

Protocol manual

JBUS/MODBUS

applicom[®] 3.7

a product of Woodhead Software & Electronics

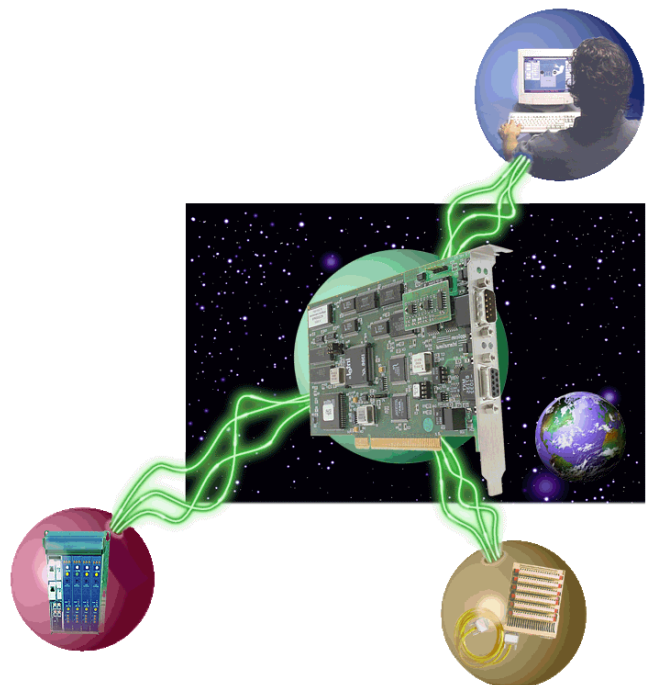


Table of contents

1.	- Generalities	1
	- Introduction	1
	- Definition of a JBUS / MODBUS exchange frame	2
2.	- Functionality	3
	- JBUS / MODBUS Master	3
	- JBUS / MODBUS slave	5
	- JBUS/MODBUS slave router	6
	- Cyclic JBUS / MODBUS Slave	8
	- JBUS/MODBUS slave gateway	11
3.	- Configuration	12
	- JBUS/MODBUS Master	12
	- JBUS/MODBUS slave	16
	- Cyclic JBUS/MODBUS slave	18
	- JBUS/MODBUS slave router	18
	- JBUS/MODBUS slave gateway	18
4.	- <i>applicom</i>® functions usable on the master channel	19
	- Wait mode	19
	- Deferred mode	19
	- Cyclic mode	20
5.	- Image variable item	21
	- Presentation	21
	- Standard descriptor	22
	- April-series 1000 descriptor	25
	- April-SMC descriptor	28
6.	- Appendices	31
	- List of extra files for this protocol	31
	- Evolution / compatibility	31
7.	- Return status of <i>applicom</i>® functions	32
	- Introduction	32
	- <i>applicom</i> ® general statuses	32
	- Statuses according to the protocol	34
8.	- Glossary of terms	36
9.	- Index	38

1. - Generalities

- Introduction

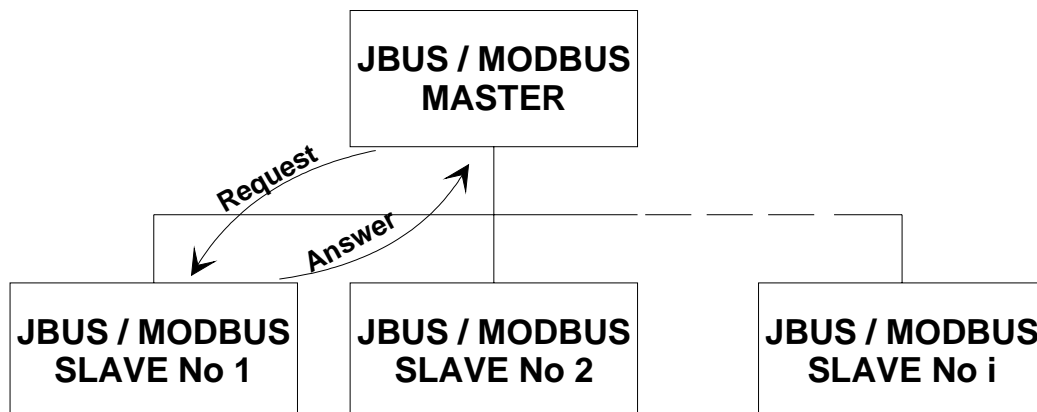
The JBUS (manufacturer: April) and MODBUS (manufacturer: Modicon) protocols are communication protocols which create a hierarchical structure. (One master and several slaves).

The **applicom**® product under JBUS / MODBUS protocols communicates in RTU (Remote Terminal Unit) mode. This type of protocol can be used to consult from the master one or several intelligent slaves. A multipoint link connects master and slave together.

Two types of communication are possible between master and slave:

- The master talks to a slave and waits for its response.
- The master talks to all the slaves without waiting for a response (general transmission principle).

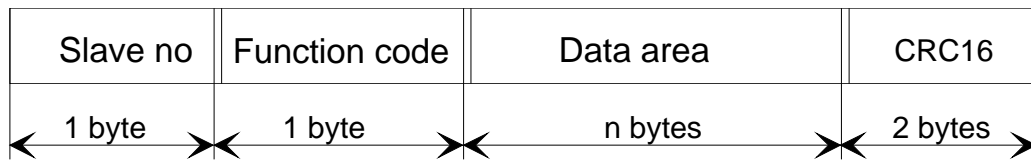
The master manages the exchanges and is the only one that has the initiative. This master repeats the question in case of incorrect exchange and declares the slave to be missing if there is no response in a given time-out. Only one device at a time can be transmitting on the line. No slave can send a message without being prompted to do so. Side communication (slave to slave) can only exist if the master software was designed to receive data and send it back from one slave to the other.



The master can address 255 slaves numbered slave n° 1 to slave n° 255. If the master uses slave number 0, this corresponds to a transmission towards all the slaves (write only).

JBUS and MODBUS protocols can be used to access devices connected on the same cable (the **applicom**® product makes it possible to disregard this constraint, see chapter JBUS/MODBUS SLAVE ROUTER).

- Definition of a JBUS / MODBUS exchange frame



Slave n° : 0 to 255 (0 = general transmission for write functions only).
 Function code : Indicates the type of exchange: 3 for read n words, 16 for write n words, for example.
 Data area : Information field holding the parameters related to the function: word address, word value, number of words, etc.
 CRC16 : Word used to detect transmission errors (Cyclic Redundancy Check).

2. - Functionality

- JBUS / MODBUS Master

The functions managed by the JBUS Master or MODBUS Master selection are identical. They correspond to the 12 functions available on the JBUS protocol.

List of supported JBUS / MODBUS functions	Corresponding <i>applicom</i> ® functions (library/DLL access)
01: Read n bits	READPACKBIT, READDIFBIT ...
02: Read n input bits	READPACKIBIT, READDIFIBIT ...
03: Read n words **	READWORD, READDIFWORD ... (if n° > 1) READDWORD, READDIFDWORD ... READFWORD, READDIFFWORD ...
04: Read n input words	READIWORD, READDIFIWORD ...
05: Write bit	WRITEPACKBIT, WRITEDIFPACKBIT ... (if n° = 1)
06: Write word	WRITEWORD, WRITEDIFWORD ... (if n° = 1)
*07: Quick read 8 bits	READQUICKBIT, READDIFQUICKBIT ...
*08: Exchange diagnostic	READDIAG
*11: Read event counter	READEVEN
*12: Read trace buffer	READTRACE
*13: Program controls - Under functions 01: Stop central processor 02: Start central processor 03: Unload memory 04: Load memory 25: Connect to PLC memory 26: Disconnect PLC memory	Only generated in JBUS if an APRIL PLC series 1000 is configured. MANUAL AUTOMATIC READWORD (10000(hex) + address) WRITEWORD (10000(hex) + address) AppConnect AppUnconnect
*14: Program control diagnostic	Automatically generated by <i>applicom</i> ® if function 13 is not run correctly.
15: Write n bits	WRITEPACKBIT, WRITEDIFPACKBIT ... (if n° > 1)
16: Write n words **	WRITEWORD, WRITEDIFWORD ... (if n° > 1) WRITEDWORD, WRITEDIFDWORD ... WRITEFWORD, WRITEDIFFWORD ...

* : Supported only by the Jbus protocol.

** : This function code is also used to access the "double word" format (READDWORD,...) and "floating word " format (READFWORD,...).

