motor is an asynchronous spindle used in V/f mode, the unit of this object is RP/s^2 (that means turns/s^2); in all other cases, the unit is 1000 NC units/s^2.

Object description	
INDEX	6083
Name	Profile Acceleration
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	Yes
Default Value	0x3E8
Lower Limit	0xA
Upper Limit	0x186A0
Unit	non V/f mode: 1000 NC unit/s^2 V/f mode: TURN/s^2 (see also chapter 5.3.1)

Object 6084h: Profile Deceleration [Axis$_n$]

This object is the linear deceleration.
Profile deceleration is given in the same units as profile acceleration. Please refer to the description of object 6083h for a description about the units used for this parameter.

Object description	
INDEX	6084
Name	Profile Deceleration [Axis n]
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	Yes
Default Value	0x3E8
Lower Limit	0xA
Upper Limit	0x186A0
Unit	non V/f mode: 1000 NC unit/s^2 V/f mode: TURN/s^2 (see also chapter 5.3.1)

Object 6085h: Quick Stop Deceleration [Axis$_n$]

The object prescribes the delay for a rapid stop (Quick stop).

Sub-index 00h
Specification of the rapid stop deceleration is in acceleration units; see object “Acceleration dimension index” (608Eh). The object is expressed in 1000 NC units/s^2.

Default value
The default value for the rapid stop deceleration represents the minimum deceleration ramp of the drive. The rapid stop deceleration cannot be adjusted.

Object description	
INDEX	6085
Name	Quick Stop Deceleration [Axis n]
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x3E8
Lower Limit	0xA
Upper Limit	0x186A0
Unit	non V/f mode: 1000 NC unit/s^2 V/f mode: TURN/s^2 (see also chapter 5.3.1)

Object 6086h: Motion Profile Type [Axis$_n$]

This object is used to select the type of the motion profile. The NUMDrive C CANopen drive only supports trapezoidal movement profiles (linear ramp, value = ”0”).

Object description	
INDEX	6086
Name	Motion Profile Type [Axis n]
Object Code	Variable
Data Type	INTEGER16
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0x0
Unit	-

Object 6098h: Homing Method [Axis$_n$]

This object determines the method that will be used during the homing mode.
-x = manufacturer specific
0 = No homing operation required
1-35 = Methods 1 to 35 (see note below)
36-127 = reserved.

Object description	
INDEX	6098
Name	Homing Method [Axis n]
Object Code	Variable
Data Type	INTEGER8
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x23
Lower Limit	0x1
Upper Limit	0x23
Unit	-

Please refer to the chapter 5.7.6 “Homing mode and homing setting” and DSP-402 for more detailed information.

Object 6099h: Homing Speeds [Axis$_n$]

This object defines a slow and a fast velocity used during the homing mode.

Object description	
INDEX	6099
Name	Homing Speeds [Axis n]
Object Code	Array
Data Type	UNSIGNED32
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries [Axis n]
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x2
Lower Limit	0x2
Upper Limit	0x2
Unit	-

Sub-Index	001
Description	Fast Homing Speed [Axis n]
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x64
Lower Limit	0x1
Upper Limit	0x57E40
Unit	non V/f mode: 1000 NC unit/min V/f mode: RPM (see also chapter 5.3.1)

Sub-Index	002
Description	Slow Homing Speed [Axis n]
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x32
Lower Limit	0x1
Upper Limit	0x8CA0
Unit	non V/f mode: 1000 NC unit/min V/f mode: RPM (see also chapter 5.3.1)

Object 609Ah: Homing Acceleration [Axis$_n$]

This object is the acceleration used in the homing mode.
Acceleration to be used for all accelerations and decelerations with the standard homing modes, in acceleration units.

Object description	
INDEX	609A
Name	Homing Acceleration [Axis n]
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x3E8
Lower Limit	0xA
Upper Limit	0x186A0
Unit	non V/f mode: 1000 NC unit/s^2 V/f mode: TURN/s^2 (see also chapter 5.3.1)

Object 60C0h: Interpolation Sub Mode Select [Axis$_n$]

This read-only object shows the interpolation modes availables.
Currently NUMDrice C CANopen only supports linear interpolation.

Object description	
INDEX	60C0
Name	Interpolation Sub Mode Select [Axis n]
Object Code	Variable
Data Type	INTEGER16
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x00
Upper Limit	0x00
Unit	-

Object 60C1h: Interpolation Data Record [Axis$_n$]

This object indicates data words, which are necessary to perform the interpolation algorithm. The number N of data words in the record is defined by interpolation data configuration.

The interpretation of the data words in interpolation data record may vary with the different possible interpolation modes as set by the interpolation_sub_mode_select.
For the linear interpolation mode each interpolation data record simply is regarded as a new position set-point.

Object description	
INDEX	60C1
Name	Interpolation Data Record [Axis n]
Object Code	Array
Data Type	INTEGER32
Category	Optional

Entry description	
Sub-Index	000
Description	Highest sub-index supported
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x1
Lower Limit	0x01
Upper Limit	0x01
Unit	-

Sub-Index	001
Description	Data Record 1
Entry Category	Optional
Data Type	INTEGER32
Access	RW
PDO Mapping	Yes
Default Value	0x00
Lower Limit	0x80000000
Upper Limit	0x7FFFFFFF
Unit	-

Object 60C2h: Interpolation Time Period [Axis$_n$]

This object indicates the interpolation cycle time.
This object includes the following sub-indices:

- sub-index 1: value of the time
- sub-index 2: dimension index to the time value in sub-index 1.

Object description	
INDEX	60C2
Name	Interpolation Time Period [Axis n]
Object Code	Record
Data Type	P402_IP_PERIOD_T
Category	Optional

Entry description	
Sub-Index	000
Description	Highest sub-index supported
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x02
Lower Limit	0x02
Upper Limit	0x02
Unit	-

Sub-Index	001
Description	Interpolation time period value
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0x2
Lower Limit	0x2
Upper Limit	0x14
Unit	(10^interpolation_time_index) s

Sub-Index	002
Description	Interpolation time index
Entry Category	Optional
Data Type	INTEGER8
Access	RW
PDO Mapping	No
Default Value	0xFD
Lower Limit	0xFD
Upper Limit	0xFD
Unit	-

Object 60F2h: Positioning Option Code

This object indicates the positioning behaviour as described by the profile position mode or the interpolated position mode.

This object is organized bit-wise. The bits have the following meaning:

- bit 0,1: relative option
- bit 2-15: not available.

Object description	
INDEX	60F2
Name	Positioning Option Code
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x0000
Upper Limit	0x0002
Unit	-

Object 60FDh: Digital Inputs [Axis$_n$]

This object reflects the current state of the digital input signals. The user may apply any signals to these inputs for special purposes like limit or reference switches.

Bit	Description
0	Negative limit switch
1	Positive limit switch
2	HOME position switch

Object description	
INDEX	60FD
Name	Digital Inputs [Axis n]
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RO
PDO Mapping	Yes
Default Value	0x00
Lower Limit	0x00000000
Upper Limit	0xFFFFFFFF
Unit	-

Object 60FFh: Target Velocity [Axis$_n$]

The object is used to enter the required speed setpoint for a drive movement in the operating mode “Profile velocity mode”.

Setpoint speed
A drive movement in Profile Velocity mode is started by deleting the bit Halt (value 1 -> 0) in the control word of the status machine: Object Controlword (6040h), bit 8.
Drive movements can only be executed from the status “Operation Enabled” if the speed mode (Profile Velocity mode) is activated.

PDO-Mapping
The object 60FFh is available for PDO mapping. It is also mapped by default configuration to bytes 2..5 of the RPDO3; see object 3rd receive PDO mapping parameter (1602h).

Object description	
INDEX	60FF
Name	Target Velocity
Object Code	Variable
Data Type	INTEGER32
Category	Optional

Entry description	
Access	RW
PDO Mapping	Yes
Default Value	0x3E8
Lower Limit	0xFFC91180
Upper Limit	0x36EE80
Unit	non V/f mode: 1000 NC unit/min V/f mode: RPM (see also chapter 5.3.1)

Object 6502h: Supported Drive Modes [Axis$_n$]

This object documents all operational modes which are supported by the drive.

- Bit 31-16 = manufacturer specific
- Bit 15-07 = reserved
- Bit 6 = ip
- Bit 5 = hm
- Bit 4 = reserved
- Bit 3 = tq
- Bit 2 = pv
- Bit 1 = vl
- Bit 0 = pp.

Object description	
INDEX	6502
Name	Supported Drive Modes [Axis n]
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RO
PDO Mapping	No
Default Value	0x65
Lower Limit	0x65
Upper Limit	0x65
Unit	-

Object 67FFh: Single Device Type [Axis$_n$]

This object describes the type of each device within one drive unit and its functionality.

- Bits 31-24: xx = Manufacturer specific
  0F = Multiple devide module
- Bits 23-16: 01 = Frequency converter
  02 = Servo drive
  04 = Stepper motor
  FF = Multiple device module
- Bits 15-00: 0192 = DSP 402 device.

Object description	
INDEX	67FF
Name	Single Device Type [Axis n]
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RO
PDO Mapping	No
Default Value	0x00200192
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

Object 6807h: Abort Connection Option Code [Axis$_{n+1}$]

This object selects the function to be performed when the connection to the network is lost.

- 0 = No action
- 1 = Malfunction
- 2 = Device control command ‘Disable Voltage’
- 3 = Device control command ‘Quick Stop’
- -x = Manufacturer specific.

Object description	
INDEX	6807
Name	Abort Connection Option Code [Axis n+1]
Object Code	Variable
Data Type	INTEGER16
Category	Optional

Entry description	
Access	RO
PDO Mapping	No
Default Value	0x1
Lower Limit	0x1
Upper Limit	0x1
Unit	-

Object 683Fh: Error Code [Axis$_{n+1}$]

This object captures the code of the last error that occured in the drive.

Object description	
INDEX	683F
Name	Error Code [Axis n+1]
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RO
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	-

Object 6840h: Controlword [Axis n+1]

Please refer to Object 6040h for the detalied description.

Object description	
INDEX	6840
Name	Controlword [Axis n+1]
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RW
PDO Mapping	Yes
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	-

Object 6841h: Statusword [Axis$_{n+1}$]

Please refer to Object 6041h for the detalied description.

Object description	
INDEX	6841
Name	Statusword [Axis n+1]
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RO
PDO Mapping	Yes
Default Value	0x0
Lower Limit	0x0
Upper Limit	0xFFFF
Unit	-

Object 685Ah: Quick Stop Option Code [Axis$_{n+1}$]

Please refer to Object 605Ah for the detalied description.

Object description	
INDEX	685A
Name	Quick Stop Option Code [Axis n+1]
Object Code	Variable
Data Type	INTEGER16
Category	Optional

Entry description	
Access	RO
PDO Mapping	No
Default Value	0x6
Lower Limit	0x6
Upper Limit	0x6
Unit	-

Object 6860h: Modes of Operation [Axis$_{n+1}$]

Please refer to Object 6060h for the detalied description.

Object description	
INDEX	6860
Name	Modes of Operation [Axis n+1]
Object Code	Variable
Data Type	INTEGER8
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0xFF
Lower Limit	0xFF
Upper Limit	0x7
Unit	-

Object 6861h: Modes of Operation Display [Axis$_{n+1}$]

Please refer to Object 6061h for the detalied description.

Object description	
INDEX	6861
Name	Modes of Operation Display [Axis n+1]
Object Code	Variable
Data Type	INTEGER8
Category	Optional

Entry description	
Access	RO
PDO Mapping	Yes
Default Value	0xFF
Lower Limit	0xFF
Upper Limit	0x6
Unit	-

Object 6864h: Position Actual Value in User Unit [Axis$_{n+1}$]

Please refer to Object 6064h for the detalied description.

Object description	
INDEX	6864
Name	Position Actual Value in User Unit [Axis n+1]
Object Code	Variable
Data Type	INTEGER32
Category	Optional

Entry description	
Access	RO
PDO Mapping	Yes
Default Value	0x0
Lower Limit	0x80000000
Upper Limit	0x7FFFFFFF
Unit	NC unit (see also chapter 5.3.1)

Object 6865h: Following Error Window [Axis$_{n+1}$]

Please refer to Object 6065h for the detalied description.

Object description	
INDEX	6865
Name	Following Error Window [Axis n+1]
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0xFFFFFFFF
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	NC unit (see also chapter 5.3.1)

Object 6866h: Following Error Time Out

Please refer to Object 6066h for the detalied description.

Object description	
INDEX	6866
Name	Following Error Time Out
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x0014
Lower Limit	0x0002
Upper Limit	0x00C8
Unit	ms

Object 6867h: Position Window [Axis$_{n+1}$]

Please refer to Object 6067h for the detalied description.

Object description	
INDEX	6867
Name	Position Window [Axis n+1]
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0xA
Lower Limit	0x0
Upper Limit	0x7FFFFFFF
Unit	NC unit (see also chapter 5.3.1)

Object 686Ch: Velocity Actual Value [Axis$_{n+1}$]

Please refer to Object 606Ch for the detalied description.

Object description	
INDEX	686C
Name	Velocity Actual Value [Axis n+1]
Object Code	Variable
Data Type	INTEGER32
Category	Optional

Entry description	
Access	RO
PDO Mapping	Yes
Default Value	0x0
Lower Limit	0x80000000
Upper Limit	0x7FFFFFFF
Unit	non V/f mode: 1000 NC unit/min V/f mode: RPM (see also chapter 5.3.1)

Object 686Dh: Velocity Window [Axis$_{n+1}$]

Please refer to Object 606Dh for the detalied description.

Object description	
INDEX	686D
Name	Velocity Window [Axis n+1]
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x64
Lower Limit	0x0
Upper Limit	0x7FFF
Unit	non V/f mode: 1000 NC unit/min V/f mode: RPM (see also chapter 5.3.1)

Object 687Ah: Target Position [Axis$_{n+1}$]

Please refer to Object 607Ah for the detalied description.

Object description	
INDEX	687A
Name	Target Position [Axis n+1]
Object Code	Variable
Data Type	INTEGER32
Category	Optional

Entry description	
Access	RW
PDO Mapping	Yes
Default Value	0x0
Lower Limit	0xC1000000
Upper Limit	0x3F000000
Unit	NC unit (see also chapter 5.3.1)

Object 687Bh: Position Range Limit [Axis$_{n+1}$]

Please refer to Object 607Bh for the detalied description.

Object description	
INDEX	687B
Name	Position Range Limit [Axis n+1]
Object Code	Array
Data Type	INTEGER32
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries [Axis n+1]
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x2
Lower Limit	0x2
Upper Limit	0x2
Unit	-

Sub-Index	001
Description	Min Position Range Limit [Axis n+1]
Entry Category	Optional
Data Type	INTEGER32
Access	RW
PDO Mapping	No
Default Value	0xC1000000
Lower Limit	0xC1000000
Upper Limit	0x3F000000
Unit	NC unit (see also chapter 5.3.1)

Sub-Index	002
Description	Max Position Range Limit [Axis n+1]
Entry Category	Optional
Data Type	INTEGER32
Access	RW
PDO Mapping	No
Default Value	0x3F000000
Lower Limit	0xC1000000
Upper Limit	0x3F000000
Unit	NC unit (see also chapter 5.3.1)

Object 687Ch: Home Offset [Axis$_{n+1}$]

Please refer to Object 607Ch for the detalied description.

Object description	
INDEX	687C
Name	Home Offset [Axis n+1]
Object Code	Variable
Data Type	INTEGER32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0xC1000000
Upper Limit	0x3F000000
Unit	NC unit (see also chapter 5.3.1)

Object 687Dh: Software Position Limit [Axis$_{n+1}$]

Please refer to Object 607Dh for the detalied description.

Object description	
INDEX	687D
Name	Software Position Limit [Axis n+1]
Object Code	Array
Data Type	INTEGER32
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries [Axis n+1]
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x2
Lower Limit	0x2
Upper Limit	0x2
Unit	-

Sub-Index	001
Description	Min Software Position Limit [Axis n+1]
Entry Category	Optional
Data Type	INTEGER32
Access	RW
PDO Mapping	No
Default Value	0xC1000000
Lower Limit	0xC1000000
Upper Limit	0x3F000000
Unit	NC unit (see also chapter 5.3.1)

Sub-Index	002
Description	Max Software Position Limit [Axis n+1]
Entry Category	Optional
Data Type	INTEGER32
Access	RW
PDO Mapping	No
Default Value	0x3F000000
Lower Limit	0xC1000000
Upper Limit	0x3F000000
Unit	NC unit (see also chapter 5.3.1)

Object 687Fh: Max Profile Velocity [Axis$_{n+1}$]

Please refer to Object 607Fh for the detalied description.

Object description	
INDEX	687F
Name	Max Profile Velocity [Axis n+1]
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0xEA60
Lower Limit	0x1
Upper Limit	0x36EE80
Unit	non V/f mode: 1000 NC unit/min V/f mode: RPM (see also chapter 5.3.1)

Object 6881h: Profile Velocity in pp-mode [Axis$_{n+1}$]

Please refer to Object 6081h for the detalied description.

Object description	
INDEX	6881
Name	Profile Velocity in pp-mode [Axis n+1]
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	Yes
Default Value	0x3E8
Lower Limit	0x0
Upper Limit	0x36EE80
Unit	non V/f mode: 1000 NC unit/min V/f mode: RPM (see also chapter 5.3.1)

Object 6883h: Profile Acceleration [Axis$_{n+1}$]

Please refer to Object 6083h for the detalied description.

Object description	
INDEX	6883
Name	Profile Acceleration [Axis n+1]
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	Yes
Default Value	0x3E8
Lower Limit	0xA
Upper Limit	0x186A0
Unit	non V/f mode: 1000 NC unit/s^2 V/f mode: TURN/s^2 (see also chapter 5.3.1)

Object 6884h: Profile Deceleration [Axis$_{n+1}$]

Please refer to Object 6084h for the detalied description.

Object description	
INDEX	6884
Name	Profile Deceleration [Axis n+1]
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	Yes
Default Value	0x3E8
Lower Limit	0xA
Upper Limit	0x186A0
Unit	non V/f mode: 1000 NC unit/s^2 V/f mode: TURN/s^2 (see also chapter 5.3.1)

Object 6885h: Quick Stop Deceleration [Axis$_{n+1}$]

Please refer to Object 6085h for the detalied description.

Object description	
INDEX	6885
Name	Quick Stop Deceleration [Axis n+1]
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x3E8
Lower Limit	0xA
Upper Limit	0x186A0
Unit	non V/f mode: 1000 NC unit/s^2 V/f mode: TURN/s^2 (see also chapter 5.3.1)

Object 6886h: Motion Profile Type [Axis$_{n+1}$]

Please refer to Object 6086h for the detalied description.

Object description	
INDEX	6886
Name	Motion Profile Type [Axis n+1]
Object Code	Variable
Data Type	INTEGER16
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x0
Lower Limit	0x0
Upper Limit	0x0
Unit	-

Object 6898h: Homing Method [Axis$_{n+1}$]

Please refer to Object 6098h for the detalied description.

Object description	
INDEX	6898
Name	Homing Method [Axis n+1]
Object Code	Variable
Data Type	INTEGER8
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x23
Lower Limit	0x1
Upper Limit	0x23
Unit	-

Object 6899h: Homing Speeds [Axis$_{n+1}$]

Please refer to Object 6099h for the detalied description.

Object description	
INDEX	6899
Name	Homing Speeds [Axis n+1]
Object Code	Array
Data Type	UNSIGNED32
Category	Optional

Entry description	
Sub-Index	000
Description	Number of Entries [Axis n+1]
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x2
Lower Limit	0x2
Upper Limit	0x2
Unit	-

Sub-Index	001
Description	Fast Homing Speed [Axis n+1]
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x64
Lower Limit	0x1
Upper Limit	0x57E40
Unit	non V/f mode: 1000 NC unit/min V/f mode: RPM (see also chapter 5.3.1)

Sub-Index	002
Description	Slow Homing Speed [Axis n+1]
Entry Category	Optional
Data Type	UNSIGNED32
Access	RW
PDO Mapping	No
Default Value	0x32
Lower Limit	0x1
Upper Limit	0x8CA0
Unit	non V/f mode: 1000 NC unit/min V/f mode: RPM (see also chapter 5.3.1)

Object 689Ah: Homing Acceleration [Axis$_{n+1}$]

Please refer to Object 609Ah for the detalied description.

Object description	
INDEX	689A
Name	Homing Acceleration [Axis n+1]
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x3E8
Lower Limit	0xA
Upper Limit	0x186A0
Unit	non V/f mode: 1000 NC unit/s^2 V/f mode: TURN/s^2 (see also chapter 5.3.1)

Object 68C0h: Interpolation Sub Mode Select [Axis$_{n+1}$]

Please refer to Object 60C0h for the detalied description.

Object description	
INDEX	68C0
Name	Interpolation Sub Mode Select [Axis n+1]
Object Code	Variable
Data Type	INTEGER16
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x00
Upper Limit	0x00
Unit	-

Object 68C1h: Interpolation Data Record [Axis$_{n+1}$]

Please refer to Object 60C1h for the detalied description.

Object description	
INDEX	68C1
Name	Interpolation Data Record [Axis n+1]
Object Code	Array
Data Type	INTEGER32
Category	Optional

Entry description	
Sub-Index	000
Description	Highest sub-index supported
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x1
Lower Limit	0x01
Upper Limit	0x01
Unit	-

Sub-Index	001
Description	Data Record 1
Entry Category	Optional
Data Type	INTEGER32
Access	RW
PDO Mapping	Yes
Default Value	0x00
Lower Limit	0x80000000
Upper Limit	0x7FFFFFFF
Unit	-

Object 68C2h: Interpolation Time Period [Axis$_{n+1}$]

Please refer to Object 60C2h for the detalied description.

Object description	
INDEX	68C2
Name	Interpolation Time Period [Axis n+1]
Object Code	Record
Data Type	P402_IP_PERIOD_T
Category	Optional

Entry description	
Sub-Index	000
Description	Highest sub-index supported
Entry Category	Optional
Access	RO
PDO Mapping	No
Default Value	0x02
Lower Limit	0x02
Upper Limit	0x02
Unit	-

Sub-Index	001
Description	Interpolation time period value
Entry Category	Optional
Data Type	UNSIGNED8
Access	RW
PDO Mapping	No
Default Value	0x2
Lower Limit	0x2
Upper Limit	0x14
Unit	(10^interpolation_time_index)

Sub-Index	002
Description	Interpolation time index
Entry Category	Optional
Data Type	INTEGER8
Access	RW
PDO Mapping	No
Default Value	0xFD
Lower Limit	0xFD
Upper Limit	0xFD
Unit	-

Object 68F2h: Positioning Option Code

This object indicates the positioning behaviour as described by the profile position mode or the interpolated position mode.

This object is organized bit-wise. The bits have the following meaning:

- bit 0,1: relative option
- bit 2-15: not available.

Object description	
INDEX	68F2
Name	Positioning Option Code
Object Code	Variable
Data Type	UNSIGNED16
Category	Optional

Entry description	
Access	RW
PDO Mapping	No
Default Value	0x00
Lower Limit	0x0000
Upper Limit	0x0002
Unit	-

Object 68FDh: Digital Inputs [Axis$_{n+1}$]

Please refer to Object 60FDh for the detalied description.

Object description	
INDEX	68FD
Name	Digital Inputs [Axis n+1]
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RO
PDO Mapping	Yes
Default Value	0x00
Lower Limit	0x00000000
Upper Limit	0xFFFFFFFF
Unit	-

Object 68FFh: Target Velocity [Axis$_{n+1}$]

Please refer to Object 60FFh for the detalied description.

Object description	
INDEX	68FF
Name	Target Velocity [Axis n+1]
Object Code	Variable
Data Type	INTEGER32
Category	Optional

Entry description	
Access	RW
PDO Mapping	Yes
Default Value	0x3E8
Lower Limit	0xFFC91180
Upper Limit	0x36EE80
Unit	non V/f mode: 1000 NC unit/min V/f mode: RPM (see also chapter 5.3.1)

Object 6D02h: Supported Drive Modes [Axis$_{n+1}$]

Please refer to Object 6502h for the detalied description.

Object description	
INDEX	6D02
Name	Supported Drive Modes [Axis n+1]
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RO
PDO Mapping	No
Default Value	0x65
Lower Limit	0x65
Upper Limit	0x65
Unit	-

Object 6FFFh: Single Device Type [Axis$_{n+1}$]

Please refer to Object 67FFh for the detalied description.

Object description	
INDEX	6FFF
Name	Single Device Type [Axis n+1]
Object Code	Variable
Data Type	UNSIGNED32
Category	Optional

Entry description	
Access	RO
PDO Mapping	No
Default Value	0x00200192
Lower Limit	0x0
Upper Limit	0xFFFFFFFF
Unit	-

5.3.3 Object dictionary access rights

Not all objects can be written at any time. Please refer to the summary as indicated on the below.

Index	Name	Access notes
1000	Device Type	Read Only
1001	Error Register	Read Only
1003	Pre-defined Error Field	Read Only
1005	COB ID SYNC	PRE-OP and SwitchOnDisabled
1006	Communication Cycle Period	PRE-OP and SwitchOnDisabled
100C	Guard Time	Read / Write
100D	Life Time Factor	Read / Write
1014	COB ID EMCY	Read / Write
1015	Inhibit Time Emergency	Read / Write
1016	Heartbeat Consumer Entries	PRE-OP, before the first heartbeat
1017	Producer Heartbeat Time	PRE-OP, before the first heartbeat
1018	Identity Object	Read / Write
1020	Verify Configuration	Read / Write
1200	Server SDO Parameter 1	Read / Write
1201	Server SDO Parameter 2	Read / Write
1400	Receive PDO Communication Parameter 1	Only when comm. FSM is in PRE-OPERATIONAL
1401	Receive PDO Communication Parameter 2	Only when comm. FSM is in PRE-OPERATIONAL
1402	Receive PDO Communication Parameter 3	Only when comm. FSM is in PRE-OPERATIONAL
1403	Receive PDO Communication Parameter 4	Only when comm. FSM is in PRE-OPERATIONAL
1440	Receive PDO Communication Parameter 65	Only when comm. FSM is in PRE-OPERATIONAL
1441	Receive PDO Communication Parameter 66	Only when comm. FSM is in PRE-OPERATIONAL
1442	Receive PDO Communication Parameter 67	Only when comm. FSM is in PRE-OPERATIONAL
1443	Receive PDO Communication Parameter 68	Only when comm. FSM is in PRE-OPERATIONAL
1600	Receive PDO Mapping Parameter 1	Only when comm. FSM is in PRE-OPERATIONAL
1601	Receive PDO Mapping Parameter 2	Only when comm. FSM is in PRE-OPERATIONAL
1602	Receive PDO Mapping Parameter 3	Only when comm. FSM is in PRE-OPERATIONAL
1603	Receive PDO Mapping Parameter 4	Only when comm. FSM is in PRE-OPERATIONAL
1640	Receive PDO Mapping Parameter 65	Only when comm. FSM is in PRE-OPERATIONAL
1641	Receive PDO Mapping Parameter 66	Only when comm. FSM is in PRE-OPERATIONAL
1642	Receive PDO Mapping Parameter 67	Only when comm. FSM is in PRE-OPERATIONAL
1643	Receive PDO Mapping Parameter 68	Only when comm. FSM is in PRE-OPERATIONAL
1800	Transmit PDO Communication Parameter 1	Only when comm. FSM is in PRE-OPERATIONAL
1801	Transmit PDO Communication Parameter 2	Only when comm. FSM is in PRE-OPERATIONAL
1802	Transmit PDO Communication Parameter 3	Only when comm. FSM is in PRE-OPERATIONAL
1803	Transmit PDO Communication Parameter 4	Only when comm. FSM is in PRE-OPERATIONAL
1840	Transmit PDO Communication Parameter 65	Only when comm. FSM is in PRE-OPERATIONAL
1841	Transmit PDO Communication Parameter 66	Only when comm. FSM is in PRE-OPERATIONAL
1842	Transmit PDO Communication Parameter 67	Only when comm. FSM is in PRE-OPERATIONAL
1843	Transmit PDO Communication Parameter 68	Only when comm. FSM is in PRE-OPERATIONAL
1A00	Transmit PDO Mapping Parameter 1	Only when comm. FSM is in PRE-OPERATIONAL
1A01	Transmit PDO Mapping Parameter 2	Only when comm. FSM is in PRE-OPERATIONAL
1A02	Transmit PDO Mapping Parameter 3	Only when comm. FSM is in PRE-OPERATIONAL
1A03	Transmit PDO Mapping Parameter 4	Only when comm. FSM is in PRE-OPERATIONAL

1A40	Transmit PDO Mapping Parameter 65	Only when comm. FSM is in PRE-OPERATIONAL
1A41	Transmit PDO Mapping Parameter 66	Only when comm. FSM is in PRE-OPERATIONAL
1A42	Transmit PDO Mapping Parameter 67	Only when comm. FSM is in PRE-OPERATIONAL
1A43	Transmit PDO Mapping Parameter 68	Only when comm. FSM is in PRE-OPERATIONAL
6007	Abort Connection Option Code [Axis^{n}]	Read Only[1]
603F	Error Code [Axis$_{n}$]	Read Only
6040	Controlword [Axis$_{n}$]	Read / Write
6041	Statusword [Axis$_{n}$]	Read Only
605A	Quick Stop Option Code [Axis$_{n}$]	Read Only[1]
6060	Modes of Operation [Axis$_{n}$]	• Only accepted if position RANGE and position LIMITs are well defined = the 2 interval do not cross each other. • Homing mode is accepted only if its homing offset is within position range AND within position limits. • Homing, PP, IP mode are accepted only if NOT sensorless. • The mode change is performed only if the preceding operation has completed, otherwise the request is refused and the error flag “statusword.13” would be set (but the SDO request will NOT return an error in this case – because the change request is checked when the SDO protocol has already been confirmed).
6061	Modes of Operation Display [Axis$_{n}$]	Read Only
6064	Position Actual Value in User Unit [Axis$_{n}$]	Read Only
6065	Following Error Window [Axis$_{n}$]	Read / Write
6066	Following Error time out [Axis$_{n}$]	Read / Write
6067	Position Window [Axis$_{n}$]	Read / Write
606C	Velocity Actual Value [Axis$_{n}$]	Read Only
606D	Velocity Window [Axis$_{n}$]	Read / Write
607A	Target Position [Axis$_{n}$]	Everytime, used on START bit edge
607B	Position Range Limit [Axis$_{n}$]	• Only changeable if in SWITCH-ON DISABLED or FAULT state. • Obviously it accepts only INF < SUP.
607C	Home Offset [Axis$_{n}$]	• Only changeable if in SWITCH-ON DISABLED or FAULT state. • Accepted only if it is within position range AND within position limits.
607D	Software Position Limit [Axis$_{n}$]	• Only changeable if in SWITCH-ON DISABLED or FAULT state. • Obviously it accepts only INF < SUP.
607F	Max Profile Velocity [Axis$_{n}$]	Read / Write
6081	Profile Velocity in pp mode [Axis$_{n}$]	Read / Write
6083	Profile Acceleration [Axis$_{n}$]	Read / Write
6084	Profile Deceleration [Axis$_{n}$]	Read / Write
6085	Quick Stop Deceleration [Axis$_{n}$]	Read / Write
6086	Motion Profile Type [Axis$_{n}$]	Read / Write
6098	Homing Method [Axis$_{n}$]	• Can be changed only if “another” homing operation is not running. Otherwise the SDO request will be refused.
6099	Homing Speeds [Axis$_{n}$]	Read / Write
609A	Homing Acceleration [Axis$_{n}$]	Read / Write
60C0	Interpolation Sub Mode Select	Read / Write
60C1	Interpolation Data Record	Read / Write

60C2	Interpolation Time Period	• Can be changed only when IP mode is not selected. When the IP mode is selected and the drive is in OPERATION ENABLED it must match V266 otherwise an alarm A16.266 will be generated.
60FD	Digital Inputs [Axis$_n$]	Read Only
60FF	Target Velocity [Axis$_n$]	Read / Write
6502	Supported Drive Modes [Axis$_n$]	Read Only
67FF	Single Device Type [Axis$_n$]	Read Only
6800.. 6FFF	Same settings as for 6000h..67FFh	Read / Write

[1] Here RO means for NUM CANopen drive this object can assume only one value.
If the CANopen specification forces to declare this object as RW we have declared MIN value = DEFAULT value = MAX value.

5.4 NUMDrive C general parameters list

N°	Parameters description	Unit	Motor utilization		
V01	CAN configuration: • Bit 0 = CAN baud. [000=50Kb] [001=100Kb] [010=125Kb] • Bit 1 = CAN baud. [011=250Kb] [100=500Kb] [101=1Mbit] • Bit 2 = CAN baud. [110=n/a] [111=n/a]	Bit	SM	AM	LM
V20*	Sensor connectors X3, X4 configuration: • Bit 0 = Sensor connectors X3, X4 [0=Absent] [1=Plugged-in] • Bit 1 = [0=> X3=S1(n+1) X4=S2(n+1)] [1=> X3=S2(n) X4=S2(n+1)]	Bit	SM	AM	LM
V21*	TP1/2 output configuration: • Bit 0 = Output S9.3 [0=(TP1 axis n)] [1=(TP1 axis n+1)] • Bit 1 = Output S9.1 [0=(TP2 axis n)] [1=(TP2 axis n+1)]				
V22*	S7.2/3 I/O configuration: • Bit 0 = I/O S7.2 [0=axis n] [1=axis n+1] • Bit 1 = I/O S7.3 [0=axis n] [1=axis n+1] • Bit 2 = S7.2 [0=out PROGO1] [1=in GP_IN_C] • Bit 3 = S7.3 [0=out PROGO2] [1=in GP_IN_D]				
V23*	Do not use				
V30	General Drive configuration : • Bit 0 = Wait for motor to stop to enable [1 = Wait] • Bit 3 = Exclusion of change of range if required [1 = Exclude] • Bit 4 = Forcing of electrical range [0 = Low] / [1 = High] • Bit 5 = Following error A37.0 exclusion [1=Excluded] • Bit 6 = I^2t handled as an alarm A49.0 [0=warning 1=alarm] • Bit 7 = MP0011 (LOAD) = 100% with load [0=rated 1=max] • Bit 10 = Short circuit to ground A06 exclusion [1=Excluded] • Bit 11 = Enabling external HW alarm Reset [1=enabled] • Bit 12 = UGA misalignment handling [0=Warning] [1=Alarm] • Bit 13 = Limited torque when reference disabled [1=ON] • Bit 16 = Brake failure A54 exclusion [1=Excluded] • Bit 18 = Integration of NUM-STO [0=No required] [1=Required] • Bit 24 = Alarm A60.268 [1=Enabled] • Bit 26 = Alarm A09/A38 ext.231 exclusion [1=Excluded] • Bit 27 = Alarm A03 exclusion [1=Excluded] • Bit 28 = Alarm A53.22 exclusion [1=Excluded] • Bit 30 = Drive parking [1=Enabled]	Bit	SM	AM	LM
V32	Configuration Braking following alarms [1=Brake] : • Bit 0 = Motor overtemperature (A03) • Bit 1 = Module overtemperature (A04) • Bit 2 = Brake resistor overtemperature (A05) • Bit 3 = Local link alarm (A08) • Bit 4 = Sensor alarm S1 (A09) • Bit 5 = Overspeed (A11) • Bit 6 = Self-calibration (A18) • Bit 7 = Following error (A37) • Bit 8 = Sensor alarm S2 (A38) • Bit 9 = Alarm I^2t (A49) • Bit 10 = Position coherence S1-S2 (A51) • Bit 11 = Tandem function alarm (A.52) • Bit 12 = Trajectory generator A90 • Bit 13 = CANopen communication alarm A91	Bit	SM	AM	LM

* Bi-Axes configutation only

N°	Parameters description	Unit	Motor utilization		
V33	Alarm Reset Configuration [1=Reset] • Bit 0 = Motor overtemperature (A03) • Bit 1 = Module overtemperature (A04) • Bit 2 = Brake Resistor overtemperature (A05) • Bit 3 = Short-circuit to earth (A06) • Bit 4 = Auxiliary Power Supply alarm (A07) • Bit 5 = Local link alarm (A08) • Bit 6 = S1 sensor alarm (A09) • Bit 7 = Timeout (A10) • Bit 8 = Overspeed (A11) • Bit 9 = Drive overcurrent (A12) • Bit 10 = Incompatible parameters (A16) • Bit 11 = Self-calibration alarm (A18) • Bit 12 = Following error (A37) • Bit 13 = S2 sensor alarm (A38) • Bit 14 = Alarm I^2t (A49) • Bit 15 = Position coherence S1-S2 (A51) • Bit 16 = Tandem function alarm (A52) • Bit 17 = Safety Module failure class 0 (A60) • Bit 21 = Brake failure (A54) • Bit 22 = Incompatible sensors (A55) • Bit 23 = V/f alarm (A56) • Bit 24 = Trajectory generator (A90) • Bit 25 = CANopen communication alarm (A91)	Bit	SM	AM	LM
V34	Configuration DF1 / DF2 relay [1 = break] • Bit 0 = BUS overvoltage (A02) • Bit 1 = Motor overtemperature (A03) • Bit 2 = Module overtemperature (A04) • Bit 3 = Brake Resistor overtemperature (A05) • Bit 4 = Short-circuit to earth (A06) • Bit 5 = Auxiliary power supply alarm (A07) • Bit 6 = Local link alarm (A08) • Bit 7 = S1 sensor alarm (A09) • Bit 8 = Timeout (A10) • Bit 9 = Overspeed (A11) • Bit 10 = Drive overcurrent (A12) • Bit 11 = Parameter Checksum Error (A15) • Bit 12 = Incompatible parameters (A16) • Bit 13 = Self-calibration (A18) • Bit 14 = Following error (A37) • Bit 15 = S2 sensor alarm (A38) • Bit 16 = Alarm I^2t (A49) • Bit 17 = Position coherence S1-S2 (A51) • Bit 18 = Tandem function alarm (A52) • Bit 19 = CANopen communication alarm (A91)	Bit	SM	AM	LM
V35	SW functions [0=not available] [1=available] • Bit 0 = Do not use • Bit 1 = Do not use • Bit 2 = Do not use • Bit 3 = Do not use • Bit 5 = Do not use • Bit 7 = Do not use • Bit 9 = Do not use • Bit 10 = Do not use • Bit 11 = S2 Sensor • Bit 12 = Linear Motor	Bit	SM	AM	LM
V37	Delay time of power detectionr	Reserved parameter			
V41	Motor zero speed threshold, enable, brake (Linear motor)	mm/ min	•	•	LM
V42	Motor zero speed threshold, enable, brake	Rpm	SM	AM	•

N°	Parameters description	Unit	Motor utilization		
V43	Braking Timeout on configuration V30, V32	s	SM	AM	LM
V44	Braking Current on configutation V30, V32	%	SM	AM	LM
V50	Torque current limit [0]	Arms	SM	AM	•
V51	Proportional speed loop gain [0]	mA/ Rpm	SM	AM	•
V52	Integral speed loop gain [0]	ms	SM	AM	LM
V53	Do not use				
V57	Torque current limit h. r. [0]	Arms	•	AM	•
V58	Proportional speed loop gain h. r. [0]	mA/ Rpm	•	AM	•
V59	Integral speed loop gain h. r. [0]	ms	•	AM	•
V60	Proportional speed loop gain [0]	Arms m/min	•	•	LM
V75	Do not use				
V80	Set Point Weight speed loop gain	[ ]	SM	AM	LM
V81	Anti-windup speed loop gain	[ ]	SM	AM	LM
V94	Speed reference Limit [% maximum speed] [0]	%	•	•	LM
V95	Maximum motor speed [0]	m/min	•	•	LM
V96	Threshold for overspeed alarm [0]	m/min	•	•	LM
V97	Speed reference limit [% maximum speed] [0]	%	SM	AM	•
V98	Maximum motor speed [0]	Rpm	SM	AM	•
V99	Overspeed alarm threshold [0]	Rpm	SM	AM	•
V100	Module current limit	Reserved parameters			
V101	Voltage limits Vd and Vq				
V102	Proportional current loop gain				
V103	Integral current loop gain				
V104	Motor EMF phase compensation				
V105	Compensation of motor Lq inductance				
V106	Compensation of motor Ld inductance				
V107	Lowpass Filter elettrical speed				
V108	Rated motor current [I^2t] h.r.	Arms	SM	AM	LM
V109	Vertical load compensation current	Arms	SM	AM	LM
V110	Limit current of module h.r.	Reserved parameters			
V111	Proportional current gain h.r.	Reserved parameters			
V112	Integral current gain h.r.	Reserved parameters			
V115	Rated motor current [i2t] h. r.	Arms	•	AM	•
V116	Lm (for compensation) low range	Reserved parameters			

N°	Parameters description	Unit	Motor utilization		
V119	Phase lead	Reserved parameters			
V120	Linear Motor EMF phase compensation	Reserved parameters			
V121	Set Point Weight current loop gain	Reserved parameters			
V122	Anti-windup speed current gain	Reserved parameters			
V123	Step of Iq ref. interpolation	Reserved parameters			
V125	Current percentage V108 for EPSM	%	SM	•	LM
V128	Locked rotor current I^2t (% V108)	%	SM	AM	LM
V129	Locked rotor current I^2t (% V115)		•	AM	•
V130	Associate MP with TP1	[ ]	SM	AM	LM
V131	Associate MP with TP2	[ ]	SM	AM	LM
V132	Associate MP with TP3	[ ]	SM	AM	LM
V133	Associate MP with TP4	[ ]	SM	AM	LM
V134	Multiplication factor for TP1	[ ]	SM	AM	LM
V135	Multiplication factor for TP2	[ ]	SM	AM	LM
V136	Offset for analog TP1	[ ]	SM	AM	LM
V137	Offset for analog TP2	[ ]	SM	AM	LM
V138	Configuration of analog TPs • Bit 0 = Wrap-around TP1 [1=active] • Bit 1 = Wrap-around TP2 [1=active] • Bit 2 =Output TP1 [0=Enabled] [1=Disabled] • Bit 3 =Output TP2 [0=Enabled] [1=Disabled]	Bit	SM	AM	LM
V139	Configuration of Programmable Relays • Bit 0 = Brake management [1=Active] • Bit 1 = Brake management output on [1=PROGO2] [0=PROGO1] • Bit 2 = Absolute value for prog. relay 1 [1= absolute] • Bit 3 = Absolute value for prog. relay 2 [1=absolute] • Bit 4 = Programmable relay 1 mode [1= band mode] [0=thresholds] • Bit 5 = Programmable relay 2 mode [1= band mode] [0=thresholds] • Bit 6 = Logic of programmable relay 1 [1=positive] [0=negative] • Bit 7 = Logic of programmable relay 2 [1=positive] [0=negative] • Bit 10 = Logic of limit switches [1=NO] [0=NC]	Bit	SM	AM	LM
V140	Associate MP with programmable relay 1	[ ]	SM	AM	LM
V141	Programmable relay 1, threshold 1	[ ]	SM	AM	LM
V142	Programmable relay 1, threshold 2	[ ]	SM	AM	LM
V143	Programmable relay 1, MP scale factor	[ ]	SM	AM	LM
V144	Programm. relay 1, tripping time constant	ms	SM	AM	LM
V150	Associate MP with programmable relay 2	[ ]	SM	AM	LM
V151	Programmable relay 2, threshold 1	[ ]	SM	AM	LM
V152	Programmable relay 2, threshold 2	[ ]	SM	AM	LM
V153	Programmable relay 2, MP scale factor	[ ]	SM	AM	LM
V154	Programm. relay 2, tripping time constant	ms	SM	AM	LM

N°	Parameters description	Unit	Motor utilization		
V160	Do not use				
V165	Do not use				
V166	Do not use				
V169	Do not use				
V170	Do not use				
V171	Do not use				
V172	Do not use				
V173	Do not use				
V175	Do not use				
V176	Do not use				
V180	Current percentage (V108) for EPS	%	SM	AM	LM
V182	Phase U/W relative weight for EPS	%	SM	AM	LM
V183	Phase V/W relative weight for EPS	%	SM	AM	LM
V190	Do not use				
V191	Do not use				
V205	Do not use				
V206	Do not use				
V207	Do not use				
V208	Do not use				
V209	Do not use				
V210	Do not use				
V217	Do not use				
V218	Do not use				
V220	Reference configuration: • Bit 0 = Reset [00=No reset] [01=Pos. Ctrl Output] • Bit 1 = Reset [10=Ref. Int. Pos.] [11=Speed Ctrl Output] • Bit 2 = Interpolator active [1=Active] • Bit 3 = Interpolator Type [0=1 order] [1=2 order] • Bit 4 = Reversal of speed reference [1=REV] • Bit 5 = Not used • Bit 6 = Reversal of current reference [1=REV] • Bit 7 = Interpolation of current reference V123 [1=Enable]	Bit	SM	AM	LM
V221	Digital enable configurations : • Bit 0 = Torque enable [0=Digital bus 1=GP_INA/B] • Bit 1 = Reference enable [0=Digital bus 1=GP_INA/B] • Bit 2 = Torque AND Digital bus enable • Bit 3 = Reference AND Digital bus enable • Bit 4 =Change of range mode [0=Digital bus 1=GP_INA/B]	Bit	SM	AM	LM

