DPR 180/DPR 250

COMMUNICATION OPTION

MANUAL

TABLE OF CONTENTS

TABLE OF CONTENTS, continued

TABLE OF CONTENTS, continued

FIGURES AND TABLES

FIGURES

TABLES

1. OVERVIEW

Before running your communication option, please read the following explanations on how to install it.

1.1 Warning

➢ **If you received your optional communication board as a spare part:**

You are going to install your optional communication board.

To run properly, this application needs a recorder firmware release upper or equal to 001AE.

To know your recorder firmware release, refer to your product manual (sub-section 3-2) or read it on the recorder in SERVICE / MISCELLANEOUS, SOFTWARE.

If your recorder firmware version is higher than 001AE, please go to the second step.

If your recorder firmware release is lower than 001AE (or if you wish to upgrade it), follow the procedure given in the **PC Configurator Kit 46190407-501 :**

1. Install the PC configurator software (included in the kit **46190407-501**) on your PC.

The minimum PC configuration required is a 486 with 4 Mb of RAM and 10 Mb free on your hard disk. The software is compatible with Windows 3.1, Windows 3.11 and Windows 95.

2. Install the new recorder firmware (included in this kit) on your PC.

3. Connect the PC - Recorder interface (Kit # 46190409-501 **not included**).

4. Upgrade the recorder firmware.

5. Install the optional communication board as described in this manual

(refer to Section 2; sub-section 2.1)

6. Check or modify the configuration switches (refer to sub-section 2.1)

7. Wire the communication board (refer to sub-section 2.2)

8. Configure the communication option as described in Section 3 of the manual.

➢ **If you received your optional communication board with your recorder:**

The communication board can run directly.

However, you have to check the following points:

1. Verify or modify the switches configuration on the communication board (refer to sub-section 2.1)

2. Wire the communication option (refer to sub-section 2.2)

3. Configure the communication option as described in Section 3 of this manual.

➢ **Note that you can only upgrade the communication firmware from the front panel of the recorder (with a jack cable).**

1.2 Protocol

- MODBUS RTU
- ASCII Modified transparent mode

1.3 Main functions

- Selectable RS232/RS422/RS485
- Reading PV's, alarm states...
- Sending Communication PV's to recorder
- Printing messages on the chart
- Reading and changing configuration parameters like alarm setpoints, ranges, trend speed, tagnames etc...(ASCII protocol only)
- Printing Communication PV's on the chart
- Applying alarm on Communication PV's
- Applying mathematical functions on Communication PV's (if Maths option present)
- Displaying Communication PV's on the recorder display
- Activating relay or displaying message on "SHEDTIME" event
- Isolation between recorder and communication wires

2. INSTALLATION

2.1 Installing the universal communication option board

➔ **WARNING**: Please use an antistatic ground strap to avoid possible electrostatic damage to the printed circuit boards.

1. Isolate the recorder from the main supply.
2. Open the recorder door and remove the chart cassette from the chassis.
3. Turn OFF the switch (See Figure 2-1)

Recorder's main powerswitch located behind the chart cassette

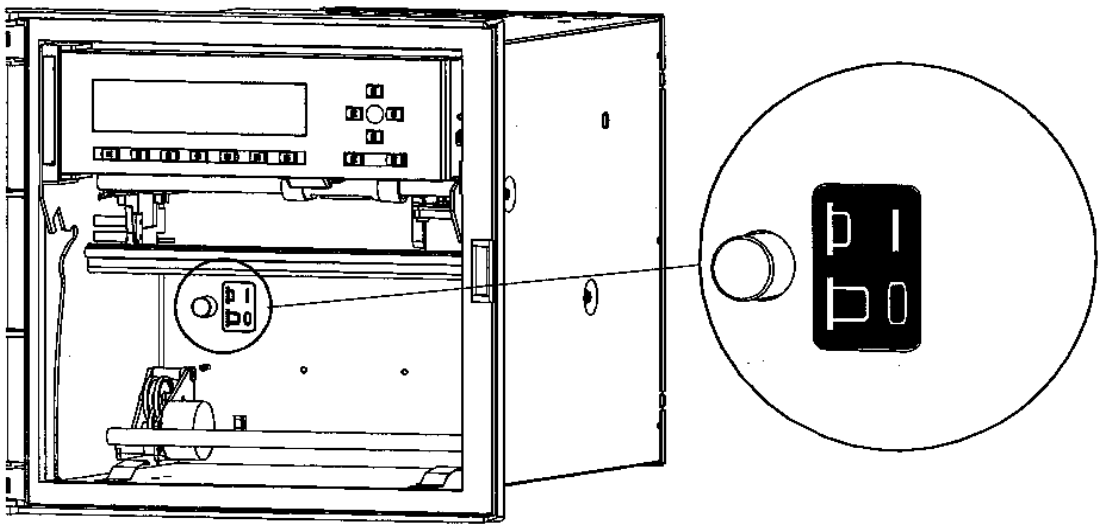


Figure 2-1

4. Loosen the 3 fixing screws from the rear cover (see ref.A, Figure 2-2)

5. Remove the rear cover. (See ref. B & C, Figure 2-2)

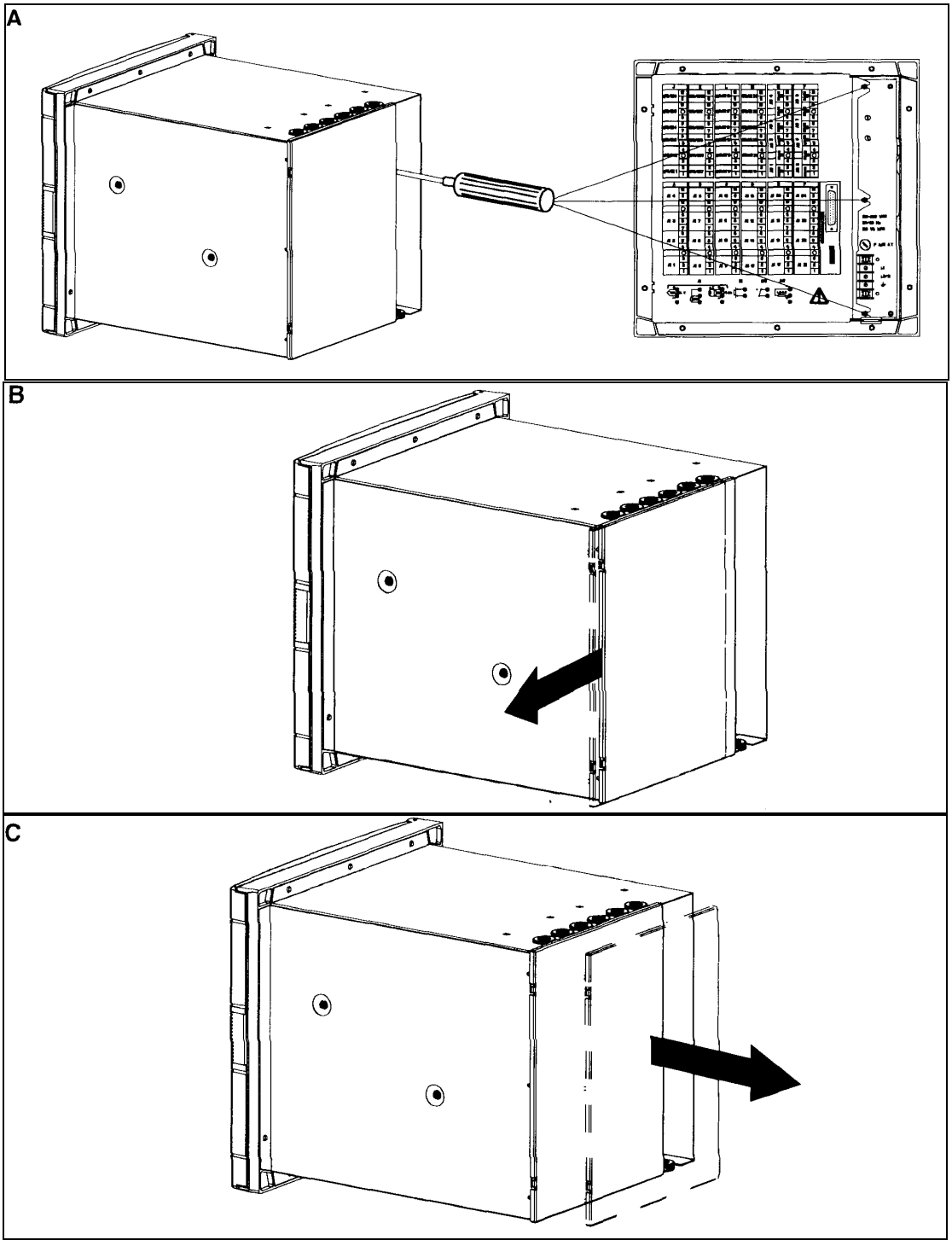


Figure 2-2

6. Remove the terminal block or slot cover location F (see ref. F, Figure 2-6)

7. Remove the cover which protects the CPU board.
   Unscrew the 3 fixing screws (see ref.A, Figure 2-3)

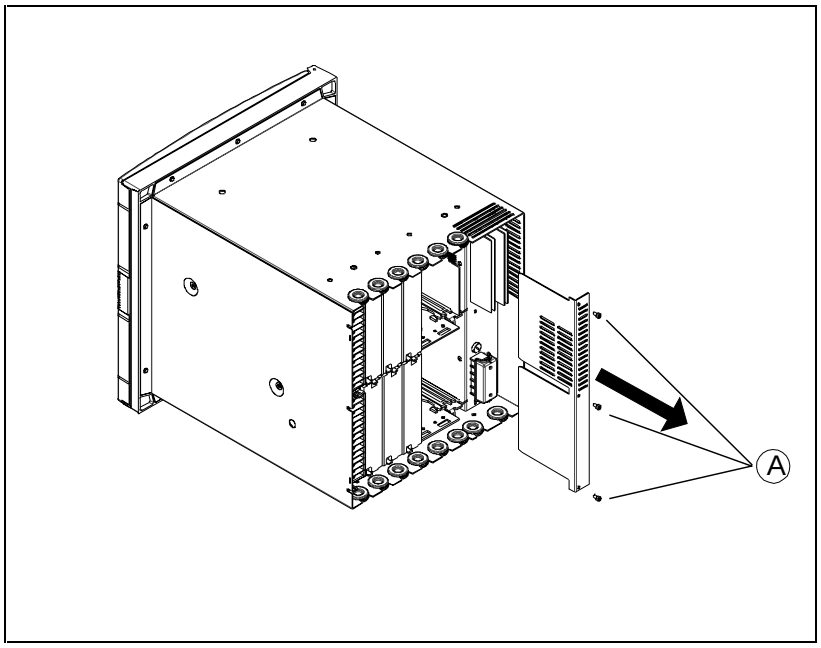


Figure 2-3

8. Ensure your antistatic ground strap is connected to earth before proceeding to avoid possible electrostatic damage to the CPU board or communication board.
9. Verify the switches configuration on communication board (See Figure 2-4)

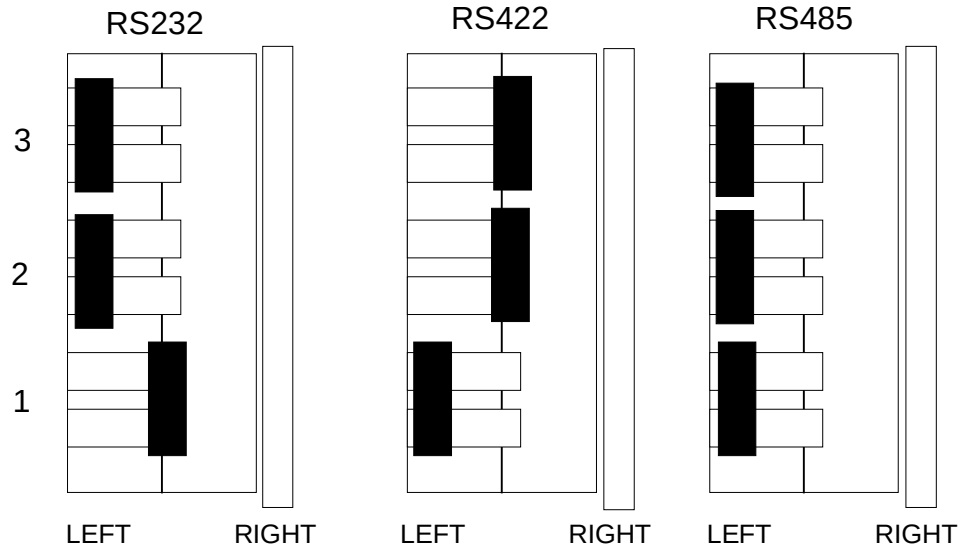


Figure 2-4 RS232 / RS422 / RS485 SWITCHES CONFIGURATION

10. Fit the new cover with communication board.

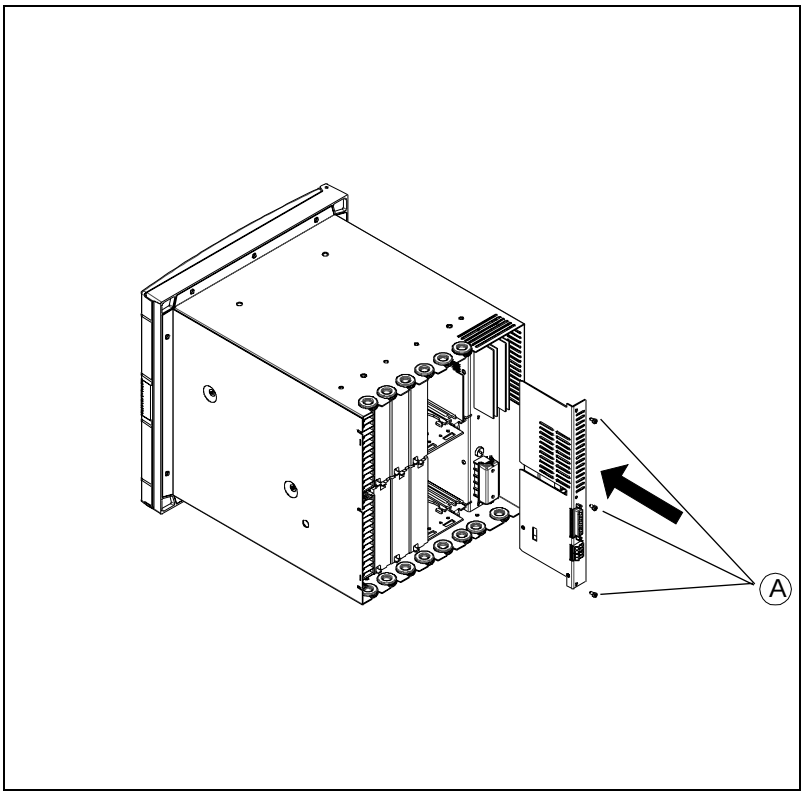


Figure 2-5

11. Replace the 3 fixing screws (see ref. A, Figure 2-5)

12. Wire the communication terminal block or connect the communication cable RS232 (see Section 2.2)

In this case, we recommend to use a right angle connector so that the rear metal cover of the recorder can be reinstalled, or to order the communication cable referenced 46210098-501.

13. Replace the slot cover or terminal block.

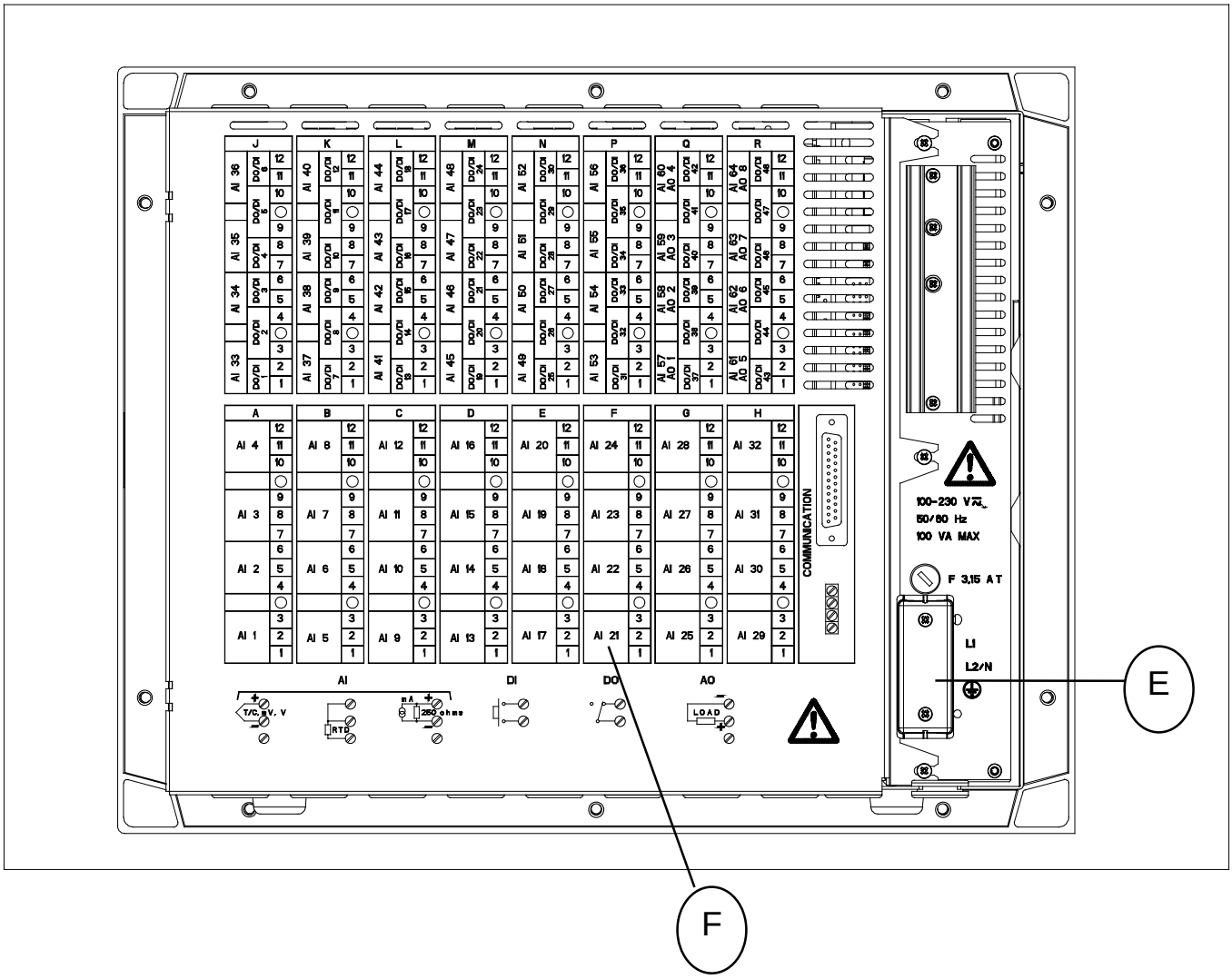


Figure 2-6

14. Replace the rear cover.

15. Turn ON the switch (see Figure 2-1)

16. Replace the chart cassette.

2.2 Wiring configuration

This software package has been designed to operate with three kinds of serial communication standards which are: RS232, RS422 and RS485. Refer to the following chapters for the wiring configuration of each of them. For more details on the wiring, please refer to your computer product manual.

2.2.1 RS232 wiring configuration

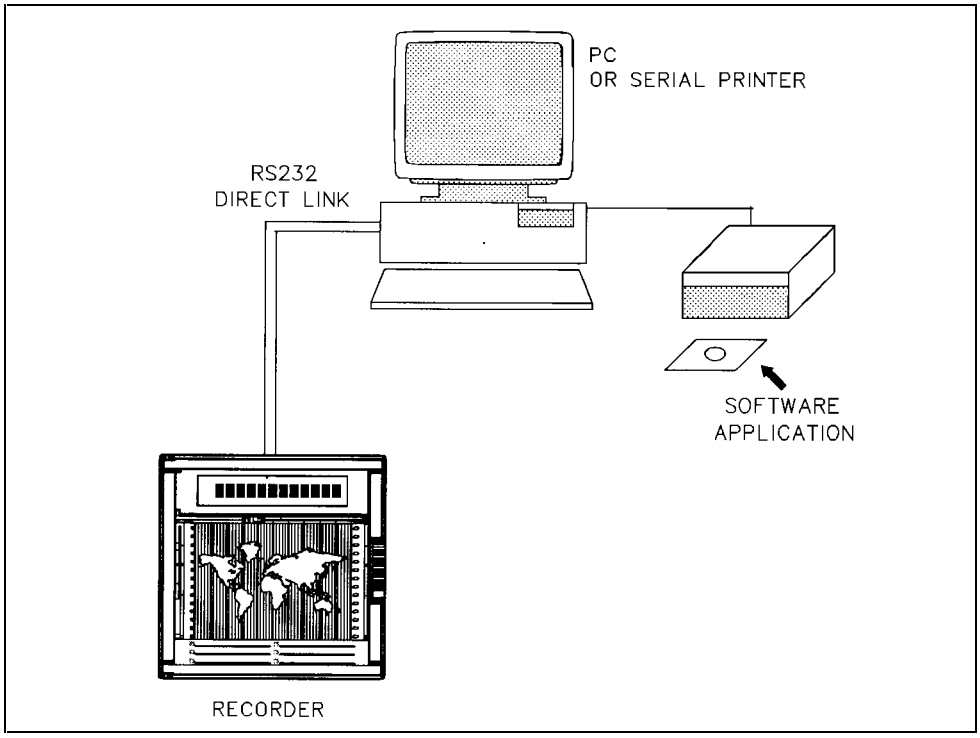
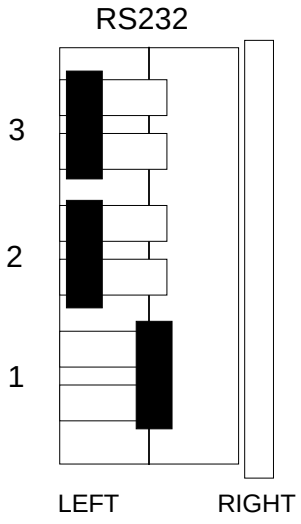


Figure 2-7 RS232 wiring configuration

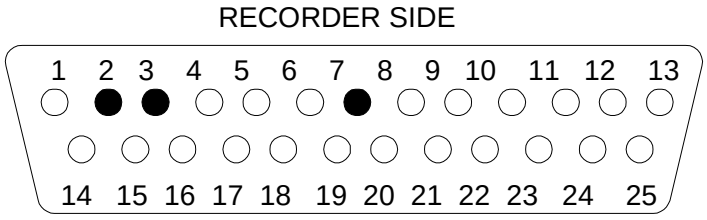
2.2.1.1 Switch configuration



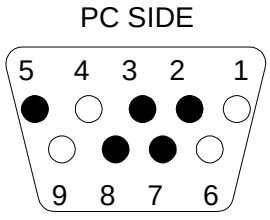
2.2.1.2 Interface connector

2.2.1.2.1 With DB9 connector

Interface cable connectors pin arrangement and signal functions.



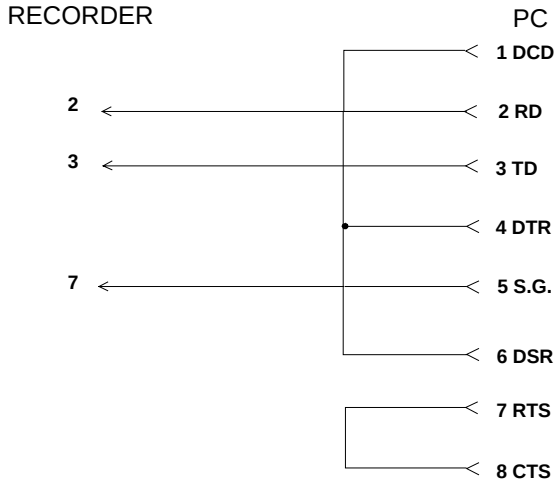
DB25 male connector face view



DB9 female connector face view

RECORDER	**PC**
Pin n°	Pin n°
2	2
3	3
5	4
7	5
20	6
20	8

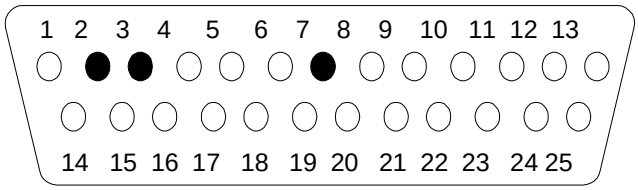
Note : Check compatibility with your PC as far as no standard for DB9 connector exists yet.



2.2.1.2.2 With DB25 connector

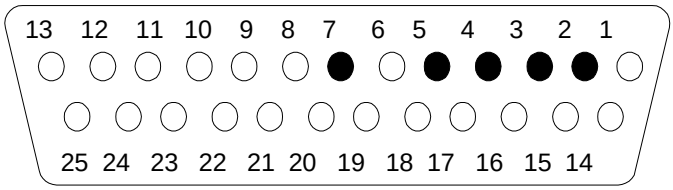
Interface cable connectors pin arrangement and signal functions.

