ATEQ

G6 Series – Modbus RTU Manual

Table of Contents

Safety advisory / Warranty

Preamble

Hardware installation

Configuration of the ATEQ device (slave)

Configuration of the master

Frame construction

Functional description of an ATEQ device

ATEQ Manufacturer Plants – Measurement Solution, Global Leader

📍	🌐	☎
ATEQ 15, rue des Dames, Z.I. 78340 LES CLAYES-SOUS-BOIS FRANCE	info@ateq.com ateq.com	T.: +33 1 30 80 1020 F.: +33 1 30 54 1100
ATEQ K.K. 3 – 41 ATEQ Building, Ikehata Chiryu-city, Aichi-pref JAPAN	info@ateq.co.jp ateq.co.jp	T.: +81 566-84-4670 F.: +81 566-84-4680
ATEQ China 98 Jian Peng Lu Shanghai CHINA	shanghai@ateq.com.cn ateq.com.cn	T.: +86 21 6763 9508 F.: +86 21 6763 9528
ATEQ SYSTEMS ANALYSIS TAIWAN CO., LTD. NO. 3, LAN 223, San Jia Dong Street 40642, TAICHUNG TAIWAN	ateqtaiwan@ateq.com.tw ateq.com.tw	T.: +886 4 2437 5278 F.: +886 4 2437 3675
ATEQ CORP. 35980 Industrial Road Suite L Livonia MI 48150 UNITED STATES	leaktest@atequsa.com atequsa.com	T.: +1 734-838-3100 F.: +1 734-838-0644

ℹ We continuously work on improving our products. This is why inforamtion contained in this manual, the device and the technical specifications may be modified without prior notification.

ℹ Pictures and figures in this manual are non contractual

Safety advisory / Warranty

GOOD PRACTICES AND SAFETY INSTRUCTIONS

Safety recommendations

⚠ If the device is supplied with 100 / 240 V AC, it is mandatory to connect it to the ground with a good link to the ground, to protect against electric hazard or electrocution.

⚠ It is dangerous to change the status of the outputs.
They can control power actuators or other equipment (mechanical, pneumatic, hydraulic, electrical or other) which can cause serious personal injury and damage to surrounding material.

⚠ For safety and quality measurement reasons, it is important, before powering on the device, to ensure that it is air supplied with a minimum operating pressure (0.6 MPa ± 15%).

Recommendations for the test environment

Keep the test area as clean as possible.

Recommendations for operators

ATEQ recommends that the operators who use the devices have training and a level of qualification that correspond to the job to perform.

General recommendations

- Read the user manual before using the device.
- All electrical connections to the device must be equipped with safety systems (fuses, circuit breakers, etc.) adapted to the needs and in accordance with the applicable standards and rules.
- To avoid electromagnetic interference, electrical connections to the device must be shorter than 2 meters.
- Power supply plug must be grounded.
- Disconnect the device from the mains before performing any maintenance work.
- Shut off the compressed air supply when working on the pneumatic assembly.
- Do not open a connected device.
- Avoid splashing water on the device.

ATEQ is at your disposal for any information concerning the use of the device under maximum safety conditions.
We draw your attention to the fact that ATEQ cannot be held responsible for any accident related to a misuse of the measuring instrument, the workstation or non-compliance of the installation with safety rules.
In addition, ATEQ declines any responsibility for the calibration or the fitting of their instruments that is not done by ATEQ.
ATEQ also declines any responsibility for any modification (program, mechanical or electrical) of the device done without their written consent.

Preamble

INTRODUCTION

This manual intends to help you for the configuration and the use of your ATEQ G6 device on the Modbus RTU network.

For more information on your ATEQ equipment, refer to the Quick Start Manual.

BASIC NOTIONS

The numerical values used in the ATEQ device are coded on a **Long** format.

> ATEQ devices are configured in **Little Endian format**. It means that the **Least Significant Byte** is **sent first** on the network.

Word

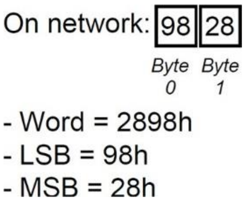
A word is a 16-bit data. It is coded with two bytes (8bits):

— The first byte is the Least Significant Byte (**LSB**)

— The second byte is the Most Significant Byte (**MSB**)

Example of a word:

> Reminder: “**h**” indicates a hexadecimal code, “**(d)**” indicates a decimal code.



On network:	98	28
	Byte 0	*Byte 1*

- Word = 2898h
- LSB = 98h
- MSB = 28h

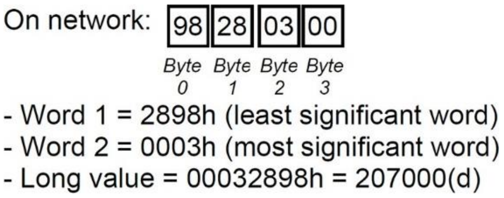
Long format (Signed Double word)

A **Long** format data is coded with two words (of 16 bits).
In the memory range of the ATEQ device or when they are transmitted, both words are coming in the following order:

— The first word is the least significant word

— The second word is the most significant word

— Example of a **Long** format:

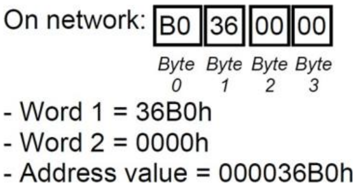


On network:	98	28	03	00
	Byte 0	*Byte 1*	*Byte 2*	*Byte 3*

- Word 1 = 2898h (least significant word)
- Word 2 = 0003h (most significant word)
- Long value = 00032898h = 207000(d)

Address value

All address values are treated with the **Long** format.
Example – address of the “millibar” unit in the Unit table (see Unit table):



On network:	B0	36	00	00
	Byte 0	*Byte 1*	*Byte 2*	*Byte 3*

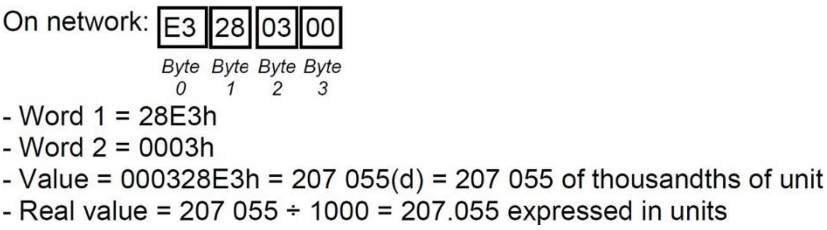
- Word 1 = 36B0h
- Word 2 = 0000h
- Address value = 000036B0h

Numerical value

All the numerical values are treated with the **Long** format with fixed comma (10^{-3}).
Thus, their value is expressed in thousandths of unit. So, this value must be multiplied by 1000 to get the value in units.
For example, a value of 207055 represents 207.055. So, any numerical value must be divided by 1000 to get the real value:

- 207.055 = 207055 ÷ 1000

Example – Pressure:



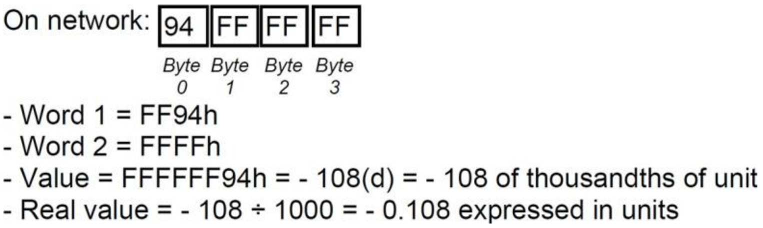
On network: E3 28 03 00

Byte 0 Byte 1 Byte 2 Byte 3

- Word 1 = 28E3h
- Word 2 = 0003h
- Value = 000328E3h = 207 055(d) = 207 055 of thousandths of unit
- Real value = 207 055 ÷ 1000 = 207.055 expressed in units

Negative numerical value

All the negative numerical values are treated with **Signed long** format with fixed comma (10^{-3}).
Thus, they must be multiplied by 1000 to get the value in units.
Example – Leak value (signed long):



On network: 94 FF FF FF

Byte 0 Byte 1 Byte 2 Byte 3

- Word 1 = FF94h
- Word 2 = FFFFh
- Value = FFFFFF94h = - 108(d) = - 108 of thousandths of unit
- Real value = - 108 ÷ 1000 = - 0.108 expressed in units

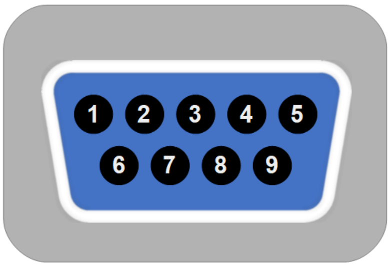
Hardware installation

HARDWARE CONFIGURATION

Connect your ATEQ equipment to the Modbus RTU network using its Modbus RTU connectors and compatible cables.
Your device has one Modbus RTU connector.

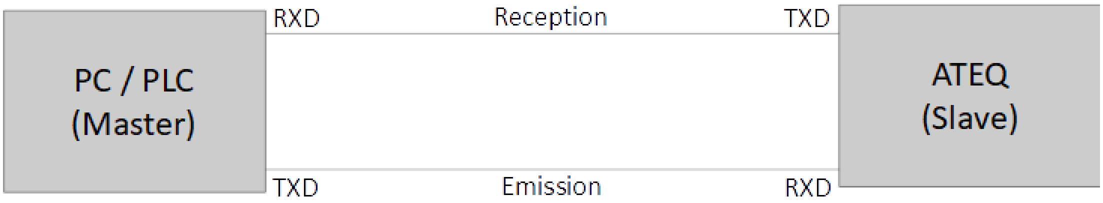
> For more information on your ATEQ equipment, refer to the Quick Start Manual.

Modbus RTU connector – 9 pins male connector

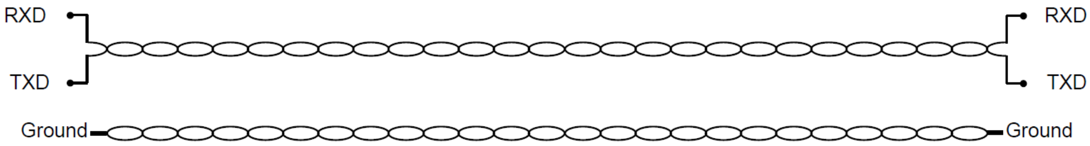


Pin number	Signal
1	-
2	RXD (receive data)
3	TXD (transmit data)
4	-
5	Ground
6	-
7	RTS (request to send)
8	CTS (clear to send)
9	-

Architecture of the Modbus RTU network



The network is built on the basis of a cable composed of two pairs of entwined and shielded wires. One pair is for the signals and the other is for the ground.



Configuration of the ATEQ device (slave)

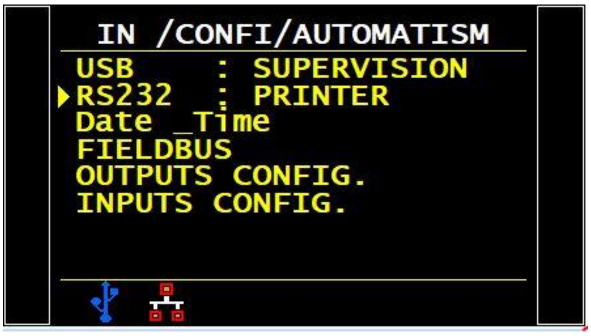
Use this procedure to configure your device.

This configuration can be done with the front panel of your ATEQ device

The Modbus RTU configuration on an ATEQ device is **always 8 bits long with one stop bit**.

SETUP OF THE RS232 MODE

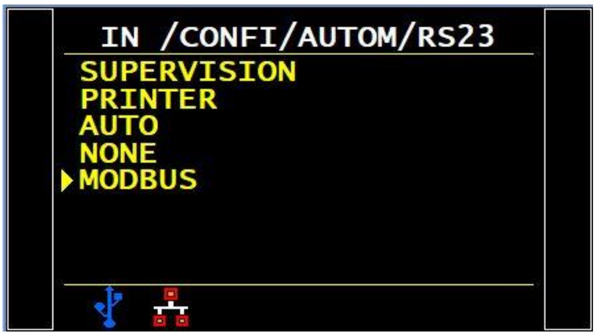
From the ATEQ device



```
 IN  /CONFI/AUTOMATISM
 USB      : SUPERVISION
▶RS232    : PRINTER
 Date _Time
 FIELDBUS
 OUTPUTS CONFIG.
 INPUTS CONFIG.
```

From the **MAIN MENU** screen of your ATEQ device:

- ➢ **CONFIGURATION**
- ➢ **AUTOMATISM**
- ➢ **RS232**



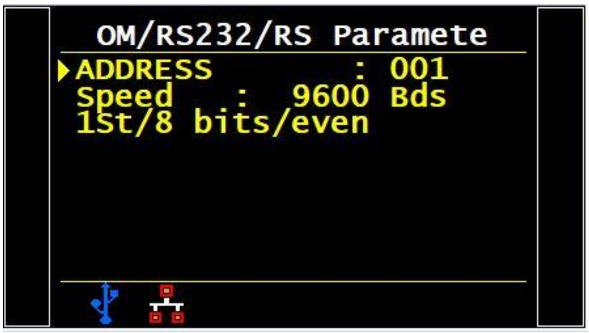
```
 IN  /CONFI/AUTOM/RS23
 SUPERVISION
 PRINTER
 AUTO
 NONE
▶MODBUS
```

Choose **MODBUS** value in the new window.
It will also give you access to the serial parameters.

