NUM 1060

USE OF THE UNI-TE PROTOCOL

en-938914/0

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.

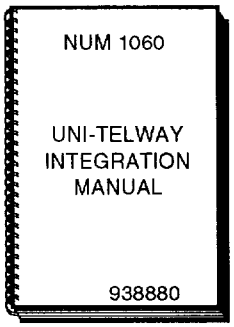
The programming examples described in this manual are intended for guidance only. They must be specially adapted before they can be used in programs with an industrial application, according to the automated system used and the safety levels required.

Table of Contents

NUM 1060 Network Documentation Structure

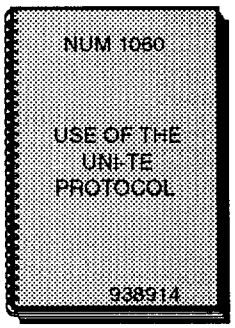
Integration Document

This document is designed for integration of NUM 1060 numerical controls in a network.

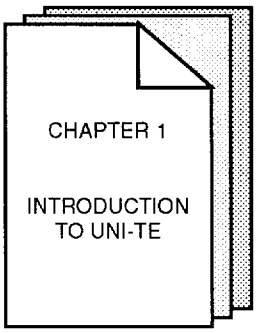


Operator Manual

This document explains network transfers with a numerical control.

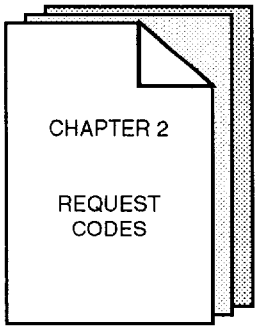


Use of the UNI-TE Protocol

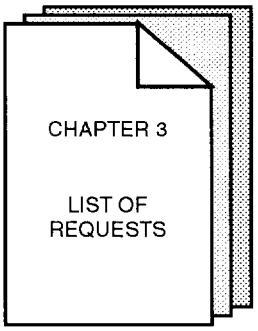


Operation of UNI-TE protocol.

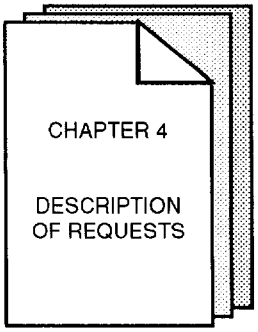
Particularities related to the use of NUM numerical controls.



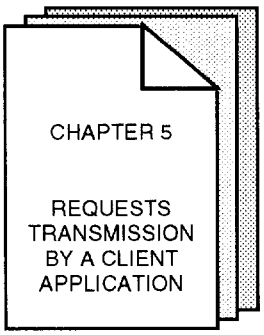
Description of the request structure:
- Request format,
- Report format..



Requests listed by type.



Detailed description of the requests and their use.



Review of sending of requests by a NUM client application.

Agencies

The list of NUM agencies is given at the end of the volume.

Questionnaire

To help us improve the quality of our documentation, we kindly request you to return the questionnaire at the end of the volume.

1 Introduction to UNI-TE

1.1 General

The application layer provides a unique communication interface for all the equipment from Telemecanique, NUM and other vendors that complies with the UNI-TE protocol.

The application layer provides standard services to which are added services specific to various units such as numerical controls.

The services are mainly implemented by a question and answer mechanism called «request/report».

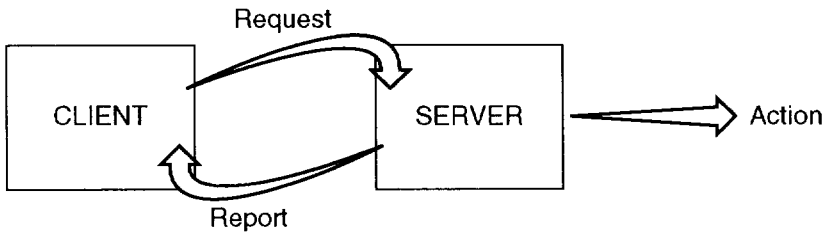
Only the requests specific to NUM 1060 numerical controls are detailed herein.

1.2 Operating Mode

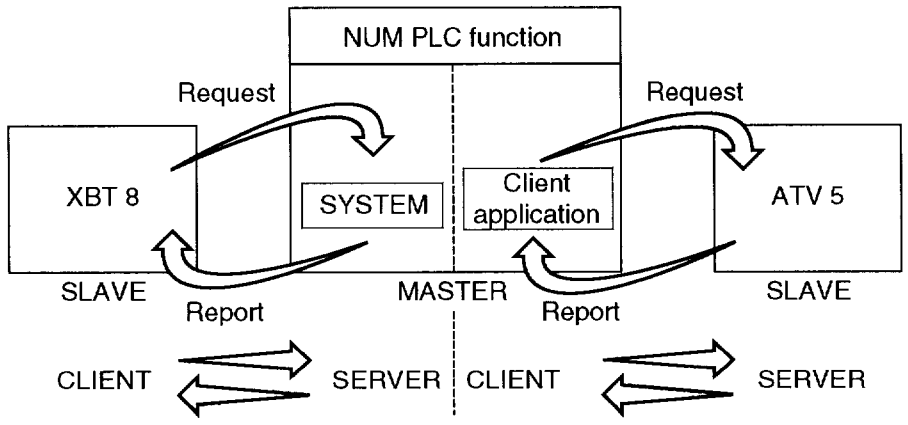
A unit supporting the UNI-TE protocol can have client and/or server status.

The client unit takes the initiative of the communication. It asks a question (read), sends data (write) or a command (RUN, STOP). The client is sometimes called «requester».

The server unit performs the service requested by the client and sends a report after execution.



Certain units can be both client and server. For instance, a PLC is a server for the system tasks (programming, settings, diagnostic) and may be a client for the NETO and NETI functions of the user program (sending of commands, read of states) for another unit.

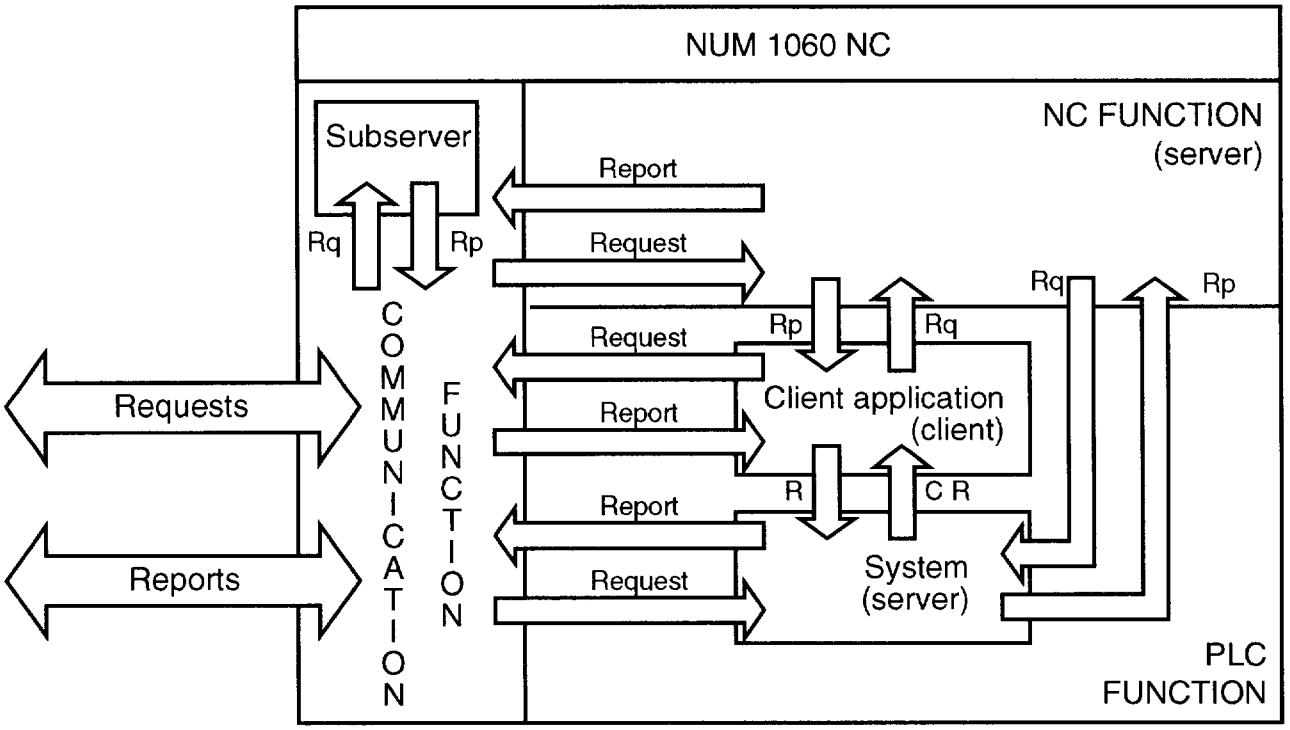


REMARK *The client and server states are independent of the master and slave modes.*

1.3 Special features of NUM 1060 Numerical Controls

NUM 1060 numerical controls include two entities that are capable of dialoguing with other units using the UNI-TE protocol:

- The NC function which manages path following in particular (execution of part programmes, management of tool and part offsets, axis and spindle measurement and control),
- The PLC function that manages the automatic controls of the numerical control.



