



# F600 Series – CC-Link Manual





# Table of Contents

## Safety advisory / Warranty

Good practices and safety instructions .....	3
--	---

## Preamble

Introduction .....	4
Basic notions .....	5

## Hardware installation

Hardware configuration .....	7
Cabling .....	8

## Configuration of the ATEQ device (slave)

Setup of the CC-Link configuration mode .....	9
Setup of the station address .....	11
Setup of the communication speed .....	12
Setup of the output mode .....	13

## Configuration of the master

Configure the MELSEC board as master .....	14
Describe the slaves to the master .....	16
Download the configuration to the master .....	18
Verification .....	19




## Functional description of an ATEQ device

Introduction .....	20
Write and read table by mode .....	21
Configuration .....	50
Cycle .....	79
Results .....	84
SLMP Parameters .....	94





## 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 information 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

3 / 95

### 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  $\pm$  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 F600 device on the CC-Link network.

**i** | 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.

5 / 95

### 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 Byte*  
0 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 Byte Byte Byte*  
0 1 2 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 Byte Byte Byte*  
0 1 2 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:

$$- 207.055 = 207055 \div 1000$$

Example – Pressure:

On network: 

E3	28	03	00
Byte	Byte	Byte	Byte
0	1	2	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	Byte	Byte	Byte
0	1	2	3

- Word 1 = FF94h

- Word 2 = FFFFh

- Value = FFFFFFFF94h = - 108(d) = - 108 of thousandths of unit

- Real value = - 108 ÷ 1000 = - 0.108 expressed in units



# Hardware installation

Connect your ATEQ equipment to the CC-Link network using its CC-Link connectors and compatible cables.

## HARDWARE CONFIGURATION

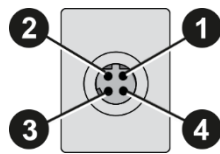
Your device has a CC-Link internal board and two CC-Link connectors.  
The CC-Link internal board is located inside your device.

Your device has two M12 D coded type connector – 4 pins female connector.



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

### M12 D coded type connector – 4 pins female connector



Pin number	Signal
1	DA (brown)
2	DB (white)
3	DG (blue)
4	SLD (black)

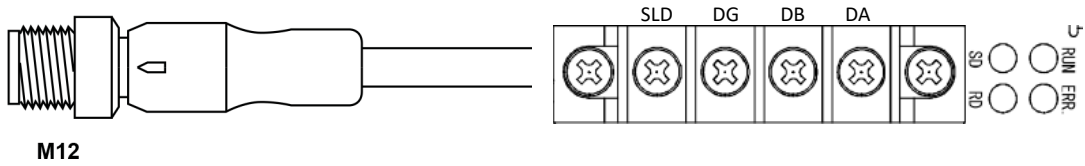




## CABLING

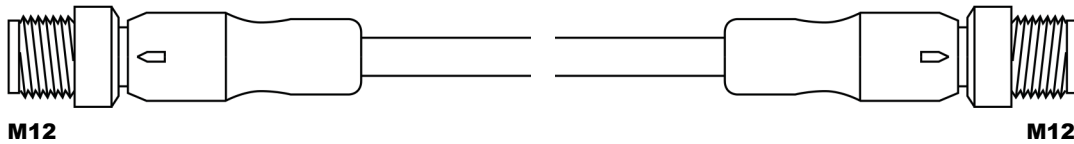
### From slave to master MELSEC board

Cable with male M12 D coded type connector (4pins) and an MELSEC connector (4pins).

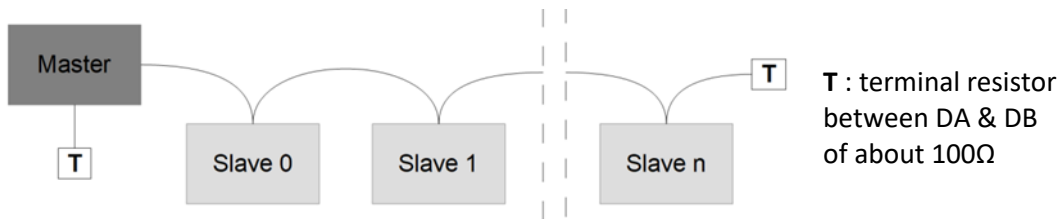


### Between two slaves

Cable with two male M12 D coded type connector (4pins).



### CC-Link network link





# Configuration of the ATEQ device (slave)

Use this procedure to set the configuration mode of your device.



This configuration can be done with the front panel of your ATEQ device or with the ATEQ Fieldbus Configurator software

## SETUP OF THE CC-LINK CONFIGURATION MODE

Five configuration modes are available:

Mode	Configuration mode	Number of occupied station	Extended cycle	Length (in bytes)	Use
5	<b>Standard mode</b> <i>(6<sup>th</sup> Serie only)</i>	4	Octuple	368	For the inputs/outputs, real time measurements, the live cycle results, 21 parameters management, functions and extended menu bits and extended cycle results
4	<b>Standard less mode</b>	3	Octuple	272	For the inputs/outputs, real time measurements, the live cycle results, 21 parameters management, functions and extended menu bits
3	<b>Medium more mode</b>	2	Octuple	176	For the inputs/outputs, the real time measurements, the live cycle results, 10 parameters management, functions and extended menu bits
2	<b>Medium mode</b>	1	Octuple	80	For the inputs/outputs, the real time measurements and cycle results
1	<b>Light mode</b>	1	Quadruple	40	For the inputs/outputs and the real time measurements

Number of bytes available for the communication depends on the selected configuration mode. Therefore, the usable functionalities also depend on it:

WRITE					
Mode	Commands	Extended menu bits	Function bits	Parameters	Program name
5	ALL	YES	YES	21	YES
4	ALL	YES	YES	21	YES
3	ALL	YES	YES	10	YES
2	Start, Reset, Auto-zero, Select Prog, FIFO & Last results	NO	NO	NO	NO
1	Start, Reset, Auto-zero, Select Prog	NO	NO	NO	NO





READ								
Mode	Echo / Error	Relay status	Real time measurements	Results	Extended menu bits	Function bits	Parameters	Program name
5	ALL	YES	YES	Extended	YES	YES	21	YES
4	ALL	YES	YES	Standard	YES	YES	21	YES
3	ALL	YES	YES	Standard	YES	YES	10	YES
2	Start, Reset, Auto-zero, Select Prog, FIFO & Last results	YES	YES	Standard	NO	NO	NO	NO
1	Start, Reset, Auto-zero, Select Prog	YES	YES	NO	NO	NO	NO	NO

```

MAIN /CONFI/FIELDBUS
>MODE : 1-Lite
NUM OCC : 1
EXT.CYC : Quadruple
  
```

5<sup>th</sup> Series:  
 From the MAIN MENU screen of your ATEQ device go to

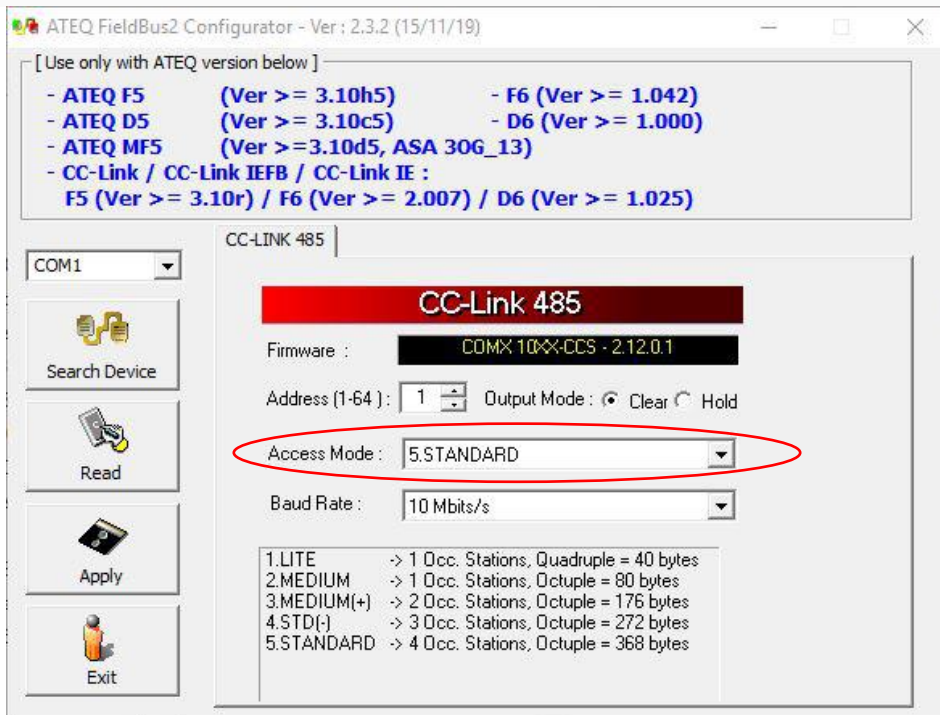
- > CONFIGURATION
- > FIELDBUS
- > MODE

```

MAIN /CONFI/AUTOM/FI
ADDRESS : 001
▶ACCESS : 1.LITE
Speed : 10 Mbit/s
MODE : Clear output
* 1 Occ. Stations
* 4 - Quadruple
* 40 bytes
  
```

6<sup>th</sup> Series:  
 From the MAIN MENU screen of your ATEQ device go to

- > CONFIGURATION
- > AUTOMATISM
- > FIELDBUS
- > ACCESS





## SETUP OF THE STATION ADDRESS

Station address goes from **1 to 64**.



The station address configured in the ATEQ edvice must be the same as the one provided to the master.

```
MAIN /CONF/FIELD BUS
>ADDRESS: 01
Speed : 10 Mbit/s
HOLD M.: Clear outp
```

### 5<sup>th</sup> Series:

From the **MAIN MENU** screen of your ATEQ device go to

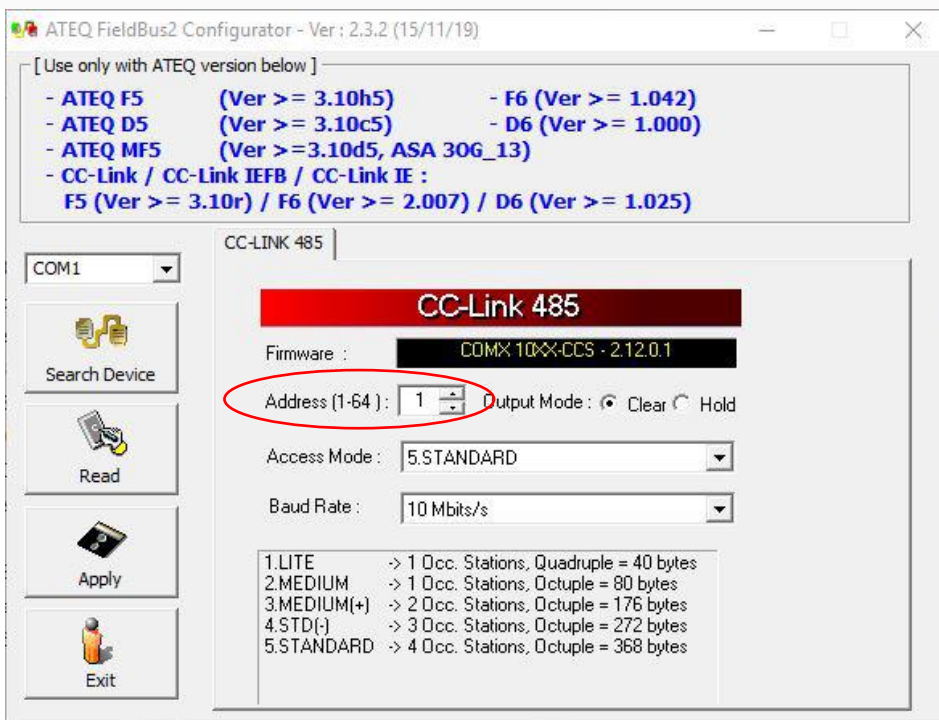
- CONFIGURATION
- FIELD BUS
- ADDRESS

```
MAIN /CONF/AUTOM/FI
ADDRESS : 001
ACCESS : 5.STANDARD
Speed : 10 Mbit/s
MODE : Clear output
* 4 Occ. Stations
* 8 - Octuple
* 368 bytes
```

### 6<sup>th</sup> Series:

From the **MAIN MENU** screen of your ATEQ device go to

- CONFIGURATION
- AUTOMATISM
- FIELD BUS
- ADDRESS





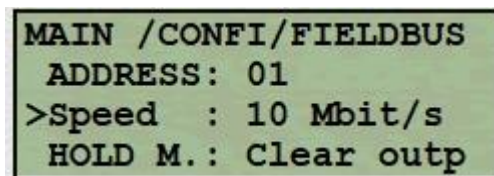
## SETUP OF THE COMMUNICATION SPEED

The communication speed can be set to:

- 156 kbits/s
- 625 kbits/s
- 2.5 Mbits/s
- 5 Mbits/s
- 10 Mbits/s



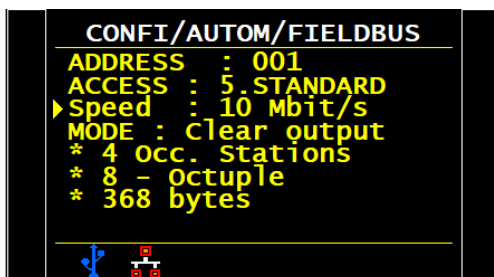
The communication speed configured in the ATEQ device must be the same as the one provided to the master.