- Maximum number of variables per frame with the library

Objet	Maximum quantity in read	Maximum quantity in write
Bit	2000	2000
Word	125	125
Double word	62	62
Floating word	62	62

General transmission in write is managed with slave n° = 0.

- Maximum number of variables per frame with PCDDE

Objet	Maximum quantity in read	Maximum quantity in write
Bit	2000	1
Word	125	1
Double word	62	1
Floating word	62	1

General transmission in write is managed with slave n° = 0.

The number given for read frames corresponds to the maximum number of points (as imposed by the server and/or the protocol) which can be grouped together during dynamic optimization of the frames carried out by the server. However, this number can be reduced to suit a specific item of equipment by configuring the length of frames in the topic (see chapter "Implementation/Topics configuration/Advanced options").

Where write operations are concerned, a variable automatically entails the formation of a frame.

- Maximum number of variables per frame with OPC server

Objet	Maximum quantity in read	Maximum quantity in write
Bit	2000	2000
Word	125	125
Double word	62	62
Floating word	62	62

General transmission in write is managed with slave n° = 0.

The number given for read frames corresponds to the maximum number of points (as imposed by the server and/or the protocol) which can be grouped together during dynamic optimization of the frames carried out by the server. However, this number can be reduced to suit a specific item of equipment by configuring the length of frames in the topic (see chapter "Implementation/Topics configuration/Advanced options").

For the write frames, see chapter "OPC Server/Optimization of synchronous and asynchronous requests".

- JBUS / MODBUS slave

An interface channel configured as JBUS/MODBUS slave behaves exactly like a standard JBUS/MODBUS slave by making a 32 kbit and 32 kword database available to the external master. This database is shared by all channels on the same **applicom®** interface.

List of supported JBUS / MODBUS functions	
01 or 02: Read n bits	
03 or 04: Read n words	
05: Write 1 bit	
06: Write 1 word	
07: Quick read 8 bits	
08: Exchange diagnostic	
11: Read event counter	
12: Read trace buffer	
13: Program control	
- Under functions	
03: Address from 0 to 7FFF = Read words	
10000(hex) + address = Read double words (INTEL format)	
20000(hex) + address = Read floating words (INTEL format)	
04: Address from 0 to 7FFF = Write words	
10000(hex) + address = Write double words (INTEL format)	
20000(hex) + address = Write floating words (INTEL format)	
15: Write n bits	
16: Write n words	

Object	Maximum quantity
Bit	2048
Word	128
Double word	64
Floating word	64

Error code sent to the JBUS / MODBUS master

- 1: Unknown function code.
Received function different from that listed above.
- 2: Wrong address.
Address of solicited bit or word variables above 32767 or below 0.
- 3: Wrong data.
- 4: Interface not ready.
Data area locked in the database (see chapter "Data-Base access lock" in the manual "Functionality").

The master can also lock some database areas for read and/or write only (see chapter "Data-Base access lock" in the manual "Functionality").

Note:

Access to double and floating words using function code 13 leads to processing of the appropriate type in the database, and therefore in the event FIFO under Windows DDE.

- JBUS/MODBUS slave router

An interface channel configured as "JBUS/MODBUS Slave Router" provides the same functionality as a channel configured in "JBUS/MODBUS Slave".

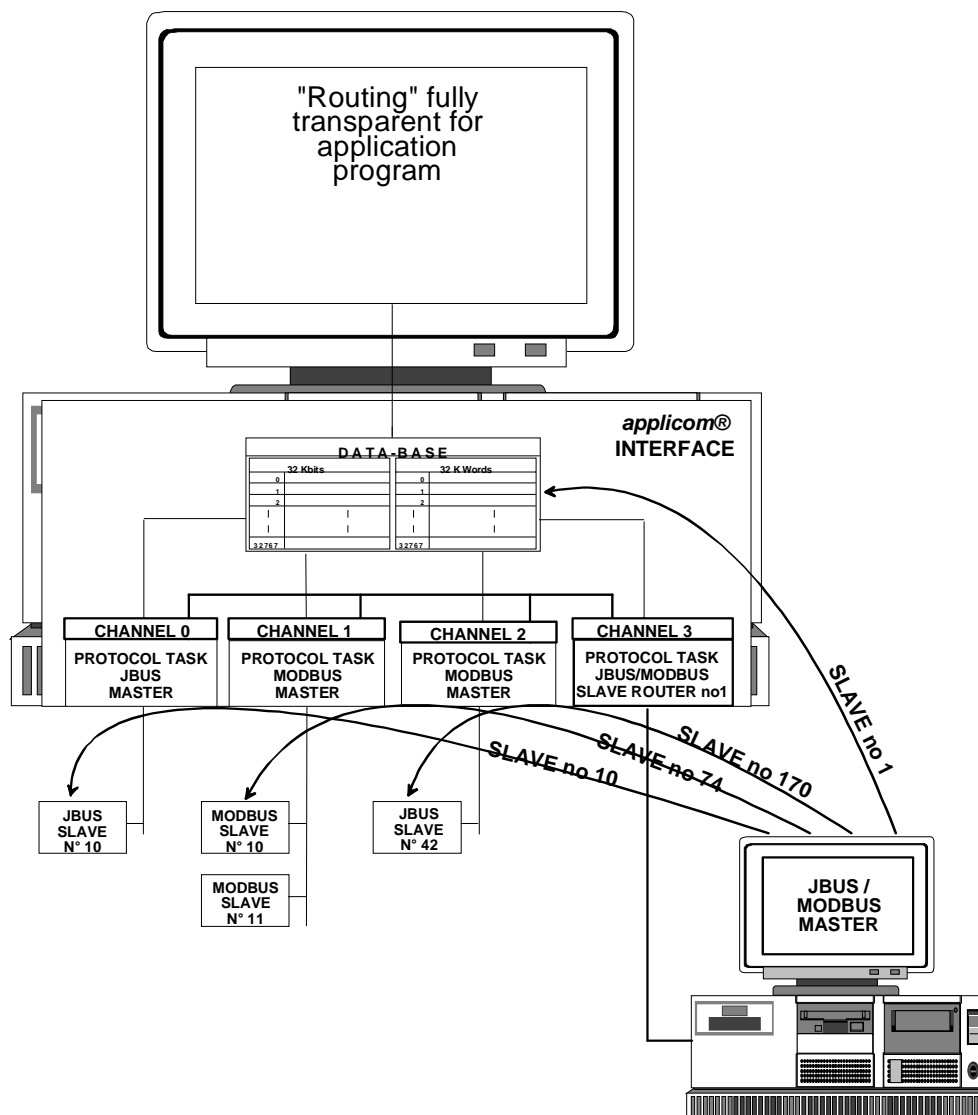
This mode also allows the JBUS/MODBUS master, connected to the **applicom®** interface, to access all data belonging to the slaves connected to the other JBUS/MODBUS master channels of the interface.

The "JBUS/MODBUS Slave Router" channel performs a frame redirection mechanism that depends on the slave number generated by the external master.

The output channel number as well as the destination slave number are coded in the slave number byte according to the principle below :

- the output channel number (master imperatively) is coded in binary 0 to 3 on the 2 most significant bits of the JBUS/MODBUS slave number byte.
- the destination slave number is defined on the 6 least significant bits 0 to 63.

Example:



The external master applies to slave n° 10 of the **applicom**® server channel n° 1. It will therefore have to aim at slave $64+(8+2)=74$.

Slave number byte								
	Channel n° 2 bits				slave n° 6 bits			
bits	0	1	0	0	1	0	1	0
signif.	128	64	32	16	8	4	2	1

Slave 10 of channel 0	: slave n° = (in binary)	00001010 = (in decimal)	10
Slave 10 of channel 1	: slave n° =	01001010 =	74
Slave 42 of channel 3	: slave n° =	11101010 =	170
Slave DATA-BASE	: slave n° = defined at configuration =		1

As JBUS/MODBUS frames are transmitted again, a higher time-out value should be defined (double the value) in the external master.

- Cyclic JBUS / MODBUS Slave

- Presentation

The functionality can be used to optimize architecture based on the concentrator principle. In fact, for read and write accesses, the master sees only one slave.

All accessed variables must be preset by cyclic read functions.