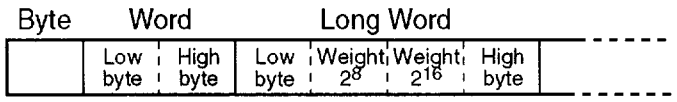
Both the NC function and PLC function (system) can act as server (see Chapter 3 for the requests sent to the NC).

The PLC function (client application) includes the client function (see Chapter 5 for the request sending procedure).

2 Request Codes

Requests are a sequence of bytes, words (2 bytes) and long words (4 bytes).

The bytes of the words and long words are sent beginning with the low byte:



2.1 Standard Request Format

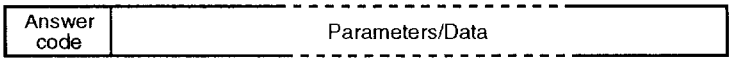
A request includes:
- the request code (one byte)
- the sender category code (one byte)
- the parameters and/or data (bytes, words or long words: maximum 126 bytes).

2.2 Standard Report Format

A report includes:
- the answer code (one byte)
- the parameters and/or data (bytes, words or long words: maximum 127 bytes).

If the report is positive, the answer code depends on the request.

For a negative report, the answer code is H'FD'.



3 List of Requests

3.1 Access to Data

These requests give access to the specific objects of the numerical control:

Requests	Request code	Answer code	Description	See
Read-Object	54 (H'36')	102 (H'66')	Read objects (bits, words)	4.1.1
Write-Object	55 (H'37')	103 (H'67')	Write objects (bits, words)	4.1.2

3.2 Unsolicited Data

This is the only service for which no report is returned. It is actually the answer to an implicit question.

This service allows a byte string to be transferred between application programmes without a prior request.

The use of these data is up to the receiver which must implicitly be on wait for this answer:

Requests	Request code	Answer code	Description	See
Unsolicited-Data	252 (H'FC')	—	Send data without first receiving a request	4.2

3.3 General Purpose Requests

3.3.1 Requests Addressed to the System

These requests are used for diagnostic and setting up. For the numerical control, they include:

Requests	Request code	Answer code	Description	See
Identification	15 (H'0F')	63 (H'3F')	Unit identification: product type, version and part number	4.3
Status	49 (H'31')	97 (H'61')	Information on the status of the unit	4.4

3.3.2 Requests Concerning the Communication Interfaces

These requests are used to access certain data of the communication interfaces. They do not include any elements specific to NUM numerical controls:

Requests	Request code	Answer code	Description	See
Mirror	250 (H'FA')	251 (H'FB')	Test of the system and communication path	4.5
Read-CPT	162 (H'A2')	210 (H'D2')	Management of the history of link faults of a unit	4.6
Etat-Station	163 (H'A3')	211 (H'D3')	Indicates the number of stations managed by the master and their status (connected or not)	4.7
Clear-CPT	164 (H'A4')	254 (H'FE')	Resets the error counters of a unit	4.8

3.4 Operating Modes

These requests are used to stop, start or initialise a unit connected to the bus:

Requests	Request code	Answer code	Description	See
Run	36 (H'24')	254 (H'FE')	Starts the tasks of a unit	4.9
Stop	37 (H'25')	254 (H'FE')	Stops the tasks of a unit	4.10
Init	51 (H'33')	99 (H'63')	Initialises a unit	4.11

3.5 File Transfers

These requests are used by a client unit to up- or download data blocks (segments), programmes or other data from or to a server.

This service is carried out in three steps:
- setting up of a sequence defining the nature of the service (upload or download) and identifying the data to be transferred
- running of the sequence at the initiative of the client
- closing of the sequence.

The requests used include:

Requests	Request code	Answer code	Description	See
Open-DownLoad-Sequence	58 (H'3A')	106 (H'6A')	Sets up a download sequence	4.12.1
Write-DownLoad-Segment	59 (H'3B')	107 (H'6B')	Transfers a segment	4.12.2
Close-DownLoad-Sequence	60 (H'3C')	108 (H'6C')	Closes the download sequence	4.12.3
Open-UpLoad-Sequence	61 (H'3D')	109 (H'6D')	Sets up an upload sequence	4.13.1
Read-UpLoad-Segment	62 (H'3E')	110 (H'6E')	Transfers a segment	4.13.2
Close-UpLoad-Sequence	63 (H'3F')	111 (H'6F')	Closes the upload sequence	4.13.3

3.6 Specific Requests

These requests are specific to NUM 1060 numerical controls.

The question and answer codes are set to H'F5'. The requests are differentiated by the additional request codes:

Requests	Additional request code	Additional answer code	Description	See
Delete-File	70 (H'46')	118 (H'76')	Deletes a programme from the RAM	4.14
Read-Memory-Free	71 (H'47')	119 (H'77')	Returns the number of available bytes in the RAM	4.15
Open-Directory	72 (H'48')	120 (H'78')	Reads the start of the list of programmes present in the NC memory	4.16.1
Directory	73 (H'49')	121 (H'79')	Reads the next part of the list or programmes present in the NC memory	4.16.2
Close-Directory	74 (H'4A')	122 (H'7A')	Ends read of the list of programmes present in the NC memory	4.16.3
Write-Message	75 (H'4B')	123 (H'7B')	Sending of messages by the supervisor	4.17
Stop-Automate	76 (H'4C')	124 (H'7C')	Stops the PLC user tasks	4.18
Init-Automate	77 (H'4D')	125 (H'7D')	Initialises the PLC	4.19
Run-Automate	75 (H'4F')	123 (H'7F')	Starts the PLC user tasks	4.20

4 Description of Requests

4.1 Read and Write Object Requests

4.1.1 Read Objects

Description

The Read-Object request is used to read objects accessible for read by the NC and PLC servers of the NUM 1060 numerical control (see Sec. 4.1.3).

The data are grouped by families of objects. Each family is identified by a number (segment number). Each object includes one or more data items.

After receiving the request, the server returns an answer code followed by the data corresponding to the type of the object(s) requested.

Transmission

Request code / Category code / Segment / Specific byte / Object address / Number of objects

Request code	1 byte: 54 (H'36')
Category code	1 byte: 0
Segment	1 byte: (see Sec. 4.1.3 for the segment numbers of objects accessible for read)
Specific byte	1 byte (size of the object(s) to be read on a PLC programmed in Ladder language, not significant in other cases)
Object address	1 word: address of the first object to be read in the family (see Sec. 4.1.3 for the address of the first object in the family)
Number of objects to be read	1 word: number of bytes to be read starting with the first (see Sec. 4.1.3 for the total number objects in the family)

REMARK *If the quantity specified is such that the answer could contain more than 126 data bytes, the request is refused (negative answer code).*

Reception

Positive answer

Answer code / Specific byte / Data

Answer code	1 byte: 102 (H'66')
Specific byte	1 byte (same value as in the request)
Data	data bytes: bits, bytes, words or long words (see Sec. 4.1.3 for the data type of the objects)

Negative answer

Answer code

Answer code	1 byte: 253 (H'FD')

A negative answer may be due to a segment number that is not recognised, an address outside the limits or an excessive number of objects.

Example

Read current programme number

Request

54 / 0 / 181 / 0 / 0 / 1 with:

- 54	read object request	(byte)
- 0	category code	(byte)
- 181	segment number corresponding to the current programme number	(byte)
- 0	not significant	(byte)
- 0	address of the single object in the family	(word)
- 1	request to read the single object in the family	(word)

Report

102 / 0 / 83 with:

- 102	read object answer	(byte)
- 0	not significant	(byte)
- 83	programme %83	(word)

4.1.2 Write Objects

Description

The Write-Object request is used to write the values of objects accessible for write by the NC and PLC servers of the numerical control (see Sec. 4.1.3).

The data are grouped by families of objects. Each family is identified by a number (segment number). Each object includes one or more data items.

After reception of the request, the server writes the data and returns an answer code.