### 5<sup>th</sup> Series:

From the **MAIN MENU** screen of your ATEQ device go to

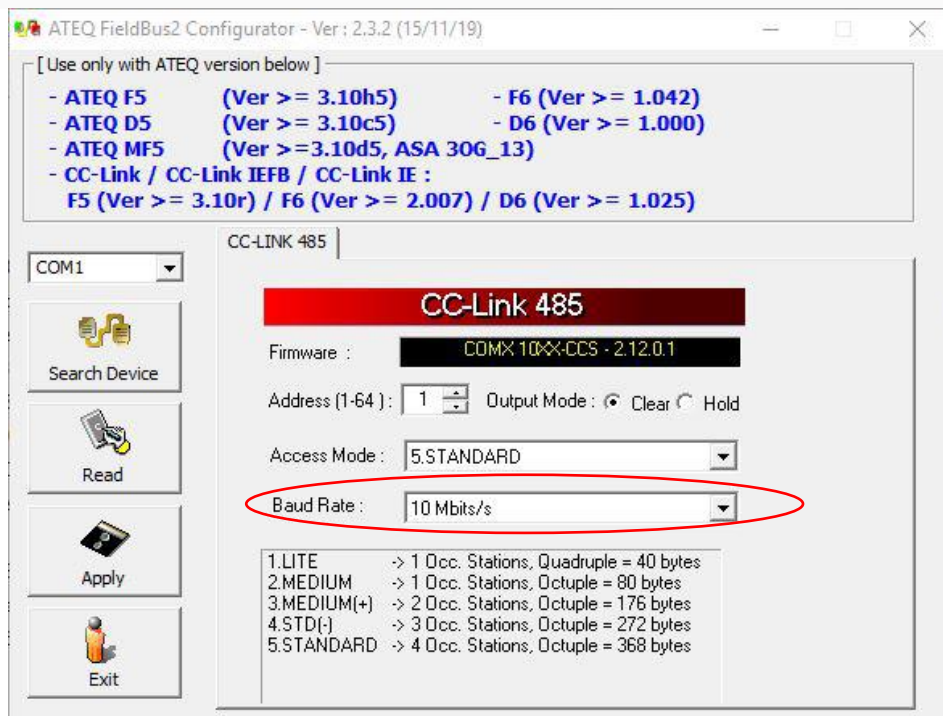
- **CONFIGURATION**
- **FIELD BUS**
- **Speed**



### 6<sup>th</sup> Series:

From the **MAIN MENU** screen of your ATEQ device go to

- **CONFIGURATION**
- **AUTOMATISM**
- **FIELD BUS**
- **Speed**





## SETUP OF THE OUTPUT MODE

The output mode is used to command the output to hold their value or to clear it after being set.

The output modes are:

- Clear output
- Hold output

13 / 95



The output mode configured in the ATEQ device must be the same as the one provided to the master.

```
MAIN /CONFI/FIELDBUS
ADDRESS: 01
Speed : 10 Mbit/s
>HOLD M.: Clear outp
```

### 5<sup>th</sup> Series:

From the **MAIN MENU** screen of your ATEQ device go to

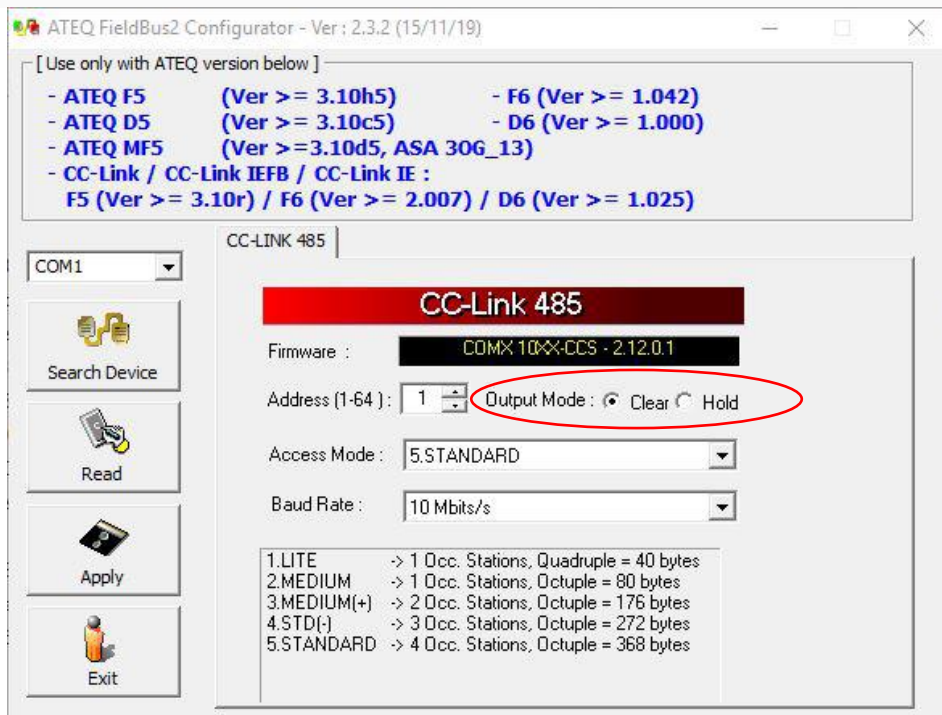
- > CONFIGURATION
- > FIELDBUS
- > HOLD M.

```
CONFI/AUTOM/FIELDBUS
ADDRESS : 001
ACCESS : 5.STANDARD
Speed : 10 Mbit/s
>MODE : Clear output
* 4 Occ. Stations
* 8 - Octuple
* 368 bytes
```

### 6<sup>th</sup> Series:

From the **MAIN MENU** screen of your ATEQ device go to

- > CONFIGURATION
- > AUTOMATISM
- > FIELDBUS
- > MODE



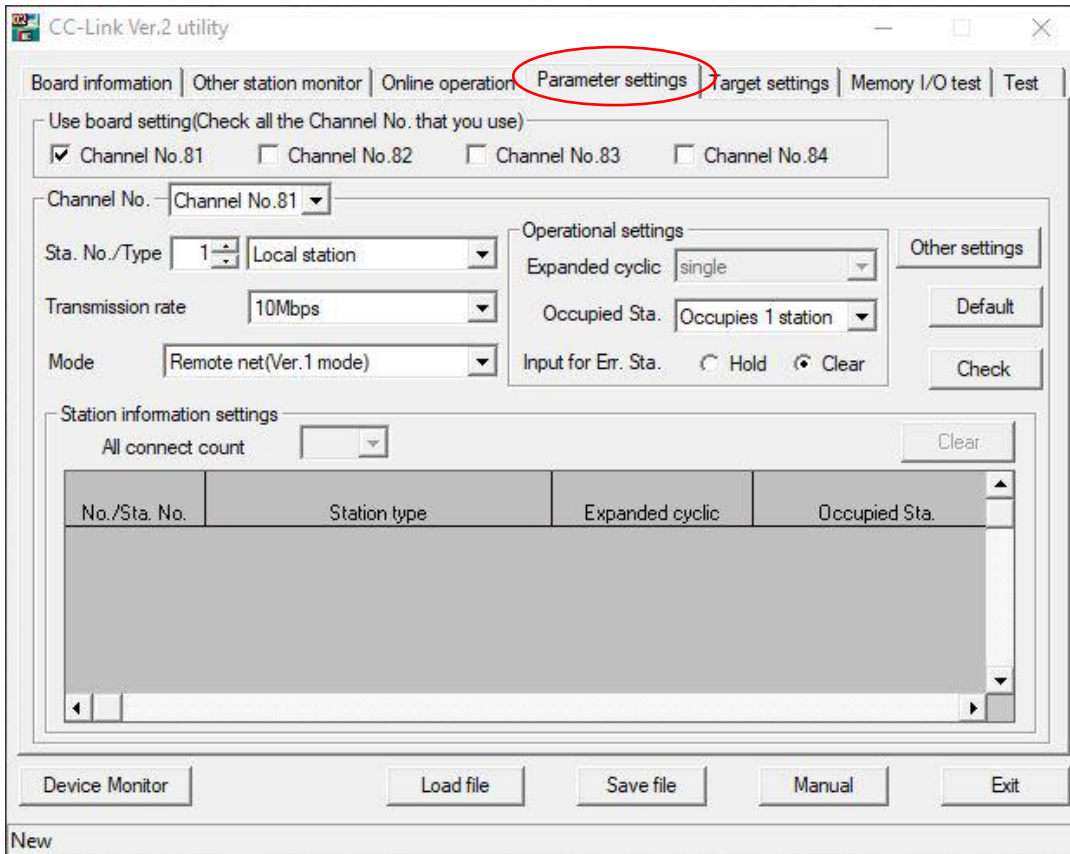


# Configuration of the master

**i** For the master configuration, you will use the “MELSEC CC-Link Ver.2 utility” software and a MELSEC master board put in your computer.

## CONFIGURE THE MELSEC BOARD AS MASTER

Open **MELSEC CC-Link Ver.2 utility** software and go into the **Parameter setting** tab





Put the **Sta. No** to 0 in order to inform the MELSEC board it will be the master (and not a slave).  
Make the **Transmission rate** correspond to the baud rate of your slaves.  
Put the **Mode** to "Remote net(Ver.2 mode)".  
Make the **Input for Err. Sta.** correspond to the output mode of your slaves.

The screenshot shows the 'CC-Link Ver.2 utility' software interface. The 'Parameter settings' tab is active. The 'Use board setting' section has checkboxes for Channel No. 81, 82, 83, and 84, with Channel No. 81 selected. The 'Channel No.' dropdown is set to 'Channel No. 81'. The 'Operational settings' section includes: 'Sta. No./Type' set to '0' (Master station), 'Transmission rate' set to '10Mbps', 'Mode' set to 'Remote net(Ver.2 mode)', 'Expanded cyclic' set to 'single', and 'Occupied Sta.' set to 'Occupies 1 station'. The 'Input for Err. Sta.' section has radio buttons for 'Hold' and 'Clear', with 'Clear' selected. The 'Station information settings' section shows 'All connect count' set to '64'. Below this is a table with 6 rows of station information.

No./Sta. No.	Station type	Expanded cyclic	Occupied Sta.
1/ 1	Ver.1 Remote I/O station	single	Occupies 1 station
2/ 2	Ver.1 Remote I/O station	single	Occupies 1 station
3/ 3	Ver.1 Remote I/O station	single	Occupies 1 station
4/ 4	Ver.1 Remote I/O station	single	Occupies 1 station
5/ 5	Ver.1 Remote I/O station	single	Occupies 1 station
6/ 6	Ver.1 Remote I/O station	single	Occupies 1 station





## DESCRIBE THE SLAVES TO THE MASTER

Indicate the number of slaves connected on your CC-Link network in **All connect count**.

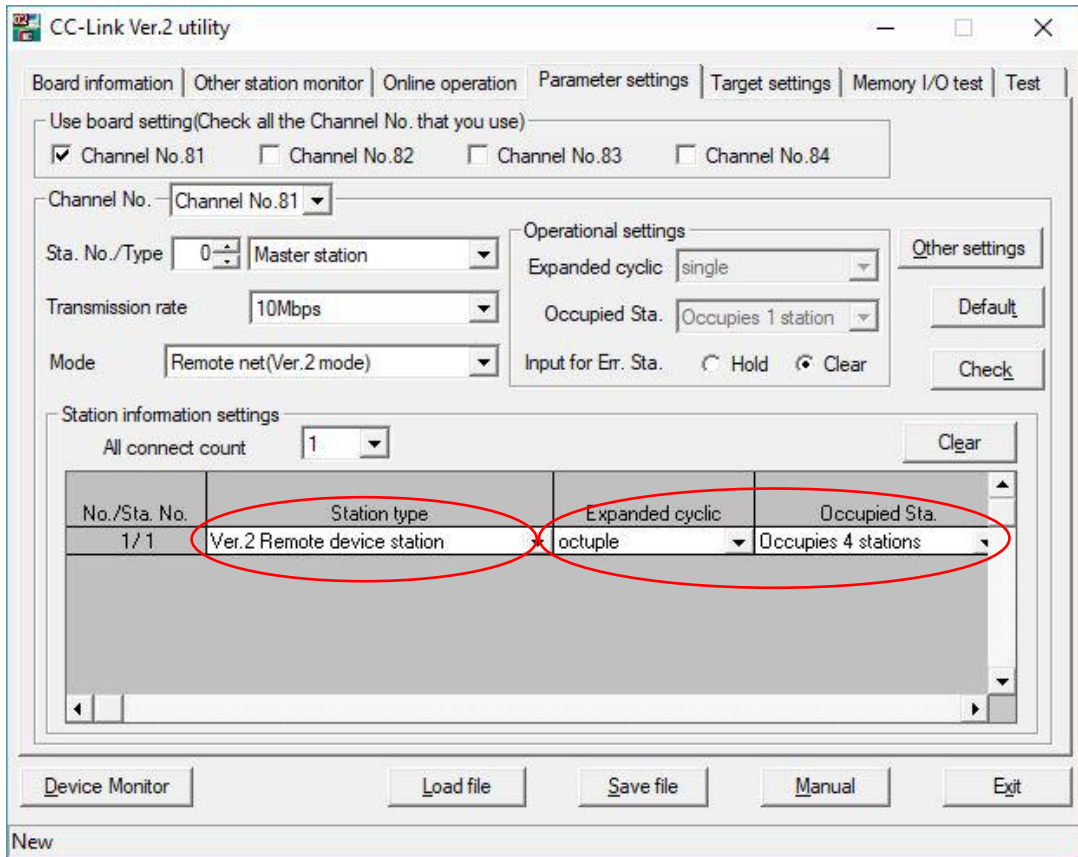
The screenshot shows the 'CC-Link Ver.2 utility' window. The 'Station information settings' section is highlighted with a red circle, showing 'All connect count' set to 1. Below this is a table with the following data:

No./Sta. No.	Station type	Expanded cyclic	Occupied Sta.
1/1	Ver.1 Remote I/O station	single	Occupies 1 station

At the bottom of the window, there are buttons for 'Device Monitor', 'Load file', 'Save file', 'Manual', and 'Exit'. The status bar at the very bottom shows the word 'New'.