SETUP OF THE STATION NUMBER

The station number must be the same on slave and master.

From the ATEQ device



```
OM/RS232/RS Paramete
▶ADDRESS        : 001
 Speed    :  9600 Bds
 1St/8 bits/even
```

From the **MAIN MENU** screen of your ATEQ device:

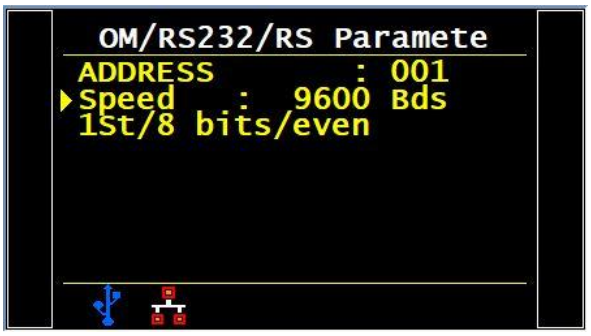
- **CONFIGURATION**
- **AUTOMATISM**
- **RS232: MODBUS**
- **ADDRESS**

The station number can be equal to a value between **1 and 255**.

SETUP OF THE COMMUNICATION SPEED

The **speed** must be the same on slave and master.

From the ATEQ device



```
OM/RS232/RS Paramete
ADDRESS        : 001
Speed   :  9600 Bds
1St/8 bits/even
```

From the **MAIN MENU** screen of your ATEQ device:

- ➢ **CONFIGURATION**
- ➢ **AUTOMATISM**
- ➢ **RS232: MODBUS**
- ➢ **Speed**

The speed can be equal to:

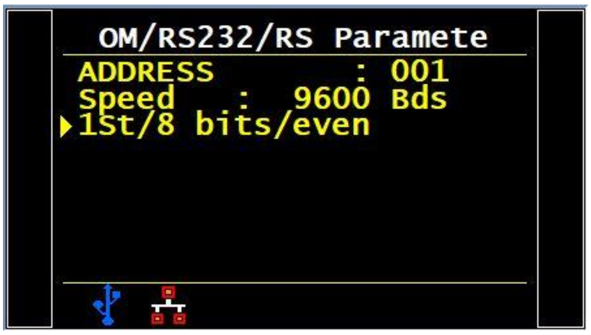
- — **4800 bauds**
- — **9600 bauds**
- — **19200 bauds**
- — **28800 bauds**
- — **38400 bauds**
- — **57600 bauds**

SETUP OF THE PARITY

The **parity** must be the same on slave and master.

The Modbus RTU configuration on an ATEQ device is **always 8 bits long with one stop bit**.

From the ATEQ device



```
OM/RS232/RS Paramete
ADDRESS        : 001
Speed    :   9600 Bds
▸1St/8 bits/even
```

From the **MAIN MENU** screen of your ATEQ device:

- ➢ **CONFIGURATION**
- ➢ **AUTOMATISM**
- ➢ **RS232: MODBUS**

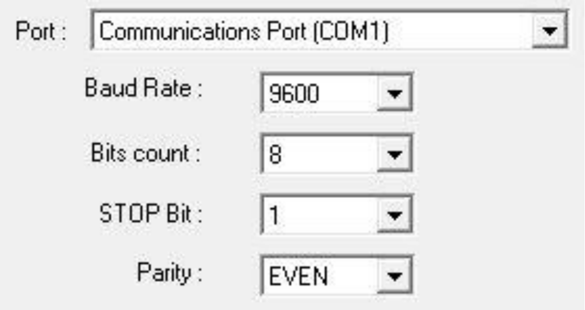
Select the last line in this menu to change the parity.

The parity can be equal to:

- **None**
- **0**
- **1**
- **Even**
- **Odd**

Configuration of the master

SETUP OF THE COMMUNICATION PORT



Port : Communications Port (COM1)

Baud Rate : 9600

Bits count : 8

STOP Bit : 1

Parity : EVEN

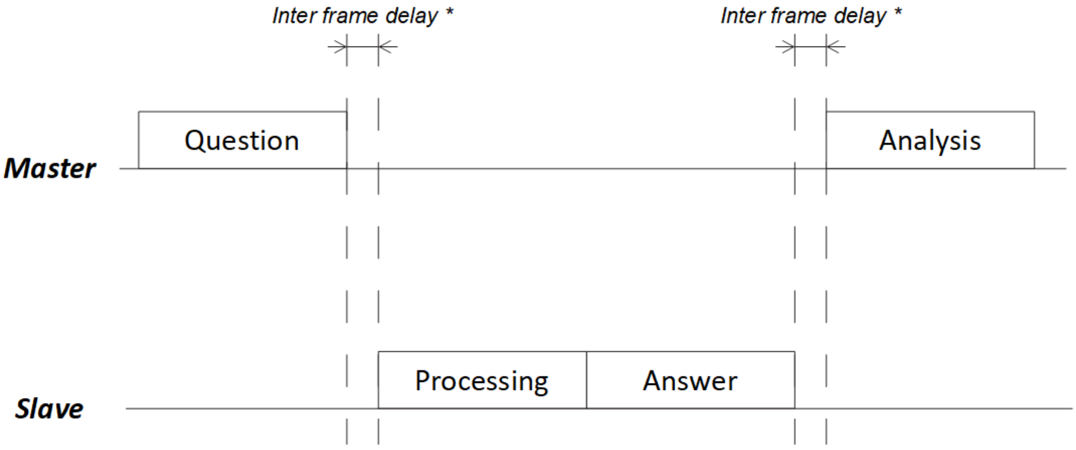
Select the connected communication port and go into its properties.

Then adjust the different settings according to those of your ATEQ device.

Frame construction

DIALOG MECHANISM (ASYNCHRONOUS LINK)

The Modbus RTU data frames do not include delimiters.
The synchronization is achieved by using a delay 3.5 times longer than the transmission time of a byte.
At the end of this delay, the first byte received is considered as the start of a new frame.



* 3,5 times the transmission time of a byte

COMMANDS

Standard access

Reminder: a **byte is 8 bits long** and a **word is 16 bits long**

Reminder: "h" indicates a hexadecimal code, "(d)" indicates a decimal code.

The **Standard access** allows the user to read/write **multiple items** in a single frame.
The ATEQ 6th series instruments support three different functions in **Standard access**.

Writing N*words: 10h

Question:

Slave address	Function number (10h)	Word address	Number of words to write	Number of bytes to write	Data 0	…	Data N	CRC
Byte	*Byte*	*Word*	*Word*	*Byte*		*N*words*		*Word*

Answer:

Slave address	Function number (10h)	Word address	Number of written words	CRC
Byte	*Byte*	*Word*	*Word*	*Word*

Reading N*words: 03h

Question:

Slave address	Function number (03h)	Word address	Number of words to read	CRC
Byte	*Byte*	*Word*	*Word*	*Word*

Answer:

Slave address	Function number (03h)	Number of read bytes	Data 0	…	Data N	CRC
Byte	*Byte*	*Byte*		*N*words*		*Word*

Writing a bit: 05h

Question:

Slave address	Function number (05h)	Bit address	Bit value Force bit to 1: FF00h Force bit to 0: 0000h	CRC
Byte	*Byte*	*Word*	*Word*	*Word*

Answer (identical to the question):

Slave address	Function number (05h)	Bit address	Bit value Force bit to 1: FF00h Force bit to 0: 0000h	CRC
Byte	*Byte*	*Word*	*Word*	*Word*

Direct access

Reminder: a **byte is 8 bits long** and a **word is 16 bits long**

Reminder: "h" indicates a hexadecimal code, "(d)" indicates a decimal code.

The **Direct access** allows the user to read/write **directly only one item** in a single frame.
The ATEQ 6th series instruments support two different functions in **Direct access**.

Writing N*words: 10h

Question:

Slave address	Function number (10h)	Direct access address	Number of words to write	Number of bytes to write	Data 0	…	Data N	CRC
Byte	*Byte*	*Word*	*Word*	*Byte*		*N*words*		*Word*

Answer:

Slave address	Function number (10h)	Direct access address	Number of written words	CRC
Byte	*Byte*	*Word*	*Word*	*Word*

Reading N*words: 03h

Question:

Slave address	Function number (03h)	Direct access address	Number of words to read	CRC
Byte	*Byte*	*Word*	*Word*	*Word*

Answer:

Slave address	Function number (03h)	Number of read bytes	Data 0	…	Data N	CRC
Byte	*Byte*	*Byte*		*N*words*		*Word*

Command error handling

ℹ Reminder: a **byte is 8 bits long** and a **word is 16 bits long**

ℹ Reminder: "h" indicates a hexadecimal code, "(d)" indicates a decimal code.

Error frame

The errors are handled in the answer of the slave to a request of the master.
When an error occurs, the slaves add **80h** to the **Function number** followed by the error code:

— Error on a **Writing N*words (10h)** request

Slave address	Function number + 80h (90h)	Error code	CRC
Byte	*Byte*	*Byte*	*Word*

— Error on a **Reading N*words (03h)** request

Slave address	Function number + 80h (83h)	Error code	CRC
Byte	*Byte*	*Byte*	*Word*

Error codes

Hexa code	Item	Meaning
02	ILLEGAL DATA ADDRESS	Address out of range
03	ILLEGAL DATA VALUE	Value out of limit / value not valid / parameter or bit unavailable

CRC CALCULATION

Definition

In Modbus RTU, the **Cyclic Redundancy Check** is calculated on 16 bits. It is therefore called **CRC16**.

The CRC16 is a calculation based on the binary value of each character composing the frame. This function translates the frame into a 16-bit binary word; this binary word is inserted at the end of the frame.

When the master or the slave receives a frame, it calculates the CRC16 of this frame and compares the result with the value of CRC16 contained in the frame (last word), in order to check that the exchange has been correctly undertaken:

- If the CRC16 corresponds, the slave responds.
- If the CRC16 is false:
  - The slave that receives the erroneous frame does not respond,
  - The master having not received a response restarts the same request for the slave.

> If the exchange is not accomplished after 2 attempts, the master declares a communication error in the network and stops the exchanges.

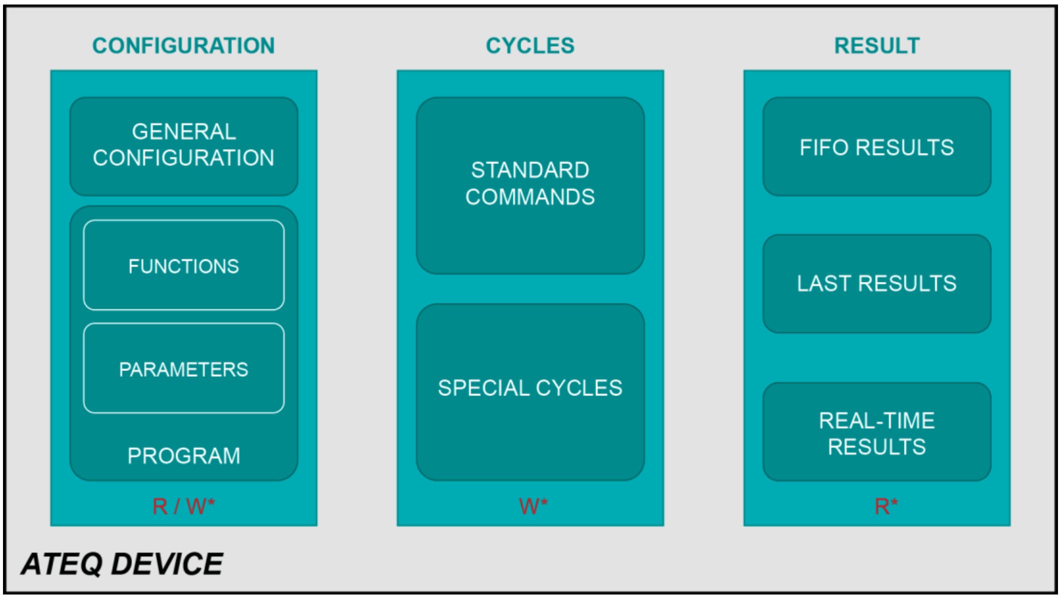
CRC16 calculation algorithm

```
CRC16 = 0FFFFh                  // Initialization at the start of each new data frame

As long as (NO(End of frame))
    CRC16=(CRC16 OR exclusive character received)
    for (i=0;i<8;i++)
    {
        CRC16=CRC16/2
        If there are remainders to the division then
            CRC16= (CRC16 XOR 0A001h)
    }
FTQ
```