RECORDER SIDE



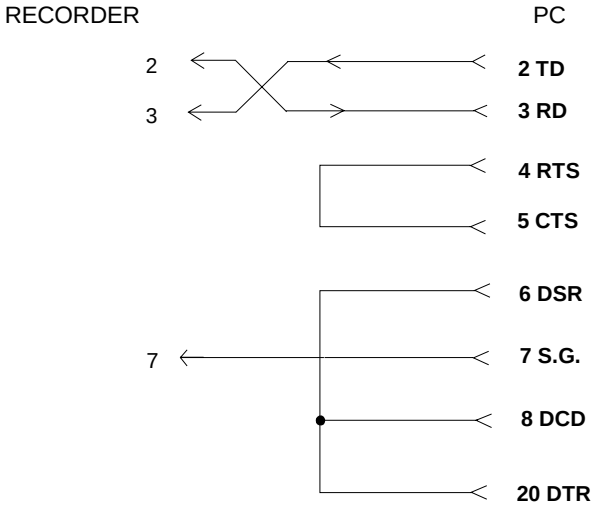
DB25 male connector face view

PC SIDE



DB25 female connector face view

RECORDER	PC		
Pin n°	Pin n°	**Direction**	**Description**
3	2	to recorder	transmitted DATA
2	3	from recorder	received DATA
-	4	from DTE	request to send
-	5	to DTE	clear to send
7	7	-	ground



2.2.2 RS422 wiring configuration

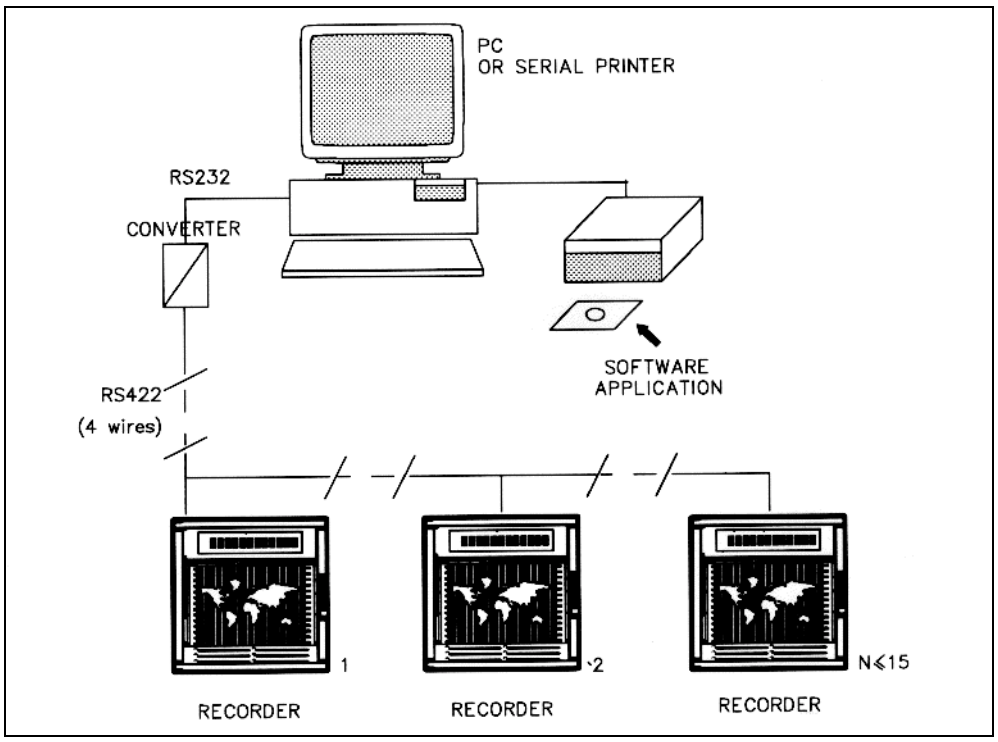
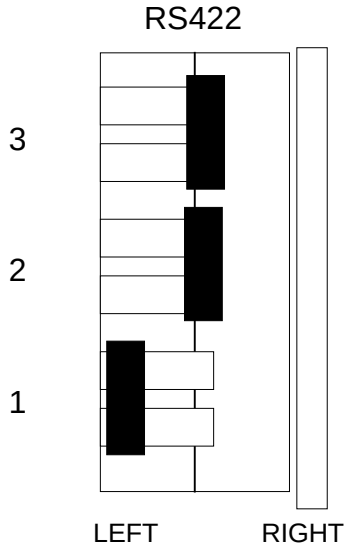


Figure 2-8 RS422 wiring configuration

2.2.2.1 Switch configuration



2.2.2.2 Interface connector

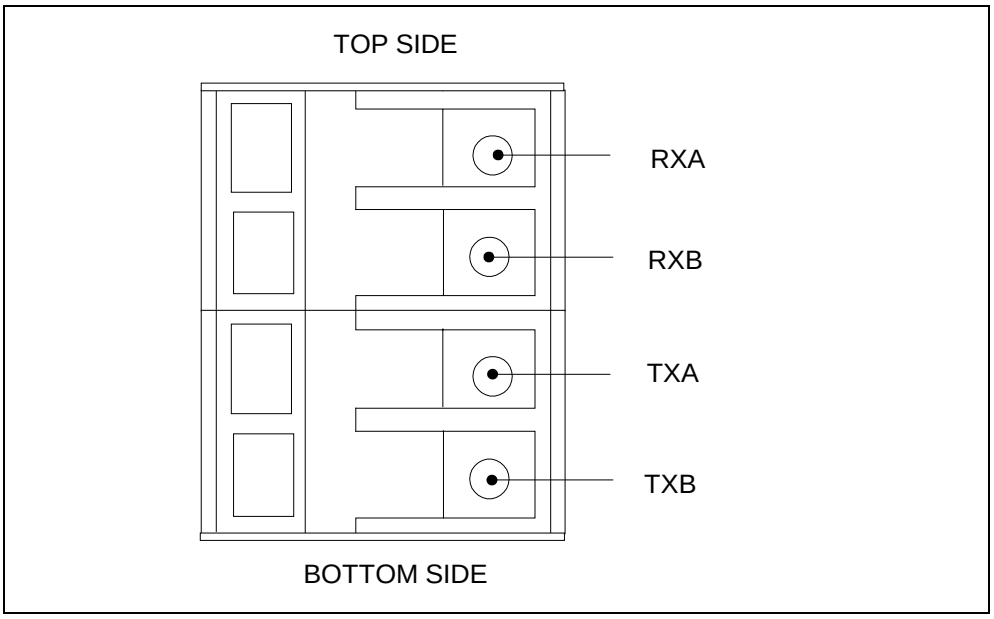


Figure 2-9

2.2.3 RS485 (2 wires) wiring configuration

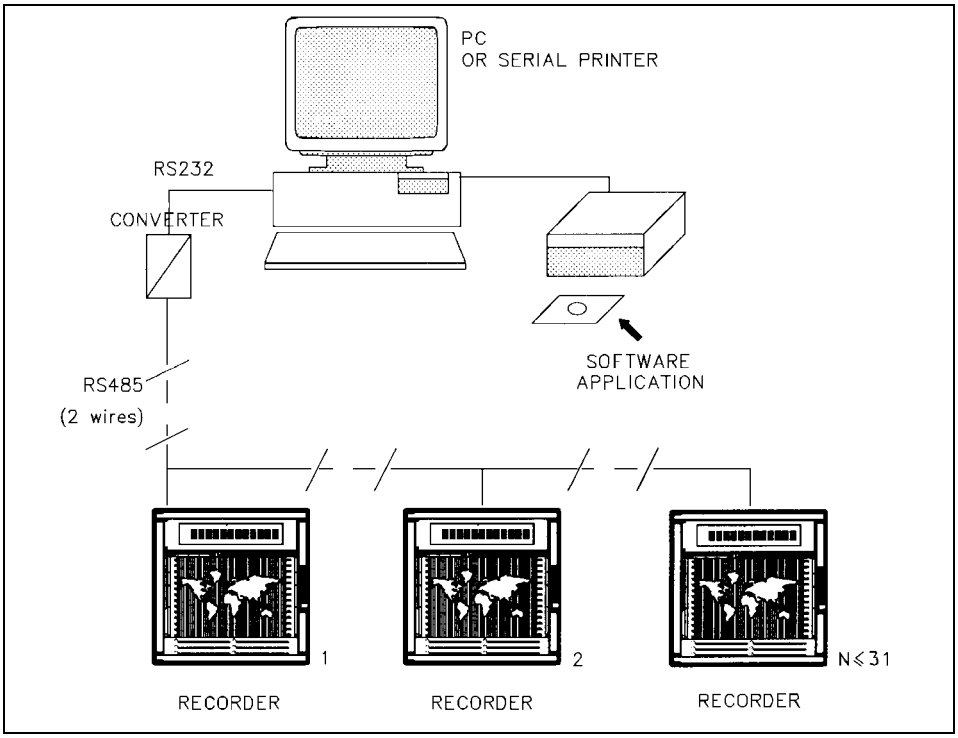
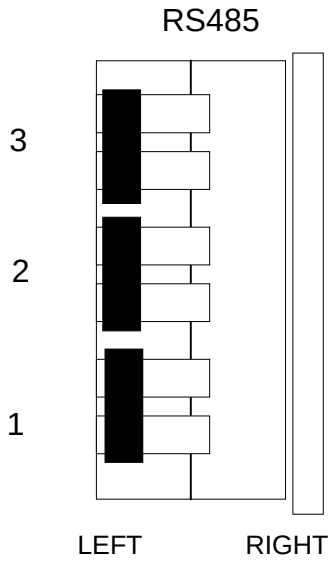


Figure 2-10 RS485 wiring configuration

2.2.3.1 Switch configuration



2.2.3.2 Interface connector

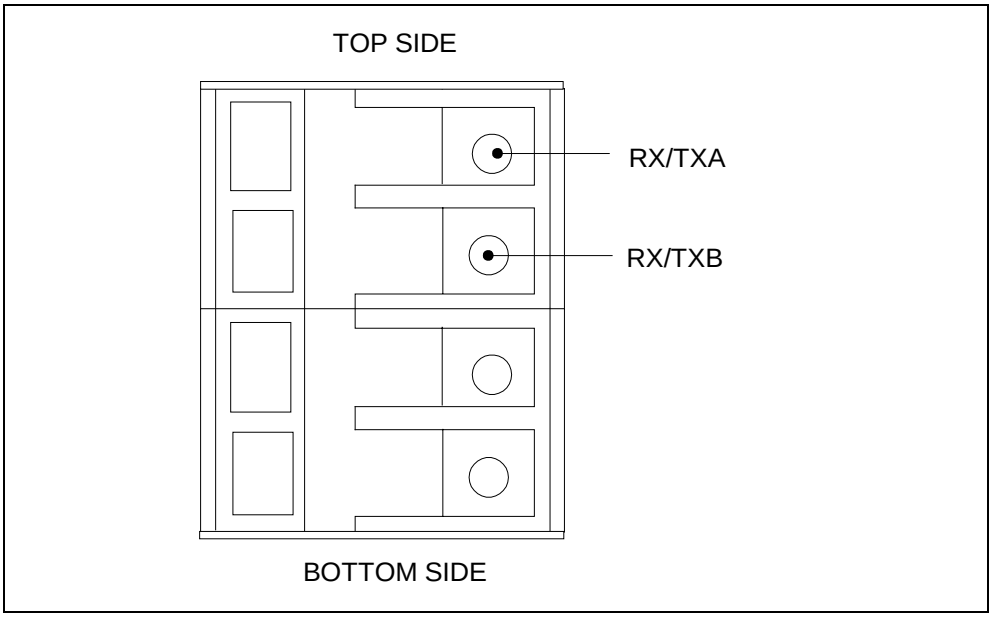


Figure 2-11 Interface connector

2.2.4 Connecting the RS422/485 link to a computer

The recorder with RS422/485 Communications option can be connected to your computer using one of two arrangements shown below.

Arrangement	**Description**
ICS plug-in I/O board	Wired directly to the RS422/485 port in your computer using an ICS plug-in I/O board which is specifically designed to interface with the IBM (or IBM compatible) PC, PC/XT; or PC/AT computer. This board is available from... ICS Computer Products, Inc. 5466 Complex Street Suite 208 San Diego, California 92123
Burr-Brown Converter	Using the RS232 port a Burr-Brown RS232 to RS422/485 converter installed between the RS232 port and the recorder. This converter is available from : Burr-Brown International Airport Industrial Park P.O. Box 11400 Tucson, Arizona 85734 Part number LDM485ST, limited distance modem

2.2.5 Rear connection

The recorder has built in circuits to reduce the effects of most electrical noise. We recommend that you review the following guidelines, to minimize the noise effects.

1. Separate the communication leadwires from the line voltage, the alarm output, contactors, motors etc...
2. For a communication distance, over 1.5 meters, use a separate metal tray, or metal conduit.
3. Use wiring cable composed of twisted pair wirings, with a shield for RS485 and RS422. Use a shielded cable for RS232.
4. Connect the shield wire to the ground, at one end only, preferably at the recorder (see screws on figure 2-5, ref. A). Use for example a wiring cable type: Belden 9271 twinax, or equivalent.
5. We recommend to install a 120 ohms resistor between TXA and TXB, on the last recorder on communication link.
6. The maximum capabilities are:

Type of communication.	Distances max.	# of Unit
RS232	15 meters	1
RS422	1000 meters	15
RS485	1200 meters	31

3. COMMUNICATION CONFIGURATION

3.1 COMMUNICATION sub-matrix parameters

3.1.1 COMMUNICATION sub-matrix parameters list

SUB-MATRIX → **COMMUNICATION**

Position of parameters

3.1.2 Explanation of the classification

♦ ♦	The configuration of parameters will stop the measuring and the printing.
♣ ♣	To modify a parameter, **PASSWORD 2** is required.

3.1.3 COMMUNICATION sub-matrix parameters description

SUB-MATRIX	*PARAMETER*	*CLASSIFICATION*
COMMUNICATION	**PROTOCOL**	♦ ♦ ♣ ♣
Definition:	Selection of protocol.	
How to modify it:	Select a new value.	
Possible values:	**RTU** **ASCII**	

SUB-MATRIX	*PARAMETER*	*CLASSIFICATION*
COMMUNICATION	**CONNECT**	♦ ♦ ♣ ♣
Definition:	Type of connection.	
How to modify it:	Select a new value.	
Possible values:	**RS232** **RS422** **RS485**	
WARNING:	**Must be the same as the switches configuration on the communication board.**	

SUB-MATRIX	*PARAMETER*	*CLASSIFICATION*
COMMUNICATION	**ADDRESS**	♦ ♦ ♣ ♣
Definition:	Address of the recorder on the network.	
How to modify it:	Enter a numeric value.	
Possible values:	**[0 .. 99]**	

SUB-MATRIX	*PARAMETER*		*CLASSIFICATION*
COMMUNICATION	**BAUDS**		♦ ♦ ♣ ♣
Definition:	Baud rate of the communication line.		
How to modify it:	Select a new value.		
Possible values:	**110** **150** **300** **600**	**1200** **2400** **4800** **9600**	**19200** **38400**

SUB-MATRIX	*PARAMETER*	*CLASSIFICATION*
COMMUNICATION	**BITS**	♦ ♦ ♣ ♣
Definition:	Number of bits per character.	
How to modify it:	**NOT POSSIBLE**	
Possible values:	**8 BITS/CHAR or 7 BITS/CHAR (see note in the parity frame)**	

SUB-MATRIX	*PARAMETER*	*CLASSIFICATION*
COMMUNICATION	**STOP**	♦ ♦ ♣ ♣
Definition:	Number of stop bits.	
How to modify it:	**NOT POSSIBLE**	
Possible values:	**1 STOP BIT only**	

SUB-MATRIX	*PARAMETER*	*CLASSIFICATION*
COMMUNICATION	**PARITY**	♦ ♦ ♣ ♣
Definition:	Parity of character.	
How to modify it:	Select a new value.	
Possible values:	**ODD** **EVEN** **NONE**	
NOTE:	**THIS SELECTION IS POSSIBLE ONLY IF THE ASCII PROTOCOL HAS BEEN SELECTED. IN RTU, THE PARITY IS AUTOMATICALLY SET TO NONE.** **Depending on the type of parity selected, the number of data bit is not always the same:** **Case ODD or EVEN** **1 start bit 7 data bits 1 parity bit 1 stop bit** **Case NONE** **1 start bit 8 data bits no parity bit 1 stop bit**	

SUB-MATRIX	*PARAMETER*	*CLASSIFICATION*
COMMUNICATION	**SHEDTIME**	♦ ♦ ♣ ♣
Definition:	This parameter allows you to control if the time elapsed between two communications is greater than the configured shedtime value or if there is no more communication with the supervisor. A shedtimer and an event are associated to shedtime parameter.	
How to modify it:	Enter a numeric value.	
Possible values:	**[1 .. 3000] seconds** 0 second means **no shedtime configured.**	
NOTE:	Each time the communication board receives a message, it resets the shedtimer to the shedtime value. If the shedtimer expires before the next message, the shedtime event is activated. Once the event is activated, depending on the configuration, a message can be displayed, or/and a relay can be activated. The configuration of the shedtime event parameter is in the **EVENT** sub-matrix.	

3.2 COMMUNICATION sub-matrix services

3.2.1 COMMUNICATION sub-matrix services list

SUB-MATRIX → **COMMUNICATION SERVICES**

Position of services

→ **COM TEST** page 20

→ **SOFTWARE** page 20

→ **PROD TEST** page 21

3.2.2 COMMUNICATION sub-matrix services description

SUB-MATRIX	*TYPE OF SERVICE*	*CLASSIFICATION*
COMMUNICATION	**COM TEST**	♦ ♦ ♣ ♣
Definition:	Allows the user to test the communication option board.	
How to use/execute it:	**Recorder configuration:** - Configure **CONNECT** parameter to RS232 - In the RS232 connector, connect reception to transmission pin. In the COMMUNICATION SERVICES sub-matrix, select COM TEST and press **ENTER.** When "**CONFIRM**" is flashing, press **ENTER** to start the test. The message "**WAIT PLEASE**" is flashing and after several seconds, one of these messages is displayed: - **TEST PASSED**: Transmission with communication board is correct. - **TEST FAILED**: Problem in Rx, Tx pins connection or in the communication board. - **NO RESPONSE**: Problem between the recorder mother board and the communication board. Check the connection with the communication board. Press **SET UP** twice to come back to main menu.	

SUB-MATRIX	*TYPE OF SERVICE*	*CLASSIFICATION*
COMMUNICATION	**SOFTWARE**	♦ ♣
Definition:	Shows the communication option software version.	
How to use/execute it:	You can only read the value. Press **SET UP** to escape.	

SUB-MATRIX	*TYPE OF SERVICE*	*CLASSIFICATION*
COMMUNICATION	**PROD TEST**	♦ ♦ ♣ ♣

Definition:

This is a more complete test which gives diagnostic about RS232/MODEM pins and also about RS422/485 connectors.

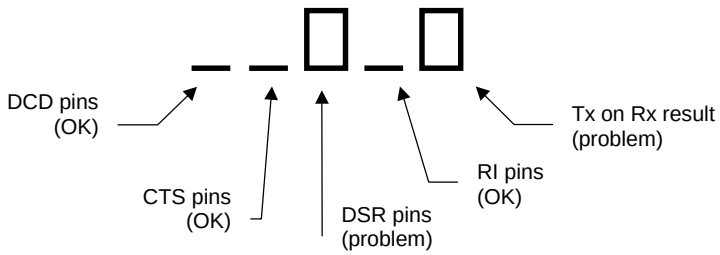
Before running the test, it is required to have connected communication board connectors as described in the diagram on page 22.

How to use/execute it:

- In the COMMUNICATION SERVICES sub-matrix, select "**PROD TEST**" and press **ENTER.** When "**CONFIRM**" is flashing, press **ENTER** to start the test.
- The message "**RS232/SWITCH**" is blinking.
- On your communication board, set red connection switch in the RS232 position.
- When ready, press **ENTER.**

The message "**WAIT PLEASE**" is flashing and after several seconds, one of these messages is displayed:

- "**RS232 PASSED**": Transmission with communication board is correct.
- In case of failure, the "**RS232**" message is displayed followed by a diagram representing the pins states (an underscore means that the connection is OK, a rectangle indicates a problem):



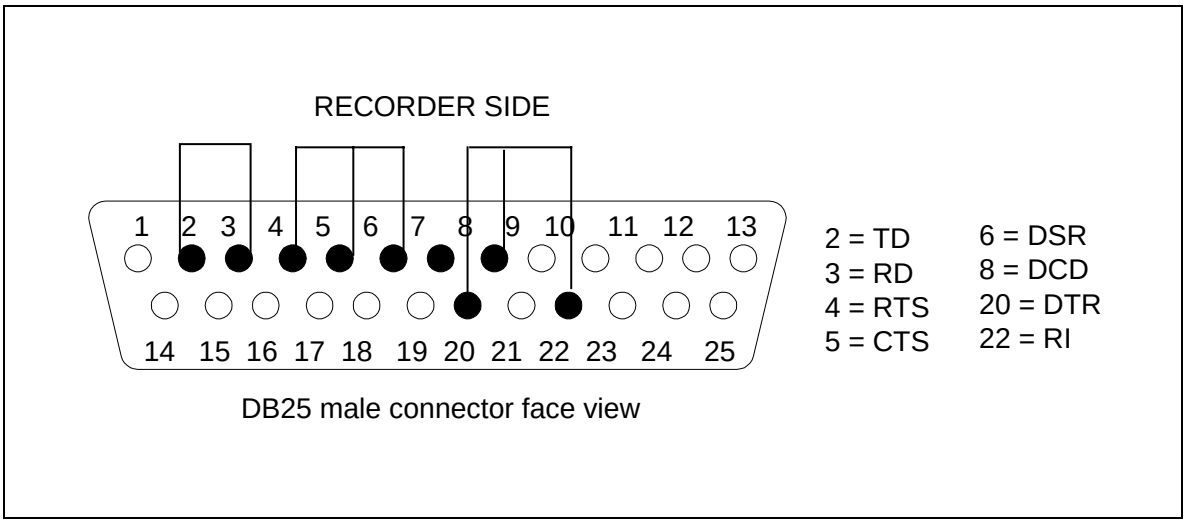
This message indicates that the DSR pins are badly wired. It also indicates a problem on either the Tx or the Rx pin.
- After this kind of message or after a PASSED message, press **ENTER** to run RS422 test.
- The message "**RS422/SWITCH**" is blinking.
- Remove all connections on the RS232 connector.
- On your communication board, set red connection switch in the RS422 position.
- When ready press **ENTER**.

The message "**WAIT PLEASE**" is flashing and after several seconds, one of these messages is displayed:
- "**RS422 PASSED**"**:** Transmission with communication board is correct in RS422.
- "**RS422 FAILED**": Problem in Rx, Tx pins connection or in the communication board.
- Remove all connections on the RS422/485 connectors (Tx on Rx is now ensured by the RS485 switch position)
- Press **ENTER** to run RS485 test.
- The way to run it is the same as in RS422 (do not forget to set connection switch in the RS485 position).
- After the PASSED or FAILED message, press **ENTER.**

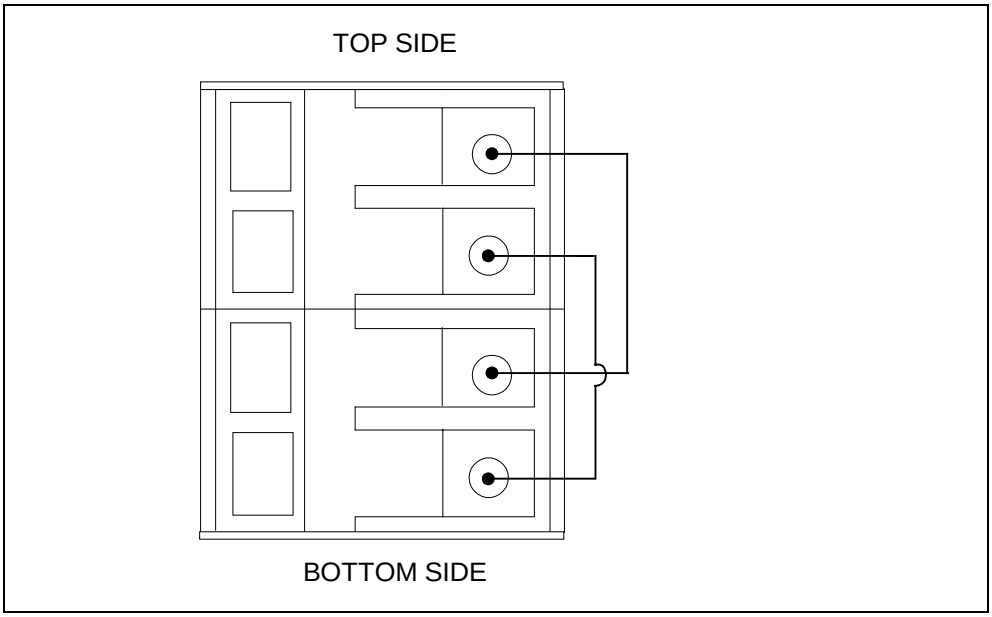
"**END OF TEST**" is displayed and after two seconds, you are back to the communication services level. Press **SET UP** to return to main menu.

Note that all through the test, the DISPLAY key makes the recorder exit and initialize again and the SETUP key makes the recorder go back to the communication services level.

DIAGRAM:



Pins to connect on the communication board DB25 connector to run RS232 PROD TEST properly. (It is important to remove those connections while running RS422/485 PROD TEST).



Pins to connect on the communication board RS422/485 connector to run RS422 PROD TEST properly (It is important to remove those connections while running RS485 PROD TEST).

4. RTU PROTOCOL DEFINITION

4.1 General

A data communication system protocol controls the language structure or message format common to all devices on a network. The protocol determines how the master and slave establish and break off contact, how the sender and receiver are identified, how the messages are exchanged in an orderly manner, and how errors are detected. The protocol controls the query and response cycle which takes place between master and slave devices, as shown in figure 4-1.

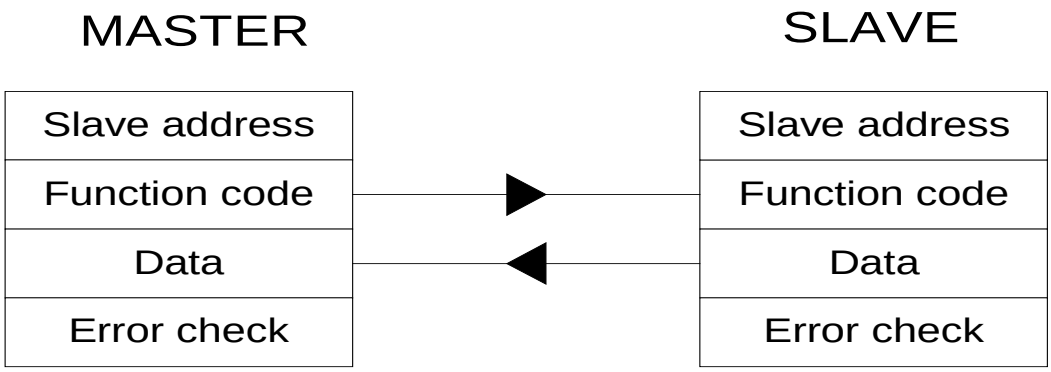


Figure 4-1 Master slave exchanges

Only the master initiates the transaction. Transactions are either a query/response type (only a single slave is addressed).

Certain characteristics of the MODBUS protocol are fixed, such as frame format, frame sequences, handling of communication errors and exception conditions, performed functions.