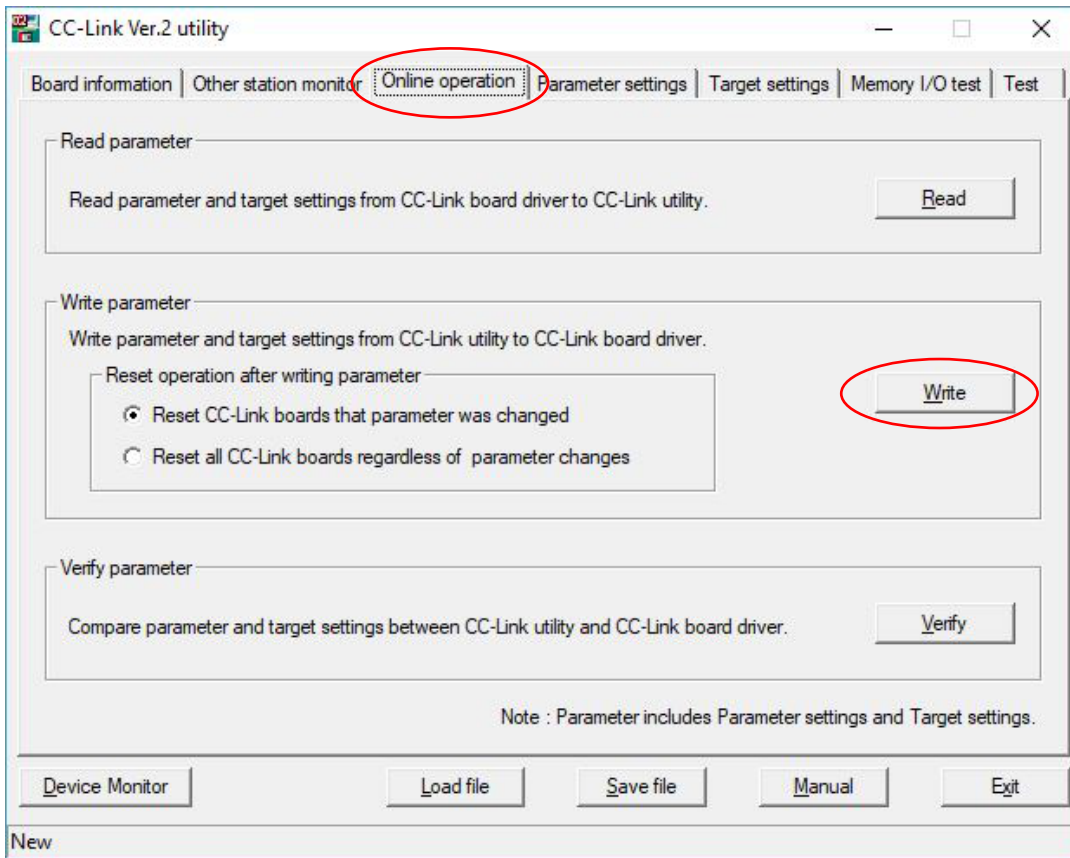
Put the **Station type** of your slaves to “Ver.2 Remote device station”.  
Make the **Expanded cyclic** and the number of **Occupied Sta.** correspond to the configuration mode of your slaves.





## DOWNLOAD THE CONFIGURATION TO THE MASTER

Go into **Online operation** tab and click on **Write** in order to write all the configuration onto the board.



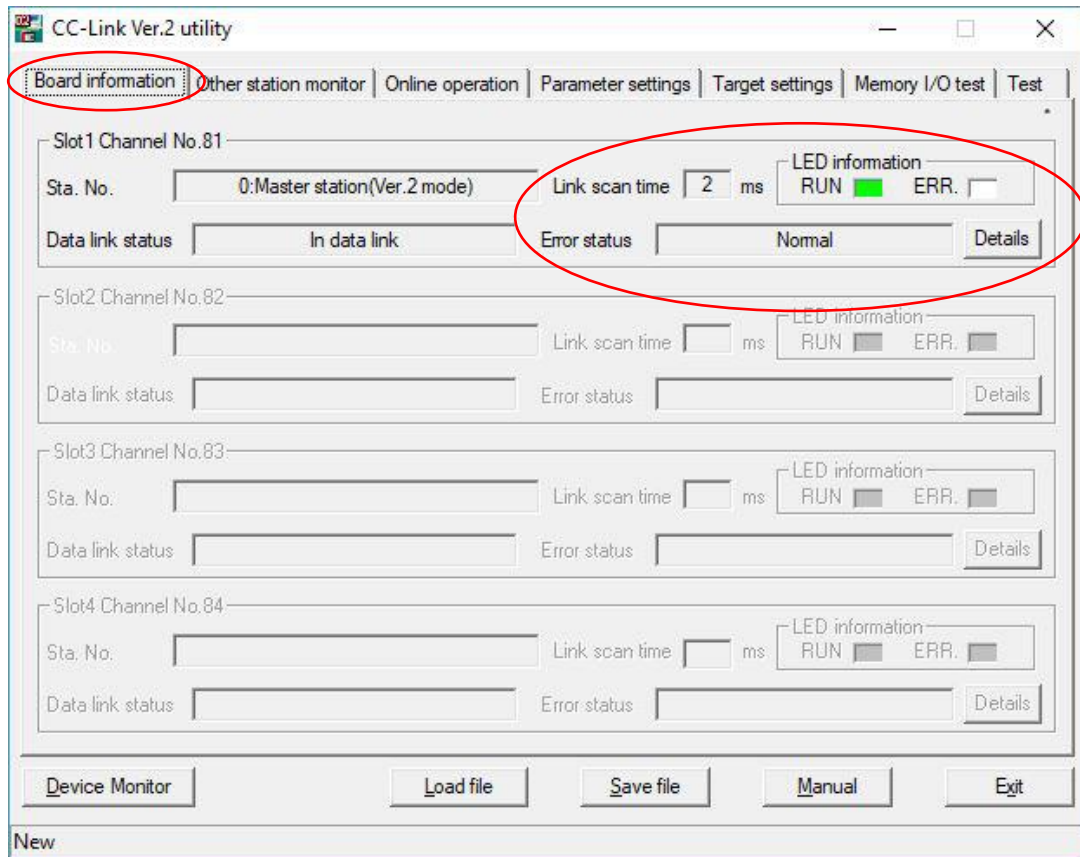


## VERIFICATION

Go into the **Board information** tab.

If all your slaves are connected to the master and if the configuration is good, the **RUN led** will be on and the **Error status** will be on “Normal”.

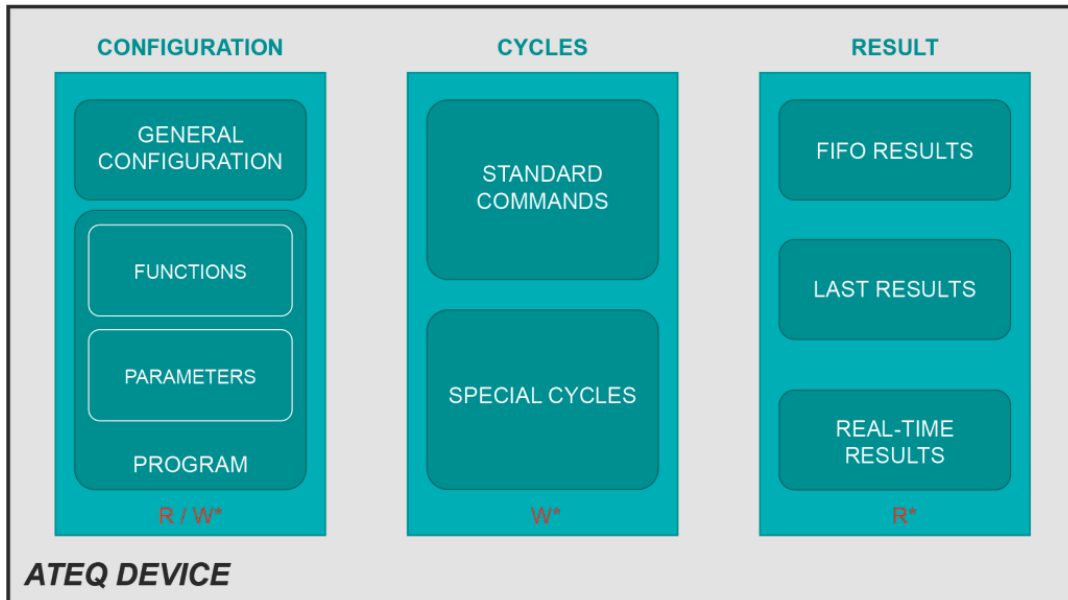
Otherwise the **ERR. led** will be on and the **Error status** will indicate “Error” (you can have details on this error by clicking on the button **Details**)





# Functional description of an ATEQ device

## INTRODUCTION



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



## WRITE AND READ TABLE BY MODE

### Write table

#### Command data

21 / 95

MASTER → ATEQ				
Offset (byte decimal)	Offset (word hexa)	Command / Status	Description	Type
0	00(h)	Commands	Start / Reset / Read Result / Read or write parameters	WORD
2	01(h)	RESERVED		
4	02(h)	RESERVED		
6	03(h)	Program number to select	Program to run / edit	WORD
8	04(h)	Special cycle code to run	Auto-zero	WORD
10	05(h)	RESERVED		
...	...			
32	0F(h)			

#### Exchange data (depends on mode)

MASTER → ATEQ				
Mode	Offset (byte decimal)	Offset (word hexa)	Command / Status	Type
1	-	-	-	-
2	-	-	-	-
3	32	10(h)	General bits functions – 8 words	ARRAY
	48	18(h)	Normal bits functions – 10 words	ARRAY
	68	22(h)	10 parameters	STRUCTURE
	132	42(h)	Program name	ARRAY
4	40	14(h)	General bits functions – 10 words	ARRAY
	60	1E(h)	Normal bits functions – 10 words	ARRAY
	80	28(h)	21 parameters	STRUCTURE
	208	68(h)	Program name	ARRAY
5	72	24(h)	General bits functions – 10 words	ARRAY
	92	2E(h)	Normal bits functions – 10 words	ARRAY
	112	38(h)	21 parameters	STRUCTURE
	240	78(h)	Program name	ARRAY



## Read table

### Data available cyclically

MASTER ← ATEQ				
Offset (byte decimal)	Offset (word hexa)	Command echo, Status/Relay/Step	Description	Type
0	00(h)	Command echo	Echo of the current command	WORD
2	01(h)	Command error code	Error code of the current command	WORD
4	02(h)	Refresh counter		WORD
6	03(h)	Number of program	Current running program	WORD
8	04(h)	Number of FIFO results	Result available in FIFO	WORD
10	05(h)	Test type	Current running test type	WORD
12	06(h)	Relays status	Current relay status	WORD
14	07(h)	Step	Current cycle step	WORD
Real time measurements				
16	08(h)	Pressure value (32 bits integer)		LONG
20	0A(h)	Pressure unit (32 bits integer)		LONG
24	0C(h)	Leak value (32 bits integer)		LONG
28	0E(h)	Leak unit (32 bits integer)		LONG

### Exchange data (depends on mode)

MASTER ← ATEQ				
Mode	Offset (byte decimal)	Offset (word hexa)	Command / Status	Type
1	-	-	-	-
2	32	10(h)	FIFO / Last results	STRUCTURE
3	32	10(h)	General bits functions – 8 words	ARRAY
	48	18(h)	Normal bits functions – 10 words	ARRAY
	68	22(h)	10 parameters	STRUCTURE
	132	42(h)	Program name	ARRAY
	144	48(h)	FIFO / Last results	STRUCTURE
4	40	14(h)	General bits functions – 10 words	ARRAY
	60	1E(h)	Normal bits functions – 10 words	ARRAY
	80	28(h)	21 parameters	STRUCTURE
	208	68(h)	Program name	ARRAY
	224	70(h)	FIFO / Last results	STRUCTURE
5	72	24(h)	General bits functions – 10 words	ARRAY
	92	2E(h)	Normal bits functions – 10 words	ARRAY
	112	38(h)	21 parameters	STRUCTURE
	240	78(h)	Program name	ARRAY
	336	80(h)	FIFO / Last results (extended)	STRUCTURE



## Details mode by mode

### Mode 1.LIGHT: 1 occupied station, Quadruple (40 byte)

For the digital input/output and the real time measurements.

- RX/RX000-03F(h) = 4 words
- WRr/WRw00-0F(h) = 16 words

23 / 95

READ							
Echo / Error	Relay status	Real time measurements	Results	Extended menu bits	Function bits	Parameters	Program name
Start, Reset, Autozero, Select Prog	YES	YES	NO	NO	NO	NO	NO

WRITE				
Commands	Extended menu bits	Function bits	Parameters	Program name
Start, Reset, Autozero, Select Prog	NO	NO	NO	NO

```

MAIN /CONFI/FIELDBUS
MODE : 1-Lite
NUM OCC : 1
>EXT.CYC : Quadruple
  
```

```

MAIN /CONFI/FIELDBUS
NUM OCC : 1
EXT.CYC : Quadruple
>I/O size: 40 bytes
  
```

```

MAIN /CONFI/AUTOM/FI
ADDRESS : 001
▶ACCESS : 1.LITE
Speed : 10 Mbit/s
MODE : Clear output
* 1 Occ. Stations
* 4 - Quadruple
* 40 bytes
  
```

ATEQ FieldBus2 Configurator - Ver : 2.3.2 (15/11/19)

[ Use only with ATEQ version below ]

- ATEQ F5 (Ver >= 3.10h5) - F6 (Ver >= 1.042)
- ATEQ D5 (Ver >= 3.10c5) - D6 (Ver >= 1.000)
- ATEQ MF5 (Ver >= 3.10d5, ASA 30G\_13)
- CC-Link / CC-Link IEFB / CC-Link IE : F5 (Ver >= 3.10r) / F6 (Ver >= 2.007) / D6 (Ver >= 1.025)

COM1

CC-LINK 485

**CC-Link 485**

Firmware : COMX 10XX-CCS - 2.12.0.1

Address (1-64) : 1 Output Mode :  Clear  Hold

Access Mode : 1.LITE

Baud Rate : 10 Mbits/s

1.LITE -> 1 Occ. Stations, Quadruple = 40 bytes  
 2.MEDIUM -> 1 Occ. Stations, Octuple = 80 bytes  
 3.MEDIUM(+) -> 2 Occ. Stations, Octuple = 176 bytes  
 4.STD(-) -> 3 Occ. Stations, Octuple = 272 bytes  
 5.STANDARD -> 4 Occ. Stations, Octuple = 368 bytes





RX/RX = 4 words – Commands						
Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)		Write table (MASTER → ATEQ)	
0	2	00(h)	<b>Command echo:</b>		<b>Command:</b>	
			X000(h)	Echo reset	Y000(h)	Reset (stop current cycle)
			X001(h)	Echo start	Y001(h)	Start (starting a test cycle)
			X002(h)	Echo start special cycle	Y002(h)	Start special cycle
			X003(h)	Echo program selection	Y003(h)	Program selection
			X004(h)	<i>RESERVED</i>	Y004(h)	<i>RESERVED</i>
			X00F(h)		Y00F(h)	
2	2	01(h)	<b>Command error code:</b>		Y010(h)	<i>RESERVED</i>
			X010(h)	Error reset		
			X011(h)	Error start		
			X012(h)	Error start special cycle		
			X013(h)	Error program selection		
			X014(h)	<i>RESERVED</i>		
X01F(h)	Y01F(h)					
4	2	02(h)	X020(h)	Refresh counter	Y020(h)	<i>RESERVED</i>
			...		...	
			X02F(h)		Y02F(h)	
6	2	03(h)	X030(h)	Number of current program	Y030(h)	Number of program to be selected
			...		...	
			X03F(h)		Y03F(h)	