Functional description of an ATEQ device

INTRODUCTION



- R/W*: reading and writing
- W*: writing only
- R*: reading only

Address tables

Word addresses

These addresses are used with the **Writing N*words (10h)** or the **Reading N*words (03h)** functions of the **Standard access**:

Hexa address	Item	Read	Write
0000	Read parameters	Y	N
0010	FIFO result	Y	N
0011	Last result	Y	N
0020	Step code in progress	Y	N
0030	Real time result (real time information)	Y	N
007F	Write parameters	N	Y
0100	Extended menu bits	Y	Y
0110	Function bits	Y	Y
0120	Personalization	Y	Y
0130	Number of results in FIFO	Y	N
0200	Program to be selected	N	Y
0201	Special cycle	N	Y
0202	Selected program	Y	N
3004	Program in edition mode	Y	Y

Bit addresses

These addresses are used with the **Writing a bit (05h)** function of the **Standard access**:

Hexa address	Item
0000	Reset
0001	Start
0002	FIFO reset

Direct access addresses

These addresses are used with the **Writing N*words (10h)** or the **Reading N*words (03h)** functions of the **Direct access**:

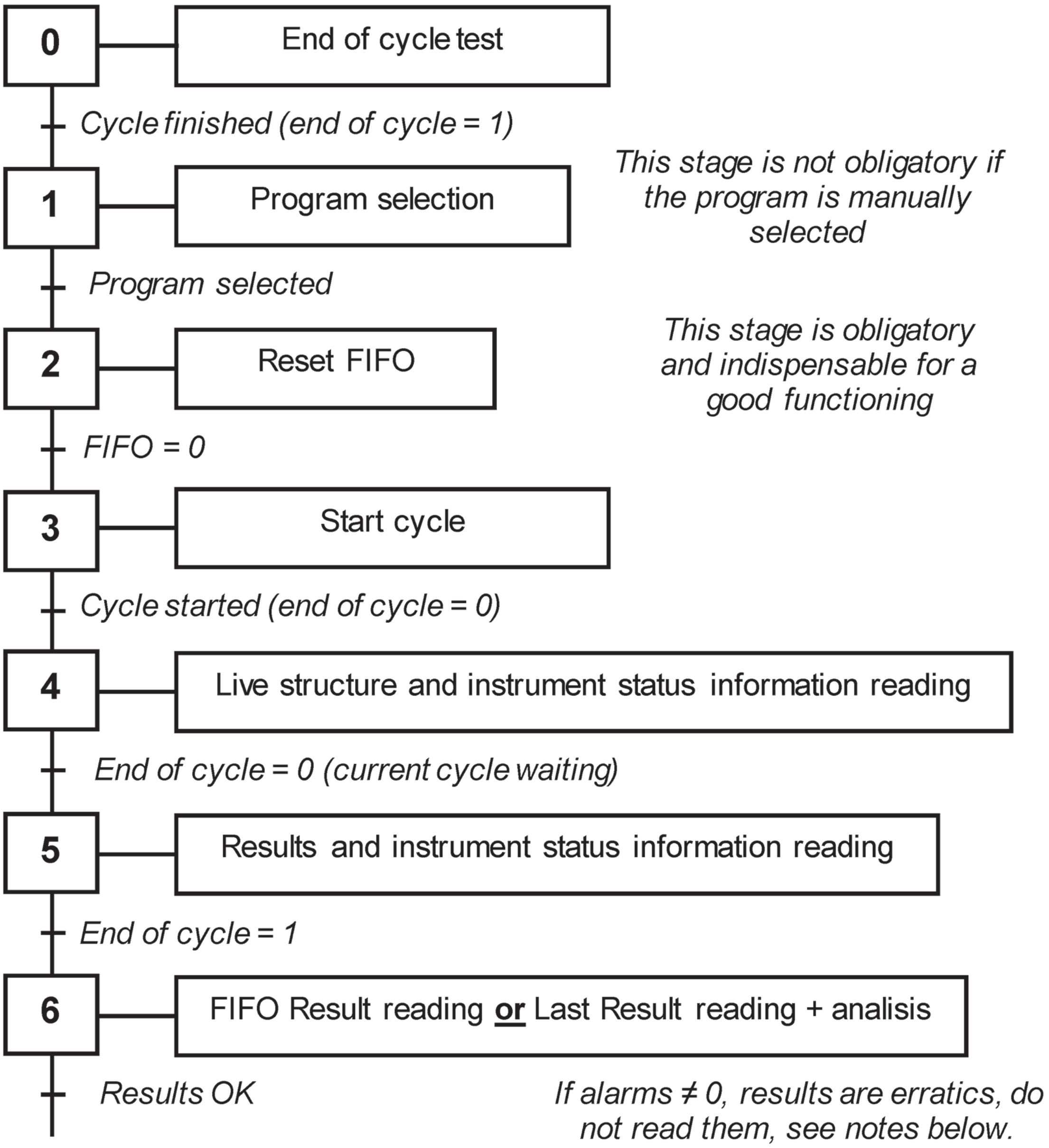
Read hexa address	Write hexa address	Item
2000	6000	Program in edition mode
2001 … 2200	6001 … 6200	Parameters
2201 … 220D	-	Status and real time measurement
2301 … 230C	-	Last result
2401 … 24FF	6401 … 64FF	Extended menu bits
2601 … 26FF	6601 … 66FF	Function bits

Treatment of the commands

Reminder: "h" indicates a hexadecimal code, "(d)" indicates a decimal code.

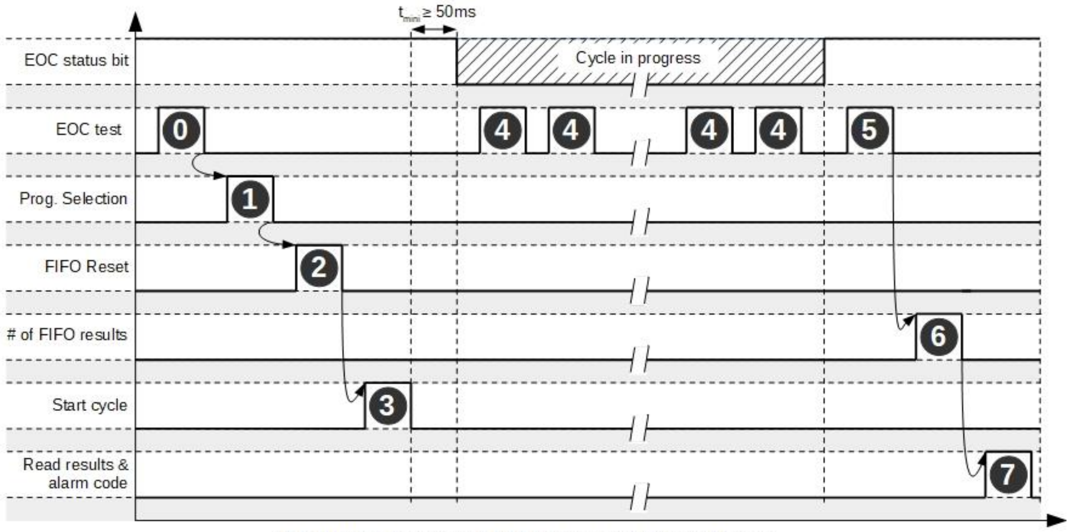
ATEQ device using

Base procedure for using an ATEQ instrument.


0
End of cycle test
Cycle finished (end of cycle = 1)
1
Program selection
This stage is not obligatory if the program is manually selected
Program selected
2
Reset FIFO
This stage is obligatory and indispensable for a good functioning
FIFO = 0
3
Start cycle
Cycle started (end of cycle = 0)
4
Live structure and instrument status information reading
End of cycle = 0 (current cycle waiting)
5
Results and instrument status information reading
End of cycle = 1
6
FIFO Result reading or Last Result reading + analisis
Results OK
If alarms ≠ 0, results are erratics, do not read them, see notes below.

If the number of results in the FIFO = 0, the results are erratic, **do not read them.**
If there's an alarm bit, read the alarm code and **do not use the measurements results (erratic results).**

Modbus progress chart



WARNING : The status bits update rate is about 50ms

0: Read 13 words @30h : word 4, bit 5 = 1 (EOC status bit)

1: Write 1 word @200h : word = n° prog (0001h = prog 2)

2: **ALWAYS RESET THE FIFO**
Write bit @02h : bit = FF (command « Reset FIFO »)

3: Write bit @01h : bit = FF (command « Start »)
t_{mini} ≥ 50ms

4: Read 13 words @30h : word 4, bit 5 = 0 (EOC status bit)

5: Read 13 words @30h : word 4, bit 5 = 1 (EOC status bit)

Use of FIFO Results

6: *Read the number of results in FIFO :*
Read 13 words @30h : **if word 2 ≥ 1** go to step 7, **else** END

7: Read 12 words @10h : 12 words (size of standard results)
if Alarm Code = 0 go to step 8, **else** END

8: Use the results recovered at step 7 (if Alarm code was equal to 0)

Use of Last Results

6: *Read the number of results in FIFO :*
Read 13 words @30h : **if word 2 ≥ 1** go to step 7, **else** END

7: Read 12 words @11h : 12 words (size of standard results)
if Alarm Code = 0 go to step 8, **else** END

8: Use the results recovered at step 7 (if Alarm code was equal to 0)

General configuration

Table of the configuration / extended menus bits

> Reminder: Direct access addresses are expressed in hexadecimal

The bits below are mostly present in the **CONFIGURATION** or **More functions...** menus.
They are only used to allow the access to other parameters according to the configuration, depending on the configuration, these are active or not.

> Acronyms used in the "Menu" column:
> - Conf: CONFIGURATION
> - +Func: FUNCTIONS > More functions...
> - RS232: CONFIGURATION > RS232

Word	Bit n°	Mask		D.A. address		Meaning	Menu
		Hexa	Dec	Read	Write		
1	0	0001	1	2404	6404	Fill type.	+Func
	1	0002	2	2403	6403	Pre-fill type.	+Func
	2	0004	4	2401	6401	Recovery thresholds.	+Func
	3	0008	8	241E	641E	Volume calculation	+Func
	4	0010	16	2413	6413	Personalization of the program name.	+Func
	5	0020	32	241F	641F	Chaining.	+Func
	6	0040	64	2420	6420	Automatic connector.	+Func
	7	0080	128	2416	6416	Valves codes (outputs codes)	+Func
	8	0100	256	2422	6422	Stamping.	+Func
	9	0200	512	2426	6426	Sending conditions: pass part	RS232
	10	0400	1024	2427	6427	Sending conditions: fail part maximum flow	RS232
	11	0800	2048	2429	6429	Sending conditions: presence of an alarm	RS232
	12	1000	4096	242A	642A	Sending conditions: pressure defect	RS232
	13	2000	8192	242B	642B	Sending conditions: end of cycle	RS232
	14	4000	16384	242C	642C	Sending conditions: recoverable	RS232
	15	8000	32768	242D	642D	Content of the frame: time	RS232

Word	Bit n°	Mask		D.A. address		Meaning	Menu
		Hexa	Dec	Read	Write		
2	16	0001	1	2412	6412	Content of the frame: personalization	RS232
	17	0002	2	242E	642E	Content of the frame: pressure	RS232
	18	0004	4	242F	642F	Security	Conf
	19	0008	8	2414	6414	External dump	Conf
	20	0010	16	2430	6430	Exportation	RS232
	21	0020	32	240F	640F	Automatic reset	Conf
	22	0040	64			*Reserved*	
	23	0080	128			*Reserved*	
	24	0100	256			*Reserved*	
	25	0200	512	2419	6419	Automatic start	+Funct
	26	0400	1024	2461	6461	Cut valve	Conf
	27	0800	2048	2409	6409	Filtering	+Funct
	28	1000	4096			*Reserved*	
	29	2000	8192	2406	6406	Pressure compensation	+Funct
	30	4000	16384			*Reserved*	
	31	8000	32768	2439	6439	Line feed (label)	RS232
3	32	0001	1	241C	641C	End of cycle	+Funct
	33	0002	2	2418	6418	Unit type	+Funct
	34	0004	4	243A	643A	Bar graph display	Conf
	35	0008	8	2462	6462	Negative rejects level	Conf
	36	0010	16			*Reserved*	
	37	0020	32	2443	6443	Bar code	RS232
	38	0040	64	249D	649D	Program selection bar code	
	39	0080	128	2492	6492	Bar code reset on end of cycle	
	40	0100	256	2435	6435	Auxiliary code activation	+Funct
	41	0200	512	24B7	64B7	Standard conditions	+Funct
	42	0400	1024			*Reserved*	
	43	0800	2048	2440	6440	Service cycle activation	
	44	1000	4096	2434	6434	Sign change activation	+Funct
	45	2000	8192	2408	6408	Peak hold	+Funct
	46	4000	16384	2477	6477	Negative flow display	+Funct
	47	8000	32768			*Reserved*	

Word	Bit n°	Mask		D.A. address		Meaning *ERD Only	Menu
		Hexa	Dec	Read	Write		
4	48	0001	1	249B	649B	Buzzer	+Funct
	49	0002	2	24C0	64C0	Display mode activation	+Funct
	50	0004	4	244B	644B	Sending conditions: fail part minimum flow	RS232
	51	0008	8	24D2	64D2	Offset	+Funct
	52	0010	16	24D3	64D3	Minimum flow activation	+Funct

Example: bit number 13 (Sending conditions: end of cycle) activated to 1, will place to "2000h" the value in the first word.