Other characteristics are user selectable. These include a choice of transmission medium and baud rate.

When the message reaches the MODBUS slave interface, it enters this addressed device through a similar "port". The addressed device removes the envelope, reads the message, and, if no errors have occurred, performs the requested task. Then it replaces the message into the saved envelope and "returns to sender". The information in the response message is:

- the slave address
- the action performed
- the data required (if a read action has been requested)
- a means of checking errors.

4.1.1 Mode of transmission: RTU (remote terminal unit)

The mode of transmission is the structure of the data within a message, and the coding system used to transmit the data.

Characteristic:

Number of bits/character :

- start bits	1
- data bits (least significant first)	8
- no parity	
- stop bit	1

NOTE: In the MODBUS RTU mode, message characters must be transmitted in a continuous stream. Data bits are transmitted and received LSB first.

A CRC (cyclic redundancy check) error check sequence is calculated on each message and added to the frame before transmission. The receiving unit recalculates the CRC and compares it to the transmitted CRC.

When the character redundancy check detects a communication error, processing of the message stops. A PLC slave will not act on or respond to the message. (The same result occurs if a non-existent slave address is used).

When a communication error occurs, the message is unreliable. The PLC slave cannot know for sure if this message was intended for it. So the CPU might be answering a message which was not its message to begin with. It is therefore essential to program the Modbus master to consider that a communication error has occurred if there is no response in a reasonable time. The length of this time period depends on baud rate, the type of message, and the scan time of the PLC slave. Once this time is determined, the master may be programmed to automatically retransmit the message.

4.1.2 CRC-16 (Cyclic Redundancy Check) error check sequence

The CRC-16 error check sequence is implemented as described in the following paragraph:

The message (data bits only, disregarding start / stop bits) is considered as one continuous binary number whose most significant bit (MSB) is transmitted first. The message is pre-multiplied by x 16 (shifted left 16 bits), then divided by (x16+x15+x2+1) expressed as a binary number (11000000000000101). The integer quotient digits are ignored and the 16-bit remainder (initialized to all ones at the start to avoid the case of all zeros being an accepted message) is appended to the message (MSB first) as the two CRC check bytes. The resulting message including CRC, when divided by the same polynomial (x16+x15+x2+1) at the receiver will give a zero remainder if no error has occurred. (The receiving unit recalculates the CRC and compares it to the transmitted CRC). All arithmetic is performed modulo two (no carries). An example of CRC-16 error check for message HEX. 207 (address 2, function 7 or a status request to slave number 2) is given in table 4-1.

The device used to serialize the data for transmission will send the conventional LSB or right-most bit of each character first. In generating the CRC, the first bit transmitted is defined as the MSB of the dividend. For convenience then, and since there are no carries used in arithmetic, let's assume while computing the CRC that the MSB is on the right. To be consistent, the byte order of the generating polynomial is dropped since it affects only the quotient and not the remainder.
This yields 1010 0000 0000 0001 (HEX A001). Note that this reversal of the bit order will have no effect whatever on the interpretation or bit order of characters external to the CRC calculations.

The step by step procedure to form the CRC-16 check bytes is as follows:

☑ Load a 16-bit register with all 1's.

☑ Exclusive OR the first 8-bit byte with the high order of the 16-bit register, putting the result in the 16-bit register.

☑ Shift the 16-bit register one bit to the right.

☑ a. If the bit shifted out to the right (flag) is one, exclusive OR the generated polynomial (1010 0000 0000 0001) with the 16-bit register.

b. If the bit shifted out to the right is a zero, return to step 3.

☑ Repeat step 3 and 4 until 8 shifts have been performed.

☑ Exclusive OR the next 8-bit byte with the 16-bit register.

☑ Repeat step 3 through 6 until all bytes of the message have been exclusive OR with the 16-bit register and shifted 8 times.

☑ The 16-bit register contains the 2 byte CRC error check and is added to the message most significant bits first.

Table 4-1 CRC-16 generation

	REGISTER 16-BITS			MSB	INDIC.
Exclusive or	1111	1111	1111	1111	
0			0000	0010	
	1111	1111	1111	1101	
shift 1	0111	1111	1111	1110	1
polynomial	1010	0000	0000	0001	
	1101	1111	1111	1111	
Shift 2	0110	1111	1111	1111	1
polynomial	1010	0000	0000	0001	
	1100	1111	1111	1110	
Shift 3	0110	0111	1111	1111	0
Shift 4	0011	0011	1111	1111	1
polynomial	1010	0000	0000	0001	
Shift 5	1001	0011	1111	1110	
	0100	1001	1111	1111	0
Shift 6	0010	0100	1111	1111	1
polynomial	1010	0000	0000	0001	
	1000	0100	1111	1110	
Shift 7	0100	0010	0111	1111	0
Shift 8	0010	0001	0011	1111	1
polynomial	1010	0000	0000	0001	
	1000	0001	0011	1110	
07			0000	0111	
	1000	0001	0011	1001	
Shift 1	0100	0000	1001	1100	1
polynomial	1010	0000	0000	0001	
	1110	0000	1001	1101	
Shift 2	0111	0000	0100	1110	1
polynomial	1010	0000	0000	0001	
	1101	0000	0010	1111	
Shift 3	0110	1000	0010	0111	1
polynomial	1010	0000	0000	0001	
	1100	1000	0010	0110	
Shift 4	0110	0100	0001	0011	0
Shift 5	0011	0010	0000	1001	1
polynomial	1010	0000	0000	0001	
	1001	0010	0000	1000	
Shift 6	0100	1001	0000	0100	0
Shift 7	0010	0100	1000	0010	0
Shift 8	0001	0010	0100	0001	0
	HEX 12			HEX 41	

TRANSMITTED MESSAGE WITH CRC-16 (Message shifted to right to transmit)							
12		41		07		02	
0001	0010	0100	0001	0000	0111	0000	0010
Last byte transmitted		TRANSMISSION ORDER				First byte transmitted	

4.2 Modbus protocol

4.2.1 Remote Terminal Unit (RTU) framing

Frame synchronization can be maintained in MODBUS RTU transmission mode only by simulating a synchronous message. The receiving device monitors the elapse time between receipt of characters. If three and one-half character time elapse without a new character or completion of the frame, then the device ignores the frame and assumes that the next received byte will be an address. See table 4-2.

Table 4-2 MODBUS RTU frame format

3.5 character	ADDRESS	FUNCTION CODE	DATA	ERROR CHECK	3.5 character
transfer time minimum	8 BITS	8 BITS	(Number of data) x 8 BITS	16 BITS	transfer time minimum

4.2.2 Address field

The address field immediately follows the beginning of frame and consists of 8-bits (MODBUS RTU). These bits indicate the user's assigned address of the slave device that is to receive the message sent by the attached master.

Each slave must be assigned a unique address and only the addressed slave will respond to a query that contains its address. When the slave sends a response, the slave address informs the master which slave is communicating.

The address value must be comprised between 0 and 99.

4.2.3 Function code field

The function code field tells the addressed slaves what function to perform. The table 4-3 lists the function codes, their meaning and the action they initiate.

Table 4-3 Function codes used by the recorder

Function code	Meaning	Action
03 / 04	READ REGISTERS	Obtain current binary value in one or more register.
06	PRESET SINGLE REGISTER	Place a specific value into a register.
08	LOOPBACK DIAGNOSTIC TEST	Diagnostic test message sent to slave to evaluate communication process.
16 (10h)	PRESET MULTIPLE REGISTERS	Place specific binary values into a series of consecutive holding registers.
20 (14h)	READ GENERAL REFERENCE	Reads information contained in memory files.
21 (15h)	WRITE GENERAL REFERENCE	Changes information contained in memory files.

The higher order bit in this field is set by the slave device to indicate that other than a normal response is being transmitted to the master device. (See Section 4.3 for a description of exception response). This bit remains zero if the message is a query or a normal response message.

4.2.4 Data field

The data field contains information needed by the slave to perform the specific function or it contains data collected by the slave in response to a query. This information may be values, address references, or limits. For example, the function code tells the slave to read a register, and the data field is needed to indicate which register to start at and how many to read. The embedded address and data information varies with the type and capacity of PC associated with the slave.

4.2.5 Error check field

The field allows the master and slave devices to check a message for errors in transmission. Sometimes, because of the electrical noise or other interference, a message may be changed slightly while it is on its way from one unit to another. The error detection assumes that the slave or master does not react to messages that have changed during transmission. This increases the safety and the efficiency of the Modbus system.

4.3 Exception responses

Programming or operation errors are those involving illegal data in a message, no response from a slave for example. These errors result in an exception response from either the master computer software (Modbus Communication Handler) or the slave, depending on the type of error. The exception response codes are listed in table 4-4. When a slave detects one of these errors, it sends a response message to the master consisting of slave address, function code, error code and error check fields. To indicate that the response is a notification of an error, the high order bit of the function code is set to 1.

Table 4-4 Exception response codes

Code	Name	Meaning
01	ILLEGAL FUNCTION	The message function received is not an allowable action for addressed slave.
02	ILLEGAL DATA ADDRESS	The address referenced in the data field is not an allowable address for the addressed slave location.
03	ILLEGAL DATA VALUE	The value referenced in the data field is not allowable in the addressed slave location.
06	BUSY, REJECTED MESSAGE	The message was received without error, but the slave is engaged in processing a long duration program command. Retransmit later.

4.4 Detailed explanation of used Modbus functions

4.4.1 Registers (codes 03 / 04h)

☑ QUERY

Function codes 03 / 04 obtain the contents of one or more register(s). The registers cannot be written by this function. They are numbered from zero up.

Table 4-5 Read register (Query)

		DATA		
ADDRESS	FUNCTION CODE	DATA START REGISTER HO	DATA START REGISTER LO	
02	04	02	00	

DATA (cont'd)			ERROR CHECK
	DATA NUMBER OF REG HO	DATA NUMBER OF REG LO	CRC
	00	02	7040

Always equal to zero for this application

☑ RESPONSE

The addressed slave responds with its address and the function code followed by the information field. The information field contains 1 byte describing the quantity of data byte to be returned. The contents of the registers requested (DATA) are 2 bytes each, with the binary content right justified within each pair of characters. The first byte includes the high order bits and the second, the low order bits.

Table 4-6 Read register (Response)

		DATA			
ADDRESS	FUNCTION	BYTE COUNT	DATA REGISTER 1		
02	04	04	42	50	

DATA (cont'd)		ERROR CHECK
DATA REGISTER 2		CRC
00	00	DCED

4.4.2 Loopback test (Code 08h)

☑ QUERY

The loopback test allows to evaluate the communication process.
The loopback test requests a simple return of the query message. (Diagnostic code 0000)

		DATA		
ADDRESS	FUNCTION	DATA DIAGNOSTIC CODE HO	DATA DIAGNOSTIC CODE LO	
01	08	00	00	

DATA (cont'd)			ERROR CHECK
	DATA VALUE HO	DATA VALUE LO	CRC
	2E	3E	7C7B

☑ RESPONSE

The loopback test response is a simple return of the query message. (Diagnostic code 0000)

		DATA		
ADDRESS	FUNCTION	DATA DIAGNOSTIC CODE HO	DATA DIAGNOSTIC CODE LO	
01	08	00	00	

DATA (cont'd)			ERROR CHECK
	DATA VALUE HO	DATA VALUE LO	CRC
	2E	3E	7C7B

4.4.3 Preset single register (code 06h)

☑ QUERY

Function 06 allows you to modify the contents of a register. Unused high order bits must be set to 0.

Table 4-7 Preset single register (Query)

		DATA		
ADDRESS	FUNCTION	REGISTER NUMBER HO	REGISTER NUMBER LO	
02	06	0A	01	

DATA (cont'd)		ERROR CHECK
DATA VALUE HO	DATA VALUE LO	CRC
00	01	1A21

☑ RESPONSE

The normal response to a preset single register request is to retransmit the query message after the register has been altered.

Table 4-8 Preset single register (Response)

		DATA		
ADDRESS	FUNCTION	REGISTER NUMBER HO	REGISTER NUMBER LO	
02	06	0A	01	

DATA (cont'd)		ERROR CHECK
DATA VALUE H0	DATA VALUE LO	CRC
00	01	1A21

4.4.4 Preset multiple registers (code 10h)

☑ QUERY

The master sends the address of the first register to be written (2 bytes), the number of data bytes transmitted (1 byte), the data bytes and the error check byte.

Table 4-9 Preset multiple register (Query)

		DATA		
ADDRESS	FUNCTION	REGISTER ADDRESS HO	REGISTER ADDRESS LO	
02	10	10	02	

DATA (cont'd)				
QUANTITY OF REGISTERS		BYTE COUNT	HO DATA	HO DATA
00	02	04	42	97
Always equal to zero for this application				

DATA (cont'd)			ERROR CHECK
	HO DATA	LO DATA	CRC
	33	33	4183

☑ RESPONSE

The normal response to a function 16 (10h) query is to echo the address, function code, starting address and number of registers to be loaded.

Table 4-10 Preset multiple register (Response)

		DATA		
ADDRESS	FUNCTION	HO ADDRESS	LO ADDRESS	
02	10	10	02	

DATA (cont'd)			ERROR CHECK
	QUANTITY		CRC
	00	02	E4FB

4.4.5 Read general reference (code 14h)

☑ QUERY

Several sub-requests can be included in one message. Each sub-request reads a contiguous group of registers.

➔ WARNING: ONLY ONE SUB-REQUEST (TO ACCESS THE ALARM SETPOINT) IS SUPPORTED BY THE COMMUNICATION

Table 4-11 Read general reference query message format

		DATA	
ADDRESS	FUNCTION	BYTE CNT	
B	B	B	

DATA (cont'd)			
SUB-REQUEST N			
REF TYPE	FILE NO	REQ ADDR	REG CNT
B	W	W	W

DATA (cont'd)				ERROR CHECK
SUB-REQUEST N+1				
REF TYPE	FILE NO	REQ ADDR	REG CNT	CRC
B	W	W	W	W (CRC)

B means Byte (8 bits long)
W means Word (16 bits long)

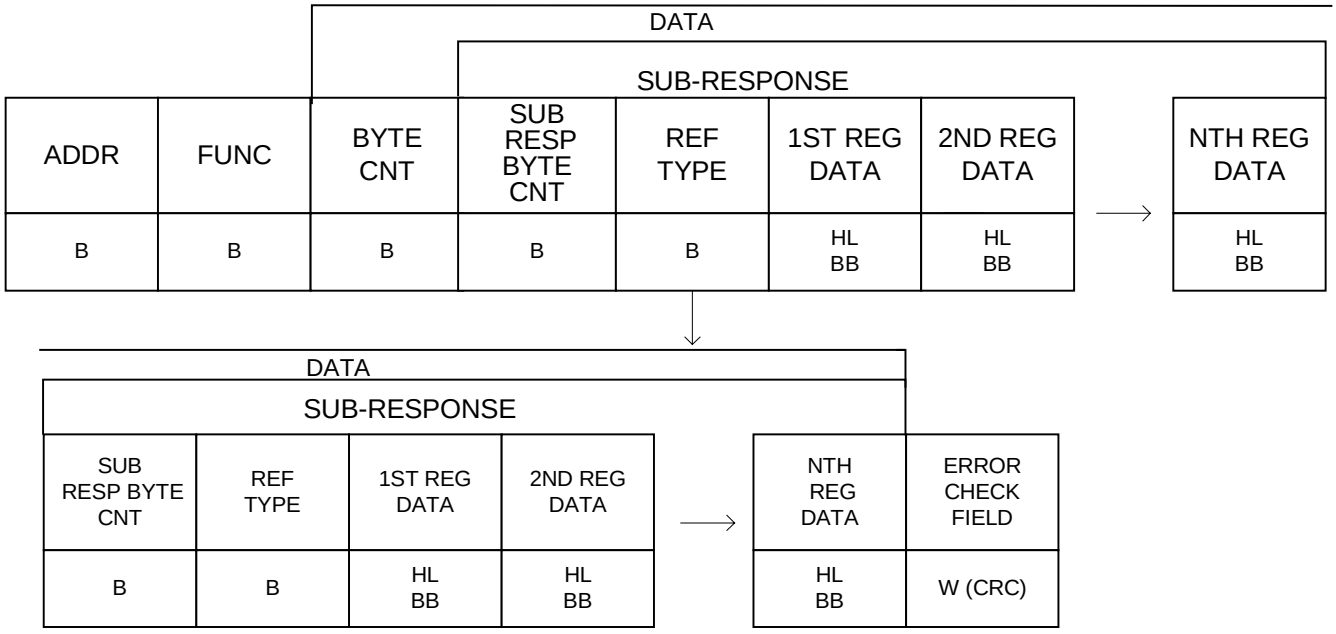
Byte count:
The total number of bytes in the read general reference response message, excluding the address, function code, byte count, and the error check fields; that is all occurrences of the following: sub-response byte count, reference type, and the first through the nth register in each sub-response.

Reference type and File N°:
Equal to zero for this application.

☑ RESPONSE

One read general reference query message can result in one or more sub-responses. The addressed slave responds with its own address, the function code, and the total byte count of one or more sub-responses. Each sub-response contains the byte count of that sub-response, its reference type, and the response data. The error check field follows the last sub-response.

Table 4-12 Read general reference response message format



Byte count:
The total number of bytes contained in all the sub-responses.

Sub-Response byte count:
The number of bytes in its sub-response.

Reference type and File N°:
Must be equal to zero for this application.

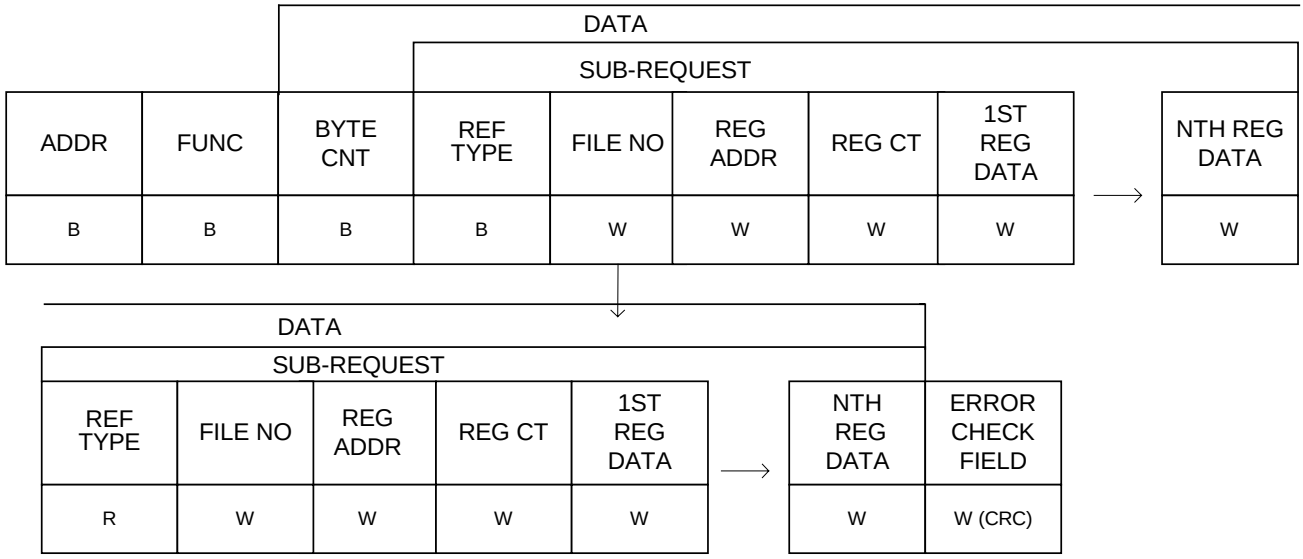
Error check:
The cyclic redundancy check if using RTU transmission.

4.4.6 Write general reference (Code 15h)

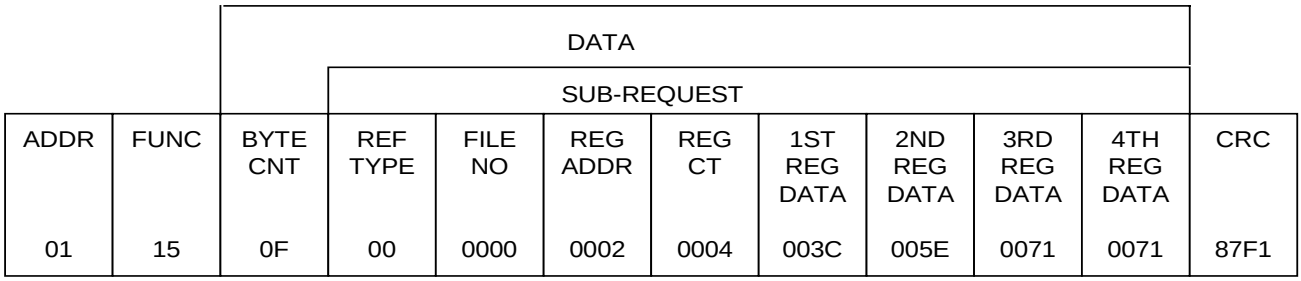
☑ QUERY

➔ WARNING: ONLY ONE SUB-REQUEST (TO ACCESS THE ALARM SETPOINT) IS SUPPORTED BY THE COMMUNICATION

Table 4-13 Write general reference query message format



EXAMPLE:



Byte count:
The total number of bytes in the write general reference message, excluding the address, function code, byte count, and the error check fields; that is all occurrences of the following: reference type, file number, register address, register count, and the first through the last register in the write general reference query message.

☑ RESPONSE

The normal response to a write general reference query message is the retransmission of the write request.

Table 4-14 Write general reference response message

		DATA								
			SUB-REQUEST							
ADDR	FUNC	BYTE CNT	REF TYPE	FILE NO	REG ADDR	REG CT	1ST REG DATA	2ND REG DATA	3RD REG DATA	4TH REG DATA
B	B	B	B	W	W	W	W	W	W	W

↓

DATA							
SUB-REQUEST							
REF TYPE	FILE NO	REG ADDR	REG CT	1ST REG DATA	2ND REG DATA	3RD REG DATA	ERROR CHECK FIELD
B	B	W	W	W	W	W	CRC

Reference type:
Equal to zero for this application.

File number:
Equal to zero for this application.

Starting register address:
The address of the first register in which information is to be entered or changed.

Register count:
The number of registers in which information will be entered or changed.

5. RTU FUNCTIONS

5.1 Loopback

This frame tests the communication system to know if the communication with the recorder is correct. Variations in the response may indicate faults in the modbus system. The information field contains 2 bytes for the designation of diagnostic code followed by 2 bytes for the information field.

☑ Function code : 08h
☑ Data diagnostic code : 0000 (Return query data)
☑ Data field : A537h for example

EXAMPLE:

01	**08**	**00**	**00**	**A5**	**37**	**DA8D**

☑ Request : loopback test return query data
- 01 : address
- 08 : function code
- A537 : data example
- DA8D : CRC

01	**08**	**00**	**00**	**A5**	**37**	**DA8D**

☑ Response : loopback test return query data response
- 01 : address
- 08 : function code
- 0000
- A537 : data example
- DA8D : CRC

5.2 Read process values

Reading of any of the analog, com. or maths process values.

☑ Function : read (code 03 / 04h)
☑ Address range : 1800h to 18FEh
☑ Number of registers : 64 max.

ADDRESS	TRANSMITTED REGISTERS	
	180 mm recorder	**250 mm recorder**
1800h	ANALOG 1 float IEEE MSB	
1801h	ANALOG 1 float IEEE LSB	
1802h	ANALOG 2 float IEEE MSB	
--	--	
--	--	
182Fh	ANALOG 24 float IEEE LSB	
1830h	(reserved)	ANALOG 25 float IEEE MSB
--	(reserved)	--
--	(reserved)	--
187Fh	(reserved)	ANALOG 64 float IEEE LSB
1880h	COM 1 float IEEE MSB	
1881h	COM 1 float IEEE LSB	
--	--	
18AFh	COM 24 float IEEE LSB	
18B0h	(reserved)	COM 25 float IEEE MSB
--	(reserved)	--
--	(reserved)	--
18BFh	(reserved)	COM 32 float IEE LSB
18C0h	MATH 1 float IEEE MSB	
18C1h	MATH 1 float IEEE LSB	
--	--	
18EFh	MATH 24 float IEEE LSB	
18F0h	(reserved)	MATH 25 float IEEE MSB
--	(reserved)	--
--	(reserved)	--
18FFh	(reserved)	MATH 32 float IEEE LSB

Note: see appendix D for IEEE information.

➔ **WARNING:** - the register address has to be EVEN. It is impossible to read only LSB of variable.
- the number of registers has to be EVEN and different from 0. Otherwise an invalid address error code will be returned. **This number also has to be lower than 64.**

EXAMPLE:

01	04	18	02	00	02	D6AB

☑ Request : read analog 2 value
- 01 : address
- 04 : function code
- 1802 : register start address
- 0002 : number of registers
- D6AB : CRC

01	04	04	42	5D	47	AE	CC62

☑ Response : analog 2 = 55.32 decimal
- 01 : address
- 04 : function code
- 04 : number of bytes
- 425D47AE : analog 2 (55.32)
- CC62 : CRC

5.3 Read digital process values

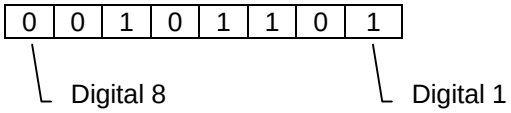
Reading of any digital inputs.

☑ Function : read (code 03 / 04h)
☑ Address range : 1A00h to 1A04h
☑ Number of registers : 5 max.