Transmission

Request code / Category code / Segment / Specific byte / Object address / Number of objects / Data

Request code	1 byte: 55 (H'37')
Category code	1 byte: 0
Segment	1 byte: (see Sec. 4.1.3 for the segment number of objects accessible for read)
Specific byte	1 byte (size of the object(s) to be written on a PLC programmed in Ladder language, not significant in other cases)
Object address	1 word: address of the first object to be written in the family (see Sec. 4.1.3 for the address of the first object in the family)
Number of objects	1 word: number of bytes to be written starting with the first (see Sec. 4.1.3 for the total number objects in the family)

REMARK *If the quantity specified is such that the request contains more than 120 data bytes, the request is refused (negative answer code).*

Data	Data bytes: bits, bytes, words or long words (see Sec. 4.1.3 for the data type of the objects)

Reception

Answer code

Positive answer

Answer code	1 byte: 254 (H'FE')

Negative answer

Answer code	1 byte: 253 (H'FD')

A negative answer may be due to a segment number that is not recognised, an address outside the limits or an excessive number of objects.

Example

On a system in which the internal unit (IU) is the micron, write the DAT1 values for the four axes of the first group:
- DAT1 X = 100 mm, i.e. 100,000 IU,
- DAT1 Y = 150 mm, i.e. 150,000 IU,
- DAT1 Z = 30 mm, i.e. 30,000 IU,
- DAT1 C = 10 degrees, i.e. 100,000°/10,000.

Request

55 / 0 / 130 / 0 / 0 / 1 / 100 000 / 150 000 / 30 000 / 0 / 0 / 0 / 0 / 0 / 100 000 where:

- 55	write-object request	(byte)
- 0	category code	(byte)
- 130	segment number corresponding to DAT1 values	(byte)
- 0	not significant	(byte)
- 0	address of the first object to be written in the family	(word)
- 1	number of objects to be written in the family	(word)
- numerical data	DAT1 values in IU for the 9 axes of the group (9 long words)	

Report

254	write object answer	(byte)

4.1.3 Objects Accessible for Read and Write

In the table below:
- all the objects are accessible for read,
- the objects accessible for write are identified,
- the address of the first object in a family is 0 unless otherwise specified,
- IU represents the internal unit of the system defined in a machine parameter.

The objects accessible for read and write can be classified in two categories:
- objects of the NC software,
- objects of the PLC software.

The objects of the PLC software differ for:
- a PLC programmed in assembler,
- a PLC programmed in Ladder language.

4.1.3.1 Objects of the NC Software

Segment number	Name of the family of objects Properties	Values Units
128 H'80'	Axis position reference One object per axis group, size 9 long words Each object corresponds to the nine possible axes of a group	-99 999 999 to 99 999 999 IU or 10^{-4} degrees
129 H'81'	Axis measurement One object per axis group, size 9 long words Each object corresponds to the nine possible axes of a group	-99 999 999 to 99 999 999 IU or 10^{-4} degrees
130 H'82'	Axis DAT1 values One object per axis group, size 9 long words accessible for write Each object corresponds to the nine possible axes of a group	-99 999 999 to 99 999 999 IU or 10^{-4} degrees
131 H'83'	Axis DAT2 values One object per axis group, size 9 long words accessible for write Each object corresponds to the nine possible axes of a group	-99 999 999 to 99 999 999 IU or 10^{-4} degrees
132 H'84'	Axis DAT3 values One object per axis group, size 9 long words accessible for write Each object corresponds to the nine possible axes of a group	-99 999 999 to 99 999 999 IU
133 H'85'	Minimum dynamic axis travel One object per axis group, size 9 long words accessible for write Each object corresponds to the nine possible axes of a group	-99 999 999 to 99 999 999 IU or 10^{-4} degrees
134 H'86'	Maximum dynamic axis travel One object per axis group, size 9 long words accessible for write Each object corresponds to the nine possible axes of a group	-99 999 999 to 99 999 999 IU or 10^{-4} degrees
135 H'87'	Inclined axis angular value One object per axis group, size 1 long word accessible for write	-99 999 999 to 99 999 999 10^{-4} degrés
136 H'88'	Machine zero point One object per physical axis, size 1 long word accessible for write	-99 999 999 to 99 999 999 IU or 10^{-4} degrees
137 H'89'	Minimum static travel One object per physical axis, size 1 long word accessible for write	-99 999 999 to 99 999 999 IU or 10^{-4} degrees
138 H'8A'	Maximum static travel One object per physical axis, size 1 long word accessible for write	-99 999 999 to 99 999 999 IU or 10^{-4} degrees

Segment number	Name of the family of objects Properties	Values Units
139 H'8B'	Current corrections of a slave axis One object per physical axis, size 1 long word	-99 999 999 to 99 999 999 IU or 10^{-4} degrees
140 H'8C'	Axis position reference One object per physical axis, size 1 long word	-99 999 999 to 99 999 999 IU or 10^{-4} degrees
141 H'8D'	Axis measurement One object per physical axis, size 1 long word	-99 999 999 to 99 999 999 IU or 10^{-4} degrees
143 H'8F'	Driven axes One object per physical axis, size 1 long word	0 or 1
144 H'90'	Measured spindle speed setting One object per spindle, size 1 long word	2^{-15} of the maximum speed, rpm
145 H'91'	Measured spindle reference position One object per spindle, size 1 long word	0 to 3 599 999 10^{-4} degrees
146 H'92'	Tool corrections One object per tool, size 7 long words, maximum 255 tools Accessible for write, address of the first object in the family: 1 The 7 long words of an object have the following meanings:	
	Length in X (turning) or tool length (milling)	-9 999 999 to 9 999 999 IU
	Length in Z (turning) or cutter tip radius (milling)	-9 999 999 to 9 999 999 IU
	Radius of the cutting part (turning) or tool radius (milling)	-9 999 999 to 9 999 999 IU
	Dynamic correction in X (turning) or length (milling)	-9 999 999 to 9 999 999 IU
	Dynamic correction in Z (turning) or radius (milling)	-9 999 999 to 9 999 999 IU
	Tool nose direction (turning)	0 to 8
	Tool type	0: milling 1: turning 2: boring
147 H'93'	H variable of the table of dynamic correctors 255 objects, size 1 long word accessible for write Address of the first object in the family: 1	-99 999 999 to 99 999 999 no unit
148 H'94'	Interpolation status One object per axis group, size 4 long words The 4 long words of an object have the following meanings:	
	Current speed	mm/sample
	Distance remaining to be travelled on the current block (in the path)	µm
	Programmed speed	mm / min, mm/rev or V / L
	Speed override coefficient	2^{-16}
149 H'95'	Homing not done on the axes One object, size 1 long word accessible for write Each bit of the object indicates the status of the corresponding physical axis	0 to H'FFFFFFFF'
150 H'96'	Local data parameters (E800xx) 51 objects, size 1 long word accessible for write	-99 999 999 to 99 999 999 no unit

Segment number	Name of the family of objects Properties	Values Units
151 H'97'	Master axis reference position for interaxis calibration Maximum 1000 objects, size 1 long word accessible for write	-99 999 999 to 99 999 999 IU
152 H'98'	Slave axis correction in interaxis calibration Maximum 1000 objects, size 1 long word accessible for write	-99 999 999 to 99 999 999 IU
153 H'99'	Programme status One object per axis group, size 22 bytes The 22 bytes of an object have the following meanings:	
	Active programme number: 1 long word	
	Active block number: 1 word	
	Programme error number: 1 word	
	Errored block number: 1 word	
	Tool number: 1 word	
	Tool direction: 1 word. The bit indicating the tool direction is set in the high byte if the direction is positive or the low byte if negative - bit 0 set indicates the direction on the X axis (G16 P+/-) - bit 1 set indicates the direction on the Y axis (G16 Q+/-) - bit 2 set indicates the direction on the Z axis (G16 R+/-) - bits 3 to 7 are not used	
	Tool corrector number: 1 word	
	List of G functions present: 1 long word. A bit is set to indicate the presence of the function - bit 0: function G00 - bit 8: function G17 - bit 15: function G22 - bit 23: function G93 - bit 1: function G01 - bit 9: function G19 - bit 16: function G40 - bit 24: function G94 - bit 2: function G02 - bit 10: function G18 - bit 17: function G41 - bit 25: function G95 - bit 3: function G03 - bit 11: function G90 - bit 18: function G42 - bit 27: function G96 - bit 4: function G04 - bit 12: function G91 - bit 19: function G53 - bit 28: function G97 - bit 6: function G38 - bit 13: function G70 - bit 20: function G54 - bit 30: function G20 - bit 7: function G09 - bit 14: function G52 - bit 21: function G29 - bit 31: function G21	
	List of processes remaining to be executed: 1 word. The highest bit set identifies the function being executed - bit 0: function G78 - bit 1: end of external movement - bit 2: encoded M functions - bit 3: M post-function - bit 4: function G04 - bit 5: function G09 - bit 6: execution of a circle - bit 7: execution of a line - bit 8: JOG - bit 11: FEED STOP - bit 13: M pre-function - bit 15: T function	
157 H'9D'	Block end dimensions One object per axis group, size 11 long words The first 9 long words give the block end dimensions for the nine possible axes The tenth and eleventh long words give the coordinates on the X and Y axes respectively of the centre in circular interpolation	-99 999 999 to 99 999 999 UI or 10^{-4} degrees