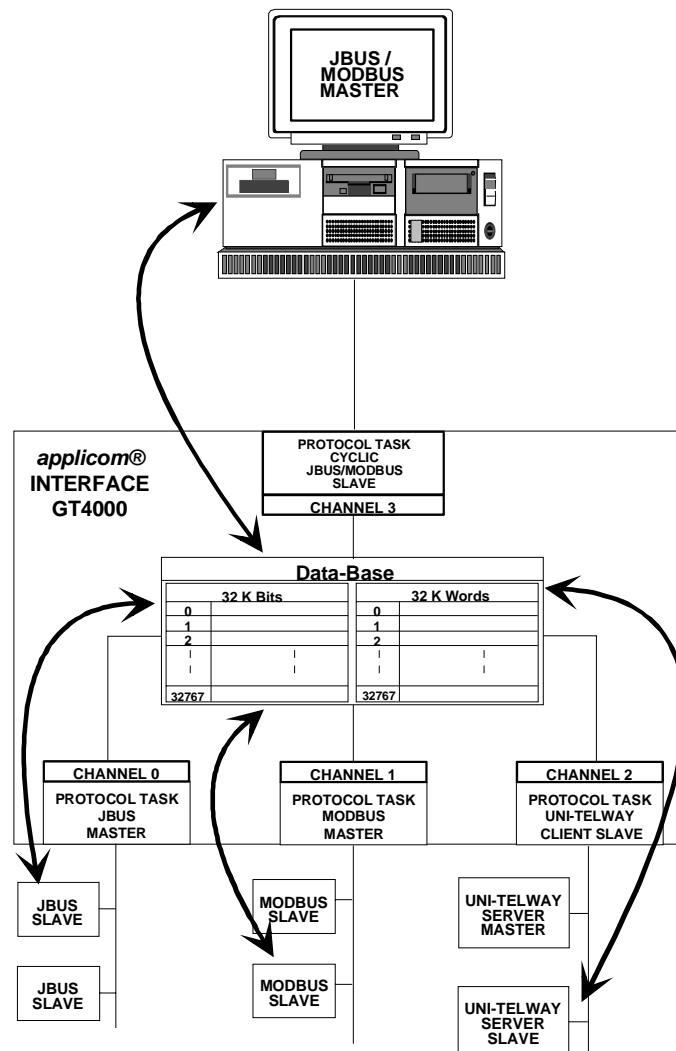
Exchanges are asynchronous for read (via Data-Base).

Exchanges are synchronous for write (directly in the equipment).

Example :

- The **applicom**® interface retrieves all the variables to its database via cyclic functions by using up to 3 master or client channels.

- The external master only queries the Data-Base and therefore has the comfortable advantage of seeing only one slave.



- Principle

The CYCLIC JBUS/MODBUS slave channel will be seen by the master as a single slave whose number is configurable.

Variables not refreshed by a cyclic read function will be called **local variables**.

Variables refreshed by a cyclic read function will be called **image variables**.

This management is ensured :

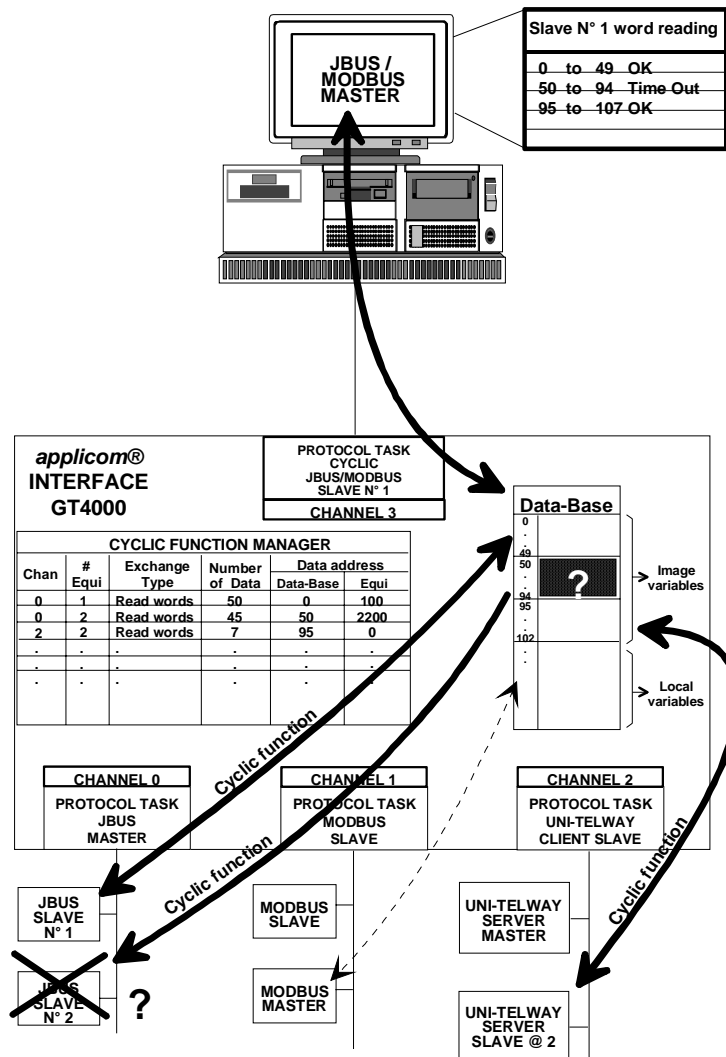
- On the first 10 000 words and bits of the database.
- If the variables are refreshed by a cyclic read function (image variables).
- On the cyclic functions configured at system initialization.

- Processing of JBUS/MODBUS read functions

All the JBUS/MODBUS read functions return data read in the Data-Base. If, in the case of image variables, they are refreshed by cyclic functions that do not end correctly (status \neq 0) the channel behavior on the accessed area is the function status image.

Example :

Cyclic function in Time-Out = No response from the " Cyclic Jbus/Modbus slave " channel.



- Processing of JBUS/MODBUS write functions

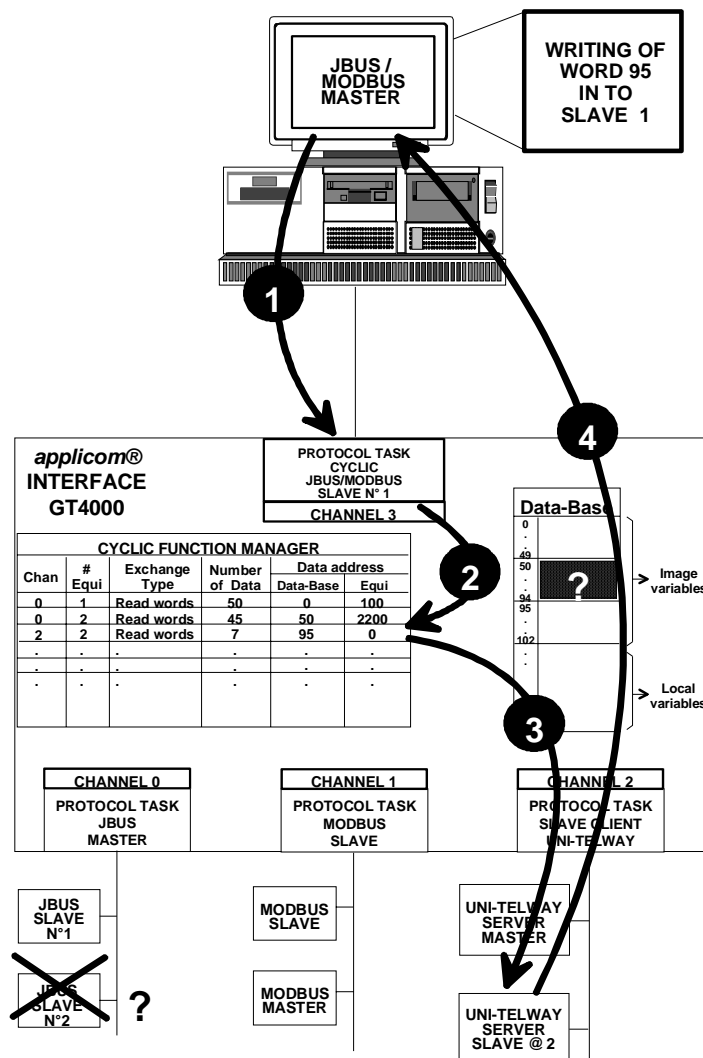
During a write function, the "Cyclic Jbus/Modbus slave" will direct the remote control:

- To the Data-Base in the case of local variables
- Directly to the equipment in the case of image variables

In the case of image variables, the remote control routing is totally transparent for the external Jbus/Modbus master.