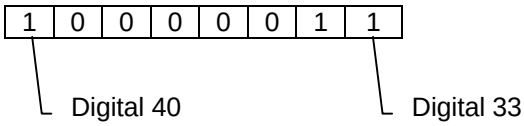
ADDRESS	TRANSMITTED REGISTERS			
	180 mm	**250 mm**	**180 mm**	**250 mm**
1A00h	digitals 1 to 8		digitals 9 to 16	
1A01h	digitals 17 to 24		digitals 25 to 32	
1A02h	digitals 33 to 36	digitals 33 to 40	(reserved)	digitals 41 to 48
1A03h	(reserved for future application)			
1A04h	(reserved for future application)			

☑ Byte meaning:

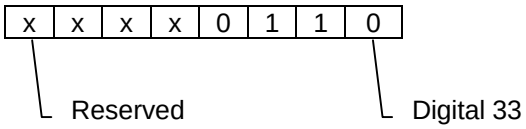
byte 1



byte 5 (for **250 mm**)



byte 5 (for **180 mm**)



EXAMPLE:

01	**04**	**1A**	**01**	**00**	**01**	**6712**

☑ Request : read value of digital inputs 17 to 32
- 01 : address
- 04 : function code
- 1A02 : register start address
- 0001 : number of registers
- D830 : CRC

01	**04**	**01**	**18**	**0A**	**3337**

☑ Response: operation performed

18h: 0001 1000 digitals 20 and 21 are equal to 1
0Ah: 0000 1010 digitals 26 and 28 are equal to 1

- 01 : address
- 04 : function code
- 02 : number of bytes
- 180A : value for digital inputs 17 to 32
- C18E : CRC

5.4 Send communication process values

Overwrite any of the communication process values existing on the recorder.

☑ Function : write (code 10h)
☑ Address range : the area is duplicated into 2 address ranges :
1000h to 103Eh
or 1880h to 18BEh
☑ Number of registers : 64 max.

ADDRESS		TRANSMITTED REGISTERS	
Range 1	Range 2	**180 mm**	**250 mm**
1000h	1880h	Write COM 1 float IEEE MSB	
1001h	1881h	Write COM 1 float IEEE LSB	
1002h	1882h	Write COM 2 float IEEE MSB	
--	--	--	
--	--	--	
102Fh	18AFh	Write COM 24 float IEEE LSB	
1030h	18B0h	(reserved)	WRITE COM 25 float IEEE MSB
--	--	(reserved)	--
--	--	(reserved)	--
103Fh	18BFh	(reserved)	WRITE COM 32 float IEEE LSB

Note: see appendix D for IEEE information.

➔ **WARNING:** - The register address has to be EVEN. It is impossible to read only LSB of variable.
- The number of registers has to be EVEN and different from 0. Otherwise an invalid address error code will be returned.

EXAMPLE:

01	**10**	**10**	**02**	**00**	**04**	**08**	**42**	**82**	**3D**	**71**	**41**	**46**	**14**	**7B**	**94E0**

☑ Request : Force COM 2 to 65.12 (42823D71h)
Force COM 3 to 12.38 (4146147Bh)

- 01 : address
- 10 : function code
- 1002 : register start address
- 0004 : number of registers
- 08 : number of bytes to be written
- 42823D71 : COM2
- 4146147B : COM3
- 94E0 : CRC

01	**10**	**10**	**02**	**00**	**04**	**64CA**

☑ Response : operation performed

- 01 : address
- 10 : function code
- 1002 : register start address
- 0004 : number of registers
- 64CA : CRC

5.5 Print a message

Allows you to print a message of maximum 64 (replace 64 by 50 for a 180 mm recorder) characters on the paper. The authorized characters are given in appendix A.

☑ Function : write (code 10h)
☑ Address range : 0300h to 030Fh
☑ Number of registers : 32 max.

ADDRESS	TRANSMITTED REGISTERS			
	180 mm	**250 mm**	**180 mm**	**250 mm**
0300h	1st character		2nd character	
0301h	3rd character		4th character	
0302h	5th character		6th character	
0303h	7th character		8th character	
--	--		--	
--	--		--	
0318h	49th character		50th character	
0319h	(reserved)	51st character	(reserved)	52nd character
--	(reserved)	--	(reserved)	--
--	(reserved)	--	(reserved)	--
031Fh	(reserved)	63rd character	(reserved)	64th character

➔ **WARNING:** - The number of characters has to be EVEN.
- The "C:" message is included in the message length. That is to say that the user message is actually limited to 62 (48) characters. A longer message will be truncated to the authorized limit.
- Starting address must be 0300h otherwise an invalid address error code will be returned.

The message will be printed under the following format: C: User text

EXAMPLE: "C : This is a user-message"

The character @ is used to place the date or time of the recorder in the user-message:

- @d will be replaced by the current date.
- @h will be replaced by the current time.
- @e the message will be printed on trace (by default, the message will be printed on blank).

When using @d or @h, keep in mind that the date and time will take respectively 9 characters and 5 characters.

EXAMPLE 1:

01	10	03	00	00	04	08	30	31	32	33	

	34	35	36	37	D830

☑ Request : send message "01234567"

- 01 : address
- 10 : function code
- 0300 : register start address
- 0004 : number of registers
- 08 : number of bytes
- 3031323334353637 : message to be printed
- D830 : CRC

01	10	03	00	00	04	C18E

☑ Response : operation performed

- 01 : address
- 10 : function code
- 0300 : register start address
- 0004 : number of registers
- C18E : CRC

EXAMPLE 2:

01	10	03	00	00	05	0A	40	64	20	40	68	

	20	44	44	44	44	77CA

☑ Request : send message "31 AUG 97 14:50 DDDD" on blank

- 01 : address
- 10 : function code
- 0300 : register start address
- 0005 : number of registers
- 0A : number of bytes
- 40642040682044444444 : message "@d @h DDDD"
- 77CA : CRC

01	10	03	00	00	05	004E

☑ Response : operation performed

- 01 : address
- 10 : function code
- 0300 : register start address
- 0005 : number of registers
- 004E : CRC

EXAMPLE 3:

01	10	03	00	00	05	0A	40	64	20	40	68	

	20	44	44	40	65	B512

☑ Request : send message "31 AUG 97 14:50 DD" on trace

- 01 : address
- 10 : function code
- 0300 : register start address
- 0005 : number of registers
- 0A : number of bytes
- 40642040682044444065 : message "@d @h DD@e"
- B512 : CRC

01	10	03	00	00	05	004E

☑ Response : operation performed

- 01 : address
- 10 : function code
- 0300 : register start address
- 0005 : number of registers
- 004E : CRC

5.6 Alarm status

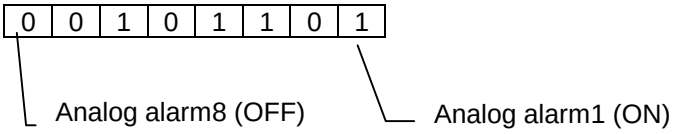
Indicates the alarm status ON or OFF of 64 analog alarms, 48 digital alarms and 6 events alarms and which analog, communication or math channels are in alarm.

☑ Function : Read (code 03 / 04h)
☑ Address range : 0100h to 0110h
☑ Number of registers : 17 max
☑ Bytes bit 0 = alarm off
bit 1 = alarm on

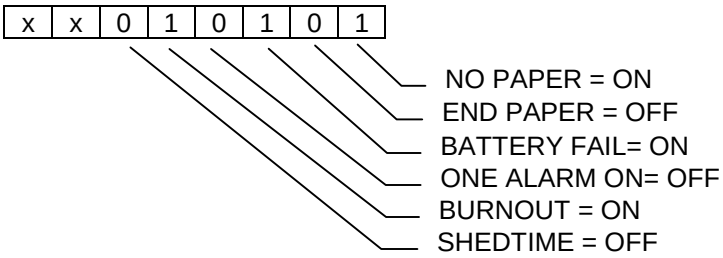
ADDRESS	TRANSMITTED REGISTERS			
	180 mm	**250 mm**	**180 mm**	**250 mm**
0100h	analog al. 1 to 8		analog al. 9 to 16	
0101h	analog al. 17 to 24		(reserved)	analog al. 25 to 32
0102h	(reserved)	analog al. 33 to 40	(reserved)	analog al. 41 to 48
0103h	(reserved)	analog al. 49 to 56	(reserved)	analog al. 57 to 64
0104h	digital al. 1 to 8		digital al. 9 to 16	
0105h	digital al. 17 to 24		digital al. 25 to 32	
0106h	digital al. 33 to 36	digital al. 33 to 40	(reserved)	digital al. 41 to 48
0107h	event al. 1 to 6		(reserved)	
0108h	(reserved)		(reserved)	
0109h	analog channel in al. 1 to 8		analog channel in al. 9 to 16	
010Ah	analog channel in al. 17 to 24		(reserved)	analog ch. in al. 25 to 32
010Bh	(reserved)	analog ch. in al. 33 to 40	(reserved)	analog ch. in al. 41 to 48
010Ch	(reserved)	analog ch. in al. 49 to 56	(reserved)	analog ch. in al. 57 to 64
010Dh	com channel in al. 1 to 8		com channel in al. 9 to 16	
010Eh	com channel in al. 17 to 24		(reserved)	com ch. in al. 25 to 32
010Fh	math channel in al. 1 to 8		math channel in al. 9 to 16	
0110h	math channel in al. 17 to 24		(reserved)	math ch. in al. 25 to 32

☑ Byte meaning:

Byte 1



Byte 15 (alarm event)



EXAMPLE:

01	04	01	00	00	02	7037

☑ Request : what is the analog alarm status from 1 to 32 ?

- 01 : address
- 04 : function code
- 0100 : register start address
- 0002 : number of registers

01	04	04	0F	03	31	00	1D00

☑ Response:

0Fh : 0000 1111 : Alarms 1 to 4 are ON
03h : 0000 0011 : Alarms 9 and 10 are ON
31h : 0011 0001 : Alarms 17, 21 and 22 are ON
00h : 0000 0000 : Alarms 25 to 32 are OFF

- 01 : address
- 04 : function code
- 04 : number of bytes
- 0F033100 : analog alarm status
- 1D00 : CRC

5.7 Print process values (snapshot log)

This frame starts the print-out of the trace, analog, math or digital PVs on the paper.

☑ Function code : write (code 06h)
☑ Address range : 0A01h
☑ Number of registers : 1
☑ Possible values : 00 = print Trace
01 = print Analog
02 = print Digital
03 = print Math

EXAMPLE:

01	**06**	**0A**	**01**	**00**	**01**	**1A12**

☑ Request : print the analog PV's on the paper.
- 01 : address
- 06 : function code
- 0A01 : start register address
- 0001 : data value
- 1A12 : CRC

01	**06**	**0A**	**01**	**00**	**01**	**1A12**

☑ Response : operation performed
- 01 : address
- 06 : function code
- 0A01 : start register address
- 0001 : data value
- 1A12 : CRC

5.8 Printer status

This frame requests information concerning the printer status and is fixed. Different parameters will induce an invalid error code.

☑ Function code : Read (code 03 / 04h)
☑ Register start address : 0800h
☑ Number of registers : 4

ADDRESS	TRANSMITTED REGISTERS	
0800h	(reserved)	Cassette state
0801h	Speed used	Printer mode
0802h 0803h	Remaining paper length float IEEE MSB Remaining paper length float IEEE LSB	

Note: see appendix D for IEEE information.

☑ Byte meaning

Byte :	0	Reserved	00
	1	Cassette state	Value: 00 = cassette out 01 = cassette in
	2	Paper speed	Value: 00 = speed 1 used 01 = speed 2 used
	3	Printer mode	Value: 00 = printer inhibit 01 = print mode
	4 - 7	Remaining paper	Value: 4 bytes in mm Value of remaining paper length IEEE format

EXAMPLE:

01	04	08	00	00	04	F3A9

☑ Request : information about the printer?

- 01 : address
- 04 : function code
- 0800 : register start address
- 0004 : number of registers
- F3A9 : CRC

01	04	08	00	01	01	01	46	AE	92	00	1115

☑ Response:

- 00: reserved
- 01: cassette in
- 01: speed 2 used
- 01: print mode
- remaining paper length : 22.3 m

- 01 : address
- 04 : function code
- 08 : number of bytes
- 0001010146AE9200 : printer status
- 1115 : CRC

5.9 Relay status

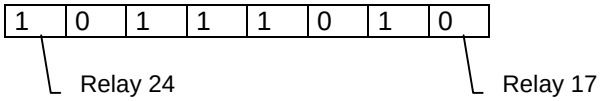
This frame is used to read from 1 to 48 relay status.

☑ Function code : read (code 03 / 04h)
☑ Register address : 0C00h
☑ Number of registers : 3 max

ADDRESS	TRANSMITTED REGISTERS			
	180 mm	**250 mm**	**180 mm**	**250 mm**
0C00h	relays 1 to 8		relays 9 to 16	
0C01h	relays 17 to 24		relays 25 to 32	
0C02h	relays 33 to 36	relays 33 to 40	(reserved)	relays 41 to 48

☑ Byte meaning

byte 3:



EXAMPLE:

01	**04**	**0C**	**00**	**00**	**01**	**329A**

☑ Request : what is the status of relays 1 to 16 ?
- 01 : address
- 04 : function code
- 0C00 : register start address
- 0001 : number of registers
- 329A : CRC

01	**04**	**02**	**00**	**35**	**7927**

☑ Response : 35h = 00110101: relays 1, 3, 5 and 6 are active
- 01 : address
- 04 : function code
- 02 : number of bytes
- 0035 : relay status
- 7927 : CRC

5.10 Read alarm setpoints (function code 03 / 04h)

Reading of any of the alarm setpoints of the recorder.

☑ Function code : read (code 03 / 04h)
☑ Address range : 1C00h 1C7Eh
☑ Number of registers : 64 max.

ADDRESS	TRANSMITTED REGISTERS	
	180 mm	**250 mm**
1C00h	Alarm 1 setpoint float IEEE MSB	
1C01h	Alarm 1 setpoint float IEEE LSB	
1C02h	Alarm 2 setpoint float IEEE MSB	
--	--	
--	--	
1C5Fh	Alarm 48 setpoint float IEEE LSB	
1C60h	(reserved)	Alarm 49 setpoint float IEEE MSB
--	(reserved)	--
--	(reserved)	--
1C7Fh	(reserved)	Alarm 64 setpoint float IEEE LSB

Note : See appendix D for IEEE information.

EXAMPLE:

01	**04**	**1C**	**02**	**00**	**02**	**D6AB**

☑ Request : read alarm 2 setpoint value
- 01 : address
- 04 : function code
- 1C02 : register start address
- 0002 : number of registers
- D6AB : CRC

01	**04**	**04**	**42**	**5D**	**47**	**AE**	**CC62**

☑ Response : alarm 2 setpoint = 55.32 decimal
- 01 : address
- 04 : function code
- 04 : number of bytes
- 425D47AE : alarm 2 setpoint 2 (55.32)
- CC62 : CRC

5.11 Write alarm setpoints (function code 10h)

Overwrite any of the alarm setpoints in the recorder configuration.

☑ Function code : write (code 10h)
☑ Address range : 1C00h to 1C7Fh
☑ Number of registers : 64 max.

ADDRESS	TRANSMITTED REGISTERS	
	180 mm	**250 mm**
1C00h	Alarm 1 setpoint float IEEE MSB	
1C01h	Alarm 1 setpoint float IEEE LSB	
1C02h	Alarm 2 setpoint float IEEE MSB	
--	--	
--	--	
1C5Fh	Alarm 48 setpoint float IEEE LSB	
1C60h	(reserved)	Alarm 49 setpoint float IEEE MSB
--	(reserved)	--
--	(reserved)	--
1C7Fh	(reserved)	Alarm 64 setpoint float IEEE LSB

Notes : see appendix D for IEEE information.

EXAMPLE:

01	**10**	**1C**	**02**	**00**	**04**	**08**	**42**	**82**	**3D**	**71**	**41**	**46**	**14**	**7B**	**94E0**

☑ Request : force alarm 2 setpoint to 65.12 (42823D71h)
force alarm 3 setpoint to 12.38 (4146147Bh)

- 01 : address
- 10 : function code
- 1C02 : register start address
- 0004 : number of registers
- 08 : number of bytes to be written
- 42823D71 : alarm 2 setpoint
- 4146147B : alarm 3 setpoint
- 94E0 : CRC

01	**10**	**1C**	**02**	**00**	**04**	**679A**

☑ Response : operation performed

- 01 : address
- 10 : function code
- 1C02 : register start address
- 0004 : number of registers
- 679A : CRC

5.12 Read alarm setpoints (function code 14h)

This parameter allows to read a single alarm setpoint of the recorder. Different parameters will induce an invalid error code.

☑ Function code : read (code 14h)
☑ Reference type : 00
☑ Number of files : 0000
☑ Register address
(must be even) : 0000h to 007Eh (see appendix D)
☑ Register count : 0002

ADDRESS	TRANSMITTED REGISTERS	
	180 mm	**250 mm**
0000h	Alarm 1 setpoint float IEEE MSB	
0001h	Alarm 1 setpoint float IEEE LSB	
0002h	Alarm 2 setpoint float IEEE MSB	
--	--	
--	--	
005Fh	Alarm 48 setpoint float IEEE LSB	
0060h	(reserved)	Alarm 49 setpoint float IEEE MSB
--	(reserved)	--
--	(reserved)	--
007Fh	(reserved)	Alarm 64 setpoint float IEEE LSB

Note : See appendix D for IEEE information.

EXAMPLE:

01	**14**	**07**	**00**	**00**	**00**	**00**	**08**	**00**	**02**	**9F27**

☑ Request : read alarm 5 setpoint
- 01 : address
- 14 : function code
- 07 : number of bytes
- 00 : reference type
- 0000 : number of files
- 0008 : register start address
- 0002 : number of registers
- 9F27 : CRC

01	**14**	**06**	**05**	**00**	**41**	**DA**	**CC**	**CD**	**C098**

☑ Response : alarm 5 setpoint = 27.35 (41DACCCDh)
- 01 : address
- 14 : function code
- 06 : number of bytes
- 05 : number of bytes in its sub-response
- 00 : reference type
- 41DACCCD : alarm 5 setpoint
- C098 : CRC

5.13 Write alarm setpoints (function code 15h)

This frame is used to modify a single alarm setpoint in the recorder configuration. Different parameters will induce an invalid error code.

☑ Function code : write (code 15h)
☑ Reference type : 00
☑ Number of files : 0000
☑ Register address range
(must be even) : 0000h to 007Eh (see appendix D)
☑ Register count : 0002

➔ **WARNING: See CONFIGURATION LOCK parameter before any use of the WRITE ALARM SETPOINT parameter.**

ADDRESS	TRANSMITTED REGISTERS	
	180 mm	**250 mm**
0000h	Alarm 1 setpoint float IEEE MSB	
0001h	Alarm 1 setpoint float IEEE LSB	
0002h	Alarm 2 setpoint float IEEE MSB	
--	--	
--	--	
005Fh	Alarm 48 setpoint float IEEE LSB	
0060h	(reserved)	Alarm 49 setpoint float IEEE MSB
--	(reserved)	--
--	(reserved)	--
007Fh	(reserved)	Alarm 64 setpoint float IEEE LSB

Notes : see appendix D for IEEE information.

EXAMPLE:

01	**15**	**0B**	**00**	**00**	**00**	**00**	**08**	**00**	**02**	**41**	**09**	

	99	**9A**	**A28B**

☑ Request : write alarm 5 setpoint = 8.6 (4109999Ah)

- 01 : address
- 15 : function code
- 0B : number of bytes
- 00 : reference type
- 0000 : number of files
- 0008 : register start address
- 0002 : number of registers
- 4109999A : alarm 5 setpoint
- A28B : CRC

01	15	0B	00	00	00	00	08	00	02	41	09	

	99	9A	A28B

☑ Response : the normal response to a write general reference query message is the retransmission of the write request.

- 01 : address
- 15 : function code
- 0B : number of bytes
- 00 : reference type
- 0000 : number of files
- 0008 : register start address
- 0002 : number of registers
- 4109999A : alarm 5 setpoint
- A28B : CRC

5.14 Configuration lock

This frame allows to lock or unlock the configuration. With a lock-set on the recorder, configuration parameters cannot be modified when using keyboard or jack connection. For this reason, the "LOCK" frame is required before any configuration changes.

☑ Function code : write (code 06h)
☑ Register value : 1
☑ Register number : 2E00h unlock the configuration access
2E01h lock the configuration access

EXAMPLE:

01	**06**	**2E**	**01**	**00**	**01**	**10E2**

☑ Request : lock configuration access request
- 01 : address
- 06 : function code
- 2E01 : register start address (LOCK request)
- 0001 : (reserved)
- 10E2 : CRC

01	**06**	**2E**	**01**	**00**	**01**	**10E2**

☑ Response : operation performed
- 01 : address
- 06 : function code
- 2E01 : register start address
- 0001 : (reserved)
- 10E2 : CRC

➔ **WARNING:** When you reboot your recorder, the lock will not be active.

5.15 Report Slave ID (function code11h)

Function code 17 (11h) is used to report the Device Information which includes information like :
Slave ID, device description and firmware version.

☑ Function code : 11h
☑ Byte count : in the response only
☑ Data field : in the response only

EXAMPLE:

01	**11**	**C0**	**2C**

☑ Request : **Report Slave ID**
- 01 : address
- 11 : function code
- C0 2C : CRC

01	**11**	**33**	**25**	**FF**	**44**	**50**	**52**	**32**	**35**	**30**	**20**
30	**30**	**31**	**41**	**4B**	**20**	**20**	**20**	**00**	**00**	**00**	**06**
00	**18**	**00**	**00**	**40**	**01**	**00**	**00**	**00**	**08**	**02**	**1A**	**00**	**00**	**30**
03	**0C**	**00**	**00**	**30**	**06**	**18**	**C0**	**00**	**20**	**08**	**1C**	**00**	**00**	**40**
069F											

☑ Response : **device information**
- 01 : address
- 11 : function code
- 33 : byte count
- 25 : slave ID from DPR250
- FF : run Indicator Status ON
- 44505232353020303031414B : DPR250 001AK (device tag + software version)
- 20202000 : 3 spaces + end of string
- 00 : Model ID
- 00 : device class ID (generic class 00 -> fixed address mapable)
- 06 : number of recorders (start of device mapping)
- 00 : **analog inputs type**
- 1800 : starting address of data
- 0040 : number of data
- 01 : **analog outputs type**
- 0000 : starting address of data (not defined at this moment)
- 0008 : number of data
- 02 : **discrete input type**
- 1A00 : starting address of data
- 30 : number of data
- 03 : **discrete output type**
- 0C00 : starting address of data
- 0030 : number of data

- 06 : **maths type**
- 18C0 : starting address of data
- 0020 : number of data
- 08 : **alarms type**
- 1C00 : starting address of data
- 0040 : number of data
- 069F : CRC

6. ASCII PROTOCOL DEFINITION

6.1 Interface functions

☑ Transmission system	: asynchronous transmission system with start-stop bits.
☑ Start bit	: 1 bit
☑ Stop bit	: 1 bit
☑ Parity	: even, odd, no parity
☑ Bit per character	: 8 bits including parity
☑ Baud rate	: 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 bits per second
☑ Electrical signal characteristics	: E.I.A. (Electronic Industries Association) - standard RS232C, RS422 and RS485.

6.2 Protocol

The protocol used for the universal communication option is based on the modified transparent mode protocol. This allows the recorder to be connected on the same serial multi-drop link RS422/RS485 as the regulator.

6.3 ASCII code set

The universal communications option uses the American Standard Code for Information Interchange (ASCII).

The characters used are:

- upper case, alphabetic characters (A-F) for hexadecimal values
- numeric characters (0-9)
- certain symbols such as comma (,), carriage return, line feed
- only for the information field of the loopback protocol are all ASCII characters allowed.

6.4 Loopback

The loopback protocol is provided for link tests. With this message exchange you can test the communication link between your computer and the recorder. The host computer sends a series of ASCII characters to the recorder, and the recorder will return the characters it received to the host computer.

6.5 Checksum

There is an optional transaction called "checksum" which is used to increase security on the link. Used with any message exchange, it enables both your computer and your recorder to detect messages that have been corrupted by line noise.

6.6 Message exchange

Each communication takes place as a message exchange: your computer sends a request message (ASCII characters), and then waits for the resulting response (ASCII characters) from the device involved. Your computer is the host, it initiates the message exchange. The recorder is a response only device. When you send a READ request, the recorder responds with the data requested. When you send a WRITE request, the recorder responds with a message advising whether the operation has been performed or not.

6.6.1 Request messages

Request messages are composed of standard fields, separated by commas. Each field contains a specific kind of information which must be entered in the specified order to obtain a valid request message.

6.6.1.1 Station address

A two digit device address in decimal value - from **00** to **99** - identifies the specific device you are addressing. You must assign a unique station address to each device on the link.

6.6.1.2 Protocol field

A four digit number selects whether or not you are going to use a Checksum Protocol with your message exchange.

☑ **4204**: selects checksum protocol
☑ **0204**: ignores checksum protocol

When a message contains other values in the protocol field, the recorder will not respond at all even if the address is correct.

6.6.1.3 Function code field

Two ASCII characters indicate an hexadecimal code telling the kind of operation to perform.

☑ **01**: READ from variable
☑ **02**: WRITE to variable
☑ **03**: READ of configuration data
☑ **04**: WRITE to configuration data
☑ **05**: service
☑ **8A**: loopback message

6.6.1.4 Parameter code field

Two ASCII characters indicate the hexadecimal value of the parameter to access. For loopback protocol, this field does not exist (see examples).

6.6.1.5 Data type field

A one character field specifies the format or data type of the data field.

☑ **0**: hexadecimal values in ASCII representation
☑ **D**: ASCII characters in data fields (loopback only)

6.6.1.6 Number field

Two ASCII characters contain the hexadecimal number of values which have to be read or written. This number depends on the number of values and the data-field length of each parameter.

Restrictions for this number:

1. Minimum value is 01.
2. The number must be smaller or equal to the maximum number of values.

The above condition should be met to obtain a valid request.

6.6.1.7 Starting index field

A two ASCII character field contains the hexadecimal number of the first value which has to be read or written.

The starting index depends on the number in the previous number field, the number of values and the data field length for the specific parameter.

Restrictions for the index number:

1. The index number should be smaller or equal to maximum number of values.
2. The minimum value is 01.
3. The sum of the starting index and number must be smaller or equal to the maximum number of values+1.

6.6.1.8 Data field

This field with variable length contains only data in case of a WRITE request. The two bytes between commas represent two hexadecimal digits in the ASCII representation (i.e. if you want to send the B2h hexadecimal value, the two bytes sent will be 42h 32h).