Segment number	Name of the family of objects Properties	Values Units
180 H'B4'	Mode selection One object, size 1 word, accessible for write The value of the word indicates the mode selected:	
	- automatic mode	H'0000'
	- single step mode	H'0001'
	- MDI mode	H'0002'
	- dryrun mode	H'0003'
	- sequence number search mode	H'0004'
	- edit mode	H'0005'
	- test mode	H'0006'
	- manual mode	H'0007'
	- homing mode	H'0008'
	- shift mode	H'0009'
	- tool set mode	H'000A'
	- load mode	H'000D'
	- unload mode	H'000F'
181 H'B5'	Current programme number One object, size 1 word, accessible for write	1 à 99 999
224 H'E0'	Data transmitted to the programme being executed One object per axis group, size 1 long word, accessible for write The value of this word is recovered by the part programme after execution of a parametric expression of type «Ln = $1»	
226 H'E2'	Acknowledgement of a blocking message One object per axis group This object acknowledges the link after transmission of a blocking message: - "$11" to the PLC application — 2 = acknowledgement - "$21" to the UNI-TELWAY slave - "$31" to MAPWAY or ETHWAY - "$41" to the UNI-TELWAY master — 1 = link acknowledged without NC release; 2 = link acknowledged and NC released	

REMARK *The receiver address is encoded in the corresponding machine parameter:*
P100 for MAPWAY / ETHWAY
P110 for UNI-TELWAY master
P111 for UNI-TELWAY slave

4.1.3.2 Objects of the PLC Software Programmed in Assembler

For certain objects, the address of the first accessible object is different for write than for read.

Segment number	Name of the family of objects Properties	Associated mnemonic
160 H'A0'	Internal Boolean variables Size: 1 byte Address of the first object in the family: 0 for read, 16 (H'10') for write Address of the last object: 1279 (H'04FF')	B
161 H'A1'	Internal digital variables Size: 1 byte. These variables are located in two areas	M
	Area 1 Address of the first object in the area: 0 for read, 37 (H'25') for write address of the last object: 5631 (H'15FF')	
	Area 2 Address of the first object in the area: 9216 (H'2400'), address of the last object: 15615 (H'3CFF')	
162 H'A2'	Double internal digital variables Size: 1 word. These variables are included in two areas	MM
	Area 1 Address of the first object in the area: 0 for read, 37 (H'25') for write address of the last object: 5630 (H'15FE')	
	Area 2 Address of the first object in the area: 9216 (H'2400'), address of the last object: 15614 (H'3CFE')	
164 H'A4'	Digital inputs Size: 1 word. The inputs are included in two areas:	EN
	Digital inputs from the terminal board: 32 objects	
	Digital inputs from the NC: 16 objects, address of the first object in the area: 32 (H'20')	
165 H'A5'	Digital outputs Size: 1 byte. The outputs are included in two areas:	AN
	Digital outputs from the terminal board: 32 objects	
	Digital outputs from the NC: 16 objects, address of the first object in the area: 32 (H'20')	
168 H'A8'	Boolean inputs Size: 1 byte. The inputs are located in two areas:	E
	Boolean inputs from the terminal board: 256 objects	
	Boolean inputs from the NC: 128 objects. Address of the first object in the area: 256 (H'0100')	
169 H'A9'	Boolean outputs Size: 1 byte. The outputs are located in two areas:	A
	Boolean outputs from the terminal board: 256 objects	
	Boolean outputs from the NC: Address of the first object in the area: 256 (H'0100') for read, 259 (H'0103') for write Address of the last object: 383 (H'017F')	
172 H'AC'	Counters 16 objects in the family, size 1 word	C
174 H'AE'	Timers 16 objects in the family, size 1 word	T

4.1.3.3 Objects of the PLC Software Programmed in Ladder Language

The value i of the specific byte of the request identifies the type and size of the objects to be read or written:
- $0 \leq i \leq 7$: the object is bit i of the byte addressed,
- i = 64: the object is a byte,
- i = 65: the object is a word,
- i = 66: the object is a long word.

Segment number	Name of the family of objects Properties	Associated mnemonic
160 H'A0'	Variables not saved Size: depends on the specific byte of the request Address of the last object: 32767 (H'7FFF')	%V
161 H'A1'	Saved variables Size: depends on the specific byte of the request Address of the last object: 30719 (H'77FF')	%M
162 H'A2'	Common word variables* Size: depends on the specific byte of the request Address of the last object: 16255 (H'3F7F')	%S
164 H'A4'	Inputs from the NC* Size: depends on the specific byte of the request Address of the last object: 3967 (H'0F7F')	%R
165 H'A5'	Outputs to the NC* Size: depends on the specific byte of the request Address of the last object: 3967 (H'0F7F')	%W
168 H'A8'	Inputs from the terminal board** Size: depends on the specific byte of the request Address of the last object: 28479 (H'6F3F')	%I
169 H'A9'	Outputs to the terminal board** Size: depends on the specific byte of the request Address of the last object: 28479 (H'6F3F')	%Q

* For these objects, only addresses in which the low byte is less than or equal to H'7F' are allowed.
** For these objects, only addresses in which the low byte is less than or equal to H'3F' are allowed.

4.2 Unsolicited Data

Description

The Unsolicited-Data request allows an entity to send data at its own initiative without first receiving a request.

No report is sent by the receiver. The request can however be associated with another request at the initiative of the receiver of the unsolicited data, used to acknowledge the unsolicited data.

Transmission

Request code / Category code / Data origin / Data length / Data

Request code	1 byte: 256 (H'FC')
Category code	1 byte: 6
Data origin	1 byte: code of the unsolicited data origin (codes 0 to 31 and FF are reserved for the NC and must not be used)
Data length	1 byte: data length in bytes
Data	Data bytes

4.3 Unit Identification

Description

The Identification request is used to identify the NC unit.

Transmission

Request code/Category code

Request code	1 byte: 15 (H'0F')
Category code	1 byte: 0

Reception

Positive answer

Answer code / Product type / Subtype / Product version / Data

Answer code	1 byte: 63 (H'3F')
Product type	1 byte: 40-49 and 100-109 reserved for NUM 100: NUM 1060 102: NUM 1040 101: NUM 1060 Series II 103: NUM 1060-7
Subtype	1 byte: ASCII code of the letter associated with the version (e.g.. H'41': version A)
Product version	1 byte: the first half-byte specifies the index associated with the version (e.g.. H'30': index 3)
Data	Data bytes: ASCII character string beginning with a length byte (e.g. «\9NUM1060S2»)

Negative Answer

No negative answer.

4.4 Unit Status Data

Description

The Status request returns the status of the complete numerical control (NC + PLC).

Transmission

Request code / Category code / Desired detail

Request code	1 byte: 49 (H'31')
Category code	1 byte: 0
Desired detail	1 byte: axis group index

Reception

Positive answer

Answer code / Current status / Status mask / Data / Specific data

Answer code	1 byte: 97 (H'61')
Current status	String of 8 bits: - bit 0: system inoperative, - bit 1: detection of a recoverable error, - bit 2: detection of an unrecoverable error, - bit 3: check the auxiliary power source, - bit 4: system reset, - bit 5: critical operation being executed, - bit 6: halt, - bit 7: local mode.
Status mask	String of 8 bits: mask giving the current significant status bits
Data	32 bytes: - 22 bytes corresponding to the programme status of the axis group specified (segment 153 of the NC objects, see Sec. 4.1.3), - 1 byte: operator panel status (EN.100 in assembler, %R3.B in Ladder language), - 1 byte: NC status (EN.101 in assembler, %R5.B in Ladder language), - 1 byte: NC mode (EN.20 in assembler, %R16.B in Ladder language), - 1 byte: machine mode (EN.22 in assembler, %R18.B in Ladder language), - 1 word: current programme number (ENN.2C in assembler, %R1A.W in Ladder language), - 1 byte: PLC status • 0: no application • 2: running • 1: stopped • 3: faulty
Specific data	16 bytes: table of the PLC memory field whose address is specified by the PLC programme

Negative answer

Answer code

Answer code	1 byte: 253 (H'FD')

Cause of rejection: axis group unknown.

4.5 Test of the System and Communication Path

Description

The Mirror request tests the system and the communication path.

The client sends a sequence to the server and the server returns the same sequence.

This request does not contain anything specific to NUM numerical controls.

Transmission

Request code / Category code / Data

Request code	1 byte: 250 (H'FA')
Category code	1 byte: 0
Data	Byte string (maximum 126)

Reception

Positive answer

Answer code / Data

Answer code	1 byte: 251 (H'FB')
Data	Byte string identical to the string sent

Negative answer

No negative answer.