N°	Parameters description	Unit	Motor utilization		
V225	Digital input configuration [GP_INA/B] • Bit 0 = Torque enable [0=GP_IN_A] [1=GP_IN_B] • Bit 1 = Reference enable [0=GP_IN_A] [1=GP_IN_B] • Bit 2 = Homing switch S1 [0=GP_IN_A] [1=GP_IN_B] • Bit 3 = Homing switch S2 [0=GP_IN_A] [1=GP_IN_B] • Bit 4 = Homing switch S1 logic inverted [1=reverse] • Bit 5 = Homing switch S2 logic inverted [1=reverse] • Bit 6 = Change of range request [0=GP_IN_A] [1=GP_IN_B]	Bit	SM	AM	LM
V238	UGA filter on S1 sensor (RCMDS20)	Reserved parameter			
V239	UGA filter on S2 sensor (RCMDS21)	Reserved parameter			
V240	First time constant for I^2t	Reserved parameter			
V241	Second time constant for I^2t	Reserved parameter			
V245	Frequency threshold. Change of time const. I^2t	Reserved parameter			
V246	Frequency threshold hysteresis V245	Reserved parameter			
V250	ST Revision	Reserved parameter			
V251	Parameter Compatibility	Reserved parameter			
V252	Drive type	Reserved parameter			
V253	Drive size	Reserved parameter			
V254	Motor number	Reserved parameter			
V255	Motor type	Reserved parameter			
V256	Motor power Connection type	Reserved parameter			
V257	Motor poles pairs number	Reserved parameter			
V258	Motor sensor type	[ ]	SM	AM	LM
V259	Linear Motor Pole Pitch [180 degrees el.]	Reserved parameter			
V260	Speed regulator Parameters exchange	[ ]	SM	AM	LM
V261	Position regulator Parameters exchange	[ ]	SM	AM	LM
V263	System sampling time	Reserved parameter			
V264	Speed loop sampling time	Reserved parameter			
V265	Position loop sampling time	Reserved parameter			
V266	Do not use				
V267	ST Peak current	Reserved parameter			
V268	Electrical phase offset Hall effect sensor	mDegree	SM	•	LM
V269	Position offset for electrical phasing	mDegree	SM	•	LM
V270	Motor PTC-NTC value	Ohm	SM	AM	LM
V271	Current loop sampling time	Reserved parameter			
V272	Motor thermal sensor configuration : • Bit 0 = Sensor Type [0=PTC] [1=NTC] • Bit 1 = Sensor wiring [0=S1] [1=S2] • Bit 2 = Motor temperature sensor control NTC open [1=act.]	Bit	SM	AM	LM

N°	Parameters description	Unit	Motor utilization		
V273	Electrical phase offset management : • Bit 0 = Phase offset exchange after EPS/Hall+HOMING [1=NO] • Bit 1 = CNC On-line el. phase offset tuning [1=ON] • Bit 2 = APEM Automatic phase offset exchange on mark [1=ON] • Bit 3 = Exclusion of selfcalibration alarm A18.3 [1=excluded] • Bit 4 = Forced EPS with absolute encoders [1=ON] • Bit 5 = Exclusion of selfcalibration alarm A18.2 [1=Excluded] • Bit 6 = EPS for highly anisotropic motors [1=ON]	Bit	SM	AM	LM
V301	Proportional position gain [0]	mm/min/ mm	SM	AM	LM
V306	S1 sensor MULTI coefficient [0]	[ ]	SM	AM	LM
V307	S1 sensor DIVI coefficient [0]	[ ]	SM	AM	LM
V310	S1 sensor offset in CNC Phys. Unit [0]	[ ]	SM	AM	LM
V311	S2 sensor MULTI coefficient [0]	[ ]	SM	AM	LM
V312	S2 sensor DIVI coefficient [0]	[ ]	SM	AM	LM
V313	S2 sensor offset in CNC Phys. Unit [0]	[ ]	SM	AM	LM
V314	Sinusoidal signals range of S1 Sensor : • Bit 0 = Not used • Bit 1 = Not used • Bit 2 = Threshold A09.254 [00=50%] [01=40%] • Bit 3 = Threshold A09.254 [10=30%] [11=20%] • Bit 4 = Not used • Bit 5 = Alarms A09.230/248/254 exclusion [1=Excluded]	[ ]	SM	AM	LM
V315	Sinusoidal signals range of S2 Sensor : • Bit 0 = Not used • Bit 1 = Not used • Bit 2 = Threshold A38.254 [00= 50%][01= 40%] • Bit 3 = Threshold A38.254 [10=30%] [11=20%] • Bit 4 = Not used • Bit 5 = Alarms A38.230/248/254 exclusion [1=Excluded]	Bit	SM	AM	LM
V317	S1 sensor power supply voltage	V	SM	AM	LM
V318	S2 sensor power supply voltage	V	SM	AM	LM
V340	Step of speed calculation S1	[ ]	SM	AM	LM
V341	Step of speed calculation S2	[ ]	SM	AM	LM
V342	Speed filter by S1 sensor	Hz	SM	AM	LM
V343	Speed filter by S2 sensor	Hz	SM	AM	LM
V344	S1 Sensor code ID	[ ]	SM	AM	LM
V345	S2 Sensor code ID	[ ]	SM	AM	LM
V346	S1 Sensor number	[ ]	SM	AM	LM
V347	S2 Sensor number	[ ]	SM	AM	LM

N°	Parameters description	Unit	Motor utilization		
V348	S1 Sensor configuration: • Bit 0 = Incr. Trace [000=None] [001=Sinus] [010=TTL] • Bit 1 = Incr. Trace [011=Resolver] [100=NC] [101=Fast Serial] • Bit 2 = Incr. Trace [110=NC] [111=Sensorless] • Bit 3 = Absolute Trace [000=None] [001=HipNUM] [010=EnDat] • Bit 4 = Absolute Trace [011=Ref.Cod][100=Hip BL][101=Hiperf] • Bit 5 = Absolut Trace [110=SSI] [111=EnDat2.2] • Bit 6 = CNC measure [0=incremental] [1=absolute] • Bit 7 = Sensor cyclical direction [1=clockwise] • Bit 8 = Sensor type [0=rotative] [1=linear] • Bit 9 = Reversal measure for position loop [1=Rev] • Bit 10 = Alarm A09.248 exclusion [1=Excluded] • Bit 11 = Oversampling of sinusoidal signals [1=ON] • Bit 13 = Sensor error signal test (pin 21) [1=ON] • Bit 14 = Compensation of sinusoidal signals [1=ON] • Bit 15 = Abs. measure translation for rotary sensor [1=OFF] • Bit 16 = Homing switch use excluded [1=Excluded] • Bit 17 = Speed filter [0=V342] [1=V340] • Bit 18 = SSI-Endat Abs position half-travel displacement • Bit 19 = Endat memory handling excluded [1=Excluded] • Bit 20 = EnDat 2.2. sensor CRC error [0=Alarm] [1=Warning]	Bit	SM	AM	LM
V349	S2 Sensor configuration : • Bit 0 = Incr. Trace [000=None] [001=Sinus] [010=TTL] • Bit 1 = Incr. Trace [011=NC] [100=Hall] [101=Fast Serial] • Bit 2 = Incr. Trace [110=NC] [111=Sensorless] • Bit 3 = Absolute Trace [000=None] [001=HipNUM] [010=EnDat] • Bit 4 = Absolute Trace [011=Ref.Cod][100=Hip BL][101=Hiperf] • Bit 5 = Absolute Trace [110=SSI] [111=EnDat2.2] • Bit 6 = CNC measure [0=incremental] [1=absol.] • Bit 7 = Sensor cyclical direction [1=Clockwise] • Bit 8 = Sensor type [0=rotative] [1=linear] • Bit 9 = Reversal measure for position loop [1=Rev]] • Bit 10 = Alarm A38.248 exclusion [1=Excluded] • Bit 11 = Oversampling of sinusoidal signals [1=ON] • Bit 12 = Not used • Bit 13 = Sensor error signal test (pin 21) [1=ON] • Bit 14 = Compensation of sinusoidal signals [1=ON] • Bit 15 = Abs. measure translation for rotary sensor [1=OFF] • Bit 16 = Homing switch use excluded [1=Excluded] • Bit 17 = Speed filter [0=V343] [1=V341] • Bit 18 = SSI-Endat Abs position half-travel displacement • Bit 19 = Endat memory handling excluded [1=Excluded] • Bit 20 = EnDat 2.2. sensor CRC error [0=Alarm] [1=Warning]	Bit	SM	AM	LM
V350	Position loop sensor configuration : • Bit 0 = Position loop sensor configuration [0=S1] [1=S2]	Bit	SM	AM	LM
V351	Speed loop sensor configuration : • Bit 0 =Speed loop sensor configuration [0=S1] [1=S2]	Bit	SM	AM	LM
V352	Current loop sensor configuration : • Bit 0 =Current loop sensor configuration [0=S1] [1=S2]	Bit	SM	AM	LM
V353	Period of position sensor/motor turn [0]	[ ]	SM	AM	•
V355	AP02: Mechanical ratio load turns	[ ]	SM	AM	•
V356	AP02 : Mechanical ratio sensor turns	[ ]	SM	AM	•
V360	Period by turn sensor S1	[ ]	SM	AM	•
V361	Absolute resolution by turn sensor S1	[ ]	SM	AM	•
V365	Number of absolute sensor turns S1	[ ]	SM	AM	•
V368	Intervals between coded references S1	[ ]	SM	AM	LM

N°	Parameters description	Unit	Motor utilization		
V370	Period by turn sensor S2	[ ]	SM	AM	•
V371	Absolute resolution by turn sensor S2	[ ]	SM	AM	•
V375	Number of absolute sensor turns S2	[ ]	SM	AM	•
V378	Intervals between coded references S2	[ ]	SM	AM	LM
V380	Period of linear sensor S1	um	SM	AM	LM
V381	Absolute resolution of linear sensor S1	nm	SM	AM	LM
V390	Period of linear sensor S2	um	SM	AM	LM
V391	Absolute resolution of linear sensor S2	nm	SM	AM	LM
V400	Rs at 120 degrees C	Reserved parameter			
V401	Rr at 150 degrees C	Reserved parameter			
V402	Sigma Ls	Reserved parameter			
V403	Lr	Reserved parameter			
V406	twotosh0	Reserved parameter			
V407	ratio_gamma	Reserved parameter			
V408	Additional extern inductance	Reserved parameter			
V409	ST flux-weakening voltage	Reserved parameter			
V410	Proportional gain flux loop l. r.	Reserved parameter			
V411	Integral gain flux loop l. r.	Reserved parameter			
V412	Lr saturation level l. r.	Reserved parameter			
V414	Observer engagement speed l. r.	Reserved parameter			
V415	Observer engagement hysteresis l. r.	Reserved parameter			
V417	Defluxing starting speed l. r.	Reserved parameter			
V425	min_flux as a %age of V427 l. r.	Reserved parameter			
V426	K deflux l. r.	Reserved parameter			
V427	Magnetization current l. r.	Reserved parameter			
V428	Magnetization current limit l. r.	Reserved parameter			
V450	Proportional flux loop gain h. r.	Reserved parameter			
V451	Integral flux loop gain h. r.	Reserved parameter			
V452	Lr saturation level Lr h. r.	Reserved parameter			
V454	Observer engagement speed h. r.	Reserved parameter			
V455	Observer engagement hysteresis h. r.	Reserved parameter			
V457	Defluxing starting speed h. r.	Reserved parameter			
V465	min_flux as a %age of V467 h. r.	Reserved parameter			
V466	K deflux h. r.	Reserved parameter			
V467	Magnetization current h. r.	Reserved parameter			

N°	Parameters description	Unit	Motor utilization		
V468	Magnetization current limit h. r.	Reserved parameter			
V480	Correction factor for MP006 Delivered Power	Reserved parameter			
V481	Control mode selection	Reserved parameter			
V488	Resistance tt at 20 degrees C	Reserved parameter			
V489	Motor Lq inductance	Reserved parameter			
V491	Id_Max (defluxing parameter)	Reserved parameter			
V492	Id current without defluxing	Reserved parameter			
V496	Defluxing starting speed	Reserved parameter			
V497	Defluxing starting speed	Reserved parameter			
V498	K Iq reduction	Reserved parameter			
V499	K flux-weakening Id	Reserved parameter			
V500	Filter configuration : • Bit 0 = 1. filter for torque current reference [1=Active] • Bit 1 = 2. filter for torque current reference [1=Active] • Bit 2 = 3. filter for torque current reference [1=Active] • Bit 3 = 4. filter for torque current reference [1=Active] • Bit 4 = 1. on speed reference [1=Active] • Bit 5 = Notch Filter for AP13 [1=Active] • Bit 6 = Not used • Bit 7 = Filters UPDATE on line • Bit 8 = TEST MODE: test on Iq reference filters	Bit	SM	AM	LM
V501	ph11 (FILTER 1 for torque current reference)	[ ]	SM	AM	LM
V502	ph12 (FILTER 1 for torque current reference)	[ ]	SM	AM	LM
V503	ph21 (FILTER 1 for torque current reference)	[ ]	SM	AM	LM
V504	ph22 (FILTER 1 for torque current reference)	[ ]	SM	AM	LM
V505	ga1 (FILTER 1 for torque current reference)	[ ]	SM	AM	LM
V506	ga2 (FILTER 1 for torque current reference)	[ ]	SM	AM	LM
V507	r*r (FILTER 1 for torque current reference)	[ ]	SM	AM	LM
V508	C1 (FILTER 1 for torque current reference)	[ ]	SM	AM	LM
V509	C2 (FILTER 1 for torque current reference)	[ ]	SM	AM	LM
V511	ph11 (FILTER 2 for torque current reference)	[ ]	SM	AM	LM
V512	ph12 (FILTER 2 for torque current reference)	[ ]	SM	AM	LM
V513	ph21 (FILTER 2 for torque current reference)	[ ]	SM	AM	LM
V514	ph22 (FILTER 2 for torque current reference)	[ ]	SM	AM	LM
V515	ga1 (FILTER 2 for torque current reference)	[ ]	SM	AM	LM
V516	ga2 (FILTER 2 for torque current reference)	[ ]	SM	AM	LM
V517	r*r (FILTER 2 for torque current reference)	[ ]	SM	AM	LM
V518	C1 (FILTER2 for torque current reference)	[ ]	SM	AM	LM
V519	C2 (FILTER 2 for torque current reference)	[ ]	SM	AM	LM

N°	Parameters description	Unit	Motor utilization		
V521	ph11 (FILTER 3 for torque current reference)	[ ]	SM	AM	LM
V522	ph12 (FILTER 3 for torque current reference)	[ ]	SM	AM	LM
V523	ph21 (FILTER 3 for torque current reference)	[ ]	SM	AM	LM
V524	ph22 (FILTER 3 for torque current reference)	[ ]	SM	AM	LM
V525	ga1 (FILTER 3 for torque current reference)	[ ]	SM	AM	LM
V526	ga2 (FILTER 3 for torque current reference)	[ ]	SM	AM	LM
V527	r*r (FILTER 3 for torque current reference)	[ ]	SM	AM	LM
V528	C1 (FILTER 3 for torque current reference)	[ ]	SM	AM	LM
V529	C2 (FILTER 3 for torque current reference)	[ ]	SM	AM	LM
V531	ph11 (FILTER 4 for torque current reference)	[ ]	SM	AM	LM
V532	ph12 (FILTER 4 for torque current reference)	[ ]	SM	AM	LM
V533	ph21 (FILTER 4 for torque current reference)	[ ]	SM	AM	LM
V534	ph22 (FILTER 4 for torque current reference)	[ ]	SM	AM	LM
V535	ga1 (FILTER 4 for torque current reference)	[ ]	SM	AM	LM
V536	ga2 (FILTER 4 for torque current reference)	[ ]	SM	AM	LM
V537	r*r (FILTER 4 for torque current reference)	[ ]	SM	AM	LM
V538	C1 (FILTER 4 for torque current reference)	[ ]	SM	AM	LM
V539	C2 (FILTER 4 for torque current reference)	[ ]	SM	AM	LM
V541	ph11 (FILTER 1 for speed reference)	[ ]	SM	AM	LM
V542	ph12 (FILTER 1 for speed reference)	[ ]	SM	AM	LM
V543	ph21 (FILTER 1 for speed reference)	[ ]	SM	AM	LM
V544	ph22 (FILTER 1 for speed reference)	[ ]	SM	AM	LM
V545	ga1 (FILTER 1 for speed reference)	[ ]	SM	AM	LM
V546	ga2 (FILTER 1 for speed reference)	[ ]	SM	AM	LM
V547	r*r (FILTER 1 for speed reference)	[ ]	SM	AM	LM
V548	r*r (FILTER 1 for speed reference)	[ ]	SM	AM	LM
V549	C2 (FILTER 1 for speed reference)	[ ]	SM	AM	LM
V551	ph11 (FILTER notch for AP13)	[ ]	SM	AM	LM
V552	ph12 (FILTER notch for AP13)	[ ]	SM	AM	LM
V553	ph21 (FILTER notch for AP13)	[ ]	SM	AM	LM
V555	ga1 (FILTER notch for AP13)	[ ]	SM	AM	LM
V556	ga2 (FILTER notch for AP13)	[ ]	SM	AM	LM
V557	r*r (FILTER notch for AP13)	[ ]	SM	AM	LM
V558	C1 (FILTER notch per AP13)	[ ]	SM	AM	LM
V559	C2 (FILTER notch per AP13)	[ ]	SM	AM	LM

N°	Parameters description	Unit	Motor utilization		
V600	V/f: Maximal motor current (% of V108)	Reserved parameter			
V601	V/f: Rated frequency	Reserved parameter			
V602	V/f: Rated voltage	Reserved parameter			
V603	V/f: V0	Reserved parameter			
V604	V/f: Stator resistance @ 20 degrees C	Reserved parameter			
V605	V/f: Intermittent overload (% of V108)	Reserved parameter			
V606	V/f: Freq. start of overload reduction	Reserved parameter			
V1050	Torque current limit [1]	Arms	SM	AM	LM
V1051	Proportional speed loop gain [1]	mA/ Rpm	SM	AM	•
V1052	Integral speed loop gain [1]	ms	SM	AM	LM
V1053	Do not use				
V1057	Torque current limit h.r. [1]	Arms	•	AM	•
V1058	Proportional speed loop gain h.r. [1]	mA/ Rpm	•	AM	•
V1059	Integral speed loop gain h.r. [1]	ms	•	AM	•
V1060	Proportional speed loop gain [1]	Arms/ m/min	•	•	LM
V1075	Do not use				
V1094	Speed reference Limit [% maximum speed] [1]	%	•	•	LM
V1095	Maximum motor speed [1]	m/min	•	•	LM
V1096	Threshold for overspeed alarm [1]	m/min	•	•	LM
V1097	Speed reference limit [% maximum speed] [1]	%	SM	AM	•
V1098	Maximum motor speed [1]	Rpm	SM	AM	•
V1099	Overspeed alarm threshold [1]	Rpm	SM	AM	•
V1301	Proportional position gain [1]	mm/min /mm	SM	AM	LM
V1306	S1 sensor MULTI coefficient [1]	[ ]	SM	AM	LM
V1307	S1 sensor DIVI coefficient [1]	[ ]	SM	AM	LM
V1310	S1 sensor offset in CNC Phys. Unit [1]	[ ]	SM	AM	LM
V1311	S2 sensor MULTI coefficient [1]	[ ]	SM	AM	LM
V1312	S2 sensor DIVI coefficient [1]	[ ]	SM	AM	LM
V1313	S2 sensor offset in CNC Phys. Unit [1]	[ ]	SM	AM	LM
V1353	Period of position sensor/motor turn [1]	[ ]	SM	AM	•

5.5 NUMDrive C parameters description

5.5.1 Mono-Axis and Bi-Axes parameters description

V01 CAN configuration

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V01	CAN configuration	Bit	0 ÷ 2	0	SM, AM, LM	YES / 2

The V01 parameter allows to configure the CAN baudrate as indicated in the table below.

Bit 2	Bit 1	Bit 0	CAN bit rate configuration	Bus length	Unterminated cable drop length	Cumulative length cable drop length
0	0	0	50kb/s	1000m	55m	275m
0	1	0	125kb/s	500m	22m	110m
0	1	1	250kb/s	250m	11m	55m
1	0	0	500kb/s (1)	100m	5.5m	27.5m
1	0	1	1Mkb/s	25m	1.5m	7.5m
(1) Default value.						

The V01 parameter must always be configured accordig to the value selected on Flexium Tools CANbus main page configuration. Please refer to the chapter 3.3.

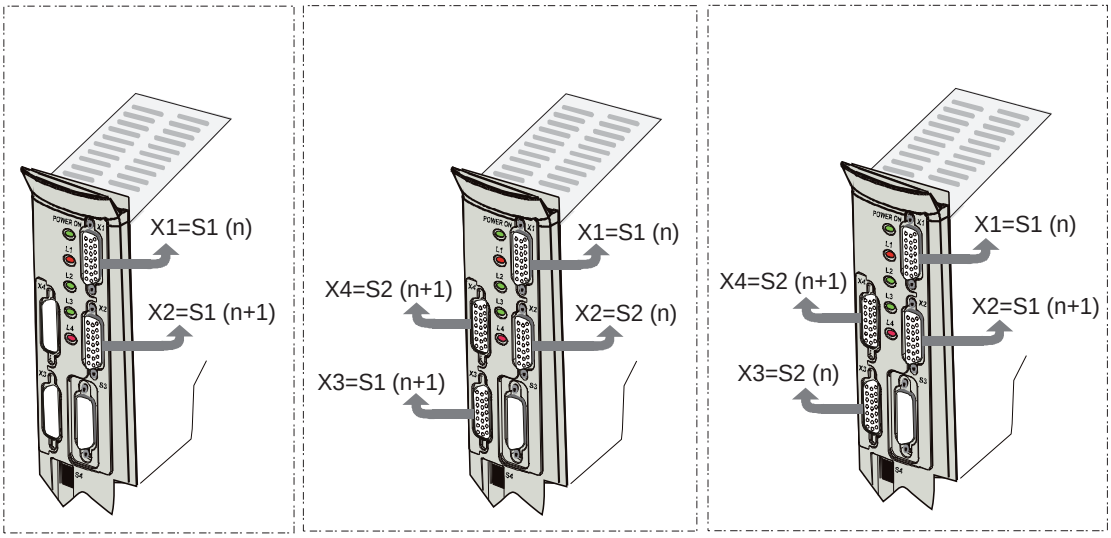
V20 Sensor connectors X3, X4 configuration (Bi-Axes configuration only)

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V20	Sensor connectors X3, X4 configuration	Bit	0 ÷ 3	0	SM, AM, LM	YES / 1

The V20 parameter allows to configure the second sensor board for Bi-Axes modules.
This parameter is taking into account in the Axis(n) configuration only.

Set the bit 0 to assign the sensor connectors X3, X4.
Set the bit 1 to assign the sensor configuration.

N° Bit	Bit function	Bit value = 0	Bit value = 1	Default value
0	Sensor connectors X3, X4 [0=Absent] [1=Plugged-in]	Sensor connectors X3, X4 absent.	Sensor connectors X3, X4 present. (Plugged-in)	0
1	[0=> X3=S1(n+1) X4=S2(n+1)] [1=> X3=S2(n) X4=S2(n+1)]	The X3 connector is assigned as S1 for Axis n+1, and the X4 connector is assigned as S2 for Axis n+1.	The X3 connector is assigned as S2 for Axis n, and the X4 connector is assigned as S2 for Axis n+1.	0



V21 TP1/2 output configuration (Bi-Axes configuration only)

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V21	TP1/2 output configuration	Bit	0 ÷ 3	0	SM, AM, LM	NO / 1

The V21 parameter allows to assign the analog output TP1 and TP2 test points and relative measure point (MP) to the physical S9.3, S9.1 outputs for Bi-Axes modules.
This parameter is taking into account in the Axis(n) configuration only.

Set the bit 0 to assign the TP1 output.
Set the bit 1 to assign the TP2 output.

N° Bit	Bit function	Bit value = 0	Bit value = 1	Default value
0	Output S9.3 : [0=(TP1 axis n)] [1=(TP1 axis n+1)]	TP1 for Axis n assigned onto pin 3 of S9 connector.	TP1 for Axis n+1 assigned onto pin 3 of S9 connector.	0
1	Output S9.1 [0=(TP2 axis n)] [1=(TP2 axis n+1)]	TP2 for Axis n assigned onto pin 1 of S9 connector.	TP2 for Axis n+1 assigned onto pin 1 of S9 connector.	0

For a correct TP1 and TP2 connection please see NUMDrive C Installation manual code AMOMAN012.

V22 PROGO1/2 output configuration (Bi-Axes configuration only)

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V22	S7.2/3 I/O configuration	Bit	0 ÷ 3	0	SM, AM, LM	YES / 1

The V22 parameter allows to assign PROGO1 / PROGO2 programmable outputs to the physical outputs S7.2, S7.3 for Axis (n) or Axis (n+1).
This parameter is taking into account in the Axis(n) configuration only.

Set the bit 0 to assign.
Set the bit 1 to assign PROGO2 output.

N° Bit	Bit function	Bit value = 0	Bit value = 1	Default value
0	I/O S7.2	PROGO1 for Axis n assigned onto pin 2 of S7 connector.	PROGO1 for Axis n+1 assigned onto pin 2 of S7 connector.	0
1	I/O Output S7.3	PROGO2 for Axis n assigned onto pin 3 of S7 connector.	PROGO2 for Axis n+1 assigned onto pin 3 of S7 connector.	0
2	I/O Output S7.2	PROGO1 assigned onto pin 2 of S7 connector.	GP_IN_C assigned onto pin 2 of S7 connector.	0
3	I/O Output S7.3	PROGO2 assigned onto pin 3 of S7 connector.	GP_IN_D assigned onto pin 3 of S7 connector.	0

For a correct PROGO1 and PROGO2 connection please see NUMDrive C Installation manual code AMOMAN012.

V23 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V23						

V30 General Drive Configuration

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V30	General Drive Configuration	BIT	0 ÷ 7FFFFFFF	1040	SM, AM, LM	YES / 1

The V30 parameter allows to sets up the main general drive configuration by means of bit configurable functions. The parameters reported on the table below are bit parameters used for the general configuration of the drive.

V30 Use of the bits

N° Bit	Bit function	Bit value = 0	Bit value = 1	Default value
0	Wait for motor stop to enable.	No wait. The torque is enabled even with the motor in motion.	Wait. The torque is enabled only with motor stopped.	0
3	Exclusion of range change if required. (Used only for the spindle motor with two ranges: low range/high range)	Range change not excluded.	Range change excluded. The electric range change is excluded and the electrical range is selected by Bit 4.	0
4	Forcing of electrical range (Associated to bit 3 =1)	Forcing low range. The low range is forced.	Forcing high range. The high range is forced.	0
5	Following error A37.0 exclusion.	Active:The drive alarm A37 is actived.	Excluded: The drive alarm A37 is excluded.	0
6	I^2t handled as an alarm A49.0.	I^2t handled as warning.	I^2t handled as alarm.	1
7	MP011 (LOAD) = 100%.	With nominal load value.	With maximum load value.	0
10	Short circuit to ground A06 exclusion.	A06 short circuit exclusion enabled.	A06 short circuit exclusion disabled.	0
11	Enabling external HW alarm Reset.	Alarm Reset disabled.	Alarm Reset enabled.	0
12	UGA misalignment handling.	As warning status.	As alarm status. (A09 /A38.230)	1
13	Limited torque when reference disabled.	Torque limitation disabled.	Torque limitation enabled.	0
16	Brake failure A54 exclusion.	Not excluded.	Excluded.	0
18	Integration of NUM-STOP	No required	Required	0
24	Alarm A60.268	Alarm disabled.	Alarm enabled.	0
26	Alarm A09/A38 ext.231 exclusion	Not excluded	Excluded	0
27	Alarm A03 exclusion	Not excluded	Excluded	0
28	Alarm A53.22 exclusion	Not excleded	Excluded	0
30	Drive parking. (1)	Drive parking disabled.	Drive parking enabled.	0

(1) Drive parking gives the possibility to hide all the drive alarms staying with the drive drive in run status and allows to disconnect completely a motor (power and measure system) without to go in drive alarm status. This function could be used on milling machine when you will have to remove a motor equipped toolhead.

This function is selected by: V30 bit 30 = 1: Drive parking enabled.
If you need to manage this function we suggest to store the parameters with the drive configured into drive parking mode, so at the power on the drive could always go in RUN state, then depending on the machine state you can write V30 bit 30 = 0 or not and RUN the drive.
Using V30.30 = 1 the motor torque remains disabled.

V32 Configuration Braking following alarms

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V32	Configuration Braking following alarms	BIT	-	-	SM, AM, LM	YES / 1

The V32 parameter can configure the motor stop in case of alarm.
The motor may stop by inertia (coasting) or by braking.

- The motor stop configuration is only possible with configurable alarms (type C).
  In the table below are reported the C alarms (configurable).
- For not configurable alarms qualified as NC the motor stops by inertia.

N° Bit	Bit function	Bit value = 0	Bit value = 1	Default value
0	Motor overtemperatur e (A03)	Stop the motor by inertia	Stop the motor by braking (see modalities)	0
1	Module overtemperature (A04)			0
2	Brake resistor overtemperature (A05)			0
3	Local link alarm (A08)			0
4	S1 sensor alarm (A09)			0
5	Overspeed (A11)			0
6	Self-calibration (A18)			0
7	Following error (A37)			0
8	S2 sensor alarm (A38)			0
9	I^2t alarm (A49)			0
10	Position coherence S1 - S2 (A51)			0
11	Tandem function alarm (A52)			0
12	Trajectory generator (A90)			0
13	CANopen communication alarm (A91)			0

For a correct use of the configuration bits see next page and see “General MDLU3 alarms / securities configuration” on chapter 5.8.5.

V32 Use of configuration bit 1 - Braking modalities

N° Bit	Bit function Configuration braking following alarms	Bit = 1 Motor stop by braking
0	Motor overtemperatur (A03)	Motor stops with brake by setting to zero the speed reference with ramp if V30.15=1 or by torque limitation if V30.13=1. Setting V30.13= 0 or V30.15=0 the motor stops with the maximum current.
1	Module overtemperature (A04)	
2	Brake resistor overtemperature (A05)	
3	Local link alarm (A08)	
9	I2T alarm (A49)	
10	Position coherence S1 - S2 (A51)	
11	Tandem function alarm (A52)	
12	Trajectory generator (A90)	
13	CANopen communication alarm (A91)	
5	Overspeed (A11)	Motor stops by braking through short-circuiting the motor phases or Synchronous motors and Linear motors, motor stop by inertia for Asynchronous motors.
6	Self-calibration (A18)	
7	Following error (A37)	
4	S1 sensor alarm (A09)	S1 or S2 closes speed or current loop: Motor stops by braking through short-circuiting the motor phases for Synchronous motors and Linear motors, motor stops by inertia for Asynchronous motors. S1 or S2 closes position loop: Motor stops with brake by setting to zero the speed reference with ramp if V30.15=1 or by torque limitation if V30.13=1. Setting V30.13= 0 or V30.15=0 the motor stops with the maximum current.
8	S2 sensor alarm (A38)	

The slave drives in torque duplication configuration will always stop the motor by braking through short- circuiting the motor phases.

V33 Alarm Reset Configuration

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V33	Alarm Reset Configuration [1=Reset]	BIT	0 ÷ 3FFFFFF	3E3FFFF	SM, AM, LM	YES / 1

Every single alarm to be reset must be previously configured by means of parameter V33 with bit 1 = 1.

An alarm reset within the drive is carried out by EMCY message. The duration of reset pulse must be >500ms and the reset must be carried out in absence of torque enable request.

The alarms are divided in two categories according to the reset procedure used :

- NC = To reset the alarm, it is mandatory to turn OFF power and auxiliary supply. (Not Configurable)
- C = The alarm reset mode may be configured by means of WPM (Configurable).

To reset an alarm, the torque enable command must be absent.
The table on the next page shows the type C alarms (Configurable).

For the alarms reset configuration, please refer to the chapter 5.8.5 “General MDLU3 alarms / securities configuration.

V33 Use of the bits

N° Bit	Bit function Alarm Reset Configuration	Bit value = 0	Bit value = 1	Default value
0	Motor overtemperatur e (A03)	Alarm reset not enable	Alarm reset enable	1
1	Module overtemperature (A04)			1
2	Brake resistor overtemperature (A05)			1
3	Short-circuit to earth (A06)			1
4	Auxiliary Power Supply alarm (A07)			1
5	Local Link Alarm (A08)			1
6	S1 sensor alarm (A09)			1
7	Timeout (A10)			1
8	Overspeed (A11)			1
9	Drive overcurrent (A12)			1
10	Incompatible parameters (A16)			1
11	Self-calibration alarRm (A18)			1
12	Following error (A37)			1
13	S2 sensor alarm (A38)			1
14	I2T alarm (A49)			1
15	Position coherence S1 - S2 (A51)			1
16	Tandem function alarm (A52)			1
17	Safety Module failure class 0 (A60)			1
21	Brake failure (A54)			1
22	Incompatible sensors (A55)			1
23	Allarme V/f (A56)			1
24	Trajectory generator (A90)			1
25	CANopen communication alarm (A91)			1

For the reset and configuration alarms list see chapter 5.8.5, 5.8.6.

V34 Configuration DF1 / DF2 relay [1=break]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V34	Configuration DF1 / DF2 relay [1 = break]	BIT	0 ÷ FFFFF	0	SM, AM, LM	YES / 1

The V34 parameter can configure the mains contactor opening through the contact DF1 / DF2 of the MDLL3 power supply when any of the alarms listed in the table below occurs.

The alarms are divided in two categories depending on whether or not the opening of line remote switch through the contact DF1 / DF2 is provided.

- NC = Opening of contact DF1 / DF2 is not possible.
- C = Opening of contact DF1 / DF2 is configurable.

The table on the next page shows the type C alarms (Configurable). For a correct use of the configuration bits see “General MDLU3 alarms / securities configuration on chapter 5.8.5.

V34 Use of the bits

N° Bit	Bit function Configuration DF1 / DF2 relay [1 = break]	Bit value = 0	Bit value = 1	Default value
0	Bus Overvoltage (A02)	DF1 / DF2 contact do not change the status in case of alarm.	DF1 / DF2 contact opened in case of alarm	0
1	Motor overtemperature (A03)			0
2	Module overtemperature (A04)			0
3	Brake Resistor overtemperature (A05)			0
4	Short-circuit to earth (A06)			0
5	Auxiliary Power Supply alarm (A07)			0
6	Local Link Alarm (A08)			0
7	S1 sensor alarm (A09)			0
8	Timeout (A10)			0
9	Overspeed (A11)			0
10	Drive overcurrent (A12)			0
11	Parameter checksum error (A15)			0
12	Incompatible parameters (A16)			0
13	Self-calibration alarm (A18)			0
14	Following error (A37)			0
15	S2 sensor alarm (A38)			0
16	I^2t alarm (A49)			0
17	Position coherence S1 - S2 (A51)			0
18	Tandem function alarm (A52)			0
19	CANopen communication alarm (A91)			0

V35 SW function

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V35	SW functions [0=Not available][1=Available]	BIT	-	0	SM, AM, LM	YES / 1

The V35 parameter allows to enable the drive applications as indicated in the table.

N° Bit	Bit function SW functions	Bit value = 0	Bit value = 1	Default value
0	Do not use these bits	Not available application.	Available application	0
1				0
2				0
3				0
5				0
7				0
9				0
10				0
11	S2 Sensor			0
12	Linear Motor (for Mono-Axis only)			0

Check the correct setting between V35 and V160 parameter.
Otherwise alarm A16.35 (Software Functions not available) will appear.
Not available on Basic Performances version for NUMDrive C.

V41 Motor zero speed threshold, enable, brake (Linear motor)

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V41	Motor zero speed threshold, enable, brake	mm/min	1 ÷ 3000	300	LM	NO / 2

The V41 parameter is used to set the linear motor zero speed threshold.

This parameter may be used in according with V30 bit 0 configuration in order to control the motor torque enable / disable. (see V30.0 use) whenever the motor zero speed threshold is reached.

V42 Motor zero speed threshold, enable, brake

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V42	Motor zero speed threshold, enable, brake	Rpm	0.1 ÷ 5000	30	SM, AM	NO / 2

The V42 parameter is used to set the rotary motor zero speed threshold.

This parameter may be used for typically tool change application and it must to be adjusted in according with V30 bit 0 configuration in order to control the motor torque enable / disable. (see V30.0 use) whenever the motor zero speed threshold is reached.

V43 Braking Timeout on configuration V30 - V32 parameters

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V43	Braking Timeout on config. V30 - V32 parameters	s	1 ÷ 120	10	SM, AM, LM	YES / 1

The V43 parameter is used to set the braking timeout value. When the time will be elapsed the drive will be disabled and the motor stops by inertia (coasting). The watchdog time starts when a braking condition is activated.

The braking time depending on the V44, V91, V92 and V93 setting.

V44 Braking Current on configuration V30 - V32 parameters

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V44	Braking Current on config. V30 - V32 parameters	%	0 ÷ 100	100	SM, AM, LM	NO / 1

The V44 parameter is used to set the percentage (%) value of torque current which will be used from the motor during the brake phase, when any alarms listed on the V32 parameters occurs or by means V30 parameter configuration “Braking when disabling (bit1).

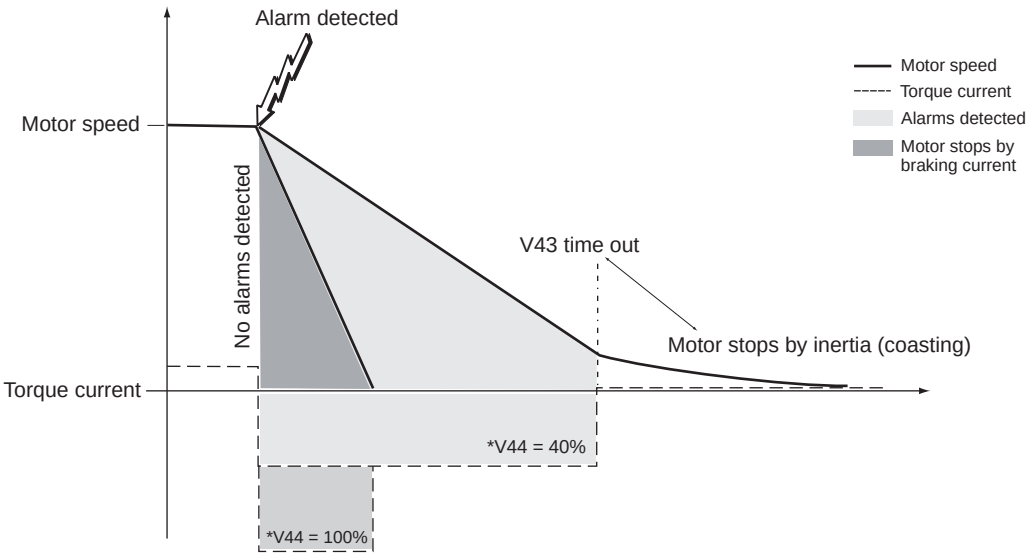
We suggest to use this configuration when the mechanics doesn’t allow to brake without ramp. Setting the current value it is possible to reduce the mechanical shock due to a high peak torque during the braking phases.

The parameter value is expressed in % and the default value 100%.
The percentage value is applied on the torque current limit value enabled during the braking phases (% value of V50, V1050, V57, V1057 parameters).

Example

- V30 bit1=1, V32 all bits to 1. (Drive and alarms with motor brake configuration)
- V50 value = 53.03 Arms
- V44 value = 40%
- V43 value = xx s.

  During the braking phases the drive stops the motor by means a torque current equal to V50*0.40 Arms.



V50 Torque current limit [0]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V50	Torque current limit [0]	Arms	0 ÷ ST value	4.30	SM, AM, LM	NO / 2

Through the V50 parameter it is possible to reduce the maximum drive current in order to limit the maximum torque delivered by the motor.

The value is expressed in Arms.

- Enter the Arms value of the desired torque current into parameter V50.
- Example of V50 adjustment for MDLU3021A ... (15 Arms. max)
- By entering the value 10.00 in parameter V50 the effective current delivered by the system will be 10 Arms max. (with subsequent reduction of motor maximum torque).

By selecting the motor/drive association, the maximum value foreseen for that motor will automatically be setup on V50 parameter.

ATTENTION! Never set-up current values higher than the standard value resulting from motor/drive association.

V51 Proportional speed loop gain [0]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V51	Proportional speed loop gain [0]	mA/Rpm	0 ÷ ST value	15	SM, AM, LM	NO / 2

The V51 parameter is used to improve the dynamic response of the system.

The value of the speed cycle's proportional gain will have to be increased by adjusting parameter V51 in general by the same percentage as the load inertia in relation to the motor.

For a correct use of the parameter see chapter 5.7.1 - Speed loop parameters optimisation procedure.

V52 Integral speed loop gain [0]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V52	Integral speed loop gain [0]	ms	0 ÷ ST value	30	SM, AM, LM	NO / 2

The V52 parameter is used to improve the dynamic response of the system.

For a correct use of the parameter see chapter 5.7.1 - Speed loop parameters optimisation procedure.

V53 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V53						

V57 Torque current limit - high range [0]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V57	Torque current limit h.r. [0]	Arms	0 ÷ ST value	4.30	AM	NO / 2

ATTENTION! To use the V57 parameter please refer to the description provided for parameter the V50 parameter. (only for asynchronous motors with range change)

The high range parameters can be used only for asynchronous motors with the electrical range change. The corresponding parameter for low range is the standard parameter used for axis motors. (Ex.V57 high range current limit adjustment - V50 lower current limit adjustment).

V58 Proportional speed loop gain - high range [0]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V58	Proportional speed loop gain h.r. [0]	mA/Rpm	0 ÷ ST value	100	AM	NO / 2

ATTENTION! To use the V58 parameter, see the description provided for the V51 parameter. (only for asynchronous motor with range change).

Parameters related to high range are used only for asynchronous motors with electrical range change. The corresponding parameter for low range is the standard parameter used for axis motors.

V59 Integral speed loop gain - high range [0]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V59	Integral speed loop gain h.r. [0]	ms	0 ÷ ST value	30	AM	NO / 2

ATTENTION! To use the V59 parameter, see the description provided for the V52 parameter. (only for asynchronous motor with range change).

Parameters related to high range are used only for asynchronous motors with electrical range change. The corresponding parameter for low range is the standard parameter used for axis motors.

V60 Proportional speed loop gain [0]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V60	Proportional speed loop gain [0]	Arms/m/min	0 ÷ ST value	15	LM	NO / 2

ATTENTION! To use the V60 parameter see the description provided for the V51 parameter. Set the parameter to a value allowing to obtain the desired dynamic performance.

To optimize this parameter it is necessary to carry out a preventive calibration of filters on currents reference.

V75 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V75						

V80 Set Point Weight speed loop gain

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V80	Set Point Weight speed loop gain	[ ]	0 ÷ 1.00	1.00	SM, AM, LM	NO / 2

The V80 parameter is used to adjust the Set Point Weight speed loop gain.
We suggesty to set V80 =1 (standard value).

ATTENTION! Using value <1 may reduce the speed feedback, but may reduce also the motor acceleration and the motor vibration.

V81 Anti-windup speed loop gain

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V81	Anti-windup speed loop gain	[ ]	1.00 ÷ 2.00	1.00	SM, AM, LM	NO / 2

The V81 parameter is used to adjust the Anti-windup speed loop gain.
We suggesty to set V81 =1 (standard value).

ATTENTION! Using value >1 may reduce the integrator saturation, but may produce also overshoot upon the integrator saturation output signal.

V94 Speed reference Limit [% maximum speed] [0]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V94	Speed reference limit (linear motor) [0]	%	0 ÷ 200	110	LM	NO / 2

To use the V94 parameter see the description provided for the V97 parameter.
The value of V94 parameter refer to a percentage (%) of V95 parameter.

V95 Maximum motor speed [0]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V95	Maximum motor speed [0]	m/min	0 ÷ 1000	10	LM	NO / 2

To use the V95 parameter see the description provided for V98 parameter.

V96 Threshold for overspeed alarm [0]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V96	Threshold for overspeed alarm [0]	m/min	0 ÷ 1000	12	LM	NO / 2

To use the V96 parameter see the description provided for V99 parameter.

V97 Speed reference limit [% max. speed] [0]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V97	Speed reference limit [% maximum speed] [0]	%	0 ÷ 200	110	SM, AM	NO / 2

During the control speed, delta position and position modes, the V97 parameter is the maximum value of the speed reference which can be reached.

The value of V97 parameter is a percentage (%) of V98 parameter.

Example

- V98 = 3000 Rpm
- V97 = 110 %
- V1097 = Parameter set [1] is a percentage (%) of the V1098 parameter.

In any case, the speed reference can never be greater than 3300 Rpm. Speed reference values greater than 3300 Rpm will be automatically limited by the system.

V98 Maximum motor speed [0]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V98	Maximum motor speed [0]	Rpm	0 ÷ V_{max} (1)	1500	SM, AM	NO / 2

Command mode - Speed control

- Digital reference : V98 parameter defines the rotation speed of the motor with the maximum digital reference ± (2^{24} -1) .
- Analog reference : V98 parameter defines the rotation speed of the motor with a reference of ± 10V.

Command mode - Position and delta position control

- Digital reference : V98 parameter is used as basic value for calculation of V97 parameter. (See also V97 parameter)
- Analog reference : See the above statement related to digital reference.