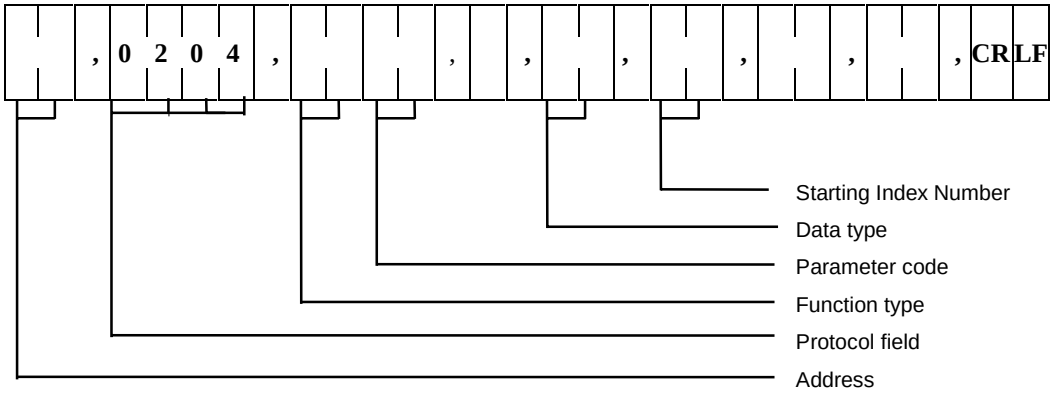
6.6.1.9 Checksum field

This field is a one byte hexadecimal value (two ASCII characters) representing the binary sum of all previous characters. This field has to be present when a checksum protocol is selected in the protocol field.

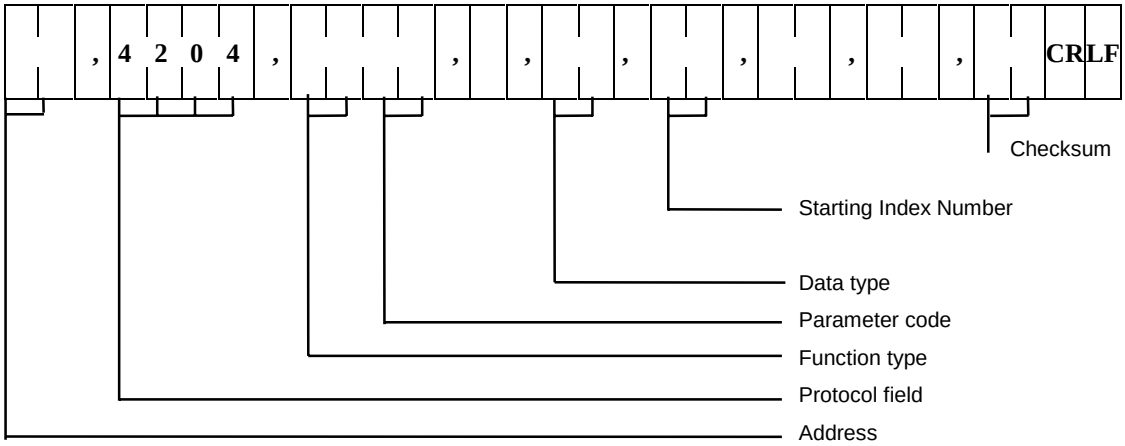
6.6.1.10 Carriage return (0Dh) / line feed (0Ah)

☑ **CR LF**: Terminates a message.

6.6.1.10.1 General request message without checksum



6.6.1.10.2 General request message with checksum



6.6.2 Response message

The response message returns data in case of a READ operation and tells your computer status or value of the operation initiated by the request message.

6.6.2.1 Request message status code

A two digit code indicates whether or not the present request message has been successfully processed.

- ☑ **00**: operation performed
- ☑ **01**: invalid request
- ☑ **02**: invalid format
- ☑ **04**: invalid checksum, parity or framing error
- ☑ **05**: invalid mode
- ☑ **06**: data out of range in the data field of a write in configuration

Refer to *Appendix C, "TROUBLESHOOTING"* to resolve invalid requests.

6.6.2.2 Device status

A two digit code indicates whether or not the addressed recorder is working correctly and has performed the requested operation.

- ☑ **00**: recorder is working correctly
- ☑ **01**: problem detected

6.6.2.3 Device mode

A two digit code indicates the mode of the recorder.

- ☑ **00**: reserved
- ☑ **01**: RUN mode (The recorder is able to print and measure data)
- ☑ **02**: reserved
- ☑ **03**: DEF mode (No printing, no measure)
- ☑ **04**: reserved
- ☑ **05**: reserved
- ☑ **06**: CAL PAP MODE (Measure OFF, printing ON)
- ☑ **07**: reserved

6.6.2.4 Data field

Contains information in case of loopback or READ request. The different two ASCII characters are separated by a comma. For more information, see each parameter code.

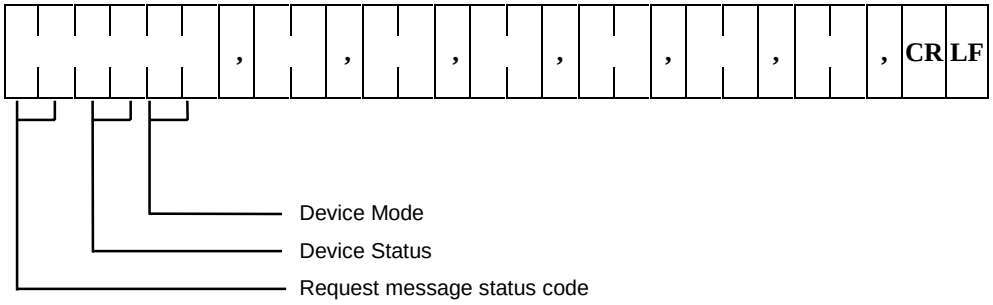
6.6.2.5 Checksum field (optional)

This field is a one byte hexadecimal value (two ASCII characters) representing the binary sum of all previous characters. This field is only present in the response when the protocol field in the request contains 4204.

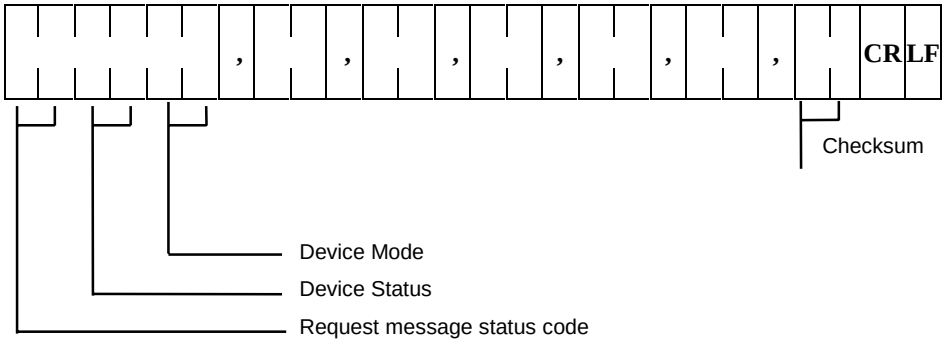
6.6.2.6 Carriage return/line feed

Terminates a message.

6.6.2.6.1 General response message without checksum protocol



6.6.2.6.2 General response message with checksum protocol

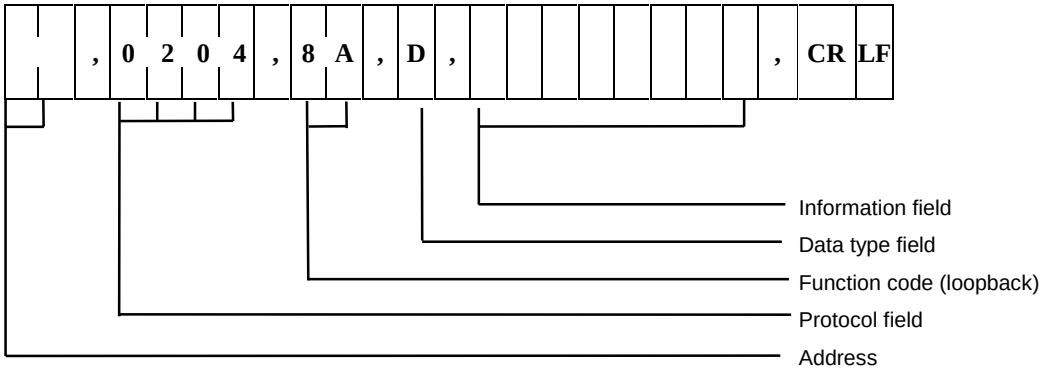


6.7 Loopback request and response

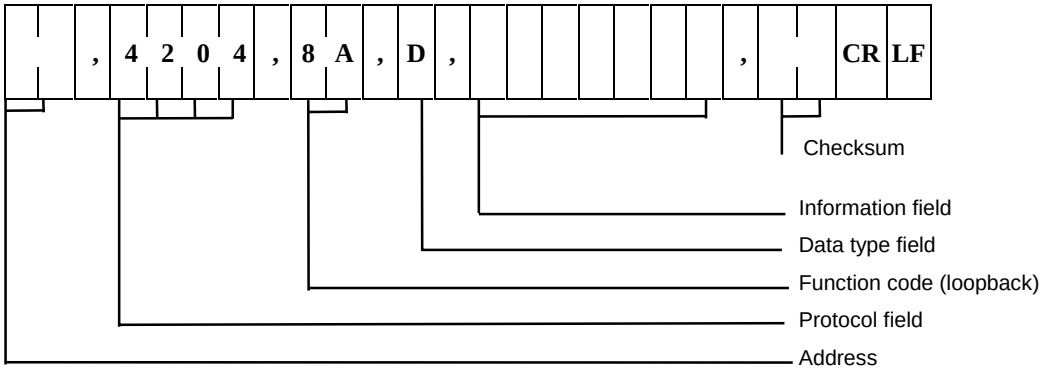
The loopback protocol can be used with or without the checksum protocol. In the information field, all ASCII characters are allowed, except the CR character (carriage return).

6.7.1 Loopback request

6.7.1.1 Loopback request message without checksum

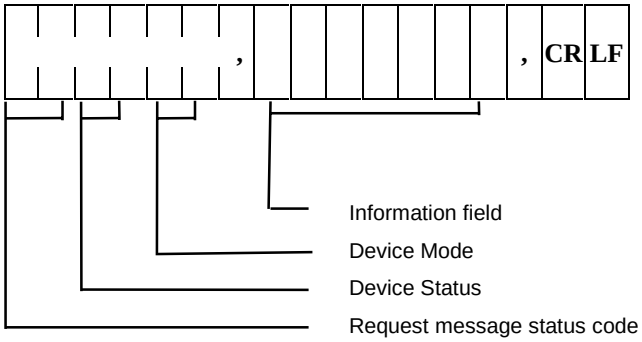


6.7.1.2 Loopback request message with checksum

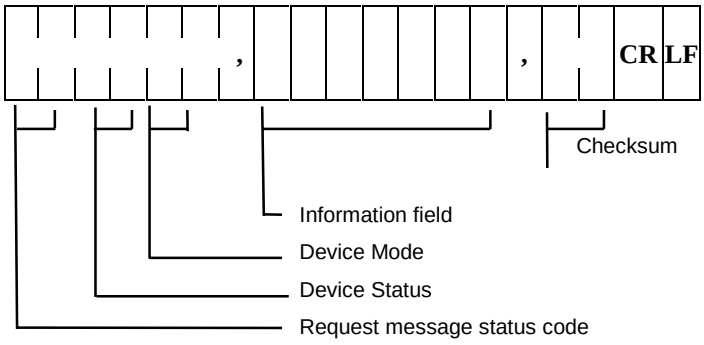


6.7.2 Loopback response

6.7.2.1 Loopback response message without checksum protocol



6.7.2.2 Loopback response message with checksum protocol



6.8 Checksum protocol (for data security)

The optional checksum protocol is used to increase security on the communication link. This protocol enables both your computer and your recorder to detect messages that have been interrupted by line noise.

6.8.1 Using checksum protocol

You can use the checksum protocol with any message exchange. The recorder uses the protocol to check the transmission of request messages. Your computer uses the protocol to check the transmission of response messages when a message exchange includes the checksum protocol.

- Your recorder can tell, with high probability, if the ASCII code in the request message has changed during transmission from your computer.
- Your computer can tell, with high probability, if the ASCII code in the response message has changed during transmission from the recorder.

To use the checksum protocol, change the format of the request message as shown in chapter *General request message with checksum – page 64.*

1. Use a 4204 in request protocol field.
2. Insert 2 hexadecimal characters which represent the checksum you have calculated from the ASCII codes in the request message as explained in the following chapter.

6.8.2 Procedure to calculate the checksum

1. Take the binary sum, ignoring carry forwards generated by the most significant bits, of the ASCII codes for each of the message's characters, ignoring parity, up to but not including the checksum field and the CR and LF characters. The final sum should not be an 8-bit binary number.
2. Convert the four least significant bits of this sum to the equivalent hexadecimal digit. This becomes the least significant digit in the checksum field.
3. Convert the four most significant bits of this sum to the equivalent hexadecimal digit. This becomes the most significant digit in the checksum field.

0	5	,	4	2	0	4	,	0	1	0	B	,	0	,	0	2	,	0	8	,	0	4	CR	LF

0	30
5	35
,	2C
4	34
2	32
0	30
4	34
,	2C
0	30
1	31
0	30
B	42
,	2C
0	30
,	2C
0	30
2	32
,	2C
0	30
8	38
,	2C +
4	04

-> Checksum = 04.

7. ASCII FUNCTIONS

7.1 Alarm status

This frame tells whether the addressed alarms are ON or OFF or informs if the addressed channels are on alarm or not.

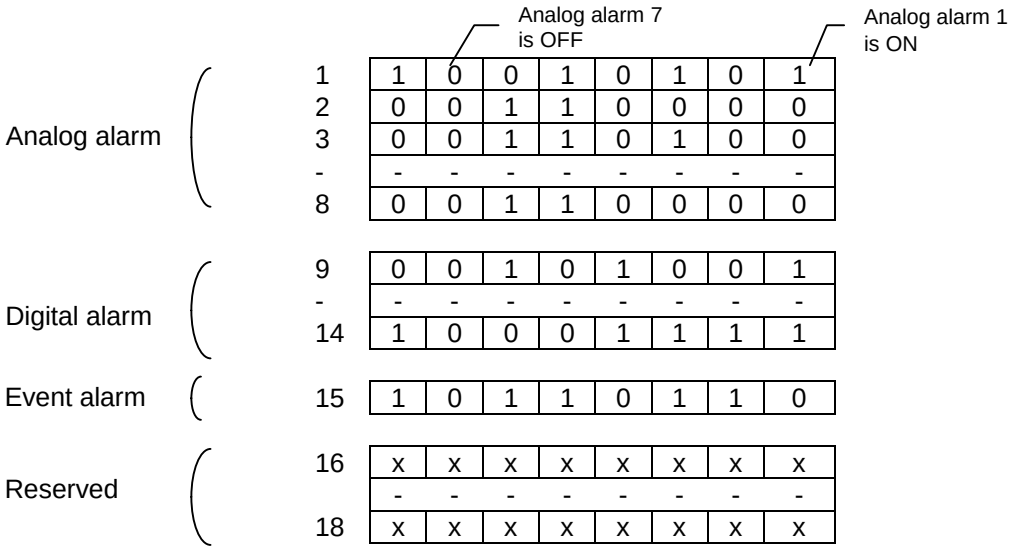
It can indicate the alarm status (ON or OFF) of:
- 64 analog alarms
- 48 digital alarms
- 8 events

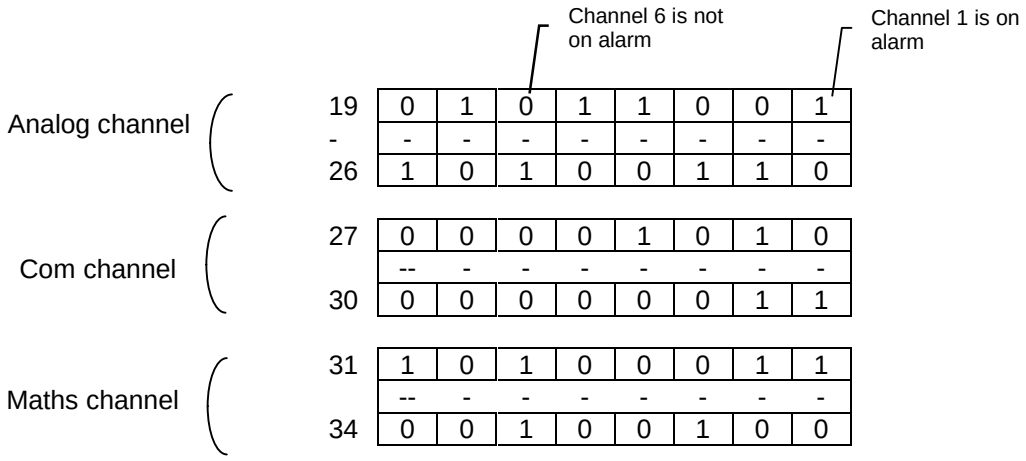
The possible channels to be addressed are:
- analog inputs 1 to 64
- com PVs 1 to 32
- maths 1 to 32

- ☑ Parameter code : 01h
- ☑ Function code : 01 read only
- ☑ Maximum number of values : 34 (22h)
- ☑ Data field length : Number of value bytes
- ☑ Data field
  - bit 0 = alarm off
  - bit 1 = alarm on

☑ Starting index position

1 to 8	Analog alarm
9 to 14	Digital alarm
15	Event alarm
16 to 18	Reserved
19 to 26	Channel analog
27 to 30	Channel Com
31 to 34	Channel Math





➔ **WARNING**: When using the communication card with a 180 mm recorder, be aware of the number of analog alarms not being 64 but 24. Consequently, bytes 1 to 3 represent analog alarm 1 to 24 and byte 4 to 8 have to be regarded as reserved locations. The same consideration is effective with the com and maths alarms number.

EXAMPLE:

0	**4**	**,**	**0**	**2**	**0**	**4**	**,**	**0**	**1**	**0**	**1**	**,**	**0**	**,**	**0**	**3**	**,**	**0**	**1**	**,**	**CR**	**LF**									

☑ Request : what is the status of analog alarm 1 to 24?

- 04 : address
- 0204 : no checksum
- 01 : read request
- 01 : parameter code (Alarm status)
- 0 : data type
- 03 : number of bytes to read
- 01 : starting index

0	**0**	**0**	**0**	**0**	**1**	**,**	**A**	**D**	**,**	**0**	**2**	**,**	**0**	**0**	**,**	**CR**	**LF**														

☑ Response:

- 00 : operation performed
- 00 : recorder is working correctly
- 01 : mode is run mode
- AD 02 00 : alarms 1, 3, 4, 6, 8 and 10 are ON

7.2 Print a message

Allows you to print a message of maximum 64 (replace 64 by 50 for a 180 mm recorder) characters on the paper. The authorized characters are given in appendix A.

➔ **WARNING**: The maximal length of the message settable by the user is fixed to 64 characters.
The message will be printed under the following format:

C: User text
Example: "C: This is a user-message"

The "C:" message is included in the message length. That is to say that the user-message is actually limited to 62 (48) characters. A longer message will be truncated to the authorized limit.

The character @ is used to place the date or time of the recorder in the user-message:

- By default, the message will be printed on blank.
- @d will be replaced by the current date.
- @h will be replaced by the current time.
- @e the message will be printed on trace.

When using @d or @h, keep in mind that the date and time message will take respectively 9 characters and 5 characters.

☑ Parameter code : 03h
☑ Function code : 02 write only
☑ Number of values : 01
☑ Data field length : 64 (40h) bytes max
☑ Data field : hexadecimal representation of the ASCII value of the character corresponding to the table in appendix A. (See also example)

EXAMPLE:

0	2	,	0	2	0	4	,	0	2	0	3	,	0	,	0	1	,	0	1	,	4	0	,	6	4	,	2	0	,

	4	0	,	6	8	,	2	0	,	3	5	,	3	5	,	3	5	,	2	0	,	4	0	,	6	5	,	CR	LF	

☑ Request : print “current date and time 555” on trace

- 02 : address
- 0204 : no checksum
- 02 : write request
- 03 : parameter code (Print a message)
- 0 : data type field
- 01 : number field (no meaning for this function)
- 01 : starting index (no meaning for this function)
- 40 64 20 ...65 : @d @h 555 @e ("31 AUG 97 14:50 555")

0	**0**	**0**	**0**	**0**	**1**	**,**	**CR**	**LF**

☑ Response:

- 00 : operation performed
- 00 : recorder is working correctly
- 01 : mode is run mode

7.3 Printer status

Indicates the status of the printer.

- ☑ Parameter code : 08h
- ☑ Function code : 01 Read only
- ☑ Number of values : 01
- ☑ Data field length : 08 bytes

Byte numbers:

Byte:	0	Reserved	00
	1	Cassette state	Value: 00 = cassette out 01 = cassette in
	2	Paper speed	Value: 00 = speed 1 used 01 = speed 2 used
	3	Printer mode	Value: 00 = printer inhibit 01 = print mode
	4-7	Remaining paper	Value: 4 bytes in mm Value of remaining paper length IEEE format

EXAMPLE:

1	2	,	0	2	0	4	,	0	1	0	8	,	0	,	0	1	,	0	1	,	CR	LF

☑ Request : what is the printer status ?

- 12 : address
- 0204 : no checksum
- 01 : read request
- 08 : parameter code
- 0 : data type field
- 01 : number of value
- 01 : starting index

0	0	0	0	0	1	,	0	0	,	0	1	,	0	1	,	0	0	,	4	4	,	4	4	,	4	4	,

4	4	,	CR	LF

☑ Response:

- 000001 : operation performed - Recorder is working correctly – Mode is run mode
- 00 : reserved
- 01 : cassette in
- 01 : speed 2 in use
- 00 : printer inhibit
- 44 44 44 44 : value of remaining paper length: 44444444h = 785.066

7.4 Print process values (snapshot Log)

Print-out of the current PV's on the paper.

☑ Parameter code	: 0Ah
☑ Function code	: 02 write only
☑ Number of values	: 01
☑ Data field length	: 01 byte
☑ Data field:	value: 00 = print Trace 01 = print Analog 02 = print Digital 03 = print Math

EXAMPLE:

0	7	,	0	2	0	4	,	0	2	0	A	,	0	,	0	1	,	0	1	,	0	1	,	CR	LF

☑ Request : print analog PV's

- 07 : address
- 0204 : no checksum
- 02 : write request
- 0A : parameter code (Print process values)
- 0 : data type field
- 01 : number of value
- 01 : starting index
- 01 : print Trace

0	0	0	0	0	1	,	CR	LF

☑ Response:

- 00 : operation performed
- 00 : recorder is working correctly
- 01 : mode is run mode

7.5 Read process values

Reading of analog, com., maths process values.

☑ Parameter code : 18h
☑ Function code : 01 read only
☑ Maximum number of values : 128 (80h)
☑ Data field length : number of values x (4 bytes) (Hexadecimal representation of floating) (IEEE)
☑ Starting index position

N° of byte	180 mm	250 mm
1 - 24	Analog inputs (1 to 24)	Analog inputs (1 to 24)
25 - 64	Reserved	Analog inputs (25 to 64)
65 - 88	Com (1 to 24)	Com (1 to 24)
89 - 96	Reserved	Com (25 to 32)
97 - 120	Math (1 to 24)	Math (1 to 24)
121 - 128	Reserved	Math (25 to 32)

EXAMPLE:

0	1	,	0	2	0	4	,	0	1	1	8	,	0	,	0	2	,	0	2	,	CR	LF

☑ Request : what are the values of analog input 2 to 3 ?

- 01 : address
- 0204 : no checksum
- 01 : read request
- 18 : parameter code (Process values)
- 0 : data type field
- 02 : number of process values to read
- 02 : starting index (Start reading at analog input 2)

0	0	0	0	0	1	,	4	4	,	A	8	,	4	9	,	4	5	,	4	4	,	5	5	,

	6	6	,	7	7	,	CR	LF

☑ Response:

- 00 : operation performed
- 00 : recorder is working correctly
- 01 : mode is run mode
- 44 A8 49 45 : analog 2 value is 1346.29
- 44 55 66 77 : analog 3 value is 853.60

➔ **WARNING**: When an analog input is set with no-entry or a maths with no function then the value should be ignored.

7.6 Read digital process values

Reading of digital inputs (48 with a 250 mm recorder, 36 with a 180 mm):

☑ Parameter code : 1Ah
☑ Function code : 01 read only
☑ Number of values max : 9 bytes
☑ Data field length : number of value bytes
☑ Data field : 0 or 1
☑ Starting index position and byte meaning

	Digital 8							Digital 1	
1	0	0	0	1	0	1	1	1	digital
	-	-	-	-	-	-	-	-	
	-	-	-	-	-	-	-	-	
6	0	0	1	0	0	0	1	1	
	x	x	x	x	x	x	x	x	reserved
	-	-	-	-	-	-	-	-	
9	x	x	x	x	x	x	x	x	

➔ **WARNING**: Since the 180 mm recorder has only 36 possible digital inputs, the array above will contain only 5 rows.

EXAMPLE:

0	1	,	0	2	0	4	,	0	1	1	A	,	0	,	0	2	,	0	1	,	CR	LF

☑ Request : what are the Process values of digital 1 to 16 ?

- 01 : address
- 0204 : no checksum
- 01 : read request
- 1A : parameter code (Digital process value)
- 0 : data type field
- 02 : number of byte to read (byte 1 and 2)
- 01 : starting index (Start reading on byte 1)

0	0	0	0	0	1	,	1	7	,	3	0	,	CR	LF

☑ Response:

- 00 : operation performed
- 00 : recorder is working correctly
- 01 : mode is run mode
- 17 : reading result of byte 1-> digital 1, 2, 3, 5 are closed (equal to 1)
- 30 : reading result of byte 2-> digital 13, 14 are closed (equal to 1)

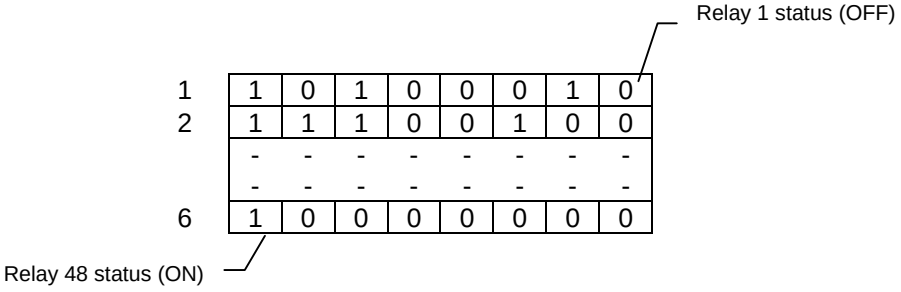
➔ **WARNING**: When a digital input is set with type NONE then the value should be ignored.