The **applicom®** interface will carry out this routing through the following phases :

- 1: Write function reception.
- 2: Analysis of configured cyclic functions and extraction of the addressee parameters
 Channel n° .
 Equipment n° .
 Variable address in the equipment.
 Remote control run.
 Activation of the cyclic read function to update the Data-Base.
- 3: Return of the remote control status to the external Jbus/Modbus master.



Note : The external master can access a refreshed area via several cyclic functions. The response will only be correct if all the cyclic functions concerned end with status = 0.

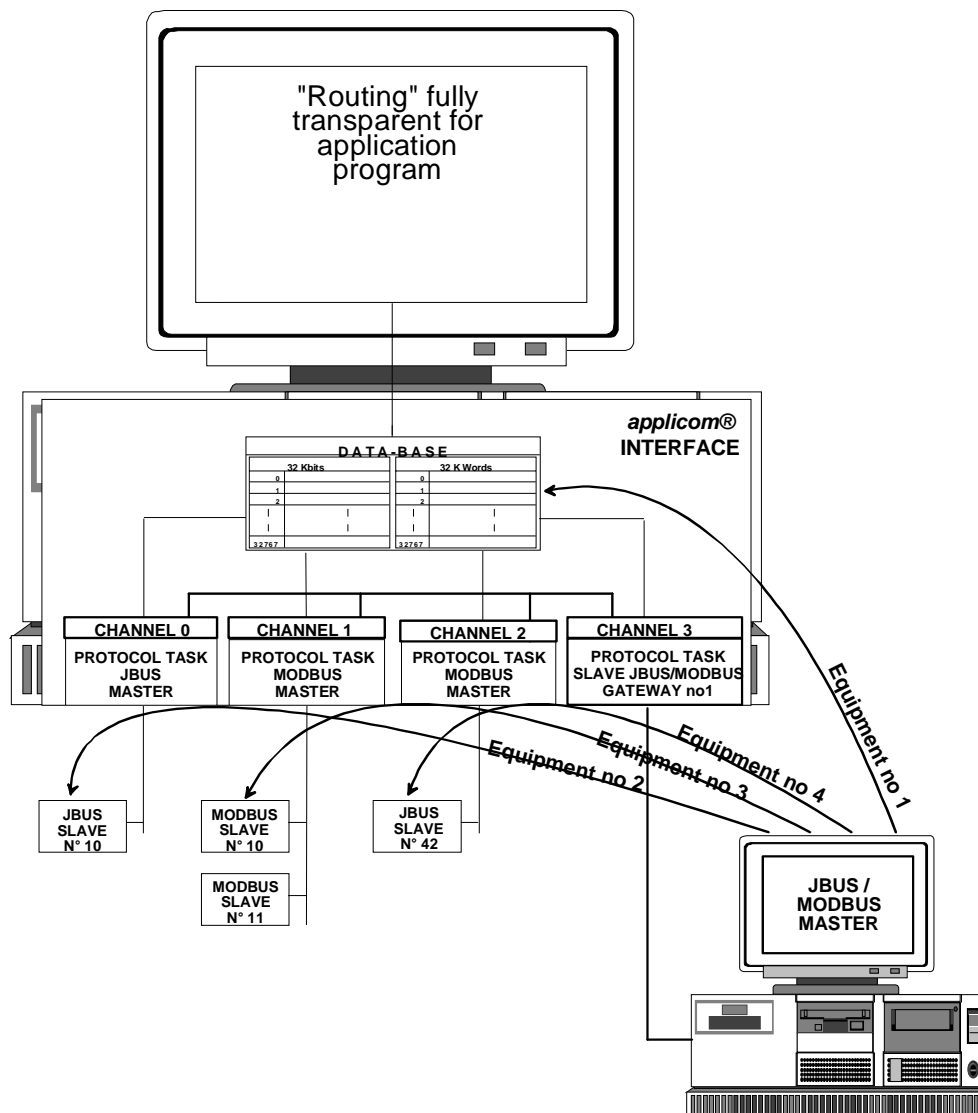
- JBUS/MODBUS slave gateway

An interface channel configured as "JBUS/MODBUS slave gateway" provides the same functionality as a channel configured in "JBUS/MODBUS Slave".

This mode also allows the JBUS/MODBUS master, connected to the **applicom®** interface, to access all data belonging to the slaves connected to the other JBUS/MODBUS master channels of the interface.

The "JBUS/MODBUS slave gateway" channel performs a frame redirection mechanism that depends on the slave number generated by the external master.

During the configuration, the output channel number and the destination slave number are associated with a slave number of the task "JBUS/MODBUS slave gateway".



The external master applies to slave n° 10 of the **applicom®** server channel n° 1. It will therefore have to aim at slave 2. This number being defined during the configuration.

Slave 10 of channel 0 : slave n° = 2
 Slave 42 of channel 3 : slave n° = 4

Slave 10 of channel 1 : slave n° = 3
 Slave DATA-BASE : slave n° = 1

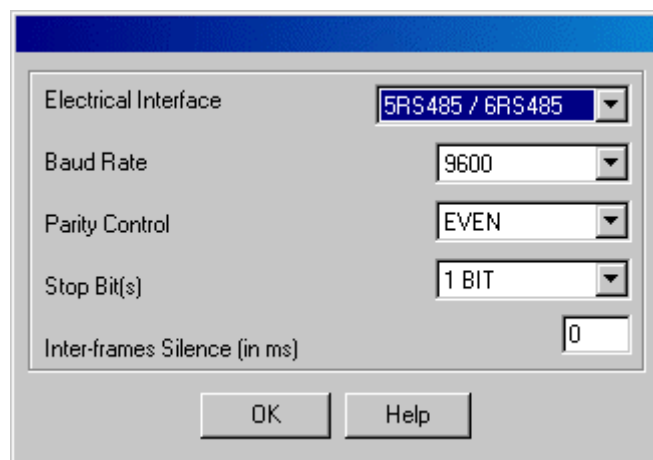
As JBUS/MODBUS frames are transmitted again, a higher time-out value should be defined (double the value) in the external master.

3. - Configuration

- JBUS/MODBUS Master

- Configuration screen

The JBUS and MODBUS protocol parameters are more or less the same. You must declare the various target slaves in your exchanges.



⇒ Electrical interface:

The interface module type is determined by the reference of the module installed on the **applicom®** interface or on the galvanic isolation option (BX4010, BX4010R).

You have then the following options:

RS232 (2 signals) (3RS232*, 5RS232*, 6RS232)

RS232 (6 signals) (3RS232*, 5RS232*, 6RS232)

BC20 mA*

3RS485*

3RS485-4*

5RS485* or 6RS485

* : Not galvanically isolated. Isolation only ensured with option BX4010 and BX4010R.

For the module compatibility, you can refer to the Wiring/Output module compatibility section of « hardware installation » manual.

RS232 (2 signals)(3RS232, 5RS232, 6RS232)

Means that only the TxD and RxD signals are exploited by the protocol task installed on the channel. RTS, DTR, CTS and DCD signals can be accessed from the application by using the **applicom®** library GETMODEM and SETMODEM functions.

RS232 (6 signals)(3RS232, 5RS232, 6RS232)

Means that the protocol task installed at the channel level manages the signals.

- TxD and RxD for data transmission and reception.
- RTS, an output signal positioned by the **applicom®** interface before each transmission.
- CTS, an input signal, must be active to allow transmission.
- DCD, an input signal, must be active to validate reception.

You can position the DTR output signal from the applicative program by using the **applicom®** library SETMODEM function.

You can query the CTS and DCD input signal status from the application program by using the **applicom®** library GETMODEM function.

BC20 mA

Conductive 20 mA current loop. The active or passive operation is determined by wiring. If active, see "5BC20 module characteristics" in wiring section of the Hardware installation manual. The active operation should only be used if BX4010 and BX4010R boxes are used.

3RS485

2-wire RS485 link. The line prepolarization as well as the load resistance are wiring-selectable.

Caution: Wiring is different from modules 3RS485-4 and 5RS485.

3RS485-4

4-prepolarized wire RS422 / RS485 link. The load resistance is wiring-selectable.

5RS485 or 6RS485

2 or 4-prepolarized wire RS422 / RS485 link. Load resistances are wiring-selectable. In addition, the 6RS485 module ensures the galvanic isolation. Your interface must be equipped for this type of PROM module V3.0 or higher.

⇒ Baud rate

You must select a transmission speed comprised between **50 and 38400 bauds**, independently on each channel. This speed must be consistent with that of the equipments connected on the channel.

⇒ Parity control

Addition or not of a parity bit. You must select the type of parity check.