4.6 Management of Unit Fault History

Description

Each station manages a history of link faults (character errors, frame errors, protocol errors) by counting four error types in counters (16-bit words):

- number of messages sent and not acknowledged,
- number of messages sent and rejected,
- number of messages received and not acknowledged,
- number of messages received and rejected.

REMARK *There is never any counter overflow. The counters remain frozen at H'7FFF' (32767) until they are reset by a Clear-CPT request (see Sec. 4.8).*

The Read-CPT request is used to read the counters.

This request does not include anything specific to NUM numerical controls.

Transmission

	Request code / Category code
Request code	1 byte: 162 (H'A2')
Category code	1 byte: 0

Reception

Positive answer

	Answer code / Data
Answer code	1 byte: 210 (H'D2')
Data	4 words: - number of messages sent and not acknowledged - number of messages sent and rejected - number of messages received and not acknowledged - number of messages received and rejected

Negative answer

Answer code	1 byte: 253 (H'FD')

Cause of rejection: request unknown.

4.7 Stations Managed by the Master

Description

The bus manager manages the number of stations (link addresses) set and the connected or unconnected status of these stations.

The Etat-Station request returns the number of stations and their status.

This request:
- is processed only on the UNI-TELWAY master
- does not include anything specific to NUM numerical controls.

Transmission

Request code / Category code

Request code	1 byte: 163 (H'A3')
Category code	1 byte: 0

Reception

Positive answer

Answer code / Number of stations / Station status

Answer code	1 byte: 211 (H'D3')
Number of stations	1 byte: number of stations managed
Station status	1 bit per station (rank = station address): - bit = 1: station connected - bit = 0: station disconnected

Negative answer

Answer code	1 byte: 253 (H'FD')

Cause of rejection: request unknown,
manager not configured.

4.8 Error Counter Reset

Description

The Clear-CPT request resets the error counters of a unit (see Sec. 4.6).

This request does not contain anything specific to NUM numerical controls.

Transmission

	Request code / Category code
Request code	1 byte: 164 (H'A4')
Category code	1 byte: 0

Reception

Positive answer

Answer code	1 byte: 254 (H'FE')

Negative answer

Answer code	1 byte: 253 (H'FD')

Cause of rejection: request unknown,
access rights insufficient.

4.9 Running the Tasks of a Unit

Description

The Run request starts an NC cycle.

Transmission

	Request code / Category code
Request code	1 byte: 36 (H'24')
Category code	1 byte: 0

Reception

	Answer code

Positive answer

Answer code	1 byte: 254 (H'FE')

Negative answer

Answer code	1 byte: 253 (H'FD')

Cause of rejection: NC status incompatible with a cycle start.

4.10 Stopping Unit Tasks

Description

The Stop request stops axis feed (FEED STOP). The spindles are not affected.

Transmission

Request code/ Category code

Request code	1 byte: 37 (H'25')
Category code	1 byte: 0

Reception

Answer code

Positive answer

Answer code	1 byte: 254 (H'FE')

Negative answer

Answer code	1 byte: 253 (H'FD')

Cause of rejection: NC status incompatible with feed stop.

4.11 Initialising a Unit

Description

The Init request resets the NC.

Transmission

Request code / Category code

Request code	1 byte: 51 (H'33')
Category code	1 byte: 0

Reception

Answer code / Result

Answer code	1 byte: 99 (H'63')

Positive answer

Result	1 byte: 0

Negative answer

Result	1 byte: 10 (H'0A')

Cause of rejection: NC status incompatible with a reset.

4.12 File Download Requests

A file is downloaded as follows:
- file opened by a first request (Open-DownLoad-Sequence) specifying the type and possibly the name of the file to be downloaded,
- writing of the file data by consecutive numbered Write-DownLoad-Segment requests,
- closing of the file by the last request (Close-DownLoad-Sequence).

REMARKS *A single file (all types) or all the Ladder modules can be downloaded.*

Part programme type files for storage are automatically closed by the NC if the requester does not act within a timeout set by machine parameter P84 (see Parameter Manual).

4.12.1 Setting up a Download Sequence

Description

The Open-DownLoad-Sequence request starts downloading a file.

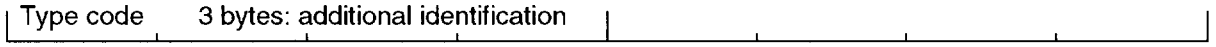
Transmission

Request Code / Category code / File identification [/Extension code / Filename]

Request code 1 byte: 58 (H'3A')

Category code 1 byte: 0

File identification 2 long words:



Long word 1: file identification — Long word 2: additional information

File type to be downloaded	Type code byte 1	Additional identification byte 2	byte 3	byte 4	Additional information long word 2
Axis calibration	H'02'	not significant			not significant
Macros	H'03'	not significant			not significant
Machine parameters	H'05'	not significant			not significant
PLC assembler binary code	H'06'	0	0	0	not significant
PLC Ladder binary code The additional identification gives the module type and number	H'07'	1 : %TSi 2 : %TFi 3 : %SPi 4 : %THi 5 : %INI 16 : all Ladder and C modules	0 0 0 0 0 0	0 ≤ i ≤ 4 0 ≤ i ≤ 15 0 ≤ i ≤ 255 0 ≤ i ≤ 15 0 0	not significant
C language files	H'06' / H'07' *	H'41'	not significant		
Text files	H'06' / H'07' *	H'61'	not significant		
Part programmes in drip feed mode	H'0C'	programme number indexed by the axis group			ring buffer size: i (i even and 2048 ≤ i ≤ 32768)
Part programmes for storage	H'12'	programme number indexed by the axis group			not significant

* The type code depends on the PLC programming language: H'06' for assembler, H'07' for Ladder.

[Extension code]	1 byte: 1 (this optional byte is used only for files in C and text files)
[Filename]	Byte string: can only be used when the extension code = 1. It is an ASCII character string beginning with a length byte. This string contains the name and, where required, the path of the file to be downloaded (e.g. «\D\APPLI\AUTO.C»)

Reception

Answer code / Status

Answer code	1 byte: 106 (H'6A')
Status	1 byte: see table below

Status	Type code of the files concerned	Description
0	all files	request executed
1	H'06', H'07', H'0C' et H'12'	file already exists
2	H'06', H'07', H'0C' et H'12'	other programme being downloaded or edited
3	H'06', H'07' et H'12'	free memory space less than 128 bytes
10	H'0C'	the NC is not in reset state
11	H'0C' et H'12'	programme number incoherent or PPP buffer size incorrect
21	H'06' et H'07'	file type not recognised
28	H'06' et H'07'	system error

4.12.2 Transferring a Segment

Description

The Write-DownLoad-Segment request is used to load data in a file that has already been opened (see Sec. 4.12.1).

A maximum of 122 data bytes can be sent per request. The complete programme is therefore sent using a sequence of requests numbered by increments of 1.

Transmission

Request code / Category code / Segment number / Segment length / Data

Request code	1 byte: 59 (H'3B')
Category code	1 byte: 0
Segment number	1 word: the number of the first request is 1 and the subsequent requests are numbered by increments of 1.
Segment length	1 word: number of data bytes (maximum 122)
Data	Character string (ASCII codes) of the part of the programme contained in the request. In the case of a part programme, each block must end with CR and LF (ASCII codes H'0D' and H'0A')

Reception

Answer code / Status / Segment number

Answer code	1 byte: 107 (H'6B')
Status	1 byte: see table below

Status	Type code of the files concerned	Description
0	all files	request executed
3	H'06', H'07' et H'12'	memory full
4	H'06', H'07', H'0C' et H'12'	no file being downloaded
7	H'0C'	ring buffer full
9	H'06', H'07', H'0C' et H'12'	data length error
10	H'0C'	NC no longer in drip feed mode
11	H'06', H'07' et H'0C'	a block has more than 120 characters, data incoherent
20	H'06', H'07', H'0C' et H'12'	other programme being downloaded, sender error
25	H'06', H'07', H'0C' et H'12'	sequence error
28	H'06' et H'07'	system error

Segment number	1 word: the number corresponds to the segment number of the data being transmitted.

4.12.3 Ending the Download Sequence

Description

The Close-DownLoad-Sequence request is used to end downloading of a file already opened.