7.7 Relay status

Status of relays (1 to 48 for a 250 mm recorder, 1 to 36 for a 180 mm recorder) on relay-board when present.

☑ Parameter code : 0Ch
☑ Function code : 01 read only
☑ Maximum number of values : 06
☑ Data field length : number of values requested
☑ Data field value :
1 = relay active (ON)
0 = relay not active (OFF)

☑ Byte meaning



➔ **WARNING**: Since the 180 mm recorder can only have 36 possible relays, the above array will have five rows and the four most significant bits of the fifth byte will have no meaning.

EXAMPLE:

0	1	,	0	2	0	4	,	0	1	0	C	,	0	,	0	1	,	0	2	,	CR	LF

☑ Request : status of relays 9 to 16

- 01 : address
- 0204 : no checksum
- 01 : read request
- 0C : parameter code (Relay status)
- 0 : data type field
- 01 : number of byte to read
- 02 : starting index (Start reading on byte 2)

0	0	0	0	0	1	,	E	4	,	CR	LF

☑ Response:

- 00 : operation performed
- 00 : recorder is working correctly
- 01 : mode is run mode
- E4 : reading result of byte 2 ->Relay 11, 14, 15, 16 are ON

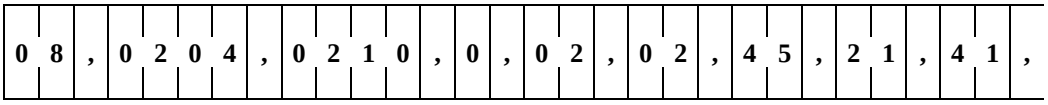
➔ **WARNING:** Response returns status of possible relays even when the relay boards are not fitted.

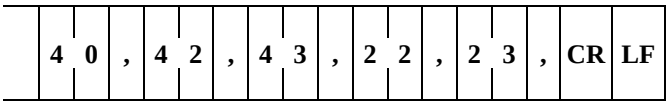
7.8 Send communication process values

Permits to send communication process values (32 for a 250 mm recorder, 24 for a 180 mm) to the recorder.

☑ Parameter code : 10h
☑ Function code : 02 write only
☑ Maximum number of values : 32 (20h)
☑ Data field length : (number of values x 4) bytes
☑ Data field info : hexadecimal representation of floating (IEEE)

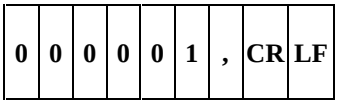
EXAMPLE:





☑ Request : override com 2 and 3

- 08 : address
- 0204 : no checksum
- 02 : write request
- 10 : parameter code (Com process values)
- 0 : data type field
- 02 : number of Com pvs to write
- 02 : starting index (start writing on Com pvs 2)
- 45 21 41 40 (2580.1) : new Com pvs 1 value (IEEE floating point)
- 42 43 22 23 (48.783) : new Com pvs 2 value (IEEE floating point)



☑ Response:

- 00 : operation performed
- 00 : recorder is working correctly
- 01 : mode is run mode

7.9 Configuration lock / unlock

Permits to lock the access (keyboard and jack) to modify parameters (See section 8). **A lock is requested before any modification of the configuration parameters.** When configuration has been changed, a unlock frame can be sent to allow configuration modifications again via the MMI.

☑ Parameter code : 2Eh
☑ Function code : 02, write only
☑ Number of values : 01
☑ Data field length : 01 byte
☑ Starting index : value:
00 = unlock configuration access
01 = lock configuration access
☑ Data field : 01

Possible responses:

- 09: operation performed
- 0A: operation already performed
- 0B: access refused

EXAMPLES:

0	**9**	**,**	**0**	**2**	**0**	**4**	**,**	**0**	**2**	**2**	**E**	**,**	**0**	**,**	**0**	**1**	**,**	**0**	**1**	**,**	**0**	**1**	**,**	**CR**	**LF**

☑ Request : LOCK requested

- 09 : address
- 0204 : no checksum
- 02 : write request
- 2E : parameter code (Configuration lock / unlock)
- 0 : data type field
- 01 : number of value
- 01 : starting index (Lock requested)
- 01 : data field

0	**0**	**0**	**0**	**0**	**1**	**,**	**0**	**9**	**,**	**CR**	**LF**

☑ Response:

- 00 : operation performed
- 00 : recorder is working correctly
- 01 : mode is run mode
- 09 : the lock has been set

➔ **WARNING**: THE LOCK WILL NOT BE ACTIVE WHEN YOU REBOOT YOUR RECORDER.

EXAMPLE:

0	9	,	0	2	0	4	,	0	2	2	E	,	0	,	0	1	,	0	0	,	0	1	,	CR	LF

☑ Request : UNLOCK requested

☑ 09 : address
☑ 0204 : no checksum
☑ 02 : write request
☑ 2E : parameter code (Configuration lock / unlock)
☑ 0 : data type field
☑ 01 : number of byte
☑ 00 : starting index (Unlock requested)
☑ 01 : data field

0	0	0	0	0	1	,	0	9	,	CR	LF

☑ Response:

- 00 : operation performed
- 00 : recorder is working correctly
- 01 : mode is run mode
- 09 : the lock has been disabled

7.10 End conf write

This frame allows you to indicate to the recorder that all the configuration changes are made. This action has to be executed after the unit configuration frame and this frame is fixed. Different parameters will induce an invalid error code.

☑ Parameter code : 06h
☑ Function code : 02 write only
☑ Number of values : 01
☑ Data field length : 01
☑ Data field : 01

EXAMPLE:

0	1	,	0	2	0	4	,	0	2	0	6	,	0	,	0	1	,	0	1	,	0	1	,	CR	LF

☑ Request : end signal of all configuration changes

- 01 : address
- 0204 : no checksum
- 02 : write request
- 06 : parameter code (End conf write)
- 0 : data type field
- 01 : number of byte
- 01 : starting index
- 01 : data field

0	0	0	0	0	1	,	CR	LF

☑ Response:

- 00 : operation performed
- 00 : recorder is working correctly
- 01 : mode is run mode

7.11 Read software version

It allows you to read in the same frame the software version of both recorder and communication board.

☑ Parameter code : 0E
☑ Function code : 05 (service)
☑ Number of values : 01
☑ Data field length : 18 (12h) bytes
☑ Data field : each Software version is given in a 9 bytes string

EXAMPLE:

0	4	,	0	2	0	4	,	0	5	0	E	,	0	,	0	1	,	0	1	,	CR	LF

- Request : what is the Software version of the recorder and the communication board?
- 04 : address
- 0204 : no checksum
- 05 : service
- 0E : parameter code (read software version)
- 0 : data type field
- 01 : number of byte
- 01 : starting index

0	0	0	0	0	1	,	2	0	,	2	0	,	2	0	,	3	0	,	3	0	,	3	1	,	4	1	,	4	5	

	,	0	0	,	2	0	,	2	0	,	2	0	,	3	1	,	3	0	,	3	0	,	4	1	,	4	1	,	0	0

	,	CR	LF

☑ Response:

- 00 : operation performed
- 00 : recorder is working correctly
- 01 : mode is run mode
- 20 20 20 30 30 31 41 45 00 : “001AE\0” recorder software version
- 20 20 20 31 30 30 41 41 00 : “100AA\0” communication card software version

7.12 Read card presence

This frame responses what kind of cards (no card, analog, alarm, current, logic, mV - mA) are connected in each slot of the recorder back panel. The bytes number in the response frame determines the slot number in the recorder (see your recorder manual for information about slot numbers).
In case of use with a 180 mm recorder, the four last bytes have to be ignored.

☑ Parameter code : 0F
☑ Function code : 05 (service)
☑ Number of values : 01
☑ Data field length : 16 (10h) bytes
☑ Data field : 00h No card
01h Analog card
02h Alarm card
03h Current card
04h Logic
05h mV - mA
22h Bad position

EXAMPLE:

0	2	,	0	2	0	4	,	0	5	0	F	,	0	,	0	1	,	0	1	,	CR	LF

☑ Request : what kind of cards are present on the 180 mm recorder?

- 02 : address
- 0204 : no checksum selection
- 05 : service
- 0F : parameter code (read card presence)
- 0 : data type field
- 01 : number of byte
- 01 : starting index

0	0	0	0	0	1	,	0	1	,	0	1	,	0	1	,	0	1	,	0	1	,	0	1	,	0	4	,	0	4	

	,	0	2	,	0	2	,	0	2	,	0	2	,	0	0	,	0	0	,	0	0	,	0	0	,	CR	LF

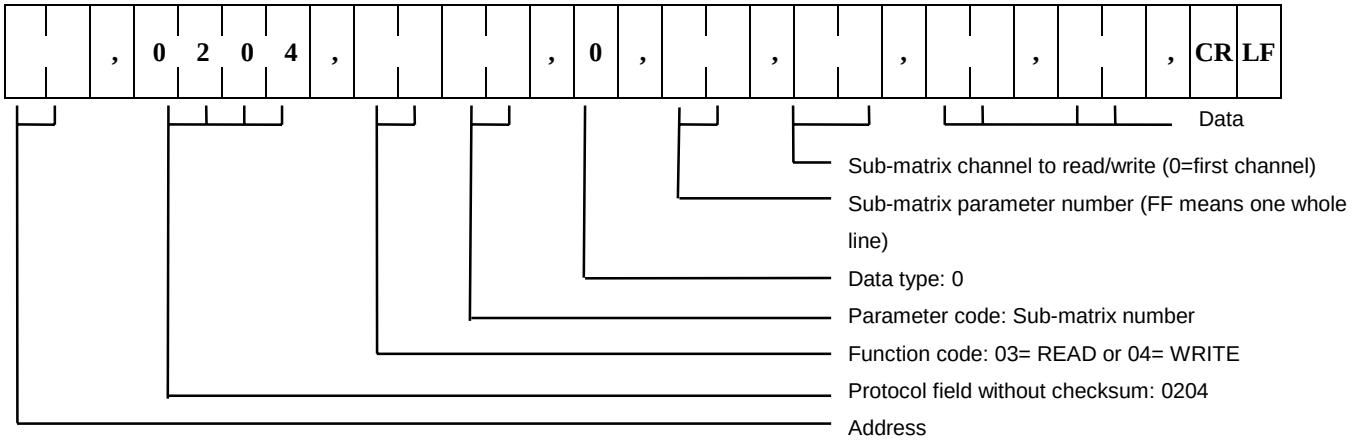
☑ Response:

- 00 : operation performed
- 00 : recorder is working correctly
- 01 : mode is run mode
- 01 01 01 01 01 01 : 6 analog cards in lower slots in slots 1 to 6
- 04 04 : 2 logic cards in slots 7, 8
- 02 02 02 02 : 4 alarm cards in slots 9 to 12
- 00 00 00 00 : no slots 13, 14, 15, 16 on a 180 mm recorder

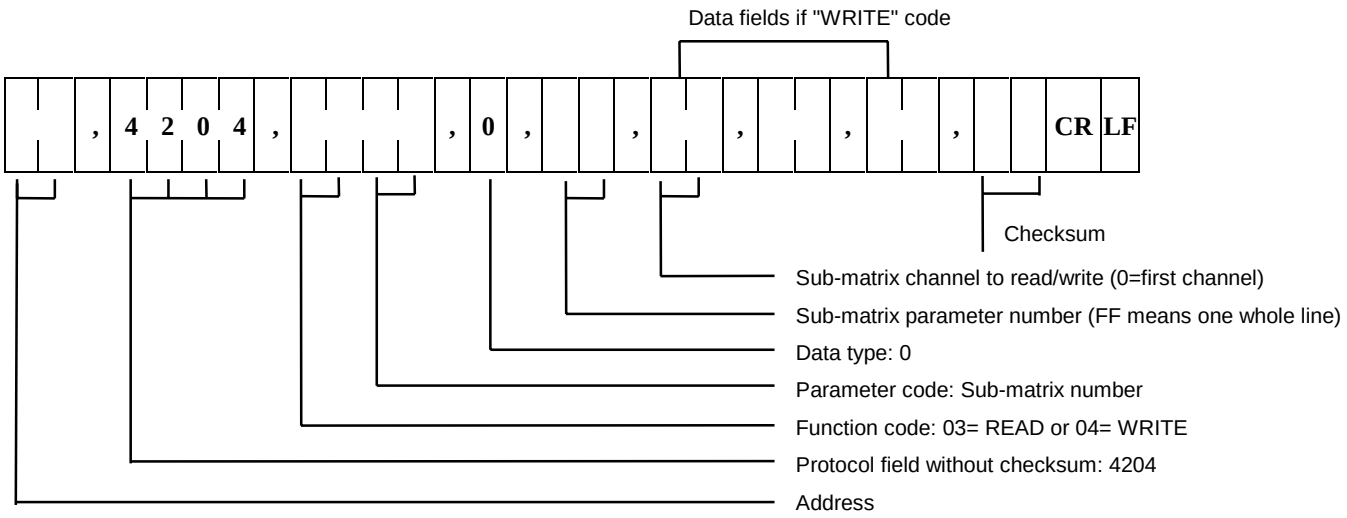
8. ASCII CONFIGURATION EXCHANGES

8.1 Overview

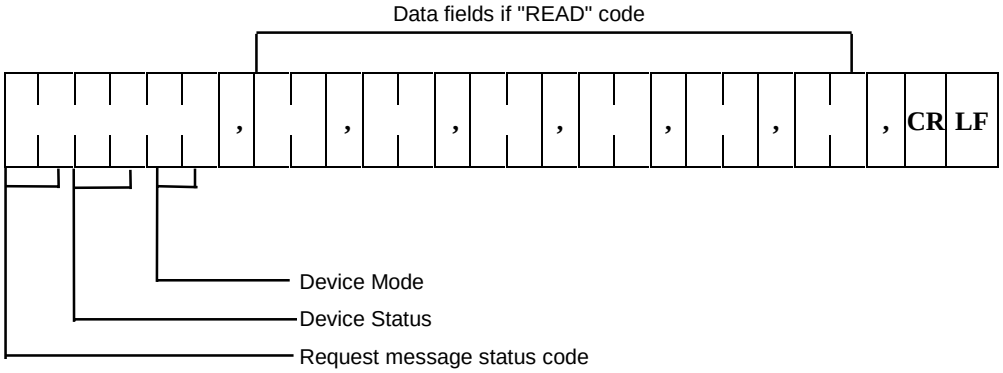
The ASCII communication allows you to READ or WRITE configuration data.
Each message allows you to exchange the complete configuration of a specified channel sub-matrix.
For each channel, you can read or write either a single parameter or the whole line (all parameters).



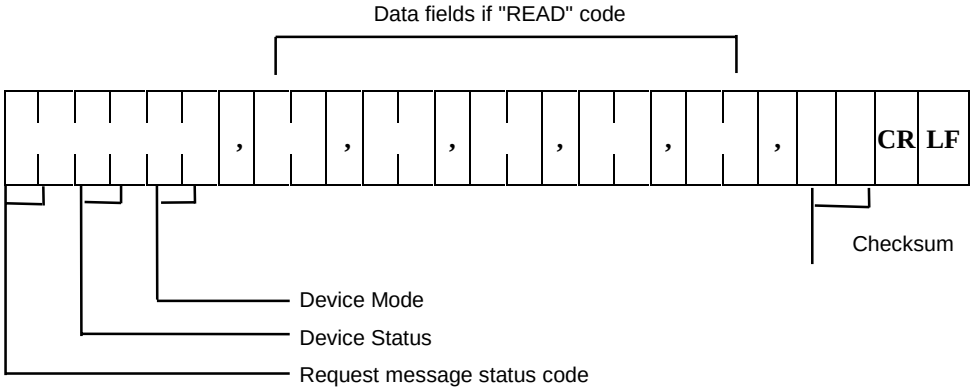
General request message without checksum



General request message with checksum



General response message without checksum



General response message with checksum

The following pages give you the code of each sub-matrix, the position of each parameter of a channel, the description of each parameter and how to interpret it.

8.2 Description

8.2.1 Instructions to follow in case of writes in configuration

Any change in configuration must follow these different instructions:

1. Lock Access to configuration
(see CONFIGURATION LOCK/UNLOCK, sub-section 7.9)

2. Writes in configuration (you can make several writes in different sub-matrices)
(see this section: ASCII CONFIGURATION EXCHANGES)

3. End of writes in configuration
(see END CONF WRITE, sub-section 7.10)

➔ **WARNING** : **Any change in configuration stops the printing and the measuring.**
In case of WRITE configuration, the transmission can take several seconds.

8.2.2 Example of writes: Changes in the MMI sub-matrix

Description:
Recorder Address: 01
Communication without checksum
MMI configuration wanted:

HOLD KEY: ENABLE (01)
DISPLAY KEY: ENABLE (01)
PRINT KEY: DISABLE (00)
RESET KEY: DISABLE (00)
ACK KEY: DISABLE (00)
F1 KEY: INHIBIT/PRINT (01)
F2 KEY: UNUSED (00)
DISPLAY HI: SPEED IN USE (07)
DISPLAY LO: DATE & TIME (08)
BRIGHT: >> 80% (04)
RESERVED: (00)
RESERVED: (00)

Communication:

1. Lock Access to configuration

Request: 01,0204,022E,0,01,01,01,CRLF
Response: 000001,09,CRLF

2. Write in MMI configuration

Request: 01,0204,0409,0,FF,00,01,01,00,00,00,01,00,07,08,04,00,00,CRLF
Response: 000001,CRLF (first write) ; 000003,CRLF (Response to the following writes)

3. End of write in configuration

Request: 01,0204,0206,0,01,01,01,CRLF
Response: 000003,CRLF

4. Unlock Access to configuration (if desired)

Request: 01,0204,022E,0,01,00,01,CRLF
Response: 000001,09,CRLF

8.3 Configuration description

8.3.1 Sub-matrices list

Number: NAME	Number of Channel 180 mm recorder	Number of Channel 250 mm recorder	
00: ALARM	48	64	Page 90
01: ANALOG INPUT	24	64	Page 95
02: DIGITAL	36	48	Page 101
05: MESSAGES	48	64	Page 106
06: CHART	24	32	Page 107
07: PRINTER	1	1	Page 113
08: CHART DOC	1	1	Page 116
09: MMI	1	1	Page 119
0Ah: EVENTS	6	6	Page 124
0Bh: MISCELLANEOUS	1	1	Page 126
0Ch: MATH	24	32	Page 129
0Eh: PERIODIC REPORT	1	1	Page 135
OFh: CURRENT 4-20 mA	8	8	Page 137

8.3.2 Parameters list of each sub-matrix

SUB-MATRIX → **00: ALARM**

Position of parameters		Number of data fields	
→	**00: SP VALUE**	**04**	page 91
→	**01: HYSTERESIS**	**04**	page 91
→	**02: RESERVED**	**04**	
→	**03: RESERVED**	**04**	
→	**04: OCCURRENCE**	**02**	page 91
→	**05: RESERVED**	**02**	
→	**06: RESERVED**	**02**	
→	**07: APPLY ON**	**01**	page 92
→	**08: ALARM TYPE**	**01**	page 92
→	**09: DIFF WITH**	**01**	page 92
→	**0Ah: ACTION**	**01**	page 93
→	**0Bh: MSG COLOR**	**01**	page 93
→	**0Ch: MSG TYPE**	**01**	page 93
→	**0Dh: MSG NUMBER**	**01**	page 94
→	**0Eh: RED IN AL**	**01**	page 94
→	**0Fh: RELAY NUM**	**01**	page 94
→	**10h: ACKNOWLEDGE**	**01**	page 94
→	**11h: RESERVED**	**01**	
→	**12h: RESERVED**	**01**	

SUB-MATRIX	*PARAMETER*
ALARM	**SP VALUE**
Definition:	The alarm switches from OFF to ON when the SP value is reached.
Parameter type:	Float (4 bytes, MSB First).
Possible values:	**[-9999999.0 ... 9999999.0]**

SUB-MATRIX	*PARAMETER*
ALARM	**HYSTERESIS**
Definition:	Establishes the alarm hysteresis. Alarms switch ON at set point but switch OFF value depends on the hysteresis setting. Hysteresis is expressed in Engineering units and is added to low alarm and subtracted from high alarm set points to establish the alarm release value.
Parameter type:	Float (4 bytes, MSB First).
Possible values:	**[0.0 ... 999.0]**

SUB-MATRIX	*PARAMETER*
ALARM	**OCCURRENCE**
Definition:	Defines the number of alarm occurrences that must occur after power on before alarm activation can actually operate. This acts as a filter for the alarm activation.
Parameter type/How to modify it:	Short Integer (2 bytes, MSB First). Select the value.
Possible values choices:	**[0...9]** **0** = No alarm occurrence (normal alarm activation) **1** = 1 alarm occurrence **2** = 2 alarm occurrences . . . **9** = 9 alarm occurrences

SUB-MATRIX	*PARAMETER*
ALARM	**APPLY ON**
Definition:	Channel on which the alarm is applied. (Analog 1 ... 24, Math 1 ... 24).
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**01: ANALOG #01 ; ANALOG # i (i = 1 ...*)** **41h: COMM #01 ; COMM # i (i = 1 ...*)** **61h: MATH #01 ; MATH # i (i = 1 ...*)**

* See Appendix E for the differences between 180 mm and 250 mm recorders

SUB-MATRIX	*PARAMETER*
ALARM	**ALARM TYPE**
Definition:	Type of alarm.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: NONE** **01: ALARM HIGH** **02: ALARM LOW** **03: CHG RATE H** **04: CHG RATE L** **05: CHG RATE H, L** **06: DIFFERENTIAL**

SUB-MATRIX	*PARAMETER*
ALARM	**DIFF WITH**
Definition:	Second channel used if alarm type is differential.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**The same as those for alarm channel.** **01: ANALOG #01; ANALOG # i (i = 1 ...*)** **41h: COMM #01; COMM # i (i = 1 ...*)** **61h: MATH #01; MATH # i (i = 1 ...*)**

* See Appendix E for the differences between 180 mm and 250 mm recorders

SUB-MATRIX	*PARAMETER*
ALARM	**ACTION**
Definition:	Action on printer in case of alarm.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: NO ACTION** **01: CHG SPD/INT** **02: CHG RANGE** **03: PRINT ON AL** **04: PRT INHIBIT** **05: TAB SQTRACE** **06: TAB SQBLANK** **08: PRT MATH LOG** **09: CHG GROUP B** **0Ah: CHG GROUP AB**

SUB-MATRIX	*PARAMETER*
ALARM	**MSG COLOR**
Definition:	Color of alarm message.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: BLACK** **01: BLUE** **02: PURPLE** **03: GREEN** **04: BROWN** **05: RED**

SUB-MATRIX	*PARAMETER*
ALARM	**MSG TYPE**
Definition:	Defines when the alarm message is printed.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: NONE** **01: STD MESSAGE** **02: MESSAGE ON** **03: MESSAGE OFF** **04: MSG ON/OFF**

SUB-MATRIX	*PARAMETER*
ALARM	**MSG NUMBER**
Definition:	Selection of the alarm message to be printed.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**01: MESSAGE #01; MESSAGE # i (i = 1 ...*)**

* See Appendix E for the differences between 180 mm and 250 mm recorders

SUB-MATRIX	*PARAMETER*
ALARM	**RED IN AL**
Definition:	Specifies if the trend trace will be printed in red during alarm condition.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: NO**
	01: YES

SUB-MATRIX	*PARAMETER*
ALARM	**RELAY NUM**
Definition:	Selection of the relay to activate in alarm condition.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: NO RELAY**
	01: RELAY #01
	RELAY #i (i = 1 ...*)

* See Appendix E for the differences between 180 and 250 mm recorders

SUB-MATRIX	*PARAMETER*
ALARM	**ACKNOWLEDGE**
Definition:	To acknowledge all alarm relay outputs.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**01: ENABLE**
	00: DISABLE

SUB-MATRIX → **01: ANALOG INPUT**

Position of parameters		Number of data fields	
→	**00: EXT COMP**	**4**	page 96
→	**01: FILTER**	**4**	page 96
→	**02: LOW VALUE**	**4**	page 96
→	**03: HIGH VALUE**	**4**	page 97
→	**04: LOW ADJUST**	**4**	page 97
→	**05: HIGH ADJUST**	**4**	page 97
→	**06: RESERVED**	**4**	
→	**07: RESERVED**	**2**	
→	**08: SENSOR**	**1**	page 97
→	**09: RANGE**	**1**	page 98
→	**0Ah: BURNOUT**	**1**	page 100
→	**0Bh: STD MATH**	**1**	page 100
→	**0Ch: DIFF WITH**	**1**	page 100
→	**0Dh: RESERVED**	**1**	
→	**0Eh: RESERVED**	**1**	
→	**0Fh: RESERVED**	**1**	

➔ **WARNING**: You have to understand well the User's Manual about ANALOG matrix: there are several correlations between parameters. For example, do not configure a Thermocouple SENSOR with a mV RANGE.