- None
- Odd
- Even

⇒ Stop bit(s)

You must select the number of stop bits used for delimiting a character. You must choose between **1 or 2 stop bits**.

This parameter must be consistent with that of the equipments connected on the channel.

⇒ Inter-frames Silence

This is the time added to the silence corresponding to the three detection characters at frame end (in milliseconds). **This value must always equal 0** unless one of the slaves does not rigorously respect the JBUS/MODBUS protocols.

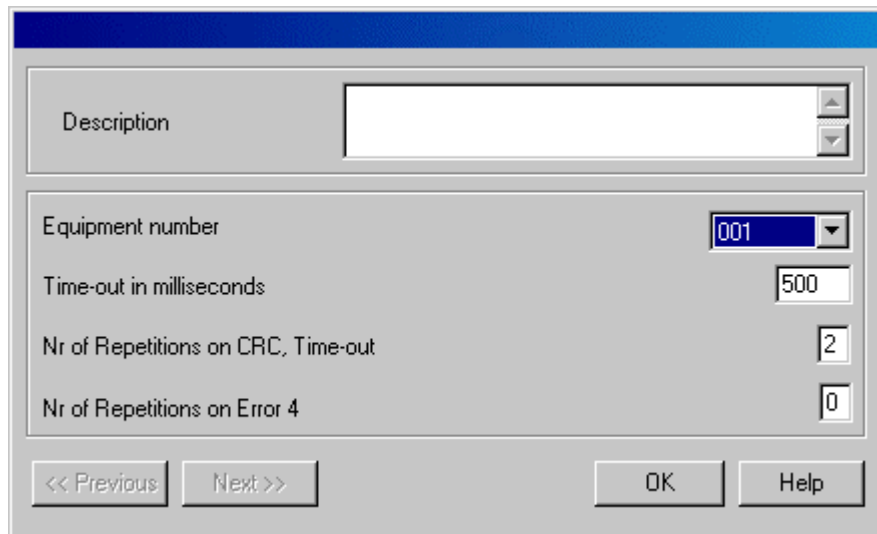
- Slaves configuration

For each slave, you must define :

⇒ Slave number :

The JBUS and MODBUS protocols do not include any addressing mechanism for non-adjacent nodes (no network layer): a slave is identified by a number (1 to 255). For further simplification, the **applicom®** equipment number is implicitly equal to the JBUS/MODBUS slave number.

You must enter a slave number (1 to 255). This number is the one that will be recognized by the JBUS/MODBUS master.



For each JBUS slave :

From the list of available equipment, select the device corresponding to the slave. If the slave is not in the list, select the device "UNDEFINED".

The protocol task will then be able to process access functions in the program area of some PLCs (APRIL series 1000).

Note: For handling "Double word" or "Floating word" variables, the "most significant" "least significant" format can be inverted depending on the "slave type ":

W x Address	W x+1 Address	Choice : Slave type
Most significant	Least significant	Undefined (or other)
Least significant	Most significant	applicom®

For each MODBUS slave :

Simply declare the equipment as "present".

⇒ Time-out

This is the maximum time between the master sending a function and the slave's response (from 10 to 60000 ms).

A slave always has different characteristics, especially a specific reaction time: this is why the time-out value must be set.

When the communication server is run, we recommend that the time-out value is set to 1000 ms minimum. This time-out value must always be higher than the inter-character time-out value.

⇒ **Number of Repetitions on CRC, Time-out**

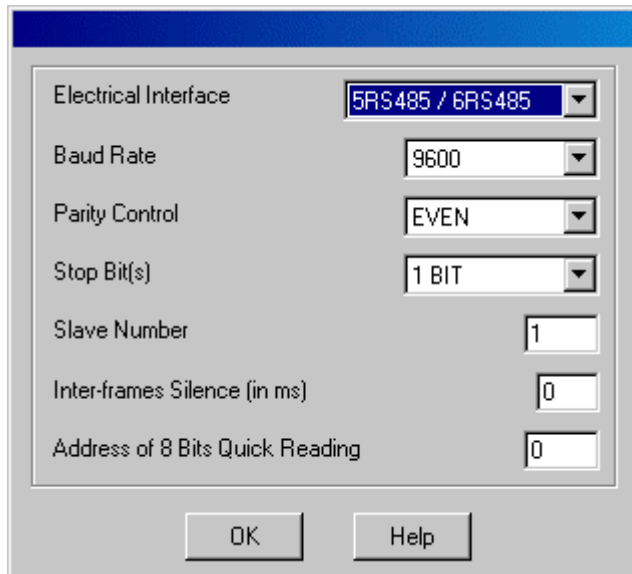
Determines the number of repetitions (**0 to 9**) if there is no response from the slave (**applicom**® Status = 33) or parity or CRC error in reception (**applicom**® Status = 34).

To privilege the communication with slaves present, the **applicom**® master manages no more repetitions after 10 frames ending with Time-Out on the same slave. Repetition management will restart automatically from the slave's first response.

⇒ **Number of Repetitions on Error 4**

Determines the number of repetitions (**0 to 9**) if the response from the slave is "data not ready". The frequency of these repetitions has a preset value of 50 ms.

- JBUS/MODBUS slave



⇒ Electrical interface :

The interface module type is determined by the reference of the module installed on the **applicom®** interface or on the galvanic isolation option (BX4010, BX4010R).

You have then the following options:

RS232 (2 signals) (3RS232*, 5RS232*, 6RS232)

RS232 (6 signals) (3RS232*, 5RS232*, 6RS232)

BC20 mA*

3RS485*

3RS485-4*

5RS485* or 6RS485

* : **Not galvanically isolated. Isolation only ensured with option BX4010 and BX4010R.**

For the module compatibility, you can refer to the Wiring/Output module compatibility section of « hardware installation » manual.

RS232 (2 signals)(3RS232*, 5RS232*, 6RS232)

Means that only the TxD and RxD signals are exploited by the protocol task installed on the channel. RTS, DTR, CTS and DCD signals can be accessed from the application by using the **applicom®** library GETMODEM and SETMODEM functions.

RS232 (6 signals)(3RS232, 5RS232, 6RS232)

Means that the protocol task installed at the channel level manages the signals.

- TxD and RxD for data transmission and reception.
- RTS, an output signal positioned by the **applicom®** interface before each transmission.
- CTS, an input signal, must be active to allow transmission.
- DCD, an input signal, must be active to validate reception.

You can position the DTR output signal from the applicative program by using the **applicom®** library SETMODEM function.

You can query the CTS and DCD input signal status from the application program by using the **applicom®** library GETMODEM function.

BC20 mA

Conductive 20 mA current loop. The active or passive operation is determined by wiring. If active, see "5BC20 module characteristics" in wiring section of the Hardware installation manual. The active operation should only be used if BX4010 and BX4010R boxes are used.

3RS485

2-wire RS485 link. The line prepolarization as well as the load resistance are wiring-selectable.

3RS485-4

4-prepolarized wire RS422 / RS485 link. The load resistance is wiring-selectable.

5RS485 or 6RS485

2 or 4-prepolarized wire RS422 / RS485 link. Load resistances are wiring-selectable. In addition, the 6RS485 module ensures the galvanic isolation. Your interface must be equipped for this type of PROM module V3.0 or higher.

⇒ Baud Rate

You must select a transmission speed comprised between **50 and 38400 bauds**, independently on each channel. This speed must be consistent with that of the equipments connected on the channel.

⇒ Parity Control

Addition or not of a parity bit. You must select the type of parity check.

- None
- Odd
- Even

⇒ Stop Bit(s)

You must select the number of stop bits used for delimiting a character. You must choose between **1 or 2 stop bits**.

This parameter must be consistent with that of the equipments connected on the channel.

⇒ Slave Number

JBUS/MODBUS slave number assigned to the **applicom®** interface channel. This slave number, **between 1 and 255**, will allow the external master to access the **applicom®** interface database.

⇒ Inter-frames Silence

This is the time added to the silence corresponding to the three detection characters at frame end (in milliseconds). **This value must always equal 0** unless the master does not rigorously respect the JBUS/MODBUS protocols.

⇒ Address of 8-bit Quick Reading

You can declare an address of the 8-bit quick reading in the **applicom®** interface database. This is a specificity of these protocols which can be used for quick access to an area of 8 contiguous bits (shorter frame). If you are not interested in this operating mode, leave the default value (0).

- Cyclic JBUS/MODBUS slave

Same configuration as the JBUS/MODBUS slave.

- JBUS/MODBUS slave router

Same configuration as the JBUS/MODBUS slave.