2000h is equivalent to 8192 in decimal and 0010000000000000 in binary.

In the Modbus frame, the words will follow each other: word 1 + word 2 + ….. + word n.

Reading of the configuration / extended menu bits

The configuration / extended menu bits are independents of the program number.

- *Standard access*

Example of reading 4 words of the “Configuration / Extended menu bits”:

Master

— Make a **Read N*words** request of 4 words at the **0100h** address.

On network:

01	03	01	00	00	04	45	F5

01	Slave address
03	Function number (Read N*words)
01 00	Word address (Extended menu bits)
00 04	Number of words to read
45 F5	CRC

Slave

— Answer to the request:

On network:

01	03	08	00	20	00	10	00
80	20	00	6D	FE			

01	Slave address
03	Function number (Read N*words)
08	Number of read bytes
00 20	Word 1: read 2000h
00 10	Word 2: read 1000h
00 80	Word 3: read 8000h
20 00	Word 4: read 0020h
6D FE	CRC

- *Direct access*

In Direct access, the master can only access to bits one by one.

Example for reading the state of the “Chaining” bit (word 1, bit 5):

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td>— Make a <b>Read N*words</b> request of 1 word at the <b>241Fh</b> address.
On network:
<table><tr><td>01</td><td>03</td><td>24</td><td>1F</td><td>00</td><td>01</td><td>BF</td><td>3C</td></tr></table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>24 1F</td><td>D.A. address of the “Chaining” bit</td></tr>
<tr><td>00 01</td><td>Number of words to read</td></tr>
<tr><td>BF 3C</td><td>CRC</td></tr>
</table></td><td></td></tr>
<tr><td></td><td>— Answer to the request:
On network:
<table><tr><td>01</td><td>03</td><td>02</td><td>01</td><td>00</td><td>B9</td><td>D4</td></tr></table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>02</td><td>Number of read bytes</td></tr>
<tr><td>01 00</td><td>Word: read 0001h (Chaining bit = 1)</td></tr>
<tr><td>B9 D4</td><td>CRC</td></tr>
</table></td></tr>
</table>

Writing of the configuration / extended menu bits

The configuration / extended menu bits are independents of the program number.

- *Standard access*

Example of writing 4 words in the “Configuration / Extended menu bits”:

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td>

— Make a **Write N*words** request of 4 words at the **0100h** address.

On network:

<table>
<tr><td>01</td><td>10</td><td>01</td><td>00</td><td>00</td><td>04</td><td>08</td><td>00</td></tr>
<tr><td>A0</td><td>00</td><td>10</td><td>00</td><td>80</td><td>20</td><td>00</td><td>CD</td></tr>
<tr><td>1A</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>01 00</td><td>Word address (Extended menu bits)</td></tr>
<tr><td>00 04</td><td>Number of words to write</td></tr>
<tr><td>08</td><td>Number of bytes to write</td></tr>
<tr><td>00 A0</td><td>Word 1: write A000h</td></tr>
<tr><td>00 10</td><td>Word 2: write 1000h</td></tr>
<tr><td>00 80</td><td>Word 3: write 8000h</td></tr>
<tr><td>20 00</td><td>Word 4: write 0022h</td></tr>
<tr><td>CD 1A</td><td>CRC</td></tr>
</table>

</td><td></td></tr>
<tr><td></td><td>

— Answer to the request:

On network:

<table>
<tr><td>01</td><td>10</td><td>01</td><td>00</td><td>00</td><td>04</td><td>C0</td><td>36</td></tr>
</table>

<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>01 00</td><td>Word address (Extended menu bits)</td></tr>
<tr><td>00 04</td><td>Number of written words</td></tr>
<tr><td>C0 36</td><td>CRC</td></tr>
</table>

</td></tr>
</table>

- *Direct access*

In Direct access, the master can only access to bits one by one.

Example for writing the “Chaining” bit to 1 (word 1, bit 5):

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td>— Make a <b>Write N*words</b> request of 1 word at the <b>641Fh</b> address.
On network:
<table>
<tr><td>01</td><td>10</td><td>64</td><td>1F</td><td>00</td><td>01</td><td>02</td><td>01</td></tr>
<tr><td>00</td><td>80</td><td>69</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>64 1F</td><td>D.A. address of the “Chaining” bit</td></tr>
<tr><td>00 01</td><td>Number of words to write</td></tr>
<tr><td>02</td><td>Number of bytes to write</td></tr>
<tr><td>01 00</td><td>Word: write 0001h (Chaining bit = 1)</td></tr>
<tr><td>80 69</td><td>CRC</td></tr>
</table>
</td><td></td></tr>
<tr><td></td><td>— Answer to the request:
On network:
<table>
<tr><td>01</td><td>10</td><td>64</td><td>1F</td><td>00</td><td>01</td><td>2F</td></tr>
<tr><td>3F</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>64 1F</td><td>D.A. address of the “Chaining” bit</td></tr>
<tr><td>00 01</td><td>Number of written words</td></tr>
<tr><td>2F 3F</td><td>CRC</td></tr>
</table>
</td></tr>
</table>

Program

Program in edition mode command on the ATEQ device

> Always subtract 1 from the value of the program number to be put in edition mode.
> Example: for putting program number 2 in edition mode, send the value 1 at the address 3004h.

- *Standard access*

Example for putting program number 3 in edition mode:

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td>

— Make a **Write N*words** request of 1 word at the address **3004h**.

On network:

<table>
<tr><td>01</td><td>10</td><td>30</td><td>04</td><td>00</td><td>01</td><td>02</td><td>02</td></tr>
<tr><td>00</td><td>96</td><td>B7</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>30 04</td><td>Word address (Program in edition mode)</td></tr>
<tr><td>00 01</td><td>Number of words to write</td></tr>
<tr><td>02</td><td>Number of bytes to write</td></tr>
<tr><td>02 00</td><td>Word: write 0002h (Program n°3)</td></tr>
<tr><td>96 B7</td><td>CRC</td></tr>
</table>

</td><td></td></tr>
<tr><td></td><td>

— Answer to the request:

On network:

<table>
<tr><td>01</td><td>10</td><td>30</td><td>04</td><td>00</td><td>01</td><td>4F</td></tr>
<tr><td>08</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>30 04</td><td>Word address (Program in edition mode)</td></tr>
<tr><td>00 01</td><td>Number of words to write</td></tr>
<tr><td>4F 08</td><td>CRC</td></tr>
</table>

</td></tr>
</table>

- *Direct access*

Example for putting program number 3 in edition mode:

Master

— Make a **Write N*words** request of 1 word at the address **6000h**.

On network:

01	10	60	00	00	01	02	02
00	C7	36					

01	Slave address
10	Function number (Write N*words)
60 00	D.A. address for program in edition mode
00 01	Number of words to write
02	Number of bytes to write
02 00	Word: write 0002h (Program n°3)
C7 36	CRC

Slave

— Answer to the request:

On network:

01	10	60	00	00	01	1F
C9						

01	Slave address
10	Function number (Write N*words)
60 00	D.A. address for program in edition mode
00 01	Number of words to write
1F C9	CRC

Function

Table of the function bits

Table of the function bits per program.

> Reminder: **Direct access addresses** are expressed in **hexadecimal**

The bits below are present in the **FUNCTIONS** menu of each program, if these have been previously validated in the **More functions...** menu.

Word	Bit n°	Mask		D.A. address		Meaning *ERD Only	Menu
		Hexa	Dec	Read	Write		
1	0	0001	1	2604	6604	Fill type activation	Funct
	1	0002	2	2603	6603	Pre-fill type activation	Funct
	2	0004	4	2601	6601	Recovery thresholds activation	Funct
	3	0008	8	261E	661E	Cycle end activation	Funct
	4	0010	16	261F	661F	Cycle end with reset and piezo reset activation	
	5	0020	32	2620	6620	Cycle end with dump and reset activation	
	6	0040	64	2621	6621	Cycle end with fill activation	
	7	0080	128	2622	6622	Chaining activation	Funct
	8	0100	256	2623	6623	Pass part chaining activation	
	9	0200	512	2625	6625	Fail part maximum flow chaining activation	
	10	0400	1024	2625	6625	Alarm chaining activation	
	11	0800	2048	2626	6626	Pressure switch error chaining activation	
	12	1000	4096	2627	6627	Cycle end chaining activation	
	13	2000	8192	262A	662A	Recovery chaining activation	
	14	4000	16384	262B	662B	Automatic connector chaining activation	Funct
	15	8000	32768	2612	6612	Valve code activation	
2	16	0001	1	2613	6613	Valve code ext. 1 activation	
	17	0002	2	2614	6614	Valve code ext. 2 activation	
	18	0004	4	2615	6615	Valve code ext. 3 activation	
	19	0008	8	2616	6616	Valve code ext. 4 activation	
	20	0010	16	2617	6617	Valve code ext. 5 activation	
	21	0020	32	2618	6618	Valve code ext. 6 activation	
	22	0040	64	2619	6619	Valve code int. 1 activation	
	23	0080	128	261A	661A	Valve code int. 8 activation	
	24	0100	256	262C	662C	Stamping activation	Funct
	25	0200	512	262D	662D	Pass part stamping activation	
	26	0400	1024	262E	662E	Fail part maximum flow stamping activation	
	27	0800	2048	2630	6630	Alarm stamping activation	
	28	1000	4096	2631	6631	Pressure switch error stamping activation	
	29	2000	8192	2632	6632	Cycle end stamping activation	
	30	4000	16384	2633	6633	Recovery stamping activation	
	31	8000	32768	261B	661B	External dump activation	Funct

Word	Bit n°	Mask		D.A. address		Meaning *ERD Only	Menu
		Hexa	Dec	Read	Write		
3	32	0001	1			Reserved	
	33	0002	2	261C	661C	Automatic start cycle activation	Funct
	34	0004	4	2606	6606	Pressure compensation activation	Funct
	35	0008	8	2609	6609	Filtering activation	Funct
	36	0010	16	261D	661D	Standard conditions activation	Funct
	37	0020	32	264D	664D	Bar code activation	
	38	0040	64	264F	664F	Start after reading bar code	
	39	0080	128	2638	6638	Auxiliaries code activation	
	40	0100	256	2639	6639	Auxiliary code 1 activation	
	41	0200	512	263A	663A	Auxiliary code 2 activation	
	42	0400	1024	263B	663B	Auxiliary code 3 activation	
	43	0800	2048	263C	663C	Auxiliary code 4 activation	
	44	1000	4096	267D	667D	Optional auxiliaries code activation	
	45	2000	8192	267E	667E	Optional auxiliary code 1 activation	
	46	4000	16384	267F	667F	Optional auxiliary code 2 activation	
	47	8000	32768	2680	6680	Optional auxiliary code 3 activation	
4	48	0001	1	2681	6681	Optional auxiliary code 4 activation	
	49	0002	2	2682	6682	Optional valve code activation	
	50	0004	4	2683	6683	Optional valve code ext. 1 activation	
	51	0008	8	2684	6684	Optional valve code ext. 2 activation	
	52	0010	16	2685	6685	Optional valve code ext. 3 activation	
	53	0020	32	2686	6686	Optional valve code ext. 4 activation	
	54	0040	64	2687	6687	Optional valve code ext. 5 activation	
	55	0080	128	2688	6688	Optional valve code ext. 6 activation	
	56	0100	256	2689	6689	Optional valve code int. 1 activation	
	57	0200	512	268A	668A	Optional valve code int. 2 activation	
	58	0400	1024	2611	6611	Sign change activation	Funct
	59	0800	2048	2608	6608	Peak hold activation	Funct
	60	1000	4096	2668	6668	Negative flow display activation	Funct
	61	2000	8192	268B	668B	Buzzer activation	
	62	4000	16384	268C	668C	Cycle end buzzer activation	
	63	8000	32768	268D	668D	Pass part buzzer activation	

Word	Bit n°	Mask		D.A. address		Meaning *ERD Only	Menu
		Hexa	Dec	Read	Write		
5	64	0001	1	268E	668E	Fail part maximum flow buzzer activation	
	65	0002	2	268F	668F	Alarm buzzer activation	Funct
	66	0004	4	2650	6650	Automatic mode activation	Funct
	67	0008	8			Reserved	
	68	0010	16			Reserved	
	69	0020	32			Reserved	
	70	0040	64	26BF	66BF	Offset activation	Funct
	71	0080	128	26C1	66C1	Minimum flow activation	Funct

Example: bit number 14 (Automatic connector chaining activation) activated on 1, will put to "4000h" the value in the first word.
4000h is equivalent to 16384 in decimal and 0100000000000000 in binary.
In the Modbus frame, the words will follow as such: word 1 + word 2 + ... + word n.