WRr/WRw = 16 words – Status & Real time measurements						
Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)		Write table (MASTER → ATEQ)	
8	2	04(h)	WRr00(h)	Number of FIFO results	WRw00(h)	Special cycle code
10	2	05(h)	WRr01(h)	Test type	WRw01(h)	RESERVED
12	2	06(h)	WRr02(h)	<b>Relay status:</b>	WRw02(h)	RESERVED
				Bit 0 = 1 > Pass part (OK)		
				Bit 1 = 1 > Fail test part (NOK)		
				Bit 2 = 1 > Fail reference part (NOK)		
				Bit 3 = 1 > Alarm		
				Bit 4 = 1 > Pressure error		
				Bit 5 = 1 > Cycle end (system ready)		
				Bit 6 = 1 > Part recoverable		
				Bit 7 = 1 > Calibration error		
				Bit 8 = 1 > Calibration check error		
				Bit 9 = 1 > ATR fault		
				Bit 10 to 15 > <i>not used, always at 0</i>		
14	2	07(h)	WRr03(h)	Step	WRw03(h)	RESERVED
Real time measurements						
16	4	08(h)	<b>Pressure value (32 bits integer)</b> Example: Pressure value = 207 On network : 98 28 03 00 00032898(H) → 207000(d)/1000 → 207			
			WRr04(h)	Low word (LSW)	WRw04(h)	RESERVED
			WRr05(h)	High word (MSW)	WRw05(h)	RESERVED
20	4	0A(h)	<b>Pressure unit (32 bits integer)</b> Example: Pressure unit Pa = 6 On network: 70 17 00 00 00001770(H) → 6000(d)/1000 → 6			
			WRr06(h)	Low word (LSW)	WRw06(h)	RESERVED
			WRr07(h)	High word (MSW)	WRw07(h)	RESERVED
24	4	0C(h)	<b>Leak value (32 bits integer)</b> Example: Leak value = -0.108 On network: 94 FF FF FF FFFFFF94(H) → -108(d)/1000 → -0.108			
			WRr08(h)	Low word (LSW)	WRw08(h)	RESERVED
			WRr09(h)	High word (MSW)	WRw09(h)	RESERVED
28	4	0E(h)	<b>Leak unit (32 bits integer)</b> Example: Leak Unit Pa/s = 8 On network: 40 1F 00 00 00001F40(H) → 8000(d)/1000 → 8			
			WRr0A(h)	Low word (LSW)	WRw0A(h)	RESERVED
			WRr0B(h)	High word (MSW)	WRw0B(h)	RESERVED
32	6	10(h)	WRr0C(h)		WRw0C(h)	
			...	RESERVED	...	RESERVED
			WRr0F(h)		WRw0F(h)	





## Mode 2.MEDIUM: 1 occupied station, Octuple (80 bytes)

For the input/output, the real time measurements and cycle results

- RX/RX000-07F(h) = 8 words
- WRr/WRw00-1F(h) = 32 words

READ							
Echo / Error	Relay status	Real time measurements	Results	Extended menu bits	Function bits	Parameters	Program name
Start, Reset, Autozero, Select Prog, FIFO & Last results	YES	YES	Standard	NO	NO	NO	NO

WRITE				
Commands	Extended menu bits	Function bits	Parameters	Program name
Start, Reset, Autozero, Select Prog, FIFO & Last results	NO	NO	NO	NO

```

MAIN /CONFI/FIELDBUS
>MODE : 2-Medium
NUM OCC : 1
EXT.CYC : Octuble
  
```

```

MAIN /CONFI/FIELDBUS
NUM OCC : 1
EXT.CYC : Octuble
>I/O size: 80 bytes
  
```

```

MAIN /CONFI/AUTOM/FI
ADDRESS : 001
▶ACCESS : 2.MEDIUM
Speed : 10 Mbit/s
MODE : Clear output
* 1 Occ. Stations
* 8 - Octuple
* 80 bytes
  
```

ATEQ FieldBus2 Configurator - Ver: 2.3.2 (15/11/19)

[ Use only with ATEQ version below ]

- ATEQ F5 (Ver >= 3.10h5) - F6 (Ver >= 1.042)
- ATEQ D5 (Ver >= 3.10c5) - D6 (Ver >= 1.000)
- ATEQ MF5 (Ver >= 3.10d5, ASA 30G\_13)
- CC-Link / CC-Link IEFB / CC-Link IE : F5 (Ver >= 3.10r) / F6 (Ver >= 2.007) / D6 (Ver >= 1.025)

COM1

Search Device

Read

Apply

Exit

CC-LINK 485

**CC-Link 485**

Firmware : COMX 10XX-CCS - 2.12.0.1

Address (1-64) : 1 Output Mode :  Clear  Hold

Access Mode : 2.MEDIUM

Baud Rate : 10 Mbits/s

1.LITE -> 1 Occ. Stations, Quadruple = 40 bytes  
 2.MEDIUM -> 1 Occ. Stations, Octuple = 80 bytes  
 3.MEDIUM(+) -> 2 Occ. Stations, Octuple = 176 bytes  
 4.STD(-) -> 3 Occ. Stations, Octuple = 272 bytes  
 5.STANDARD -> 4 Occ. Stations, Octuple = 368 bytes





RX/RX = 8 words – Commands & Status						
Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)		Write table (MASTER → ATEQ)	
0	2	00(h)	<b>Command echo:</b>		<b>Command:</b>	
			X000(h)	Echo reset	Y000(h)	Reset (stop current cycle)
			X001(h)	Echo start	Y001(h)	Start (starting a test cycle)
			X002(h)	Echo start special cycle	Y002(h)	Start special cycle
			X003(h)	Echo program selection	Y003(h)	Program selection
			X004(h)	Echo read FIFO results	Y004(h)	Read FIFO results
			X005(h)	RESERVED	Y005(h)	RESERVED
			X006(h)	RESERVED	Y006(h)	RESERVED
			X007(h)	Echo reset FIFO results	Y007(h)	Reset FIFO results
			X008(h)	RESERVED	Y008(h)	RESERVED
			...			
			X00E(h)		Y00E(h)	
X00F(h)	Echo read Last results	Y00F(h)	Read Last results			
2	2	01(h)	<b>Command error code:</b>			
			X010(h)	Error reset	Y010(h)	RESERVED
			X011(h)	Error start		
			X012(h)	Error start special cycle		
			X013(h)	Error program selection		
			X014(h)	Error read FIFO results		
			X015(h)	RESERVED		
			X016(h)	RESERVED		
			X017(h)	Error reset FIFO results		
			X018(h)	RESERVED	...	RESERVED
			...			
			X01E(h)			
X01F(h)	Error read Last results	Y01F(h)				
4	2	02(h)	X020(h)	Refresh counter	Y020(h)	RESERVED
			...		...	
			X02F(h)		Y02F(h)	
6	2	03(h)	X030(h)	Number of current program	Y030(h)	Number of program to be selected
			...		...	
			X03F(h)		Y03F(h)	
8	2	04(h)	X040(h)	Number of FIFO results	Y040(h)	Special cycle code
			...		...	
			X04F(h)		Y04F(h)	
10	2	05(h)	X050(h)	Test type	Y050(h)	RESERVED
			...		...	
			X05F(h)		Y05F(h)	





Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)	Write table (MASTER → ATEQ)	
12	2	06(h)	<b>Relay status:</b>		
			X060(h) Pass part (OK)	Y060(h)	RESERVED
			X061(h) Fail test part (NOK)		
			X062(h) Fail reference part (NOK)		
			X063(h) Alarm		
			X064(h) Pressure error		
			X065(h) Cycle end (system ready)		
			X066(h) Part recoverable	...	
			X067(h) Calibration error		
			X068(h) Calibration check error		
			X069(h) ATR fault		
			X06A(h) ... <i>Not used, always at 0</i>		
			X06F(h)	Y06F(h)	
14	2	07(h)	X070(h)	Y070(h)	
			...	...	
			X07F(h)	Y07F(h)	



WRr/WRw = 32 words – Real time measurements & FIFO / Last results						
Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)		Write table (MASTER → ATEQ)	
16	4	08(h)	<b>Pressure value (32 bits integer)</b> Example: Pressure value = 207 On network : 98 28 03 00 00032898(H) → 207000(d)/1000 → 207			
			WRr00(h)	Low word (LSW)	WRw00(h)	RESERVED
			WRr01(h)	High word (MSW)	WRw01(h)	RESERVED
20	4	0A(h)	<b>Pressure unit (32 bits integer)</b> Example: Pressure unit Pa = 6 On network: 70 17 00 00 00001770(H) → 6000(d)/1000 → 6			
			WRr02(h)	Low word (LSW)	WRw02(h)	RESERVED
			WRr03(h)	High word (MSW)	WRw03(h)	RESERVED
24	4	0C(h)	<b>Leak value (32 bits integer)</b> Example: Leak value = -0.108 On network: 94 FF FF FF FFFFFF94(H) → -108(d)/1000 → -0.108			
			WRr04(h)	Low word (LSW)	WRw04(h)	RESERVED
			WRr05(h)	High word (MSW)	WRw05(h)	RESERVED
28	4	0E(h)	<b>Leak unit (32 bits integer)</b> Example: Leak Unit Pa/s = 8 On network: 40 1F 00 00 00001F40(H) → 8000(d)/1000 → 8			
			WRr06(h)	Low word (LSW)	WRw06(h)	RESERVED
			WRr07(h)	High word (MSW)	WRw07(h)	RESERVED
FIFO / Last results						
32	36	10(h)	WRr08(h)	FIFO / Last results	WRw08(h)	RESERVED
			...		...	
68	6	22(h)	WRr19(h)	RESERVED	WRw19(h)	RESERVED
			WRr1A(h)		WRw1A(h)	
			...		...	
			WRr1F(h)		WRw1F(h)	



### Mode 3.MEDIUM(+): 2 occupied stations, Octuple (176 bytes)

For the input/output, the real time measurements, 10 parameters managements, functions and extended menu bits and cycle result.

- RX/Ry000-17F(h) = 24 words
- WRr/WRw00-3F(h) = 64 words

READ							
Echo / Error	Relay status	Real time measurements	Results	Extended menu bits	Function bits	Parameters	Program name
ALL	YES	YES	Standard	YES (8 words)	YES (10 words)	YES (10 parameters)	YES

WRITE				
Commands	Extended menu bits	Function bits	Parameters	Program name
ALL	YES (8 words)	YES (10 words)	YES (10 parameters)	YES

```

MAIN /CONFI/FIELDBUS
MODE : 3-Medium(+)
NUM OCC : 2
>EXT.CYC : Octuble

MAIN /CONFI/FIELDBUS
NUM OCC : 2
EXT.CYC : Octuble
>I/O size: 176 bytes
  
```

```

MAIN /CONFI/AUTOM/FI
ADDRESS : 001
>ACCESS : 3.MEDIUM(+)
Speed : 10 Mbit/s
MODE : Clear output
* 2 Occ. Stations
* 8 - octuple
* 176 bytes
  
```

ATEQ FieldBus2 Configurator - Ver: 2.3.2 (15/11/19)

[ Use only with ATEQ version below ]

- ATEQ F5 (Ver >= 3.10h5) - F6 (Ver >= 1.042)
- ATEQ D5 (Ver >= 3.10c5) - D6 (Ver >= 1.000)
- ATEQ MF5 (Ver >= 3.10d5, ASA 30G\_13)
- CC-Link / CC-Link IEFB / CC-Link IE : F5 (Ver >= 3.10r) / F6 (Ver >= 2.007) / D6 (Ver >= 1.025)

COM1

CC-LINK 485

**CC-Link 485**

Firmware : COMX10XX-CCS - 2.12.0.1

Address (1-64) : 1 Output Mode :  Clear  Hold

Access Mode : 3.MEDIUM(+)

Baud Rate : 10 Mbits/s

1.LITE -> 1 Occ. Stations, Quadruple = 40 bytes  
 2.MEDIUM -> 1 Occ. Stations, Octuple = 80 bytes  
 3.MEDIUM(+) -> 2 Occ. Stations, Octuple = 176 bytes  
 4.STD(-) -> 3 Occ. Stations, Octuple = 272 bytes  
 5.STANDARD -> 4 Occ. Stations, Octuple = 368 bytes





RX/RW = 24 words – Commands, Status, Real time measurements & Extended menu bits handling						
Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)		Write table (MASTER → ATEQ)	
0	2	00(h)	<b>Command echo:</b>		<b>Command:</b>	
			X000(h)	Echo reset	Y000(h)	Reset (stop current cycle)
			X001(h)	Echo start	Y001(h)	Start (starting a test cycle)
			X002(h)	Echo start special cycle	Y002(h)	Start special cycle
			X003(h)	Echo program selection	Y003(h)	Program selection
			X004(h)	Echo read FIFO results	Y004(h)	Read FIFO results
			X005(h)	Echo read parameters	Y005(h)	Read parameters
			X006(h)	Echo write parameters	Y006(h)	Write parameters
			X007(h)	Echo reset FIFO results	Y007(h)	Reset FIFO results
			X008(h)	RESERVED	Y008(h)	RESERVED
			X009(h)	Echo read ext. menu bits	Y009(h)	Read extended menu bits
			X00A(h)	Echo read functions bits	Y00A(h)	Read functions bits
			X00B(h)	Echo write ext. menu bits	Y00B(h)	Write extended menu bits
			X00C(h)	Echo write functions bits	Y00C(h)	Write functions bits
			X00D(h)	Echo read program name	Y00D(h)	Read program name
			X00E(h)	Echo write program name	Y00E(h)	Write program name
			X00F(h)	Echo read Last results	Y00F(h)	Read Last results
2	2	01(h)	<b>Command error code:</b>		Y010(h)	RESERVED
			X010(h)	Error reset		
			X011(h)	Error start		
			X012(h)	Error start special cycle		
			X013(h)	Error program selection		
			X014(h)	Error read FIFO results		
			X015(h)	Error read parameters		
			X016(h)	Error write parameters		
			X017(h)	Error reset FIFO results		
			X018(h)	RESERVED		
			X019(h)	Error read ext. menu bits		
			X01A(h)	Error read functions bits		
			X01B(h)	Error write ext. menu bits		
			X01C(h)	Error write functions bits		
			X01D(h)	Error read program name		
			X01E(h)	Error write program name		
			X01F(h)	Error read Last results		
4	2	02(h)	X020(h)	Refresh counter	Y020(h)	RESERVED
			... X02F(h)		... Y02F(h)	
6	2	03(h)	X030(h)	Number of current program	Y030(h)	Number of program to be selected
			... X03F(h)		... Y03F(h)	
8	2	04(h)	X040(h)	Number of FIFO results	Y040(h)	Special cycle code
			... X04F(h)		... Y04F(h)	
10	2	05(h)	X050(h)	Test type	Y050(h)	RESERVED
			... X05F(h)		... Y05F(h)	







Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)	Write table (MASTER → ATEQ)
12	2	06(h)	<b>Relay status:</b>	
			X060(h) Pass part (OK)	Y060(h)
			X061(h) Fail test part (NOK)	
			X062(h) Fail reference part (NOK)	
			X063(h) Alarm	
			X064(h) Pressure error	
			X065(h) Cycle end (system ready)	
			X066(h) Part recoverable	...
			X067(h) Calibration error	
			X068(h) Calibration check error	
			X069(h) ATR fault	
			X06A(h) ... <i>Not used, always at 0</i>	
			X06F(h)	Y06F(h)
14	2	07(h)	X070(h) ... Step	Y070(h) ... RESERVED
			X07F(h)	Y07F(h)



Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)	Write table (MASTER → ATEQ)		
<b>Real time measurements</b>						
16	4	08(h)	<b>Pressure value (32 bits integer)</b> Example: Pressure value = 207 On network : 98 28 03 00 00032898(H) → 207000(d)/1000 → 207			
			X080(h)	Low word (LSW)	Y080(h)	RESERVED
			...		...	
			X08F(h)	Y08F(h)		
X090(h)	High word (MSW)	Y090(h)	RESERVED			
...		...				
X09F(h)	Y09F(h)					
20	4	0A(h)	<b>Pressure unit (32 bits integer)</b> Example: Pressure unit Pa = 6 On network: 70 17 00 00 00001770(H) → 6000(d)/1000 → 6			
			X0A0(h)	Low word (LSW)	Y0A0(h)	RESERVED
			...		...	
			X0AF(h)	Y0AF(h)		
X0B0(h)	High word (MSW)	Y0B0(h)	RESERVED			
...		...				
X0BF(h)	Y0BF(h)					
24	4	0C(h)	<b>Leak value (32 bits integer)</b> Example: Leak value = -0.108 On network: 94 FF FF FF FFFFFFF94(H) → -108(d)/1000 → -0.108			
			X0C0(h)	Low word (LSW)	Y0C0(h)	RESERVED
			...		...	
			X0CF(h)	Y0CF(h)		
X0D0(h)	High word (MSW)	Y0D0(h)	RESERVED			
...		...				
X0DF(h)	Y0DF(h)					
28	4	0E(h)	<b>Leak unit (32 bits integer)</b> Example: Leak Unit Pa/s = 8 On network: 40 1F 00 00 00001F40(H) → 8000(d)/1000 → 8			
			X0E0(h)	Low word (LSW)	Y0E0(h)	RESERVED
			...		...	
			X0EF(h)	Y0EF(h)		
X0F0(h)	High word (MSW)	Y0F0(h)	RESERVED			
...		...				
X0FF(h)	Y0FF(h)					



Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)	Write table (MASTER → ATEQ)		
Extended menu bits						
32	16	10(h)	<b>Extended menu bits reading:</b>		<b>Extended menu bits writing:</b>	
			X100(h)	Extended menu bits - word 0	Y100(h)	Extended menu bits - word 0
			...		...	
			X10F(h)	Extended menu bits - word 1	Y10F(h)	Extended menu bits - word 1
			...		...	
			X110(h)	Extended menu bits - word 2	Y110(h)	Extended menu bits - word 2
			...		...	
			X11F(h)	Extended menu bits - word 3	Y11F(h)	Extended menu bits - word 3
			...		...	
			X120(h)	Extended menu bits - word 4	Y120(h)	Extended menu bits - word 4
			...		...	
			X12F(h)	Extended menu bits - word 5	Y12F(h)	Extended menu bits - word 5
			...		...	
			X130(h)	Extended menu bits - word 6	Y130(h)	Extended menu bits - word 6
			...		...	
			X13F(h)	Extended menu bits - word 7	Y13F(h)	Extended menu bits - word 7
...	...					
X140(h)	Extended menu bits - word 7	Y140(h)	Extended menu bits - word 7			
...		...				
X14F(h)	Extended menu bits - word 7	Y14F(h)	Extended menu bits - word 7			
...		...				
X150(h)	Extended menu bits - word 7	Y150(h)	Extended menu bits - word 7			
...		...				
X15F(h)	Extended menu bits - word 7	Y15F(h)	Extended menu bits - word 7			
...		...				
X160(h)	Extended menu bits - word 7	Y160(h)	Extended menu bits - word 7			
...		...				
X16F(h)	Extended menu bits - word 7	Y16F(h)	Extended menu bits - word 7			
...		...				
X170(h)	Extended menu bits - word 7	Y170(h)	Extended menu bits - word 7			
...		...				
X17F(h)	Extended menu bits - word 7	Y17F(h)	Extended menu bits - word 7			
...		...				



WRr/WRw = 64 words – Function bits handling, Parameters handling, Program name & FIFO / Last results						
Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)		Write table (MASTER → ATEQ)	
48	20	18(h)	<b>Function bits reading:</b>		<b>Function bits writing:</b>	
			WRr00(h)	Function bits - word 0	WRw00(h)	Function bits - word 0
			WRr01(h)	Function bits - word 1	WRw01(h)	Function bits - word 1
			WRr02(h)	Function bits - word 2	WRw02(h)	Function bits - word 2
			WRr03(h)	Function bits - word 3	WRw03(h)	Function bits - word 3
			WRr04(h)	Function bits - word 4	WRw04(h)	Function bits - word 4
			WRr05(h)	Function bits - word 5	WRw05(h)	Function bits - word 5
			WRr06(h)	Function bits - word 6	WRw06(h)	Function bits - word 6
			WRr07(h)	Function bits - word 7	WRw07(h)	Function bits - word 7
			WRr08(h)	Function bits - word 8	WRw08(h)	Function bits - word 8
WRr09(h)	Function bits - word 9	WRw09(h)	Function bits - word 9			
68	64	22(h)	WRr0A(h)	Parameters reading	WRw0A(h)	Parameters writing
			...		...	
			WRr29(h)		WRw29(h)	
132	12	42(h)	WRr2A(h)	Program name reading	WRw2A(h)	Program name writing
			...		...	
			WRr2F(h)		WRw2F(h)	
144	32	48(h)	WRr30(h)	FIFO / Last results	WRw30(h)	<i>RESERVED</i>
			...		...	
			WRr3F(h)		WRw3F(h)	



## Mode 4.STD(-): 3 occupied stations, Octuple (272 bytes)

For the input/output, the real time measurements, 21 parameters managements, functions and extended menu bits and cycle result.

- RX/Ry000-27F(h) = 40 words
- WRr/WRw00-5F(h) = 96 words

READ							
Echo / Error	Relay status	Real time measurements	Results	Extended menu bits	Function bits	Parameters	Program name
ALL	YES	YES	Standard	YES (10 words)	YES (10 words)	YES (21 parameters)	YES

WRITE				
Commands	Extended menu bits	Function bits	Parameters	Program name
ALL	YES (10 words)	YES (10 words)	YES (21 parameters)	YES

```

MAIN /CONFI/FIELDBUS
MODE : 4-Std(-)
NUM OCC : 3
>EXT.CYC : Octuble

MAIN /CONFI/FIELDBUS
NUM OCC : 3
EXT.CYC : Octuble
>I/O size: 272 bytes
  
```

```

MAIN /CONFI/AUTOM/FI
ADDRESS : 001
▶ACCESS : 4.STD(-)
Speed : 10 Mbit/s
MODE : Clear output
* 3 Occ. Stations
* 8 - Octuple
* 272 bytes
  
```

ATEQ FieldBus2 Configurator - Ver: 2.3.2 (15/11/19)

[ Use only with ATEQ version below ]

- ATEQ F5 (Ver >= 3.10h5) - F6 (Ver >= 1.042)
- ATEQ D5 (Ver >= 3.10c5) - D6 (Ver >= 1.000)
- ATEQ MF5 (Ver >= 3.10d5, ASA 30G\_13)
- CC-Link / CC-Link IEFB / CC-Link IE : F5 (Ver >= 3.10r) / F6 (Ver >= 2.007) / D6 (Ver >= 1.025)

COM1

Search Device

Read

Apply

Exit

CC-LINK 485

**CC-Link 485**

Firmware : COMX 10XX-CCS - 2.12.0.1

Address (1-64) : 1 Output Mode :  Clear  Hold

Access Mode : 4.STD(-)

Baud Rate : 10 Mbits/s

1.LITE -> 1 Occ. Stations, Quadruple = 40 bytes  
 2.MEDIUM -> 1 Occ. Stations, Octuple = 80 bytes  
 3.MEDIUM(+) -> 2 Occ. Stations, Octuple = 176 bytes  
 4.STD(-) -> 3 Occ. Stations, Octuple = 272 bytes  
 5.STANDARD -> 4 Occ. Stations, Octuple = 368 bytes





RX/RX = 40 words – Commands, Status, Real time measurements, Extended menu bits handling & Function bits handling						
Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)		Write table (MASTER → ATEQ)	
0	2	00(h)	<b>Command echo:</b>		<b>Command:</b>	
			X000(h)	Echo reset	Y000(h) Reset (stop current cycle)	
			X001(h)	Echo start	Y001(h) Start (starting a test cycle)	
			X002(h)	Echo start special cycle	Y002(h) Start special cycle	
			X003(h)	Echo program selection	Y003(h) Program selection	
			X004(h)	Echo read FIFO results	Y004(h) Read FIFO results	
			X005(h)	Echo read parameters	Y005(h) Read parameters	
			X006(h)	Echo write parameters	Y006(h) Write parameters	
			X007(h)	Echo reset FIFO results	Y007(h) Reset FIFO results	
			X008(h)	RESERVED	Y008(h) RESERVED	
			X009(h)	Echo read ext. menu bits	Y009(h) Read extended menu bits	
			X00A(h)	Echo read functions bits	Y00A(h) Read functions bits	
			X00B(h)	Echo write ext. menu bits	Y00B(h) Write extended menu bits	
			X00C(h)	Echo write functions bits	Y00C(h) Write functions bits	
			X00D(h)	Echo read program name	Y00D(h) Read program name	
			X00E(h)	Echo write program name	Y00E(h) Write program name	
			X00F(h)	Echo read Last results	Y00F(h) Read Last results	
2	2	01(h)	<b>Command error code:</b>			
			X010(h)	Error reset	Y010(h)	
			X011(h)	Error start		
			X012(h)	Error start special cycle		
			X013(h)	Error program selection		
			X014(h)	Error read FIFO results		
			X015(h)	Error read parameters		
			X016(h)	Error write parameters		
			X017(h)	Error reset FIFO results		
			X018(h)	RESERVED	...	
			X019(h)	Error read ext. menu bits		
			X01A(h)	Error read functions bits		
			X01B(h)	Error write ext. menu bits		
			X01C(h)	Error write functions bits		
			X01D(h)	Error read program name		
			X01E(h)	Error write program name		
			X01F(h)	Error read Last results	Y01F(h)	
4	2	02(h)	X020(h)	Refresh counter	Y020(h)	
			...		...	RESERVED
			X02F(h)		Y02F(h)	
6	2	03(h)	X030(h)	Number of current program	Y030(h)	
			...		...	Number of program to be selected
			X03F(h)		Y03F(h)	
8	2	04(h)	X040(h)	Number of FIFO results	Y040(h)	
			...		...	Special cycle code
			X04F(h)		Y04F(h)	
10	2	05(h)	X050(h)	Test type	Y050(h)	
			...		...	RESERVED
			X05F(h)		Y05F(h)	





Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)	Write table (MASTER → ATEQ)
12	2	06(h)	<b>Relay status:</b>	
			X060(h) Pass part (OK)	Y060(h)
			X061(h) Fail test part (NOK)	
			X062(h) Fail reference part (NOK)	
			X063(h) Alarm	
			X064(h) Pressure error	
			X065(h) Cycle end (system ready)	
			X066(h) Part recoverable	...
			X067(h) Calibration error	
			X068(h) Calibration check error	
			X069(h) ATR fault	
			X06A(h) ... <i>Not used, always at 0</i>	
			X06F(h)	Y06F(h)
14	2	07(h)	X070(h) ... Step	Y070(h) ... RESERVED
			X07F(h)	Y07F(h)



Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)	Write table (MASTER → ATEQ)		
<b>Real time measurements</b>						
16	4	08(h)	<b>Pressure value (32 bits integer)</b> Example: Pressure value = 207 On network : 98 28 03 00 00032898(H) → 207000(d)/1000 → 207			
			X080(h)	Low word (LSW)	Y080(h)	RESERVED
			...		...	
			X08F(h)	Y08F(h)		
X090(h)	High word (MSW)	Y090(h)	RESERVED			
...		...				
X09F(h)	Y09F(h)					
20	4	0A(h)	<b>Pressure unit (32 bits integer)</b> Example: Pressure unit Pa = 6 On network: 70 17 00 00 00001770(H) → 6000(d)/1000 → 6			
			X0A0(h)	Low word (LSW)	Y0A0(h)	RESERVED
			...		...	
			X0AF(h)	Y0AF(h)		
X0B0(h)	High word (MSW)	Y0B0(h)	RESERVED			
...		...				
X0BF(h)	Y0BF(h)					
24	4	0C(h)	<b>Leak value (32 bits integer)</b> Example: Leak value = -0.108 On network: 94 FF FF FF FFFFFFF94(H) → -108(d)/1000 → -0.108			
			X0C0(h)	Low word (LSW)	Y0C0(h)	RESERVED
			...		...	
			X0CF(h)	Y0CF(h)		
X0D0(h)	High word (MSW)	Y0D0(h)	RESERVED			
...		...				
X0DF(h)	Y0DF(h)					
28	4	0E(h)	<b>Leak unit (32 bits integer)</b> Example: Leak Unit Pa/s = 8 On network: 40 1F 00 00 00001F40(H) → 8000(d)/1000 → 8			
			X0E0(h)	Low word (LSW)	Y0E0(h)	RESERVED
			...		...	
			X0EF(h)	Y0EF(h)		
X0F0(h)	High word (MSW)	Y0F0(h)	RESERVED			
...		...				
X0FF(h)	Y0FF(h)					
32	8	10(h)	X100(h)	Y100(h)		
			...	RESERVED	...	RESERVED
			X13F(h)	Y13F(h)		





Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)	Write table (MASTER → ATEQ)		
Extended menu & Functions bits						
40	20	14(h)	<b>Extended menu bits reading:</b>	<b>Extended menu bits writing:</b>		
			X140(h)	Y140(h)		
			...	Extended menu bits - word 0	...	Extended menu bits - word 0
			X14F(h)	Y14F(h)		
			X150(h)	Y150(h)		
			...	Extended menu bits - word 1	...	Extended menu bits - word 1
			X15F(h)	Y15F(h)		
			X160(h)	Y160(h)		
			...	Extended menu bits - word 2	...	Extended menu bits - word 2
			X16F(h)	Y16F(h)		
			X170(h)	Y170(h)		
			...	Extended menu bits - word 3	...	Extended menu bits - word 3
			X17F(h)	Y17F(h)		
			X180(h)	Y180(h)		
			...	Extended menu bits - word 4	...	Extended menu bits - word 4
			X18F(h)	Y18F(h)		
			X190(h)	Y190(h)		
			...	Extended menu bits - word 5	...	Extended menu bits - word 5
			X19F(h)	Y19F(h)		
			X1A0(h)	Y1A0(h)		
...	Extended menu bits - word 6	...	Extended menu bits - word 6			
X1AF(h)	Y1AF(h)					
X1B0(h)	Y1B0(h)					
...	Extended menu bits - word 7	...	Extended menu bits - word 7			
X1BF(h)	Y1BF(h)					
X1C0(h)	Y1C0(h)					
...	Extended menu bits - word 8	...	Extended menu bits - word 8			
X1CF(h)	Y1CF(h)					
X1D0(h)	Y1D0(h)					
...	Extended menu bits - word 9	...	Extended menu bits - word 9			
X1DF(h)	Y1DF(h)					



Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)		Write table (MASTER → ATEQ)	
60	20	1E(h)	<b>Function bits reading:</b>		<b>Function bits writing:</b>	
			X1E0(h)	Function bits - word 0	Y1E0(h)	Function bits - word 0
			...		X1EF(h)	
			X1F0(h)	Function bits - word 1	Y1F0(h)	Function bits - word 1
			...		X1FF(h)	
			X200(h)	Function bits - word 2	Y200(h)	Function bits - word 2
			...		X20F(h)	
			X210(h)	Function bits - word 3	Y210(h)	Function bits - word 3
			...		X21F(h)	
			X220(h)	Function bits - word 4	Y220(h)	Function bits - word 4
			...		X22F(h)	
			X230(h)	Function bits - word 5	Y230(h)	Function bits - word 5
			...		X23F(h)	
			X240(h)	Function bits - word 6	Y240(h)	Function bits - word 6
			...		X24F(h)	
			X250(h)	Function bits - word 7	Y250(h)	Function bits - word 7
			...		X25F(h)	
			X260(h)	Function bits - word 8	Y260(h)	Function bits - word 8
			...		X26F(h)	
			X270(h)	Function bits - word 9	Y270(h)	Function bits - word 9
...	X27F(h)					

WRr/WRw = 96 words – Parameters handling, Program name & FIFO / Last results						
Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)		Write table (MASTER → ATEQ)	
80	128	28(h)	WRr00(h)	Parameters reading	WRw00(h)	Parameters writing
			...		WRw39(h)	
			WRr39(h)			
208	16	68(h)	WRr40(h)	Program name reading	WRw40(h)	Program name writing
			...		WRw47(h)	
			WRr47(h)			
224	36	70(h)	WRr48(h)	FIFO / Last results	WRw48(h)	RESERVED
			...		WRw59(h)	
			WRr59(h)			
260	12	82(h)	WRr5A(h)	RESERVED	WRw5A(h)	RESERVED
			...		WRw5F(h)	
			WRr5F(h)			





## Mode 5.STANDARD: 4 occupied stations, Octuple (368 bytes)

For the input/output, the real time measurements, 21 parameters managements, functions and extended menu bits and extended cycle result.

- RX/Ry000-37F(h) = 56 words
- WRr/WRw00-7F(h) = 128 words

READ							
Echo / Error	Relay status	Real time measurements	Results	Extended menu bits	Function bits	Parameters	Program name
ALL	YES	YES	Extended	YES (10 words)	YES (10 words)	YES (21 parameters)	YES

WRITE				
Commands	Extended menu bits	Function bits	Parameters	Program name
ALL	YES (10 words)	YES (10 words)	YES (21 parameters)	YES

```

MAIN /CONFI/FIELDBUS
>MODE : 5-Standard
NUM OCC : 4
EXT.CYC : Octuple
  
```

```

MAIN /CONFI/FIELDBUS
NUM OCC : 4
EXT.CYC : Octuple
>I/O size: 368 bytes
  
```

```

MAIN /CONFI/AUTOM/FI
ADDRESS : 001
▶ACCESS : 5.STANDARD
Speed : 10 Mbit/s
MODE : Clear output
* 4 Occ. Stations
* 8 - octuple
* 368 bytes
  
```

ATEQ FieldBus2 Configurator - Ver: 2.3.2 (15/11/19)

[ Use only with ATEQ version below ]

- ATEQ F5 (Ver >= 3.10h5) - F6 (Ver >= 1.042)
- ATEQ D5 (Ver >= 3.10c5) - D6 (Ver >= 1.000)
- ATEQ MF5 (Ver >= 3.10d5, ASA 30G\_13)
- CC-Link / CC-Link IEFB / CC-Link IE : F5 (Ver >= 3.10r) / F6 (Ver >= 2.007) / D6 (Ver >= 1.025)

COM1

Search Device

Read

Apply

Exit

CC-LINK 485

**CC-Link 485**

Firmware : COMX 10XX-CCS - 2.12.0.1

Address (1-64) : 1 Output Mode :  Clear  Hold

Access Mode : 5.STANDARD

Baud Rate : 10 Mbits/s

1.LITE -> 1 Occ. Stations, Quadruple = 40 bytes  
 2.MEDIUM -> 1 Occ. Stations, Octuple = 80 bytes  
 3.MEDIUM(+) -> 2 Occ. Stations, Octuple = 176 bytes  
 4.STD(-) -> 3 Occ. Stations, Octuple = 272 bytes  
 5.STANDARD -> 4 Occ. Stations, Octuple = 368 bytes





RX/RX = 40 words – Commands, Status, Real time measurements, Extended menu bits handling & Function bits handling						
Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)		Write table (MASTER → ATEQ)	
0	2	00(h)	<b>Command echo:</b>		<b>Command:</b>	
			X000(h)	Echo reset	Y000(h) Reset (stop current cycle)	
			X001(h)	Echo start	Y001(h) Start (starting a test cycle)	
			X002(h)	Echo start special cycle	Y002(h) Start special cycle	
			X003(h)	Echo program selection	Y003(h) Program selection	
			X004(h)	Echo read FIFO results	Y004(h) Read FIFO results	
			X005(h)	Echo read parameters	Y005(h) Read parameters	
			X006(h)	Echo write parameters	Y006(h) Write parameters	
			X007(h)	Echo reset FIFO results	Y007(h) Reset FIFO results	
			X008(h)	RESERVED	Y008(h) RESERVED	
			X009(h)	Echo read ext. menu bits	Y009(h) Read extended menu bits	
			X00A(h)	Echo read functions bits	Y00A(h) Read functions bits	
			X00B(h)	Echo write ext. menu bits	Y00B(h) Write extended menu bits	
			X00C(h)	Echo write functions bits	Y00C(h) Write functions bits	
			X00D(h)	Echo read program name	Y00D(h) Read program name	
			X00E(h)	Echo write program name	Y00E(h) Write program name	
			X00F(h)	Echo read Last results	Y00F(h) Read Last results	
2	2	01(h)	<b>Command error code:</b>			
			X010(h)	Error reset	Y010(h)	
			X011(h)	Error start		
			X012(h)	Error start special cycle		
			X013(h)	Error program selection		
			X014(h)	Error read FIFO results		
			X015(h)	Error read parameters		
			X016(h)	Error write parameters		
			X017(h)	Error reset FIFO results		
			X018(h)	RESERVED	...	
			X019(h)	Error read ext. menu bits		
			X01A(h)	Error read functions bits		
			X01B(h)	Error write ext. menu bits		
			X01C(h)	Error write functions bits		
			X01D(h)	Error read program name		
			X01E(h)	Error write program name		
			X01F(h)	Error read Last results	Y01F(h)	
4	2	02(h)	X020(h)	Refresh counter	Y020(h)	
			...		...	RESERVED
			X02F(h)		Y02F(h)	
6	2	03(h)	X030(h)	Number of current program	Y030(h)	
			...		...	Number of program to be selected
			X03F(h)		Y03F(h)	
8	2	04(h)	X040(h)	Number of FIFO results	Y040(h)	
			...		...	Special cycle code
			X04F(h)		Y04F(h)	
10	2	05(h)	X050(h)	Test type	Y050(h)	
			...		...	RESERVED
			X05F(h)		Y05F(h)	





Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)	Write table (MASTER → ATEQ)
12	2	06(h)	<b>Relay status:</b>	
			X060(h) Pass part (OK)	Y060(h)
			X061(h) Fail test part (NOK)	
			X062(h) Fail reference part (NOK)	
			X063(h) Alarm	
			X064(h) Pressure error	
			X065(h) Cycle end (system ready)	
			X066(h) Part recoverable	...
			X067(h) Calibration error	
			X068(h) Calibration check error	
			X069(h) ATR fault	
			X06A(h) ... <i>Not used, always at 0</i>	
			X06F(h)	Y06F(h)
14	2	07(h)	X070(h) ... Step	Y070(h) ... RESERVED
			X07F(h)	Y07F(h)



Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)	Write table (MASTER → ATEQ)		
<b>Real time measurements</b>						
16	4	08(h)	<b>Pressure value (32 bits integer)</b> Example: Pressure value = 207 On network : 98 28 03 00 00032898(H) → 207000(d)/1000 → 207			
			X080(h)	Low word (LSW)	Y080(h)	RESERVED
			...		...	
			X08F(h)	Y08F(h)		
X090(h)	High word (MSW)	Y090(h)	RESERVED			
...		...				
X09F(h)	Y09F(h)					
20	4	0A(h)	<b>Pressure unit (32 bits integer)</b> Example: Pressure unit Pa = 6 On network: 70 17 00 00 00001770(H) → 6000(d)/1000 → 6			
			X0A0(h)	Low word (LSW)	Y0A0(h)	RESERVED
			...		...	
			X0AF(h)	Y0AF(h)		
X0B0(h)	High word (MSW)	Y0B0(h)	RESERVED			
...		...				
X0BF(h)	Y0BF(h)					
24	4	0C(h)	<b>Leak value (32 bits integer)</b> Example: Leak value = -0.108 On network: 94 FF FF FF FFFFFFF94(H) → -108(d)/1000 → -0.108			
			X0C0(h)	Low word (LSW)	Y0C0(h)	RESERVED
			...		...	
			X0CF(h)	Y0CF(h)		
X0D0(h)	High word (MSW)	Y0D0(h)	RESERVED			
...		...				
X0DF(h)	Y0DF(h)					
28	4	0E(h)	<b>Leak unit (32 bits integer)</b> Example: Leak Unit Pa/s = 8 On network: 40 1F 00 00 00001F40(H) → 8000(d)/1000 → 8			
			X0E0(h)	Low word (LSW)	Y0E0(h)	RESERVED
			...		...	
			X0EF(h)	Y0EF(h)		
X0F0(h)	High word (MSW)	Y0F0(h)	RESERVED			
...		...				
X0FF(h)	Y0FF(h)					
32	40	10(h)	X100(h)	Y100(h)		
			...	RESERVED	...	RESERVED
			X13F(h)	Y13F(h)		



Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)	Write table (MASTER → ATEQ)		
Extended menu & Functions bits						
72	20	24(h)	<b>Extended menu bits reading:</b>	<b>Extended menu bits writing:</b>		
			X240(h)	Y240(h)		
			...	Extended menu bits - word 0	...	Extended menu bits - word 0
			X24F(h)	Y24F(h)		
			X250(h)	Y250(h)		
			...	Extended menu bits - word 1	...	Extended menu bits - word 1
			X25F(h)	Y25F(h)		
			X260(h)	Y260(h)		
			...	Extended menu bits - word 2	...	Extended menu bits - word 2
			X26F(h)	Y26F(h)		
			X270(h)	Y270(h)		
			...	Extended menu bits - word 3	...	Extended menu bits - word 3
			X27F(h)	Y27F(h)		
			X280(h)	Y280(h)		
			...	Extended menu bits - word 4	...	Extended menu bits - word 4
			X28F(h)	Y28F(h)		
			X290(h)	Y290(h)		
			...	Extended menu bits - word 5	...	Extended menu bits - word 5
			X29F(h)	Y29F(h)		
			X2A0(h)	Y2A0(h)		
...	Extended menu bits - word 6	...	Extended menu bits - word 6			
X2AF(h)	Y2AF(h)					
X2B0(h)	Y2B0(h)					
...	Extended menu bits - word 7	...	Extended menu bits - word 7			
X2BF(h)	Y2BF(h)					
X2C0(h)	Y2C0(h)					
...	Extended menu bits - word 8	...	Extended menu bits - word 8			
X2CF(h)	Y2CF(h)					
X2D0(h)	Y2D0(h)					
...	Extended menu bits - word 9	...	Extended menu bits - word 9			
X2DF(h)	Y2DF(h)					



Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)		Write table (MASTER → ATEQ)	
92	20	2E(h)	<b>Function bits reading:</b>		<b>Function bits writing:</b>	
			X2E0(h)	Function bits - word 0	Y2E0(h)	Function bits - word 0
			...		X2EF(h)	
			X2F0(h)	Function bits - word 1	Y2F0(h)	Function bits - word 1
			...		X2FF(h)	
			X300(h)	Function bits - word 2	Y300(h)	Function bits - word 2
			...		X30F(h)	
			X310(h)	Function bits - word 3	Y310(h)	Function bits - word 3
			...		X31F(h)	
			X320(h)	Function bits - word 4	Y320(h)	Function bits - word 4
			...		X32F(h)	
			X330(h)	Function bits - word 5	Y330(h)	Function bits - word 5
			...		X33F(h)	
			X340(h)	Function bits - word 6	Y340(h)	Function bits - word 6
			...		X34F(h)	
			X350(h)	Function bits - word 7	Y350(h)	Function bits - word 7
			...		X35F(h)	
			X360(h)	Function bits - word 8	Y360(h)	Function bits - word 8
			...		X36F(h)	
			X370(h)	Function bits - word 9	Y370(h)	Function bits - word 9
...	X37F(h)	Y37F(h)				

WRr/WRw = 96 words – Parameters handling, Program name & FIFO / Last results						
Offset (byte)	Length (byte)	Offset (word)	Read table (MASTER ← ATEQ)		Write table (MASTER → ATEQ)	
112	128	38(h)	WRr00(h)	Parameters reading	WRw00(h)	Parameters writing
			...		WRw39(h)	
240	16	78(h)	WRr40(h)	Program name reading	WRw40(h)	Program name writing
			...		WRw47(h)	
256	80	80(h)	WRr48(h)	FIFO / Last results	WRw48(h)	RESERVED
			...		WRw6F(h)	
336	12	A8(h)	WRr70(h)	RESERVED	WRw70(h)	RESERVED
			...		WRw7F(h)	





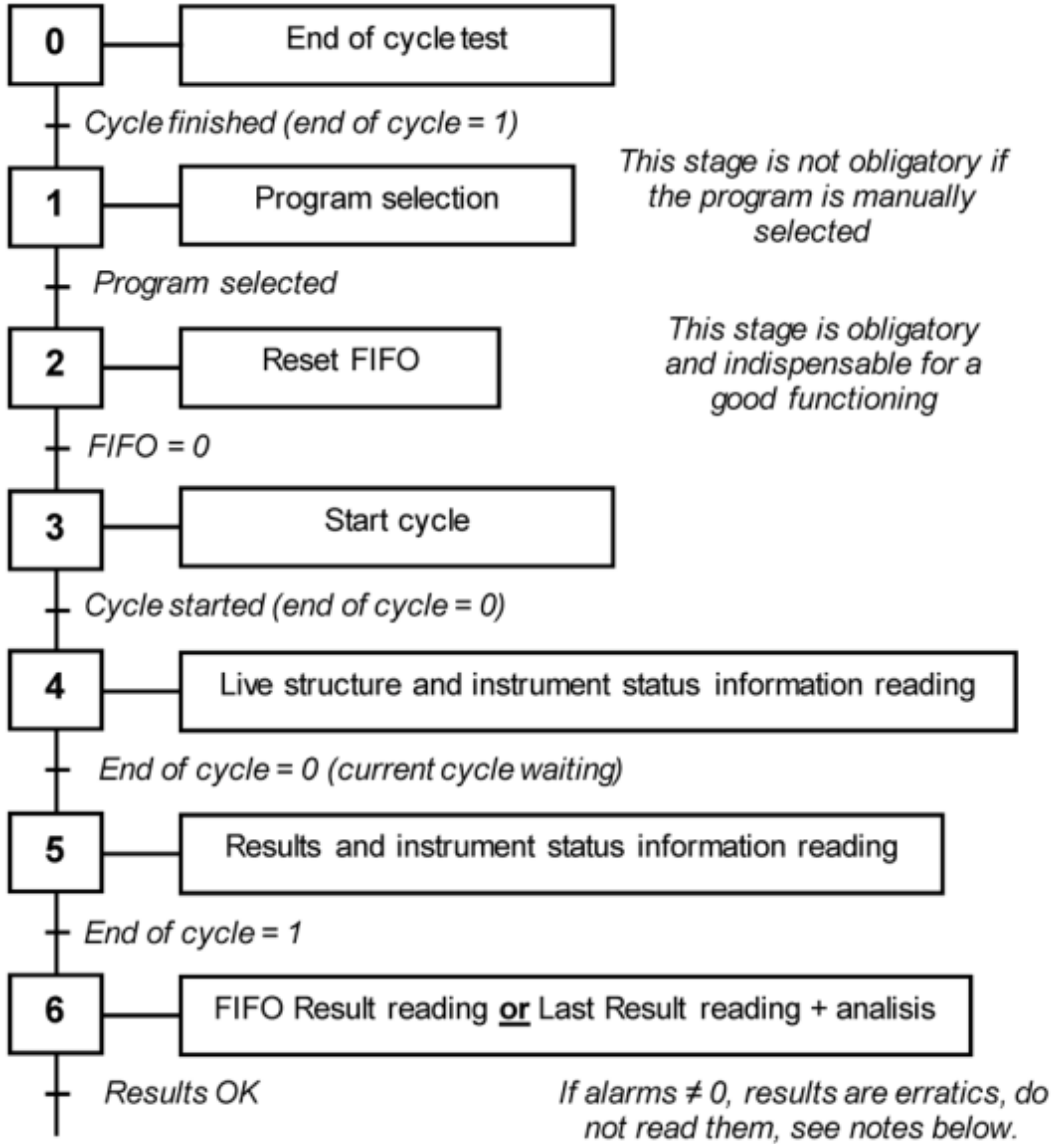


## Treatment of the commands

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

### ATEQ using

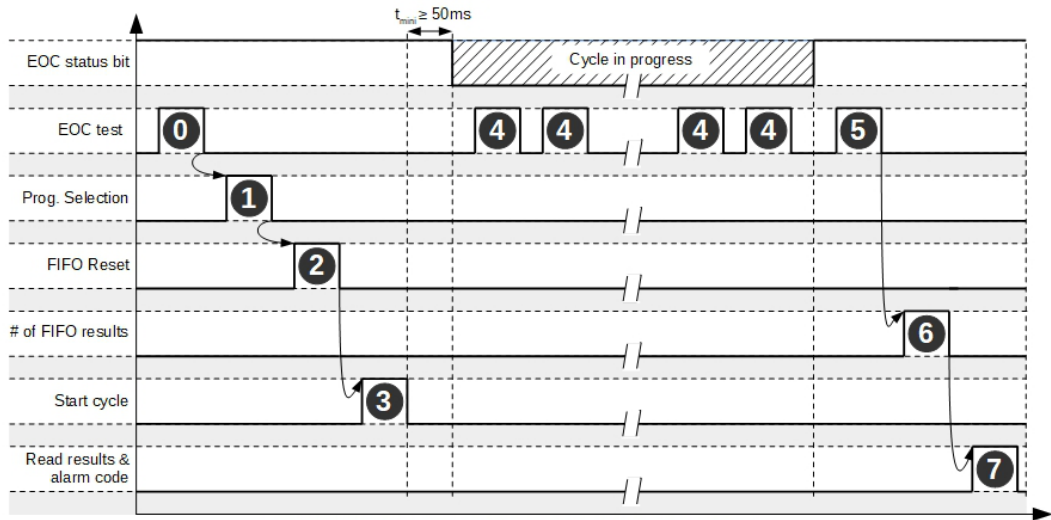
Base procedure for using an ATEQ instrument:



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



## Fieldbus progress chart



**WARNING : The status bits update rate is about 50ms**

<p><b>0</b> : Read @06(h) : Status bit 5 = 1 (EOC status bit)</p>	<p><b>6</b> : Read the number of results in FIFO : Read @04(h) : <b>if &gt; 0</b> go to step 7, <b>else</b> END</p>	<p><i>Use of FIFO Results</i></p>
<p><b>1</b> : Write @03(h) : 1 word = n° prog (0001h = prog 2) Write @00(h) : Y003(h) = 1 (cmd « Prog. Selection »)</p>	<p><b>7</b> : Write @00(h) : Y004(h) = 1 (command « Read FIFO results ») Read Exchange datas : FIFO/Last results (standard or extended) <b>if Alarm Code = 0</b> go to step 8, <b>else</b> END</p>	
<p><b>2</b> : <b>ALWAYS RESET THE FIFO</b> Write @00(h) : Y007(h) = 1 (cmd « Reset FIFO »)</p>	<p><b>8</b> : Use the results recovered at step 7 (if Alarm code was equal to 0)</p>	
<p><b>3</b> : Write @00(h) : Y001(h) = 1 (cmd « Start ») <math>t_{min} \geq 50ms</math></p>	<p><b>6</b> : Read the number of results in FIFO : Read @04(h) : <b>if ≠ 1</b> go to step 7, <b>else</b> END</p>	<p><i>Use of Last Results</i></p>
<p><b>4</b> : Read @06(h) : Status bit 5 = 0 (EOC status bit)</p>	<p><b>7</b> : Write @00(h) : Y00F(h) = 1 (command « Read Last results ») Read Exchange datas : FIFO/Last results (standard or extended) <b>if Alarm Code = 0</b> go to step 8, <b>else</b> END</p>	
<p><b>5</b> : Read @06(h) : Status bit 5 = 1 (EOC status bit)</p>	<p><b>8</b> : Use the results recovered at step 7 (if Alarm code was equal to 0)</p>	



## CONFIGURATION

### General configuration

#### Table of configuration / extended menu bits

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

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.

- i** Acronyms used in the “Menu” column:
- Conf: CONFIGURATION
  - +Func: FUNCTIONS > More functions...
  - RS232: CONFIGURATION > RS232

Word	Bit n°	Mask		Meaning	Menu
		Hexa	Dec		
1	0	0001	1	Permanent blowing activation.	Conf
	1	0002	2	Reserved.	
	2	0004	4	Fill type.	+Func
	3	0008	8	Pre-fill type.	+Func
	4	0010	16	Recovery thresholds.	+Func
	5	0020	32	Cycle end.	+Func
	6	0040	64	Reserved.	
	7	0080	128	Peak meter.	+Func
	8	0100	256	Reserved.	
	9	0200	512	Reference volume.	+Func
	10	0400	1024	ATR 0.	+Func
	11	0800	2048	ATR 1.	+Func
	12	1000	4096	ATR 2.	+Func
	13	2000	8192	Program name.	+Func
	14	4000	16384	Chaining.	+Func
15	8000	32768	Automatic connector.	+Func	





Word	Bit n°	Mask		Meaning	Menu
		Hexa	Dec		
2	16	0001	1	Calibration check.	+Funct
	17	0002	2	Valve codes (output codes).	+Funct
	18	0004	4	Sealed component (Leak unit always PA, not changeable).	+Funct
	19	0008	8	Stamping.	+Funct
	20	0010	16	Reserved.	
	21	0020	32	N test.	+Funct
	22	0040	64	Reserved.	
	23	0080	128	Sending cond.: pass part.	RS232
	24	0100	256	Sending cond.: fail test part.	RS232
	25	0200	512	Sending cond.: fail ref. part.	RS232
	26	0400	1024	Sending cond.: alarm presence.	RS232
	27	0800	2048	Sending cond.: pressure error.	RS232
	28	1000	4096	Sending cond.: end of cycle.	RS232
	29	2000	8192	Sending cond.: recoverable.	RS232
	30	4000	16384	Sending cond.: calibration.	RS232
3	31	8000	32768	Frame content: time stamp.	RS232
	32	0001	1	Frame content: name.	RS232
	33	0002	2	Content of the frame: pressure.	RS232
	34	0004	4	Security.	Conf
	35	0008	8	External dump.	Conf
	36	0010	16	Exportation.	RS232
	37	0020	32	Automatic reset.	Conf
	38	0040	64	Reserved.	
	39	0080	128	Reserved.	
	40	0100	256	Reserved.	
	41	0200	512	Reserved.	
	42	0400	1024	Reserved.	
	43	0800	2048	Parameters automatic setting.	Conf
	44	1000	4096	Reserved.	
	45	2000	8192	Page feed.	RS232
	46	4000	16384	Sign change.	+Funct
	47	8000	32768	After sale service cycle.	+Funct



Word	Bit n°	Mask		Meaning	Menu
		Hexa	Dec		
4	48	0001	1	Unit type.	+Funct
	49	0002	2	Automatic reset piezo 2.	Conf
	50	0004	4	Reserved.	
	51	0008	8	Electronic regulator mode.	Conf
	52	0010	16	Auxiliary codes activation.	+Funct
	53	0020	32	Filtering.	+Funct
	54	0040	64	Reserved.	
	55	0080	128	Quick automatic reset activation.	Conf
	56	0100	256	Permanent electronic regulator.	Conf
	57	0200	512	Bar code.	Conf
	58	0400	1028	Flow reject.	+Funct
	59	0800	2048	No negative.	+Funct
	60	1000	4096	Dump threshold.	+Funct
	61	2000	8192	ATR 3.	+Funct
	62	4000	16384	In 7 test configuration.	Conf
5	63	8000	32768	Reserved.	
	64	0001	1	Absolute value.	Conf
	65	0002	2	Leak display mode.	+Funct
	66	0004	4	By pass valve.	Conf
	67	0008	8	Reserved.	
	68	0010	16	Reserved.	
	69	0020	32	Reserved.	
	70	0040	64	Reserved.	+Funct
	71	0080	128	Dump Off.	+Funct
	72	0100	256	Program selection on bar code reading.	+Funct
	73	0200	512	Bar code reset on end of cycle.	+Funct
	74	0400	1024	Cut Off.	+Funct
	75	0800	2048	ATF.	+Funct
6	76	1000	4096	Reserved.	
	77	2000	8192	Reserved.	
	78	4000	16384	Reserved.	
	79	8000	32768	Reserved.	
6	80 > 95			Word Reserved.	