SUB-MATRIX	*PARAMETER*
ANALOG INPUT	**EXT COMP**
Definition:	The thermocouple sensor is directly connected to a remote temperature compensation box. Then the connections are made with copper leadwires. Two types of wiring are possible: 1. At a fixed temperature compensation box with temperature configurable from 0 to 80°C (32 to 176°F). 2. On variable temperature compensation box. We use 1 channel to measure the temperature of the box.
Parameter type:	Float (4 bytes, MSB First).
Possible values:	1. **[0.0 ...80.0]** : Fixed Temperature in °C. 2. **[ANALOG # i] :** variable temperature; select the channel used to measure the temperature of the box. **101.0 : ANALOG #1** **102.0 : ANALOG #2; ANALOG # i (i = 1 ... *)**

* See Appendix E for the differences between 180 mm and 250 mm recorders

SUB-MATRIX	*PARAMETER*
ANALOG INPUT	**FILTER**
Definition:	You may wish to apply a filter to noisy signals. However if pulses, square waves or other rapidly changing inputs are to be displayed and recorded without damping, choose 0 filter value.
Parameter type:	Float (4 bytes, MSB First).
Possible values:	**[0.0 ... 99.0] seconds** **0.0 = No filter** **10.0 = 10 seconds**

SUB-MATRIX	*PARAMETER*
ANALOG INPUT	**LOW VALUE**
Definition:	Engineering value corresponding to low limit of the selected input actuation range.
Parameter type:	Float (4 bytes, MSB First).
Possible values:	**[-9999.0 ... 9999.0]**

SUB-MATRIX	*PARAMETER*
ANALOG INPUT	**HIGH VALUE**
Definition:	Engineering value corresponding to high limit of the selected input actuation range.
Parameter type:	Float (4 bytes, MSB First).
Possible values:	**[-9999.0 ... 9999.0]**

SUB-MATRIX	*PARAMETER*
ANALOG INPUT	**LOW ADJUST** **HIGH ADJUST**
Definition:	Zero adjust and span adjust are values used to calibrate a temperature loop. Otherwise choose zero value = factory calibration. Adjustments are made directly in engineering unit. (ex: 5 = 5°C)
Parameter type:	Float (4 bytes, MSB First).
Possible choices:	**[-99.0 ... 99.0]**

SUB-MATRIX	*PARAMETER*
ANALOG INPUT	**SENSOR**
Definition:	Basic sensor type used on each channel.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: T/C INT COMP** **01: T/C EXT COMP** **02: RTD** **03: TR NL 0-5V** **04: TR NL 1-5V** **05: TR NL 0-20mA** **06: TR NL 4-20mA** **07: LINEAR** **08: SPECIAL** **09: NO ENTRY**

SUB-MATRIX	*PARAMETER*
ANALOG INPUT	**RANGE**
Definition:	DISPLAY ACTUATION RANGE For directly connected temperature sensors and non-linear temperature transmitters, the actuation selection defines the linearization routine used to produce a linear chart scale. For linear transmitters, the selection simply defines the transmitter's electrical range/span. The choice of actuation offered by the recorder during configuration will depend upon the selected sensor. The allowed ranges will depend on whether you have selected a thermocouple, Linear or Non Linear or RTD.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	Depend on the type of sensor connected. Possible selections are listed below.

AVAILABLE RANGES

LINEAR	RTD/OHMS	
RANGE	RANGE	RANGE
mV: 58h: 0, 10 mV 59h: -10, 0, 10 mV 5Ah: 0, 20 mV 5Bh: -20, 0, 20 mV 5Ch: 0, 50 mV 5Dh: -50, 0, 50 mV 5Eh: 10, 50 mV 5Fh: 0, 100 mV 60h: -100, 0, 100 mV 61h: 0, 500 mV 62h: -500, 0, 500 mV Volt: 65h: 0, 1 V 66h: -1, 0, 1 V 67h: 0, 2 V 68h: -2, 0, 2 V 69h: 0, 5 V 6Ah: -5, 0, 5 V 6Bh: 1, 5 V 6Ch: 0, 10 V 6Dh: -10, 0, 10 V mA: 63h: 0, 20 mA* 64h: 4, 20 mA*	Pt 100 Ω at 0°C: 3Ch: -50, 0, 150 °C 3Dh: -58, 0, 302 °F 3Eh: 0, 100 °C** 3Fh: 32, 212 °F** 40h: 0, 200 °C 41h: 32, 392 °F 42h: 0, 400 °C 43h: 32, 752 °F 44h: -200, 0, 800 °C 45h: -328, 0, 1472 °F Ni 50 ohms: 50h: -80, 0, 320 °C 51h: -112, 0, 608 °F Ni 508 ohms: 52h: -80, 0, 150 °C 53h: -112, 0, 302 °F Cu 10 ohms: 54h: -20, 0, 250 °C*** 55h: -4, 0, 482 °F*** Ohms: 56h: 0, 200 Ω 57h: 0, 2000 Ω	JIS: 46h: -50, 0, 150 °C 47h: -58, 0, 302 °F 48h: 0, 100 °C** 49h: 32, 212 °F** 4Ah: 0, 200 °C 4Bh: 32, 392 °F 4Ch: 0, 400 °C 4Dh: 32, 752 °F 4Eh: -200, 0, 500 °C 4Fh: -328, 0, 932 °F

* The mA inputs have to be connected on a 250 Ω input resistor across the input terminals.
** Accuracy: 0.25 %
*** Accuracy: 0.5 %

AVAILABLE RANGES (continued)

THERMOCOUPLES						
	RANGE		RANGE		RANGE	
	J: 00: -50, 0, 150 °C 01: -58, 0, 302 °F 02: 0, 400 °C 03: 32, 752 °F 04: -200, 0, 870 °C 05: -328, 0, 1598 °F L: 06: -50, 0, 150 °C 07: -58, 0, 302 °F 08: 0, 400 °C 09: 32/752 °F 0Ah: -200/870 °C 0Bh: -328/1598 °F K: 0Ch: 0, 400 °C 0Dh: 32, 752 °F 0Eh: 0, 800 °C 0Fh: 32, 1472 °F 10h: 0, 1200 °C 11h: 32, 2192 °F 12h: -200, 0, 1370 °C 13h: -328, 0, 2498 °F R : 1Ch: -20, 0, 1760 °C 1Dh: -4, 0, 3200 °F		S: 1Eh: 0, 1600 °C 1Fh: 32, 2912 °F 20h: -20, 0, 1760 °C 21h: -4, 0, 3200 °F N: 14h: 0, 400 °C 15h: 32, 752 °F 16h: 0, 800 °C 17h: 32, 1472 °F 18h: 0, 1200 °C 19h: 32, 2192 °F 1Ah: -200, 0, 1300 °C 1Bh: -328, 0, 2372 °F T: 22h: -50, 0, 150 °C 23h: -58, 0, 302 °F 24h: 0, 150 °C 25h: 32, 302 °F 26h: 50, 150 °C 27h: 122, 302 °F 28h: -200, 0, 400 °C 29h: -328, 0, 752°F USER ACTUATION: 70h: User actuation 1 71h: User actuation 2		U: 2Ah: -50, 0, 150 °C 2Bh: -58, 0, 302 °F 2Ch: 0, 150 °C 2Dh: 32, 302 °F 2Eh: 50, 150 °C 2Fh: 122, 302 °F 30h: -200, 0, 400 °C 31h: -328, 0, 752 °F NiMo: 32h: 0, 1400 °C 33h: 32, 2552 °F Moco: 6Eh: 0, 1400 °C 6Fh: 32, 2552 °F W-W26: 34h: -20, 0, 2320 °C 35h: -4, 0, 4208 °F W5-W26: 36h: -20, 0, 2320 °C 37h: -4, 0, 4208 °F PR20-40: 38h: 0, 1800 °C 39h: 32, 3272 °F B: 3Ah: 400, 1820 °C 3Bh: 752, 3308 °F	

SUB-MATRIX	*PARAMETER*
ANALOG INPUT	**BURNOUT**
Definition:	Allows you to define the safety backup position to activate ANALOG ALARM PARAMETERS (if configured) in case of sensor burnout. The trace can go either to the right (high) or to the left (low).
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: NO BURNOUT** **01: B.OUT LOW** **02: B.OUT HIGH** **<u>Not configurable selections</u>**: **FIX LOW** **FIX HIGH** **FIX NONE**

SUB-MATRIX	*PARAMETER*
ANALOG INPUT	**STD MATH**
Definition:	2 mathematical functions are included as standard in the recorder. These functions apply only to analog inputs.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: NO OPT MATH** **01: SQUARE ROOT** **02: CHANNEL DIFF**

SUB-MATRIX	*PARAMETER*
ANALOG INPUT	**DIFF WITH**
Definition:	Second channel used when **STD MATH = CHANNEL DIFF.**
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**01: ANALOG #01; ANALOG # i (i = 1 ... *)** **00: NONE**

***** See Appendix E for the differences between 180 mm and 250 mm recorders

SUB-MATRIX → **02: DIGITAL**

Position of parameters		Number of data fields	
→	**00: RESERVED**	**2**	
→	**01: OFF POSITN**	**2**	page 102
→	**02: ON POSITN**	**2**	page 102
→	**03: RESERVED**	**2**	
→	**04: RESERVED**	**2**	
→	**05: TYPE**	**1**	page 102
→	**06: DIFF WITH**	**1**	page 102
→	**07: ACTION**	**1**	page 103
→	**08: RELAY NUM**	**1**	page 103
→	**09: MSG NUM**	**1**	page 103
→	**0Ah: MSG COLOR**	**1**	page 104
→	**0Bh: MSG TYPE**	**1**	page 104
→	**0Ch: RED IN AL**	**1**	page 104
→	**0Dh: ACKNOWLEDGE**	**1**	page 105
→	**0Eh: TRACE COLOR**	**1**	page 105
→	**0Fh: TRACE**	**1**	page 105
→	**10h: RESERVED**	**1**	
→	**11h: RESERVED**	**1**	
→	**12h: RESERVED**	**1**	

SUB-MATRIX	*PARAMETER*
DIGITAL	**OFF POSITN**
Definition:	Defines the trace position (DI = OFF) on the chart. (In %) One increment = 1%
Parameter type:	Short Integer (2 bytes, MSB First).
Possible choices:	**[0 ... 100]**

SUB-MATRIX	*PARAMETER*
DIGITAL	**ON POSITN**
Definition:	Defines the trace position (DI = ON) on the chart. (In %) One increment = 1%
Parameter type:	Short Integer (2 bytes, MSB First).
Possible choices:	**[0 ... 100]**

SUB-MATRIX	*PARAMETER*
DIGITAL	**TYPE**
Definition:	Type of digital input.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: NONE** **01: DIG CLOSED** **02: DIG OPENED** **03: DIFFERENTIAL**

SUB-MATRIX	*PARAMETER*
DIGITAL	**DIFF WITH**
Definition:	Second digital input to be used if the first digital input type is differential.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**01: DIGITAL #01; DIGITAL # i (i = 1 ... *)**

* See Appendix E for the differences between 180 mm and 250 mm recorders

SUB-MATRIX	*PARAMETER*
DIGITAL	**ACTION**
Definition:	Action on printer in case of digital input change.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: NO ACTION**
	01: CHG SPD/INT
	02: CHG RANGE
	03: PRINT ON AL
	04: PRT INHIBIT
	05: TAB SQTRACE
	06: TAB SQBLANK
	08: PRT MATH LOG
	09: CHG GROUP B
	0Ah: CHG GROUP AB

SUB-MATRIX	*PARAMETER*
DIGITAL	**RELAY NUM**
Definition:	Selection of the relay activated with digital input.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: NO RELAY**
	01: RELAY #01 **RELAY #i (i = 1 ... *)**

* See Appendix E for the differences between 180 mm and 250 mm recorders

SUB-MATRIX	*PARAMETER*
DIGITAL	**MSG NUM**
Definition:	Selection of the message to be printed.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**01: MESSAGE #01** **MESSAGE # i (i = 1... *)**

* See Appendix E for the differences between 180 mm and 250 mm recorders

SUB-MATRIX	*PARAMETER*
DIGITAL	**MSG COLOR**
Definition:	Color of the message.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: BLACK**
	01: BLUE
	02: PURPLE
	03: GREEN
	04: BROWN
	05: RED

SUB-MATRIX	*PARAMETER*
DIGITAL	**MSG TYPE**
Definition:	Defines when the digital message is printed.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: NONE**
	01: STD MESSAGE
	02: MESSAGE ON
	03: MESSAGE OFF
	04: MSG ON/OFF

SUB-MATRIX	*PARAMETER*
DIGITAL	**RED IN AL**
Definition:	Specifies if the digital input trace will be printed in red with digital action.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: NO**
	01: YES

SUB-MATRIX	*PARAMETER*
DIGITAL	**ACKNOWLEDGE**
Definition:	To acknowledge all alarm latching relay outputs.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**01: ENABLE**
	00: DISABLE

SUB-MATRIX	*PARAMETER*
DIGITAL	**TRACE COLOR**
Definition:	Defines the color of the digital trace.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: BLACK**
	01: BLUE
	02: PURPLE
	03: GREEN
	04: BROWN
	05: RED

SUB-MATRIX	*PARAMETER*
DIGITAL	**TRACE**
Definition:	Enable/disable the digital trace.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**01: ENABLE**
	00: DISABLE

SUB-MATRIX → **05: MESSAGES**

Position of parameters		Number of data fields
→	**00: MESSAGE** page 106	**72**

SUB-MATRIX *PARAMETER*

MESSAGES	**MESSAGE**
Definition:	To configure the messages (1 to 48)
Parameter type:	String Char (MSB First).
Possible choices:	**50 characters + 22 «\0» for 180 mm recorders** **(Data field for «\0» is 00)** **or** **64 characters + 8 «\0» for 250 mm recorders**

SUB-MATRIX → **06: CHART**

Position of parameters		Number of data fields	
→	**00: MIN RANGE 1**	**4**	page 108
→	**01: MAX RANGE 1**	**4**	page 108
→	**02: MIN RANGE 2**	**4**	page 108
→	**03: MAX RANGE 2**	**4**	page 108
→	**04: RESERVED**	**4**	
→	**05: RESERVED**	**4**	
→	**06: 0% ZONE**	**2**	page 109
→	**07: 100% ZONE**	**2**	page 109
→	**08: SUB DIV**	**2**	page 109
→	**09: RESERVED**	**2**	
→	**0Ah: TAG NAME**	**9**	page 110
→	**0Bh: ENG UNIT**	**6**	page 110
→	**0Ch: TRACE**	**1**	page 110
→	**0Dh: RG 1 COLOR**	**1**	page 111
→	**0Eh: RG 2 COLOR**	**1**	page 111
→	**0Fh: RANGE USED**	**1**	page 112
→	**10h: GROUP DEF**	**1**	page 112
→	**11h: FORMAT**	**1**	page 112
→	**12h: RESERVED**	**1**	
→	**13h: RESERVED**	**1**	
→	**14h: RESERVED**	**1**	

SUB-MATRIX	*PARAMETER*
CHART	**MIN RANGE 1**
Definition:	Lower limit of chart range 1.
Parameter type:	Float (4 bytes, MSB First).
Possible choices:	**[- 9999999.0 ... 9999999.0]**

SUB-MATRIX	*PARAMETER*
CHART	**MAX RANGE 1**
Definition:	Upper limit of chart range 1.
Parameter type:	Float (4 bytes, MSB First).
Possible choices:	**[- 9999999.0 ... 9999999.0]**

SUB-MATRIX	*PARAMETER*
CHART	**MIN RANGE 2**
Definition:	Lower limit of chart range 2.
Parameter type:	Float (4 bytes, MSB First).
Possible choices:	**[- 9999999.0 ... 9999999.0]**

SUB-MATRIX	*PARAMETER*
CHART	**MAX RANGE 2**
Definition:	Upper limit of chart range 2.
Parameter type:	Float (4 bytes, MSB First).
Possible choices:	**[- 9999999.0 ... 9999999.0]**

SUB-MATRIX	*PARAMETER*
CHART	**0% ZONE**
Definition:	Defines chart zone for printing.
Parameter type:	Short Integer (2 bytes, MSB First).
Possible choices:	**[0 ... 80]** **0 ... 100% ----> 0 ... 80% for 0% zone of paper ----> 20 ... 100% for 100% zone** **The choice of parameters for 0% and 100% zone allows you to define the datum such that the width of the chart paper is less than the calibrated width of the paper.** **This function permits the segregation of input traces into zones to avoid the problem of input signals using the same paper scale and having the same values printed on top of each other.** **The minimum width per zone is 20% of the chart.**

SUB-MATRIX	*PARAMETER*
CHART	**100 % ZONE**
Definition:	Defines chart zone for printing.
Parameter type:	Short Integer (2 bytes, MSB First).
Possible choices:	**[20 ... 100]** **0 ... 100% ----> 0 ... 80% for 0% zone of paper ----> 20 ... 100% for 100% zone** **The choice of parameters for 0% and 100% zone allows you to define the datum such that the width of the chart paper is less than the calibrated width of the paper.** **This function permits the segregation of input traces into zones to avoid the problem of input signals using the same paper scale and having the same values printed on top of each other.** **The minimum width per zone is 20% of the chart.**

SUB-MATRIX	*PARAMETER*
CHART	**SUB DIV**
Definition:	This parameter allows the visualization of a scale on the paper in addition to the classic **RANGE** message.
Parameter type:	Short Integer (2 bytes, MSB First).
Possible choices:	**[0 ... 9]** **00: NO DIVISION**

SUB-MATRIX	*PARAMETER*
CHART	**TAG NAME**
Definition:	Name of the chart channel.
Parameter type:	String char.
Possible choices:	**8 characters + «\0» (Data field for «\0» is 00)**

SUB-MATRIX	*PARAMETER*
CHART	**ENG UNIT**
Definition:	Chart channel units.
Parameter type:	String char.
Possible choices:	**5 characters + «\0» (Data field for «\0» is 00)**

SUB-MATRIX	*PARAMETER*
CHART	**TRACE**
Definition:	Defines the variable to be printed on the chart. (None, Analog inputs, Comm inputs, Maths results)
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: NO TRACE** **01: ANALOG #01; ANALOG # i (i = 1 ...*)** **41h: COMM #01; COMM # i (i = 1 ...*)** **61h: MATH #01; MATH # i (i = 1 ...*)**

***** See Appendix E for the differences between 180 mm and 250 mm recorders

SUB-MATRIX	*PARAMETER*
CHART	**RG 1 COLOR**
Definition:	Color of range 1.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: BLACK**
	01: BLUE
	02: PURPLE
	03: GREEN
	04: BROWN
	05: RED
	06: BLACK THICK
	07: BLUE THICK
	08: PURPLE THICK
	09: GREEN THICK
	0Ah: BROWN THICK
	0Bh: RED THICK

SUB-MATRIX	*PARAMETER*
CHART	**RG2 COLOR**
Definition:	Color of range 2.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: BLACK**
	01: BLUE
	02: PURPLE
	03: GREEN
	04: BROWN
	05: RED
	06: BLACK THICK
	07: BLUE THICK
	08: PURPLE THICK
	09: GREEN THICK
	0Ah: BROWN THICK
	0Bh: RED THICK

SUB-MATRIX	*PARAMETER*
CHART	**RANGE USED**
Definition:	You may select whether the input channel will be printed normally (range 1 or 2) or on alarm (with range 1 or 2).
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: WITH RG1**
	01: WITH RG2
	02: RG1 ON ALARM
	03: RG2 ON ALARM

SUB-MATRIX	*PARAMETER*
CHART	**GROUP DEF**
Definition:	This parameter defines two separate groups of channels that will be printed together as a group.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: NO GROUP**
	01: GROUP A
	02: GROUP B

SUB-MATRIX	*PARAMETER*
CHART	**FORMAT**
Definition:	Format used for the printing of trend and range and information, and the display of trace value.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: AUTOMATIC**
	01: XXXXX (no decimal point) 10000
	02: XXX.X (1/10) 100.0
	03: XX.XX (1/100) 10.00
	04: X.XXX(1/1000) 1.000

SUB-MATRIX → **07: PRINTER**

Position of parameters		Number of data fields	
→	**00: CHART LG**	**4**	page 114
→	**01: SPEED 1**	**4**	page 114
→	**02: SPEED 2**	**4**	page 114
→	**03: RESERVED**	**4**	
→	**04: RESERVED**	**2**	
→	**05: SPEED UNIT**	**1**	page 114
→	**06: SP/INT USED**	**1**	page 114
→	**07: INTERVAL 1**	**2**	page 115
→	**08: INTERVAL 2**	**2**	page 115
→	**09: RECORD MODE**	**1**	page 115
→	**0Ah: PRINT MODE**	**1**	page 115
→	**0Bh: RESERVED**	**1**	
→	**0Ch: RESERVED**	**1**	

SUB-MATRIX	*PARAMETER*
PRINTER	**CHART LG**
Definition:	Specifies the chart length of the chart roll or fanfold which actuates the event alarm. This is used with the recorder **EVENTS** to signal when the chart paper has reached the preconfigured chart length.
Parameter type:	Float (4 bytes, MSB First).
Possible choices:	**[0.0 ... 35000.0] mm (always in mm)**

SUB-MATRIX	*PARAMETER*
PRINTER	**SPEED 1**
Definition:	Value of speed 1.
Parameter type:	Float (4 bytes, MSB First).
Possible choices:	**[0.0 ... 5000.0] mm/h (always in mm/h)**

SUB-MATRIX	*PARAMETER*
PRINTER	**SPEED 2**
Definition:	Value of speed 2.
Parameter type:	Float (4 bytes, MSB First).
Possible choices:	**[0.0 ... 5000.0] mm/h (always in mm/h)**

SUB-MATRIX	*PARAMETER*
PRINTER	**SPEED UNIT**
Definition:	Speed unit.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: UNIT = mm/h** **01: UNIT = inch/h**

SUB-MATRIX	*PARAMETER*
PRINTER	**SP/INT USED**
Definition:	Defines speed in use in alarm off.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: SPEED 1** **01: SPEED 2**

SUB-MATRIX	*PARAMETER*
PRINTER	**INTERVAL 1**
Definition:	Tab 1 prints interval. (In minutes)
Parameter type:	Short Integer (2 bytes, MSB First).
Possible choices:	**[1 ... 1440] minutes**

SUB-MATRIX	*PARAMETER*
PRINTER	**INTERVAL 2**
Definition:	Tab 2 prints interval.
Parameter type:	Short Integer (2 bytes, MSB First).
Possible choices:	**[1 ... 1440] minutes**

SUB-MATRIX	*PARAMETER*
PRINTER	**RECORD MODE**
Definition:	Recording mode allows you to print normally or to stop the printer.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: INHIBIT**
	01: PRINT

SUB-MATRIX	*PARAMETER*
PRINTER	**PRINT MODE**
Definition:	Under this heading you must choose whether recording will be in **TREND** mode or **TABULAR** mode.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: TREND**
	01: TABULAR

SUB-MATRIX → **08: CHART DOC**

Position of parameters		Number of data fields	
→	**00: RESERVED**	**4**	
→	**01 [0]: INFORMATION 01**	**2**	page 117
→	**01 [1]: INFORMATION 02**	**2**	page 117
→	**01 [2]: INFORMATION 03**	**2**	page 117
→	**01 [3]: INFORMATION 04**	**2**	page 117
→	**01 [4]: INFORMATION 05**	**2**	page 117
→	**01 [5]: INFORMATION 06**	**2**	page 117
→	**01 [6]: INFORMATION 07**	**2**	page 117
→	**01 [7]: INFORMATION 08**	**2**	page 117
→	**01 [8]: INFORMATION 09**	**2**	page 117
→	**01 [9]: INFORMATION 10**	**2**	page 117
→	**02: PRINT INTRVAL**	**2**	page 117
→	**03: TRACE REF**	**1**	page 117
→	**04: FUNCT MSG**	**1**	page 118
→	**05: RESERVED**	**1**	
→	**06: RESERVED**	**1**	

SUB-MATRIX	*PARAMETER*
CHART DOC	**INFORMATION 01...10**
Definition:	Next information is printed at a distance which depends on **PRT INTRVAL** value and print speed.
Parameter type:	Array of Short Integer (10 * 2 bytes, MSB First).
Possible choices:	**00: NO INFORMATION**
	01: MESSAGE #01; MESSAGE # i (i = 1 ... *)
	41h: RANGE #01; RANGE # i (Traces, i = 1...*) and chart certification
	61h: NEXT RANGE
	62h: BLANK
	63h: SNAP SHOT TRACE
	64h: SNAP SHOT ANALOG
	65h: SNAP SHOT MATH
	66h: SNAP SHOT LOGIC

***** See Appendix E for the differences between 180 mm and 250 mm recorders

SUB-MATRIX	*PARAMETER*
CHART DOC	**PRT INTRVAL**
Definition:	Separation between two consecutive information print-outs # 1 to 10 printed on the chart.
Parameter type:	Short Integer (2 bytes, MSB First).
Possible choices:	**[1 ... 1440] minutes**

SUB-MATRIX	*PARAMETER*
CHART DOC	**TRACE REF**
Definition:	Specifies whether a number or number plus tag name is printed alongside the trace to identify the channel.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: NUMBER**
	01: NUMBER & TAG

SUB-MATRIX	*PARAMETER*
CHART DOC	**FUNCT MSG**
Definition:	Change range, change speed, change group and print inhibit messages are the functional messages of the recorder. The **FUNCT MSG** parameter allows the customer to select whether to print these messages on the paper.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**01: ENABLE** **00: DISABLE**

SUB-MATRIX → **09: MMI**

Position of parameters		Number of data fields	
→	**00: HOLD KEY**	**1**	page 120
→	**01: DISPLAY KEY**	**1**	page 120
→	**02: PRINT KEY**	**1**	page 120
→	**03: RESET KEY**	**1**	page 120
→	**04: ACK KEY**	**1**	page 121
→	**05: F1 KEY**	**1**	page 121
→	**06: F2 KEY**	**1**	page 121
→	**07: DISPLAY HI**	**1**	page 122
→	**08: DISPLAY LO**	**1**	page 122
→	**09: BRIGHT**	**1**	page 123
→	**0Ah: RESERVED**	**1**	
→	**0Bh: RESERVED**	**1**	

SUB-MATRIX	*PARAMETER*
MMI	**HOLD KEY**
Definition:	To enable the Hold key so the operator can modify the display scanning from the keyboard and Hold the upper display on a desired channel.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**01: ENABLE**
	00: DISABLE

SUB-MATRIX	*PARAMETER*
MMI	**DISPLAY KEY**
Definition:	To enable the Display key so the operator can modify the display from the keyboard.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**01: ENABLE**
	00: DISABLE

SUB-MATRIX	*PARAMETER*
MMI	**PRINT KEY**
Definition:	To enable the Printer key so the operator can modify the printer action from the keyboard.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**01: ENABLE**
	00: DISABLE

SUB-MATRIX	*PARAMETER*
MMI	**RESET KEY**
Definition:	To reset the maths functions or alarm occurrence.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**01: ENABLE**
	00: DISABLE

SUB-MATRIX	*PARAMETER*
MMI	**ACK KEY**
Definition:	To release ALL energized alarms that have **ACKNOWLEDGE** enabled in **ALARM** and **DIGITAL** sub-matrices.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**01: ENABLE**
	00: DISABLE