Reading of the function bits

The functions bits are dependents of the program number.
Put the wanted program in edition before executing command.

- *Standard access*

Example for reading 5 words of the "Function bits":

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td>

— Put in edition the program number on which the functions bits have to be read

— Make a **Read N*words** request of 5 words at the **0110h** address.

On network:

<table><tr><td>01</td><td>03</td><td>01</td><td>10</td><td>00</td><td>05</td><td>85</td><td>F0</td></tr></table>

<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>01 10</td><td>Word address (Function bits)</td></tr>
<tr><td>00 05</td><td>Number of words to read</td></tr>
<tr><td>85 F0</td><td>CRC</td></tr>
</table>

</td><td></td></tr>
<tr><td></td><td>

— Answer to the request:

On network:

<table>
<tr><td>01</td><td>03</td><td>0A</td><td>00</td><td>80</td><td>00</td><td>00</td><td>10</td></tr>
<tr><td>00</td><td>00</td><td>10</td><td>00</td><td>00</td><td>46</td><td>25</td><td></td></tr>
</table>

<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>0A</td><td>Number of read bytes</td></tr>
<tr><td>00 80</td><td>Word 1: read 8000h</td></tr>
<tr><td>00 00</td><td>Word 2: read 0000h</td></tr>
<tr><td>10 00</td><td>Word 3: read 0010h</td></tr>
<tr><td>00 10</td><td>Word 4: read 1000h</td></tr>
<tr><td>00 00</td><td>Word 5: read 0000h</td></tr>
<tr><td>46 25</td><td>CRC</td></tr>
</table>

</td></tr>
</table>

- *Direct access*

In Direct access, the master can only access to bits one by one.

Example for reading the state of the “Sequencing activation” bit (word 1, bit 7):

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr>
<td>
— Put in edition the program number on which the functions bits have to be read
— Make a <b>Read N*words</b> request of 1 word at the <b>2622h</b> address.
On network:
<table><tr><td>01</td><td>03</td><td>26</td><td>22</td><td>00</td><td>01</td><td>2F</td><td>48</td></tr></table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>26 22</td><td>D.A. address of the “Sequencing act.” bit</td></tr>
<tr><td>00 01</td><td>Number of words to read</td></tr>
<tr><td>2F 48</td><td>CRC</td></tr>
</table>
</td>
<td></td>
</tr>
<tr>
<td></td>
<td>
— Answer to the request:
On network:
<table><tr><td>01</td><td>03</td><td>02</td><td>01</td><td>00</td><td>B9</td><td>D4</td></tr></table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>02</td><td>Number of read bytes</td></tr>
<tr><td>01 00</td><td>Word: read 0001h (Sequencing act. bit = 1)</td></tr>
<tr><td>B9 D4</td><td>CRC</td></tr>
</table>
</td>
</tr>
</table>

Writing of the function bits

> The functions bits are dependents of the program number.
> Put the wanted program in edition before executing command.

- *Standard access*

Example of writing 5 words in the “Function bits”:

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td>
— Put in edition the program number on which the functions bits have to be read
— Make a <b>Write N*words</b> request of 5 words at the <b>0110h</b> address.
On network:
<table>
<tr><td>01</td><td>10</td><td>01</td><td>10</td><td>00</td><td>05</td><td>0A</td><td>00</td></tr>
<tr><td>A0</td><td>00</td><td>00</td><td>10</td><td>00</td><td>00</td><td>10</td><td>00</td></tr>
<tr><td>00</td><td>2A</td><td>7A</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>01 10</td><td>Word address (Function bits)</td></tr>
<tr><td>00 05</td><td>Number of words to write</td></tr>
<tr><td>0A</td><td>Number of bytes to write</td></tr>
<tr><td>00 A0</td><td>Word 1: write A000h</td></tr>
<tr><td>00 00</td><td>Word 2: write 0000h</td></tr>
<tr><td>10 00</td><td>Word 3: write 0010h</td></tr>
<tr><td>00 10</td><td>Word 4: write 1000h</td></tr>
<tr><td>00 00</td><td>Word 5: write 0000h</td></tr>
<tr><td>2A 7A</td><td>CRC</td></tr>
</table>
</td><td></td></tr>
<tr><td></td><td>
— Answer to the request:
On network:
<table>
<tr><td>01</td><td>10</td><td>01</td><td>10</td><td>00</td><td>05</td><td>00</td><td>33</td></tr>
</table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>01 10</td><td>Word address (Function bits)</td></tr>
<tr><td>00 05</td><td>Number of written words</td></tr>
<tr><td>00 33</td><td>CRC</td></tr>
</table>
</td></tr>
</table>

- *Direct access*

In Direct access, the master can only access to bits one by one.

Example for writing the state of the “Sequencing activation” bit (word 1, bit 7):

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td>
— Put in edition the program number on which the functions bits have to be read
— Make a <b>Write N*words</b> request of 1 word at the <b>6622h</b> address.
On network:
<table>
<tr><td>01</td><td>10</td><td>66</td><td>22</td><td>00</td><td>01</td><td>02</td><td>01</td></tr>
<tr><td>00</td><td>A7</td><td>44</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>66 22</td><td>D.A. address of the “Sequencing act.” bit</td></tr>
<tr><td>00 01</td><td>Number of words to write</td></tr>
<tr><td>02</td><td>Number of bytes to write</td></tr>
<tr><td>01 00</td><td>Word: write 0001h (Sequencing act. = 1)</td></tr>
<tr><td>A7 44</td><td>CRC</td></tr>
</table>
</td><td></td></tr>
<tr><td></td><td>
— Answer to the request:
On network:
<table>
<tr><td>01</td><td>10</td><td>66</td><td>22</td><td>00</td><td>01</td><td>BF</td></tr>
<tr><td>4B</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>66 22</td><td>D.A. address of the “Sequencing act.” bit</td></tr>
<tr><td>00 01</td><td>Number of written words</td></tr>
<tr><td>BF 4B</td><td>CRC</td></tr>
</table>
</td></tr>
</table>

Parameters

Downloading of the parameters

Reminder: **Direct access addresses** are expressed in **hexadecimal**

All the parameters values below have a tratment by the ATEQ device as **Long** format with fixed comma (10^{-3}). A **Long** is a two words set.

Identifier N°		D.A. address		Meaning	Value
Dec	Hexa	Read	Write		
1	0001			"FILL TIME" Fill time	0 > 650 seconds
2	0002			"STAB TIME": Stabilization time	0 > 650 seconds
3	0003			"TEST TIME" Test time	0 > 650 seconds
6	0006			"PRE FILL" Pre fill time	0 > 650 seconds
9	0009			"DUMP TIME" Dump time	0 > 650 seconds
10	000A			"COUPL. A": Coupling time 1	0 > 650 seconds
11	000B			"COUPL. B": Coupling time 2	0 > 650 seconds
20	0014			"VOLUME" Part volume.	0 > 9999
21	0015			"TYPE": Test type	Invalid 0000 Direct 1000 Operator 2000
29	001D			"Inter-Cycle": Time between 2 chained cycles	0 > 650 seconds
48	0030			"DURATION" Maintain time of the result during stamp	0 > 650 seconds
50	0032			"Min FILL" Minimum pressure value	- 9999 > 9999.
51	0033			"Max FILL" Maximum pressure value	- 9999 > 9999.
53	0035			"Press. UNIT" Pressure unit.	Refer to Unit table.
60	003C			"Test FAIL" Natural reject value of the test part	0 > 9999
61	003D			"TestREWORK" Natural reject level of the test part in recovery	0 > 9999
62	003E			"Ref. FAIL" Natural reject level of the reference part	0 > 9999
63	003F			"Ref.REWORK" Natural reject value of the reference part in recovery:	0 > 9999
66	0042			"Set FILL" Fill instruction value:	-9999 > 9999

Identifier N°		D.A. address		Meaning	Value	
Dec	Hexa	Read	Write			
80	0050			“Diff A-Z” Differential auto reset time.	0 > 650 seconds	
103	0067			“FILL MODE” Type of fill.	Standard Instruction Ballistic Ramp Adjust EASY EASY Auto	0000 1000 2000 3000 4000 5000 6000
110	006E			“EXT. DUMP” Type of external dump.	Normally close Normally open	0000 1000
112	0070			‘IN7:” Function attributed to the entry of the special cycles (output 7)	Refer to the “Configure input value” table at the end of the chapter	
123	007B			“LANGUAGE” Choice of the language.	Default language 2nd language	0000 1000
126	007E			“Max PreFILL” Maximum pressure value in pre-fill.	-9999 > 9999	
127	007F			“Flow Unit” Reject unit.	Refer to Unit table.	
128	0080			“Leak Rate” Instruction value during a calibration.	0 > 9999	
148	0094			“FILTER” Filtering.	0 > 650 seconds	
149	0095			“UNITS” Unit type	SI SAE CUSTOM	0000 1000 2000
158	009E			“Max rej.” Percents of the bar graph.	70% 50% 30%	0000 1000 2000
161	00A1			“Volume UNIT” Volume unit.	Refer to Unit table.	
164	00A4			“NEXT PROG.” Number of the following program in sequencing.	1 > 128	
165	00A5			“N. OF CYCLES”(PIEZO AUTO AZ menu) Number of cycles between two automatic reset.	0 > 9999	
166	00A6			“N. OF MINUTES”(PIEZO AUTO AZ menu) Time between two automatic reset.	0 > 999 minutes	
249	00F9			“DELAY EXT1” Programmed external output 1 delay time.	0 > 650 seconds	
250	00FA			“DELAY EXT2” Programmed external output 2 delay time.	0 > 650 seconds	
251	00FB			“DELAY EXT3” Programmed external output 3 delay time.	0 > 650 seconds	
252	00FC			“DELAY EXT4” Programmed external output 4 delay time.	0 > 650 seconds	
253	00FD			“DELAY EXT5” Programmed external output 5 delay time.	0 > 650 seconds	
254	00FE			“DELAY EXT6” Programmed external output 6 delay time.	0 > 650 seconds	

Identifier N°		D.A. address		Meaning	Value	
Dec	Hexa	Read	Write			
255	00FF			“DELAY INT2” Programmed internal output 2 delay time.	0 > 650 seconds	
256	0100			“DELAY INT1” Programmed internal output 1 delay time.	0 > 650 seconds	
257	0101			“DELAY AUX1” Programmed auxiliary output 1 delay time.	0 > 650 seconds	
258	0102			“DELAY AUX2” Programmed auxiliary output 2 delay time.	0 > 650 seconds	
259	0103			“DELAY AUX3” Programmed auxiliary output 3 delay time.	0 > 650 seconds	
260	0104			“DELAY AUX4” Programmed auxiliary output 4 delay time.	0 > 650 seconds	
261	0105			“TIME EXT1” Programmed external output 1 duration time.	0 > 650 seconds	
262	0106			“TIME EXT2” Programmed external output 2 duration time.	0 > 650 seconds	
263	0107			“TIME EXT3” Programmed external output 3 duration time.	0 > 650 seconds	
264	0108			“TIME EXT4” Programmed external output 4 duration time.	0 > 650 seconds	
265	0109			“TIME EXT5” Programmed external output 5 duration time.	0 > 650 seconds	
266	010A			“TIME EXT6” Programmed external output 6 duration time.	0 > 650 seconds	
267	010B			“TIME INT2” Programmed internal output 2 duration time.	0 > 650 seconds	
268	010C			“TIME INT1” Programmed internal output 1 duration time.	0 > 650 seconds	
269	010D			“TIME AUX1” Programmed auxiliary output 1 duration time.	0 > 650 seconds	
270	010E			“TIME AUX2” Programmed auxiliary output 2 duration time.	0 > 650 seconds	
271	010F			“TIME AUX3” Programmed auxiliary output 3 duration time.	0 > 650 seconds	
272	0110			“TIME AUX4” Programmed auxiliary output 4 duration time.	0 > 650 seconds	
274	0112			“FILTER” Pressure filtering.	0 > 650 seconds	
281	0119			“RANGE” Capillary number with dual capillaries option only:	Capillary 1 Capillary 2	0000 1000
287	011F			“First Char.” Start on bar code.	0 > 40	
288	0120			“Char. Number” Number of character of bar code.	0 > 40	
289	0121			“Pr “ Program bar code.	1 > 128	
353	0161			“Press. UNIT” (configuration/pneumatique menu) General pressure unit	Refer to Unit table.	
354	0162			“LINE P. MIN” Minimum line pressure level	-9999 > 9999	