Application example

Spindle motors: The parameter value is strictly connected to the value set for object 607F. (maximum spindle speed expressed in rpm)

Example.
- Spindle speed 2000 rpm
- Reduction ratio 1:2
- V98 parameter value = 4000.

Axis motors: The parameter value is strictly connected to the value set for object 607F. (maximum axes speed expressed in mm/min)

Example.
- Axis speed 20000 mm/min
- Screw pitch 10 mm with motor directly connected on screw
- V98 parameter value = 2000.

1098 Parameter set [1] of V98 parameter.

V99 Overspeed alarm threshold [0]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V99	Overspeed alarm threshold [0]	Rpm	0 ÷ V_{max} (1)	2000	SM, AM	NO / 2

The motor overspeed control is one of the possible functions associated with the speed loop; in fact, the triggering of alarm A11 (motor overspeed) can be controlled through the adjustment of V99 parameter. By means of this parameter, it is possible for example to stop the motor in the event of an anomaly (motor coast or incorrect programming of parameters associated with the speed loop).

Setup V99 parameter with the desiderated overspeed value in Rpm. If the speed value in V99 parameter is exceeded, the alarm A11 will be activated. Always check that V99 parameter is compatible with the value setup in V98 parameter.

Overspeed values

- Axis motor overspeed : normally, the overspeed threshold value is calibrated at the rated motor speed +20%.
- Spindle motors overspeed : normally, the overspeed threshold value is calibrated at the rated motor speed +20%.

In case of overspeed alarm activation, the motor will stop by inertia or through short-circuiting phases and the alarm A11 will be displayed. (See V32 use of the bit configuration)

Application example with motor BPH 095/3/V.

- Motor operating speed 4000 Rpm + 20% = 4800 (value to be entered into V99)
- Select V99 parameter, then enter 4800.

For speed value >4800 Rpm the drive will display the alarm state A11.
V1099 = Parameter set [1] of V99 parameter.

V108 Rated motor current [I^2t]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V108	Rated motor current [I^2t]	Arms	3% $I_{max\ drive}$ ÷ $I_{nom\ drive}$	1	SM, AM, LM	YES/ 2

The use of these parameters is strictly reserved to NUM Technical Service. The motor rated current Inom [Arms] is automatically selected on V108 (V115) parameters during the motor / drive association.

The value of V108 (V115 using high range) parameters determines two adjustment thresholds of the current limit. (See operating diagram fig.1 on the following page)

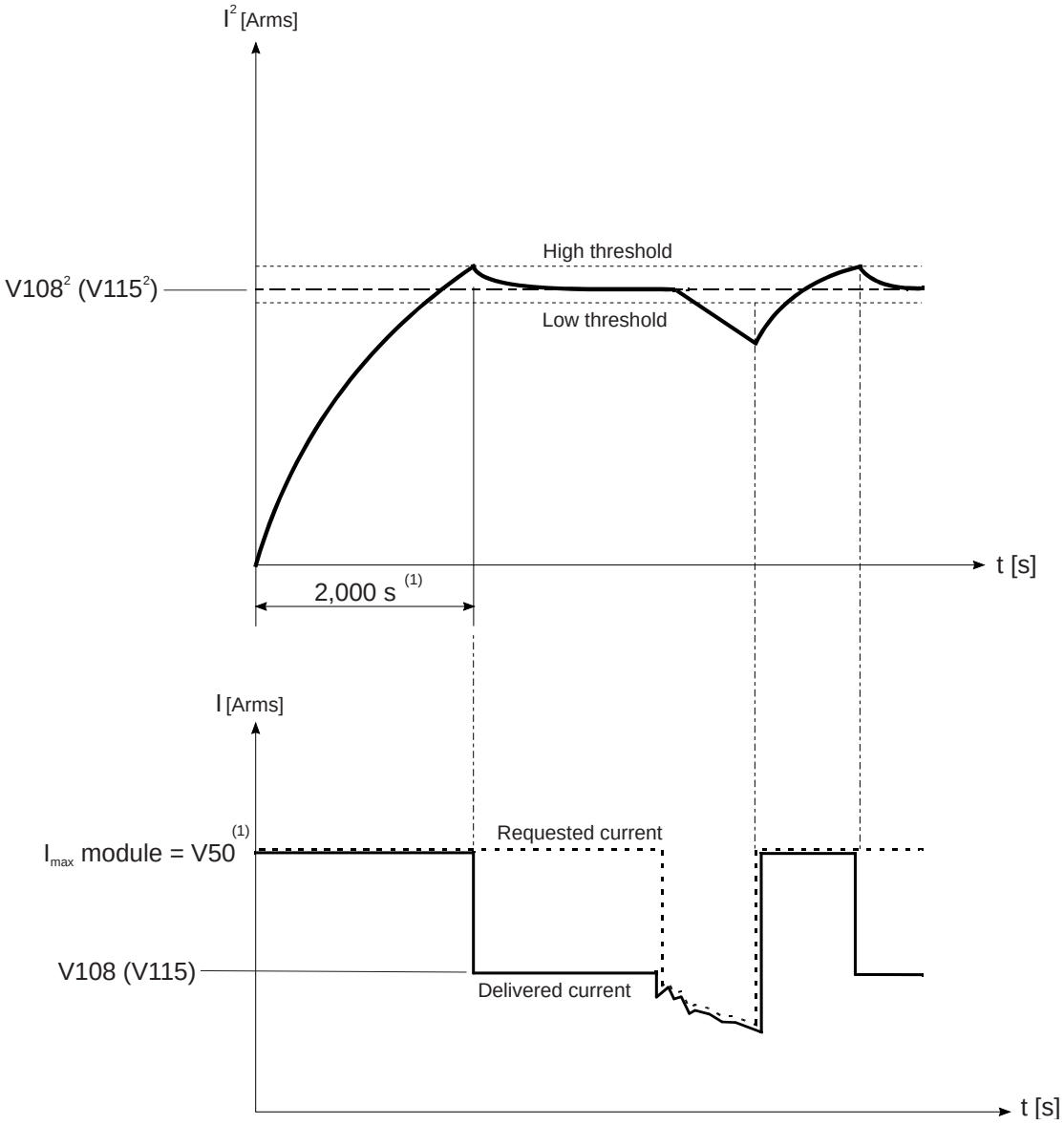
Motor thermal image management

Definitions :
- High threshold is given from : $(I_{nom} \times 1.1)^2$
- Low threshold is given from : $(I_{nom} \times 0.95)^2$.

High threshold definition. If the product I^2t reaches the high threshold value, the maximum current (Iq_{max}) delivered from the drive is reduced to the rated value as indicated on V108 (V115) parameters.
(See figure 1 on the following page)
It is possible to display the high threshold set on the measure point MP007 when the value is > 121%. (See also chapter 5.6)

Low threshold definition. If the product I^2t goes below the low threshold, the drive is able, if requested, to deliver again the peak value that can be delivered from the drive; as indicated on V50 (V57) parameters.
(See figure 1 on the following page)
It is possible to display the low threshold set on the on the measure point MP007 when the value is < 90.25%. (See also chapter 5.6)

V108 (V115) Thermal image management



(1) If V50 < I_{max} module t > 2s.

V109 Vertical load compensation current

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V109	Vertical load compensation current	Arms	- V108 ÷ +V108	1	SM, AM, LM	NO / 2

The V109 parameter allows to setup the current value delivered by the drive when enabling the torque.

The V109 parameter may be used to compensate the reaction due to the load applied to unbalanced vertical axes. The compensation value is expressed in Arms.

Select the right compensation value to prevent an unexpected axis movement during the system enable phase.

V115 Rated motor current [I^2t]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V115	Rated motor current [I^2t] h.r.	Arms	3% $I_{max\ drive}$ ÷ $I_{nom\ drive}$	3.20	AM	YES/ 2

To use the V115 parameter see the description provided for V108 parameter.

V125 Current percentage V108 for EPSM

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V125	Current percentage V108 for EPSM	%	0 ÷ (1)	50	SM, LM	NO / 2

$$(1) = \frac{100 * V100}{V108}$$

The V125 parameter sets the current percentage (%) to use for EPSM.

The parameter value is expressed in % and the default value 50%.

Increase the parameter value only if using special motors or when there are strong frictions preventing to reach the correct position during EPSM operation.
(Electrical Position Synchronisation with movement).

V128 Locked rotor current I^2t (% V108)

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V128	Locked rotor current I^2t (% V108)”	%	13 ÷ 100	100	SM, LM	YES / 2

The V128 define the maximum current value which could be maintened by the motor when the speed is lower than value setted in the V245.

V129 Locked rotor current I^2t (% V115)

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V129	Locked rotor current I^2t (% V115)	%	13 ÷ 100	100	AM	YES / 2

To use the V129 parameter see the description provided for the V128 parameter.

V130 Associate MP with TP1 (Analog / Digital)

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V130	Associate MP with TP1	[ ]	1 ÷ 500	50	SM, AM, LM	NO / 2

TP1 is used to display the internal measure point of the drive in analog form.
The measure point (MP) value is available in analog form on pin 3, S9 connector.

By means V130 TP1 test point it is possible to select the measure point which has to be associated to the TP1. (See chapter 5.6)

Example

V130 = 050 TP1 with Iq current measure point association. (Default)

V131 Associate MP with TP2 (Analog / Digital)

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V131	Associate MP with TP2	[ ]	1 ÷ 500	11	SM, AM, LM	NO / 2

TP2 is used to display the internal measure point of the drive in analog form.
The measure point (MP) value is available in analog form on pin 1, S9 connector.

By means V131 TP2 test point it is possible to select the measure point which has to be associated to the TP2. (See chapter 5.6)

Example

V131 = 011 TP2 with LOAD signal measure point association. (Default)

V132 Associate MP with TP3 (Digital)

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V132	Associate MP with TP3	[ ]	1 ÷ 500	8	SM, AM, LM	NO / 2

TP3 test point is used to display the internal measure point (MP) of the drive in digital form.

By means V132 TP3 test point it is possible to select the measure point which has to be associated to the TP3. (See chapter 5.6)

Example

V132 = 008 TP3 with BUS Voltage measure point association. (Default)

V133 Associate MP with TP4 (Digital)

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V133	Associate MP with TP4	[ ]	1 ÷ 500	260	SM, AM, LM	NO / 2

TP4 test point is used to display the internal measure point (MP) of the drive in digital form.

By means V133 TP4 test point it is possible to select the measure point which has to be associated to the TP4. (See chapter 5.6)

Example

V133 = 260 TP4 with motor PTC-NTC measure point association. (Default)

V134 Multiplication factor for TP1

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V134	Multiplication factor for TP1	[ ]	$-2^{31} \div +(2^{31}-1)$	134219776	SM, AM, LM	NO / 2

The multiplication factor for TP1 is used to display in analog scale the measure point selected through the V130 parameter. The TP1 test point value is available in analog form on pin 3, S9 connector.

By entering into V130 parameter the analog test point conversion factor (see table on chapter 5.6), it is possible to obtain on the test point the measure point selected and automatically the appropriate scale.

For example, selecting on the V130 the measure point code 001 (mechanical motor speed) and selecting 163840 value on the V134 (conversion factor - see table chapter 5.6) it is possible to obtain 1V output = 1000 Rpm on the pin 3, S9 connector.

Example

We want to obtain 2V output =1000 Rpm on the pin 3 :

- Select V130 = 001 code (mechanical motor speed selection)
- Conversion factor = 163840
- Tp measure point analog scale = 1V - 1000 Rpm
- Multiply the conversion factor value * 2 = 163840 * 2 = 327680
- Insert V134 = 327680 to obtain 2V output =1000 Rpm.

V135 Multiplication factor for TP2

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V135	Multiplication factor for TP2	[ ]	$-2^{31} \div +(2^{31}-1)$	107374182	SM, AM, LM	NO / 2

The multiplication factor for TP2 is used to display, in analog scale, the measure point selected through the V131 parameter.

The TP2 test point value is available in analog form on pin 1, S9 connector.

To use the V135 parameter see the description provided for V134.

V136 Offset for analog TP1 test points

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V136	Offset for analog TP1 test points	[ ]	$-2^{31} \div +(2^{31}-1)$	0	SM, AM, LM	NO / 2

The offset for TP1 can introduce a bias to the TP1 test analog test point.
The function can display with analog resolution scale the drive measure point (MP) with average values different from zero.

The analog test points are used in some particular cases, to display in the best way the drive quantities, and it is necessary to add to the quantities an offset before transforming them to an analog signal. V136 parameter concerns the test point 1 (pin 3 of the S9 connector).

Example

Optimized display of ripple present on speed signal with the motor running at high average speed, i.e. at 1000 Rpm. Without using V136 parameter it is necessary to use a suitable scale on the analog test point in such a way to obtain, at 1000 Rpm, an analog value lower than 10V (9V scale = 1000 Rpm).

Obviously, a ripple of few Rpm's on speed signal, cannot be evaluated, as it corresponds to a voltage variation of few tens of mV.
Using an offset, it is possible to obtain the analog value 0, even in case of average speeds different from 0.

Knowing the speed scale in program units, it is possible to setup V136 or V137 parameters with the speed value in program units at 1000 Rpm (reversed sign) and to use a larger scale on the analog test point so that a ripple of few Rpm corresponds to a voltage variation of several Volts.

- speed scale: 1 Rpm = 1342.18 in program units
- 1000 Rpm = 1342180 in program units
- current scale = maximum current deliverable by the drive = 13107

Entering the value (+1342180) in V136 parameter at a speed of 1000 Rpm, on the analog output of test point 1 we will obtain 0V.

In this case we will have the possibility to use a test point scale 1000V = 1000 Rpm so that a variation of 1 Rpm will correspond to an analog variation of 1V, thus obtaining an optimized display of the speed ripple.

If the measure point scales are unknown, we suggest to use the parameters V138 (Wrap-around TP1/2)

V137 Offset for analog TP2 test points

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V137	Offset for analog TP2 test points	[ ]	$-2^{31} \div +(2^{31}-1)$	0	SM, AM, LM	NO / 2

The offset for TP2 can introduce a bias to the TP2 analog test point.
The function can display with analog resolution scale the drive quantities with average values different from zero.

The analog test points are used in some particular cases to display in the best way the drive quantities, and it is necessary to add to the quantities an offset before transforming them to an analog signal.

The V137 parameter concerns TP2 (pin 1 of the S9 connector).

To use the V137 parameter see the description provided for V136 parameter.

V138 Configuration of analog TP

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V138	Configuration of analog TP	Bit	0 ÷ F	C	SM, AM, LM	NO / 2

The V138 parameter allows to keep the right visualization of the most important measure point signal part, this maintaining the correct TP scale selection (Wrap-around function).

Setting bit0 or bit1 is possible to make the Wrap-around configuration on TP1 or TP2.
Setting bit2 or bit3 is possible to enable the output on TP1, TP2.

N° Bit	Bit function	Bit value = 0	Bit value = 1	Default value
0	Wrap-around TP1	Wrap-around not active	Wrap-around not active	0
1	Wrap-around TP2			0
2	Output TP1 (1)	Enabled	Disabled	1
3	Output TP2 (1)	Enabled	Disabled	1

V139 Configuration of Programmable Relays

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V139	Configuration of Programmable Relays	Bit	0 ÷ FF	BC	SM, AM, LM	YES / 2

By means of the programmable relays, it is possible to control the brake management, control quantities using two thresholds, logic intervention and a time constant.
The drive has two programmable relays outputs: PROGO1 and PROGO2 on pins 2/3 of drive's S7 connectors.

The V139 parameter sets the programmable relays for different application by means of bit configurations (see table below). The bits use and descriptions are shown on following pages.

N° Bit	Bit function	Bit value = 0	Bit value = 1	Default value
0	Brake management	Brake management not active	Brake management active	0
1	Brake management output on PROGO1 or PROGO2	Brake management output on PROGO1	Brake management output on PROGO2	0
2	Absolute value for prg. relay no. 1	Not absolute value (Relative)	Absolute value	1
3	Absolute value for prg. relay no. 2			1
4	Prg. relay no.1 mode	Threshold mode	Band mode	1
5	Prg. relay no.2 mode			1
6	Prg. relay no.1 logic	Negative logic	Positive logic	0
7	Prg. relay no.2 logic			1
8, 9	Not used			
10	Logic of limit switches	Normal close	Normal open	0

- **Motors with brake management only:**
  If the limit switch on $Axis_n$ is used, it is not possible to use PROGOx for the brake management. Please also refer to the S7/S8 connector on NUMDrive C Installation manual.

- **Motors with brake management only:**
  If one PROGOx is used to manage the brake, only the other could be linked to the measure point (MP). In Bi-Axes configuration V22 must be taken into account.

- **Motors with range change only:**
  PROGO1 / PROGO2 are used for control the low and high range of the motor and V139 is not take into account.
  If bit 3 of V30 is setted to 1 the range change is disable and PROGO1 / PROGO2 are controlled by V139 parameter.

- **Default values of programmable relays 1 and 2 PROGO1/2 :**
  Relay1 = relay output high(1) with motor running at a speed >30 Rpm
  Relay2 = relay output high(1) with motor running at a speed ranging from 30 to 100 Rpm.

(1) Output high means: on pin 2 or 3 of S7 connector a +24Vdc is present.

V139 Use of the configuration bits

Bit 0 = Brake management [1=active]

Bit 0 allows to obtain the motor holding brake management in case of a alarm. The default value is 0. See utilisation on the chapter 7.		
N° Bit	Bit function	Default value
Bit 0 = 0	Holding brake management not active.	0
Bit 0 = 1	Holding brake management active.	

Bit 1 = Brake management output on [0 = PROGO1, 1 = PROGO2]

Bit 1 allows to select the association between the programmable relay and brake management. (Output on pin2 PROGO1 or on pin 3 PROGO2 of S7 connector) The defaul value is 0.		
N° Bit	Bit function	Default value
Bit 1 = 0	Brake output on PROGO1	0
Bit 1 = 1	Brake output on PROGO2	

Bit 2 = Absolute value for programmable relay 1

Bit 2 allows to define the logic operation (absolute or not absolute) of the relay 1 considering either the absolute or not absolute control value of the selected quantity on the relay.		
N° Bit	Bit function	Default value
Bit 2 = 0	Not absolute value	1
Bit 2 = 1	Absolute value	

Bit 3 = Absolute value for programmable relay 2

Bit 3 allows to define the logic operation (absolute or not absolute) of the relay 2 considering either the absolute or not absolute control value of the selected quantity on the relay.		
N° Bit	Bit function	Default value
Bit 3 = 0	Not absolute value	1
Bit 3 = 1	Absolute value	

Bit 4 or 5 = Programmable relay 1 / 2 mode -

Bit 4 or 5 allows to define the logic operation of the programmable relay 1 or 2 as either threshold or band mode. Bit 4 PROGO1, Bit 5 PROGO2.		
N° Bit	Bit function	Default value
Bit 4 or 5 = 0	Threshold mode	1
Bit 4 or 5 = 1	Band mode	

- **Threshold mode : bit 4 or bit 5 = 0**

  The working range of the selected quantity is monitored.
  The programmable relay switches ON / OFF as the quantity crosses the band created by the two thresholds.
  The relay switches ON as the quantity exceeds the value of threshold 1 and OFF as it falls below the value of threshold 2. The relay switching will occur only if the condition is present for a time longer than value set in the V144 parameter.

- **Band mode: bit 4 or 5 = 1**

  In this mode the programmable relay switches OFF when the quantity value is outside the band created by the two thresholds. The relay switching will occur only if the condition is present for a time longer than value set in the V144 parameter.

Bit 6 or 7 = Logic of prog. relay 1 [1=positive, 0=negative]

Bit 6 or 7 sets the output status condition of the programmable relay 1/2 threshold mode with negative or positive logic control. Bit 6 relay1 PROGO1, Bit 7 relay2 PROGO2		
N° Bit	Bit function	Default value
Bit 6 or 7 = 0	Negative logic	1
Bit 6 or 7 = 1	Positive logic	

V139 Use of the Programmable relays

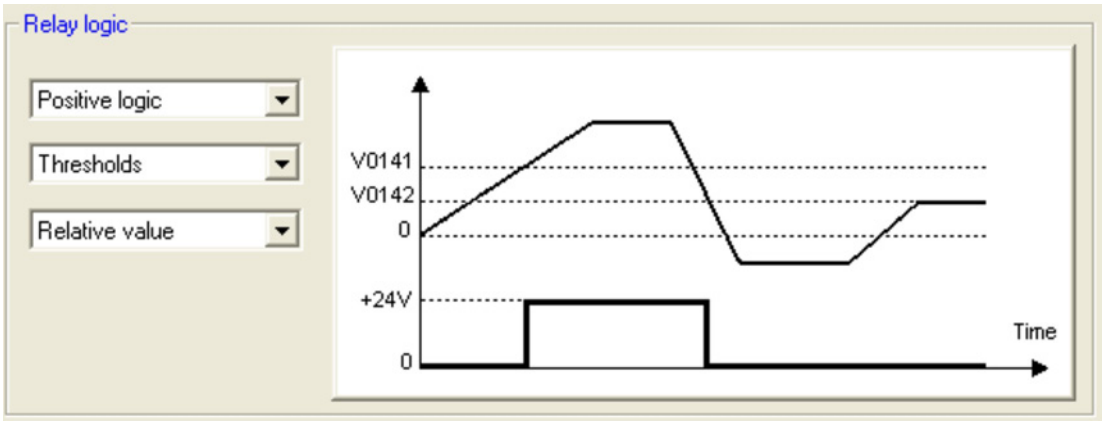
PROGO1 use in modality brake management

- Bit 0 = 1
- Bit 1 = 0

For more informations on the modality brake management please refer to chapter 11.3.

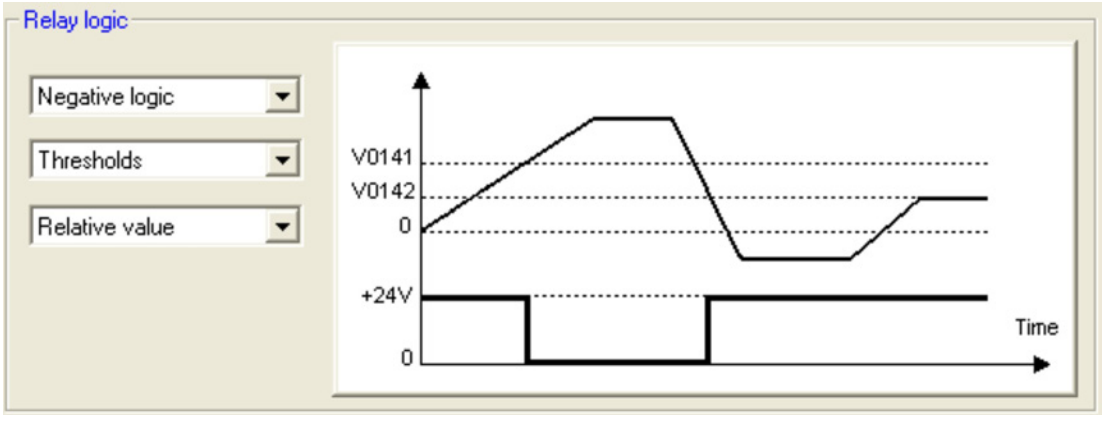
PROGO1 use in modality Positive logic, Thresholds, Relative value

- Bit 6 = 1
- Bit 4 = 0
- Bit 2 = 0



PROGO1 use in modality Negative logic, Thresholds, Relative value

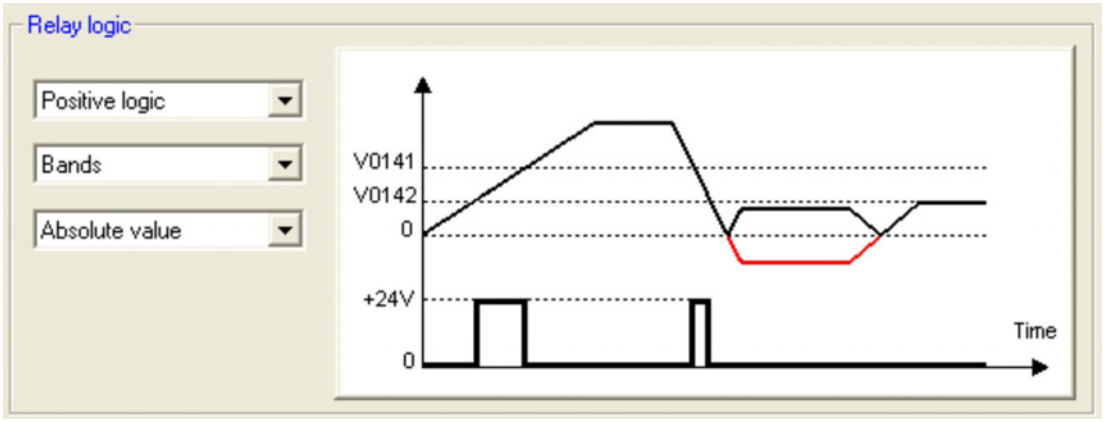
- Bit 6 = 0
- Bit 4 = 0
- Bit 2 = 0



The above examples are available only if the brake management and the motor range change are not used.
These examples may be used for PROGO2 too. See bits 3, 5, 7.

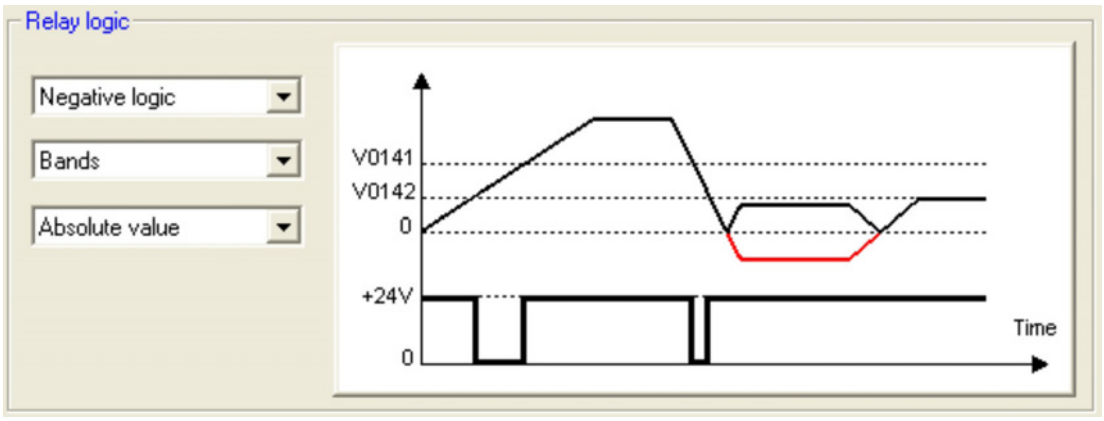
PROGO1 use in modality Positive logic, Bands, Absolute value

- Bit 6 = 1
- Bit 4 = 1
- Bit 2 = 1



PROGO1 use in modality Negative logic, Bands, Absolute value

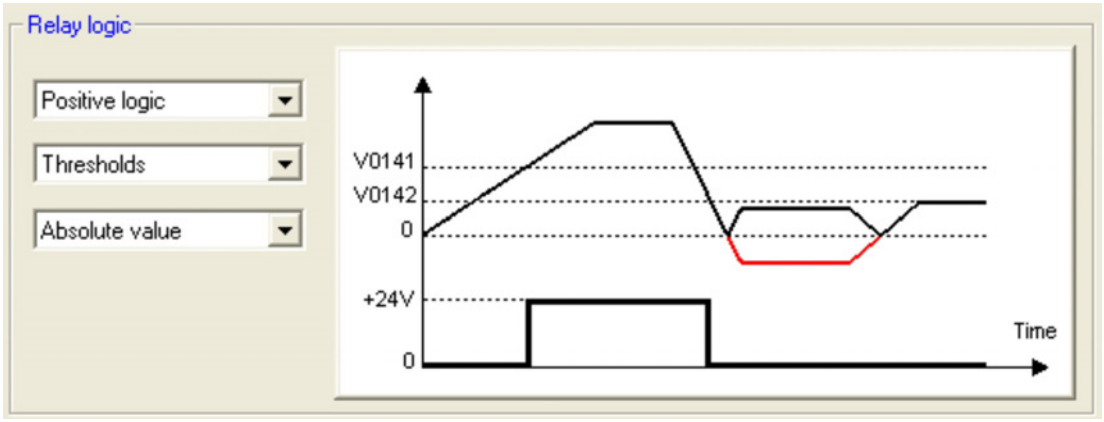
- Bit 6 = 0
- Bit 4 = 1
- Bit 2 = 1



The above examples are available only if the brake management and the motor range change are not used.
These examples may be used for PROGO2 too. See bits 3, 5, 7.

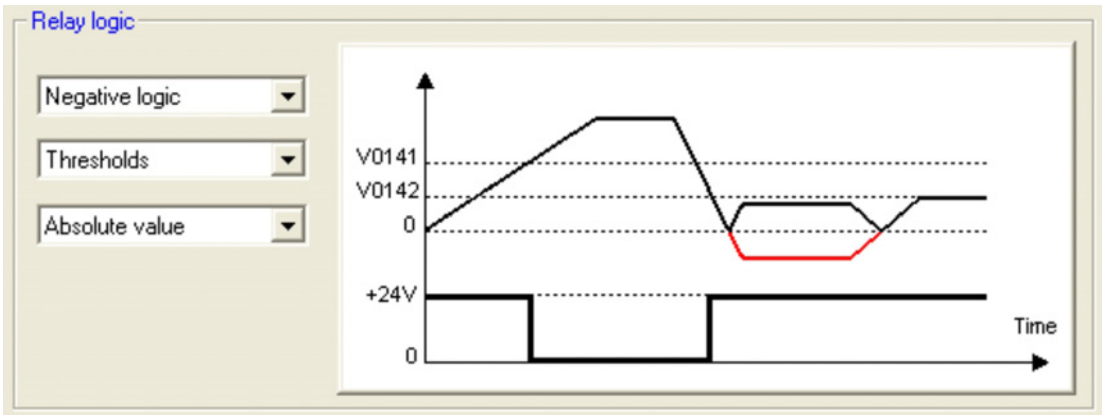
PROGO1 use in modality Positive logic, Thresholds, Absolute value

- Bit 6 = 1
- Bit 4 = 0
- Bit 2 = 1



PROGO1 use in modality Negative logic, Thresholds, Absolute value

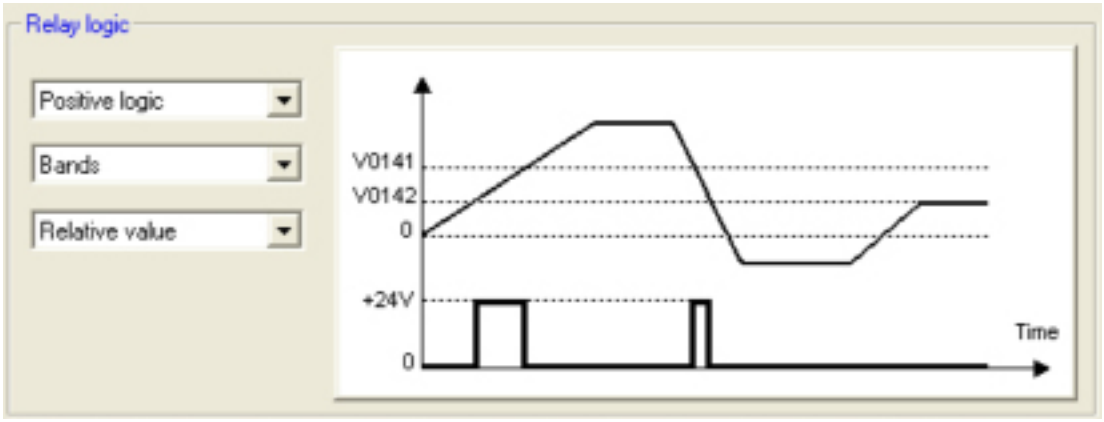
- Bit 6 = 0
- Bit 4 = 0
- Bit 2 = 1



The above examples are available only if the brake management and the motor range change are not used.
These examples may be used for PROGO2 too. See bits 3, 5, 7.

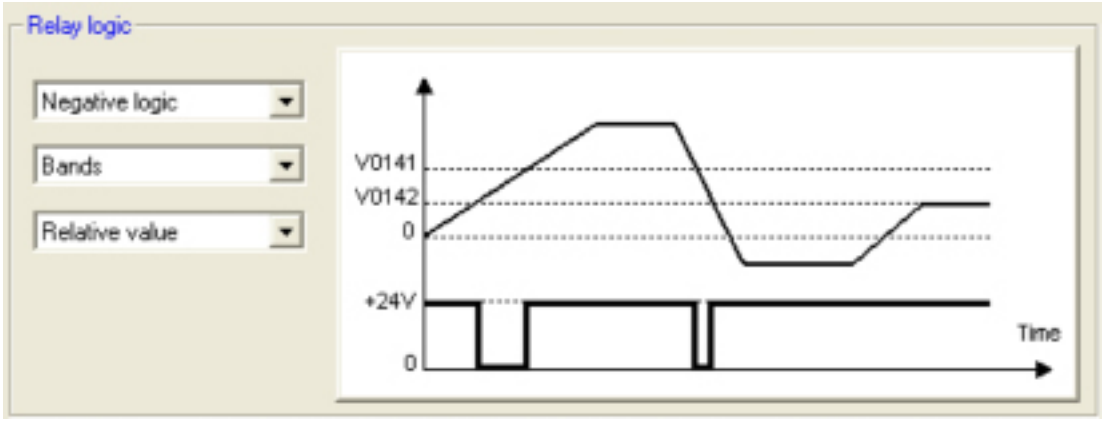
PROGO1 use in modality Positive logic, Bands, Relative value

- Bit 6 = 1
- Bit 4 = 1
- Bit 2 = 0



PROGO1 use in modality Negative logic, Bands, Relative value

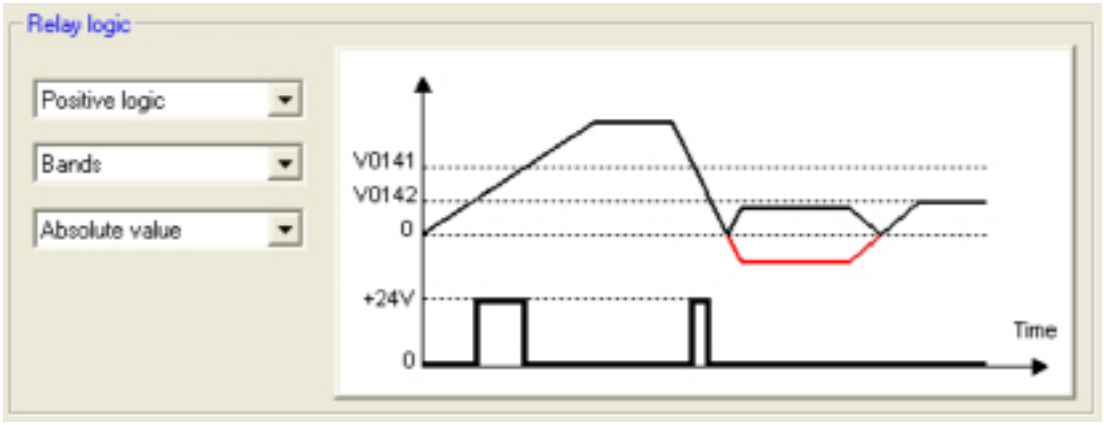
- Bit 6 = 0
- Bit 4 = 1
- Bit 2 = 0



The above examples are available only if the brake management and the motor range change are not used.
These examples may be used for PROGO2 too. See bits 3, 5, 7.

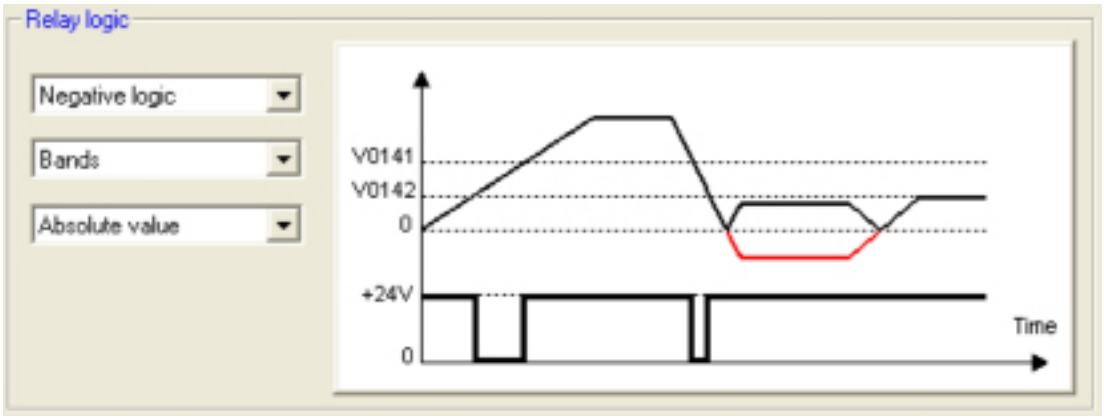
PROGO1 use in modality Positive logic, Bands, Absolute value

- Bit 6 = 1
- Bit 4 = 1
- Bit 2 = 1



PROGO1 use in modality Negative logic, Bands, Absolute value

- Bit 6 = 0
- Bit 4 = 1
- Bit 2 = 1



The above examples are available only if the brake management and the motor range change are not used.
These examples may be used for PROGO2 too. See bits 3, 5, 7.

V140 Associate MP with programmable relay 1

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V140	Associate MP with programmable relay 1	[ ]	1 ÷ 500	1	SM, AM, LM	NO / 2

The V140 programmable relay 1 test point can associate a measure point MP to the programmable relay 1. The measure point and programmable relay associations is shown on chapter 5.6.

Example.

We want to associate to the relay 1 the measure point code 001 (mechanical motor speed selection)

- Select V140 = 1

For a correct use of the relay, always set correctly the programmable relay 1 scale factor V143 parameter.

V141 Programmable relay 1 threshold 1

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V141	Programmable relay 1 threshold 1	[ ]	$-2^{31} \div +(2^{31}-1)$	30	SM, AM, LM	NO / 2

The V141 and V142 parameters define the threshold values to be used with V139 parameter.
Please always observe this rule: V141 value must be > V142 value.

Once the measure point to be associated to programmable relay No.1 has been selected and the application mode has been configured using the V139 parameter, it is necessary to setup the values of both thresholds as well as the intervention time constant.

The thresholds 1/2 measurement unit is automatically adapted to the selected quantity.
If, for example, an item representing a speed has been selected, the threshold will be configured in Rpm.

- If the selected measure point is a speed or a speed reference :
  enter the thresholds in Rpm.
- If the selected measure point is an iq, id current or an iq or id current reference :
  enter the thresholds in Arms.
- If the selected measure point MP007 Image I^2t (see chapter 6.3) :
  enter the thresholds in %.
- If the selected measure point is a vq, vd, bus voltage :
  enter the thresholds in Volts.

For all quantities (i.e. position, position reference ...) for which is impossible to establish the correspondence between physical unit and internal unit, a unit in V142 and V143 parameters corresponds to a program internal unit of the selected measure point.

Further information concerning the relay scale factor values and measure point unit measurement are shown on the chapter 5.6.

V142 Programmable relay 1 threshold 2

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V142	Programmable relay 1 threshold 2	[ ]	$-2^{31} \div +(2^{31}-1)$	0	SM, AM, LM	NO / 2

To use V142 parameter see the description provided for V141.

V143 Programmable relay 1, MP scale factor

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V143	Programmable relay 1, MP scale factor	[ ]	$-2^{31} \div +(2^{31}-1)$	1342.18	SM, AM, LM	NO / 2

The measure point scale factor of the programmable relays 1/2 is to be entered in the V143 and V153 parameters.

The correct scale factor settings define the measurement units for the V141/V142 and V151/V152 parameters. See chapter 5.6 for the standard setting values.

Example Association of the mechanical speed to programmable relay1

- Select V140 parameter and enter 1
  (MP code = speed)
- Select V143 parameter and enter 1342.18
  (scale adjust.)
- Check the scale factor in Rpm on the parameters V141/142
  (threshold1/2)

Example MP002 (iq current) reference scale factor calculation using MDLU3075 module.

- The scale factor value is equal to : $\frac{18536.097}{75} = 247.$
- The scale factor value is equal to : $\frac{18536.097}{\text{module size (A)}}$

Example. MP007 (Imagine I^2t) scale factor calculation using MDLU3014 module and V108 = 4A (Motor nominal current)

- The scale factor value is equal to : $\frac{4^2 * 419.4}{14^2} = 34.24$
- The scale factor value is equal to : $\frac{\text{Nominal current}^2}{\text{module size } (A^2)} * 419.4$

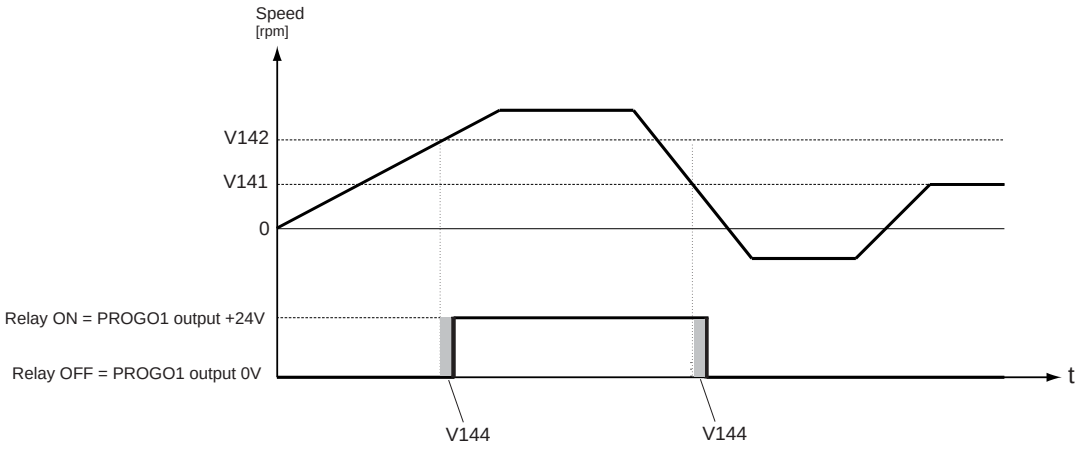
V144 Programmable relay 1 tripping time constant

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V144	Programmable relay 1 tripping time constant	ms	5 ÷ 1500	15	SM, AM, LM	NO / 2

The programmable relay time constant manages the relay ON / OFF switching and viceversa. When the value of the quantity controlled has reach for a time longer than value set in the V144 parameter the relay will switch ON or OFF.

The relay thresholds are selected by the V142 / V142 parameters.
The programmable relay time constant is expressed in ms.

See relay utilisation on chapter 5.7.2.



V150 Associate MP with programmable relay 2

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V150	Associate MP with programmable relay 2	[ ]	1 ÷ 500	1	SM, AM, LM	NO / 2

To use the V150 parameter see the description provided for the V140 parameter.

V151 Programmable relay 2, threshold 1

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V151	Programmable relay 2, threshold 1	[ ]	$-2^{31} \div +(2^{31}-1)$	100	SM, AM, LM	NO / 2

To use the V151 parameter see the description provided for the V141 parameter.

V152 Programmable relay 2, threshold 2

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V152	Programmable relay 2, threshold 2	[ ]	$-2^{31} \div +(2^{31}-1)$	30	SM, AM, LM	NO / 2

To use the V152 parameter see the description provided for the V142 parameter.

V153 Programmable relay 2, MP scale factor

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V153	Programmable relay 2, MP scale factor	[ ]	$-2^{31} \div +(2^{31}-1)$	1342.18	SM, AM, LM	NO / 2

To use the V153 parameter see the description provided for the V143 parameter.

V154 Programmable relay 2 tripping time constant

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V154	Programmable relay 2 tripping time constant	ms	5 ÷1500	15	SM, AM, LM	NO / 2

To use the V154 parameter see the description provided for the V144 parameter.

V160 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V160						

V165 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V165						

V166 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V166						

V169 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V169						

V170 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V170						

V171 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V171						

V172 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V172						

V173 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V173						

V175 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V175						

V176 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V176						

V180 Current percentage (V108) for EPS

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V180	Current percentage (V108) for EPS	%	0 ÷ (1)	100	SM, AM, LM	YES/ 2

$$(1)\ \max = \frac{100\% * I_{nom.\ module}}{V108}$$

The V180 paramente allows to set up the medium current value which has to be reached to obtain a correct phase value for the EPS function.
(EPS=Electrical Position Sinchronization without moviment).

The parameter is expressed in percentage of the nominal current motor (see parameter V108).
The default value suggested is 50%. However, if necessary, increase the value step of 10%.

V182 U/V Phase relative weight for EPS

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V182	U/V Phase relative weight for EPS	%	0 ÷ 200	100	SM, AM, LM	YES / 2

The V182 parameter improve the U/V phases relative weight during the EPS function.
(EPS=Electrical Position Sinchronization without movement).

The parameter is expressed as a percentage. The minimum value is 0% and the maximum is 200%.
The default value suggested is 100%.

The parameter has to be modified if there is a great unbalance in the U/V/W phases inductance.
This effect is usually noticeed in presence of linear motors.

V183 Phases V/W relative weight for EPS

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V183	Phases V/W relative weight for EPS	%	0 ÷ 200	100	SM, LM	YES / 2

The V183 parameter improve the V/W phases relative weight during the EPS function.
(EPS = Electrical Position Sinchronization without movement).