As JBUS/MODBUS frames are retransmitted, a higher time-out value should be defined (double the value) in the external master.

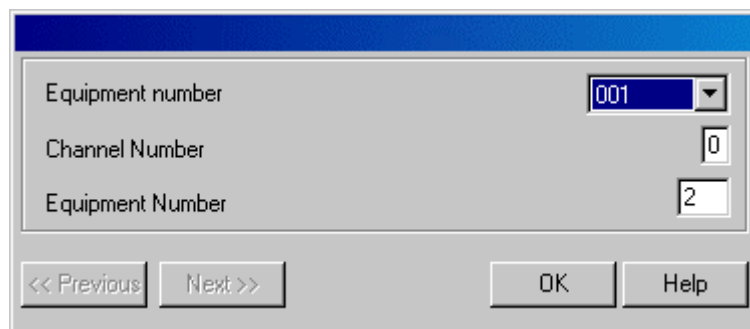
- JBUS/MODBUS slave gateway

Same configuration as the JBUS/MODBUS slave. A button allows gateway equipments configuration.

As JBUS/MODBUS frames are retransmitted, a higher time-out value should be defined (double the value) in the external master.

⇒ Gateway equipments configuration

Indicate for each slave number of the task "JBUS/MODBUS slave gateway" the output channel number and the number of destination equipment bound.



The screenshot shows a configuration dialog box with a blue title bar. It contains three input fields: 'Equipment number' with a dropdown menu showing '001', 'Channel Number' with a text box showing '0', and 'Equipment Number' with a text box showing '2'. At the bottom, there are four buttons: '<< Previous', 'Next >>', 'OK', and 'Help'.

4. - *applicom*® functions usable on the master channel

- Wait mode

- Standard functions

readpackbit	writepackbit
readpackibit	
readword	writeword
readiword	
readwordbcd	writewordbcd
readdword	writedword
readfword	writefword
readmes	writemes
AppConnect	AppUnconnect
manual	automatic

- Specific functions

iocounter	resetiocounter
readdiag	
readeven	
readtrace	
readquickbit	
statjbus	

- Deferred mode

readdifbit	writedifpackbit	
readdifibit		
readdifword	writedifword	
readdifiword		
readdifdword	writedifdword	
readdiffword	writediffword	
readdifquickbit		
readdifmes	writedifmes	
testtransdif	transdif	transdifpack

- Cyclic mode

createcyc
startcyc **stopcyc** **actcyc**
transcyc **transcycpack**

Cyclic function Type:

TYPE OF VARIABLE IN EQUIPMENT	Function type	
	Reading	Writing
Packed bits	X	X
Input packed bits	X	
Quick reading of 8 bits	X	
Words	X	X
BCD words	X	X
Input words	X	
32 bit double words	X	X
32 bit IEEE floating words	X	X

5. - Image variable item

- Presentation

The "item of image variables" are the syntaxes which allow to access to the variables through the DDE server "pcdde" or the OPC server.

Report you to sections "DDE server/Principles regarding access to Data" or "OPC server/Data Access Principle".

According to the configuration of the equipment, the descriptor is determined by default by the server.

In case of MODBUS protocol, the default descriptor is the standard descriptor.

In case of JBUS protocol, the default descriptor is:

- April SMC descriptor for the SMC equipment,
- April serie 1000 descriptor for the APRIL equipment,
- **applicom®** standard descriptor for the other equipments.

You however have the possibility of using another descriptor (in particular the **applicom®** standard descriptor) by the means of the advanced options.

- Standard descriptor

The standard descriptor can be used to access all equipment which does not have a specific descriptor. The address field in the « item » name can go up to 10 digits. A linear address can therefore be composed ranging from 0 to 4 giga.

This descriptor is the default descriptor determined by the OPC and DDE server in these cases:

- MODBUS protocol
- JBUS protocol with different equipment than SMC and APRIL serie 1000 equipment.

	Unitary mode	Table Mode, Matrix Mode
Internal bits	Bx	Bx_n, Bx_n_l
Input bits	Blx	Blx_n, Blx_n_l
Output bits	BOx	BOx_n, BOx_n_l
Internal words	Wx	Wx_n, Wx_n_l
Bits in the internal words	Wx.b	
Input words	Wlx	Wlx_n, Wlx_n_l
ASCII string in the internal words		M_Wx_n
Internal double words	Dx	Dx_n, Dx_n_l
Internal floating words	Fx	Fx_n, Fx_n_l

Remark: Limits values for n and l parameters are depending on the protocol. However, in case of PCDDE, limits cannot never be superior than 128 for bits and bytes, 64 for words, 32 for double words and floating words.

For variable addressing, refer to the section "- JBUS / MODBUS Master" on page 3

To consult the read and write limits, refer to the sections:

"- Maximum number of variables per frame with PCDDE" on page 4

"- Maximum number of variables per frame with OPC" on page 4

- Internal bits => Bx (type: BIT)

x: Number of the first bit.

Example: B4

- Internal bits => Bx_n, Bx_n_l

n: Number of bits.

l: Number of bits per line (Matrix mode only)

Examples: B4_10, B4_10_5

- Input bits => Blx (type: BIT)

x: Number of the first bit.

Example: Bl4

- Input bits => Blx_n, Blx_n_l

n: Number of bits.

l: Number of bits per line (Matrix mode only).

Examples: BI4_10, BI4_10_5

- Output bits => BOx (type: BIT)

x: Number of the first bit.

Example: BO4

- Output bits => BOx_n, BOx_n_l

n: Number of bits.

l: Number of bits per line (Matrix mode only).

Examples: BO4_10, BO4_10_5

- Internal words => Wx (type: 16 bit WORD)

x: Number of the first word.

Example: W4

- Internal words => Wx_n, Wx_n_l

n: Number of words.

l: Number of words per line (Matrix mode only).

Examples: W4_10, W4_10_5

- Bits in the internal words => Wx.b

For this syntax, the maximum numbers of variables per frame are

- in read: 125*16

- in write: 1

x: Number of the first word.

b: Rank of the bit in the word (0 to 15).

Example: W4.5

- Input words => Wlx (type: 16 bit WORD)

x: Number of the first word.

Example: Wl4

- Input words => Wlx_n, Wlx_n_l

n: Number of words.

l: Number of words per line (Matrix mode only).

Examples: WI4_10, WI4_10_5

- ASCII string in the internal words => M_Wx_n

For more information on the use of the ASCII string, you can consult the chapter "Use of message mode".

x: Number of the first word containing the string

n: Number of potential words that could contain the ASCII string (1 to 65).

Example: M_W100_30

In the example, the word array W100 to W129 can contain the string.

- Internal double words => Dx (type: 32 bit WORD)

x: Number of the first double word.

Example: D4

- Internal double words => Dx_n, Dx_n_l

n: Number of double words.

l: Number of double words per line (Matrix mode only).

Examples: D4_10, D4_10_5

- Internal floating words => Fx (type: 32 bit IEEE REAL)

x: Number of the first floating word.

Example: F4

- Internal floating words => Fx_n, Fx_n_l

n: Number of floating words.

l: Number of floating words per line (Matrix mode only).

Examples: F4_10, F4_10_5

- April-series 1000 descriptor

This descriptor can only be used when accessing April-series 1000 PLCs.

This descriptor is the default descriptor determined by the OPC and DDE server for APRIL equipment.

	Unitary mode	Table Mode, Matrix Mode
Internal bits not saved	%MXx	%MXx_n, %MXx_n_l
Internal bits saved	%RXx	%RXx_n, %RXx_n_l
Input bits	%IXwxyz	%IXwxyz_n, %IXwxyz_n_l
Output bits	%QXwxyz	%QXwxyz_n, %QXwxyz_n_l
Data words	%MWx	%MWx_n, %MWx_n_l
Bits in the internal words	%MWx.b	
Double words in the data words	%MDx	%MDx_n, %MDx_n_l
Floating words in the data words	%FDx	%FDx_n, %FDx_n_l
ASCII string in the data words		M_MWx_n

Remark: Limits values for n and l parameters are depending on the protocol. However, in case of PCDDE, limits cannot never be superior than 128 for bits and bytes, 64 for words, 32 for double words and floating words.

For variable addressing, refer to the section "- JBUS / MODBUS Master" on page 3

To consult the read and write limits, refer to the sections:

"- Maximum number of variables per frame with PCDDE" on page 4

"- Maximum number of variables per frame with OPC" on page 4

- Internal bits not saved => %MXx (type: BIT)

x: Number of the first bit (0 to 16383 - 4095 max for April 5000).