Identifier N°		D.A. address		Meaning	Value	
Dec	Hexa	Read	Write			
364	016C			“DISPLAY MODE“ Leak display management	XXXX XXX.X XX.XX X.XXX	0000 1000 2000 3000
375	0177			‘IN8:” Function attributed to the entry of the special cycles (output 8)	Refer to the “Configure input value” table at the end of the chapter	
376	0178			‘IN9:” Function attributed to the entry of the special cycles (output 9)	Refer to the “Configure input value” table at the end of the chapter	
379	017B			“USB:” USB mode (printer or supervision)	Supervision Printer Bar code Auto None	0000 1000 2000 3000 4000
412	019C			“SAVE ON” Mode of Results stocking.	None Internal USB	0000 1000 2000
413	019D			“ACCESS” Access parameters mode.	None USB Password	0000 1000 2000
414	019E			“YEAR” Year configuration.	2000 > 9999	
415	019F			“MONTH” Month configuration.	1 > 12	
416	01A0			“DAY” Day configuration.	1 > 31	
417	01A1			“HOUR” Hour configuration.	0 > 59	
418	01A2			“MINUTE” Minute configuration.	0 > 59	
419	01A3			“SECOND” Second configuration.	0 > 59	
459	01CB			“N. OF CYCLES” Number of learning cycle	2 > 9999	
460	01CC			“INTER-CYCLE” Time between 2 learning cycle	0 > 650 seconds	
461	01CD			“MAX OFFSET” Offset max for learning cycle	0 > 9999	
462	01CE			“FLOW MASTER” Value of Flow master for learning cycle	0 > 9999	
463	01CF			“PRESS MASTER” Value of Pressure master for learning cycle	-9999 > 9999	
464	01D0			“Min. Vol.” Minimum Volume for learning	0 > 9999	
465	01D1			“Max. Vol.” Maximum Volume for learning	0 > 9999	
486	01E6			“OFFSET” Offset Learning	-9999 > 9999	

Configurable input values

Input value	Value code
Program Selection	0000
Capil. Temp. Check (*)	10000
Temperature Check (*)	11000
Atm Pressure Check (*)	12000
P1 Sensor Check (*)	13000
Flow Check Cap 1(*)	14000
Flow Check Cap 2(*)	15000
Line P. Sensor Check (*)	16000
Regulator Adjust.	17000
Infinite Fill	18000
Piezo Az	19000
Code Reader	20000
Pre-Regul. Adjust.	21000
Print Results	22000
Volume Comp.	23000
Leak Offset Learn	24000
Offset+Vol. Learn	25000

(*) Available when the **Service special cycle** function is checked.

Unit table

This list gives all the units used in the instrument in hexadecimal code.

Unit code		Unit
Decimal	Hexadecimal	
0000	0000	cm^3/s
1000	03E8	cm^3/min
2000	07D0	cm^3/h
6000	1770	Pascal
11000	2AF8	Bar
12000	2EE0	Kilopascal
13000	32C8	PSI
14000	36B0	Millibar
15000	3A98	Megapascal
30000	7530	Liter/hour
46000	B3B0	Inch^3/s
47000	B798	Inch^3/min
48000	BB80	Inch^3/hour
49000	BF68	Feet^3/hour
50000	C350	Milliliter/second
51000	C738	Milliliter/minute
52000	CB20	Milliliter/hour
55000	D6D8	mm^3
56000	DAC0	cm^3
61000	EE48	Milliliter
62000	F230	Liter
63000	F618	inch^3
64000	FA00	feet^3
84000	01 4820	SCCM
92000	01 6760	Points

Reading of the parameters

> The parameters are dependents of the program number.
> Put the wanted program in edition before executing command.

- *Standard access*

This is an example based on the reading of three parameters:

— **Test type** (identifier number 21)
— **Fill time** (identifier number 1)
— **Stabilization time** (identifier number 2)

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td>
— Put in edition the program number on which the parameters have to be read
— Make a <b>Write N*words</b> request of 4 words at the <b>0000h</b> address, with the number of parameters to read (Word 1) and their identifiers (Word 2, 3 and 4).
On network:
<table>
<tr><td>01</td><td>10</td><td>00</td><td>00</td><td>00</td><td>04</td><td>08</td><td>03</td></tr>
<tr><td>00</td><td>15</td><td>00</td><td>01</td><td>00</td><td>02</td><td>00</td><td>F4</td></tr>
<tr><td>36</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>00 00</td><td>Word address (Read parameters)</td></tr>
<tr><td>00 04</td><td>Number of words to write</td></tr>
<tr><td>08</td><td>Number of bytes to write</td></tr>
<tr><td>03 00</td><td>Word 1: write 0003h (3 param. to read)</td></tr>
<tr><td>15 00</td><td>Word 2: write 0015h (identifier n°21)</td></tr>
<tr><td>01 00</td><td>Word 3: write 0001h (identifier n°1)</td></tr>
<tr><td>02 00</td><td>Word 4: write 0002h (identifier n°2)</td></tr>
<tr><td>F4 36</td><td>CRC</td></tr>
</table>
</td><td></td></tr>
<tr><td></td><td>
— Answer to the request:
On network:
<table>
<tr><td>01</td><td>10</td><td>00</td><td>00</td><td>00</td><td>04</td><td>C1</td><td>CA</td></tr>
</table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>00 00</td><td>Word address (Read parameters)</td></tr>
<tr><td>00 04</td><td>Number of written words</td></tr>
<tr><td>C1 CA</td><td>CRC</td></tr>
</table>
</td></tr>
<tr><td>
— Make a <b>Read N*words</b> request of 9 words at the <b>0000h</b> address, to retrieve the read parameters with their identifier on a word and their value on a long. (3 parameters * (1 + 2) words = 9 words)
On network:
<table>
<tr><td>01</td><td>03</td><td>00</td><td>00</td><td>00</td><td>09</td><td>85</td><td>CC</td></tr>
</table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>00 00</td><td>Word address (Read parameters)</td></tr>
<tr><td>00 09</td><td>Number of words to read</td></tr>
<tr><td>85 CC</td><td>CRC</td></tr>
</table>
</td><td></td></tr>
</table>

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td></td><td>
— Answer to the request:
On network:
<table>
<tr><td>01</td><td>03</td><td>12</td><td>15</td><td>00</td><td>E8</td><td>03</td><td>00</td></tr>
<tr><td>00</td><td>01</td><td>00</td><td>F4</td><td>01</td><td>00</td><td>00</td><td>02</td></tr>
<tr><td>00</td><td>E8</td><td>03</td><td>00</td><td>00</td><td>9B</td><td>C2</td><td></td></tr>
</table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>12</td><td>Number of read bytes</td></tr>
<tr><td>15 00</td><td>Word 1: read 0015h (identifier n°21)</td></tr>
<tr><td>E8 03</td><td>Word 2 &amp; 3: read 000003E8h</td></tr>
<tr><td>00 00</td><td>(value of test type = 1000(d) → Direct)</td></tr>
<tr><td>01 00</td><td>Word 4: read 0001h (identifier n°1)</td></tr>
<tr><td>F4 01</td><td>Word 5 &amp; 6: read 000001F4h</td></tr>
<tr><td>00 00</td><td>(value of fill time = 500(d) → 0.5 sec.)</td></tr>
<tr><td>02 00</td><td>Word 7: read 0002h (identifier n°2)</td></tr>
<tr><td>E8 03</td><td>Word 8 &amp; 9: read 000003E8h</td></tr>
<tr><td>00 00</td><td>(value of stab. time = 1000(d) → 1 sec)</td></tr>
<tr><td>9B C2</td><td>CRC</td></tr>
</table>
</td></tr>
</table>

- *Direct access*

In Direct access, the master can only access to parameters one by one.

This is an example based on the reading of three parameters:

— **Test type** (D.A. address: 2015h)

— **Fill time** (D.A. address: 2001h)

— **Stabilization time** (D.A. address: 2002h)

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td>
— Put in edition the program number on which the parameters have to be read
— Make a <b>Read N*words</b> request of 2 words at the <b>2015h</b> address.
On network:
<table>
<tr><td>01</td><td>03</td><td>20</td><td>15</td><td>00</td><td>02</td><td>DE</td><td>0F</td></tr>
</table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>20 15</td><td>D.A. address</td></tr>
<tr><td>00 02</td><td>Number of words to read</td></tr>
<tr><td>DE 0F</td><td>CRC</td></tr>
</table>
</td><td></td></tr>
<tr><td></td><td>
— Answer to the request:
On network:
<table>
<tr><td>01</td><td>03</td><td>04</td><td>E8</td><td>03</td><td>00</td><td>00</td><td>3F</td></tr>
<tr><td>93</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>04</td><td>Number of read bytes</td></tr>
<tr><td>E8 03</td><td>Word 1 &amp; 2: read 000003E8h</td></tr>
<tr><td>00 00</td><td>(value of test type = 1000(d) → Direct)</td></tr>
<tr><td>3F 93</td><td>CRC</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td>

— Make a **Read N*words** request of 2 words at the **2001h** address.

On network:

<table><tr><td>01</td><td>03</td><td>20</td><td>01</td><td>00</td><td>02</td><td>9E</td><td>0B</td></tr></table>

<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>20 01</td><td>D.A. address</td></tr>
<tr><td>00 02</td><td>Number of words to read</td></tr>
<tr><td>9E 0B</td><td>CRC</td></tr>
</table>

</td><td></td></tr>
<tr><td></td><td>

— Answer to the request:

On network:

<table><tr><td>01</td><td>03</td><td>04</td><td>F4</td><td>01</td><td>00</td><td>00</td><td>99</td></tr><tr><td>C3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>

<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>04</td><td>Number of read bytes</td></tr>
<tr><td>F4 01</td><td>Word 1 &amp; 2: read 000001F4h</td></tr>
<tr><td>00 00</td><td>(value of fill time = 500(d) → 0.5 sec.)</td></tr>
<tr><td>3F 93</td><td>CRC</td></tr>
</table>

</td></tr>
<tr><td>

— Make a **Read N*words** request of 2 words at the **2002h** address.

On network:

<table><tr><td>01</td><td>03</td><td>20</td><td>02</td><td>00</td><td>02</td><td>6E</td><td>0B</td></tr></table>

<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>20 02</td><td>D.A. address</td></tr>
<tr><td>00 02</td><td>Number of words to read</td></tr>
<tr><td>6E 0B</td><td>CRC</td></tr>
</table>

</td><td></td></tr>
<tr><td></td><td>

— Answer to the request:

On network:

<table><tr><td>01</td><td>03</td><td>04</td><td>E8</td><td>03</td><td>00</td><td>00</td><td>3F</td></tr><tr><td>93</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>

<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>04</td><td>Number of read bytes</td></tr>
<tr><td>E8 03</td><td>Word 1 &amp; 2: read 000003E8h</td></tr>
<tr><td>00 00</td><td>(value of stab. time = 1000(d) → 1 sec)</td></tr>
<tr><td>3F 93</td><td>CRC</td></tr>
</table>

</td></tr>
</table>

Writing of the parameters

> The parameters are dependents of the program number.
> Put the wanted program in edition before executing command.

- *Standard access*

This is an example based on the writing of two parameters:

— **Fill time** (identifier number 1)
— **Stabilization time** (identifier number 2)

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td>
— Put in edition the program number on which the parameters have to be written
— Make a <b>Write N*words</b> request of 7 words at the <b>0000h</b> address, with the number of parameters to read (Word 1), their identifiers (Word 2 and 5) and their value on a long (Word 3 & 4 and 6 & 7).
On network:
<table>
<tr><td>01</td><td>10</td><td>00</td><td>7F</td><td>00</td><td>07</td><td>0E</td><td>02</td></tr>
<tr><td>00</td><td>01</td><td>00</td><td>E8</td><td>03</td><td>00</td><td>00</td><td>02</td></tr>
<tr><td>00</td><td>E8</td><td>03</td><td>00</td><td>00</td><td>87</td><td>AC</td><td></td></tr>
</table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>00 7F</td><td>Word address (Write parameters)</td></tr>
<tr><td>00 07</td><td>Number of words to write</td></tr>
<tr><td>0E</td><td>Number of bytes to write</td></tr>
<tr><td>02 00</td><td>Word 1: write 0002h (2 param. to read)</td></tr>
<tr><td>01 00</td><td>Word 2: write 0001h (identifier n°1)</td></tr>
<tr><td>E8 03</td><td>Word 3 &amp; 4: write 000003E8h</td></tr>
<tr><td>00 00</td><td>(value of fill time = 1000(d) → 1 sec)</td></tr>
<tr><td>02 00</td><td>Word 5: write 0002h (identifier n°2)</td></tr>
<tr><td>E8 03</td><td>Word 6 &amp; 7: write 000003E8h</td></tr>
<tr><td>00 00</td><td>(value of stab. time = 1000(d) → 1 sec)</td></tr>
<tr><td>87 AC</td><td>CRC</td></tr>
</table>
</td><td></td></tr>
<tr><td></td><td>
— Answer to the request:
On network:
<table>
<tr><td>01</td><td>10</td><td>00</td><td>7F</td><td>00</td><td>07</td><td>B0</td><td>13</td></tr>
</table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>00 7F</td><td>Word address (Write parameters)</td></tr>
<tr><td>00 07</td><td>Number of written words</td></tr>
<tr><td>B0 13</td><td>CRC</td></tr>
</table>
</td></tr>
</table>

- *Direct access*

This is an example based on the writing of two parameters:

- **Fill time** (D.A. address: 6001h)
- **Stabilization time** (D.A. address: 6002h)

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td>

— Put in edition the program number on which the parameters have to be written

— Make a **Write N*words** request of 2 words at the **6001h** address.