The parameter is expressed in percentage. The minimum value is 0% and the maximum is 200%. The default value suggested is 100%.

The parameter has to be modified if there is a great unbalance in the U/V/W phases inductance. This effect is usually noticed in presence of linear motors.

V190 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V190						

V191 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V191						

V205 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V205						

V206 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V206						

V207 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V207						

V208 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V208						

V209 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V209						

V210 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V210						

V217 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V217						

V218 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V218						

V220 Reference configuration

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V220	Reference configuration	Bit	0 ÷ FF	4	SM, AM, LM	YES / 2

N° Bit	Bit function	Bit value = 0	Bit value = 1	Default value
0, 1	Clear items see table 1			0
2	Interpolator active	Interpolator not active	Interpolator active	1
3	Interpolator type	1st order	2nd order	0
4	Reversal of the speed reference	Reference not reversed	Reference	0
5	Not used			
6	Reversal of the current reference	Reference not reversed	Reference reserved	0
7	Interpolation of current reference V123	Interpolation not enable	Interpolation enable	0

Bit 1	Bit 0	Table 1 - Clear quantities
0	0	No clear
0	1	Clear of position regulator output
1	0	Clear of interpolated position reference
1	1	Clear of speed regulator output

For a correct use of the configuration bits see the next page.

V220 Use of the configuration bits

N° Bit	Bit function Reference configuration	Bit utilization
0, 1	Clear items	See table 1.
2	Interpolator active	Always leave the interpolator activate.
3	Interpolator type	Always use the interpolator of 1°order.
4	Reversal of speed reference	It is used when with an additional position sensor, the position loop is in a positive reaction (following error on the first movement). Set to 1 the bit 4 if using the same sensor to close the speed and position loop, and if the bit 9 = 1 in the parameters V348 and V349 is selected.
5	Not used	
6	Reversal of current reference	Always leave the bit to 0, if the speed and current loops are closed with the same sensor (see parameter V351 / V352 in standard configuration). If the speed and current loops are closed with different sensors which have not the same cyclic direction, (the position of one sensor increases while the position of the other decreases)it is therefore necessary to select the bit to 1 to reverse the current reference. If the above described conditions are not followed, the motor will be “racing”.
7 (1)	Interpolation of current reference V123	V220 bit 7 is used to activate the «fine interpolation» on the current reference used by current loop 0 = no fine interpolation active, 1 = current reference fine interpolation active such fine interpolation on the current reference used by current loop is parametrized by mean V123.

(1) **V123** : Step of computation (pas de calcul) of the fine interpolation on current reference:
- **min = 1** The fine-interpolator averages two current references delayed by 50 µs each other,so doing it implements a FINE INTERPOLATOR on current-loop-reference.
- **def = 2** The fine-interpolator averages two current references delayed by 100µs each other, so doing it implements a «filter» to make same phase lag (50µs) and same current-loop response magnitude (-6dB @ 1600Hz)
- **max = 3** The fine-interpolator averages two current references delayed by 150µs each other, so trying to eliminate the 3.4KHz resonance due to Stegmann encoder coupling to NUM motors.

It exists also a measure point displaying the result of the «fine interpolation» of the current reference:
- MP048: displays the «interpolated Iq reference», it is the output of the current-loop-reference fine interpolation described above. The input of such a filter is the usual MP002 (Iq reference)
  Obviously if bit V220.7 is 0, then the two MP coincide, apart a little displaying delay of 50µs.
  Such displaying delay does not influence the current-loop behaviour, it is only a MP displaying issue.
  The current-loop uses the interpolated Iq reference without any delay.

V221 Enable configurations

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V221	Enable configuration	Bit	0 ÷ 1F	0	SM, AM, LM	YES / 2

N° Bit	Bit function	Bit value = 0	Bit value = 1	Default value
0	Torque enable	From CANopen command (PLC program)	With GP_IN_A or B digital input (see V225)	0
1	Reference enable			0
2	Torque AND Digital bus enable	Not active	See note below	0
3	Reference AND Digital bus enable			0
4	Change of range mode (1)	Not active	GP_ IN_A or B mode	0

(1) The motor change of range is enable with:
- Bit 4 = Not active.
- Bit 4 = 1 using V225 bit 6 configuration.

Note. It is possible to configure the drive in such a way as to obtain the torque or the reference speed enable only if both PLC and GP_IN_A / B (see pins 4,5 S7 connector) logical input commands are simultaneously present.

Example

To obtain the torque enable only when both PLC + GP_IN_A / B commands are simultaneously present, set V221 bit 0 = 1 and bit 2= 1.

V225 Digital input configuration [GP_IN_A or B]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V225	Digital input configuration [GP_IN_A / B]	[ ]	0 ÷ 7F	C	SM, AM, LM	YES / 2

N° Bit	Bit function	Bit value = 0	Bit value = 1	Default value
0	Torque enable	With GP_IN_A digital input (pin 4 S7connector)	With GP_IN_B digital input (pin 5 S7connector)	0
1	Reference enable			0
2	S1 HOMING switch			1
3	S2 HOMING switch			1
4	S1 HOMING switch logic reversed	Not reversed	Reversed	0
5	S2 HOMING switch logic reversed			0
6	Change of range request	GP_IN_A	GP_IN_B	0

V258 Motor sensor type

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V258	Motor sensor type	[ ]	0 ÷ Z	X	SM, AM, LM	YES / 2

The drive can control different types of sensor. The type of motor sensor to be used may be configured by means of the table below. We suggest to use V258 parameter in presence of new type of sensor only. Otherwise, make this operation by WPM. (see chapter 5.1)

Sensor type	Sensor description example
0	Not configured
A	High resolution Multiturn encoder (absolute) - 512
B	High resolution Singleturn encoder (incremental) - 512
P	High resolution Multiturn encoder (absolute) - 1024 [2 gen.]
Q	High resolution Singleturn encoder (incremental) - 1024 [2 gen.]
H	Wheel 256 teeth
C	Wheel 256 teeth with Top
V	Wheel 256 teeth with Top [screw fix.]
G	Wheel 512 teeth with Top
W	S2 SENSOR
X	S1 SENSOR
J	SinCos SKM 36 Multiturn encoder (absolute) - 128
K	SinCos SKS 36 Singleturn encoder (incremental) -128

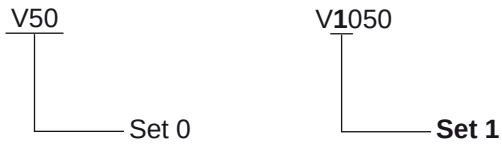
V260 Speed regulator Parameters exchange

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V260	Speed regulator Parameters exchange	[ ]	0 ÷ 1	0	SM, AM, LM	NO / 2

The V260 parameter allows to select one particular speed loop parameters set of the two available This selection is possible with WPM.

The speed loop parameters set ranges from V50 to V99 parameters.
The sets 0 or 1 of available parameters are defined by the thousand's digit indicated on the specific parameter.

Example



- Selecting 0 on the V260 parameter will automatically select the parameter set 0.
- Selecting 1 on the V260 parameter will automatically select the parameter set 1.

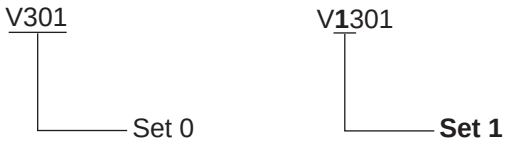
V261 Position regulator Parameters exchange

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V261	Position regulator Parameters exchange	[ ]	0 ÷ 1	0	SM, AM, LM	NO / 2

The V261 parameter allows to select any particular position loop parameters set.
This selection is possible with WPM.

The position loop parameters range from V301 to V399 parameters.
The sets 0,1... of available parameters are defined by the thousand's digit indicated on the specific parameter.

Example



- Selecting 0 on the V261 parameter will automatically select the parameter set 0.
- Selecting 1 on the V261 parameter will automatically select the parameter set 1.

V266 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V266						

V268 Electrical phase offset Hall effect sensor

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V268	Electrical phase offset Hall effect sensor	mDegree	0 ÷ 359999	0	SM, LM	YES/ 2

The V268 parameter sets the electrical phase offset between the Hall sensor and the 0 motor electrical phase. (U motor phase).

V269 Position offset for electrical phasing

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V269	Position offset for electrical phasing	mDeg	0 ÷ 359999	0	SM, LM	YES / 2

The V269 parameter allows to modify the electrical phasing. For NUM motors the V269 parameter value must be 0. This parameter is also used for motors having EPS function.

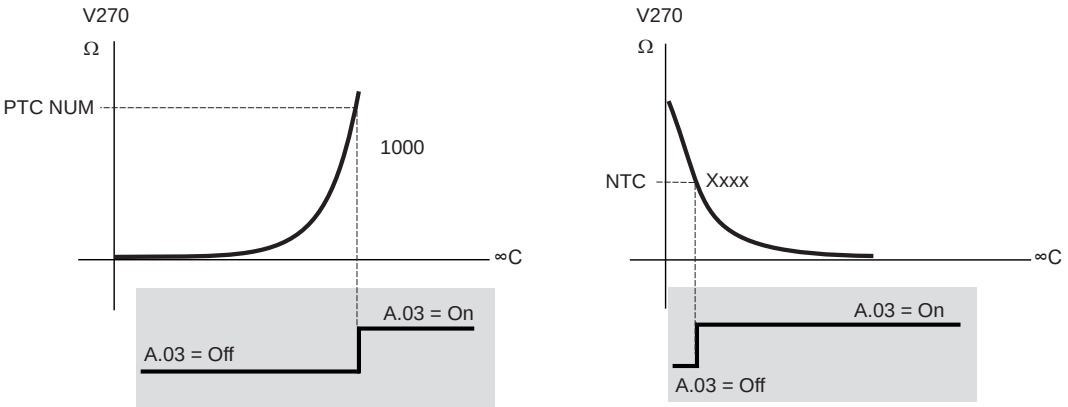
With EPS in use if the parameter value is 0 (zero), and if HOMING is required from the system, on the drive will appear A18.04 alarm.

V270 Motor PTC-NTC value

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V270	Motor PTC-NTC value	Ohm	0 ÷ 7000	1000	SM, AM, LM	YES / 2

The V270 parameter sets the alarm threshold of A.03 motor overtemperature alarm as indicated below. The V270 parameter value is expressed in Ohms.

- For NUM motors BPH, BPG, BPL and BPX055 with PTC the thermal probe default value is 1000. (See use example as indicated in figure below)
- For NUM motors BPX, BHX the thermal probe default value is 1340.
- For non NUM motors check the PTC or NTC thermal probe technical data sheet and insert the correct Ohms values on the V270 parameter.



For the thermal probe configuration see V272 parameter.

V272 Motor thermal sensor configuration

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V272	Motor thermal sensor configuration	[ ]	0 ÷ 7	0	SM, AM, LM	YES / 2

The V272 parameter sets the motor thermal sensor type as indicated in the table below.

N° Bit	Bit function	Bit value = 0	Bit value = 1	Default value
0	Sensor Type	PTC sensor	NTC sensor	0
1	Sensor wiring	Wiring on S1	Wiring on S2	0
2	Motor temperature sensor control NTC open	Control not active	Control active	0

V273 Electrical phase offset management

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V273	Electrical phase offset management	[ ]	0 ÷ 7F	0	SM, AM, LM	YES / 2

The V273 parameter sets the electrical phase offeset as indicated il the table below.

N° Bit	Bit function	Bit value = 0	Bit value = 1	Default value
0	Phase offset exchange after EPS/Hall + HOMING	Offset exchange enabled	Offset exchange not enabled	0
1	CNC On-line electrical phase offset tuning	On-line electrical not enabled	On-line electrical enabled	0
2	APEM Automatic phase offset exchange on mark	APEM not enabled	APEM enabled	0
3	Exclusion of selfcalibration alarm A18.3 [1=Excluded]	Not excluded	Excluded	0
4	Forced EPS with absolute encoders	Forced EPS not enabled	Forced EPS enabled	0
5	Exclusion of selfcalibration alarm A18.2	Not excluded	Excluded	0
6	EPS for highly anisotropic motors	EPS not enabled	EPS enabled	0

V301 Proportional position gain [0]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V301	Proportional position gain [0]	mm/min/mm	0 ÷ 50000	1000	SM, AM, LM	NO / 2

The V301 parameter allows to adjust the position loop gain.
The parameter is expressed in mm/min/mm.

If a different sensors for speed loop closure is used to obtain a correct gain it is necessary to properly set the V353 parameter.
To optimize the adjustment of proportional position gain it is necessary to optimize first the speed loop gains.

V306 S1 sensor MULTI coefficient [0]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V306	S1 sensor MULTI coefficient [0]	[ ]	- 32767 ÷ +32767	1	SM, AM, LM	YES / 2

The MULTI / DIVI parameters are measurement conversion coefficients.
The MULTI (V306) / DIVI (V307) parameters convert the feedback provided by motor position sensor to the internal measurement units needed by the application selected as “NC Units” .
(See also “NC Units” on chapters 5.2.1, 5.3.1)

The V306 and V307 parameters are needed when the S1 sensor is used to close the position loop as well as in the AP01 special application during the initialization of absolute position provided by S1 sensor.

Enter as V306 (MULTI) parameter an axis displacement in “NC Units” .
Enter as V307 (DIVI) parameter the number of feedback counts needed to carry out the displacement indicated in V306 parameter.
The measure in “NC Units” (Measure point MP Point 215) is:

$$\frac{\text{S1 sensor measurement} * \text{V306}}{\text{V307}}$$

Parameters V1306 and V1307 replace V306 and V307 parameters when parameter set 1 is activated.
The maximum value for V306 and V307 parameter is 32767.

If the values to be used are higher, it is advisable to simplify the ratio by eliminating the factors common to numerator and denominator. By selecting the negative sign (-) on V306 parameter (ex. -10.000) the measurement direction is inverted.

The inversion of measurement direction must be used if the measurement decreases when the axis moves in the positive direction.

V307 S1 sensor DIVI coefficient [0]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V307	S1 sensor DIVI coefficient [0]	[ ]	1 ÷ 32767	1	SM, AM, LM	YES / 2

To use the V307 parameter see the description provided for the V306 parameter.

V310 S1 sensor offset in Controller physical unit [0]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V310	S1 sensor offset in Controller Phys. Unit [0]	[ ]	$-2^{31} \div +(2^{31}-1)$	0	SM, AM, LM	YES / 2

The V310 parameter is active when the S1sensor is used to close the positioning loop as well as in the AP01 special application, during the initialization of absolute position performed by S1 motor sensor.

The value of the V310 parameter is added to the position value in “NC Units”.
(See also “NC Units” on chapters 5.2.1, 5.3.1)

Application.

- S1 sensor managed as incremental for the Controller it is not used.
  During the HOMING execution (machine origin detection) the object 607Ch or 687Ch is assigned to the position value in “NC Units”.

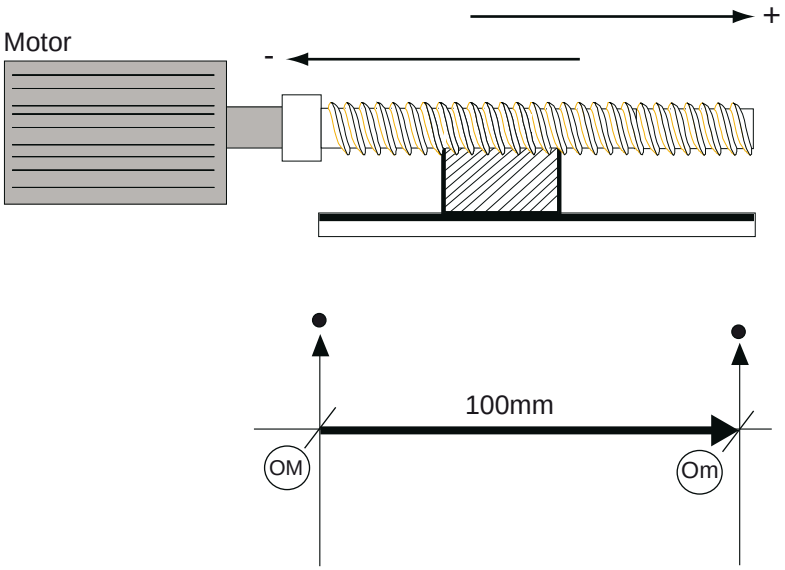
- S1 sensor managed as absolute for the Controller (V348 bit6 =1) :
  The value of V310 is added to the position value in Controller physical unit during the initialization of the absolute position automatically performed at drive power on and after every RUN command or by means the alarms reset.

AP01 special application.
During the initialization of absolute position by S1 sensor the V310 parameter is used as described in the second case.

V310 Application example S1 sensor type P (High resolution Multiturn encoder)

If the machine zero does not correspond to zero encoder it is necessary to enter in V310 parameter the measure read on the Controller when the axis is on the desired machine zero position.

The measurement sign must be the opposite of measurement read on the Controller in machine zero position.



(OM) Position due to zero sensor position
(Om) Machine zero position

- If the CNC measurement unit is expressed in µm enter the value +100000 in the V310 parameter.
- The offset value must always be entered using the “NC Unit”.
  i.e: if the “NC Unit” is the µm, then the offset value must be expressed in µm.

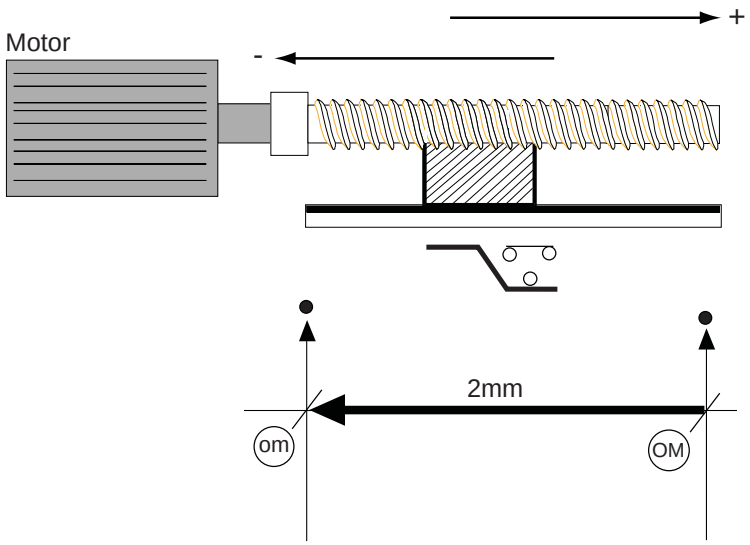
V310 Application example S1 sensor type Q (High resolution Singleturn encoder)

If the machine zero does not correspond to zero encoder it is necessary to enter in V310 parameter the measure read on the Controller when the axis is on the desired machine zero position.

The measurement sign must be the opposite of measurement read on on the Controller in machine zero position.

If the machine zero does not correspond to zero encoder it is necessary to enter in V310 parameter the measure read on the Controller when the axis is on the desired machine zero position.

The measurement sign must be the opposite of measurement read on on the Controller in machine zero position.



OM Position due to zero top

om Machine zero position

- If the “NC Unit” is expressed in µm enter the value +2000 in the V310 parameter.
- The offset value must always be entered using the “NC Unit”.
  i.e: if the “NC Unit” is the µm, then the offset value must be expressed in µm.

V311 S2 sensor MULTI coefficient [0]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V311	S2 sensor MULTI coefficient [0]	[ ]	- 32767 ÷ +32767	1	SM, AM, LM	YES / 2

The MULTI / DIVI parameters are conversion coefficients of the measurement.
The MULTI (V311) / DIVI (V312) parameters allow to adapt the measurement provided by motor position sensor to the internal “NC Unit”.
(See also “NC Units” on chapters 5.2.1, 5.3.1)

The V311 and V312 parameters are needed when the S2 sensor is used to close the position loop as well as in the AP01 special application during the initialization of absolute position provided by S2 sensor.

Enter as V311 (MULTI) parameter the measure of shift performed by the system.
Enter as V312 (DIVI) parameter the counts number of measurement system needed to carry out the shift indicated in V311 parameter. The measure in “NC Unit” (measure point MP 215) is :

$$\frac{\text{S2 sensor measurement} * \text{V311}}{\text{V312}}$$

The maximum setting value of V311 and V312 parameters is 32767.
If the values to be setup are higher it will be necessary to rationalize such values by reducing them to the minimum terms.

The V1311 (MULTI) and V1312 (DIVI) parameters refer to parameter set 1.
Selecting the negative sign (-) on V311 parameter (i.e. -10.000) causes the inversion of measurement direction. If the values to be used are higher, it is advisable to simplify the ratio by eliminating the factors common to numerator and denominator.

The inversion of measurement direction must be used if the measurement decreases when the mobile element moves to the axis positive direction.

V312 S2 sensor DIVI coefficient [0]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V312	S2 sensor DIVI coefficient [0]	[ ]	1 ÷ 32767	1	SM, AM, LM	YES / 2

To use the V312 parameter see the description provided for the V311 parameter.

V313 S2 sensor offset in Controller physical unit [0]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V313	S2 sensor offset in Controller physical unit [0]	[ ]	$-2^{31} \div +(2^{31}-1)$	0	SM, AM, LM	YES / 2

If the machine zero does not correspond to the cam zero it is necessary to enter in V313 parameter the measure read on the Controller when the axis is on the desired machine zero position. The measurement sign must be the opposite of measurement read on the Controller in machine zero position.

The V313 parameter is active when the S2 sensor is used to close the positioning loop as well as during the initialization of absolute position performed by S2 sensor.
The value of V313 parameter is added to the position value in “NC Unit”.
(See also “NC Units” on chapters 5.2.1, 5.3.1)

Application.

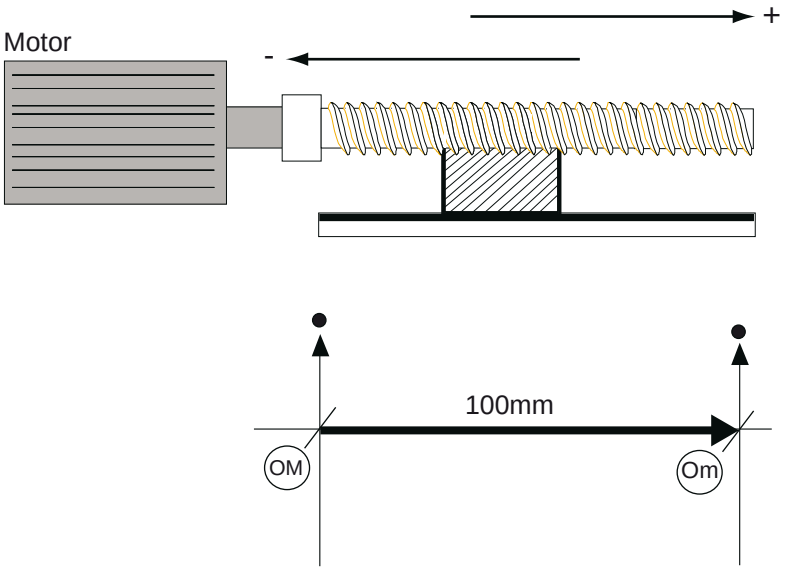
- S2 sensor managed as incremental for the Controller it is not used.
  During the HOMING execution (machine origin detection) the object 607Ch or 687Ch is assigned to the position value in physical unit.

- S2 sensor managed as absolute for the Controller (V349 bit6 =1) :
  The value of V313 is added to the position value in “NC Unit” during the initialization of the absolute position automatically performed at drive power on and after every RUN command or by means the alarms reset.

V313 Application example S1 sensor type P (High resolution Multiturn encoder)

If the machine zero does not correspond to zero encoder it is necessary to enter in V313 parameter the measure read on the Controller when the axis is on the desired machine zero position.

The measurement sign must be the opposite of measurement read on the Controller in machine zero position.



(OM) Position due to zero sensor position
(Om) Machine zero position

- If the NC unit is expressed in µm enter the value +100000 in the V313 parameter.
- The offset value must always be entered using the NC unit. i.e: if the NC Unit is the µm, then the offset value must be expressed in µm.

V314 Sinusoidals signal range of S1 sensor

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V314	Sinusoidals signals range of S1 Sensor	Bit	0 ÷ 3F	0	AM, SM, LM	YES / 2

The V314 parameter is used to select the sinusoidal signals range for a correct use of S1 sensor.

The parameter allows to select the A09.254 alarm threshold which controls the sinusoidal signal absence. (see table below)

N° Bit	Bit function	Bit value = 0	Bit value = 1	Default value
0, 1, 4	Not used			0
2	See table 2 - A09.254 alarm threshold			0
3				0
5	Alarms A09.230/248/254 exclusion	Alarms not excluded	Alarms excluded	0

Bit 3	Bit 2	A09.254 alarm threshold
0	0	50% signal amplitude
0	1	40% signal amplitude
1	0	30% signal amplitude
1	1	20% signal amplitude

V315 Sinusoidals signal range of S2 sensor

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V315	Sinusoidals signals range of S2 Sensor	Bit	0 ÷ 3F	0	AM, SM, LM	YES / 2

The V315 parameter is used to select the sinusoidal signals range for a correct use of S2 sensor.

The parameter allows to select the A38.254 alarm threshold which controls the sinusoidal signal absence. (see table below)

N° Bit	Bit function	Bit value = 0	Bit value = 1	Default value
0, 1, 4	Not used			0
2	See table 2 - A38.254 alarm threshold			0
3				0
5	Alarms A38.230/248/254 exclusion	Alarms not excluded	Alarms excluded	0

Bit 3	Bit 2	A38.254 alarm threshold
0	0	50% signal amplitude
0	1	40% signal amplitude
1	0	30% signal amplitude
1	1	20% signal amplitude

V317 S1 sensor power supply voltage

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V317	S1 sensor power supply voltage	V	1 ÷ 9.20	5	SM, AM, LM	YES / 2

The parameter V317 sets the S1 sensor power supply voltage.
The value of V317 is expressed in Volt.

For a correct use of the sensor before connecting it always take into consideration the sensor data sheet and select the appropriate power supply voltage.

V318 S2 sensor power supply voltage

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V318	S2 sensor power supply voltage	V	1 ÷ 9.20	5	SM, AM, LM	YES / 2

The parameter V318 sets the S1 sensor power supply voltage.
The value of V318 is expressed in Volt.

For a correct use of the sensor before connecting it always take into consideration the sensor data sheet and select the appropriate power supply voltage.

V340 Step of speed calculation S1

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V340	Step of speed calculation S1	[ ]	0 ÷ 4	2	SM, AM, LM	NO / 2

Using the V340 parameter it is possible to change the value of calculation step on the speed loop. The speed calculation step parameter is used in case of high motor noise due to disturbances on speed signal.

The standard value of V340 parameter is 2. Recommended values of V340 depending on sensor type as indicated in the table.

S1 sensor (close the speed loop)	Recommended V340 value
A, B, P, Q encoder	2
J , K encoder	3
Toothed wheel, TTL and SSI sensor	4
Optical linear scale	2

Higher values (max 4) reduce the disturbances on speed signal, but provide a worse dynamic response of the system (reduction of speed loop passband).

ATTENTION : After the modification of V340 parameter it is necessary to repeat the speed loop adjustment operations (V51,V52 and/or V1051,V1052).

V341 Step of speed calculation S2

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V341	Step of speed calculation S2	[ ]	0 ÷ 4	2	SM, AM, LM	NO / 2

Using the V341 parameter it is possible to change the value of calculation step on the speed loop. The speed calculation step parameter is used in case of high motor noise, due to disturbances on speed signal.

The standard value of V341 parameter is 2. Recommended values of V341 depending on sensor type as indicated in the table.

S2 sensor (close the speed loop)	Recommended V341 value
A, B, P, Q encoder	2
J , K encoder	3
Toothed wheel, TTL and SSI sensor	4
Optical linear scale	2

Higher values (max 4) reduce the disturbances on speed signal, but provide a worse dynamic response of the system (reduction of speed loop passband).

ATTENTION : After the modification of V341 parameter it is necessary to repeat the speed loop adjustment operations (V51,V52 and/or V1051,V1052).

V342 Speed filter by S1 sensor

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V342	Speed filter by S1 sensor	Hz	50 ÷ 3000	600	SM, AM, LM	NO / 2

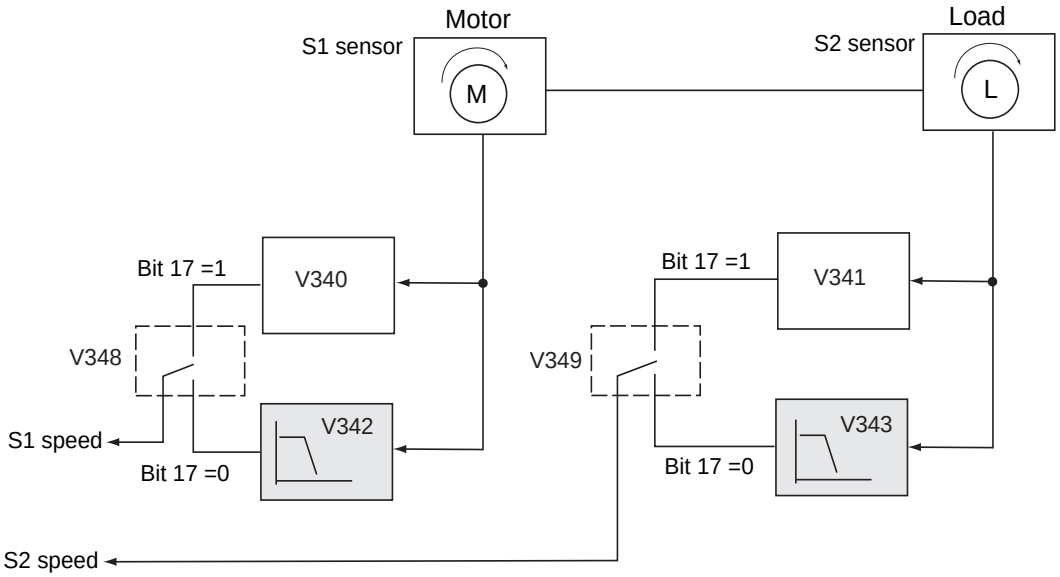
The V342 parameter allows to set up the speed filter by S1 sensor and has to be enabled by means V348 bit 17 = 0. Recommended values of V342 depenting on sensor type. (See table below)

V343 Speed filter by S2 sensor

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V343	Speed filter by S2 sensor	Hz	50 ÷ 3000	600	SM, AM, LM	NO / 2

The V343 parameter allows to set up the speed filter by S2 sensor and has to be enabled by means V349 bit 17 = 0. Recommended values of V343 depenting on sensor type. (See table below)

S1 / S2 Sensor	Recommended V342 / V343 value [Hz]
A, B, P, Q encoder, Optical linear scale	600
J, K encoder	250
TTL and Toothed wheel	150
SSI	50



V344 S1 Sensor code ID

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V344	S1 Sensor code ID	[ ]	word	0	SM, AM, LM	YES / 2

The V344 parameter allows to identified the compatibilty between the S1 sensor configuration (set by means V346) and the sensor if it is defined as Hiperface NUM type (See V348 bits 3,4,5).
The identification word ID is automatically loaded during the sensor configuration by FlexiumTool and WPM as indicated in the chapter 5.1 “Sensor configuration”.

If an incompatibility error occors the alarm A9.253 will appear..

V345 S2 Sensor code ID

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V345	S2 Sensor code ID	[ ]	word	0	SM, AM, LM	YES / 2

The V345 parameter allows to identified the compatibilty between the S1 sensor configuration (setted by means V347) and the sensor if it is defined as Hiperface NUM type (See V349 bits 3,4,5).
The identification word ID is automatically loaded during the sensor configuration by FlexiumTool and WPM as indicated in the chapter 5.1 “Sensor configuration”.

If an incompatibility error occors the alarm A38.253 will appear.

V346 S1 Sensor number

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V346	S1 Sensor number	[ ]	0 ÷ 9999	ST sensor	AM, SM, LM	YES / 2

Some S1 sensors have been codified associating them to a reference number as indicated in the chapter 7.

If the sensor is not configurated enter 0 in the V346 parameter.
If a not codified sensor is used enter 9999 in the V346 parameter or require the codification.

If the sensor is not shown in the table on chapter 5.1, before connecting it, always take into consideration the sensor data sheet and select the appropriate values and adjusting appropriate parameters as indicated on chapter 5.7.5 " Enter new sensor configuration with WPM".

V347 S2 Sensor number

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V347	S2 Sensor number	[ ]	0 ÷ 9999	ST sensor	AM, SM, LM	YES / 2

Some S2 sensors have been codified associating them to a reference number as indicated in the chapter 7.

If the sensor is not configurated enter 0 in the V346 parameter.
If a not codified sensor is used enter 9999 in the V346 parameter or require the codification.

If the sensor is not shown in the table on chapter 5.1, before connecting it, always take into consideration the sensor data sheet and select the appropriate values and adjusting appropriate parameters as indicated on chapter 5.7.5 " Enter new sensor configuration with WPM".

V348 S1 Sensor configuration

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V348	S1 Sensor configuration	BIT	0 ÷ 1FFFFF	0	AM, SM, LM	YES / 2

N° Bit	Bit function	Bit value = 0	Bit value = 1	Default value
0, 1, 2 3, 4, 5	See table on next page			0
6	CNC measure	Incremental	Absolute	0
7	Sensor cyclical direction	CCW	CW	0
8	Sensor type	Rotary sensor	Linear sensor	0
9	Reversal measure for position loop	Without reversal	With reversal	0
10	Disabling alarm A09.248	Alarm actived	Alarm not actived	0
11	Oversampling of sinusoidal signals (1)	Not enabled	Enabled	0
13	Sensor error signal test (pin 21 connector X1 or S1) Alarm A09.257.	Not enabled	Enabled	0
14	Compensation of sinusoidals signals	Not actived	Actived	0
15	Absolute measure translation for rotary sensor	Actived	Not actived	0
16	Homing switch use excluded	Not excluded	Excluded	0
17	Speed filter	With V342	With V340	0
18	SSI-EnDat Absolute position half-travel displacement	Not enabled	Enabled	0
19	EnDat memory handling excluded	Not excluded	Excluded	0
20	EnDat 2.2. sensor CRC error. Alarm A09.259	Alarm	Warning	0

(1) Function available on Mono-Axis drive only.

On the NUMDrive C Basic Performances version the bit 17 “speed filter” is available with V342 only (V348 = 17.0).
For a correct use of the configuration bits see the next pages.

V348 Use of configuration bits from bit 0 to bit 5

Bit 2	Bit 1	Bit 0	Table n. 1 - Incremental trace
0	0	0	Incremental trace absent
0	0	1	Sinusoidal trace
0	1	0	TTL
1	0	1	Fast serial*
1	1	1	Sensorless
* To receive the absolute position via serial link with sensor EnDat 2.2 or SSI only.			

Bit 5	Bit 4	Bit 3	Table n. 2 - Absolute trace
0	0	0	Absolute trace absent
0	0	1	Hiperface NUM type
0	1	0	EnDat type
0	1	1	Coded reference
1	0	0	Hiperface SER type
1	0	1	Standard Hiperface type
1	1	0	SSI type
1	1	1	EnDat 2.2 type

Warning for TTL sensors only.
If TTL sensors with coded reference are equipped in the system, the use of interpolator signals EXE type like it is not allowed.

Warning for SSI sensors only.
The SSI sensors type are allowed in compliance with the following specification:

- position sent in binary format with GRAY code
- MSB (Most Significant Bit) as first bit
- frame without CRC
- no incremental traces
- 333 kHz clock frequency
- RS485 electrical specification for DATA and CLOCK signals
- (V361*V365) less or equal to 2^31.

The sensor must be declared as rotary sensor only.
For linear sensor application contact NUM service.

V348 Use of configuration bits from bit 6 to bit 20

N° Bit	Bit function Sensor type S1	Bit utilization
6	CNC measure	By selecting the bit to 1 the CNC acquires the sensor measure as absolute and HOMING procedure is not required. Selecting the bit to 0 the CNC treats the sensor measure as incremental and therefore the HOMING procedure is required.
7	Sensor cyclical direction	This bit must be set = 0 or = 1 in accordance to comply the following sentence: turning the motor shaft clockwise (seen from out shaft) the axis position increases. (CCW = Counter Clockwise / CW = Clockwise)
8	Sensor type	Through the bit 8 it is possible to select the type of sensor used. Bit 8 = 0 selects a rotary sensor. Bit 8 = 1 selects a linear sensor (optical scale)
9	Reversal measure for position loop	Through the bit 9 it is possible to reverse the measure of the position loop sensor.
10	Disabling alarm A09.248	If the sensor is a sinusoidal type, through the bit 10 it is possible to activate the saturation control on the analog/ digital conversion of the sine/cosine signals. The test will be made with: Bit = 0 test enabled - Bit = 1 test disabled. ATTENTION: with active test and signals sine/cosine in saturation conditions the alarm A.09.248 will appear. It is always suggested to select the active test.
13	Sensor error signal test (pin 21 sensor connector X_ or S1) Alarm A09.257	This feature gives the possibility to check the presence of the sensor connector inserted making a jumper between pin 21 and pin 5. In case of TTL sensor could be connected to the UaS /signal. This feature could be enabled by: V348 (V349) bit 13 = 1. If this bit is enabled and pin 21 isn't at 0V, alarm A9.257 will appear.
14	Sinusoidal signal compensation	The sinusoidal signal compensation sets a compensation on the sine and cosine signals. These compensations are due to a offset or different amplitude presence.
15	Absolute measure translation for rotary sensor	Using an absolute sensor measure with translation direct measure active, the 0 value of measure (without use of V310/ V1310 parameters) will be established at the absolute sensor mid-point. I.e: Absolute sensor with 4096 sensor turns will keep 0 measure with active measure translation as 2048turns.
16	Homing switch use excluded	By selecting the bit = 0 the Homing switch use during the axis homing sequence it will be required. By selecting the bit = 1 the Homing switch use during the axis homing sequence it will be not required.
17	Speed filter	By selecting the bit to 0 or to 1 is possible to select the speed filter on V342 or V340. On the NUMDrive C Basic Performances version the bit 17 "speed filter" is available with V342 only (V348 bit 17 = 0).

18	SSI-Endat Absolute position half-travel displacement	When you use absolute encoders (multi-turns or single-turn), the linear axis stroke or of the rotary axis with limited stroke mustn't pass through the 0 absolute position of the encoder. For example with our "P" encoders the turns counter ranges from 0 to 4095. This means that inside the axis stroke the turns counter must not pass through the transition 4095->0 or viceversa 0->4095. For this reason we sell our motors with the turns counter set on 2047 - 2048 value (middle of the range). Starting from this fact and assuming all the parameters creating position offset like V310 (V1310, P16 Nx are set to 0 (zero), we see the axis position value = 0 (zero) when the turns counter is = 2048 and the absolute position into the turn is = 0. This result is due to the status of V348 bit 15 = 0 by which we subtract an offset equal to the half of the max value of the encoder position so the axis position ranges from minus half max value to plus half max value (with multi/divi = 1/1). You can check the values using MP223 and MP215. This feature gives the possibility to manage an axis where the transition 0 -> max value and max value -> 0 of the encoder stays inner the axis stroke. This feature works on encoders with SSI or EnDat protocol only and it was done because we cannot write inside the encoder memory to change the absolute position. This feature is enabled by: V348 bit 18 = 1.
19	EnDat memory handling excluded.	This feature gives the possibility to define where to store the position offset for the rotor electrical phasing. In NUM motors this value is stored into the encoder memory at the end of the EPSM procedure. - bit 19 = 0: at the end of the EPSM procedure, we store the position offset for the rotor electrical phasing into the encoder memory. - bit 19 = 1: at the end of the EPSM procedure, we store the position offset for the rotor electrical phasing into V269 parameter.
20	EnDat 2.2. sensor CRC error. Alarm A09.259	This feature gives the possibility to define how to manage the CRC error detected during the EnDat 2.2 communication. - bit 20 = 0: in case of CRC error the alarm A09 ext 259 will be shown. - bit 20 = 1: in case of CRC error a warning bit into the drive warning register word (0x1A72) will be set: - 0x1A72 bit 9 = 1 in case of CRC error from S1 sensor.

V349 S2 Sensor configuration

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V349	S2 Sensor configuration	BIT	0 ÷ 1FFFFF	0	AM, SM, LM	YES / 2

N° Bit	Bit function	Bit value = 0	Bit value = 1	Default value
0, 1, 2 3, 4, 5	See table on next page			0
6	CNC measure	Incremental	Absolute	0
7	Sensor cyclical direction	CCW	CW	0
8	Sensor type	Rotary sensor	Linear sensor	0
9	Reversal measure for position loop	Without reversal	With reversal	0
10	Disabling alarm A38.248	Alarm actived	Alarm not actived	0
11	Oversampling of sinusoidal signals (1)	Not enabled	Enabled	0
13	Sensor error signal test (pin 21 connector X2 or S2) Alarm A38.257.	Not enabled	Enabled	0
14	Compensation of sinusoidals signals	Not actived	Actived	0
15	Absolute measure translation for rotary sensor	Actived	Not actived	0
16	Homing switch use excluded	Not excluded	Excluded	0
17	Speed filter	With V343	With V341	0
18	SSI-EnDat Absolute position half-travel displacement	Not enabled	Enabled	0
19	EnDat memory handling excluded	Not excluded	Excluded	0
20	EnDat 2.2. sensor CRC error. Alarm A38.259	Alarm	Warning	0

(1) Function available on Mono-Axis drive only.

On the NUMDrive C Basic Performances version the bit 17 “speed filter” is available with V343 only (V349 bit 17 =0).
For a correct use of the configuration bits see the next pages.

V349 Use of configuration bits from bit 0 to bit 5

Bit 2	Bit 1	Bit 0	Table n. 1 - Incremental trace
0	0	0	Incremental trace absent
0	0	1	Sinusoidal trace
0	1	0	TTL
1	0	1	Fast serial*
1	1	1	Sensorless
* To receive the absolute position via serial link with sensor EnDat 2.2 or SSI only.			

Bit 5	Bit 4	Bit 3	Table n. 2 - Absolute trace
0	0	0	Absolute trace absent
0	0	1	Hiperface NUM type
0	1	0	EnDat type
0	1	1	Coded reference
1	0	0	Hiperface SER type
1	0	1	Standard Hiperface type
1	1	0	SSI type
1	1	1	EnDat 2.2 type

Warning for TTL sensors only.
If TTL sensors with coded reference are equipped in the system, the use of interpolator signals EXE type like it is not allowed.

Warning for SSI sensors only.
The SSI sensors type are allowed in compliance with the following specification:

- position sent in binary format with GRAY code
- MSB (Most Significant Bit) as first bit
- frame without CRC
- no incremental traces
- 333 kHz clock frequency
- RS485 electrical specification for DATA and CLOCK signals
- (V371*V375) less or equal to 2^31.

The sensor must be declared as rotary sensor only.
For linear sensor application contact NUM service.

V349 Use of configuration bits from bit 6 to bit 20

N° Bit	Bit function Sensor type S2	Bit utilization
6	CNC measure	By selecting the bit to 1 the CNC acquires the sensor measure as absolute and HOMING procedure is not required. Selecting the bit to 0 the CNC treats the sensor measure as incremental and therefore the HOMING procedure is required.
7	Sensor cyclical direction	This bit be must be set = 0 or = 1 in accordance to comply the following sentence: turning the motor shaft clockwise (seen from out shaft) the axis position increases. (CCW = Counter Clockwise / CW = Clockwise)
8	Sensor type	Through the bit 8 it is possible to select the type of sensor used. Bit 8 = 0 selects a rotary sensor. Bit 8 = 1 selects a linear sensor (optical scale).
9	Reversal measure for position loop	Through the bit 9 it is possible to reverse the measure of the position loop sensor.
10	Disabling alarm A38.248	If the sensor is a sinusoidal type, through the bit 10 it is possible to activate the saturation control on the analog/ digital conversion of the sine/cosine signals. The test will be made with: Bit = 0 test enabled - Bit = 1 test disabled. ATTENTION: with active test and signals sine/cosine in saturation conditions the alarm A.38.248 will appear. It is always suggested to select the active test.
13	Sensor error signal test (pin 21 sensor connector X_ or S2) Alarm A38.257	This feature gives the possibility to check the presence of the sensor connector inserted making a jumper between pin 21 and pin 5. In case of TTL sensor could be connected to the UaS /signal. This feature could be enabled by: V349 bit 13 = 1. If this bit is enabled and pin 21 isn't at 0V, alarm A38.257 will appear.
14	Sinusoidal signal compensation	The sinusoidal signal compensation sets a compensation on the sine and cosine signals. These compensations are due to a offset or different amplitude presence.
15	Absolute measure translation for rotary sensor	Using an absolute sensor measure with translation direct measure active, the 0 value of measure (without use of V310/ V1310 parameters) will be established at the absolute sensor mid-point. I.e: Absolute sensor with 4096 sensor turns will keep 0 measure with active measure translation as 2048turns.
16	Homing switch use excluded	By selecting the bit = 0 the Homing switch use during the axis homing sequence it will be required. By selecting the bit = 1 the Homing switch use during the axis homing sequence it will be not required.
17	Speed filter	By selecting the bit to 0 or to 1 is possible to select the speed filter on V343 or V341. On the NUMDrive C Basic Performances version the bit 17 "speed filter" is available with V343 only (V349 bit 17 =0).