SUB-MATRIX	*PARAMETER*
MMI	**F1 KEY** **F2 KEY**
Definition:	Defines the action of the F1 and F2 keys in operation mode. The user can define the action of these keys.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: UNUSED**
	01: INHIBIT/PRINT
	02: RESERVED
	03: RESET PAPER LENG
	04: CHANGE SPEED
	05: PRINT DATE&TIME
	06: SNAP SHOT TRACE
	07: CHART ADVANCE
	08: CHG GROUP A
	09: CHG GROUP B
	0Ah: CHG GROUP A+B
	0Bh: SNAP SHOT LOGIC
	0Ch: SNAP SHOT MATH

SUB-MATRIX	*PARAMETER*
MMI	**DISPLAY HI**
Definition:	Type of information displayed on the upper line of the display in the run mode at power on.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: ANALOG INPUTS**
	01: 2 PVS TRACE
	02: MATH RESULTS
	03: COMM RESULTS
	04: ALARM STATUS
	05: RESERVED
	06: RESERVED
	07: SPEED IN USE
	08: DATE & TIME
	09: TRACE & TAG
	0Ah: RESERVED
	0Bh: TRACE IN ALARM
	0Ch: LOGIC STATES

SUB-MATRIX	*PARAMETER*
MMI	**DISPLAY LO**
Definition:	Type of information displayed on the lower line of the display in the run mode at power on.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: ANALOG INPUTS**
	01: 2 PVS TRACE
	02: NOT ALLOWED
	03: COMM RESULTS
	04: ALARM STATUS
	05: RESERVED
	06: RESERVED
	07: SPEED IN USE
	08: DATE & TIME

SUB-MATRIX	*PARAMETER*
MMI	**BRIGHT**
Definition:	To modify the display brightness during operation.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: OFF (0%)** **01: >> (20%)** **02: MEDIUM (40%)** **03: >> (60%)** **04: >> (80%)** **05: HIGH (100%)**

SUB-MATRIX → **0Ah: EVENTS**

Position of parameters		Number of data fields	
→	**00: RELAY NUM**	**1**	page 125
→	**01: DISPLAY**	**1**	page 125
→	**02: RESERVED**	**1**	
→	**03: RESERVED**	**1**	

EV01 **NO PAPER**
EV02 **END PAPER**
EV03 **BATTERY FAIL**
EV04 **ONE ALARM ON**
EV05 **BURNOUT**
EV06 **SHEDTIME**

SUB-MATRIX	*PARAMETER*
EVENTS	**RELAY NUM**
Definition:	Selection of relay activated for each event condition.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: NO RELAY**
	01: RELAY #01; RELAY # i (i = 1 ...*)

* See appendix E for the differences between 180 mm and 250 mm recorders

SUB-MATRIX	*PARAMETER*
EVENTS	**DISPLAY**
Definition:	Enable/disable a display indication of the event occurrence.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**01: ENABLE**
	00: DISABLE

SUB-MATRIX → **0Bh: MISCELLANEOUS**

Position of parameters		Number of data fields	
→	**00: RESERVED**	**2**	
→	**01: IDENTIF #**	**2**	page 127
→	**02: TIME**	**3**	page 127
→	**03: DATE**	**3**	page 127
→	**04: PASSWORD 1**	**9**	page 127
→	**05: PASSWORD 2**	**9**	page 128
→	**06: OPTIONS**	**13**	page 128
→	**07: LANGUAGE**	**1**	page 128
→	**08: FREQUENCY**	**1**	page 128
→	**09: RESERVED**	**1**	
→	**0Ah: RESERVED**	**1**	
→	**0Bh: RESERVED**	**1**	

SUB-MATRIX	*PARAMETER*
MISCEL	**IDENTIF #**
Definition:	Identification number of the instrument which will be printed on the chart.
Parameter type:	Short Integer (2 bytes, MSB First).
Possible values:	**[0 ... 99]**

SUB-MATRIX	*PARAMETER*
MISCEL	**TIME**
Definition:	To set the real time clock.
Parameter type:	Array of 3 Unsigned Char (MSB First).
Possible choices:	**00:00:00 up to 23:59:59**
	- The first character for hours: enter the numeric value
	- The second character for minutes: enter the numeric value
	- The third character for seconds: enter the numeric value

SUB-MATRIX	*PARAMETER*
MISCEL	**DATE**
Definition:	To set the real time clock date.
Parameter type:	Array of 3 Unsigned Char (MSB First).
Possible choices:	- The first character for day: enter the numeric value
	- The second character for month: enter the numeric value
	- The third character for year: enter the numeric value

SUB-MATRIX	*PARAMETER*
MISCEL	**PASSWORD 1**
Definition:	Used to provide a limited access to configuration parameters.
Parameter type:	String char.
Possible choices:	**8 characters + «\0» (Data field for «\0» is 00).**

SUB-MATRIX	*PARAMETER*
MISCEL	**PASSWORD 2**
Definition:	Used to provide full access to configuration.
Parameter type:	String char.
Possible choices:	**8 characters + «\0» (Data field for «\0» is 00).**

SUB-MATRIX	*PARAMETER*
MISCEL	**OPTIONS**
Definition:	Type of optional maths package.
Parameter type:	String char.
Possible choices:	**12 characters + «\0» (Data field for «\0» is 00)**

SUB-MATRIX	*PARAMETER*
MISCEL	**LANGUAGE**
Definition:	Operator information and configuration language.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: ENGLISH**
	01: FRENCH
	02: GERMAN
	03: SPANISH
	04: ITALIAN

SUB-MATRIX	*PARAMETER*
MISCEL	**FREQUENCY**
Definition:	To select the line frequency.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: 50 HZ**
	01: 60 HZ

SUB-MATRIX → **0Ch: MATH**

Position of parameters		Number of data fields	
→	**00: COEF A**	**4**	page 130
→	**01: COEF B**	**4**	page 130
→	**02: COEF C**	**4**	page 130
→	**03: COEF D**	**4**	page 130
→	**04: RESERVED**	**4**	
→	**05: RESERVED**	**4**	
→	**06: TAG NAME**	**9**	page 130
→	**07: ENG UNIT**	**6**	page 131
→	**08: FUNCTION**	**1**	page 131
→	**09: START**	**1**	page 132
→	**0Ah: RESET**	**1**	page 132
→	**0Bh: FORMAT**	**1**	page 133
→	**0Ch: VARIAB A**	**1**	page 133
→	**0Dh: VARIAB B**	**1**	page 134
→	**0Eh: VARIAB C**	**1**	page 134
→	**0Fh: VARIAB D**	**1**	page 134
→	**10h: RESERVED**	**1**	
→	**11h: RESERVED**	**1**	
→	**12h: RESERVED**	**1**	

SUB-MATRIX	*PARAMETER*
MATH	**COEF A**
Definition:	First coefficient of a function.
Parameter type:	Float (4 bytes, MSB First).
Possible values:	**[-9999999.0 ... 9999999.0**]

SUB-MATRIX	*PARAMETER*
MATH	**COEF B**
Definition:	Second coefficient of a function.
Parameter type:	Float (4 bytes, MSB First).
Possible values:	**[-9999999.0 ... 9999999.0]**

SUB-MATRIX	*PARAMETER*
MATH	**COEF C**
Definition:	Third coefficient of a function.
Parameter type:	Float (4 bytes, MSB First).
Possible values:	**[-9999999.0 ... 9999999.0]**

SUB-MATRIX	*PARAMETER*
MATH	**COEF D**
Definition:	Fourth coefficient of a function.
Parameter type:	Float (4 bytes, MSB First).
Possible values:	**[-9999999.0 ... 9999999.0]**

SUB-MATRIX	*PARAMETER*
MATH	**TAG NAME**
Definition:	Maths result name.
Parameter type:	String char.
Possible values:	**8 characters + «\0» (Data field for «\0» is 00)**.

SUB-MATRIX	*PARAMETER*
MATH	**ENG UNIT**
Definition:	Maths result unit.
Parameter type:	String char.
Possible values:	**5 characters + «\0» (Data field for «\0» is 00)**. Ex: • Deg F • Deg C • PSI • BAR • µA • mm/h

SUB-MATRIX	*PARAMETER*
MATH	**FUNCTION**
Definition:	Selection of a function.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible values:	**00: NO FUNCTION** **01: SPECIAL** **02: ADDITION** **03: BASIC MATH 1** **04: BASIC MATH 2** **05: SQUARE ROOT** **06: FO STERILIZATION** **07: GROUP SUM** **08: TOTALIZATION** **09: GAS MAS FLOW** **0Ah: LIQ MAS FLOW** **0Bh: STEAM FLOW** **0Ch: ENERGY CONSP** **0Dh: GRP AVERAGE** **0Eh: SIMPLE AVRG** **0Fh: ELAPSED TIME** **10h: LAP TIME** **11h: CUMUL TIME** **12h: COUNT DOWN** **13h: PERIOD PULSE** **14h: TRANSLOG** **15h: RUNNING AVRG** **16h: SIGMA A** **17h: REL HUMIDITY** **18h: GROUP MIN** **19h: GROUP MAX** **1Ah: GRP MAX MIN** **1Bh: % CARBON** **1Ch: ALARM COUNT** **1Dh: VACUUM 10**

SUB-MATRIX	*PARAMETER*
MATH	**START**
Definition:	Start condition of maths computation.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible values:	**00: CONTINUOUSLY** **01: DI #01 CLOSED** **DI CLOSED #i (i = 1 ...*)** **31h: AL #01 ON** **AL ON #i (i = 1 ...*)**
NOTE:	Stop conditions are the following: **DI OPENED#** **AL OFF #**

***** See Appendix E for the differences between 180 mm and 250 mm recorders

SUB-MATRIX	*PARAMETER*
MATH	**RESET**
Definition:	Reset condition of maths computation.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible values:	**00: NO RESET** **01: DI #01 OPENED** **DI OPENED #i (i = 1 ...*)** **31h: AL #01 OFF** **AL OFF #i (i = 1 ...*)**

SUB-MATRIX	*PARAMETER*
MATH	**FORMAT**
Definition:	Selection of a format.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible values:	**00: AUTOMATIC** **01: XXXXXXXX** **02: XXXXXX.X** **03: XXXXX.XX** **04: XXXX.XXX** **05: EXPONENT**

SUB-MATRIX	*PARAMETER*
MATH	**VARIAB A**
Definition:	First variable of a function.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible values:	**01: ANALOG #01** **ANALOG # i (i = 1 ...*)** **41h: COMM #01** **COMM # i (i = 1 ... *)** **61h: MATH #01** **MATH # i (i = 1 ... *)**

***** See Appendix E for the differences between 180 mm and 250 mm recorders

SUB-MATRIX	*PARAMETER*
MATH	**VARIAB B**
Definition:	Second variable of a function.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible values:	**01: ANALOG #01** **ANALOG # i (i = 1 ...*)** **41h: COMM #01** **COMM # i (i = 1 ...*)** **61h: MATH #01** **MATH # i (i = 1 ...*)**

SUB-MATRIX	*PARAMETER*
MATH	**VARIAB C**
Definition:	Third variable of a function.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible values:	**01: ANALOG #01** **ANALOG # i (i = 1 ...*)** **41h: COMM #01** **COMM # i (i = 1 ...*)** **61h: MATH #01** **MATH # i (i = 1 ...*)**

SUB-MATRIX	*PARAMETER*
MATH	**VARIAB D**
Definition:	Fourth variable of a function.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible values:	**01: ANALOG #01** **ANALOG # i (i = 1 ...*)** **41h: COMM #01** **COMM # i (i = 1 ...*)** **61h: MATH #01** **MATH # i (i = 1 ...*)**

***** See appendix E for the differences between 180 mm and 250 mm recorders

SUB-MATRIX → **0Eh: PERIODIC REPORT**

Position of parameters		Number of data fields	
→	**00: PERIOD**	**4**	page 136
→	**01: RESERVED**	**2**	
→	**02: RESERVED**	**2**	
→	**03: SELECTION 1, ... 20**	**32**	page 136
→	**04: RESERVED**	**64**	
→	**05: SYNCHRO AT**	**3**	page 136
→	**06: RESERVED**	**1**	
→	**07: RESERVED**	**1**	
→	**08: DESTINATION**	**1**	page 136
→	**09: RESERVED**	**1**	
→	**0Ah: RESERVED**	**1**	

SUB-MATRIX	*PARAMETER*
PERIODIC REPORT	**PERIOD**
Definition:	This parameter corresponds to the computation period of a paragraph relative to minimum, average and maximum values for each analog input and maths result.
Parameter type:	Float (4 bytes, MSB First).
Possible choices:	**0** means no Periodic Report **[0.25 ...720.0]** Report Hours in cents of hours

SUB-MATRIX	*PARAMETER*
PERIODIC REPORT	**SELECTION 01, ... 20**
Definition:	Defines every TRACE or MATH which will be computed in the periodic report.
Parameter type:	Array of 20 Char (MSB First).
Possible choices:	**00: NONE** **01: TRACE #01; TRACE # i (i= 1 ...*)** **21h: MATH #01; MATH # i (i= 1 ...*)**

***** See Appendix E for the differences between 180 mm and 250 mm recorders

SUB-MATRIX	*PARAMETER*
PERIODIC REPORT	**SYNCHRO AT**
Definition:	Synchronization time of first periodic report and following.
Parameter type:	Array of **3** Unsigned Char (MSB First).
Possible choices:	**00:00:00 up to 23:59:00** - The first character for hours: enter the numeric value - The second character for minutes: enter the numeric value - **The third character for seconds is NOT USED**

SUB-MATRIX	*PARAMETER*
PERIODIC REPORT	**DESTINATION**
Definition:	Destination where the report will be sent.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: ON PAPER**

SUB-MATRIX → **0Fh: CURRENT 4/20 mA**

Position of parameters		Number of data fields	
→	**00: 4mA VALUE**	**4**	page 138
→	**01: 20mA VALUE**	**4**	page 138
→	**02: RESERVED**	**4**	page 138
→	**03: APPLY ON**	**1**	page 138
→	**04: RESERVED**	**1**	page 138

SUB-MATRIX	*PARAMETER*
CURRENT 4/20mA	**4mA VALUE**
Definition:	Determines the value associated with 4mA.
Parameter type:	Float (4 bytes, MSB first)
Possible values:	**[-9999999.0 ... 9999999.0]**

***** See Appendix E for the differences between 180 mm and 250 mm recorders

SUB-MATRIX	*PARAMETER*
CURRENT 4/20mA	**20mA VALUE**
Definition:	Determines the value associated with 20 mA.
Parameter type:	Float (4 bytes, MSB first)
Possible values:	**[-9999999.0 ... 9999999.0]**

SUB-MATRIX	*PARAMETER*
CURRENT 4/20mA	**APPLY ON**
Definition:	Defines the channel from which the current output will be calculated.
Parameter type:	Unsigned Char (1 byte, MSB First).
Possible choices:	**00: NONE**
	01: ANALOG #01; ANALOG #i (i = 1 ... *)
	41h: COMM #01; COMM #i (i = 1 ... *)
	61h: MATH #01; MATH #i (i = 1 ... *)

* See Appendix E for the differences between 180 mm and 250 mm recorders

9. APPENDIX A

9.1 Ascii conversion table

Control				Figures				Uppercase				Lowercase			
ASCII	Recorder	HEX	DEC	ASCII	Recorder	HEX	DEC	ASCII	Recorder	HEX	DEC	ASCII	Recorder	HEX	DEC
NUL (CTL@)	◆	00	0	Space	Space	20	32	@	@	40	64	\	\	60	96
SOH (CTLA)	◆	01	1	!	!	21	33	A	A	41	65	a	a	61	97
STX (CTLB)	◆	02	2	"	"	22	34	B	B	42	66	b	b	62	98
ETX (CTLC)	◆	03	3	#	#	23	35	C	C	43	67	c	c	63	99
EOT (CTLD)	◆	04	4	$	$	24	36	D	D	44	68	d	d	64	100
ENQ (CTLE)	◆	05	5	%	%	25	37	E	E	45	69	e	e	65	101
ACK (CTLF)	◆	06	6	&	&	26	38	F	F	46	70	f	f	66	102
BEL (CTLG)	◆	07	7	’	’	27	39	G	G	47	71	g	g	67	103
BS (CTLH)	◆	08	8	(	(	28	40	H	H	48	72	h	h	68	104
HT (CTLI)	◆	09	9	)	)	29	41	I	I	49	73	i	i	69	105
LF (CTLJ)	◆	0A	10	*	*	2A	42	J	J	4A	74	j	j	6A	106
VT (CTLK)	◆	0B	11	+	+	2B	43	K	K	4B	75	k	k	6B	107
FF (CTLL)	◆	0C	12	,	,	2C	44	L	L	4C	76	l	l	6C	108
CR (CTLM)	◆	0D	13	-	-	2D	45	M	M	4D	77	m	m	6D	109
SO (CTLN)	◆	0E	14	.	.	2E	46	N	N	4E	78	n	n	6E	110
SI (CTLO)	◆	0F	15	/	/	2F	47	O	O	4F	79	o	o	6F	111
DLE (CTLP)	◆	10	16	0	0	30	48	P	P	50	80	p	p	70	112
DC1 (CTLQ)	◆	11	17	1	1	31	49	Q	Q	51	81	q	q	71	113
DC2 (CTLR)	◆	12	18	2	2	32	50	R	R	52	82	r	r	72	114
DC3 (CTLS)	◆	13	19	3	3	33	51	S	S	53	83	s	s	73	115
DC4 (CTLT)	◆	14	20	4	4	34	52	T	T	54	84	t	t	74	116
NAK (CTLU)	◆	15	21	5	5	35	53	U	U	55	85	u	u	75	117
SYN(CTLV)	◆	16	22	6	6	36	54	V	V	56	86	v	v	76	118
ETB (CTLW)	◆	17	23	7	7	37	55	W	W	57	87	w	w	77	119
CAN (CTLX)	◆	18	24	8	8	38	56	X	X	58	88	x	x	78	120
EM (CTLY)	◆	19	25	9	9	39	57	Y	Y	59	89	y	y	79	121
SUB (CTLZ)	◆	1A	26	:	:	3A	58	Z	Z	5A	90	z	z	7A	122
ESC (CTL[)	◆	1B	27	;	;	3B	59	[	◆	5B	91	{	{	7B	123
FS (CTL\)	◆	1C	28	<	<	3C	60	\	◆	5C	92	|	|	7C	124
GS (CTL])	◆	1D	29	=	=	3D	61	]	◆	5D	93	}	}	7D	125
RS (CTL^)	◆	1E	30	>	>	3E	62	^	◆	5E	94	-	-	7E	126
US (CTL_)	◆	1F	31	?	?	3F	63	_	_	5F	95	DEL	◆	7F	127

◆ : unauthorized character

10. APPENDIX B

10.1 Programming example

When using a personal computer, this program can be used to test the loopback protocol. (ASCII)
In line 50 the serial communication slot is configured as :
9600 baud even parity, 7 bits per character and 1 stop bit.

```
10      DIM TRANSMIT$ (256)
20      DIM RECEIVES$ (256)
30      DEFINT A-Z
40      CLS
50      OPEN "COM1 : 9600, E, 7, 1; CS, LF" AS #1 LEN=256
60      TRANSMIT$="01,0204,8A,D,ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890,.?@#$%^&*()_+,
70      GOSUB 200
80      GOTO 70
200     PRINT TRANMIT$
210     PRINTS #1, TRANSMIT$
220     LINE INPUT #1, RECEIVE$
230     PRINT RECEIVE$
240     RETURN
```

11. APPENDIX C

11.1 Troubleshooting

Communication problems can be divided into 2 parts:

1. The recorder is not responding at all. To solve this problem you have to check:

   - The polarity +/- for the RS422/RS485 link.
   - The transmission line of the host has to be connected to the reception line of the recorder and vice versa.
   - The protocol, connection, baud rate, parity, bits per character and stop bits of the host have to be the same as configured on the recorder. Verify the communication matrix.
   - In case of RS485 (2 wires) the RTS (Request To Send) signal of the host must be enabled during transmission.
   - Make sure that you have selected the proper jumper selection for RS232/RS422/RS485 on the universal communication board.
   - Verify that the address slave is correct, that CRC is correct for RTU.
   - Make sure that the format of your request is correct. Use for example the loopback protocol. (See sub-section 6.4)
   - Make sure your communication board works properly, use the Communication Services sub-matrix, software service (see sub-section 3.2).
     In case of "BOOT XXXX" version, try to download the firmware (a previous download has failed).

2. The recorder responds with an error code in the request message status code.

ASCII RESPONSES:

01 Invalid request:
- Request for a variable which does not exist.
- READING from a variable which is WRITE only.
- WRITING to a variable which is READ only.
- Selection of a starting index which is too large.
- Selection of a number which is too large.
- Using invalid data in the data field (check the possible values for the appropriate parameter).

02 Invalid format:
- The request does not contain hexadecimal data.
- No comma delimited.

04 Invalid checksum, parity or framing error.
- Make sure you have calculated the correct checksum when using 4204 in the protocol field.
- Check the quality of your communication link.

05 Invalid mode

- To write configuration parameters, the "LOCK" frame is required. (See CONFIGURATION LOCK parameter)
- Communication between recorder mother board and communication board is temporarily impossible, the slave is engaged in process of a long duration program command. Retransmit later.

06 Data is out of range

- This code is returned only during write in configuration. This means that the data in the data field is not within the acceptable range of the parameter to be written. Note that even if during a line write a value is out of range, the other valid data will be loaded in the different parameters.

RTU RESPONSES:

01 Illegal function:

- The function code field is incorrect.
- It can only equal to 03 / 04, 06, 08, 16, 20 or 21.

02 Illegal data address:

- The address referenced in the data field is not allowed for the function code.
- The address of floating value is odd instead of even.
- The number of registers is incorrect for the specific action. (Too large or too small)

03 Illegal data value:

- The data field value is not allowed for the function code.
- The number of data byte is too large or too small.

06 Busy, rejected message

- To write configuration parameters, the "LOCK" frame is required. (See CONFIGURATION LOCK parameter for RTU)
- Communication between recorder mother board and communication board is temporarily impossible, the slave is engaged in process of a long duration program command. Retransmit later.

In any case, look at the manual examples.

12. APPENDIX D

12.1 IEEE 32 bit floating point information

12.1.1 Introduction

The recorder support IEEE 32 bit floating point information for several of the function codes. Each IEEE 32 bit floating point number requires two consecutive registers (four bytes) starting with the register defined as the starting register for the information.

12.1.2 Bit order

The bit order for the IEEE floating point is shown below :

- Reg N HIGH BYTE SIGN, EXP7, EXP6, EXP5, EXP4, EXP3, EXP2, EXP1
- Reg N LOW BYTE EXP0, M22, M21, M20, M19, M18, M17, M16

- Reg N+1 HIGH BYTE M15, M14, M13, M12, M11, M10, M9, M8
- Reg N+1 LOW BYTE M7, M6, M5, M4, M3, M2, M1, M0

Where :

SIGN is the mantissa sign bit
EXP7 through EXP0 are exponent bits
M22 through M0 are mantissa bits

a) If Exponent = 255 and Mantissa $\neq$ 0 then
Value is Not a Number (NaN) regardless of SIGN.

b) If Exponent = 255 and Mantissa = 0 then
$V = (-1)^{sign} \infty$

c) If 0 <Exponent < 255 then
$V = (-1)^{sign} \ 2^{Exp-127} \ (1.Mantissa)$

d) If Exponent = 0 and Mantissa $\neq$ 0 then
$V = (-1)^{sign} \ 2^{-126} \ (0.Mantissa)$ (DENORMALIZED NUMBER)

e) If Exponent = 0 and Mantissa = 0 then
$V = (-1)^{sign} \ 0$ (zero)

12.1.3 Examples

The following examples of IEEE floating point are provided for reference.

Value	IEEE FP		REG N		REG N+1	
(decimal)	MSB	LSB	High	Low	High	Low
1.0	3F800000h		3fh	80h	00h	00h
2.0	40000000h		40h	00h	00h	00h
-1.0	BF800000h		BFh	80h	00h	00h
100.0	42C80000h		42h	C8h	00h	00h

12.1.4 Warning

In an IBM compatible PC, floating point values are normally stored in byte and word swapped order. In an IBM machine, the value 1.0 is stored in ascending address locations as 00h, 00h, 80h, 3fh.

13. APPENDIX E

13.1 Differences between 180 and 250 mm recorders

	180 mm recorder	250 mm recorder
Number of slots	12 (2*6)	16 (2*8)
Analog inputs	24 (6*4)	64 (16*4)
Digital inputs	36 (6*6)	48 (8*6)
Traces (Comm, Maths or Analog)	24	32
Analog alarms (SP)	48	64
Mathematic channels	24	32
Communication channels (input)	24	32
Relays	36 (6*6)	48 (8*6)
Messages Number of characters/message	48 msg 50 char	64 msg 64 char

14. PROMPTS TRANSLATION

EN	FR	GE	SP	IT
COMMUNICATION OPTION	OPTION COMMUNICATION	OPTION KOMMUNIKATION	OPCION COMUNICACION	OPZIONE COMUNICAZIONE
PARAMETERS				
PROTOCOL	PROTOCOLE	PROTOKOLL	PROTOCOLO	PROTOCOLO
CONNECT	CONNECT	ANSCHLUSS	CONECTADO	COLLEGAM
ADDRESS	ADRESSE	ADRESSE	DIRECCION	INDIRIZO
BAUDS	BAUDS	BAUDS	BAUDIOS	BAUDS
BITS	BITS	BITS	BITS	BITS
STOP	STOP	STOP	PARAR	STOP
PARITY	PARITE	PARITÄT	PARIDAD	PARITA
SHEDTIME	SHEDTIME	COMAUSFZ	TIEMP ABAND	TEMPO ATT
COM TEST	TEST COMM	KOMM TEST	PRUEBA COM	TEST COM
PROD TEST	PROD TEST	PROD TEST	PRUEBA PROD	TEST PROD
POSSIBLE VALUES				
ODD	IMPAIRE	UNGE	IMP	DI
EVEN	PAIRE	GERA	PAR	PARI
NONE	SANS PARITE	KEI	NADA	NO
FAILED	ECHEC	FEHLERHAFT	MALA	NO RIUSCITO
PASSED	TEST BON	ERFOLGREICH	BUENA	TEST BUONO
WAIT PLEASE	ATTENDEZ SVP	BITTE WARTEN	ESPERE P.FAVOR	ATTENDERE
SWITCH	COMMUTATEUR	UMSHALTG	CONMUTADOR	SCAMBIO