On network:

<table>
<tr><td>01</td><td>10</td><td>60</td><td>01</td><td>00</td><td>02</td><td>04</td><td>F4</td></tr>
<tr><td>01</td><td>00</td><td>00</td><td>F9</td><td>91</td><td></td><td></td><td></td></tr>
</table>

<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>60 01</td><td>D.A. address</td></tr>
<tr><td>00 02</td><td>Number of words to write</td></tr>
<tr><td>04</td><td>Number of bytes to write</td></tr>
<tr><td>F4 01</td><td>Word 1 &amp; 2: read 000001F4h</td></tr>
<tr><td>00 00</td><td>(value of fill time = 500(d) → 0.5 sec)</td></tr>
<tr><td>F9 91</td><td>CRC</td></tr>
</table>

</td><td></td></tr>
<tr><td></td><td>

— Answer to the request:

On network:

<table>
<tr><td>01</td><td>10</td><td>60</td><td>01</td><td>00</td><td>02</td><td>0E</td><td>08</td></tr>
</table>

<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>60 01</td><td>D.A. address</td></tr>
<tr><td>00 02</td><td>Number of written words</td></tr>
<tr><td>0E 08</td><td>CRC</td></tr>
</table>

</td></tr>
<tr><td>

— Make a **Write N*words** request of 2 words at the **6002h** address.

On network:

<table>
<tr><td>01</td><td>10</td><td>60</td><td>02</td><td>00</td><td>02</td><td>04</td><td>F4</td></tr>
<tr><td>01</td><td>00</td><td>00</td><td>B9</td><td>84</td><td></td><td></td><td></td></tr>
</table>

<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>60 02</td><td>D.A. address</td></tr>
<tr><td>00 02</td><td>Number of words to write</td></tr>
<tr><td>04</td><td>Number of bytes to write</td></tr>
<tr><td>F4 01</td><td>Word 1 &amp; 2: read 000001F4h</td></tr>
<tr><td>00 00</td><td>(value of fill time = 500(d) → 0.5 sec)</td></tr>
<tr><td>B9 84</td><td>CRC</td></tr>
</table>

</td><td></td></tr>
<tr><td></td><td>

— Answer to the request:

On network:

<table>
<tr><td>01</td><td>10</td><td>60</td><td>02</td><td>00</td><td>02</td><td>FE</td><td>08</td></tr>
</table>

<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>60 02</td><td>D.A. address</td></tr>
<tr><td>00 02</td><td>Number of written words</td></tr>
<tr><td>FE 08</td><td>CRC</td></tr>
</table>

</td></tr>
</table>

Reading of the program name

This functionality is only available in Standard access.

The personalization is dependent of the program number.
Put the wanted program in edition before executing command.

Example of reading the personalization of a program named “PROGRAMME”:

Master

- Put in edition the program number on which the program name has to be read
- Make a **Read N*words** request of 6 words at the **0120h** address

On network:

01	03	01	20	00	06	C5	FE

01	Slave address
03	Function number (Read N*words)
01 20	Word address (Program name)
00 06	Number of words to read
C5 FE	CRC

Slave

- Answer to the request:

On network:

01	03	0C	50	52	4F	47	52
41	4D	4D	45	00	41	44	AF
43							

01	Slave address
03	Function number (Read N*words)
0C	Number of read bytes
50	ASCII code for ‘P’ character
52	ASCII code for ‘R’ character
4F	ASCII code for ‘O’ character
47	ASCII code for ‘G’ character
52	ASCII code for ‘R’ character
41	ASCII code for ‘A’ character
4D	ASCII code for ‘M’ character
4D	ASCII code for ‘M’ character
45	ASCII code for ‘E’ character
00	ASCII code for NULL character
41 44	The datas following ‘00’ until the last word (CRC) do not have any meaning.
AF 43	CRC

If your program name length is less than 12 characters, you will have a NULL character ‘00’ in the received string that mark the end of personalization. Every data following are meaningless, except for the last word of the frame, that still corresponds to the CRC.

Writing of the program name

This functionality is only available in Standard access.

The personalization is dependent of the program number.
Put the wanted program in edition before executing command.

Example of writing the personalization of a program as “PROG. FLOW”:

Master

- Put in edition the program number on which the program name has to be written
- Make a **Write N*words** request of 7 words at the **0120h** address

On network:

01	10	01	20	00	07	0E	50
52	4F	47	2E	20	46	4C	4F
57	00	00	00	00	75	F6	

01	Slave address
10	Function number (Write N*words)
01 20	Word address (Program name)
00 07	Number of words to write
0E	Number of bytes to write
50	ASCII code for ‘P’ character
52	ASCII code for ‘R’ character
4F	ASCII code for ‘O’ character
47	ASCII code for ‘G’ character
2E	ASCII code for ‘.’ character
20	ASCII code for space character
46	ASCII code for ‘F’ character
4C	ASCII code for ‘L’ character
4F	ASCII code for ‘O’ character
57	ASCII code for ‘W’ character
00	ASCII code for NULL character
00	ASCII code for NULL character
00	ASCII code for NULL character
00	ASCII code for NULL character
75 F6	CRC

Slave

- Answer to the request:

On network:

01	10	01	20	00	07	81	FD

01	Slave address
10	Function number (Write N*words)
01 20	Word address (Program name)
00 07	Number of written words
81 FD	CRC

The program name has a maximum 12 characters length (without the NULL characters).
Always end your program name with at least one NULL character (00h).

Standard command cycle

Program selection command on the ATEQ device

ℹ This functionality is only available in Standard access.

⚠ Always subtract 1 from the value of the program number to be selected.
Example: for selecting program number 2, send the value 1 at the address 0200h.

Example for selecting program number 3:

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td>— Make a <b>Write N*words</b> request of 1 word at the address <b>0200h</b>.
On network:
<table>
<tr><td>01</td><td>10</td><td>02</td><td>00</td><td>00</td><td>01</td><td>02</td><td>02</td></tr>
<tr><td>00</td><td>84</td><td>F0</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>02 00</td><td>Word address (Program name)</td></tr>
<tr><td>00 01</td><td>Number of words to write</td></tr>
<tr><td>02</td><td>Number of bytes to write</td></tr>
<tr><td>02 00</td><td>Word 1: write 0002h (program n°3)</td></tr>
<tr><td>81 FD</td><td>CRC</td></tr>
</table>
</td><td></td></tr>
<tr><td></td><td>— Answer to the request:
On network:
<table>
<tr><td>01</td><td>10</td><td>02</td><td>00</td><td>00</td><td>01</td><td>00</td><td>71</td></tr>
</table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>02 00</td><td>Word address (Program name)</td></tr>
<tr><td>00 01</td><td>Number of written words</td></tr>
<tr><td>00 71</td><td>CRC</td></tr>
</table>
</td></tr>
</table>

Start cycle command on the ATEQ device

This functionality is only available in Standard access.

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td>

— Select the program you want to start
— Make a **Write bit** request at the address **0001h** and force the Start bit to 1**.**
On network:

<table><tr><td>01</td><td>05</td><td>00</td><td>01</td><td>FF</td><td>00</td><td>DD</td><td>FA</td></tr></table>

<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>05</td><td>Function number (Write bit)</td></tr>
<tr><td>00 01</td><td>Bit address (Start bit)</td></tr>
<tr><td>FF 00</td><td>Force bit to 1</td></tr>
<tr><td>DD FA</td><td>CRC</td></tr>
</table>

</td><td></td></tr>
<tr><td></td><td>

— Answer to the request (exactly the same as the request):
On network:

<table><tr><td>01</td><td>05</td><td>00</td><td>01</td><td>FF</td><td>00</td><td>DD</td><td>FA</td></tr></table>

<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>05</td><td>Function number (Write bit)</td></tr>
<tr><td>00 01</td><td>Bit address (Start bit)</td></tr>
<tr><td>FF 00</td><td>Force bit to 1</td></tr>
<tr><td>DD FA</td><td>CRC</td></tr>
</table>

</td></tr>
</table>

Reset command on the ATEQ device

> ℹ This functionality is only available in Standard access.

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td>

— Make a **Write bit** request at the address **0000h** and force the Reset bit to 1**.**

On network:

<table><tr><td>01</td><td>05</td><td>00</td><td>00</td><td>FF</td><td>00</td><td>8C</td><td>3A</td></tr></table>

<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>05</td><td>Function number (Write bit)</td></tr>
<tr><td>00 00</td><td>Bit address (Reset bit)</td></tr>
<tr><td>FF 00</td><td>Force bit to 1</td></tr>
<tr><td>8C 3A</td><td>CRC</td></tr>
</table>

</td><td></td></tr>
<tr><td></td><td>

— Answer to the request (exactly the same as the request):

On network:

<table><tr><td>01</td><td>05</td><td>00</td><td>00</td><td>FF</td><td>00</td><td>8C</td><td>3A</td></tr></table>

<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>05</td><td>Function number (Write bit)</td></tr>
<tr><td>00 00</td><td>Bit address (Reset bit)</td></tr>
<tr><td>FF 00</td><td>Force bit to 1</td></tr>
<tr><td>8C 3A</td><td>CRC</td></tr>
</table>

</td></tr>
</table>

Special cycles

Special cycle table

Write the identifier number of the wanted special cycle at the address 0201h and its instruction if necessary.
Word 1 = identifier number of the special cycle
Word 2 = instruction for the special cycle

Numb	Special cycle
1	ATR learning Cycle.
4	Custom Unit Learn.
5	Custom Unit Check.
9	Piezo auto zero.
13	Regulator adjust.
25	Capil. Temp. Check (*).
26	Temperature Check (*).
27	Atm Pressure Check (*).
28	P1 Sensor Check (*).
29	Flow 1 Check (*).
30	Flow 2 Check (*).
31	Line P. Sensor check (*).

(*) Appears with the **Service special cycle** function checked.

Auto-zero on the ATEQ device

This functionality is only available in Standard access.

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td>

— Select the program on which you want to make the auto zero

— Make a **Write N*words** request of 1 word at the address **0201h** and pass the value of the auto zero special cycle (n°9).

On network:

<table>
<tr><td>01</td><td>10</td><td>02</td><td>01</td><td>00</td><td>01</td><td>02</td><td>09</td></tr>
<tr><td>00</td><td>82</td><td>11</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>02 01</td><td>Word address (Special cycle)</td></tr>
<tr><td>00 01</td><td>Number of words to write</td></tr>
<tr><td>02</td><td>Number of bytes to write</td></tr>
<tr><td>09 00</td><td>Word 1: write 0009h (spe. cycle n°9)</td></tr>
<tr><td>82 11</td><td>CRC</td></tr>
</table>

</td><td></td></tr>
<tr><td></td><td>

— Answer to the request:

On network:

<table>
<tr><td>01</td><td>10</td><td>02</td><td>01</td><td>00</td><td>01</td><td>51</td><td>B1</td></tr>
</table>

<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>10</td><td>Function number (Write N*words)</td></tr>
<tr><td>02 01</td><td>Word address (Special cycle)</td></tr>
<tr><td>00 01</td><td>Number of written words</td></tr>
<tr><td>51 B1</td><td>CRC</td></tr>
</table>

</td></tr>
<tr><td>— Make a start request to launch the special cycle.</td><td></td></tr>
</table>

FIFO results

FIFO list results structure

At the end of each cycle, a result is stored as an array of 12 words contained in a FIFO of 8 results. This result includes the final state of the instrument (relays position, alarm signal, indicators state…), but also of the test (units, values measured for pressure and flow).
The results are in the memory of the instrument. To obtain them, it is necessary to carry out a "Read FIFO results" request.

Words	Meaning	Type	Bytes	Coeff
1	Program number.	Word	2	
2	Test type.	Word	2	
3	Image of the relays: Bit 0 = 1: pass part. Bit 1 = 1: fail part, maximum flow reject. Bit 2 = 1: fail part, minimum flow reject. Bit 3 = 1: alarm. Bit 4 = 1: unused. Bit 5 = 1: reserved. Bit 6 = 1: unused. Bit 7 = 1: unused.	Word	2	
4	Alarm code (refer to the alarm codes table).	Word	2	
5	Pressure low part word.	Long	4	x1000
6	Pressure high part word.			
7	Pressure unit code low part word (refer to units table).	Long	4	x1000
8	Pressure unit code high part word (refer to units table).			
9	Flow low section word.	Long	4	x1000
10	Flow high section word.			
11	Flow unit code low part word (refer to. Units table).	Long	4	x1000
12	Flow unit code high part word (refer to. Units table).			