Example: %MX4

- Internal bits not saved => %MXx_n, %MXx_n_l

n: Number of bits.

l: Number of bits per line (Matrix mode only).

Examples: %MX4_10, %MX4_10_5

- Internal bits saved => %RXx (type: BIT)

x: Number of the first bit (0 to 8191 ; 2047max for April 5000).

Example: %RX4

- Internal bits saved => %RXx_n, %RXx_n_l

n: Number of bits.

l: Number of bits per line (Matrix mode only).

Examples: %RX4_10, %RX4_10_5

- Input bits => %IXwxyz (type: BIT)

w: Channel number (0 to 9; 0 for an April 5000).
x: Rack number (0 to 6; 4 max for an April 5000).
y: Slot number (0 to 9).
z: Channel number on 2 digits (00 to 31)
Example: %IX26507

- Input bits => %IXwxyz_n, %IXwxyz_n_l

n: Number of bits.
l: Number of bits per line (Matrix mode only).
Examples: %IX26507_10, %IX26507_10_5

- Output bits => %QXwxyz (type: BIT)

w: Channel number (0 to 9; 0 for an April 5000).
x: Rack number (0 to 6; 4 max for an April 5000).
y: Slot number (0 to 9).
z: Channel number on 2 digits (00 to 31)
Example: %QX26507

- Output bits => %QXwxyz_n, %QXwxyz_n_l

n: Number of bits.
l: Number of bits per line (Matrix mode only).
Examples: %QX26507_10, %QX26507_10_5

- Data words => %MWx (type: 16 bit WORD)

x: Number of the first word (0 to 409599).
Example: %MW4

- Data words => %MWx_n, %MWx_n_l

n: Number of words.
l: Number of words per line (Matrix mode only).
Examples: %MW4_10, %MW4_10_5

- Bits in the internal words => %MWx.b (type: BIT)

For this syntax, the maximum numbers of variables per frame are:

- in read: 125*16
- in write: 1

x: Word number (0 to 409599).

b: Rank of the bit in the word (0 to 15).

Example: %MW4.5

- Double words in the data words => %MDx (type: 32 bit WORD)

x: Number of the first double word (0 to 409598).

Example: %MD4

- Double words in the data words => %MDx_n, %MDx_n_l

n: Number of double words.

l: Number of double words per line (Matrix mode only).

Examples: %MD4_10, %MD4_10_5

- Floating words in the data words => %FDx (type: 32 bit IEEE REAL)

x: Number of the first floating word (0 to 409598).

Example: %FD4

- Floating words in the data words => %FDx_n, %FDx_n_l

n: Number of floating words.

l: Number of floating words per line (Matrix mode only).

Examples: %FD4_10, %FD4_10_5

- ASCII string in the data words => M_MWx_n (type: 16 bit WORD)

For more information on the use of the ASCII string, you can consult the chapter "Use of message mode".

x: Number of the first word containing the string (0 to 409599).

n: Number of potential words that could contain the ASCII string (1 to 65).

Example: M_MW100_30

In the example the word array %MW100 to %MW129 can contain the string.

- April-SMC descriptor

The descriptor of **April-SMC** PLCs can only be used when accessing April-SMC PLCs.

This descriptor is the default descriptor determined by the OPC and DDE server for SMC equipment.

	Unitary mode	Table Mode, Matrix Mode
Internal bits (B Bits)	Bx	Bx_n, Bx_n_l
Internal bits (R Bits)	Rx	Rx_n, Rx_n_l
Input bits	Ex	Ex_n, Ex_n_l
Output bits	Ax	Ax_n, Ax_n_l
Data words	Mx	Mx_n, Mx_n_l
Bits in the data words	Mx.b	
Double words in the data words	MDx	MDx_n, MDx_n_l
Floating words in the data words	FDx	FDx_n, FDx_n_l
ASCII string in the data words		M_Mx_n

Remark: Limits values for n and l parameters are depending on the protocol. However, in case of PCDDE, limits cannot never be superior than 128 for bits and bytes, 64 for words, 32 for double words and floating words.

For variable addressing, refer to the section "- JBUS / MODBUS Master" on page 3

To consult the read and write limits, refer to the sections:

"- Maximum number of variables per frame with PCDDE" on page 4

"- Maximum number of variables per frame with OPC" on page 4

- Internal bits (B Bits) => Bx (type: BIT)

x: Bit number (0 to 1023).

Example: B4

Caution: There is a break between bit B511 and bit B512

- Internal bits (B Bits) => Bx_n, Bx_n_l

n: Number of bits.

l: Number of bits per line (Matrix mode only)

Examples: B4_10, B4_10_5

Caution: There is a break between bit B511 and bit B512

- Internal bits (R Bits) => Rx (type: BIT)

x: Bit number (0 to 1023).

Example: R4

Caution: There is a break between bit R511 and bit R512

- Internal bits (R Bits) => Rx_n, Rx_n_I

n: Number of bits.

I: Number of bits per line (Matrix mode only).

Examples: R4_10, R4_10_5

Caution: There is a break between bit R511 and bit R512

- Input bits => Ex (type: BIT)

x: Bit number (0 to 1023 and 2000 to 3023).

Example: E4

Caution: There is a break between bit E127 and bit E128, between E511 and E512, between E1023 and E2000, between E2127 and E2128 and between E2511 and E2512.

- Input bits => Ex_n, Ex_n_I

n: Number of bits.

I: Number of bits per line (Matrix mode only).

Examples: E4_10, E4_10_5

Caution: There is a break between bit E127 and bit E128, between E511 and E512, between E1023 and E2000, between E2127 and E2128 and between E2511 and E2512.

- Output bits => Ax (type: BIT)

x: Bit number (0 to 1023).

Example: A4

Caution: There is a break between bit A127 and bit A128 and between A511 and A512.

- Output bits => Ax_n, Ax_n_I

n: Number of bits.

I: Number of bits per line (Matrix mode only).

Examples: A4_10, A4_10_5

Caution: There is a break between bit A127 and bit A128 and between A511 and A512.

- Data words => Mx (type: 16 bit WORD)

x: Word number (0 to 13823).

Example: M4

- Data words => Mx_n, Mx_n_I

n: Number of words.

I: Number of words per line (Matrix mode only).

Examples: M4_10, M4_10_5

- Bits in the data words => Mx.b (type: BIT)

For this syntax, the maximum numbers of variables per frame are:

- in read: 125*16

- in write: 1

x: Word number (0 to 13823).

b : Rank of the bit in the word (0 to 15).

Example: M4.5

- Double words in the data words => MDx (type: 32 bit WORD)

x: Number of the double word (0 to 13822).

Example: MD4

- Double words in the data words => MDx_n, MDx_n_I

n: Number of double words.

I: Number of double words per line (Matrix mode only).

Examples: MD4_10, MD4_10_5

- Floating words in the data words => FDx (type: 32 bit IEEE REAL)

x: Number of the double word (0 to 13822).

Example: FD4

- Floating words in the data words => FDx_n, FDx_n_I

n: Number of floating words.

I: Number of floating words per line (Matrix mode only).

Examples: FD4_10, FD4_10_5

- ASCII string in the data words => M_Mx_n (type: 16 bit WORD)

For more information on the use of the ASCII string, you can consult the chapter "Use of message mode".

x: Number of the first word containing the string (0 to 13823).

n: Number of potential words that could contain the ASCII string (1 to 65).

Example: M_M100_30

In the example the word array M100 to M129 can contain the string.

6. - Appendices

- List of extra files for this protocol

MASTER.	Jbus/Modbus Master Task
ROUTERSL.	Jbus/Modbus Slave router task
SLAVE.	Jbus/Modbus Slave Task
SLJBCYC.	Jbus/Modbus cyclic slave task

- Evolution / compatibility

This protocol necessitates on **applicom**® interfaces a version EPROM minimum V2.5.

7. - Return status of applicom® functions

-Introduction

The various **applicom®** functions return a status word that:

- Guarantees the request quality.
- Can be used to diagnose the cause of a failure.

The significance of the status word value is given in the following tables. As well as the general significance, « Further details » allow you to guide your diagnostic according to the protocol used.