18	SSI-Endat Absolute position half-travel displacement	When you use absolute encoders (multi-turns or single-turn), the linear axis stroke or of the rotary axis with limited stroke mustn't pass through the 0 absolute position of the encoder. For example with our "P" encoders the turns counter ranges from 0 to 4095. This means that inside the axis stroke the turns counter must not pass through the transition 4095->0 or viceversa 0->4095. For this reason we sell our motors with the turns counter set on 2047 - 2048 value (middle of the range). Starting from this fact and assuming all the parameters creating position offset like V313 (V1313, P16 Nx are set to 0 (zero), we see the axis position value = 0 (zero) when the turns counter is = 2048 and the absolute position into the turn is = 0. This result is due to the status of V349 bit 15 = 0 by which we subtract an offset equal to the half of the max value of the encoder position so the axis position ranges from minus half max value to plus half max value (with multi/divi = 1/1). You can check the values using MP225 and MP215. This feature gives the possibility to manage an axis where the transition 0 -> max value and max value -> 0 of the encoder stays inner the axis stroke. This feature works on encoders with SSI or EnDat protocol only and it was done because we cannot write inside the encoder memory to change the absolute position. This feature is enabled by: V349 bit 18 = 1.
19	EnDat memory handling excluded.	This feature gives the possibility to define where to store the position offset for the rotor electrical phasing. In NUM motors this value is stored into the encoder memory at the end of the EPSM procedure. - bit 19 = 0: at the end of the EPSM procedure, we store the position offset for the rotor electrical phasing into the encoder memory. - bit 19 = 1: at the end of the EPSM procedure, we store the position offset for the rotor electrical phasing into V269 parameter.
20	EnDat 2.2. sensor CRC error. Alarm A38.259	This feature gives the possibility to define how to manage the CRC error detected during the EnDat 2.2 communication. - bit 20 = 0: in case of CRC error the alarm A38 ext 259 will be shown. - bit 20 = 1: in case of CRC error a warning bit into the drive warning register word (0x1A72) will be set: - 0x1A72 bit 10 = 1 in case of CRC error from S2 sensor.

V350 Position loop sensor configuration

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V350	Position loop sensor configuration	Bit	0 ÷ 1	0	SM, AM, LM	YES / 2

The V350 parameter allows by means of a bit to select the position loop sensor configuration on S1 or S2 .

N° Bit	Bit function	Bit value = 0	Bit value = 1	Default value
0	Position loop sensor configuration	on S1	on S2	0

V351 Speed loop sensor configuration

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V351	Speed loop sensor configuration	Bit	0 ÷ 1	0	SM, AM, LM	YES / 2

TheV351 parameter allows by means of a bit, to select the speed loop sensor configuration on S1 or S2 .

N° Bit	Bit function	Bit value = 0	Bit value = 1	Default value
0	Speed loop sensor configuration	on S1	on S2	0

V352 Current loop sensor configuration

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V352	Current loop sensor configuration	Bit	0 ÷ 1	0	SM, AM, LM	YES / 2

The V352 parameter allows by means of a bit, to select the current loop sensor configuration on S1 or S2

N° Bit	Bit function	Bit value = 0	Bit value = 1	Default value
0	Current loop sensor configuration	on S1	on S2	0

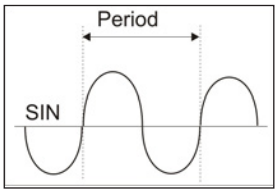
V353 Period of position sensor/motor turn [0]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V353	Period of position sensor/motor turn	[ ]	1 ÷ (1)	1024	SM, AM	YES / 2

$(1) = \frac{2^{11} - 1}{100}$

Enter in V353 parameter the number of direct measurement sensor periods per motor revolution. The V1353 parameter is enabled if the parameter set [1] has been selected by means the V261 =1. (Position loop regulator parameters exchange)

The adjustment of V353 parameter is essential when two different sensors are used to close the speed and position loops. The V353 parameter is not used when the same sensor closes the speed and position loops.

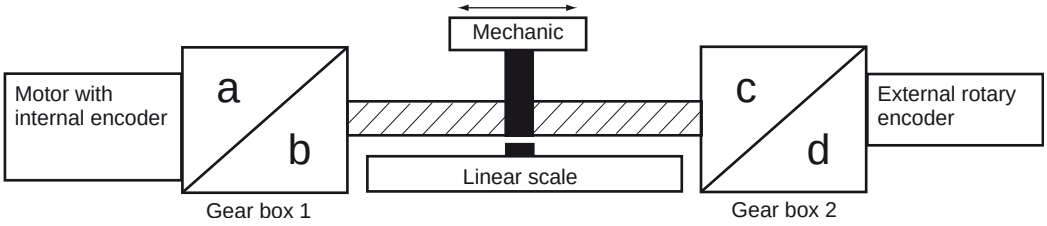


Generally you can calculate with the following formulas.
If no EXE (additional interpolation electronic) is used, the corresponding value in the formula becomes “1”.

By use of rotary axis you can calculate with a screw pitch of 360. The calculated values for V353 must be rounded to two digits after the comma. The following drawing should clarify both cases (direct linear scale or direct rotary encoder).

In case of the external rotary encoder, the possibility of an additional gear ratio is considered. If there don not exist an additional gear ratio, the corresponding values in the formula (gear ratio parameter c and d) become “1”.

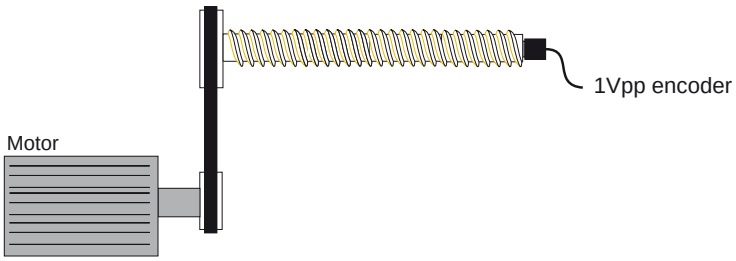
- Direct linear scale
  $V353 = i_{mechanic}(b) / i_{motor}(a) * \text{screw pitch [mm]} / \text{grating optical scale [µm]} * EXE * 1000$
- Direct ratio encoder
  $V353 = (i_{mechanic}(b) * i_{encoder}(d) * \text{encoder periods} * EXE) / (i_{motor}(a) * i_{mechanic}(c))$



V353 Application example with 1Vpp sinusoidal encoder

Enter the number of sinusoids generated by the encoder in one motor revolution.

- 1Vpp encoder, 1024 sinusoids/rev.
- Reduction ratio between motor and encoder = 2
  (for each motor revolution the encoder will generate 512 sinusoids)
- V353 = 512.00 (it is possible insert two decimal number after the comma)



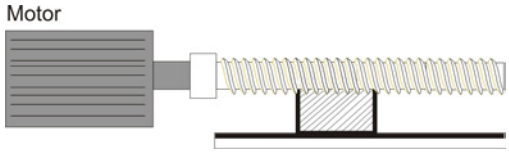
- The number of periods per motor revolution is not the number of counting pulses used to set-up the MULTI/DIVI measurement parameter.
- A wrong parameter setting causes wrong gains of the position loop.
- It is possible insert on V353 parameter two decimal number after the comma.

V353 Application example with 1Vpp sinusoidal optical scale

Enter the number of sinusoids generated by theoptical scale in one motor revolution.

- 1Vpp sinusoidal scale (one sinusoid every 16m)
- Screw pitch = 10mm (10.000m) with motor directly connected on screw

$$V353 = \frac{10000}{16} = 625$$ (Number of scale sinusoids for each motor revolution)



- The number of periods per motor revolution is not the number of counting pulses used to set-up the MULTI/DIVI measurement parameter.
- A wrong parameter setting causes wrong gains of the position loop.

V355 AP02: Mechanical ratio load turns

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V355	AP02: Mechanical ratio load turns	[ ]	0 ÷ 32767	0	SM, AM	YES / 2

The AP02 application manages the absolute position management (without to perform the HOMING procedure at each machine power on) of rotary axes with a mechanical ratio between the position sensor (S1 or S2) and axis (load) which is not 2^n.

The V355 parameter refers to the AP02 special application and must be used together with V356 parameter.
The AP02 special application is activated by setting bit 1 =1 on V160 parameter and bit 1 = 1 on V35 parameter. Enter into V355 the load turns value of the application.

V356 AP02: Mechanical ratio sensor turns

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V356	AP02: Mechanical ratio sensor turns	[ ]	1 ÷ 32767	1	SM, AM	YES / 2

The V356 parameter refers to the AP02 special application and must be used together with V355 parameter. The AP02 special application is activated by setting bit 1 =1 on V160 parameter.

The AP02 application manages the absolute position management (without to perform the HOMING procedure at each machine power on) of rotary axes with a mechanical ratio between the position sensor (S1 or S2) and axis (load) which is not 2^n.

Enter into V356 the sensor turns value of the application.

V360 Period by turn sensor S1

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V360	Period by turn sensor S1	[ ]	1 ÷ 131072	512	SM, AM	YES / 2

The parameter V360 sets the numbers of periods for S1 sensor revolution.

This parameter has to be adjusted in present of new sensor only, otherwise see note here below.

If the sensor is not shown in the table on chapter 5.1, before connecting it, always take into consideration the sensor data sheet and select the appropriate values and adjusting appropriate parameters as indicated on chapter 5.7.5 “ Enter new sensor configuration with WPM”.

V361 Absolute resolution by turn sensor S1

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V361	Absolute resolution by turn sensor S1	[ ]	1 ÷ 2147483647	16384	SM, AM	YES / 2

The V361 parameter sets the absolute resolution for S1 sensor revolution.
This parameter has to be adjusted in present of new sensor only, otherwise see note here below.
Using an incremental sensor type without digital communication observe this rule:

V361 = V360 * 4.

If the sensor is not shown in the table on chapter 5.1, before connecting it, always take into consideration the sensor data sheet and select the appropriate values and adjusting appropriate parameters as indicated on chapter 5.7.5 “ Enter new sensor configuration with WPM”.

V365 Number of absolute sensor turns S1

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V365	Number of absolute sensor turns S1	[ ]	1 ÷ 32767	1	SM, AM	YES / 2

The V365 parameter sets the number of absolute resolutions for S1 sensor.
If the sensor has not the absolute digital communication insert 1.

This parameter has to be adjusted in present of new sensor only, otherwise see note here below.

If the sensor is not shown in the table on chapter 5.1, before connecting it, always take into consideration the sensor data sheet and select the appropriate values and adjusting appropriate parameters as indicated on chapter 5.7.5 " Enter new sensor configuration with WPM".

V368 Intervals between coded references S1

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V368	Intervals between coded references S1	[ ]	1 ÷ 32767	1000	SM, AM, LM	YES / 2

The V368 parameter is used when we want to obtain the absolute position from an optical scale with coded references.

Check the number of intervals generated by the optical scale between two reference marks.
(Usually declared by the optical scale manufacturer).

In V368 parameter it is necessary to enter the number of periods generated by the optical scale between two reference marks. The default value is 1000.
Enter the number of periods in V368 parameter.

Application example with optical scale using coded references.

- Optical scale with coded references
- Codification every 20mm
- Period 20m.

$$V368 = \frac{20*1000}{20} = 1000$$

V370 Number of absolute sensor turns S2

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V370	Period by turn sensor S2	[ ]	1 ÷ 131072	512	SM, AM	YES / 2

The 370 parameter sets the number of periods for S2 sensor revolution.

This parameter has to be adjusted in present of new sensor only, otherwise see note here below.

If the sensor is not shown in the table on chapter 5.1, before connecting it, always take into consideration the sensor data sheet and select the appropriate values and adjusting appropriate parameters as indicated on chapter 5.7.5 “ Enter new sensor configuration with WPM”.

V371 Absolute resolution by turn sensor S2

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V371	Absolute resolution by turn sensor S2	[ ]	1 ÷ 2147483647	16384	SM, AM	YES / 2

The V371 parameter sets the absolute resolution for S2 sensor revolution.
This parameter has to be adjusted in present of new sensor only, otherwise see note here below.

Using an incremental sensor type without digital communication observe this rule:

V371 = V370 * 4.

If the sensor is not shown in the table on chapter 5.1, before connecting it, always take into consideration the sensor data sheet and select the appropriate values and adjusting appropriate parameters as indicated on chapter 5.7.5 “ Enter new sensor configuration with WPM”.

V375 Number of absolute sensor turns S2

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V375	Number of absolute sensor turns S2	[ ]	1 ÷ 32767	1	SM, AM	YES / 2

The V375 parameter sets the number of absolute resolutions for S1 sensor.
If the sensor has not the absolute digital communication insert 1.

This parameter has to be adjusted in present of new sensor only, otherwise see note here below.

If the sensor is not shown in the table on chapter 5.1, before connecting it, always take into consideration the sensor data sheet and select the appropriate values and adjusting appropriate parameters as indicated on chapter 5.7.5 “ Enter new sensor configuration with WPM”.

V378 Intervals between coded references S2

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V378	Intervals between coded references S2	[ ]	1 ÷ 32767	1000	SM, AM, LM	YES / 2

To use the V378 parameter see the description provided for V368 parameter.

V380 Period of linear sensor S1

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V380	Period of linear sensor S1	µm	1 ÷ 8000	16	SM, AM, LM	YES / 2

The V380 parameter sets the dimension of the linear S1 sensor period.
The parameter value is expressed in micron (µm).

This parameter has to be adjusted in present of new sensor only, otherwise see note here below.

If the sensor is not shown in the table on chapter 5.1, before connecting it, always take into consideration the sensor data sheet and select the appropriate values and adjusting appropriate parameters as indicated on chapter 5.7.5 “ Enter new sensor configuration with WPM”.

V381 Absolute resolution of linear sensor S1

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V381	Absolute resolution of linear sensor S1	nm	1 ÷ 32767	100	SM, AM, LM	YES / 2

The V381 parameter sets the absolute resolution of linear S1 sensor.

The parameter value is expressed in nm (nanometer).This parameter has to be adjusted in present of new sensor only, otherwise see note here below.

If the sensor is not shown in the table on chapter 5.1, before connecting it, always take into consideration the sensor data sheet and select the appropriate values and adjusting appropriate parameters as indicated on chapter 5.7.5 “ Enter new sensor configuration with WPM”.

V390 Period of linear sensor S2

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V390	Period of linear sensor S2	µm	1 ÷ 8000	16	SM, AM, LM	YES / 2

The V390 parameter sets the dimension of the linear S2 sensor period.
The parameter value is expressed in micron (µm).

This parameter has to be adjusted in present of new sensor only, otherwise see note here below.

If the sensor is not shown in the table on chapter 5.1, before connecting it, always take into consideration the sensor data sheet and select the appropriate values and adjusting appropriate parameters as indicated on chapter 5.7.5 “ Enter new sensor configuration with WPM”.

V391 Absolute resolution of linear sensor S2

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V391	Absolute resolution of linear sensor S2	nm	1 ÷ 32767	100	SM, AM, LM	YES / 2

The V391 parameter sets the absolute resolution of linear S2 sensor.

The parameter value is expressed in nm (nanometer).This parameter has to be adjusted in present of new sensor only, otherwise see note here below.

If the sensor is not shown in the table on chapter 5.1, before connecting it, always take into consideration the sensor data sheet and select the appropriate values and adjusting appropriate parameters as indicated on chapter 5.7.5 “ Enter new sensor configuration with WPM”.

V500 Filter configuration

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V500	Filter configuration	[ ]	0 ÷ 1FF	0	SM, AM, LM	NO / 2

The V500 parameter is used to enable the filters on torque and speed reference.

The filters on torque reference work always in cascade connection(1), and they can be loaded independently between them.

(1) See also functional diagram on following pages to give more information upon the use of the filter.

N° Bit	Bit function	Bit value = 0	Bit value = 1	Default value
0	1° - Filter for torque current reference	Filter not active	Filter active	0
1	2° - Filter for torque current reference			0
2	3° - Filter for torque current reference			0
3	4° - Filter for torque current reference			0
4	1° - Filter for speed reference			0
5	Notch Filter for AP13			0
6	Not used			0
7	Filters UPDATE on line See note for bit 7 use on next page.			0
8	TEST MODE: test on Iq reference filters			0

The filters 3 and 4 are not available on NUMDrive C Basic Performances version.

V500 Use of bit 7 - Filters UPDATE on line

It is possible to modify the filters in two different ways :

- By setting the drive in HALT.
  The new filters will take effect when the drive is set back in RUN.
- By setting bit 7 it is possible to enable a new filter configuration without setting HALT in the drive
- By selecting bit 7 =1 (with the drive in RUN) it will be possible to introduce modifications to the filters after the last RUN.

or

- By means WPM the bit 7 is automatically managed if the filters are required.

Example

- On start up, if bit 0 of parameter V500 is operating, the drive will use the filter characterized by parameters V501÷V509.
- If you also want to enable filter 2 (V511÷ V519) by setting V500 bit 1 =1, filter 2 will be enabled simply by setting bit 7 to 1.
- The filter update in the new configuration is carried out during the transfer phase from 0 to 1 of bit 7 (V500). Bit 7 can then be reset to zero, keeping the new configuration operating.
- When the tests are finished, carry out the parameters STORE paying attention to keep bit 7 at 0.

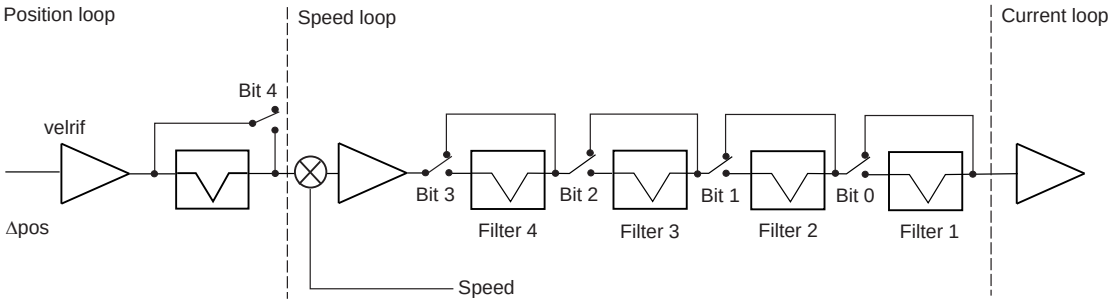
V500xx - V501 ÷ V549 Filter on torque and speed reference

The drive is provided with 4 filters on torque current and speed reference which may be adjusted by means of parameters listed before.

The filters are used in some particular cases, i.e. when the motor is directly connected on screw, and therefore, even with motor in torque from stationary, high noisiness situations (resonance) may occur.

Generally, if we analize, in this case, the wave form of the torque current, we will see a cyclic oscillation which approximately indicate the resonance frequency. As regards the frequency analysis, with WPM, it is possible to characterize all the mentioned filters.

Functional diagram



The filters can be used as lowpass filters or as notch filter.
The filters 3 and 4 are not available on NUMDrive C Basic Performances version.

V500xx - FILTER 1, 2 for torque current reference

N°	Parameters description	Unit	Adjustment range	Default	Motor utilization			HALT/ Level
V501	ph11 (FILTER 1 for torque current reference	[ ]	-32767÷32767	25956	SM	AM	LM	NO/2
V502	ph12 (FILTER 1 for torque current reference)	[ ]	0÷65535	0				
V503	ph21 (FILTER 1 for torque current reference)	[ ]	0÷65535	0				
V504	ph22 (FILTER 1 for torque current reference)	[ ]	-32767÷32767	32735				
V505	ga1 (FILTER 1 for torque current reference)	[ ]	-32767÷32767	15676				
V506	ga2 (FILTER 1 for torque current reference)	[ ]	-32767÷32767	0				
V507	r*r (FILTER 1 for torque current reference)	[ ]	0÷32767	12978				
V508	C1 (FILTER 1 for torque current reference)	[ ]	-32767÷32767	32767				
V509	C2 (FILTER 1 for torque current reference)	[ ]	-32767÷32767	0				
V511	ph11 (FILTER 2 for torque current reference)	[ ]	-32767÷32767	23442				
V512	ph12 (FILTER 2 for torque current reference)	[ ]	0÷65535	0				
V513	ph21 (FILTER 2 for torque current reference)	[ ]	0÷65535	0				
V514	ph22 (FILTER 2 for torque current reference)	[ ]	-32767÷32767	32735				
V515	ga1 (FILTER 2 for torque current reference)	[ ]	-32767÷32767	15057				
V516	ga2 (FILTER 2 for torque current reference)	[ ]	-32767÷32767	0				
V517	r*r (FILTER 2 for torque current reference)	[ ]	0÷32767	11721				
V518	C1 (FILTER 2 for torque current reference)	[ ]	-32767÷32767	32767				
V519	C2 (FILTER 2 for torque current reference)	[ ]	-32767÷32767	0				

V500xx - FILTER 3, 4 for torque current reference

N°	Parameters description	Unit	Adjustment range	Default	Motor utilization			HALT/ Level
V521	ph11 (FILTER 3 for torque current reference)	[ ]	-32767÷32767	15287	SM	AM	LM	NO/2
V522	ph12 (FILTER 3 for torque current reference)	[ ]	0÷65535	0				
V523	ph21 (FILTER 3 for torque current reference)	[ ]	0÷65535	0				
V524	ph22 (FILTER 3 for torque current reference)	[ ]	-32767÷32767	32735				
V525	ga1 (FILTER 3 for torque current reference)	[ ]	-32767÷32767	11721				
V526	ga2 (FILTER 3 for torque current reference)	[ ]	-32767÷32767	0				
V527	r*r (FILTER 3 for torque current reference)	[ ]	-32767÷32767	7643				
V528	C1 (FILTER 3 for torque current reference)	[ ]	-32767÷32767	32767				
V529	C2 (FILTER 3 for torque current reference)	[ ]	-32767÷32767	0				
V531	ph11 (FILTER 4 for torque current reference)	[ ]	-32767÷32767	8834				
V532	ph12 (FILTER 4 for torque current reference)	[ ]	0÷65535	0				
V533	ph21 (FILTER 4 for torque current reference)	[ ]	0÷65535	0				
V534	ph22 (FILTER 4 for torque current reference)	[ ]	-32767÷32767	32735				
V535	ga1 (FILTER 4 for torque current reference)	[ ]	-32767÷32767	7643				
V536	ga2 (FILTER 4 for torque current reference)	[ ]	-32767÷32767	0				
V537	r*r (FILTER 4 for torque current reference)	[ ]	0÷32767	4417				
V538	C1 (FILTER 4 for torque current reference)	[ ]	-32767÷32767	32767				
V539	C2 (FILTER 4 for torque current reference)	[ ]	-32767÷32767	0				

V500xx - FILTER 1 for speed reference and FILTER notch for AP13

N°	Parameters description	Unit	Adjustment range	Default	Motor utilization			HALT/ Level
V541	ph11 (FILTER 1 for speed reference)	[ ]	-32767÷32767	3870	SM	AM	LM	NO/2
V542	ph12 (FILTER 1 for speed reference)	[ ]	0÷65535	0				
V543	ph21 (FILTER 1 for speed reference)	[ ]	0÷65535	0				
V544	ph22 (FILTER 1 for speed reference)	[ ]	-32767÷32767	32735				
V545	ga1 (FILTER 1 for speed reference)	[ ]	-32767÷32767	3641				
V546	ga2 (FILTER 1 for speed reference)	[ ]	-32767÷32767	0				
V547	r*r (FILTER 1 for speed reference)	[ ]	0÷32767	1935				
V548	C1 (FILTER 1 for speed reference)	[ ]	-32767÷32767	32767				
V549	C2 (FILTER 1 for speed reference)	[ ]	-32767÷32767	0				
V551	ph11 (FILTER notch for AP13)	[ ]	-32767÷32767	269				
V552	ph12 (FILTER notch for AP13)	[ ]	0÷65535	6956				
V553	ph21 (FILTER notch for AP13)	[ ]	0÷65535	2380				
V554	ph22 (FILTER notch for AP13)	[ ]	-32767÷32767	269				
V555	ga1 (FILTER notch for AP13)	[ ]	-32767÷32767	-277				
V556	ga2 (FILTER notch for AP13)	[ ]	-32767÷32767	27				
V557	r*r (FILTER notch for AP13)	[ ]	0÷32767	32628				
V558	C1 (FILTER notch for AP13)	[ ]	-32767÷32767	32767				
V559	C2 (FILTER notch for AP13)	[ ]	-32767÷32767	0				

V1050 Torque current limit [1]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1050	Torque current limit [1]	Arms	0 ÷ ST value	ST value	SM, AM, LM	NO

Through the V1050 parameter it is possible to reduce the maximum drive current in order to limit the maximum torque delivered by the motor.
The value is expressed in Arms.

Enter the Arms value of the desired torque current into parameter V1050.

- Example of V1050 adjustment for MDLU3021A... (15 Arms. max)

- By entering the value 10.00 in parameter V50 the effective current delivered by the system will be 10 Arms max. (with subsequent reduction of motor maximum torque).

By selecting the motor/drive association, the maximum value foreseen for that motor will automatically be setup on V50 parameter.

ATTENTION! Never set-up current values higher than the standard value resulting from motor/drive association.

V1051 Proportional speed loop gain [1]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1051	Proportional speed loop gain [1]	mA/Rpm	0 ÷ ST value	15	SM, AM, LM	NO / 2

The V1051 parameter is used to improve the dynamic response of the system.

The value of the speed cycle's proportional gain will have to be increased by adjusting parameter V1051 in general by the same percentage as the load inertia in relation to the motor.

For a correct use of the parameter see chapter 5.7.1 Speed loop parameters optimisation procedure.

V1052 Integral speed loop gain [1]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1052	Integral speed loop gain [1]	ms	0 ÷ ST value	ST value	SM, AM, LM	NO / 2

The V1052 parameter is used to improve the dynamic response of the system.

For a correct use of the parameter see chapter 5.7.1 Speed loop parameters optimisation procedure.

V1053 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1053						

V1057 Torque current limit - High range [1]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1057	Torque current limit high range [1]	Arms	0 ÷ ST value	4.30	AM	NO / 2

ATTENTION! To use the V1057 parameter please refer to the description provided for parameter the V1050 parameter. (only for asynchronous motors with range change)

The high range parameters can be used only for asynchronous motors with the electrical range change.

The corresponding parameter for low range is the standard parameter used for axis motors.
(Ex. V1057 high range current limit adjustment - V1050 lower current limit adjustment).

V1058 Proportional speed loop gain - High range [1]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1058	Proportional speed loop gain h. r. [1]	mA/Rpm	0 ÷ ST value	100	AM	NO / 2

ATTENTION! To use the V1058 parameter see the description provided for the V1051 parameter. (only for asynchronous motor with range change).

Parameters related to high range are used only for asynchronous motors with electrical range change.
The corresponding parameter for low range is the standard parameter used for axis motors.

V1059 Integral speed loop gain - High range [1]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1059	Integral speed loop gain h. r. [1]	msec	0 ÷ ST value	30	AM	NO

ATTENTION! To use the V1059 parameter see the description provided for the V1052 parameter. (only for asynchronous motor with range change).

Parameters related to high range are used only for asynchronous motors with electrical range change.
The corresponding parameter for low range is the standard parameter used for axis motors.

V1060 Proportional speed loop gain [1]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1060	Proportional speed loop gain [1]	Arms/m/min	0 ÷ ST value	15	LM	NO / 2

ATTENTION! To use the V1060 parameter see the description provided for the V1051 parameter. Set the parameter to a value allowing to obtain the desired dynamic performance.

To optimize this parameter it is necessary to carry out a preventive calibration of filters on currents reference.

V1075 Do not use

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1075						

V1094 Speed reference Limit [% maximum speed] [1]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1094	Speed reference limit [1]	%	0 ÷ 200	110	LM	NO / 2

To use the V1094 parameter see the description provided for V1097 parameter.

V1095 Maximum motor speed [1]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1095	Maximum motor speed [1]	m/min	0 ÷ 1000	10	LM	NO / 2

To use the V1095 parameter see the description provided for V1098 parameter.
The speed is expressed in m/min.

V1096 Threshold for overspeed alarm [1]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1096	Threshold for overspeed alarm [1]	m/min	0 ÷ 1000	12	LM	NO / 2

To use the V1096 parameter see the description provided for V1099 parameter.
The speed is expressed in m/min.

V1097 Speed reference limit [% max. speed] [1]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1097	Threshold for overspeed alarm [1]	%	0 ÷ 200	110	AM, SM	NO / 2

During the control speed, delta position and position modes, the V1097 parameter is the maximum value of the speed reference which can be reached.

The value of V1097 parameter is a percentage (%) of V1098 parameter.

Example

- V1098 = 3000 rpm
- V1097 = 110 %

In any case the speed reference can never be greater than 3300 rpm.
Speed reference values greater than 3300 Rpm will be automatically limited by the system.

V1098 Maximum motor speed [1]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1098	Maximum motor speed [1]	Rpm	0 ÷ V_{max} (1)	1500	SM, AM	NO / 2

(1) $V_{max} = \frac{140000 * 50}{(V257 * V271)}$

Command mode - Speed control

- Digital reference : V1098 parameter defines the rotation speed of the motor reference with the maximum digital reference ± (2^{24} -1) .
- Analog reference : V1098 parameter defines the rotation speed of the motor reference with a reference of ± 10V.

Command mode - Position and delta position control

- Digital reference : V1098 parameter is used as basic value for calculation of reference V1097 parameter.
- Analog reference : See the above statement related to digital reference.

Application example

Spindle motors: The parameter value is strictly connected to the value set for object 607F. (maximum spindle speed expressed in rpm)

Example.
- Spindle speed 2000 rpm
- Reduction ratio 1:2
- V1098 parameter value = 4000.

Axis motors: The parameter value is strictly connected to the value set for object 607F. (maximum axes speed expressed in mm/min)

Example.
- Axis speed 20000 mm/min
- Screw pitch 10 mm with motor directly connected on screw
- V1098 parameter value = 2000.

1098 Parameter set [1] of V98 parameter.

V1099 Overspeed alarm threshold [1]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1099	Overspeed alarm threshold [1]	Rpm	0 ÷ V_{max} (1)	2000	SM, AM	NO / 2

$$\frac{140000 * 50}{(V257 * V271)}$$

The motor overspeed control is one of the possible functions associated with the speed loop; in fact, the triggering of alarm A11 (motor overspeed) can be controlled through the adjustment of V1099 parameter.
By means of this parameter it is for example possible to stop the motor in the event of an anomaly (motor coast or incorrect programming of parameters associated with the speed loop).

Setup V1099 parameter with the desired overspeed value in rpm. If the speed value in V1099 parameter is exceeded the alarm A11 will be activated. Always check that V1099 parameter is compatible with the value setup in V1098 parameter.

Overspeed values

- Axis motor overspeed: normally, the overspeed threshold value is calibrated at the rated motor speed +20%.
- Spindle motors overspeed: normally, the overspeed threshold value is calibrated at the rated motor speed +10%.

In case of overspeed alarm activation, the motor will stop by inertia or through short-circuiting phases and the alarm A11 will be displayed. (See V32 use of the bit configuration)

Application example with motor BPH 095/3/V.

- Motor operating speed 4000 rpm + 20% = 4800 (value to be entered into V99)
- Select V1099 parameter, then enter 4800.

For speed value >4800 rpm the drive will go into alarm state A11.

V1301 Proportional position gain [1]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1301	Proportional position gain [1]	mm/min/mm	0 ÷ 50000	1000	SM, AM, LM	NO / 2

The V1301 parameter allows to adjust the position loop gain.
The parameter is expressed in mm/min/mm.

If a different sensors for speed loop closure is used, to obtain a correct gain it is necessary to properly set the V1353 parameter.

To optimize the adjustment of proportional position gain it is necessary to optimize first the speed loop gains.

V1306 S1 sensor MULTI coefficient [1]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1306	S1 sensor MULTI coefficient [1]	[ ]	- 32767 ÷ +32767	1	SM, AM, LM	YES / 2

The MULTI / DIVI parameters are measurement conversion coefficients.
The MULTI (V1306) / DIVI (V1307) parameters convert the feedback provided by motor position sensor to the internal measurement units needed by the application selected as “NC Units”.
(See also “NC Units” on chapters 5.2.1, 5.3.1)

The V1306 and V1307 parameters are needed when the S1 sensor is used to close the position loop as well as in the AP01 special application during the initialization of absolute position provided by S1 sensor.

Enter as V1306 (MULTI) parameter, an axis displacement in in “NC Units”.
Enter as V1307 (DIVI) parameter, the number of feedback counts needed to carry out the displacement indicated in V1306 parameter.
The measure in in “NC Units” (Measure point MP Point 215) is:

$$\frac{\text{S1 sensor measurement} * \text{V1306}}{\text{V1307}}$$

Parameters V1306 and V1307 replace V306 and V307 parameters when parameter set 1 is activated.
The maximum value for V1306 and V1307 parameter is 32767.

If the values to be used are higher, it is advisable to simplify the ratio by eliminating the factors common to numerator and denominator. By selecting the negative sign (-) on V1306 parameter (ex. -10.000) the measurement direction is inverted.

The inversion of measurement direction must be used if the measurement decreases when the axis moves in the positive direction.

V1307 S1 sensor DIVI coefficient [1]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1307	S1 sensor DIVI coeffcient [1]	[ ]	1 ÷ 32767	1	SM, AM, LM	YES / 2

To use the V1307 parameter see the description provided for the V1306 parameter.

V1310 S1 sensor offset in Controller physical unit [1]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1310	S1 sensor offset in CNC Phys. Unit [1]	[ ]	$-2^{31} \div +(2^{31}-1)$	0	SM, AM, LM	YES / 2

The V1310 parameter is active when the S1sensor is used to close the positioning loop as well as in the AP01 special application, during the initialization of absolute position performed by S1 motor sensor.

The value of the V1310 parameter is added to the position value in “NC Units”.
(See also “NC Units” on chapters 5.2.1, 5.3.1)

Application.

- S1 sensor managed as incremental for the Controller it is not used.
  During the HOMING execution (machine origin detection) the object 607Ch or 687Ch is assigned to the position value in “NC Units”.

- S1 sensor managed as absolute for the Controller (V348 bit6 =1) :
  The value of V1310 is added to the position value in Controller physical unit during the initialization of the absolute position automatically performed at drive power on and after every RUN command or by means the alarms reset.

AP01 special application.
During the initialization of absolute position by S1 sensor, the V1310 parameter is used as described in the second case.

V1311 S2 sensor MULTI coefficient [1]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1311	S2 sensor MULTI coefficient [1]	[ ]	- 32767 ÷ +32767	1	SM, AM, LM	YES / 2

The MULTI / DIVI parameters are conversion coefficients of the measurement.
The MULTI (V1311) / DIVI (V1312) parameters allow to adapt the measurement provided by motor position sensor to the internal “NC Unit”.
(See also “NC Units” on chapters 5.2.1, 5.3.1)

The V1311 and V1312 parameters are needed when the S2 sensor is used to close the position loop as well as in the AP01 special application during the initialization of absolute position provided by S2 sensor.

Enter as V1311 (MULTI) parameter the measure of shift performed by the system.
Enter as V1312 (DIVI) parameter the counts number of measurement system needed to carry out the shift indicated in V1311 parameter. The measure in “NC Unit” (measure point MP 215) is :

$$\frac{\text{S2 sensor measurement} * \text{V1311}}{\text{V1312}}$$

The maximum setting value of V1311 and V1312 parameters is 32767.
If the values to be setup are higher it will be necessary to rationalize such values by reducing them to the minimum terms.

Selecting the negative sign (-) on V311 parameter (i.e. -10.000) causes the inversion of measurement direction. If the values to be used are higher, it is advisable to simplify the ratio by eliminating the factors common to numerator and denominator.

The inversion of measurement direction must be used if the measurement decreases when the mobile element moves to the axis positive direction.

V1312 S2 sensor DIVI coefficient [1]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1312	S2 sensor DIVI coefficient [1]	[ ]	1 ÷ 32767	1	SM, AM, LM	YES / 2

To use the V1312 parameter see the description provided for the V1311 parameter.

V1313 S2 sensor offset in Controller physical unit [1]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1313	S2 sensor offset in Controller physical unit [1]	[ ]	$-2^{31} \div +(2^{31}-1)$	0	SM, AM, LM	YES / 2

If the machine zero does not correspond to the cam zero it is necessary to enter in V1313 parameter the measure read on the Controller when the axis is on the desired machine zero position.
The measurement sign must be the opposite of measurement read on the Controller in machine zero position.

The V1313 parameter is active when the S2 sensor is used to close the positioning loop as well as during the initialization of absolute position performed by S2 sensor.
The value of V1313 parameter is added to the position value in “NC Units”.
(See also “NC Units” on chapters 5.2.1, 5.3.1)

To use the V1313 parameter see the description provided for V313 parameter.

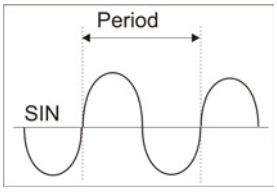
V1353 Period of position sensor/motor turn [1]

Parameter	Description	Unit	Adjustment range	Default	Utilization motor	HALT / Level
V1353	Period of position sensor/motor turn [1]	[ ]	1 ÷ (1)	1024	SM, AM	YES / 2

$(1) = \frac{2^{31}-1}{100}$

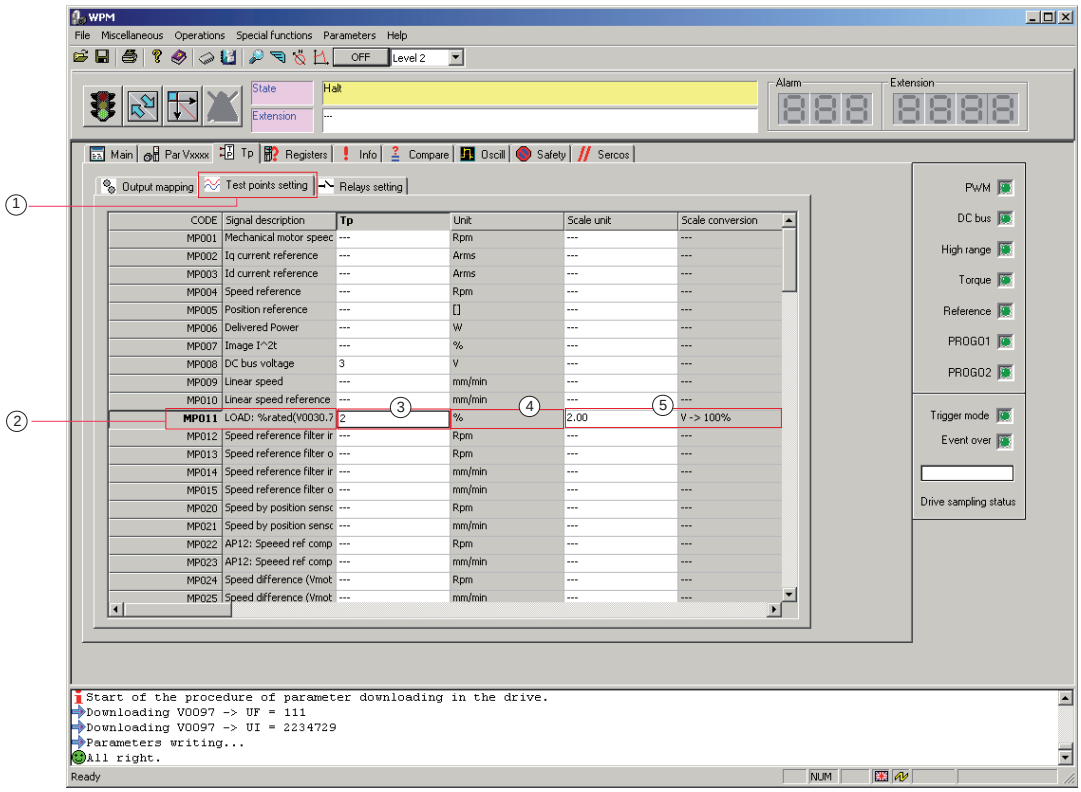
Enter in V1353 parameter the number of direct measurement sensor periods per motor revolution.
The V1353 parameter is enabled if the parameter set1 has been selected by means the V261 =1. (Position loop regulator parameters exchange)

The adjustment of V1353 parameter is essential when two different sensors are used to close the speed and position loops. The V1353 parameter is not used when the same sensor closes the speed and position loops.



5.6 NUMDrive C test points with WPM

5.6.1 Drive test point



1	Click on the icon to open the “Test point setting” page.
2	Measure point (MP) page (1). See MP list on next page
3	Measure point with test point (TP) association
4	Measure point unit
5	Measure point scale unit and scale conversion

(1) Display and modify all drive test points and associated measure points (MP) at LEVEL 1, 2 of modification only.

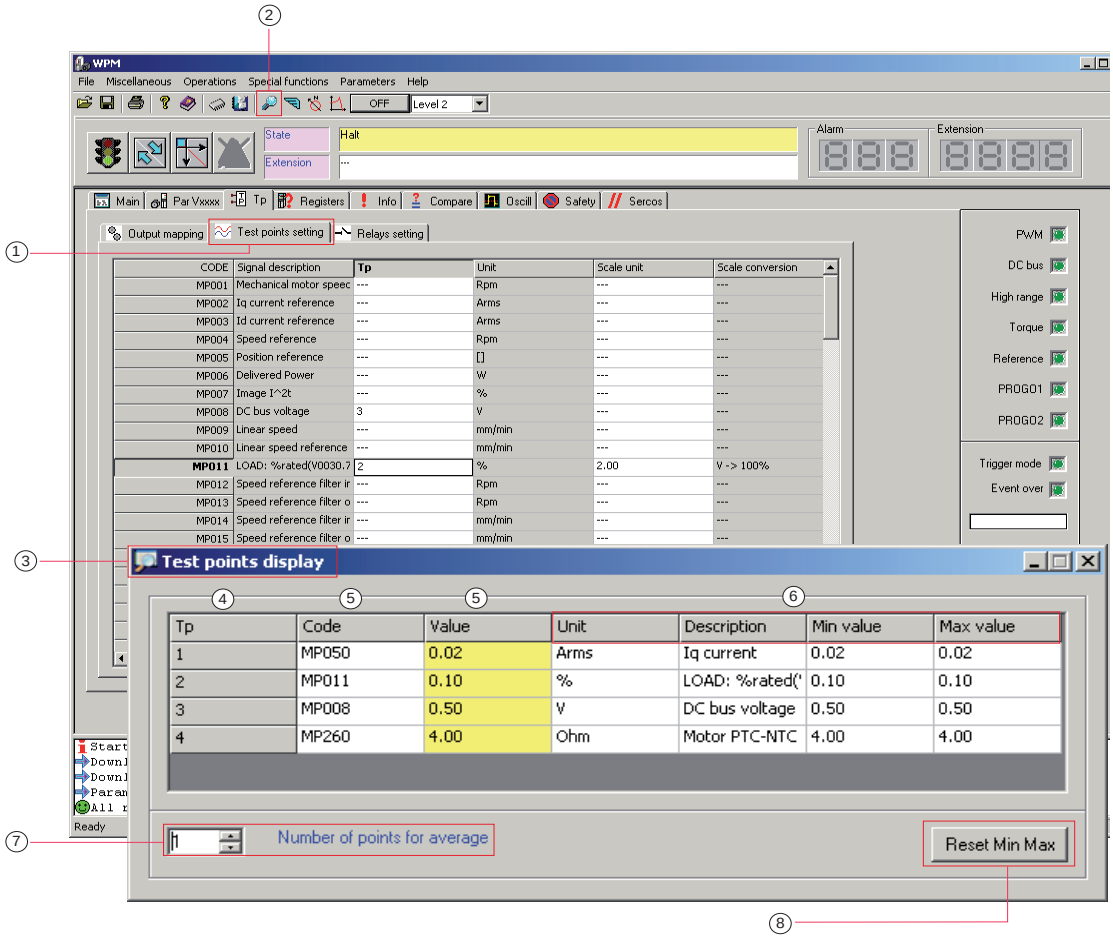
MDLU3 drive is provided of 4 test point: 2 analog/digital TP1/2 and 2 digital TP3/4:

- TP1 is enable on pin 3 of S9 connector.
- TP2 is enable on pin 1 of S9 connector.
- The drive test points are associated to the V130, V131, V132, V133 parameters. (see chapter 4 of the manual)

The measure points (MP) tables are indicated on next pages.

For more detailed information about the use of WPM tool, please refer to the WPM Instruction manual upon “Help” windows.

5.6.2 Test Points (TP) and Measure Points (MP) display



1	Click on the icon to open the “Test point setting” page.
2	Measure point (MP) page. See MP list on next page
3	Test point display page
4	Test point (TP) number
5	Associated measure point MP
6	MP value in real time
7	Number of point for average modification
8	Reset of the Min Max measure point value

For more detailed information about the use of WPM tool, please refer to the WPM Instruction manual upon “Help” windows.