All the numerical values are treated with **Long** format with fixed comma (10^{-3}). Thus, they must be multiplied by 1000 to get the value in units (see examples in "Basic notions" section).

Step table

This table represents the codes of the steps in the cycle.

Code		Steps
Decimal	Hexadecimal	
0	0000	Pre-fill.
1	0001	Fill
2	0002	Zero Diff.
3	0003	Stabilization
4	0004	Test
5	0005	Dump
65535	FFFF	No step in progress

Alarm codes table

This list gives all the alarms in hexadecimal code.

Identifier n°		Alarm
Decimal	Hexadecimal	
0	0000	No alarm.
1	0001	Pressure switched alarm (test pressure too high).
2	0002	Pressure switch (test pressure too small).
3	0003	Large leak on TEST (EEEE).
4	0004	Large leak on REF (MMMM).
7	0007	Sensor out of order (overrun).
43	002B	Pressure too high.
44	002C	Pressure too low.
45	002D	Piezo sensor out of order.
46	002E	Dump error.
47	002F	Calibration drift.
73	0049	Atmospheric pressure error.
74	004A	Temperature error.

Cycle results reading (last 8 results in FIFO)

This functionality is only available in Standard access.

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td>— Make a <b>Read N*words</b> request of 12 words at the <b>0010h</b> address.
On network:
<table><tr><td>01</td><td>03</td><td>00</td><td>10</td><td>00</td><td>0C</td><td>44</td><td>0A</td></tr></table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>00 10</td><td>Word address (FIFO result)</td></tr>
<tr><td>00 0C</td><td>Number of words to read</td></tr>
<tr><td>44 0A</td><td>CRC</td></tr>
</table></td><td></td></tr>
<tr><td></td><td>— Answer to the request:
On network:
<table><tr><td>01</td><td>03</td><td>50</td><td><i>FIFO result structure on 12 words</i></td><td><i>CRC</i></td></tr></table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>50</td><td>Number of read bytes</td></tr>
<tr><td></td><td>FIFO result structure on 12 words</td></tr>
<tr><td></td><td>CRC</td></tr>
</table></td></tr>
</table>

Reset FIFO results

This functionality is only available in Standard access.

This command resets the 8 last cycle's results available in the FIFO.

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr>
<td>
— Make a <b>Write bit</b> request at the address <b>0002h</b> and force the Reset FIFO bit to 1<b>.</b>
On network:
<table><tr><td>01</td><td>05</td><td>00</td><td>02</td><td>FF</td><td>00</td><td>2D</td><td>FA</td></tr></table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>05</td><td>Function number (Write bit)</td></tr>
<tr><td>00 00</td><td>Bit address (Reset FIFO bit)</td></tr>
<tr><td>FF 00</td><td>Force bit to 1</td></tr>
<tr><td>2D FA</td><td>CRC</td></tr>
</table>
</td>
<td></td>
</tr>
<tr>
<td></td>
<td>
— Answer to the request (exactly the same as the request):
On network:
<table><tr><td>01</td><td>05</td><td>00</td><td>02</td><td>FF</td><td>00</td><td>2D</td><td>FA</td></tr></table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>05</td><td>Function number (Write bit)</td></tr>
<tr><td>00 00</td><td>Bit address (Reset FIFO bit)</td></tr>
<tr><td>FF 00</td><td>Force bit to 1</td></tr>
<tr><td>2D FA</td><td>CRC</td></tr>
</table>
</td>
</tr>
</table>

Last results

Last results structure

> Reminder: **Direct access addresses** are expressed in **hexadecimal**

At the end of each cycle, the last result is as an array of 40 words. This result includes the final state of the instrument (relays position, alarm signal, indicators state...), but also of the test (units, values measured for the pressure and the flow).
The last result is in the memory of the instrument. To obtain them, it is necessary to carry out a "Read last results" request.

Words	D.A. address Read	Meaning	Type	Bytes	Coeff
1	2301	Program number.	Word	2	
2	2302	Test type.	Word	2	
3	2303	Image of the relays: Bit 0 = 1: pass part. Bit 1 = 1: fail part, maximum flow reject. Bit 2 = 1: fail part, minimum flow reject. Bit 3 = 1: alarm. Bit 4 = 1: unused. Bit 5 = 1: reserved. Bit 6 = 1: unused. Bit 7 = 1: unused.	Word	2	
4	2304	Alarm code (refer to the alarm codes table).	Word	2	
5	2305	Pressure low part word.	Long	4	x1000
6	2306	Pressure high part word.			
7	2307	Press. unit code low part word (refer to units table).	Long	4	x1000
8	2308	Press. unit code high part word (refer to units table).			
9	2309	Flow low section word.	Long	4	x1000
10	230A	Flow high section word.			
11	230B	Flow unit code low part word (refer to. Units table).	Long	4	x1000
12	230C	Flow unit code high part word (refer to. Units table).			

> All the numerical values are treated with **Long** format with fixed comma (10^{-3}). Thus, they must be multiplied by 1000 to get the value in units (see examples in "Basic notions" section).

Last results reading

> For using this function, it is important to:
> — Having done a start on the instrument before (“End of cycle” bit on in the relay status)
> — Not having done a reset of the FIFO

- *Standard access*

Example of reading the entire last result structure:

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td>— Make a <b>Read N*words</b> request of 12 words at the <b>0011h</b> address.
On network:
<table><tr><td>01</td><td>03</td><td>00</td><td>11</td><td>00</td><td>0C</td><td>15</td><td>CA</td></tr></table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>00 11</td><td>Word address (Last result)</td></tr>
<tr><td>00 0C</td><td>Number of words to read</td></tr>
<tr><td>15 CA</td><td>CRC</td></tr>
</table></td><td></td></tr>
<tr><td></td><td>— Answer to the request:
On network:
<table><tr><td>01</td><td>03</td><td>50</td><td><i>Last result structure on 12 words</i></td><td><i>CRC</i></td></tr></table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>50</td><td>Number of read bytes</td></tr>
<tr><td></td><td>Last result structure on 12 words</td></tr>
<tr><td></td><td>CRC</td></tr>
</table></td></tr>
</table>

- *Direct access*

In Direct access, the master can only access to parameters one by one.

This is an example on the reading of the pressure unit code in the last result:

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td>— Make a <b>Read N*words</b> request of 2 words at the <b>2307h</b> address.
On network:
<table><tr><td>01</td><td>03</td><td>23</td><td>07</td><td>00</td><td>02</td><td>7E</td><td>4E</td></tr></table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>23 07</td><td>D.A. address</td></tr>
<tr><td>00 02</td><td>Number of words to read</td></tr>
<tr><td>7E 4E</td><td>CRC</td></tr>
</table></td><td></td></tr>
<tr><td></td><td>— Answer to the request:
On network:
<table><tr><td>01</td><td>03</td><td>04</td><td>F8</td><td>2A</td><td>00</td><td>00</td><td>EA</td></tr><tr><td>9B</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Write N*words)</td></tr>
<tr><td>04</td><td>Number of read bytes</td></tr>
<tr><td>F8 2A</td><td>Word 1 &amp; 2: read 00002AF8h</td></tr>
<tr><td>00 00</td><td>(Pressure unit code = 11000 → bar)</td></tr>
<tr><td>EA 9B</td><td>CRC</td></tr>
</table></td></tr>
</table>

Real time

Status and real time measures structure

> Reminder: **Direct access addresses** are expressed in **hexadecimal**

The real time measurement is used for display curve or values during the cycle and not for the final measurement.

> Do not take or use the final results in this section, it is just to see the status of the device for the "Cycle end" (bit 5) and "Key presence" (bit 15) information.
> For the results, use only the FIFO list results structure or the Last results structure (see above)

Words	D.A. address Read	Meaning	Type	Bytes	Coeff
1	2201	Program number.	Word	2	
2	2202	Number of results waiting in the results FIFO memory.	Word	2	
3	2203	Test type.	Word	2	
4	2204	Status: Bit 0 = 1: pass part. Bit 1 = 1: fail part maximum flow. Bit 2 = 1: fail part minimum flow. Bit 3 = 1: alarm. Bit 4 = 1: pressure error.	Do not use these results while the Bit 5 (cycle end is not 1). Use only Bit 5 (cycle end) and Bit 15 (key presence).		
		Bit 5 = 1: cycle end.	Word	2	
		Bit 6 = 1: recoverable part. Bit 7 = 1: CAL error or drift. Bit 8 = 1: *Unused* Bit 9 = 1: ATR error or drift. Bits 10 / 11 / 12 / 13 / 14 = 1: *Unused*. Bit 15 = 1: key presence.	Do not use these results while the Bit 5 (cycle end is not 1). Use only Bit 5 (cycle end) and Bit 15 (key presence).		
5	2205	Step code (refer to steps table).	Word	2	
6	2206	Low pressure section word.	Long	4	x1000
7	2207	High pressure section word.			
8	2208	Pressure unit code low part word (see units table).	Long	4	x1000
9	2209	Pressure unit code high part word (see units table).			
10	220A	Flow low section word.	Long	4	x1000
11	220B	Flow high section word.			
12	220C	Flow unit code low part word (refer to. Units table).	Long	4	x1000
13	220D	Flow unit code high part word (refer to. Units table).			

Status and real time measures reading

> For using this function, it is important to:
> — Having done a start on the instrument before (“End of cycle” bit on in the relay status)
> — Not having done a reset of the FIFO

- *Standard access*

Example of reading the entire status and real time measures structure:

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td>
— Make a <b>Read N*words</b> request of 13 words at the <b>0010h</b> address.
On network:
<table>
<tr><td>01</td><td>03</td><td>00</td><td>30</td><td>00</td><td>0D</td><td>84</td><td>00</td></tr>
</table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>00 30</td><td>Word address (Real time result)</td></tr>
<tr><td>00 0D</td><td>Number of words to read</td></tr>
<tr><td>84 00</td><td>CRC</td></tr>
</table>
</td><td></td></tr>
<tr><td></td><td>
— Answer to the request:
On network:
<table>
<tr><td>01</td><td>03</td><td>1A</td><td>02</td><td>00</td><td>00</td><td>00</td><td>01</td></tr>
<tr><td>00</td><td>21</td><td>80</td><td>FF</td><td>FF</td><td>00</td><td>00</td><td>00</td></tr>
<tr><td>00</td><td>F8</td><td>2A</td><td>00</td><td>00</td><td>08</td><td>CF</td><td>00</td></tr>
<tr><td>00</td><td>70</td><td>17</td><td>00</td><td>00</td><td>AE</td><td>95</td><td></td></tr>
</table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>1A</td><td>Number of read bytes</td></tr>
<tr><td>02 00</td><td>Word 1: read 0002h (prog. N°3)</td></tr>
<tr><td>00 00</td><td>Word 2: read 0000h (num. of res. in FIFO)</td></tr>
<tr><td>01 00</td><td>Word 3: read 0001h (type test = leak)</td></tr>
<tr><td>21 80</td><td>Word 4: read 8021h (status)</td></tr>
<tr><td>FF FF</td><td>Word 5: read FFFFh (step code)</td></tr>
<tr><td>00 00
00 00</td><td>Word 6 &amp; 7: read 00000000h
(pressure value = 0)</td></tr>
<tr><td>F8 2A
00 00</td><td>Word 8 &amp; 9: read 00002AF8h
(pressure unit = 11000 → bar)</td></tr>
<tr><td>08 CF
00 00</td><td>Word 10 & 11: read 000008CFh
(leak value = 53000 → 53)</td></tr>
<tr><td>70 17
00 00</td><td>Word 12 & 13: read 00001770h
(leak unit = 6000 → Pascal)</td></tr>
<tr><td>AE 95</td><td>CRC</td></tr>
</table>
</td></tr>
</table>

- *Direct access*

In Direct access, the master can only access to parameters one by one.

This is an example on the reading of the end of cycle bit in the status:

<table>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td>— Make a <b>Read N*words</b> request of 1 word at the <b>2204h</b> address.
On network:
<table><tr><td>01</td><td>03</td><td>22</td><td>04</td><td>00</td><td>01</td><td>CF</td><td>B3</td></tr></table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Read N*words)</td></tr>
<tr><td>22 04</td><td>D.A. address</td></tr>
<tr><td>00 01</td><td>Number of words to read</td></tr>
<tr><td>CF B3</td><td>CRC</td></tr>
</table></td><td></td></tr>
<tr><td></td><td>— Answer to the request:
On network:
<table><tr><td>01</td><td>03</td><td>02</td><td>21</td><td>80</td><td>A1</td><td>B4</td></tr></table>
<table>
<tr><td>01</td><td>Slave address</td></tr>
<tr><td>03</td><td>Function number (Write N*words)</td></tr>
<tr><td>02</td><td>Number of read bytes</td></tr>
<tr><td>21 80</td><td>Word 1: read 8021h
(cycle end = 8021 & 0020 = 1)</td></tr>
<tr><td>A1 B4</td><td>CRC</td></tr>
</table></td></tr>
</table>