Transmission

Request code / Category code

Request code	1 byte: 60 (H'3C')
Category code	1 byte: 0

Reception

Answer code / Status

Answer code	1 byte: 108 (H'6C')
Status	1 byte: see table below

Status	Type code of the files concerned	Description
0	all files	request executed
4	H'06', H'07', H'0C' et H'12'	no file being downloaded
11	H'12'	file deleted: the last block loaded did not end with LF
20	H'06', H'07', H'0C' et H'12'	other file being downloaded, sender error
28	H'06' et H'07'	system error

4.12.4 Example of File Download

The following programme is to be sent for storage:

```
%32.1
N10 X0 Y0
N20 X-140
N30 Y-60
```

4.12.4.1 Opening the File (Open-DownLoad-Sequence request)

Request

58 / 0 / H'12000141' / 0 where:

- 58	file open request	(byte)
- 0	category code	(byte)
- H'12000141'	H'12': part programme type file for storage H'000141' = 321: indexed programme number	(long word)
- 0	not significant	(long word)

Report

106 / 0 where:

- 106	file open answer	(byte)
- 0	request executed	(byte)

4.12.4.2 Writing the Start of the Programme (Write-DownLoad-Segment request)

Request

59 / 0 / 1 / 16 / H'4E - 31 - 30 - 20 - 58 - 30 - 20 - 59 - 30 - 0D - 0A - 4E - 32 - 30 - 20 - 58' where:

- 59	segment transfer request	(byte)
- 0	category code	(byte)
- 1	segment number	(word)
- 16	number of data bytes	(word)
- H'4E - 31 - ... - 20 - 58'	conversion of the first byte and part of the second to ASCII code "N 1 0 _ X 0 _ Y 0 CR LF N 2 0 _ X"	(byte string)

Report

107 / 0 / 1 where:

- 107	segment transfer answer	(byte)
- 0	request executed	(byte)
- 1	segment number	(word)

4.12.4.3 Writing the Next Part of the Programme (Write-DownLoad-Segment request)

REMARK *The programme was short enough to be loaded by a single Write-DownLoad-Segment request. It was divided only to make the example more complete.*

Request

59 / 0 / 2 / 16 / H'2D - 31 - 34 - 30 - 0D - 0A - 4E - 33 - 30 - 20 - 59 - 2D - 36 - 30 - 0D - 0A' where:

- 59	segment transfer request	(byte)
- 0	category code	(byte)
- 2	segment number	(word)
- 16	number of data bytes	(word)
- H'4E - 32 - ... - 0D - 0A'	conversion of the end of the second block and all of the third to ASCII code "- 1 4 0 CR LF N 3 0 _ Y - 6 0 CR LF"	(byte string)

Report

107 / 0 / 2 where:

- 107	segment transfer answer	(byte)
- 0	request executed	(byte)
- 2	segment number	(word)

4.12.4.4 Closing the File (Close-DownLoad-Sequence request)

Request

60 / 0 where:

- 60	file close request	(byte)
- 0	category code	(byte)

Report

108 / 0 where:

- 108	file close answer	(byte)
- 0	request executed	(byte)

4.13 File Upload Requests

A file is uploaded as follows:
- file opened by a first request (Open-Upload-Sequence) specifying the type and possibly the name of the file to be uploaded
- reading of the file data by consecutive numbered Read-Upload-Segment requests
- closing of the file by the last request (Close-Upload-Sequence).

REMARKS *A single file (all types) or all the Ladder modules can be uploaded.*

Part programme type files for storages are automatically closed by the NC if the requester does not act within a timeout set by machine parameter P84 (see Parameter Manual).

4.13.1 Setting up an Upload Sequence

Description

The Open-Upload-Sequence request opens the file to be uploaded.

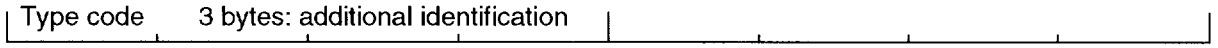
Transmission

Request Code / Category code / File identification [/Extension code / Filename]

Request code 1 byte: 61 (H'3D')

Category code 1 byte: 0

File identification 2 long words:



Long word 1: file identification — Long word 2: additional information

File type to be uploaded	Type code byte 1	Additional identification byte 2	byte 3	Additional information byte 4	long word 2
Axis calibration	H'02'	not significant			not significant
Macros	H'03'	not significant			not significant
Machine parameters	H'05'	not significant			not significant
PLC assembler binary code	H'06'	0	0	0	not significant
PLC Ladder binary code The additional identification gives the module type and number	H'07'	1 : %TSi 2 : %TFi 3 : %SPi 4 : %THi 5 : %INI 16 : all Ladder and C modules	0 0 0 0 0 0	0 ≤ i ≤ 4 0 ≤ i ≤ 15 0 ≤ i ≤ 255 0 ≤ i ≤ 15 0 0	not significant
C language files	H'06' / H'07' *	H'41'	not significant		
Text files	H'06' / H'07' *	H'61'	not significant		
Part programmes for storage	H'12'	Programme number indexed by the axis group			not significant

* The type code depends on the PLC programming language: H'06' for assembler, H'07' for Ladder.

[Extension code] 1 byte: 1 (this optional byte is used only for files in C and text files)

[Filename] Byte string: can only be used when the extension code = 1. It is an ASCII character string beginning with a length byte. This string contains the name and, where required, the path of the file to be uploaded (e.g. «\D\APPLI\AUTO.C»)

Reception

Answer code / Status

Answer code 1 byte: 109 (H'6D')

Status 1 byte: see table below

Status	Type code of the files concerned	Description
0	all files	request executed
2	H'06', H'07' et H'12'	other programme being uploaded or edited
5	H'06', H'07' et H'12'	no such programme number
15	H'12'	programme empty
21	H'06' et H'07'	erreur in filename
28	H'06' et H'07'	system error

4.13.2 Transferring a Segment

Description

The Read-Upload-Segment request is used to upload data from an NC file that has already been opened (see Sec. 4.13.1).

A maximum of 122 data bytes can be read per request. The complete programme is therefore read using a sequence of requests numbered by increments of 1.

Transmission

Request code/Category code/Segment number

Request code	1 byte: 62 (H'3E')
Category code	1 byte: 0
Segment number	1 word: the number of the first request is 1 and the subsequent requests are numbered by increments of 1.

Reception

Answer code/Status/Segment number/Segment length/Data

Answer code	1 byte: 110 (H'6E')
Status	1 byte: see table below

Status	Type code of the files concerned	Description
0		request executed, data remaining to be transmitted
4	H'06', H'07' et H'12'	no file being uploaded
15		request executed, no more data, file automatically closed
20	H'06', H'07' et H'12'	other filed being uploaded, sender error
25	H'06', H'07' et H'12'	sequence error
28	H'06' et H'07'	system error

Segment number	1 word: the number corresponds to the segment number of the data being transmitted.
Segment length	1 word: number of data bytes
Data	character string (ASCII codes) of the part of the programme transmitted

4.13.3 Ending the Upload Sequence

Description

The Close-Upload-Sequence request is used to end uploading of a file already opened.

Transmission

Request code/Category code

Request code	1 byte: 63 (H'3F')
Category code	1 byte: 0

Reception

Answer code/Status

Answer code	1 byte: 111 (H'6F')
Status	1 byte: see table below

Status	Type code of the files concerned	Description
0		request executed
4	H'06', H'07' et H'12'	file already closed
20	H'06', H'07' et H'12'	other file being uploaded
28	H'06' et H'07'	system error

4.13.4 Example of File Upload

Programme %146.1 is to be uploaded.

4.13.4.1 Opening the File (Open-Upload-Sequence request)

Request

61 / 0 / H'120005B5' / 0 where:

- 61	file open request	(byte)
- 0	category code	(byte)
- H'120005B5'	H'12': part programme type file for storage H'0005B5' = 1461: indexed programme number	(long word)
- 0	not significant	(long word)

Report

109 / 0 where:

- 109	file open answer	(byte)
- 0	request executed	(byte)

4.13.4.2 Reading the Start of the Programme (Read-Upload-Segment request)

Request

62 / 0 / 1 where:

- 62	segment transfer request	(byte)
- 0	category code	(byte)
- 1	segment number	(word)

Report

110 / 0 / 0 / 1 / 121 / Data where:

- 110	segment transfer answer	(byte)
- 0	request executed	(byte)
- 0	request executed, data remaining	(byte)
- 1	request number	(word)
- 121	number of data bytes	(word)
- Data	the data consist of a character string (ASCII code) of the part of the programme transmitted	(bytes)

A test is made on the status byte to determine whether the complete programme has been transmitted (status = 15).

If not (status = 0), it is necessary to:
- issue another read-upload-segment request, or
- issue a close-upload-sequence request if it is desired to upload only part of the program.

4.13.4.3 Writing the Next Part of the Programme (Read-Upload-Segment request)

Request

62 / 0 / 2 where:

- 62	segment transfer request	(byte)
- 0	category code	(byte)
- 2	request number	(word)

Report

110 / 0 / 15 / 2 / 35 / Data where:

- 110	segment transfer answer	(byte)
- 0	request executed	(byte)
- 15	request executed, no more data, file closed	(byte)
- 2	request number	(word)
- 35	number of data bytes	(word)
- Data	the data consist of a character string (ASCII code) of the part of the programme transmitted	(bytes)

A test is made on the status byte to determine whether the complete programme has been transmitted (status = 15).