5.6.3 Measure points general list (MP)

MP	Description	Internal Unit [IU]	Physical Unit [PU]	V143, V153 digital scale factor (DSF) [PU] = [ ] / DSF	Analog scale	V134, V135 analog scale factor (ASF)	Motor utilisat. / Level
001	Mechanical motor speed	32Bit	rpm	1342.18	1V - 1000rpm	163840	AM, SM / 1
002	Iq current reference	16Bit	Arms	(1)	8V - Imax	134219776	AM, SM, LM / 1
003	Id current reference	16Bit	Arms	(1)	8V - Imax	134219776	AM, SM, LM / 1
004	Speed reference	32Bit	rpm	1342.18	1V - 1000rpm	163840	AM, SM / 1
005	Position reference	32Bit	[ ]	1	10V - Max (5)	1024	AM, SM, LM / 1
006	Delivered Power	32Bit	W	(6)	2 V -10 KW	(8)	AM, SM, LM / 2
007	Image I^2t	32Bit	%	(2)	6V - 100%	(9)	AM, SM, LM / 1
008	DC bus voltage	16Bit	V	8	1V - 100V	274877907	AM, SM, LM / 1
009	Linear speed	32Bit	mm/min	163.84	1V - 10m/min	134217.728	LM / 1
010	Linear speed reference	32Bit	mm/min	163.84	1V - 10m/min	134217.728	LM / 1
011	LOAD (see also V30.7 on chapter 4)	16Bit	%	40.96	2V - 100% rated (V30.7=0) 2V - 100% max (V30.7=1)	107374182	AM, SM, LM / 2
012	Speed ref. filter input	32Bit	rpm	1342.18	1V- 1000rpm	163840	AM, SM / 1
013	Speed ref. filter output	32Bit	rpm	1342.18	1V- 1000rpm	163840	AM, SM / 1
014	Speed ref. filter input	32Bit	mm/min	163.84	1V - 10m/min	134217.728	LM / 1
015	Speed ref. filter output	32Bit	mm/min	163.84	1V - 10m/min	134217.728	LM / 1
020	Speed by position sensor	32Bit	rpm	1342.18	1V - 1000rpm	163840	AM, SM / 1
021	Speed by position sensor	32Bit	mm/min	163.84	1V - 10m/min	134217.728	LM / 1
022	AP12: Speeed ref comp. by act. damping 2	32Bit	rpm	1342.18	1V - 1000rpm	163840	AM, SM / 1
023	AP12: Speeed ref comp. by act. damping 2	32Bit	mm/min	163.84	1V - 10m/min	134217.728	LM / 1
024	Speed difference (Vmot - Vload)	32Bit	rpm	1342.18	1V - 100Rpm	1638400	AM, SM / 1
025	Speed difference (Vmot - Vchar)	32Bit	mm/min	163.84	1V - 1m/min	1342177.28	LM / 1
030	AP13: modulation factor of AP12	16Bit	%	327.67	10V = 100%	67108864	AM, SM, LM / 1
048	Interpolated Iq current reference limit	16Bit	Arms	(1)	8V - Imax	134219776	AM, SM, LM / 2
049	Iq current reference limit						AM, SM, LM / 2
050	Iq current						AM, SM, LM / 1
051	Id current						AM, SM, LM / 2
052	Alpha current	16Bit	Apeak	(3)	8V - Imax	134219776	AM, SM, LM / 2
053	Beta current						AM, SM, LM / 2
054	Phase U current						AM, SM, LM / 1
055	Phase V current						AM, SM, LM / 2
058	Iq^2+Id^2 current limit	16Bit	Arms	(1)	8V - Imax	134219776	AM, SM, LM / 2
061	Rotor alpha flux	16Bit	[ ]	1	10V - Max (4)	67108864	AM / 2
062	Rotor beta flux						

MP	Description	Internal Unit [IU]	Physical Unit [PU]	V143, V153 digital scale factor (DSF) [PU] = [ ] / DSF	Analog scale	V134, V135 analog scale factor (ASF)	Motor utilisat. / Level
063	Observer alpha flux	16Bit	[ ]	1	10V - Max (4)	67108864	AM / 2
064	Observer beta flux						
065	Magnetic Flux						
066	Flux reference						
081	Sin electrical position				10V - 90°		AM, SM, LM / 2
082	Cos electrical position				10V - 0°		
100	Vqreg	16Bit	V	8	1V - 100V	274877907	AM, SM, LM / 2
101	Vdreg						
102	Vq						
103	Vd						
104	Ref Va						
105	Ref Vb						
106	Inst Va						
107	Inst Vb						
108	Va observer						AM / 3
109	Vb observer						
110	delta_Va						
111	delta_Vb						
150	AP03: Mst: Torque equaliz. output	32Bit	rpm	1342.18	100V -1000rpm	16384000	AM, SM / 1
151	AP03: Mst: Torque equaliz. output	32Bit	mm/min	163.84	10V - 1 m/min	13421772.8	LM / 1
180	V/f: Total motor current	16Bit	Arms “	[1]	8 V - Imax	134219776	AM / 2
181	V/f: Current limiter outpu	32Bit	Hz	26843.5456	2 V - 500 Hz	32768	AM / 2
190	AP06: displacement measure S1-S2	32Bit	[PU]	1	10V - Max (5)	1024	AM, SM, LM / 1
191	AP06: displacement measure S1-S2 (LSW)	16Bit					
200	Position loop feedback	32Bit					
202	S1 sensor cosine	16Bit	%	122.88	6V - 100% (7)	107374182	AM, SM, LM / 2
203	S1 sensor sine						
204	S2 sensor cosine						
205	S2 sensor sine						
206	S1 sensor position	32Bit	[ ]	1	10V - Max (5)	1024	AM, SM, LM / 2
207	S2 sensor position						
208	Position in the motor rev.	16Bit	mDeg	0.1820444444	8V - 360 Degree	26843546	AM, SM, LM / 2
209	Electric motor position						
210	Electric motor speed	16Bit	Hz	13.1072	2V - 500Hz	67108864	AM, SM, LM / 2

MP	Description	Internal Unit [IU]	Physical Unit [PU]	V143, V153 digital scale factor (DSF) [PU] = [ ] / DSF	Analog scale	V134, V135 analog scale factor (ASF)	Motor utilisat. / Level
211	Reference in Physical Units	32Bit	[PU]	1	10V - Max (5)	1024	AM, SM, LM / 2
212	Reference in Internal Units	32Bit	[ ]				
213	Reference in Internal Units interp.	32Bit	[ ]				
215	Position [Measure] in Physical Units	32Bit	[PU]	1	10V - Max (5)	1024	AM, SM, LM / 1
216	S1 position [Measure] in Physical Units						
217	S2 position [Measure] in Physical Units						
218	(Sensor -> motor) phase offset	16Bit	mDeg	0.1820444444	8V - 360 Degree	26843546	AM, SM, LM / 2
220	Position error	32Bit	[ ]	1	10V - Max (5)	1024	AM, SM, LM / 2
221	Phased electric position	16Bit	mDeg	0.1820444444	8V - 360 Degree	26843546	AM, SM, LM / 2
222	Planned electrical position						SM, LM / 2
223	S1 encoder turn counter	32Bit	[ ]	1	1V -1000 turns	219902326	AM, SM / 2
224	AP02 encoder overflow counter	32Bit	[ ]		1V - 1000 overflow		AM, SM / 2
225	S2 encoder turn counter	32Bit	[ ]		1V -1000 turns		AM, SM / 2
226	Compensated S1 sensor cosine	16Bit	[ ]	1	10V - Max (4)	134217728	AM, SM, LM / 2
227	Compensated S1 sensor sine	16Bit	[ ]		10V - Max (4)	134217728	AM, SM, LM / 2
228	Offset of S1 sensor cosine	16Bit	%	122.88	6V - 100% (7)	107374182	AM, SM, LM / 2
229	Offset of S1 sensor sine						
230	Amplitude of S1 sensor cosine						
231	Amplitude of S1 sensor sine						
232	Compensated S2 sensor cosine	16Bit	[ ]	1	10V - Max (4)	134217728	AM, SM, LM / 2
233	Compensated S2 sensor sine			1	10V - Max (4)	134217728	
234	Offset of S2 sensor cosine	16Bit	%	122.88	6V - 100% (7)	107374182	AM, SM, LM / 2
235	Offset of S2 sensor sine						
236	Amplitude of S2 sensor cosine						
237	Amplitude of S2 sensor sine						
238	(sin^2+cos^2) sensor S1	16Bit	%	122.88	6V - 100% (7)	107374182	AM, SM, LM / 2
239	(sin^2+cos^2) sensor S2						
240	Top Mark S1 sensor		[ ]	1	10V = Top Mark present	67108864	
241	Top Mark S2 sensor						
242	Resolver carrier amplitude		Vrms	2457.6	1 V - 1 Vrms	89478485	
243	Resolver carrier phase		Deg	1.42	1 V -180°	858993459	
245	Sinusoidal Frequency encoder S1	32Bit	Hz	1	1V - 10kHz	21990233	AM, SM, LM / 2
246	Sinusoidal Frequency encoder S2						
MP	**Description**	**Internal Unit [IU]**	**Physical Unit [PU]**	**V143, V153 digital scale factor (DSF) [PU] = [ ] / DSF**	**Analog scale**	**V134, V135 analog scale factor (ASF)**	**Motor utilisat. / Level**

252	Iq reference at Filter input	16Bit	Arms	(1)	8V - I max	134219776	AM, SM, LM / 2
253	Proportional Iq comp. from speed reg.						
254	Integral Iq comp. from speed reg.	32Bit	Arms	(1)	8V - I max	134219776	AM, SM, LM / 2
255	AP08: Iq component by active damping						
256	AP10: Integral Iq comp. from speed reg.	16Bit	Arms				
260	Motor PTC-NTC	16Bit	Ω	1	1 V -1000 Ohm	219902325.6	AM, SM, LM / 1
261	(Sensor -> motor) phase offset [HOMING]		mDeg	0.1820444444	8V - 360 Degree	26843546	SM, LM / 2
262	Final pulse dimension for EPS		[ ]	1	10V - Max (4)	67108864	
264	Hall sensor electric position	16Bit	mDeg	0.1820444444	8V - 360 Degree	26843546	
265	Hall sensor electric phase offset (EPSM)						
312	Notch n° 1 input TP for Test	16Bit	Arms	(1)	8V - Imax	134219776	AM, SM, LM / 1
313	Notch n° 1 output TP for Test						
314	Debug Broadcast SBC	16Bit	[ ]	1	10V - Max (4)	67108864	AM, SM, LM / 3
315	Debug Task CNC						
316	Motor number						AM, SM, LM / 2
318	Hiperface S1 encoder ID code						
319	Hiperface S2 encoder ID code						
320	Glitch HOMING OK	16Bit	[ ]	1	10V = GLITCH ON	67108864	AM, SM, LM / 2
321	Glitch APEM OK (V273.2 = 1)						
322	Background time running	32Bit	uS	1	1 V -10 ms	219902326	AM, SM, LM / 3
324	Torque enable request	16Bit	[ ]	1	10V - ON	67108864	AM, SM, LM / 2
325	Torque enable (0x1AF2.1)				10V - ON	67108864	
326	Drive OK (0x1AF2.14)				10V - ON	67108864	

( ) = See “Use of measure points: notes ( )” on the next page.

5.6.4 Use of measure points: notes ( )

(1)	The scale factor is equal: $\frac{18536.097}{\text{Module size}}$
(2)	The scale factor is equal: $\frac{(\text{Nominal motor current V108, V115})^2}{(\text{Module size})^2} * 419.4$
(3)	The scale factor is equal: $\frac{13107}{\text{Module size}}$
(4)	Max 2^{15}
(5)	Max 2^{31}
(6)	The scale factor is equal: $\frac{856114.5655}{\text{Module size}}$
(7)	100% = Peak sensor signal
(8)	The scale factor is equal: 518.7 * Module size
(9)	The scale factor is equal:
Digital Value	Internal drive unit values
[PU]	Physical unit selection on CNC
[IU], [ ]	Internal drive unit values
ASP	Analog scale factor
DSP	Digital scale factor
Physical Unit	Physical unit
Unit	See also the use of V141, V142, V151, V152 parameters
Module size	007, 014, 021, 034, 050, 075, 100, 130, 150, 200, 400
I_{max} (Apeak)	Module size
I_{max} (Arms)	Module size / , $\sqrt{2}$
16Bit	16 Bit = -2^{15} to $+2^{15}-1$
32Bit	32 Bit = -2^{31} to $+2^{31}-1$

5.7 NUMDrive C application - User example

5.7.1 Speed loop parameters optimization V51 / V52

The procedure described below refers to the adjustment of axis motors.
For spindle motors the parameters related to speed loop are normally modified only when there is a need to improve particular performances
(positioning precision, overshoot on speed, excessive mechanical noise).

Before starting the speed loop parameter optimization, ensure that the following preliminary operations have been carried out:

- Motor / drive association correctly carried out.
- Correct configuration of bit parameters concerning the specific application.

5.7.1.1 Speed loop parameters optimization procedure

- Open WPM and select the digital oscilloscope
- Select motor speed on TP1 (MP code 001) and torque current on TP2 (MP code 050)
- Operate the motor with small reference steps (~10% of maximum speed) and check the response of the speed and current signals.

The figures 1, 2, 3, 4 of the following pages are referred to the use of WPM digital oscilloscope.

5.7.1.2 Speed loop parameters optimization procedure A

If the inertia of the load (axis) seen from the motor is greater than that of the motor itself and the drive line is sufficiently rigid, the quantities should run a trace as shown in figure 1, characterized by large overshoot and long re-entry time.

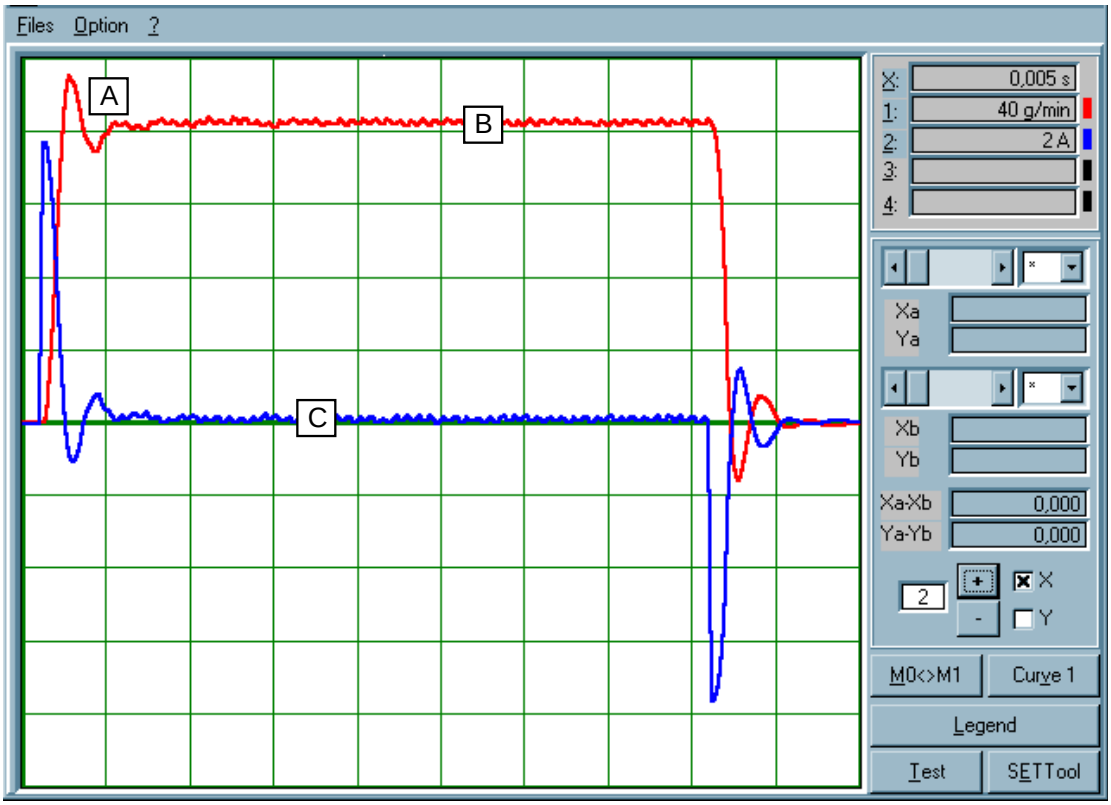


Fig. 1

A = Overshoot
B = Speed
C = Torque current

To improve the dynamic response of the system (see fig. 1) the value of the speed cycle's proportional gain will have to be increased by adjusting the V51 parameter in general by the same percentage of the load inertia in relation to the motor.

Enter the value of V51 parameter so as to obtain a response similar to that of figure 2. Pay particular attention to the motor noise and avoid setting the V51 parameter with excessively high values with could cause performance instability.

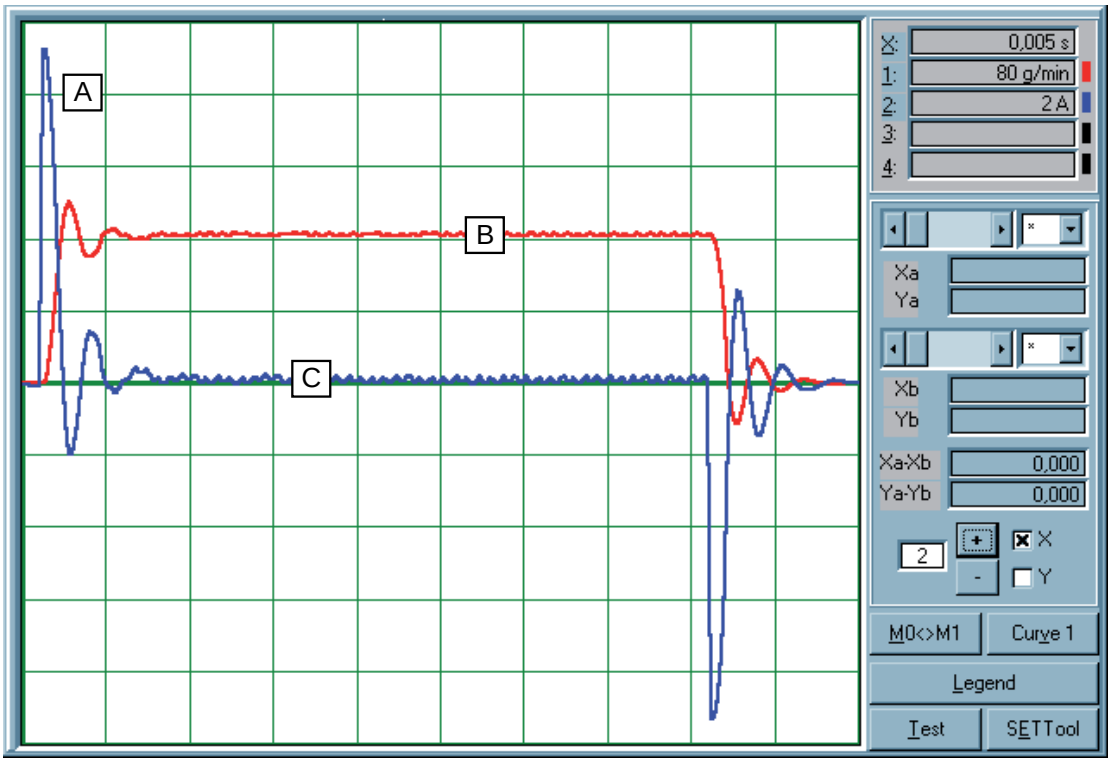


Fig. 2

A = Overshoot
B = Speed
C = Torque current

The parameters adjustment set [1] V1051, V1052 is carried out as for the V51 and V52 parameters.

5.7.1.3 Speed loop parameters optimization procedure B

If on the contrary the drive line is more flexible the trace should be similar to that shown on figure 3.

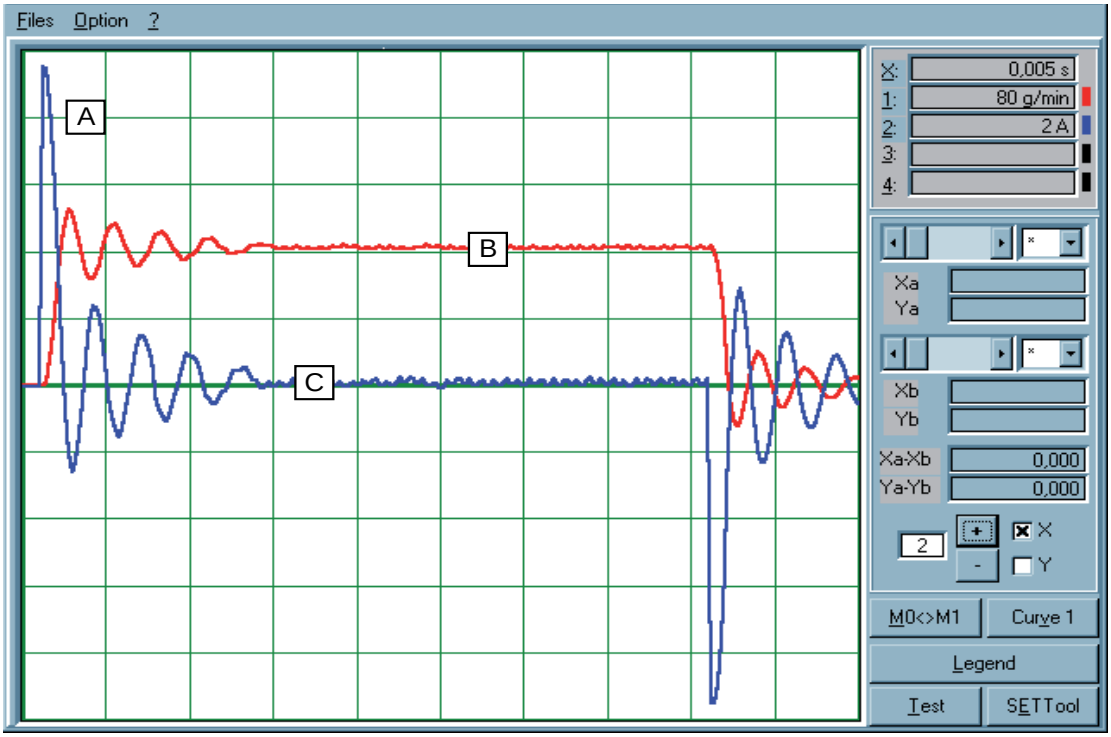


Fig. 3

A = Overshoot
B = Speed
C = Torque current

To improve this response (fig. 3) the value of the speed cycle integral gain must be reduced by adjusting the V52 parameteri n order to obtain a response similar to that in figure 4.

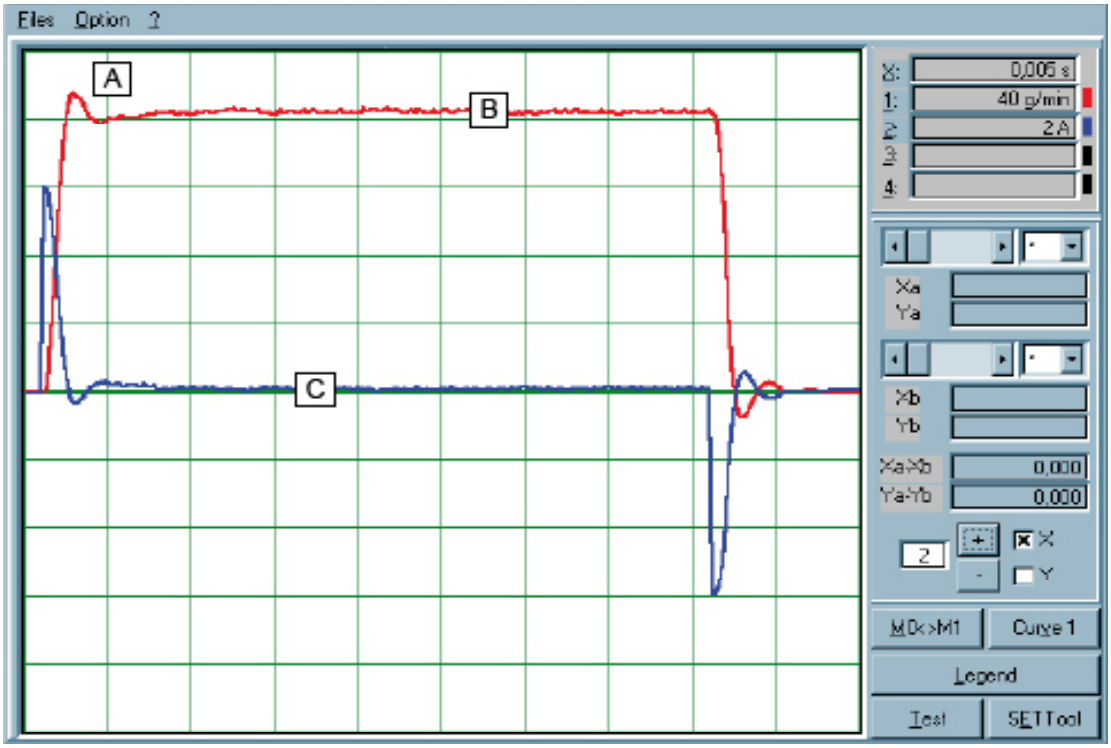


Fig. 4

A = Overshoot
B = Speed
C = Torque current

The parameters adjustment set [1] V1051, V1052 is carried out as for the V51 and V52 parameters.

5.7.2 Programmable relais 1 or 2 application examples

5.7.2.1 Control of an axis in motion (using relay no. 1 - Output on PROGO1)

We give to manage the programmable relay 1 in threshold mode with 0.100s delay in such a way that its signals to the PLC:
- relay ON condition (axis in motion) when the speed value is >100 rpm
- relay OFF condition (axis stationary) when the speed is < 80 rpm.

Adjusting procedure

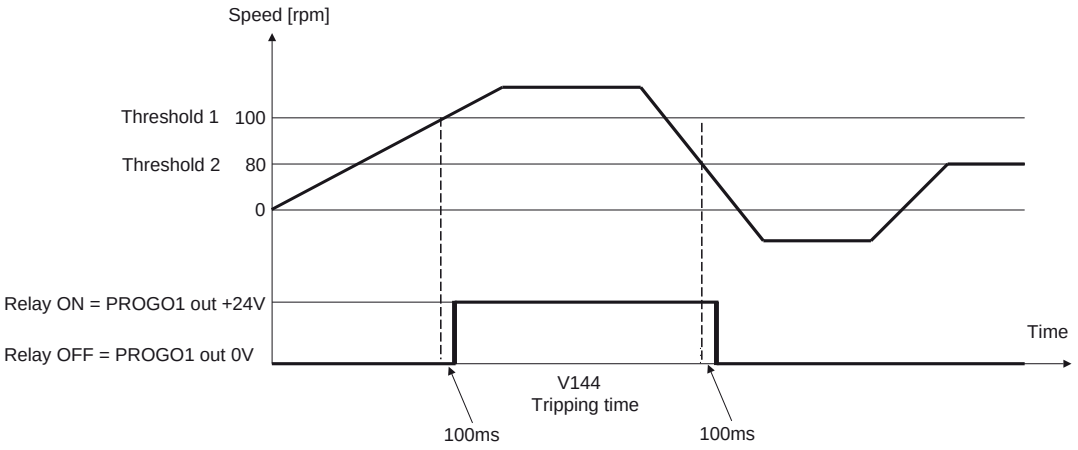
- Select the quantity to be associated to programmable relay by means V140 =1 (Mechanical motor speed)

- Select V139 parameter (Programmable relay configuration) as follows:
  - Configure Bit 2 = 1 (Absolute value for progr. relay no. 1)
  - Configure Bit 4 = 0 (Programmable relay 1 in threshold mode)
  - Configure Bit 6 = 1 (Programmable relay 1 as source mode)

- Select V141 parameter (Threshold 1) and enter 100 (speed value in rpm)
- Select V142 parameter (Threshold 2) and enter 80 (speed value in rpm)
- Select V143 parameter (Scale factor) and enter 1342.18

- Select V144 parameter (Time threshold) and enter 100 (trip to 100ms)

- By programming the relay as indicated above we will obtain the following motor speed control as in figure.



5.7.3 Motor brake management

5.7.3.1 Management of motor holding brake

It is possible to directly manage the motor holding brake command dedicating one of the two programmable relay outputs to this function.

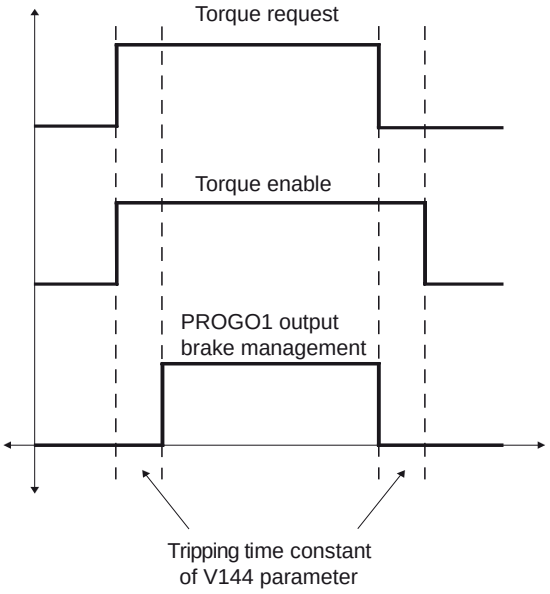
The motor holding brake may be configured by means of V139 bit 0.
By configuring bit 0 = 1 the brake management is activated.
With bit 0 = 0 the brake management is disabled.

Bit 1 allows to assign the brake management to either PROGO1 or PROGO2 relay output.
(pin2 and 3 of S7 connector)

Adjusting procedure

Application example with PROGO1 relay output.

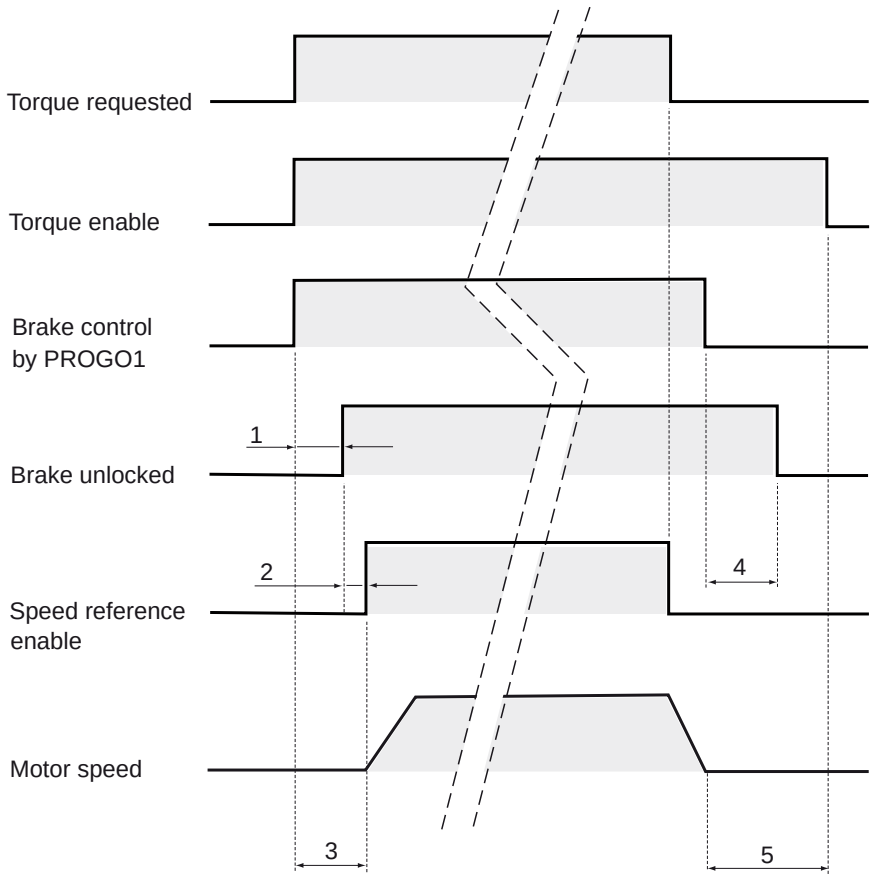
- By setting V139 bit 0 = 1 the brake management function is activated.
- By setting V139 bit 1 = 0 the brake management function can be associated to one of the relay output PROGO1.
- By means of V144 parameter it is possible to define the delay time between torque enable and holding brake unlock events when the relay output selected is PROGO1.



5.7.3.2 Use of the Programmable relays in modality brake management

PROGO1 use in modality brake management

- Bit 0 = 1
- Bit 1 = 0



1 Brake unlocked time (see V144).
2 Time between “ Brake unlocked and Speed reference enable. This time may be null.
3 Total time of torque enable at null speed, before the motor start.
4 Brake locked time.
5 Total time of torque enable at null speed (see V144).

In case of alarm with motor coast down the brake is de-energized without waiting the motor at null speed.

5.7.3.3 Management of motor holding brake in case of alarms

Adjusting procedure

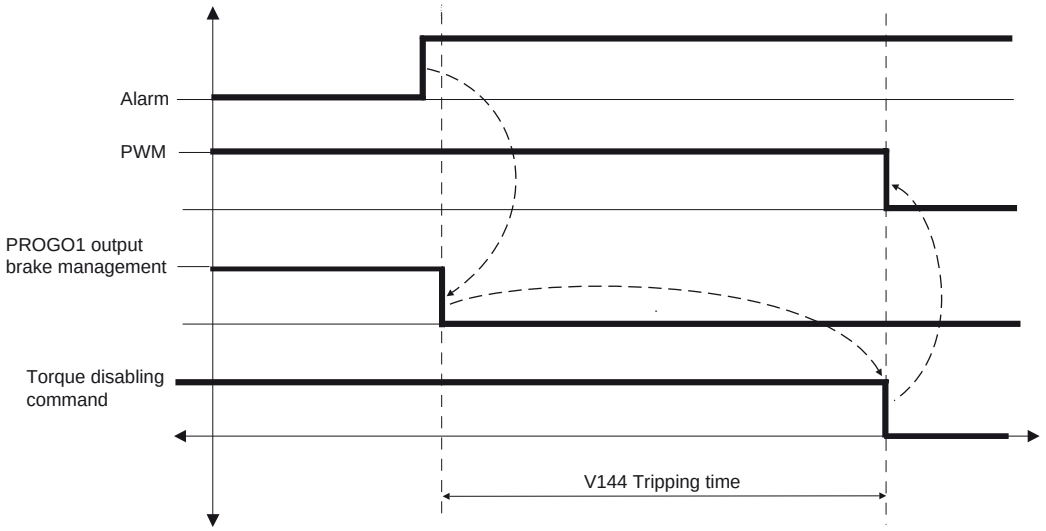
Operation of brake management in case of fatal alarms (A02, A06, A12)

- For all fatal alarms, in case of alarm, it is possible to follow the brake insertion sequence with subsequent disabling operation but the brake will not be operated till the drive is disabled.

Adjusting procedure

Operation of brake management in case of non fatal alarms

- For all non fatal alarms, in case of alarm, the brake command is immediately active while the torque will be subsequently disabled at the end of the time selected on V144/V154 parameters (time constant). In case of alarm configured with braking operation, the axis will first be stopped with torque enabled, then the holding brake will be inserted and finally the torque will be disabled.



5.7.4 MULTI /DIVI parameters utilization V306, V307

5.7.4.1 MULTI/DIVI with S1 sensor type P, Q

Adjusting procedure

The NUMDrive C CANopen drive uses “NC units” to measure position, speed, acceleration.
One “NC unit” is the minimum increment the user want to appreciate (beware, it is not an indication of the maximum resolution achievable with the system: is an expression of the desired resolution).
The “Multi/Divi” selection is done to have all the measure-related objects expressed in this unit.
With an example the things are simpler:

- the axis is a linear one, with a rotative motor and a screw;
- the measure should be expressed in micrometers: this means, for example, object 6064h will be directly “micron”;
- the sensor is a 1024 pulse encoder;
- there is a 1:1 gearbox;
- the screw is 10mm/turn;
- this means the measure will increment by 10.000 microns every turn;
- the suggested Multi/Divi is 10000/4096
  (Please refer to the chapter 5.5 Parameters description)

Application example of MULTI/DIVI parameters with S1 sensor P, Q (high resolution multiturn)

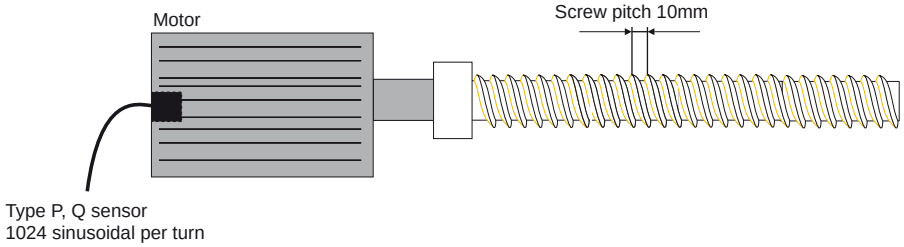
- Physical units resolution desidered on position - related objectes like 6064h: (m).
- Screw pitch: 10mm.
- Axis / motor ratio: motor fitted directly on the screw.
- Motor sensor (P or Q type) 1024 sinusoids per rev. therefore the counting pulse number for one motor sensor revolution = 1024 * 4 = 4096.
  Where: 1024 = Number of sinusoids per motor rev.
  4 = Internal multiplier coefficient.

See figure here-under.
Taking into consideration that one motor revolution is equal to a linear shift of 10mm (10.000µ), to convert the sensor feedback to “NC Unit” it is necessary to set-up the following MULTI/DIVI parameters to have the measure incremented by 10.000 at every turn (i.e. every 1024*4 counting pulses):

- V306 (MULTI MEASURE) = 10000
  (Shift in CNC internal measurement unit for one motor rev.)
- V307 (DIVI MEASURE) = 4096
  (Counts occurred during the movement selected into V306 parameter)

ATTENTION! Only for type P motor encoder.

Check that during the complete axis stroke the value detected on test point MP223 (type S1 encoder revs counter) does not reset from 4095 to 0 or vice-versa. Otherwise it is necessary to remove the motor and change the mechanical position in order to avoid the problem described above.
If the above condition is not met, move the axis to the middle of the stroke, remove the motor turn the motor shaft until the value read on test point MP223 ~4096. Then, re-assemble the motor. (Otherwise carry out the writing procedure of the absolute position for STEGMANN)



The resolutions obtained by counting are used exclusively for MULTI/DIVI calculation purposes, therefore the values do not indicate the system resolution.

5.7.4.2 MULTI/DIVI with type Q motor sensor for spindle motor with C axis application

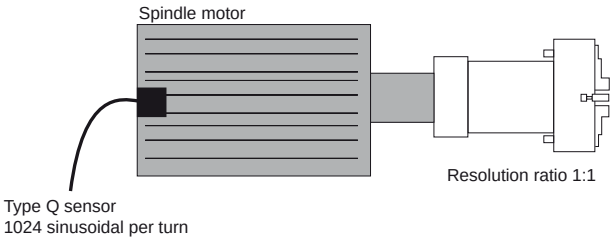
Adjusting procedure

Application example of MULTI/DIVI parameters with S1 sensor type Q for spindle motor reduction ratio 1:1 with C axis application

- Type Q motor sensor with 1024 periods for motor rev, therefore the counting pulse number for one spindle revolution = 1024 * 4 = 4096
- Resolution = 1/10000° (this means 360° *10.000 increments every turn, see “NC Unit” on chapter 5.2.1, 5.3.1)

By dividing numerator and denominator by 128 we have:

- V1306 (MULTI MEASURE) = 3.600.000 = >28125
- V1307 (DIVI MEASURE) = 4096 = >32

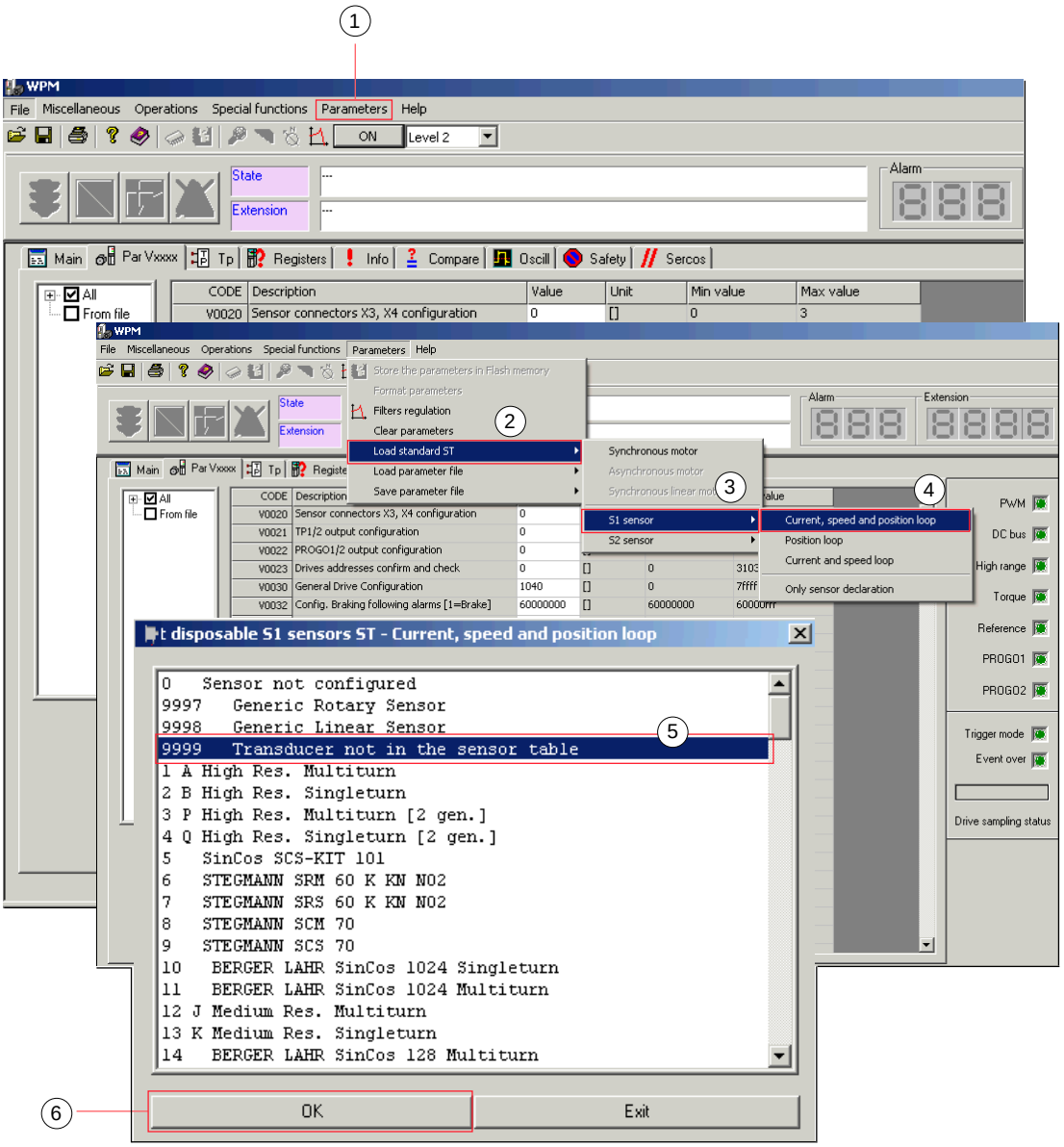


5.7.5 Sensors configuration and applications

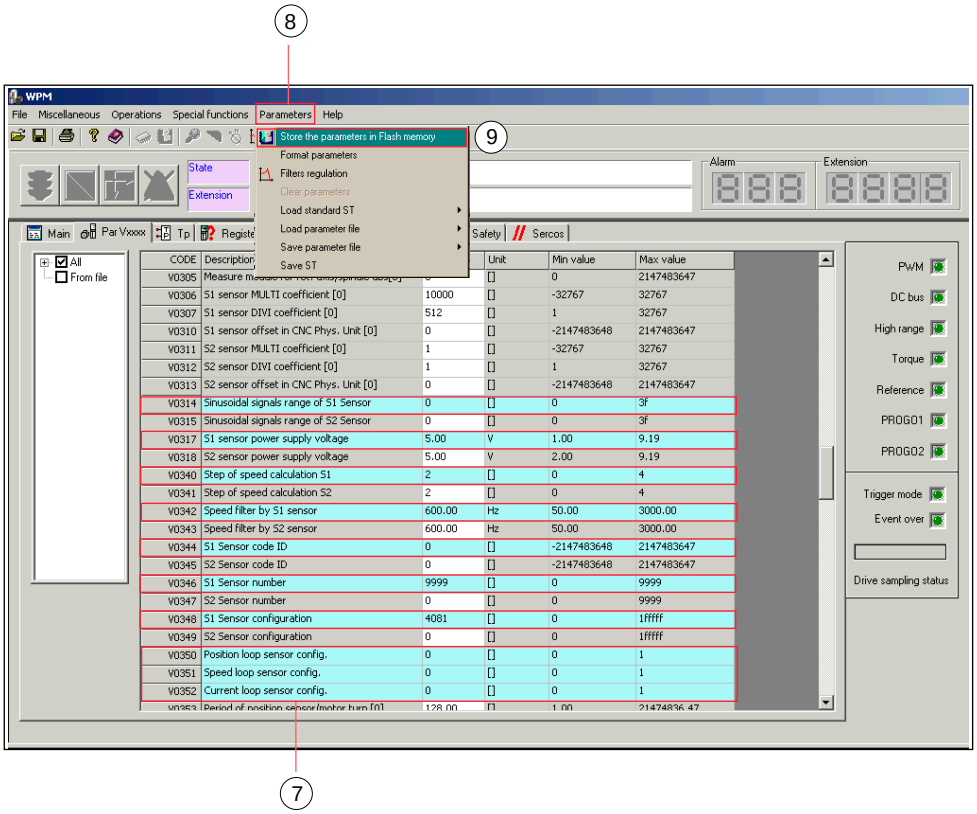
5.7.5.1 Enter new sensor configuration

If your sensor is not shown in the WPM sensor table (see chapter 5.1), therefore enter new sensor configuration is required, before connecting it, please load the standard sensor setup parameters as indicated here-below.