- **applicom®** general statuses

-6	The TRANSCYC (or TRANSCYCPACK) function is used with a cyclic function number that is no longer activated.
-5	The user program tries to perform a TRANSDIF (or TRANSDIFPACK) deferred transfer although the deferred request in progress is not completed.
-1	TRANSDIF (or TRANSDIFPACK) deferred transfer request related to a write that took place correctly.
0	No anomaly detected. The function took place correctly.
1	Unknown function. The requested function is not supported.
2	Incorrect address. The address of the variable you are soliciting is incorrect.
3	Incorrect data. Further details : Function: BINBCD, BCDBIN. - At least one of the accessed values is not in BCD format ($0 \leq \text{value} \leq 9999$).
4	Irretrievable data.
32	Bad parameter passed into the function. Incorrect number of variables.
35	Data not available in cyclic read. Attempt to transfer data with TRANSCYC (or TRANSCYCPACK) before it has been read in the equipment.
40	Deferred read or write attempt when the deferred request register is full. Another task must free the resources by making an exitbus.
41	Deferred read or write attempt when the deferred request register is full Perform deferred request transfers with TRANSDIF (or TRANSDIFPACK) in order to release the register (64 positions).
42	Deferred request transfer attempt with TRANSDIF (or TRANSDIFPACK) when the latter is empty (no deferred requests in progress).
45	Non-resident communication software. Initialize the applicom® interface before using it by typing command applicom (or PCINIT under Windows).
46	Board number not configured, or Master/client applicom® function aiming at a channel configured as slave/server, or vice versa.
47	No applicom® interface.
51	Driver system problem.
59	Protection key missing on the applicom® interface.. Use a applicom® function without previously call the INITBUS function.
66	Insufficient applicom® interface memory.

255	Used by the « PCDDE » MS-Windows server. Initial value of « STATUS_READ » and « STATUS_WRITE ». This value indicates that no transaction has been made between « PCDDE » and applicom® interface.
-----	--

Comments :

Negative function codes are information codes.

- Statuses according to the protocol

-6	The TRANSCYC (or TRANSCYCPACK) function is used with a cyclic function number that is no longer activated.
-5	The user program tries to perform a TRANSDIF (or TRANSDIFPACK) deferred transfer although the deferred request in progress is not completed.
-3	Transmitted write function in general diffusion (No report). General diffusion is carried out in Jbus/Modbus with an equipment number equal to 0.
-1	TRANSDIF (or TRANSDIFPACK) deferred transfer request related to a write that took place correctly.
0	No anomaly detected. The function took place correctly.
1	Unknown function. The requested function is not supported.
2	Incorrect address. The address of the variable you are soliciting is incorrect.
3	Incorrect data. Further details : Jbus / Modbus protocol: - Error report returned by the equipment. - Reception of a correct frame but not corresponding to the question: increase Time-Out. Function: BINBCD, BCDBIN, READWORDBCD, WRITEWORDBCD. - At least one of the accessed values is not in BCD format ($0 \leq \text{value} \leq 9999$).
4	Irretrievable data. Further details : Jbus / Modbus Protocol: - Error report returned by the equipment "Equipment not ready".
32	Bad parameter passed into the function. Incorrect number of variables.
33	Response time fault (Time-Out). Further details : Jbus / Modbus Protocol: - No response to the request within time limit. Increase the Time-Out value. - Check wiring. Check the channel and slave configuration. - For a write, if the values are positioned in the equipment, check the reception loop. Increase the Time-Out value.
34	Parity fault in reception or of control word (CRC16, BCC). Check the configuration of the channel and target equipment. Check the wiring (RC at end of line for the RS485).
35	Data not available in cyclic read. Attempt to transfer data with TRANSCYC (or TRANSCYCPACK) before it has been read in the equipment.
36	Equipment not configured. Define the equipment configuration with PCCONF and start again the applicom® product.
40	Deferred read or write attempt when the deferred request register is full. Another task must free the resources by making an exitbus.
41	Deferred read or write attempt when the deferred request register is full Perform deferred request transfers with TRANSDIF (or TRANSDIFPACK) in order to release the register (64 positions).
42	Deferred request transfer attempt with TRANSDIF (or TRANSDIFPACK) when the latter is empty (no deferred requests in progress).
43	Jbus / Modbus Protocol Unknown slave status since no exchange took place since the program started.
45	Non-resident communication software. Initialize the applicom® interface before using it by typing command applicom (or PCINIT under Windows).
46	Board number not configured, or Master/client applicom® function aiming at a channel configured as slave/server, or vice versa.
47	No applicom® interface.

48	The applicom® interface RS232 channel configured as " 6 signals " cannot transmit since it waited for the Modem CTS for more than 3 seconds during a transmission.
50	Reserved by applicom® .
51	Driver system problem.
59	Protection key missing on the applicom® interface.
66	Insufficient applicom® interface memory.
255	Used by the « PCDDE » MS-Windows server. Initial value of « STATUS_READ » and « STATUS_WRITE ». This value indicates that no transaction has been made between « PCDDE » and applicom® interface.

Comments :

Negative function codes are information codes.

In Jbus / Modbus :

Status codes 1 to 31 correspond to error messages detected by the slave, the others are detected by the communication software.

8. - Glossary of terms

***applicom*® interface**

Communication card, ISA or PCI type, with the ***applicom*®** real time multi-task kernel.

ASCII

American Standard Code for Information Interchange

bauds

Bits per second

BX4010

4 channels distribution box, with galvanic insulation.

BX4010R

4 channels rack distribution box, with galvanic insulation.

Chain

Byte string where each variable (word, byte, character) is separated by a delimiter.

channel

Physical output of an ***applicom*®** card.

client

Node which has communication initiative.

CRC16

Cyclic Redundancy Check.

CTS

Clear To Send.

DATA-BASE

***applicom*®** database, with 32k bits and 32k words.

DCD

Data Carrier Detect.

DDE poke

DDE write request of a variable.

Gateway

***applicom*®** interfaces which access to server equipments on another channel.

Image variable

Data zone of the DATA BASE updated by a cyclic function.

INTEL format

The right word in the double word or in the floating word represents the least significant word.

item

Defines an element

JBUS

Communication protocol of type master-slaves. JBUS is a registered trademark of the APRIL company.

Local variable

Data zone of the DATA BASE not updated by a cyclic function.

master

On a master-slaves network, characterizes the device which initiates the exchanges.

Matrix

Byte table where each variable (word, byte, character) is separated by a delimiter and each line by a 'return' character.

MODBUS

Communication protocol of type master-slaves. JBUS is a registered trademark of the GOULD MODICON company.

MOTOROLA format

The right word in the double word or in the floating word represents the most significant word.

OPC

Ole for Process Control

PCDDE

applicom® DDE server.

EPROM

Read only memory on *applicom*® interfaces which contains programs.

Router

Node which transmits a frame from a network different to the one it is sent.

RTS

Request To Send

RTU

Remote Terminal Unit, binary mode (8 bits).

RxD

Recept signal

Slave

On a Master-slaves bus, characterize devices which have none initiative on the bus. They answer only to the master requests.

TxD

Transmit signal.

9. - Index

- Communication master-slaves, 1
- Configuration Cyclic JBUS/MODBUS slave, 18
- Configuration JBUS / MODBUS master
 - Baud rate, 12
 - Electrical interface, 12
 - Inter-frame silence, 12
 - Parity control, 12
 - Slaves configuration, 14
 - Stop bit, 12
- Configuration JBUS / MODBUS slave
 - Address of 8-bit Quick Reading, 16
 - Baud rate, 16
 - Electrical interface, 16
 - Inter-frame silence, 16
 - Parity control, 16
 - Slave number, 16
 - Stop bit, 16
- Configuration JBUS/MODBUS slave router, 18
- Configuration Slave JBUS/MODBUS gateway, 18
- CRC16, 1
- Cyclic JBUS / MODBUS Slave
 - presentation, 8
 - principle, 9
- Data area, 1
- EPROM, 37
- Error, 38
- Function code, 1
- functions usable
 - cyclic mode, 19
 - deferred mode, 19
 - wait mode, 19
- Gateway, 11
- Item
 - April series 1000 descriptor, 27
 - April SMC descriptor, 32
 - standard Descriptor, 22
- Item descriptor
 - April series 1000 descriptor, 27
 - April SMC descriptor, 32
 - standard descriptor, 22
- JBUS / MODBUS exchange frame, 1
- JBUS/MODBUS master, 2
- JBUS/MODBUS slave, 5
- JBUS/MODBUS slave router, 6
- Limits
 - library, 3
 - OPC, 3
 - PCDDE, 3
- Maximum number of variables per frame
 - library, 3
 - OPC, 3
 - PCDDE, 3

Presentation, 1
RTU, 1
Status, 38
Supported functions, 2, 5