If not (status = 0), it is necessary to:
- issue another read-upload-segment request, or
- issue a close-upload-sequence request if it is desired to upload only part of the program.

4.13.4.4 Ending the Upload (Close-Upload-Sequence request)

Read of status 0 after the last program read request indicates that upload is not completed.

Request

63 / 0 where:

- 63	file close request	(byte)
- 0	category code	(byte)

Report

111 / 0 where:

- 111	file close answer	(byte)
- 0	request executed	(byte)

4.14 Deleting a Programme from the RAM

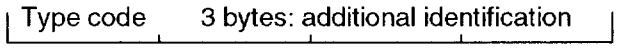
Description

The Delete-File request is a special NUM request used to delete a part programme from the NC RAM or a PLC file.

Transmission

Request code / Category code / Additional request code / File identification [/Extension code / Filename]

Request code	1 byte: 245 (H'F5')
Category code	1 byte: 0
Additional request code	1 byte: 70 (H'46')
File identification	1 long word:



Type of file to be deleted	Type code byte1	Additional identification byte2	byte3	byte4
PLC assembler binary code	H'06'	0	0	0
PLC Ladder binary code	H'07'	1 : %TSi	0	$0 \le i \le 4$
		2 : %TFi	0	$0 \le i \le 15$
The additional identification gives		3 : %SPi	0	$0 \le i \le 255$
the module type and number		4 : %THi	0	$0 \le i \le 15$
		5 : %INI	0	0
		16 : all	0	0
		Ladder and C modules		
C language files	H'06' / H'07' *	H'41'	not significant	
Text files	H'06' / H'07' *	H'61'	not significant	
Part programmes for storage	H'12'	programme number indexed by the axis group		

* The type code depends on the PLC programming language: H'06' for assembler, H'07' for Ladder.

[Extension code]	1 byte: 1 (this optional byte is used only for files in C and text files)
[Filename]	Byte string: can only be used when the extension code = 1. It is an ASCII character string beginning with a length byte. This string contains the name and, where required, the path of the file to be uploaded (e.g. «\D\APPLI\AUTO.C»)

Reception

Answer code / Additional answer code / Status

Answer code	1 byte: 245 (H'F5')
Additional answer code	1 byte: 118 (H'76')

Status	1 byte: see table below

Status	Description
0	request executed
2	request rejected, operation in the programme area
5	request rejected, no such file
10	request rejected, programme or subroutine being executed, PLC status incompatible with file deletion

Example

Deletion of part programme %44.1

Request

245 / 0 / 70 / 441 where:

- 245	specific NUM request	(byte)
- 0	category code	(byte)
- 70	additional file deletion request code	(byte)
- H'120001B9'	H'12': part programme type file for storage H'0001B9' = 441: indexed programme number	(long word)

Report

245 / 118 / 0 where:

- 245	specific NUM answer	(byte)
- 118	additional file deletion answer code	(byte)
- 0	request executed	(byte)

4.15 Reading the Number of Bytes Available in the RAM

Description

The Read-Memory-Free request is a specific NUM request that returns the number of bytes available in the NC RAM.

Transmission

Request code / Category code / Additional request code

Request code	1 byte: 245 (H'F5')
Category code	1 byte: 0
Additional request code	1 byte: 71 (H'47')

Reception

Answer code / Additional answer code / Status / Data

Answer code	1 byte: 245 (H'F5')
Additional answer code	1 byte: 119 (H'77')
Status	1 byte: see table below

Status	Description
H'00'	request executed
H'02'	request rejected, operation in the programme area

Data	1 long word: number of bytes available

4.16 NC Memory Directory Read Requests

Reading the list of programmes present in the NC RAM begins by the Open-Directory request.

The list of programmes is read by increasing numerical order.

If the list is too long to be completely contained in the answer, the rest can be read by the Directory request (this request can be used as many times as necessary to read the complete list).

The Close-Directory request is used to close the read operation at any time before reaching the end of the list.

4.16.1 Reading the Start of the List of Programmes Present in the NC RAM

Description

The Open-Directory request is a specific NUM request that returns the list of part programmes present in the NC RAM (or only part of the list if the list is too long to be contained completely in the answer: the list can contain a maximum of 15 programme names).

If the list is completely contained in the answer to the request, the operation is automatically closed and the Close-Directory request is unnecessary.

Transmission

Request code / Category code / Additional request code / Programme number

Request code	1 byte: 245 (H'F5')
Category code	1 byte: 0
Additional request code	1 byte: 72 (H'48')
Programme number	1 long word: programme number on which the list is to be started (programme number x 10 + axis group number). If no programme with this number exists, the list begins on the next higher number.

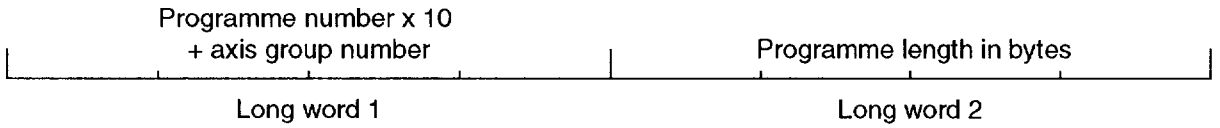
Reception

Answer code/Additional answer code/Status/Data

Answer code	1 byte: 245 (H'F5')
Additional answer code	1 byte: 120 (H'78')
Status	1 byte: see table below

Status	Description
0	request executed, data remaining to be transmitted
2	request rejected, operation in the programme area
9	request rejected, buffer too small
15	request executed, no more data, read operation automatically closed

Data String of long words (2 per programme) containing the list of programmes



4.16.2 Reading the Next Part of the List of Programmes Present in the NC Memory

Description

The Directory request is a specific NUM request which returns the next part of the list of part programmes after an Open-Directory request or a previous Directory request (the list can contain a maximum of 15 programme names).

If the end of the list is included in the answer to the request, the operation is automatically closed and the Close-Directory request is unnecessary.

Transmission

Request code / Category code / Additional request code

Request code	1 byte: 245 (H'F5')
Category code	1 byte: 0
Additional request code	1 byte: 73 (H'49')

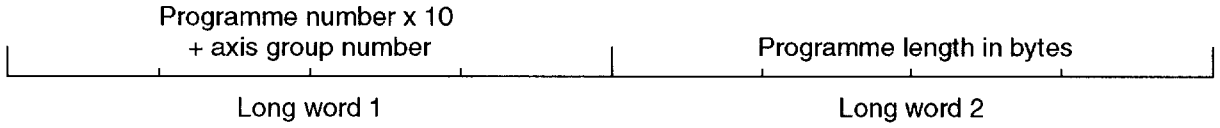
Reception

Answer code/Additional answer code/Status/Data

Answer code	1 byte: 245 (H'F5')
Additional answer code	1 byte: 121 (H'79')
Status	1 byte: see table below

Status	Description
0	request executed, data remaining to be transmitted
2	request rejected, operation in the programme area
9	request rejected, buffer too small
15	request executed, no more data, read operation automatically closed

Data String of long words (2 per programme) containing the list of programmes



4.16.3 Closing a Directory Read

Description

The Close-Directory request is a specific NUM request used to close read of the list of part programmes.

Transmission

Request code / Category code / Additional request code

Request code	1 byte: 245 (H'F5')
Category code	1 byte: 0
Additional request code	1 byte: 74 (H'4A')

Reception

Answer code / Additional answer code / Status

Answer code	1 byte: 245 (H'F5')
Additional answer code	1 byte: 122 (H'7A')
Status	1 byte: see table below

Status	Description
0	request executed
4	request rejected, operation already closed

4.16.4 Example of Directory Read

Read the list of programmes %9xxx.x.

4.16.4.1 Open-Directory Request

Request

245 / 0 / 72 / 441 where:

- 245	specific NUM request	(byte)
- 0	category code	(byte)
- 72	additional open directory request code	(byte)
- 90 000	the list is to begin with programme %9000.0	(long word)

Report

245 / 118 / 0 / Data where:

- 245	specific NUM answer	(byte)
- 120	additional open directory answer code	(byte)
- 0	request executed, data remaining to be transmitted	(byte)
- Data	the consecutive data are the part programme numbers indexed by the axis group and the length in bytes of the programmes	(long words)

A test is made on the last received programme number (programme number ≥ 100,000) to determine whether all the %9xxx.x programmes are included in the list.

If not, a directory request must be made to continue reading the directory.