Always take into consideration the sensor data sheet and select the appropriate values and adjusting appropriate parameters.



1	Open WPM , than select: → Parameters
2	→ Load standard ST
3	→ Select the sensor → S1 or S2 sensor
4	→ Select the loop → Current, Speed or Position to open the sensor selection table
5	→ Select “9999 “Transducer not in the sensor table → Press OK to open the sensor configuration page



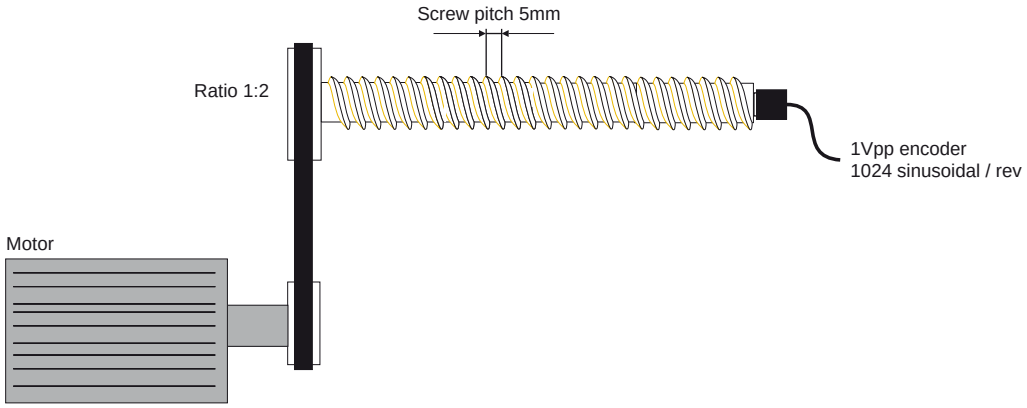
7	→ Fill with a right values all the highlight sections taking into consideration the sensor data sheet.
8	→ Select “Parameters” to make the storage of the parameters
9	→ Than click on “Store the parameters in Flash memory.

These standard values will be loaded for example on the S2 sensor parameters.
The same loading procedure is given for S1 sensor i.e. V317, V348, 368....

Standard sensor parameters	
V318, V349, V353, V1353	For all type of new sensor.
V370, V371, V375	For rotative sensor only.
V378	For coded reference sensor only.
V390, V391	For linear scale sensor only.

On the following pages the most common sensors application examples are introduced.

5.7.5.2 Sinusoidal rotary encoder



- Desidered units for position measure = (µm)
- Screw pitch : 5mm (5000µm)
- 1Vpp singletour sinusoidal encoder with 1024 sinusoid/rev therefore the number of counts for each encoder revolution = 1024 * 4 = 4096
  Where: 1024 = Number of sinusoids for one encoder rev.
  4 = Internal multiplier coefficient.

Sinusoidal rotary sensor with 1 top mark HOMING on S2 for position loop closure		
Basic parameters used		**Parameters adjustment value**
V311	S2 sensor MULTI coefficient	[5000]
V312	S2 sensor DIVI coefficient	[4096]
V315	- Sinusoidal signal S2 - bit 2 = 0 - bit 3 = 0	Standard alarm A.38.254 threshold 50%
V318	Enter the appropriate sensor voltage value	See sensor data sheet
V343	Speed filter by S2 sensor	[600]
V349	- S2 sensor type - bit 2,1,0 = 001 - bit 5,4,3 = 000 - bit 7 = 1 - bit 14 = 1	[00004081]
V350	- Position loop sensor config. - bit 0 = 1	-
V353	Periods of position sensor / motor turn	[512]
V370	Periods by turn sensor S2	[1024]
V371	Absolute resolution by turn sensor S2	[4096]
V375	Number of absolute sensor turns S2	[1]

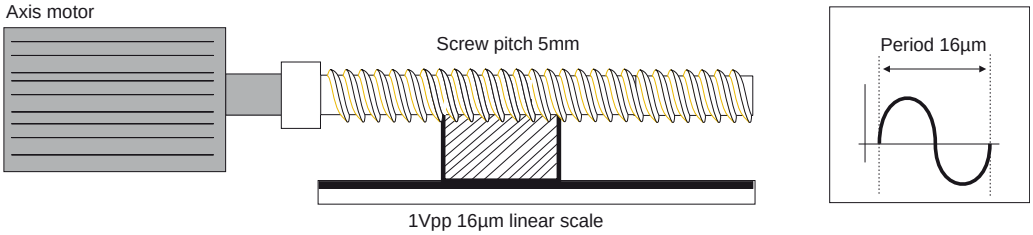
With this setting, the measure on the object 6064h will incremented by 5000 units every screw turn having a scale of 1µm.

Different parameters used if ENDAT		Parameters adjustment value
V349	- S2 sensor type - bit 2,1,0 = 001 - bit 5,4,3 = 010 - bit 6 = 1 - bit 7 = 1 - bit 14 = 1 - bit 16 = 1	[000040D1]
V370	Periods by turn sensor S2	[1024]
V371	Absolute resolution by turn sensor S2	See sensor data sheet
V375	Number of absolute sensor turns S2	

Different parameters used if Coded reference		Parameters adjustment value
V349	- S2 sensor type - bit 2,1,0 = 001 - bit 5,4,3 = 011 - bit 7 = 1 - bit 14 = 1 - bit 16 = 1	[00014099]
V378	Intervals between coded reference S2	See sensor data sheet

The resolutions obtained by counting are used exclusively for MULTI/ DIVI calculation purposes, therefore the values do not indicate the system resolution.

5.7.5.3 Sinusoidal linear scale encoder



- Desidered units for position measure = (µm)
- 1Vpp sinusoidal scale with one sinusoid every 16µm
  therefore the number of counts for each sinusoid = 4

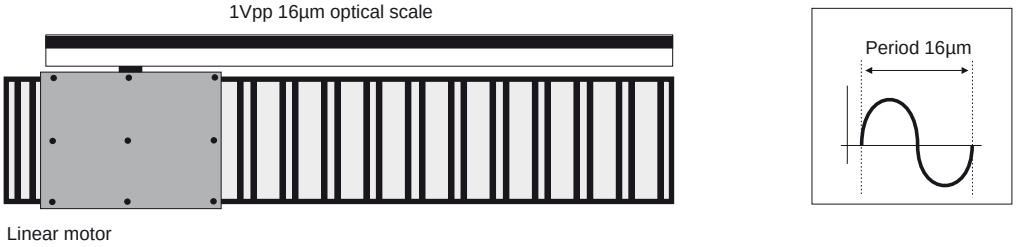
Sinusoidal linear scale with 1 top mark HOMING sensor on S2 for position loop closure		
Basic parameters used		**Parameters adjustment value**
V311	S2 sensor MULTI coefficient	[16]
V312	S2 sensor DIVI coefficient	[4]
V315	- Sinusoidal signal S2 - bit 2 = 0 - bit 3 = 0	Standard alarm A.38.254 threshold 50%
V318	Enter the appropriate sensor voltage value	See sensor data sheet
V343	Speed filter by S2 sensor	[600]
V347	S2 sensor number	[9999]
V349	- S2 sensor type - bit 2,1,0 = 001 - bit 5,4,3 = 000 - bit 7 = 1 - bit 8 = 1 - bit 14 = 1	[00004181]
V350	- Position loop sensor config. - bit 0 = 1	[1]
V353	Periods of position sensor / motor turn	[312.5]
V390	Periods of linear sensor S2	[16]

Different parameters used if ENDAT		Parameters adjustment value
V349	- S2 sensor type - bit 2,1,0 = 001 - bit 5,4,3 = 010 - bit 6 = 1 - bit 7 = 1 - bit 8 = 1 - bit 14 = 1 - bit 16 = 1	[000141D1]
V390	Periods of linear sensor S2	[16]
V391	Absolute resolution of linear sensor S2	See sensor data sheet

Different parameters used if Coded reference		Parameters adjustment value
V349	- S2 sensor type - bit 2,1,0 = 001 - bit 5,4,3 = 011 - bit 7 = 1 - bit 8 = 1 - bit 14 = 1 - bit 16 = 1	[00014199]
V378	Intervals between coded reference S2	See sensor data sheet

The resolutions obtained by counting are used exclusively for MULTI/ DIVI calculation purposes, therefore the values do not indicate the system resolution.

5.7.5.4 Linear sinusoidal encoder for linear motor



- Desidered units for position measure = (µm)
- 1Vpp sinusoidal scale with one sinusoid every 16µm
  therefore the number of counts for each sinusoid = 4

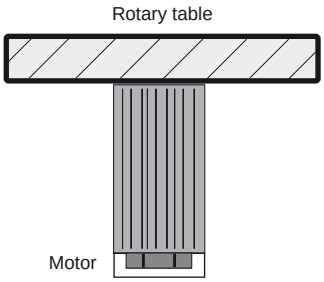
Sinusoidal linear scale with 1 top mark HOMING sensor on S1 for current, position and speed loop closure		
Basic parameters used		**Parameters adjustment value**
V258	S1 Motor sensor type	[X]
V306	S1 sensor MULTI coefficient	[16]
V307	S1 sensor DIVI coefficient	[4]
V314	- Sinusoidal signal S1 - bit 2 = 0 - bit 3 = 0	Standard alarm A.09.254 threshold 50%
V317	Enter the appropriate sensor voltage value	See sensor data sheet
V342	Speed filter by S1 sensor	[600]
V346	S1 sensor number	[9999]
V348	- S1 sensor type - bit 0 = 1 - bit 7 = 1 - bit 8 = 1 - bit 14 = 1	[00004181]
V350	- Position loop sensor config. - bit 0 = 1	[0]
V351	Speed loop sensor configuration	[0]
V352	Current loop sensor configuration	[0]
V380	Periods by turn sensor S1	[16]

Different parameters used if ENDAT		Parameters adjustment value
V348	- S1 sensor type - bit 2,1,0 = 001 - bit 5,4,3 = 010 - bit 6 = 1 - bit 7 = 1 - bit 8 = 1 - bit 14 = 1 - bit 16 = 1	[000141D1]
V380	Periods of linear sensor S1	[16]
V381	Absolute resolution of linear sensor S1	See sensor data sheet

Different parameters used if Coded reference		Parameters adjustment value
V348	- S1 sensor type - bit 2,1,0 = 001 - bit 5,4,3 = 011 - bit 7 = 1 - bit 8 = 1 - bit 14 = 1 - bit 16 = 1	[00014199]
V368	Intervals between coded reference S1	See sensor data sheet

The resolutions obtained by counting are used exclusively for MULTI/ DIVI calculation purposes, therefore the values do not indicate the system resolution.

5.7.5.5 Rotary sinusoidal encoder for direct torque motor



- Desired resolution on object 6064h(1) = 1/10000°
- 1Vpp encoder with 32768 sinusoids per revolution
  therefore the number of counts for each table
  rotation = 32768 * 4 = 131072
  Where: 32768 = number of counts for each encoder revolution
  4 = Internal multiplier coefficient.

(1) To select the desired resolution, the value on 6064h should be increment by 360*10.000 for each turn, so the values for MULTI / DIVI are:

$$\frac{MULTI}{DIVI} = \frac{360° * 10.000}{32768 * 4} = \frac{28125}{1024}$$

Sinusoidal rotary sensor with 1 top mark HOMING on S1 for current, speed and position loop closure.		
Basic parameters used		**Parameters adjustment value**
V258	S1 Motor sensor type	[X]
V306	S1 sensor MULTI coefficient	[28125]
V307	S1 sensor DIVI coefficient	[1024]
V314	- Sinusoidal signal S1 - bit 2 = 0 - bit 3 = 0	Standard alarm A.09.254 threshold 50%
V317	Enter the appropriate sensor voltage value	See sensor data sheet
V342	Speed filter by S1 sensor	[600]
V346	S1 sensor number	[9999]
V348	- S1 sensor type - bit 2,1,0 = 001 - bit 5,4,3 = 000 - bit 7 = 1 - bit 14 = 1	[00004081]
V350	- Position loop sensor config. - bit 0 = 0	[0]
V351	Speed loop sensor configuration bit 0 = 0	[0]
V352	Current loop sensor configuration bit 0 = 0	[0]
V353	Periods of position sensor / motor turn	[32768]
V360	Periods by turn sensor S1	[32768]
V361	Absolute resolution by turn sensor S1	[131072]
V365	Number of absolute sensor turns S1	[1]

Different parameters used if ENDAT		Parameters adjustment value
V348	- S1 sensor type - bit 2,1,0 = 001 - bit 5,4,3 = 010 - bit 6 = 1 - bit 7 = 1 - bit 14 = 1 - bit 16 = 1	[000140D1]
V360	Periods by turn sensor S1	[32768]
V361	Absolute resolution by turn sensor S1	See sensor data sheet
V365	Number of absolute sensor turns S1	See sensor data sheet

Different parameters used if Coded reference		Parameters adjustment value
V348	- S1 sensor type - bit 2,1,0 = 001 - bit 5,4,3 = 011 - bit 7 = 1 - bit 14 = 1 - bit 16 = 1	[00014099]
V368	Intervals between coded reference S1	See sensor data sheet

The resolutions obtained by counting are used exclusively for MULTI/ DIVI calculation purposes, therefore the values do not indicate the system resolution.

5.7.6 Homing mode and homing setting

5.7.6.1 Homing mode information and functional description

General information

This clause describes the method by which a drive seeks the home position (also called, the datum, reference point or zero point). There are various methods of achieving this using limit switches at the ends of travel or a home switch (zero point switch) in mid-travel, most of the methods also use the index (zero) pulse train from an incremental encoder.

Functional description

Figure below shows the defined input objects as well as the output objects. The user may specify the speeds, acceleration and the method of homing. There is a further object home offset, which allows the user to displace zero in the user's coordinate system from the home position.
There is no output data except for those bits in the statusword, which return the status or result of the homing process and the demand to the position control loops.

There are two homing speeds; in a typical cycle, the faster speed is used to find the home switch and the slower speed is used to find the index pulse. The manufacturer is allowed some discretion in the use of these speeds as the response to the signals may be dependent upon the hardware used.

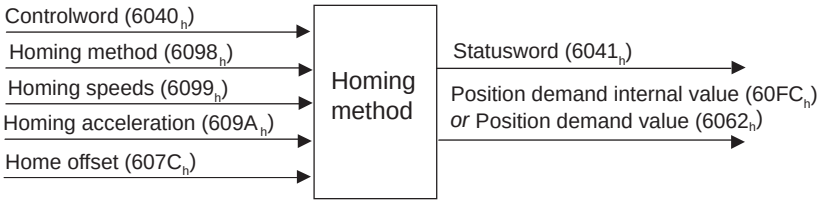


Fig.1 Homing mode function

By choosing a homing method, the following behaviour is determined:
- the homing signal (positive limit switch, negative limit switch, home switch),
- the direction of actuation and where appropriate,
- the position of the index pulse.

The home position and the zero position are offset by the home offset; see the definition of home offset for how this offset is used. An encircled number in the figures Figure 2 to Figure 9 indicates the code for selection of this homing position. The direction of movement is also indicated.

There are four sources of homing signal available: These are the negative and positive limit switches, the home switch and the index pulse from an encoder.

In case, that a limit switch has reached the drive shall move in the other direction to leave the position.

In the diagrams of homing sequences shown below, the encoder count increases as the axis's position moves to the right, in other words, the left is the minimum position and the right is the maximum position.
For the operation of positioning drives, an exact knowledge of the absolute position is normally required. Since for cost reasons, drives often do not have an absolute encoder, a homing operation is necessary. There are several, application-specific methods. The homing method is used for selection.
The exact sequence of the homing operation is clearly described by the method. In some circumstances, a drive device has several methods to choose from, using the homing method.

5.7.6.2 General definitions and Homing Methods

Method 1: Homing on negative limit switch and index pulse

Using this method as shown in Figure 2, the initial direction of movement shall be leftward if the negative limit switch is inactive (here: low). The home position shall be at the first index pulse to the right of the position where the negative limit switch becomes inactive.

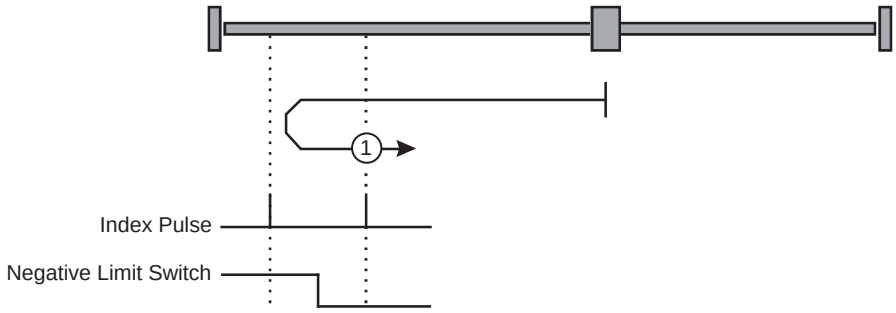


Fig.2 Homing on negative limit switch and index pulse

Method 2: Homing on positive limit switch and index pulse

Using this method as shown in Figure 3, the initial direction of movement shall be rightward if the positive limit switch is inactive (here: low). The position of home shall be at the first index pulse to the left of the position where the positive limit switch becomes inactive.

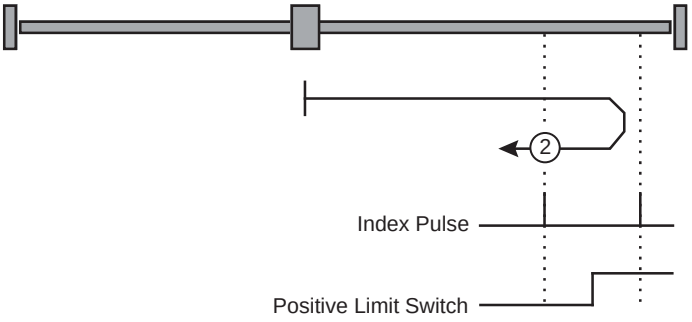


Fig.3 Homing on positive limit switch and index pulse

Method 3 and 4: Homing on positive home switch and index pulse

Using these methods as shown in Figure 4, the initial direction of movement shall be dependent on the state of the home switch. The home position shall be at the index pulse to either to the left or the right of the point where the home switch changes state. If the initial position is situated so that the direction of movement shall reverse during homing, the point at which the reversal takes place is anywhere after a change of state of the home switch.

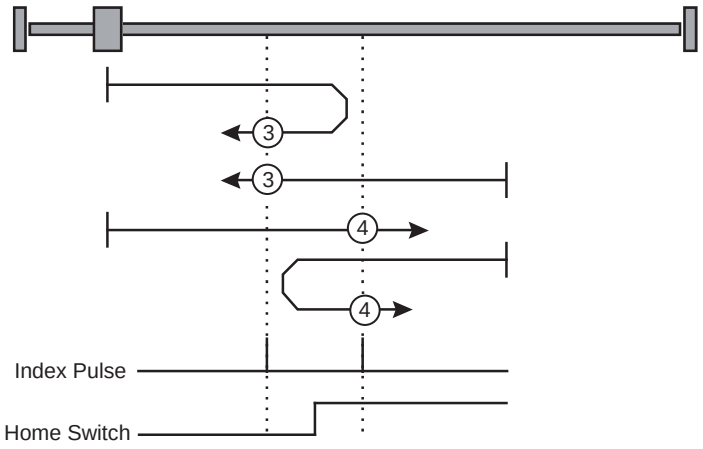


Fig.4 Homing on positive limit switch and index pulse

Method 5 and 6: Homing on negative home switch and index pulse

Using these methods as shown in Figure 5, the initial direction of movement shall be dependent on the state of the home switch. The home position shall be at the index pulse to either to the left or the right of the point where the home switch changes state. If the initial position is situated so that the direction of movement shall reverse during homing, the point at which the reversal takes place is anywhere after a change of state of the home switch.

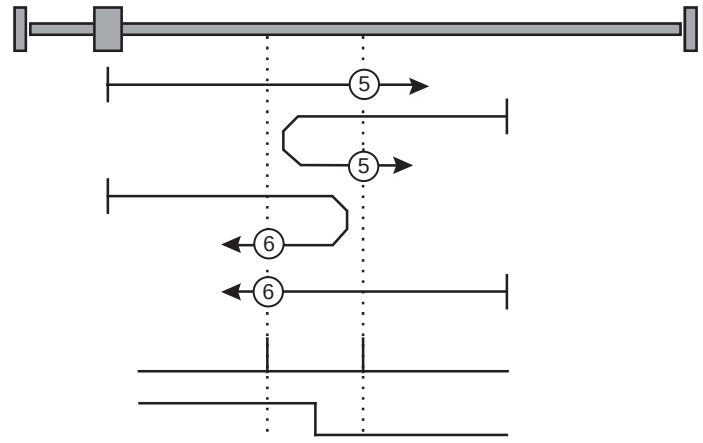


Fig.5 Homing on negative home switch and index pulse

Method 7 to 14: Homing on home switch and index pulse

These methods use a home switch, which is active over only a portion of the travel, in effect the switch has a ‘momentary’ action as the axis’s position sweeps past the switch.

Using the methods 7 to 10, the initial direction of movement shall be to the right, and using methods 11 to 14, the initial direction of movement shall be to the left except if the home switch is active at the start of the motion. In this case, the initial direction of motion shall be dependent on the edge being sought.
The home position shall be at the index pulse on either side of the rising or falling edges of the home switch, as shown in Figure 6 and Figure 7. If the initial direction of movement leads away from the home switch, the drive shall reverse on encountering the relevant limit switch.

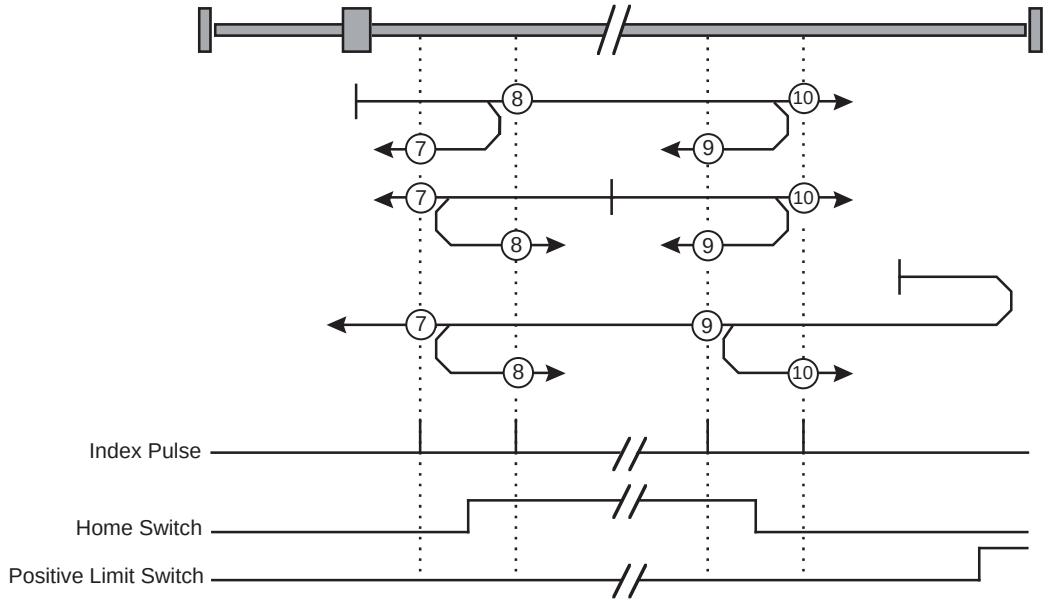


Fig.6 Homing on home switch and index pulse – positive initial motion

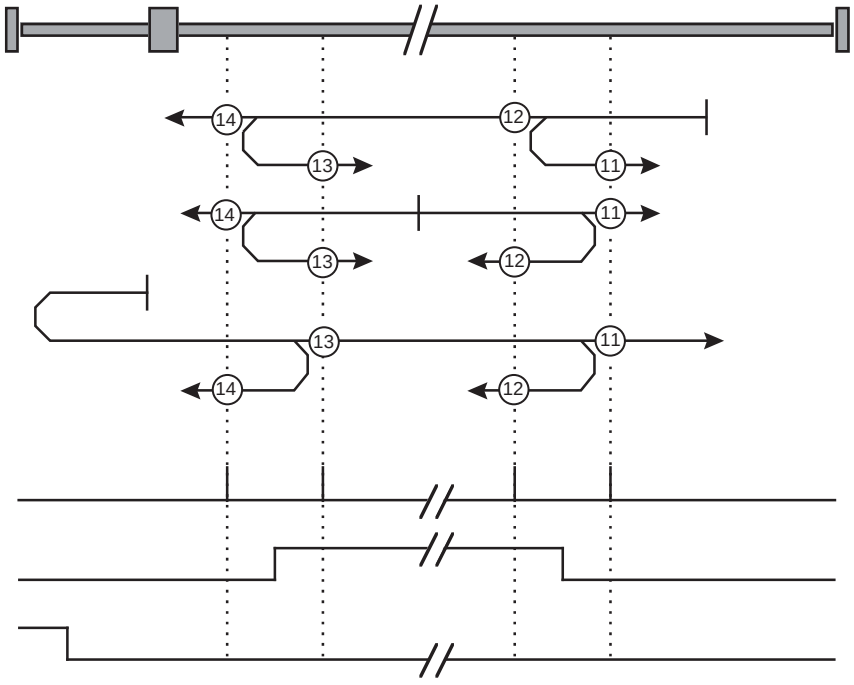


Fig.7 Homing on home switch and index pulse – negative initial motion

Method 15 and 16: Reserved

These methods are reserved.

Method 17 to 30: Homing without index pulse

These methods are similar to methods 1 to 14 except that the home position is not dependent on the index pulse but only dependent on the relevant home or limit switch transitions. For example methods 19 and 20 are similar to methods 3 and 4 as shown in Figure 8.

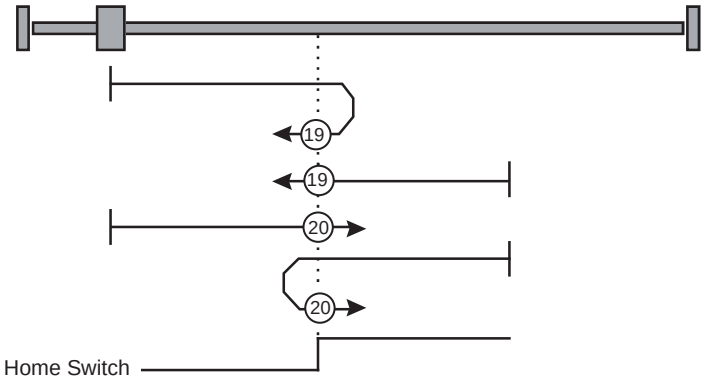


Fig.8 Homing on positive home switch

Method 31 and 32: Reserved

These methods are reserved.

Method 33 and 34: Homing on index pulse

Using these methods, the direction of homing is negative or positive respectively. The home position shall be at the index pulse found in the selected direction as shown in Figure 9.

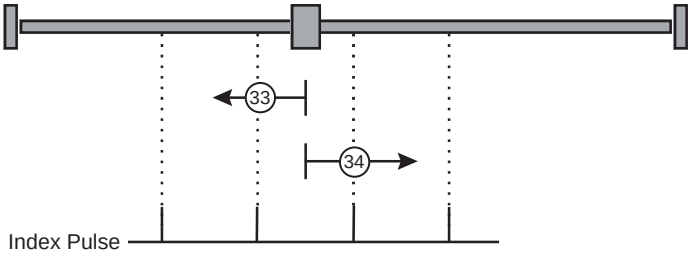


Fig.9 Homing on index pulse

Method 35: Homing on index pulse

In this method, the current position shall be taken to be the home position. This method does not require the drive device to be in *operational enabled* state.

5.7.7 Index Pulse identification (Pulse Mark)

The electrical contact opening signal must be clean without bounce.
The switch must be set so that the contact opens between one-quarter and three-quarters of the distance separating two zero pulses. This is to avoid coincidence between switch operation and the zero pulse, which could cause a random shift by a distance equal to that separating two zero pulses.

The switch size should be such that the contact opens before detection of the sensor zero pulse and remains open until the axis stops after detection of the open contact on the switch input.

When the impulse of Pulse Mark is ON (MP240/241) check that the Sine and Cosine signals (see MP202/203 - MP204/205) must be both positive.

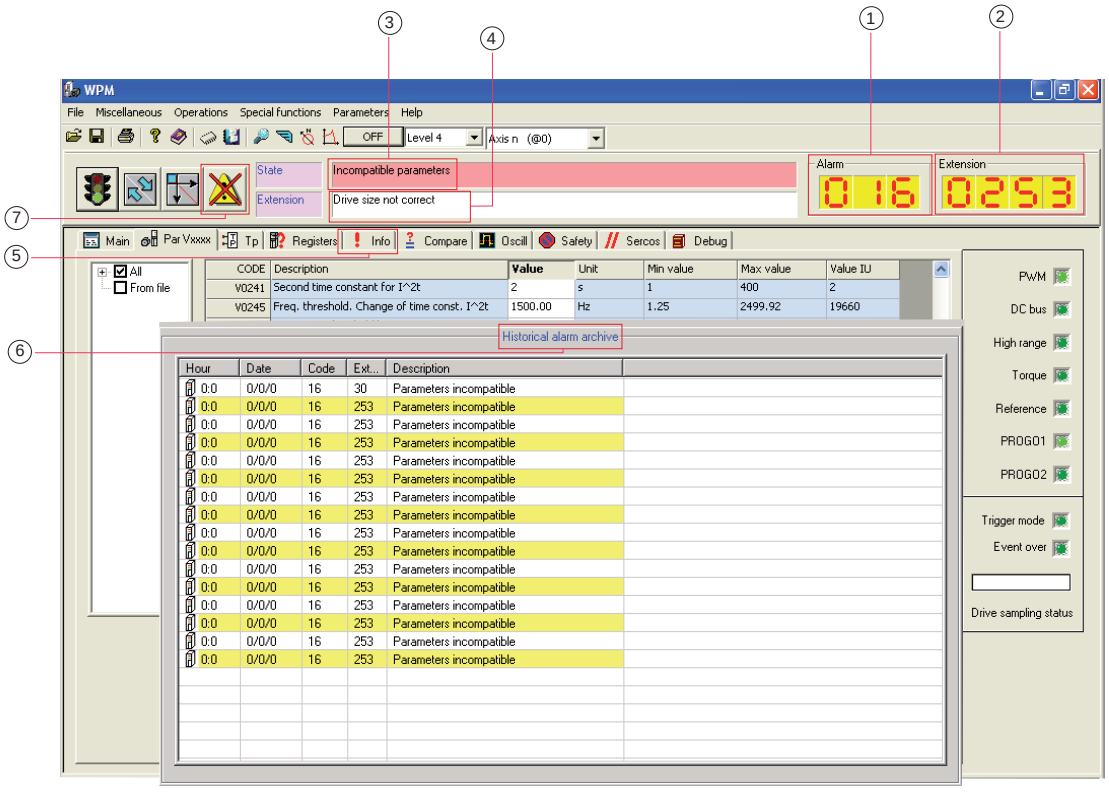
5.8 Diagnostic signalling and software alarms

5.8.1 Alarm signalling

By means the CANopen Library as indicated on chapters 3.9.11, 3.10.11 Drive Control “Alarms” and “AlarmReset” or by WPM as indicated in the figure below, it possible to control in real time the drive status and all drive devices information.
If a fault occurs, which causes an alarm to operate, the drive’s red fault Led illuminates and the alarm is displayed on the Controller or upon WPM as indicated in the figure below .

Even, for any alarms, additional informations relating to the alarm extensions and some warning informations are available. (See “Drive alarms and alarms extension” on chapter 5.8.8)

For example, if we consider an alarm generated by “Incompatible parameters (A.16)”, by programming and enabling the above-mentioned parameters in an appropriate way, it is possible to stop the motor by inertia or with braking, with or without tripping of the line remote switch, just as it is possible, if previously programmed, or to reset the alarm memorized by Controller as indicated on the chapter 3.9.11, 3.10.11 “AlarmReset” or by WPM as indicated here below.

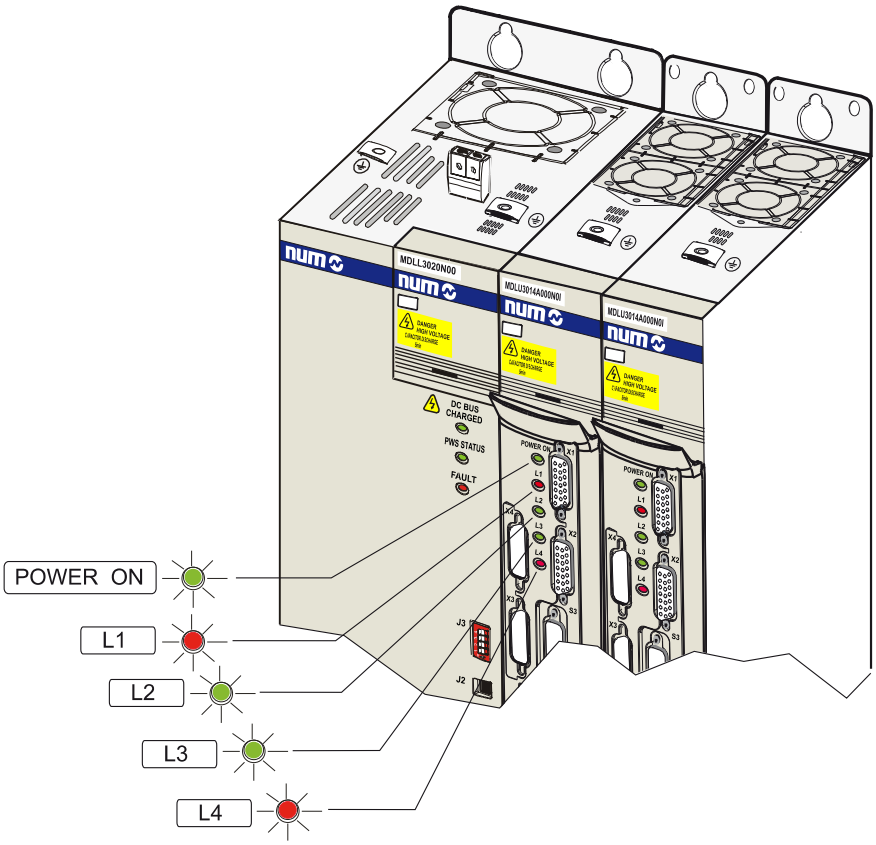


1	Alarm display
2	Alarm extension
3	Alarm description
4	Alarm extension description
5	Click on “Info” icon to display the “Historical alarm archive”
6	“Historical alarm archive” page
7	Click on the icon Reset icon to reset an alarm occurred

For more detailed information about the use of WPM tool, please refer to the WPM Instruction manual upon “Help” windows.

5.8.2 Alarm signalling

If a fault occurs which causes an alarm to operate, the drive's FAULT LED illuminates, and the alarm is displayed by WPM or by the Controller.
Also, for any alarms additional information relating to the alarm involved is available.
(See "Drive alarms extension" at chapter 5.8.8)

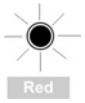


5.8.2.1 Led POWER ON status

MDLU3 LED "POWER ON" status		
LED condition	LED status "ON"	- Module auxiliary voltage OK.
	LED status "OFF"	- Module auxiliary voltage not present. - Power board failure on auxiliary voltage regulator.
	LED status "Flashing"	- Overload on auxiliary voltage regulator. - A fault on the Power Unit or on the Control Unit causes the overload. One of these parts must be substituted.

The SERIAL LINK FAULT led can display two diffent drive status:
- LED always ON: Heartbeat fault or heartbeat never started.

5.8.3 Mono-Axis alarm signalling status

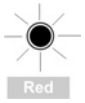
LEDs		Led signalling status			
FAULT condition	Descriptions	**L1 (SLF)** 	**L2**	**L3**	**L4 (Fault)**
	Drive in alarm status.	X	OFF	X	ON
	Local link in waiting status (tandem).	X	OFF	X	Flashing
	No Software in DSP.	ON	ON	ON	2 Flashing
	Kernel FPGA (KFPGA) Software storing fault	ON	ON	ON	3 Flashing
	No Software in Kernel FPGA and eventually no Software in DSP also.	ON	ON	ON	4 Flashing
	DPS Software incompatible with BOOT revision	ON	ON	ON	5 Flashing
	Heartbeat fault or heartbeat never started.	ON	X	OFF	X
Working conditions	Torque enabled (we could have torque enable request without torque enabled).	OFF	ON	X	OFF
	Speed enabled (we could have speed enable request without speed enabled).	OFF	X	ON	X
Software dowload status (1)	Flash memory erase before the download of the new Software.	OFF	OFF	Flashing	OFF
	Software download / update running.	OFF	Flashing	ON	OFF
	Software download / update ended.	OFF	Flashing	Flashing	OFF

LEGENDA.
(1) = Available status only if there is a Software in the drive (previous Sw version)
ON / OFF = Led status
X = Status ON or OFF negligible condition.
_ Flashing = N° of led flash.

5.8.3.1 Mono-Axis alarm possible causes/remedy

LEDs		Possible causes / remedy
	No Software in DSP	- Check the presence of the software. - Load the software.
	Kernel FPGA (KFPGA) Software storing fault	
	No Software in Kernel FPGA and eventually no Software in DSP also	
	DPS Software incompatible with BOOT revision	- Inform NUM Technical Service.

5.8.4 Bi-Axes alarm signalling status

LEDs		Led signalling status			
FAULT condition	Descriptions	**L1 (SLF)** 	**L2**	**L3**	**L4 (Fault)**
	Drive in alarm status for $Axis_n$	X	OFF	X	ON
	Drive in alarm status for $Axis_{n+1}$	X	X	OFF	ON
	Internal Local link in waiting status (tandem).	X	OFF	OFF	Flashing
	No Software in DSP.	ON	ON	ON	2 Flashing
	Kernel FPGA (KFPGA) Sw storing fault.	ON	ON	ON	3 Flashing
	No Software in Kernel FPGA and eventually no Sw in DSP also.	ON	ON	ON	4 Flashing
	DPS Software incompatible with BOOT revision.	ON	ON	ON	5 Flashing
	Heartbeat fault or heartbeat never started.	ON	OFF	OFF	X
Working conditions	Torque enabled for $Axis_n$ (we could have torque enable request without torque enabled).	OFF	Flashing	X	X(2)
	Torque enabled for $Axis_{n+1}$ (we could have torque enable request without torque enabled).	OFF	X	Flashing	X(2)
	Speed enabled with torque enabled for $Axis_n$.				
	(we could have speed enable request without speed enabled).	OFF	ON	X	X
Software dowload status (1)	Flash memory erase before the download of the new Software	OFF	OFF	Flashing	OFF
	Software download / update running	OFF	Flashing	ON	OFF
	Software download / update ended	OFF	Flashing	Flashing	OFF

LEGENDA.
(1) = Available status only if there is a Software in the drive (previous Software version)
(2) = If ON this alarm is related to the other Axis.
ON / OFF = Led status
X = Status ON, OFF or flashing condition.
_ Flashing = N° of led flash.

5.8.4.1 Bi-Axes alarm possible causes/remedy

LEDs		Possible causes / remedy
	No Software in DSP	- Check the presence of the software. - Load the software.
	Kernel FPGA (KFPGA) Software storing fault	
	No Software in Kernel FPGA and eventually no Software in DSP also	
	DPS Software incompatible with BOOT revision	- Inform NUM Technical Service.

5.8.5 Alarms configuration

A series of parameters configured according to specific needs make it possible to manage the drive at all times, even in the event of an alarm.

The drive alarms configuration are setted by means V32, V33, V34 parameter and are divided in two different categories:

Parameters	Alarm configuration
V32	NC = Motor stop not configurable (motor stop by inertia only) C = Motor stop configurable by inertia or by braking.
V33	NC = To reset the alarm, it is mandatory to turn OFF power and auxiliary supply (Not configurable) C = The alarm reset mode may be configured as indicated on the chapter 3.9.11, 3.10.11 “AlarmReset” or by WPM
V34	NC = Opening of contact DF1 / DF2 (1) for mains contactor command is not possible. C = Opening of contact DF1 / DF2 for mains contactor command is configurable.
(1) See MDLL3 power supply. NC : Not configurable C : Configurable	

See the general drive / alarms configuration table in the next page.
The A53 alarm “Self-test Hardware alarm” is not configurable alarm, therefore is mandatory turn OFF the system.

5.8.5.1 General drive alarms / securities configuration

Alarms description		Alarms / securities configuration		
Nº alarm		V32 (Braking config.)	V33 (Reset config.)	V34 (Relay config.)
2	BUS overvoltage	NC	NC	C
3	Motor overtemperature	C	C	C
4	Module overtemperature	C	C	C
5	Brake resistor or PWS overtemperature	C	C	C
6	Short circuit to earth	NC	C	C
7	Auxiliary power supply alarm	NC	C	C
8	Local link alarm	C	C	C
9	S1 sensor alarm	C	C	C
10	Timeout	NC	C	C
11	Overspeed	C	C	C
12	Drive overcurrent	NC	C	C
14	Flash checksum error	NC	NC	NC
15	Data memory checksum error	NC	NC	C
16	Incompatible parameters	NC	C	C
18	Self-calibration alarm	C	C	C
36	Boot revision incomplete	NC	NC	NC
37	Following error	C	C	C
38	S2 sensor alarm	C	C	C
49	I^2t alarm	NC	C	C
51	Check congruence position S1-S2 KO	C	C	C
52	Tandem function KO	C	C	C
53	Self-test Hardware	NC	NC	NC
54	Brake failure	NC	C	NC
55	Incompatible sensors	NC	NC	NC
56	V/f alarm	NC	C	NC
90	Trajectory generator	C	C	NC
91	CANopen communication	C	C	C

5.8.6 Alarms reset procedure

Alarms reset procedure	
From Controller (C alarms) see “AlarmReset” chapter 3.9.11, 3.10.11	- Check the type of alarm signalled by consulting the alarm tables - - Disable the drive. - Eliminate the cause if possible (see chapter 5.8.7). - Reset the alarm by means the Controller. If the cause of the alarm has been removed, the red fault led will go out, otherwise the previous alarm fault will reappear.
Hardware reset of the configurable alarm (C alarms)	If the Controller is not available, the alarm memorized can be reset as follow : - Disable the system. - Eliminate the cause if possible (see chapter 5.8.7). - Reset the alarm by means of CF1 / CF2 input on MDLL3 J1 connector.
Hardware reset of the not configurable alarm (NC alarms)	- Switch off the mains 3 phases and auxiliary power supply. - Eliminate the cause if possible (see chapter 5.8.7). - Switch on the mains 3 phases and auxiliary power supply.

5.8.7 Alarms signalling

All alarms extensions indicated and described in this chapter are referred to the drive software release till 3.2x and 4.2x version only.

Alarm display	MDLU3 alarm n°	Possible cause	Possible remedy
Overvoltage	**02**	DC BUS voltage higher than the allowed value	
		A - Wrong operation of internal or external braking resistor.	A - Check connection and functioning of the internal/external braking resistor.
		B - Mains voltage too high.	B - Check the line voltage.
		C - Peak of power braking not enough.	C - Check the dimensioning time.
Motor overtemperature (1)	**03**	The motor temperature is higher than the maximum allowed	
		A - The motor has worked in overcharge status for too long and the thermal switch has operated.	A - Check the working thermal cycle of the motor.
		B - The thermal switch wiring is not connected present or is incorrect.	B - Check the connector thermal switch wiring and the continuity of the thermal switch connections cables.
Module overtemperature	**04**	The temperature of the heat-sink power module is higher than the maximum allowed (about 90°C)	
		A -The internal temperature of the electric box or the environment temperature is too high (> 45°C) or the module heat-sink temperature is too high (>90°C).	A - Address to the machine manufacturer to check the correct thermal exchange.
		B - Incorrect system dimensioning.	B - Check the dimensioning of the drives and electrica cabinet.
		C -The force ventilation fan does not turn.	C - Inform NUM Technical Assistance.
Brake resistor or PWS overtemperature	**05**	The temperature of the braking resistor or of the heatsink's power supply module is higher than the maximum allowed. This alarm signal is issued by Power Supply MDLLx (flat cable level)	
		A - The working cycle is charactrized with very close accelerations/decelerations and the external braking resistor has reached a too hig > 400°C.	A - Use a less heavy working cycle by increasing the time between the accelerations or by increasing the braking resistor power.
		B - The power supply heatsink has reached a temperature > 90°C.	B - Check the correct thermal heat exchange inside the enclosure the electrical cabinet.
Short circuit to earth	**06**	Short circuit to the ground of the power chapter. This alarm signal is issued by the power supply (flat cable level)	
		DC BUS ground leakage.	Check the motor cable/ cables and drive ground leakage.
		Drive Power Supply.	Change Drive Power Supply (MDLLx) that could be damaged.
Auxiliary Power Supply alarm (1)	**07**	The auxiliary voltage is not available	
		Connection error on one or more drive internan auxiliary power supplies.	Inform NUM Technical Assistance.

Local link alarm (1)	**08**	This alarm is due to fault on the serial link between MDLU3 drives used to create generate the Tandem function.	
		This alarm is due to fault on the serial link between drives used for Tandem function.	
Alarm display	**MDLU3 alarm n°**	**Possible cause**	**Possible remedy**
S1 sensor alarm (1)	**09**	This alarms are related to the S1 sensor	
		Wrong connection of one or more sensors, sensor damage or fault sensor.	Check the sensor and check the sensor connections on the connector.
Timeout (1)	**10**	Time out expired	
		Inform NUM Technical Assistance.	
Overspeed (1)	**11**	The motor exceeded the maximum speed permitted for the motor or the motor did not stay at 0 speed during the initialization phase of the drive	
		A - The motor speed has exceeded the max.speed threshold foreseen with parameter V96, V99, V1096, V1099.	A - Check the present in the parameters managing the maximun speed and V96, V99, V1096, V1099 parameter value if compatible. i.e: Motor speed 3000 rpm, over speed threshold 2000 rpm.
		B - Wrong set up of the speed loop parameters.	B - Check the values present in the parameter managing the speed loop control.
		C - It has been requested to enable the drive while the motor is moving at speed above its flux-weakening starting speed (V417, V457,V497 reserved parameters).	C - Check enabling condition: wait for motor speed has reached a value below its flux-weakening starting speed (V417, V457, V496, V497) before enabling it.
Drive overcurrent	**12**	The current exceeded the maximum value allowable for the drive	
		A - Error on the power connection motor side.	A - Check motor power wiring U, V, W.
		B - Error in the sensor wiring connection used for feedback in the current loop management.	B - Check the wiring of the sensor used as feedback in the current loop management.
		C - Error in motor / drive association; used a wrong ST. (Technical specification).	C - Load the right ST.
		D - EPS function started with a wrong set up of the parameters. (param. V180, V181, V182, V183)	D - Make a new and right set up of the EPS function.
Bad flash checksum	**14**	Alarm due to SW checksum incompatibility in Flash memory (managed by Boot program)	
		Inform NUM Technical Service.	
Data memory checksum error	**15**	The checksum of the E2PROM memory of the drive is wrong. This alarm is issued during the initialization phase of the drive.	
		Inform NUM Technical Service.	
Parameters incompatibility (1)	**16**	Alarm for parameter incompatibility	
		A - Motor / drive association incompatibility.	A - Load the correct parameters configuration for the motor / drive association and verify the parameter value.
		B - Parameters not available. (i.e. parameter memory formatted)	B - Load the correct ST.

Self-calibration alarm (1)	**18**	Alarm related to the self-calibration phases of the EPS and EPSM functions	
		See alarms extension on chapter 5.8.8.	
		Inform NUM Technical Service.	
Alarm display	**MDLU3 alarm n°**	**Possible cause**	**Possible remedy**
Following error (1)	**37**	The motor exceeded the allowable maximum following error	
		- No good tuning of the control speed loop. - No good tuning of the position speed loop. - No good working of the sensor used for the feedback of the control speed loop. - No good working of the sensor used for the feedback of the control position loop. It is possible to exclude this alarm setting to 1 the bit 5 of the V30 parameter.	Check the right tuning of the system or inform NUM Technical Service.
S2 sensor alarm (1)	**38**	This alarms are related to the S2 sensor	
		Wrong connection on sensor S2 problem or sensor fault.	Check the sensor connection. See alarms extension on chapter 5.8.8.
I^2t alarm (2)	**49**	This alarm is issued whenever the I^2t condition has been reached and the drive is configured to handle such situation as an alarm instead of a warning (V30)	
		A - The working cycle of the machine-tool involves a phase where the drive current delivered to the motor is too high for a too long time. B - Some damage/defect has caused friction to increase, so requiring more current to be delivered to the motor in order to follow Controller acceleration commands. C - Some parameters on Controller require very high accelerations which can't be respected by the drive and motor which the machine-tool is equipped with.	A - Check the working cycle of the machine-tool. B - Check the machine-tool mechanics for any cause of increased friction. C - Check Controller accelerations parameter and, possibly, relax them to reduce required working speed.
		NOTE. It is always possible to eliminate such alarm by configuring parameter V30 of the drive in order to handle the I^2t condition as a simple warning. But be careful! In case of I^2t handled as a simple warning, the Controller is NEVER informed that such condition has occurred and may continue to requires so high working speed as to generate "following error" due to the fact that the drive has limited its torque current in order not to burn the motor and/or itself.	
Check congruence position S1-S2 KO	**51**	This alarm is related to the AP06 special application.The difference between the measures, in physical units, coming from the S1 and S2 sensors exceedes the value stored in V190 parameter and this condition stays active for a time longer than the value stored in V191 parameter.	
		A - Broken of the kinematical chain motor / load. B - Broken of one or both sensors. C - Wrong set up of the AP06 special application.	A - Check the kinematical chain motor / load. B - Check the connections of the sensors, their mechanical fit, check the fullfill sensors are good or not. C - Make a new tuning of the AP06 special application. (V190, V191 parameters)

Tandem function KO (1)	**52**	This alarm is related to the tandem function: AP03 special application	
		This alarm summarizes all the fault conditions emerging from mechanical problems and/or electronic problems in the Master and Slave drives.	
Self-test Hardware (1)	**53**	Alarm due to bad hardware functionality. This alarm is issued during the initialization phase of the drive.	
		This alarm is detected if any Hardware problems occurs. Inform NUM Technical Assistance.	
Alarm display	**MDLU3 alarm n°**	**Possible cause**	**Possible remedy**
Brake failure (NUMDrive C only)	**54**	This alarm is related to the Brake internal functionalities. See alarms extention on next pages.	
Incompatible sensor	**55**	This alarm is related to the incompatible sensor declaration (see parameters V348 and V349 declaration).	
V/f regulation	**56**	This alarm is related to the V/f regulation motor control.	
Trajectory generator (3)	**90**	Alarm due to the trajectory generator SW module.	
CANopen communication (3)	**91**	Alarm due to communication problems.	

(1) See “Alarms extension signalling” on next page.

(2) This alarm is issued whenever the I^2t condition has been reached and the drive is configured to handle such situation as an alarm instead of a warning (V30). It is always possible to eliminate such alarm by configuring parameter V30 of the drive in order to handle the I^2t condition as a simple.
WARNING. In case of I^2t handled as a simple warning, the Controller is NEVER informed that such condition has occurred and may continue to requires so high working speed as to generate «following error» due to the fact that the drive has limited its torque current in order not to burn the motor and/or itself.

(3) Only for MDLU3 CANopen MDLU3000ACCxN00 / MDLU3000BCBxN00 / MDLU3000BCCxN00 modules. Alarm due to the trajectory generator SW module.
Please refer to the next pages.

5.8.8 Alarms extension signalling

5.8.8.1 Extension for 02, 03, 07, 08 alarms

Alarm display	Alarm extension	
BUS overvoltage 02	**00**	Overvoltage from PWS signal. This alarm signal is issued by Power Supply MDLLx (flat cable level).
	01	Overvoltage bus voltage image. This alarm signal is issued directly by the drive starting from a own bus voltage image that it calculates by means of an analogical signal from PWS (flat cable level).
Motor overtemperature 03	**00**	Motor over temperature. The allowable max value is putted into the V270 parameter that contain the value, in Ohm, of the resistance of the thermal probe after that the alarm will go out. Check the technical data sheets of the thermal probe for to find the relation betwe-en ohmic resistance and motor temperature. This extension is applicable to both PTC and NTC thermal probes. The resistance of the thermal probe PTC is bigger then V270 value. The resistance of the thermal probe NTC is lower then V270 value.
	01	NTC thermal probe not wired. This alarm is issued when an NTC thermal probe is used and it is not wired or it is wrong wired. The resistance of the thermal probe NTC is bigger then 100kOhm.
Auxiliary Power Supply alarm 07	**00**	+/-15V auxiliary power supply missing. This alarm is issued whenever +/-15V power supply is missing. Such power supply is used to power the AD conversion circuitry of sensors and BUS voltage; furthermore it is used to supply power to the sensor itself.
	01	3.3V auxiliary power supply overvoltage This alarm is issued when 3.3V power supply exhibits an overvoltage condition. Such power supply is used to power the DSP, the FPGA and all computation circuitry on Kernel board. In case of 3.3V overvoltage condition is safer to turn-off the drive and subsitute it as soon as possible.
Local link alarm 08	**00**	The slave drive doesn't answer. This alarm is issued only by the master drive. If bits 1,2,3 in V170 are resetted =0, but in V169 drives are declared like master or slave, drives become independently but the Link is still active. Alarms of one drive are not transferred to the other one but the Link alarms are active. E.g.: If the Link is physically interrupted, alarm A08.0 issues from master and A08.42 from slave like Tandem function was active.
	42	Local communication link interrupted. The master drive started the communication but failed to continue. This alarm is issued only by the slave drive. If bits 1,2,3 in V170 are resetted =0, but in V169 drives are declared like master or slave, drives become independently but the Link is still active. Alarms of one drive are not transferred to the other one but the Link alarms are active. E.g.: If the Link is physically interrupted, alarm A08.0 issues from master and A08.42 from slave like Tandem function was active.
	51	No slave drive answered on the local communication link. This alarm is issued only by the Master drive.
	52	The master drive sends consecutive bad data packets. The master drive, after the start of the communication, sends consecutive data packets with errors inside. The timeout amount is related to the status of the drive torque enable: yes or no. This alarm is issued only by the slave drive.
	53	The master drive has never started the communication with the slave drive. This alarm is issued only by the slave drive.
	61	The drive configuration has changed from master to slave and viceversa without passing through power-up.
	63	Both the axes are declared as masters. This alarm is issued only by NUMDRIVE C Bi-Axes.
	64	Both the axes are declared as slaves. This alarm is issued only by NUMDRIVE C Bi-Axes.

5.8.8.2 Extension for 09 alarm Stegmann encoder

Alarm display	Alarm extension	
S1 Sensor alarm 09	**01**	Stegmann alarm: Analogue signal out of specification.
	02	Stegmann alarm: Wrong internal angle offset erroneous.
	03	Stegmann alarm: Data field partition destroyed.
	04	Stegmann alarm: Analogue limit not available.
	05	Stegmann alarm: Internal I2C bus not working.
	06	Stegmann alarm: Internal checksum error.
	07	Stegmann alarm: Encoder reset by program monitoring.
	08	Stegmann alarm: Counter overflow.
	09	Stegmann alarm: Parity error.
	10	Stegmann alarm: Wrong checksum of transmitted data.
	11	Stegmann alarm: Command unknown.
	12	Stegmann alarm: Number of transmitted data wrong.
	13	Stegmann alarm: Transmitted command not allowed.
	14	Stegmann alarm: Data can’t be written to selected datafield.
	15	Stegmann alarm: Wrong access code.
	16	Stegmann alarm: Size of data fiels can’t be changed.
	17	Stegmann alarm: Specified address is outside data field.
	18	Stegmann alarm: Access to non-existent data field.
	28	Stegmann alarm: Monitoring the magnitude of analogue signals..
	30	Stegmann alarm: Critical encoder temperature.
	31	Stegmann alarm: Speed too high, position acquisition impossible.
	32	Stegmann alarm: Position of single-turn not allowed.
	33	Stegmann alarm: Position error multi-turn, 1.
	34	Stegmann alarm: Position error multi-turn, 2.
	35	Stegmann alarm: Position error multi-turn, 3.
	40	Stegmann alarm: Error building absolute position (linear system).

5.8.8.3 Extension for 09 alarms EnDat encoder

Alarm display	Alarm extension	
S1 Sensor alarm 09	**00**	EnDat alarm: Absolute position error.
	01	EnDat alarm: Lighting.
	02	EnDat alarm: Signal amplitude.
	03	EnDat alarm: Position error.
	04	EnDat alarm: Supply voltage too high.
	05	EnDat alarm: Supply voltage too low.
	06	EnDat alarm: Current too high.
	07	EnDat alarm: Battery failure.

5.8.8.4 Alarm 09 common extensions from 200 to 261

Alarm display	Alarm extension	
S1 Sensor alarm 09	**200**	HOMING with coded references. The distance between two references is higher than the value in the V368 parameter.
	201	HOMING. The first zero reference position was detected in a wrong position (not multiple of 4).
	202	HOMING with coded references. The second zero reference position was detected in a wrong position (not multiple of 4).
	230	Sensor: repeated pulse counter UGA misalignments. (1)
	231	Sensor: excess of movement detected while waiting for UGA alignment. The axis has done an uncontrolled movement during the analogue & encoder position components alignement phase.
	232	Sensor: movement exceeded 30 electr. Degree while waiting for UGA alignment. This alarm can be issued only if the alarm ext.231 is excluded. The axis has done an uncontrolled movement during the analogue & encoder position components alignment phase bigger then the equivalent 30° electrical. In this case not only the position is wrong but there is a high risk of loss motor phase.
	238	It is issued by a wrong set up of MODE_SINCOS RS485 encoder.
	239	It is issued by a wrong set up Baud Rate RS485 encoder.
	240	It is issued by a wrong data field format Stegmann sensor: No data fields defined ! No data fileld necessary to manage an Hiperface NUM o SER sensor.
	241	It is issued by a wrong initialisation Timeout sensore position. No digital communication with the sensor. Possible special cause: Wrong power supply connection. Possible remedy: Check sensor connections on the connector. Verify the amplitude of sine and cosine signals using MP202 amd MP203, if they are both simultaneusly around zero could be a power supply problem. MP238 must be around 100% if power supply is good.
	242	It is issued by a wrong data field format for NUM Stegmann sensor.
	243	It is issued by a wrong data field format for SER Stegmann sensor.
	244	It's not possible to find a correct value for the resolver carrier gain.
	245	Resolver signals saturation while searching best carrier phase.
	246	The resolver cable requires the High-Voltage power supply enable connection.
	247	Timeout while searching best carrier phase for resolver.
	248	Encoder signals sine or cosine saturated. It is possible to exclude the alarm setting V348.10 = 1.
	249	Either the resolution bit number of the absolute position or the number of turns uploaded from encoder memory do not match with ones declared.
	250	Phase offset not present in the encoder.
	251	Checksum error in the EEPROM memory of the encoder.
	252	Motor poles pairs / Sensor poles pairs can't match for Permanent Magnet Synchronous Motor+resolver.
	253	Stegmann encoder type unknown. This alarm extension is issued when a Stegmann encoder has the sensor code ID declared in the V344 parameter not coherent.
	254	Sensor no connected or sine or cosine signal too low. Check the sensor wiring and if it will be ok, set the right value in the V314 parameter. (1)
(1) These alarms (on S1) may be enable / disable by V314.5 bits parameter. These alarms on S2 may be enable / disable by V315.5 bits parameter.		
Alarm display	**Alarm extension**	

S1 Sensor alarm 09	**255**	Fault in the encoder communication. The drive cannot communicate with the encoder through the digital interface during the initialization phase. It is possible that the encoder is badly connected or the wiring is wrong. Possible special cause: Wrong power supply connection. Possible remedy: Check sensor connections on the connector. Verify the amplitude of sine and cosine signals using MP202 amd MP203, if they are both simultaneusly around zero could be a power supply problem. MP238 must be around 100% if power supply is good.
	256	The absolute position displacement in the middle of the travel is not allowed for EnDAt linear encoder. Check the bit 18 in the V348 parameter.
	257	Sensor error signal (connector pin 21 state = low). Check sensor connections on the connector.
	258	Fast serial sensor missing data. The drive does not communicate correctly with the sensor through the digital interface. It is possible that the sensor is badly connected or the wiring is wrong.
	259	EnDat 2.2 sensor CRC error. The drive does not communicate correctly with the sensor through the digital interface. It is possible that the encoder or the wiring electromagnetic shield is badly connected.
	260	SSI sensor not connected. Check sensor connections on the connector. It is possible that the encoder is not correctly connected.
	261	High-speed communication (5MHz) with EnDat2.2 not possible. The drive does not communicate correctly with the sensor through the digital interface.

5.8.8.5 Extension for alarms from 10 to 15

Alarm display	Alarm extension	
Timeout 10	**01**	Timeout on Braking (config. V30.1 / V32), the speed motor has not been gone down speed V41 / V42 within the time V43.
	04	PWM timing error: missing PWM refresh.
Overspeed 11	**00**	Motor overspeed.
	01	The motor didn't stay at 0 speed during the initialization phase of the drive: a drive re-initialization was tried while the motor was moving; wait for motor stop before retry a drive reset/re-initialization.
	02	Speed > flux-weakening starting speed : the drive was requested to enable while the motor was running at speed higher than the flux-weakening starting speed (V417, V457, V496, V497). Other possibilities: for synchronous motor V42 > V497, for linear motor V41 > V496, for asynchronous motor V42 > V417 (low range) or V457 (high range).
Flash checksum error 14	**00**	Error in the SW DSP checksum (SW missing).
	01	Error in the KFPGA programming (possible HW error).
	02	Error in the KFPGA checksum (SW missing).
	03	DSP software version incompatible.
Data Memory checksum error 15	**00**	Error in the parameters table checksum
	01	Error in the alarm table checksum
	02	Error in the mechanical ratio checksum (it is correlated to the AP02 special application). This alarm is issued if the Checksum of AP02 informations stored in Flash memory is wrong. The first time the function is enabled on a new drive this alarm is issued (not resettable alarm). It needs to switch off and restart the drive.

5.8.8.6 Extension for alarm 16

Alarm display	Alarm extension	
Parameters incompatible 16	**00**	Speed/Position regulator parameters exchanges set 0,1 not correct.
	04	HW / SW incompatibility.
	06	Inom dereating not correct. The alarm issues if the Inom value defined into V108 /V115 doesn't take into account the dereating owing to the PWM frequency value (5khz -> 10khz).
	07	Oversampling configuration modified: it requires OFF->ON of the drive to be effective.
	08	Sensor - FPGAsensors not compatible: it requires OFF->ON of the drive. The alarm issues when Resolver is declared or undeclared modifying V348. This modifie requires always the OFF-> ON of the drive.
	20	This alarm can issue (only for NUMDRIVE C) for two reasons: 1) Wrong configuration coherence of V20 (axis#0) with V252. If bit 0 of V20 (axis#0) is set (Sensor connectors X3, X4 plugged-in) V252 has to be equal to "G": that means NUMDRIVE C Bi-Axes HP. If bit 0 of V20 (axis#0) is reset (Sensor connectors X3, X4 absent) V252 has to be equal to "H": that means NUMDRIVE C Bi-Axes BP. 2) Wrong configuration coherence of V20 (axis#0) with V350,V351 or V352. If bit 0 of V20 (axis#0) is reset (Sensor connectors X3, X4 absent V350, V351, V352 have to be equal to 0: that means position, speed or current sensor loop configurated on the connectors X1, X2.
	35	SW functions not available. The alarm issues if e.g.: AP01 is enabled in V160 but V35.0=0, AP02 is enabled in V160 but V35.1=0, Tandem function AP03 is enabled in V170 but V35.2=0, AP06 is enabled in V160 but V35.3=0, AP08 is enabled in V160 but V35.5=0, AP10 is enabled in V160 but V35.7=0, AP12 is enabled in V160 but V35.9=0, AP13 is enabled in V160 but V35.10=0. With V35.11=0 it is possible to manage two sensors only if all loops are linked to only one sensor, V350=V351=V352=0 or V350=V351=V352=1. If the motor is a linear motor and V35.12=0 the alarm issues. If the drive is NUMDRIVE C Bi-Axes BP the only one SW function available is AP02.
	160	Incompatibility between special application and others drive parameters. E.g. : AP01: bit V348/V349.6 of the sensor that doesn't close the position loop has to set =1 otherwise the alarm issues. AP02:this alarm can issue for many reasons: 1) The motor is a linear motor. 2) The sensor selected for AP02 is not present. 3) The sensor selected for AP02 is not an absolute sensor V348/V349 Bit3=Bit4=Bit5=0. 4) The sensor selected for AP02 is coded reference. AP06: in parameters V348/V349 only one sensor is declared (AP06 needs two sensors). AP08: this alarm can issue for many reasons: 1) It is not possible activates AP08 if all loops are closed by the same sensor. 2) V160.7=1. It is not possible to activate contemporaneously AP08 and AP10. AP12: this alarm can issue for many reasons: 1) It is not possible activates AP12 if all loops are closed by the same sensor. 2) The axis is Slave in Tandem function V169.1=1 e V170.1=1. If the axis is Slave V169.1=1 but Tandem function is not active V170.1=0 the alarm doesn't issue. AP13: this alarm can issue for many reasons: 1) AP12 is not active, V160.9=0 2) Notch filter is not active, V500.5=0
Alarm display	**Alarm extension**	

Parameters incompatible 16	**170**	The torque duplication is no longer available but older SETTools can try to configure it e.g.: if bit V170.1=0 and bit V170.2=1 (Torque duplication function declaration not implemented in MDLU3) the alarm A16.170 issues.
	220	V220.7 = 1 with V271 = 100. V220.7 = 1 only with V271 = 50
	251	ST Parameters incompatibility
	252	Drive type not correct
	253	Drive size not correct
	255	Motor type not supported (only NUMDRIVE C Bi-Axes BP version)
	257	Motor poles pairs number not correct (only NUMDRIVE C Bi-Axes BP version)
	266	CANopen different from the value sent by the master in 60C2h.1
	267	ST Peak current not correct.
	271	FPGA re-synchronisation after sampling time current loop change.
	272	Motor thermal sensor configuration not allowed (only NUMDRIVE C Bi-Axes BP version). This alarm is issued if the thermal sensor is wiring on S2 (see V272.1)
	305	AP02 requires the setting of V305 “Measure module for rot. axis/spindle abs[0]”. V305 has not to be equal to 0.
	307	S1 overflow error on multi-divi.
	310	S1 overflow error on multi-divi + offset
	312	S2 overflow error on multi-divi
	313	S2 overflow error on multi-divi + offset
	317	V317 power supply > 5.5V but S1 sensor not available for high power supply (otherwise Stegmann sensor not connected). The jumper between pin 5 and 6 of S1 that enables the sensor power supply > 5.5V is not present.
	318	V318 power supply > 5.5V but S2 sensor not available for high power supply (otherwise Stegmann sensor not connected). The jumper between pin 5 and 6 of S2 that enables the sensor power supply > 5.5V is not present.
	348	S1 configuration error. This alarm can issue for many reasons: 1) S1 is configurated like current loop sensor and V348.9=1 2) Hall sensor on S1 is not allowed 3) Linear sensor V348.8=1 is declared like corrent loop sensor of a rotary motor. 4) Sensorless encoders declared on Linear and Synchronous motors 5) Sensorless on current AND speed loop ONLY (...both !!!) 6) Sinusoidal signal compensation (CSS) on Sensorless encoders.
	349	S2 configuration error. This alarm can issue for many reasons: 1) S2 is configurated like current loop sensor and V349.9=1. 2) Resolver-Sensor on S2 not allowed. 3) Linear sensor V349.8=1 is declared like corrent loop sensor of a rotary motor. 4) Sensorless encoders declared on Linear and Synchronous motors. 5) Sensorless on current AND speed loop ONLY (...both !!!). 6) Sinusoidal signal compensation (CSS) on Sensorless encoders.
	350	P-loop not allow with non-configured sensor or Hall sensor.
	351	V-loop not allow with non-configured sensor or Hall sensor.
	352	C-loop not allow with non-configured sensor.
	355	The parameters V355=0 and V356≠1.
	1035	AP02 requires the setting of V1305 “Measure module for rot. axis/spindle abs[1]” . V1305 has not to be equal to 0.
	1307	Overflow error on multi-divi.
Alarm display	**Alarm extension**	

Parameters incompatible 16	**1310**	Overflow error on multi-divi + offset.
	1312	Overflow error on multi-divi.
	1313	Overflow error on multi-divi + offset.
	#parameter: #parameter identifies the missing parameter number in flash table	

5.8.8.7 Extension for alarm 18, 37, 38, 49

Alarm display	Alarm extension	
Self-calibration 18	**02**	Error during EPS self-phasing phase: The EPS procedure was not able to find the electrical phase of the motor. E.g. the electric current used during EPS function is too low. It is necessary to tune well the V180, V182, V183 parameters.
	03	Incoherency between the acceleration sign and torque current sign. This alarm condition is monitored from the start of the EPS procedure till 1 second after the HOMING end, this motoring is performed for security motivation. E.g. acceleration versus not coerent with current versus owing to phase offset error. If this alarm is issued it is necessary a new tuning of the EPS procedure. To enable this new tuning it is necessary to disable the monitoring, setting V273.3 = 1.
	04	It was requested a POM after an EPS procedure V273.0 = 0 with the parameter V269 = 0 . It is necessary input the correct value in the V269 parameter as described in the Procedure for the EPS / EPSM functions.
	05	Timeout during the EPSM function. This alarm is issued about 6 seconds after the start of the EPSM command, in case one or more conditions needed for to enable the EPSM command are not satisfied. The conditions to satisfy for enable the EPSM are: - Drive in Run or Alarm status - DC BUS energized: voltage present. - Drive with torque and speed disabled. - Safety relays enabled.
	06	Timeout during the EPSM function Timeout on EPSM state machine: background NOT running.
	07	Timeout during the EPSM function Timeout on EPSM offset writing on encoder.
	08	Timeout during the EPS function Conditions for EPS not satisfied.
Following error alarm 37	**00**	CNC parameter P23 threshold exceeded.
	01	Speed dynamic check threshold exceeded. See values parameters V45, V1045 for linear motor set [0] and set [1], and V46, V1046 for sync/async motor set [0] and set [1].
	02	Return from Controller control suspend procedure. This alarm is always issued (for safety reasons) when WPM leaves again the drive control to the Controller.
	For MDLU3 CANopen MDLU3000ACCxN00 / MDLU3000BCBxN00 / MDLU3000BCxN00 the text description is: CAN controller COBid 6065h threshold exceeded	
S2 sensor alarm 38	Please refer to S1 sensor alarm 09 extensions description.	
I^2t (I2T) alarm 49	**00**	Threshold exceeded
	108	Wrong value for V108 rated motor current l.r.
	115	Wrong value for V115 rated motor current h.r.
	128	Wrong value for V128 stall motor current l.r.
	129	Wrong value for V129 stall motor current h.r.

5.8.8.8 Extension for alarm 52

Alarm display	Alarm extension	
Tandem function KO 52	**00**	The slave drive is in fault status.
	41	Torque equalizer saturation timeout. The timeout value is stored in the V176 parameter. The alarm has been created with the scope to detect electrical and/or mechanical problems of the anti-backlash system: this means that the two motors, mechanically coupled to realize the anti-backlash function, don't supply the same torque to move the machine. Check the kinematical chain, the sensors fitting, the mechanical joints, etc. This alarm is issued only by the master drive.
	42	AB balancer timeout: Partner not enabled due to unknown reasons. E.g.: master enables current, flux and speed loops, then it waits 500 msec the slave communication of current, flux and speed loops enabled. If the answer is o.k., the master activates torque balancer and position loop: axis is completely operative. If the answer is not o.k. owing to e.g. 1) Slave SAM board is not in status 6 and the slave hasn't completed the own loops enable phase 2) Slave Safety Light is open and then the slave hasn't completed the own loops enable phase, the alarm issues.
	50	The slave is in Halt state while the master is torque enabled. The master drive goes immediately in the alarm status if the system is torque enabled and the slave drive has been put in Halt status using Controller or WPM. To go out from this alarm condition, it is imperative to put in the same status all the drives involved in the Tandem function, master and all the slaves: 1. Put in alarm status the master drive and all the slave drives 2. Send an Alarm Reset command, at same time, to the master and to all the slaves. This alarm is issued only by the master drive.
	60	The master is in Halt state while the slave is torque enabled. The Slave drive will go immediately in alarm status if the system is torque enabled and the Master has been put in Halt status using Controller or WPM. To go out from this alarm condition, it is imperative to put in the same state all the drives involved in the Tandem function, master and all the slaves: 1. Put in alarm status the master drive and all the slave drives 2. Send an Alarm Reset command, at same time, to the master and to all the slaves. This alarm is issued only by the slave drive.
	62	This alarm has been created only in the slave drive to show that master drive is, for any cause, in the alarm status.

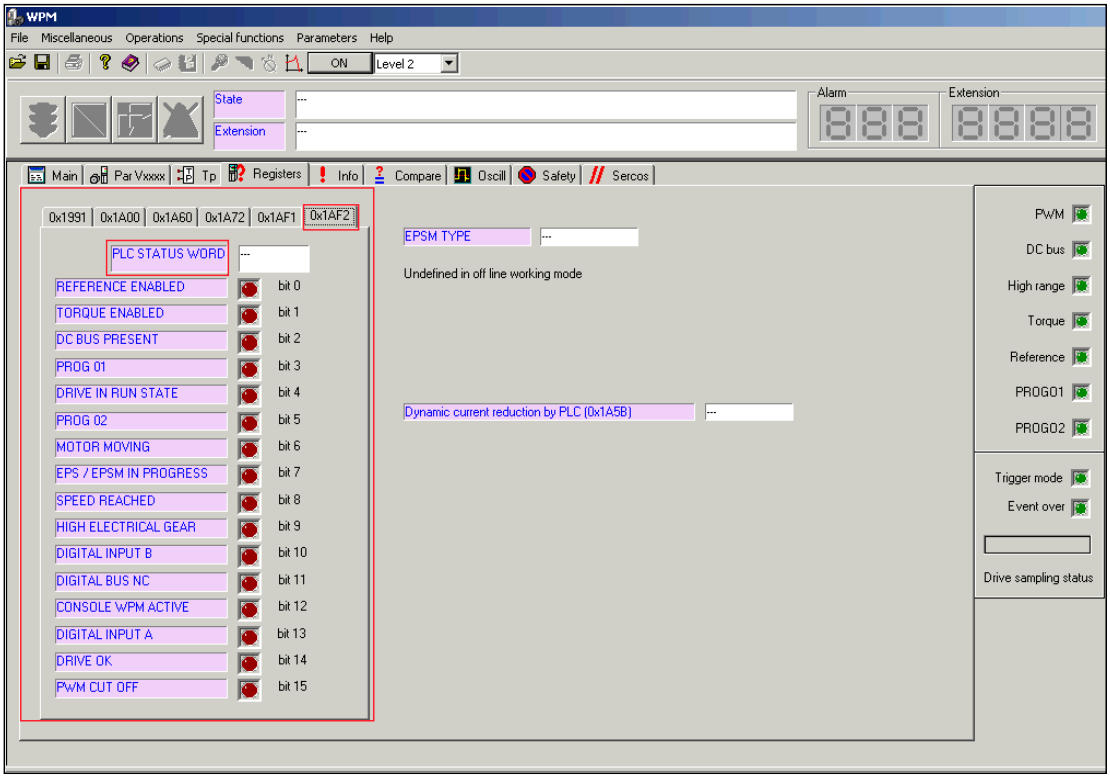
5.8.8.9 Extension for alarm 53

Alarm display	Alarm extension	
Self-test Hardware 53	**00**	Current sensing A/D converter not properly initialized.
	01	An error has occurred in the FPGAs reset procedure.
	02	Checksum error on control FPGA program in flash memory.
	03	The DONE pin hasn't gone high after the configuration of the control FPGA .
	04	Checksum error of the FPGA sensor board 1 program in flash memory.
	05	Checksum error of the FPGA sensor board 2 program in flash memory (only for NUMDRIVE C Bi-Axes).
	06	The DONE pin hasn't gone high after the configuration of FPGA sensor board 1 or 2.
	07	The control board PLL isn't working.
	08	The sensor board 1 PLL isn't working.
	09	The sensor board 2 PLL isn't working (only for NUMDRIVE C Bi-Axes).
	10	The control board address and data buses aren't working properly.
	11	The sensor board 1 address and data buses aren't working properly.
	12	The sensor board 2 address and data buses aren't working properly (only for NUMDRIVE C Bi-Axes).
	13	The control board programming code is missing.
	14	The sensor board 1 programming code is missing.
	15	The sensor board 2 programming code is missing (only for NUMDRIVE C Bi-Axes).
	16	PWR EEprom I2C not detected
	17	PWR EEprom I2C: Checksum error
	18	PWR EEprom I2C: Missing Imax data
	19	PWR EEprom I2C: Missing Inom data
	20	Wrong sensor connectors X3, X4 configuration (see V20 axis#0 only) (only for NUMDRIVE C Bi-Axes).
	21	Sensor connectors X3, X4 configuration modified (Changements not allowed) (see V20 axis#0 only): this alarm disappears after RESET and POWER UP (only for NUMDRIVE C Bi-Axes).
	22	Wrong Power board: Power board not coherent with drive type (only for NUMDRIVE C).
	23	Reading/writting information in Flash memory failed. This alarm is issued if the Flash memory is not accessible for more then 2 sec. It needs to switch off and restart.
	24	Drives addresses are equal.
	25	Tandem link deactivation needs power-up.
	26	SAM board presence not coherent with drive type.
	27	Hardware incompatiblity between control and power boards (only for NUMDRIVE C Bi-Axes).
	28	NUM-STO safety function not allowed for hardware incompatibility.
	29	Wrong dip switches selection for axis address. All dip setting OFF.

5.8.8.10 Extension for alarm 54, 55, 56, 90, 91

Alarm display	Alarm extension	
Brake failure A54	**00**	24 Volt for brake not connected or open fuse.
	01	Brake not connected or Overcurrent or Short Circuit on the output.
Incompatible sensors 55	**00**	TTL sensor on S1 and S2 not allowed.
	01	Sensorless on S1 and S2 not allowed.
	02	Sensor error signal test on both sensors not available.
	03	Fast Serial sensor on S1 and S2 not allowed.
	04	Fast Serial sensor with oversampling not allowed.
	05	Fast serial sensor - Endat sensor on S1, S2 not allowed.
	06	TTL sensor on both connectors X1, X2 (X3, X4) not allowed.
	07	Sensor error signal test on both connectors X1, X2 (X3, X4) not available.
	08	Fast Serial sensor on both connectors X1, X2 (X3, X4) not allowed.
	09	Fast serial sensor - Endat sensor on connectors X1, X2 (X3, X4) not allowed.
	10	SSI sensor on S1 and S2 not allowed.
V/f regulation alarm 56	**00**	Config: motor type not allowed (V255).
	01	Config: electrical gear switch not allowed (V256).
	02	Config: set parameters 1 not allowed (V260, V261)
	03	Config: I^2t must be configured as alarm (V30.6).
	04	Config: no vertical payloads allowed (V109).
	05	Config: special applications not allowed (V160.0/3/5/7/9/10).
	06	Config: anti-backlash not allowed (V169, V170).
	07	Config: acceleration deceleration ramps mandatory (V30.14/15).
	08	Config: encoder type for current loop not allowed.
	09	Config: encoder type for speed loop not allowed.
	10	Config: encoder type for position loop not allowed.
	11	Math err: wrong [V/Hz] slope: check V601, V602, V603.
	12	Math err: wrong gain of current limiter: check V91, V600, V605.
	For MDLU3 CANopen MDLU3000ACCxN00 / MDLU3000BCBxN00 / MDLU3000BCCxN00 the text description is: Math err: wrong gain of current limiter: check COBs 6083h 6084h, V600, V605.	
Trajectory generator 90	**00**	Unexpected HW limit switch encountered during movement.
	01	Reference outside limits.
	02	Homing method not supported.
	03	Operation mode not supported.
	04	Unexpected HW Positive limit switch encountered during movement.
	05	Unexpected HW Negative limit switch encountered during movement.
	06	Change to operation mode -1 is not allowed.
	07	Change of operation mode NOT allowed because previous movement is not completed.
	08	Operation mode NOT allowed for SENSORLESS motor.
CANopen Communication 91	**00**	HeartBeat lost.
	01	NodeGuarding lost.
	02	Slave address not allowed.

5.8.9 Meaning of the Word 0x1AF2 by WPM (PLC Status word)



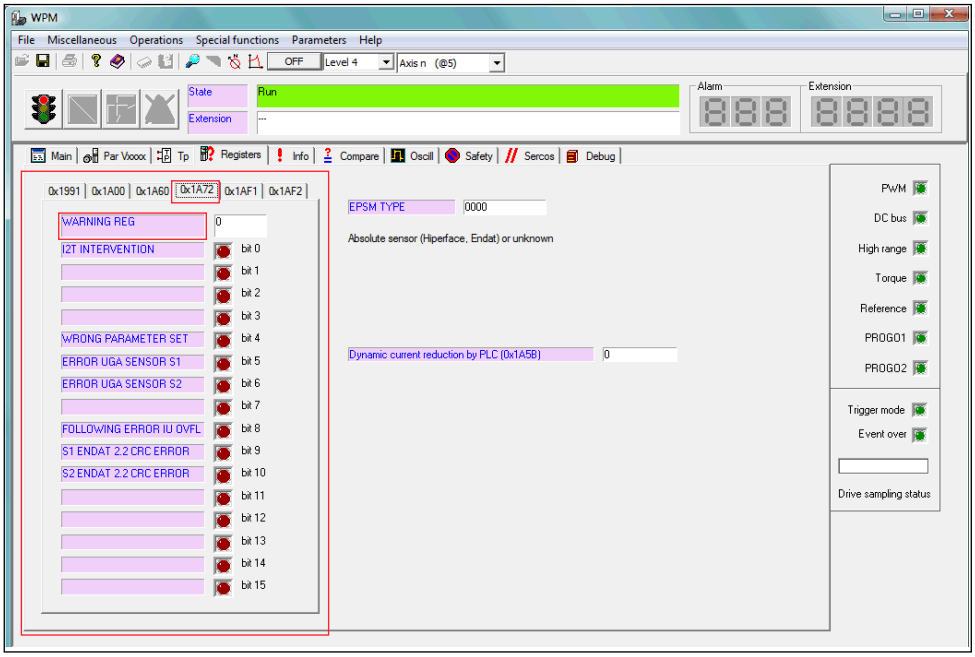
Bit*	Meaning	Bit value	Causes	Status
0	Reference Enabled	0: Reference (speed) not enabled	- Reference request = 0 - Controller fault"	- Led "SPEED READY" OFF - Drive not able to turn the motor"
		1: Reference (speed) enabled	- Reference request = 1 - No Controller fault"	- Led "SPEED READY" ON - Drive is able to turn the motor if torque enabled.
1	Torque Enabled	0: Drive not enabled	- Torque request = 0 - DC-BUS OFF - Security relais opened - Halt Status - Alarm"	- Led "TORQUE READY" OFF - No motor current - Drive not able to move motor
		1: Drive enabled	- Torque request = 1 - DC-BUS ON - Security relais closed - Run Status - No Alarm"	- Led "POWER ON" ON - Led "FAULT" OFF - Led "TORQUE READY" ON - Motor current, if motor connected - Drive is able to move motor - Drive is able to follow ref, if ref. enable will be given.
2	DC BUS Present	0: Bus voltage not present	- Power Supply not connected. to the mains. - Line switch opened	- Led "TORQUE READY" OFF - Drive disabled"
		1: Bus voltage present	- Line switch closed	
3	PROGO1	0: Output low		- S7 connector pin 2 = 0V (1)
		1: Output high		- S7 connector pin 2 = 24V (1)
4	Drive in Run state	0: Drive stopped	- Halt Request = 1 - Alarm	- Drive is able to receive commands for parameter handling. - Drive is not able to move motor.
		1: Drive start	- Halt Request = 0	- Drive is able to move motor

5	PROGO2	0: Output low		- S7 connector pin 3 = 0V (2)
		1: Output high		- S7 connector pin 3 = 24V (2)
6	Motor Moving	0: Motor stopped	- Motor stopped or slowly moving, speed is less then +/- V41 or V42	
		1: Motor running	- Motor speed is higher then +/- V41 or V42	
7	EPS /EPSM in progress	0: Self-calibration completed	- Phasing Request =0	- Drive normal status
		1: Self-calibration in progress	- Phasing Request =1	- Procedure used to phase synchronous motor sensor is in progress
8	Speed reached	0: Speed not reached		- The absolute value of (ref. speed - motor speed) >V40. See V40 use.
		1: Speed reached		- The absolute value of (ref. speed - motor speed) <=V40. See V40 use.
9	High Electrical Gear	0: Low speed range	- Spindle Electrical Configuration = 0	- Drive is using low range parameter.
		1: High speed range	- Spindle Electrical Configuration = 1	- Drive is using high range parameter.
10	Digital Input B	0: OFF	- S7 connector pin 5 = 0V	
		1: ON	- S7 connector pin 5 = 24V	
11	SBCE OK	0: Error	- Link cable not connected or wrong cabled - Controller fault - Link fault.	- Led “SERIAL FAULT” ON - Drive not synchronised with Controller sampling - SBCE status KO. - Not able to follow ref. - Not able to receive commands or to send status.
		1: OK		- Drive synchronised with Controller sampling. - Drive able to follow ref. correctly. - SBCE status OK.
12	Consol WPM Active	0: OFF	- WPM not connected or not in console mode.	- User is able to move the motor using WPM.
		1: ON	- WPM connected or in console mode	- Drive is able to move motor
13	Digital Input A	0: Output low		- S7 connector pin 3 = 0V (2)
		1: Output high		- S7 connector pin 3 = 24V (2)
14	Drive OK	0: Drive error	- Alarm condition.	- LED “POWER ON” ON - LED “FAULT” ON - LED “TORQUE READY”OFF - No motor current - Drive not able to move motor - Drive not able to follow ref., - HALT status
		1: No error	- No alarm.	- LED “POWER ON” ON - LED “FAULT” OFF - LED “TORQUE READY”OFF - Drive able to move motor - Drive able to follow ref.

15	PWM Cut Off	0: OK		
		With Stop Function enabled V30 bit 9 = 1 1: PWM cutted by Stop Function intervention"	Stop Function intervention: S8 pin 1 without 24Vdc S8 pin 2 without 0Vdc S8 pin 3-4 contact opened S8 pin 5-6 contact opened	- Led "TORQUE READY" OFF - No motor current - Drive not able to move motor - Drive not able to follow ref."
		With SAM 1: PWM cutted by SAM	SAM before start	- Led "TORQUE READY" OFF - No motor current - Drive not able to move motor - Drive not able to follow ref."
		With SAM 1: PWM cutted by SAM"	SAM intervention	- Led "FAULT" ON - Led "TORQUE READY" OFF - No motor current - Drive not able to move motor - Drive not able to follow ref. - HALT status"

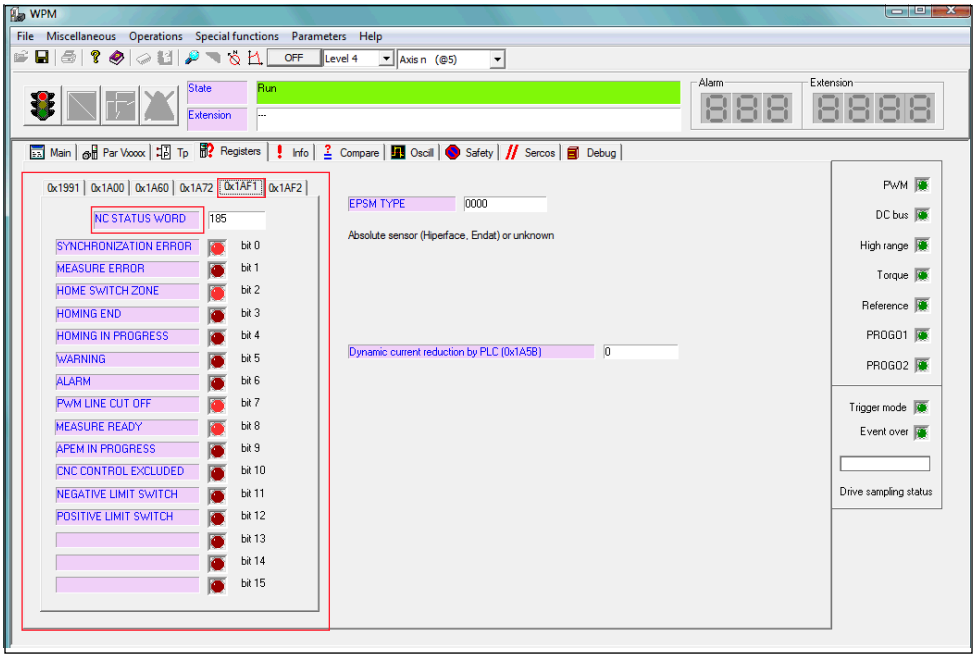
(1) On Bi-Axes drive the physical ouput depends from V22.0 too.
(2) On Bi-Axes drive the physical ouput depends from V22.1 too.
* Bit Status Word

5.8.10 Meaning of the Word 0x1A72 by WPM (Warning register)



Bit	Meaning
0	I^2 t INTERVENTION
1 2 3	Nor used bits
4	WRONG PARAMETER SET
5	ERROR UGA SENSOR 1
6	ERROR UGA SENSOR 2
7	Not used bit.
8	FOLLOWING ERROR IU OVERFLOW
9	S1 ENDAT 2.2 CRC ERROR
10	S2 ENDAT 2.2 CRC ERROR
11 12 13 14 15	Not used bits.

5.8.11 Meaning of the Word 0x1AF1 by WPM (NC Status word)



Bit	Meaning
0	SYNCRONIZATION ERROR
1	MEASURE ERROR
2	HOME SWITCH ZONE
3	HOMING END
4	HOMING IN PROGRESS
5	WARNING
6	ALARM
7	PWM LINE CUT OFF
8	MEASURE READY
9	APEM IN PROGRESS
10	CNC CONTROL EXCLUDED
11	NEGATIVE LIMIT SWITCH
12	POSITIVE LIMIT SWITCH

Page deliberately empty.

Page deliberately empty.

Published by NUM Spa

Via F. Somma 62
20012 Cuggiono (MI) (Italy)
Tel. : +39 02 979 691
website: www.num.com

Printed February 2017

CNC HighEnd Applications