4.16.4.2 Reading the Next part of the Directory (Directory request)

Request

245 / 0 / 73 where:

- 245	specific NUM request	(byte)
- 0	category code	(byte)
- 73	additional directory request code	(byte)

Report

245 / 121 / 0 / Data where:

- 245	specific NUM answer	(byte)
- 121	additional directory answer code	(byte)
- 0	request executed, data remaining to be transmitted	(byte)
- Data	the consecutive data are the part programme numbers indexed by the axis group and the length in bytes of the programmes	(long words)

A test is made on the last received programme number (programme number ≥ 100,000) to determine whether all the %9xxx.x programmes are included in the list.

If not, a directory request must be made to continue reading the directory and so forth until the complete list has been read.

4.16.4.3 Close-Directory Request

Read of status 0 after the last directory read request means that the directory has not been closed.

Request

245 / 0 / 74 where:

- 245	specific NUM request	(byte)
- 0	category code	(byte)
- 74	additional directory close request code	(byte)

Report

245 / 122 / 0 / Data where:

- 245	specific NUM answer	(byte)
- 122	additional close directory request code	(byte)
- 0	request executed, directory closed	(byte)

4.17 Sending of Messages by the Supervisor

Description

The Write-Message request is a specific NUM request allowing the supervisor to send one to three lines of 32 alphanumeric characters. These messages are displayed on the receiver NC.

Transmission

Request code / Category code / Question code / Message index / Length / Data

Request code	1 byte: 245 (H'F5')
Category code	1 byte: 0
Question code	1 byte: 75 (H'4B')
Message index	1 byte: not significant on NUM 1060
Length	1 byte: 1 to 3 (number of message lines sent)
Data	Data bytes: ASCII character string (H'1F' < ASCII code < H'80': the characters outside this interval are not displayable) including 32 bytes per message line.

Reception

Answer code/Additional answer code

Answer code	1 byte: 245 (H'F5')

Positive answer

Additional answer code	1 byte: 254 (H'FE')

Negative answer

Additional answer code	1 byte: 253 (H'FD')

Cause of rejection: request unknown

4.18 Stopping the PLC User Tasks

Description

The Stop request is a specific NUM request used to stop the PLC tasks.

Transmission

Request code/Category code/Additional request code/Spare

Request code	1 byte: 245 (H'F5')
Category code	1 byte: 0
Additional request code	1 byte: 76 (H'4C')
Spare	1 byte: 0

Reception

Answer code/Additional answer code/Status

Answer code	1 byte: 245 (H'F5')
Additional answer code	1 byte: 124 (H'7C')

Positive answer

Status	1 byte: 0

Negative answer

Status	1 byte: 10 (PLC status incompatible)

4.19 Initialising the PLC

Description

The Init request is a specific NUM request used to initialise the PLC.

Transmission

Request code/Category code/Additional request code/Init type

Request code	1 byte: 245 (H'F5')
Category code	1 byte: 0
Additional request code	1 byte: 77 (H'4D')
Init type	1 byte: see table below

Init type	Description
0	complete init
1	partial init
2	init

Reception

Answer code/Additional answer code/Status

Answer code	1 byte: 245 (H'F5')
Additional answer code	1 byte: 125 (H'7D')

Positive answer

Status	1 byte: 0

Negative answer

Status	1 byte: 10 (PLC status incompatible)

4.20 Running The PLC User Tasks

Description

The Run request is a specific NUM request used to run the PLC user tasks.

Transmission

Request code/Category code/Additional request code/Spare

Request code	1 byte: 245 (H'F5')
Category code	1 byte: 0
Additional request code	1 byte: 79 (H'4F')
Spare	1 byte: 0

Reception

Answer code/Additional answer code/Status

Answer code	1 byte: 245 (H'F5')
Additional answer code	1 byte: 127 (H'7F')

Positive answer

Status	1 byte: 0

Negative answer

Status	1 byte: 10 (PLC status incompatible)

5 Request Transmission by a Client Application

Requests are sent and reports received by the client application of a NUM 1060 numerical control using two functions:
- one requester function: Neto
- one reception function: Neti.

5.1 Requester Function

Description

The Neto function is used to send a request to a distant station. The request is sent on one of the 16 source ports (H'50' to H'5F') in the case of a master station or source port H'50' in the case of a slave station.

Before using the Neto function, the request must be stored in a transmission buffer.

This function is non-blocking. It can be used in a sequential task %TS0 to %TS4.

Several requests can exist simultaneously if they use different source ports.

Transmission

Syntax in Ladder Language

```
Neto (Source port, Buffer address)
```

Syntax in Assembler

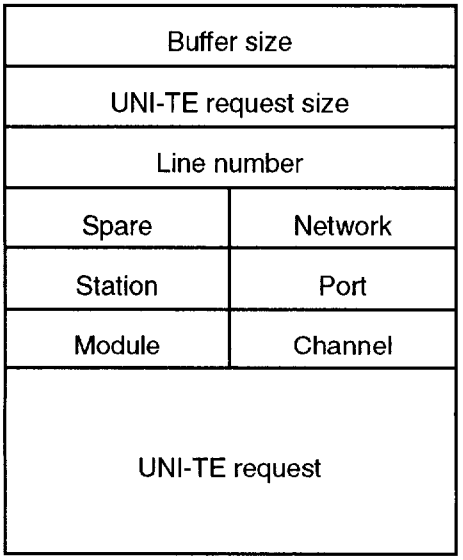
```
LDB     Source port          Store in accumulator B
LX@     Buffer address       Store in index register X
NETO                         Send the request
```

Where

Source port	Source port number (H'50' to H'5F' or only H'50' for a slave).
Buffer address	Address of the buffer containing the request to be sent.

Transmission Buffer Structure

Buffer address →



Buffer size	1 word: not significant
Request size	1 word: request length in bytes (maximum 128)
Line number	1 word: - H'20', H'21', H'24' to H'2B' (UNI-TELWAY lines) - H'30' (MAPWAY or ETHWAY) - H'40' (ETHERNET)
Server address	6 bytes: - Spare: not significant - Network: network number - Station: station number - Port: port number - Module: module number - Channel: channel number
UNI-TE request	Data: bytes, words or long words (maximum 128 bytes) in the request

Code returned after the request

Answer code 1 byte: see table below

Answer code	Description
0	transmission OK
1	buffer too long
2	zero buffer length
3	queue full, all buffers full
4	port number error
5	option not valid for this request
8	line number not valid

5.2 Reception Function

Description

The Neti function is used to receive a report on a request sent earlier or unsolicited data. The answer is received on one of the 16 source ports (H'50' to H'5F') in the case of a master station or source port H'50' in the case of a slave station.

Before using the Neti function, the reception buffer size must be specified.

This function is non-blocking. It can be used in a sequential task %TS0 to %TS4.

Several requests can exist simultaneously if they use different source ports.

Transmission

Syntax in Ladder Language

```
Neti (Source port, Buffer address)
```

Syntax in Assembler

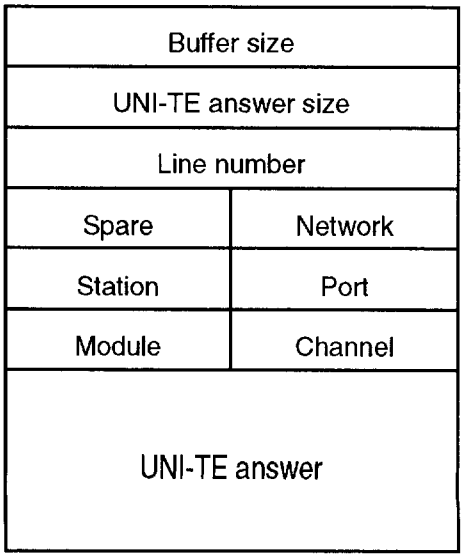
```
LDB     Source port          Store in accumulator B
LX@     Buffer address       Store in index register X
NETI                         Send the request
```

Where

Source port	Source port number (H'50' to H'5F' or only H'50' for a slave) same number as the port used to send the request
Buffer address	Address of the buffer to receive the answer.

Reception Buffer Structure

Buffer address →



Buffer size	1 word: the size in byte specified by the user does not include the word «buffer size»
Request size	1 word: request size in bytes
Line number	1 word: - H'20', H'21', H'24 to H'2B' (UNI-TELWAY lines) - H'30' (MAPWAY or ETHWAY) - H'40' (ETHERNET)
Server address	6 bytes: - Spare: not significant - Network: network number - Station: station number - Port: port number - Module: module number - Channel: channel number
UNI-TE answer	Data: bytes, words or long words (maximum 128 bytes) in the answer

Code Returned after the request

Answer code 1 byte: see table below

Answer code	Description
0	read OK
4	port number error
6	no request received on this port
7	buffer too small to contain the answer
8	line number not valid

REMARK *If the answer code is 6, the request must be repeated until the expected answer is received.*