Word	Bit n°	Mask		Meaning	Menu
		Hexa	Dec		
7	96	0001	1	Buzzer function.	+Funct
	97	0002	2	Long test (x100) function.	+Funct
	98	0004	4	Permanent blowing.	Conf
	99	0008	8	Reserved.	
	100	0010	16	Reserved.	
	101	0020	32	Display optional.	+Funct
	102	0040	64	Pressure Drop.	+Funct
	103	0080	128	Pressure correction ( $\geq v1.400$ ).	+Funct
	104	0100	256	Standard conditions ( $\geq v1.400$ ).	+Funct
	105	0200	512	Ref No Dump.	+Funct
	106	0400	1024	Reserved.	
	107	0800	2048	Offset.	+Funct
	108	1000	4096	Reserved.	
	109	2000	8192	Reserved.	
	110	4000	16384	Auto Selection Prog.	Conf
111	8000	32768	Save Volume Selection.	Conf	

Example: bit number 13 (automatic mode) 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

Master	Slave
<ul style="list-style-type: none"> <li>— Activate the “Read extended menu bits” command:</li> <li>— Write at the address 00(h), the value <b>0200(h)</b> Byte 0 = 00(h) Byte 1 = 02(h) (Y009(h) = 1)</li> </ul>	
	<p style="text-align: center;"><u>Acknowledgement</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 00(h)</li> <li>— Byte 1 = 02(h) (X009(h) = 1)</li> </ul> <p>Command error code:</p> <ul style="list-style-type: none"> <li>— Byte 2 = FF(h)</li> <li>— Byte 3 = FF(h)</li> </ul> <p>(if command error code = FFFF(h), command is in progress)</p> <p>Running “Read extended menu bits” command</p>
	<p style="text-align: center;"><u>Command finished</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 00(h)</li> <li>— Byte 1 = 02(h) (X009(h) = 1)</li> </ul> <p>Command error code if the command is correctly carried out:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 00(h)</li> <li>— Byte 3 = 00(h)</li> </ul> <p>OR if an error occurred during the command:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 00(h)</li> <li>— Byte 3 = 02(h) (X019(h) = 1)</li> </ul>
<ul style="list-style-type: none"> <li>— Wait the end of the command: command echo = 0200(h) (X009(h) = 1) command error code ≠ FFFF(h) (end of command)</li> </ul>	
<ul style="list-style-type: none"> <li>— Deactivate the “Read extended menu bits” command: Write at the address 00(h) the value <b>0000(h)</b> Byte 0 = 00(h) Byte 1 = 00(h) (Y009(h) = 0)</li> </ul>	
<ul style="list-style-type: none"> <li>— Read the extended menu bits in the corresponding area depending on the configuration mode of the slave</li> </ul>	

**i** The configuration / extended menu bits are defined in the table above for the “Extended menus” of each specific chapter for the instruments.

**i** The configuration / extended menu bits are independent of the program number.

**!** The master instrument must always set to zero the command bit. If it is not done, the slave instrument will not detect the following command on this bit. It has detection on the rising edge (when the bit state goes from 0 to 1).





## Writing of the configuration / extended menu bits

Master	Slave
<ul style="list-style-type: none"> <li>— Write the extended menu bits in the corresponding area depending on the configuration mode of the slave</li> <li>— Activate the “Write extended menu bits” command: Write at the address 00(h), the value <b>0800(h)</b> Byte 0 = 00(h) Byte 1 = 08(h) (Y00B(h) = 1)</li> </ul>	
	<p align="center"><u>Acknowledgement</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 00(h)</li> <li>— Byte 1 = 08(h) (X00B(h) = 1)</li> </ul> <p>Command error code:</p> <ul style="list-style-type: none"> <li>— Byte 2 = FF(h)</li> <li>— Byte 3 = FF(h)</li> </ul> <p>(if command error code = FFFF(h), command is in progress)</p>
	Running “Write extended menu bits” command
	<p align="center"><u>Command finished</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 00(h)</li> <li>— Byte 1 = 08(h) (X00B(h) = 1)</li> </ul> <p>Command error code if the command is correctly carried out:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 00(h)</li> <li>— Byte 3 = 00(h)</li> </ul> <p>OR if an error occurred during the command:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 00(h)</li> <li>— Byte 3 = 08(h) (X01B(h) = 1)</li> </ul>
<ul style="list-style-type: none"> <li>— Wait the end of the command: command echo = 0200(h) (X00B(h) = 1) command error code ≠ FFFF(h) (end of command)</li> </ul>	
<ul style="list-style-type: none"> <li>— Deactivate the “Write extended menu bits” command: Write at the address 00(h) the value <b>0000(h)</b> Byte 0 = 00(h) Byte 1 = 00(h) (Y00B(h) = 0)</li> </ul>	



The configuration / extended menu bits are defined in the table above for the “Extended menus” of each specific chapter for the instruments.



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



The master instrument must always set to zero the command bit. If it is not done, the slave instrument will not detect the following command on this bit. It has detection on the rising edge (when the bit state goes from 0 to 1).







## Program

### Program selection command on the ATEQ device

Master	Slave
<ul style="list-style-type: none"> <li>— Write 1 word at the address 03(h) corresponding to the program number to be selected: @03(h) = 0001(h) (= program n°2)</li> <li>— Activate the “Program selection” command: Write at the address 00(h), the value <b>0008(h)</b> Byte 0 = 08(h) (Y003(h) = 1) Byte 1 = 00(h)</li> </ul>	
	<p style="text-align: center;"><u>Acknowledgement</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 08(h) (X003(h) = 1)</li> <li>— Byte 1 = 00(h)</li> </ul> <p>Command error code:</p> <ul style="list-style-type: none"> <li>— Byte 2 = FF(h)</li> <li>— Byte 3 = FF(h)</li> </ul> <p>(if command error code = FFFF(h), command is in progress)</p>
	Running “Program selection” command
	<p style="text-align: center;"><u>Command finished</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 08(h) (X003(h) = 1)</li> <li>— Byte 1 = 00(h)</li> </ul> <p>Command error code if the command is correctly carried out:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 00(h)</li> <li>— Byte 3 = 00(h)</li> </ul> <p>OR if an error occurred during the command:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 08(h) (X013(h) = 1)</li> <li>— Byte 3 = 00(h)</li> </ul>
<ul style="list-style-type: none"> <li>— Wait the end of the command: command echo = 0008(h) (X003(h) = 1) command error code ≠ FFFF(h) (end of command)</li> </ul>	
<ul style="list-style-type: none"> <li>— Deactivate the “Program selection” command: Write at the address 00(h) the value <b>0000(h)</b> Byte 0 = 00(h) (Y003(h) = 0) Byte 1 = 00(h)</li> </ul>	



The master instrument must always set to zero the command bit. If it is not done, the slave instrument will not detect the following command on this bit. It has detection on the rising edge (when the bit state goes from 0 to 1).





## Function

### Table of the function bits

Table of the function bits per program.



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

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		Meaning	Menu
		Hexa	Dec		
1	0	0001	1	Fill regulator Number.	Funct
	1	0002	2	Pre-fill regulator Number.	Funct
	2	0004	4	Fill type function.	Funct
	3	0008	8	Pre-fill type function.	Funct
	4	0010	16	Recovery level function.	Funct
	5	0020	32	End of cycle function.	Funct
	6	0040	64	Automatic reset end cycle function.	Funct
	7	0080	128	Reset and dump end of cycle function.	Funct
	8	0100	256	Fill mode end of cycle function.	Funct
	9	0200	512	Peak hold function.	Funct
	10	0400	1024	Reference volume function.	Funct
	11	0800	2048	ATR0 function.	Funct
	12	1000	4096	ATR1 function.	Funct
	13	2000	8192	ATR2 function.	Funct
	14	4000	16384	Sequencing function.	Funct
15	8000	32768	Pass part sequencing function.	Funct	
2	16	0001	1	Fail test part sequencing function.	Funct
	17	0002	2	Fail reference part sequencing function.	Funct
	18	0004	4	Alarm sequencing function.	Funct
	19	0008	8	Pressure fault sequencing function.	Funct
	20	0010	16	End of cycle sequencing function.	Funct
	21	0020	32	Reserved.	
	22	0040	64	Recovery part sequencing function.	Funct
	23	0080	128	Calibration check sequencing function.	Funct
	24	0100	256	Automatic connector function.	Funct
	25	0200	512	Calibration check function.	Funct
	26	0400	1024	Valve code function.	Funct
	27	0800	2048	External valve code 1 function.	Funct
	28	1000	4096	External valve code 2 function.	Funct
	29	2000	8192	External valve code 3 function.	Funct
	30	4000	16384	External valve code 4 function.	Funct
	31	8000	32768	External valve code 5 function.	Funct





Word	Bit n°	Mask		Meaning	Menu
		Hexa	Dec		
3	32	0001	1	External valve code 6 function.	Funct
	33	0002	2	Internal valve code 1 function.	Funct
	34	0004	4	Internal valve code 2 function.	Funct
	35	0008	8	Stamp function.	Funct
	36	0010	16	Pass part stamp function.	Funct
	37	0020	32	Fail test part stamp function.	Funct
	38	0040	64	Fail reference part stamp function.	Funct
	39	0080	128	Alarm stamp function.	Funct
	40	0100	256	Pressure fault stamp function.	Funct
	41	0200	512	End of cycle stamp function.	Funct
	42	0400	1024	Recovery part stamp function.	Funct
	43	0800	2048	Calibration check stamp function.	Funct
	44	1000	4096	N test function.	Funct
	45	2000	8192	Reserved.	
	46	4000	16384	Reserved.	
47	8000	32768	Reserved.		
4	48	0001	1	External dump function.	Funct
	49	0002	2	Reserved.	
	50	0004	4	Reserved.	
	51	0008	8	Reserved.	
	52	0010	16	Sign change function.	Funct
	53	0020	32	Obligatory reset on end of cycle.	Funct
	54	0040	64	Auxiliaries codes function.	Funct
	55	0080	128	Auxiliaries codes 1 function.	Funct
	56	0100	256	Auxiliaries codes 2 function.	Funct
	57	0200	512	Auxiliaries codes 3 function.	Funct
	58	0400	1024	Auxiliaries codes 4 function.	Funct
	59	0800	2048	Reserved.	
	60	1000	4096	Filtering function.	Funct
	61	2000	8192	Bar code function.	Funct
	62	4000	16384	Flow reject function.	Funct
	63	8000	32768	No negative function.	Funct



Word	Bit n°	Mask		Meaning	Menu
		Hexa	Dec		
5	64	0001	1	Start after reading bar code function.	Funct
	65	0002	2	ATR3 function.	Funct
	66	0004	4	Absolute value function.	Funct
	67	0008	8	Bypass valve function.	Funct
	68	0010	16	Reserved.	
	69	0020	32	Reserved.	
	70	0040	64	Reserved.	
	71	0080	128	Dump off function.	Funct
	72	0100	256	Cut off function.	Funct
	73	0200	512	ATF function.	Funct
	74	0400	1024	Reserved.	
	75 > 79			Reserved.	
6	80 > 95			Word Reserved.	
7	96	0001	1	Optional auxiliaries codes function.	Funct
	97	0002	2	Optional auxiliaries codes 1 function.	Funct
	98	0004	4	Optional auxiliaries codes 2 function.	Funct
	99	0008	8	Optional auxiliaries codes 3function.	Funct
	100	0010	16	Optional auxiliaries codes 4 function.	Funct
	101	0020	32	Optional valves codes function.	Funct
	102	0040	64	Optional external valves codes 1.	Funct
	103	0080	128	Optional external valves codes 2.	Funct
	104	0100	256	Optional external valves codes 3.	Funct
	105	0200	512	Optional external valves codes 4.	Funct
	106	0400	1024	Optional external valves codes 5.	Funct
	107	0800	2048	Optional external valves codes 6.	Funct
	108	1000	4096	Optional internal valves codes 1.	Funct
	109	2000	8192	Optional internal valves codes 2.	Funct
110	4000	16384	Buzzer function.	Funct	
111	8000	32768	Pass part buzzer function.	Funct	
8	112	0001	1	Fail part buzzer function.	Funct
	113	0002	2	Alarm buzzer function.	Funct
	114	0004	4	End of cycle buzzer function.	Funct
	115	0008	8	Long Test Time function.	Funct
	116	0010	16	Permanent dump function.	Funct
	117	0020	32	Input 7 test function.	Funct
	118	0040	64	Reserved.	
	119	0080	128	Reserved.	
	120	0100	256	Reserved.	
	121	0200	512	Pressure Drop.	Funct
	122	0400	1024	Auto Verif Etal.	Funct
	123	0800	2048	Pressure correction (≥v1.400).	Funct
	124	1000	4096	Standard conditions (≥v1.400).	Funct
	125	2000	8192	Ref No Dump.	Funct
	126	4000	16384	Offset.	Funct
	127	8000	32768	Permanent Fill.	Funct



Word	Bit n°	Mask		Meaning	Menu
		Hexa	Dec		
9	128	0001	1	Reserved.	
	129>143			Reserved.	

Example: bit number 46 (Offset function) activated on 1, will put to "4000h" the value in the third 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

Master	Slave
<ul style="list-style-type: none"> <li>Select the program number on which the functions bits have to be read</li> <li>Activate the "Read functions bits" command: Write at the address 00(h), the value <b>0400(h)</b> Byte 0 = 00(h) Byte 1 = 04(h) (Y00A(h) = 1)</li> </ul>	
	<p align="center"><u>Acknowledgement</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>Byte 0 = 00(h)</li> <li>Byte 1 = 04(h) (X00A(h) = 1)</li> </ul> <p>Command error code:</p> <ul style="list-style-type: none"> <li>Byte 2 = FF(h)</li> <li>Byte 3 = FF(h)</li> </ul> <p>(if command error code = FFFF(h), command is in progress)</p>
	Running "Read functions bits" command
	<p align="center"><u>Command finished</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>Byte 0 = 00(h)</li> <li>Byte 1 = 04(h) (X00A(h) = 1)</li> </ul> <p>Command error code if the command is correctly carried out:</p> <ul style="list-style-type: none"> <li>Byte 2 = 00(h)</li> <li>Byte 3 = 00(h)</li> </ul> <p>OR if an error occurred during the command:</p> <ul style="list-style-type: none"> <li>Byte 2 = 00(h)</li> <li>Byte 3 = 04(h) (X01A(h) = 1)</li> </ul>
<ul style="list-style-type: none"> <li>Wait the end of the command: command echo = 0400(h) (X00A(h) = 1) command error code ≠ FFFF(h) (end of command)</li> </ul>	
<ul style="list-style-type: none"> <li>Deactivate the "Read functions bits" command: Write at the address 00(h) the value <b>0000(h)</b> Byte 0 = 00(h) Byte 1 = 00(h) (Y00A(h) = 0)</li> </ul>	
<ul style="list-style-type: none"> <li>Read the functions bits in the corresponding area depending on the configuration mode of the slave</li> </ul>	



The functions bits are dependents of the program number.  
A program selection has to be realised before executing command.



The master instrument must always set to zero the command bit. If it is not done, the slave instrument will not detect the following command on this bit. It has detection on the rising edge (when the bit state goes from 0 to 1).



## Writing of the function bits

Master	Slave
<ul style="list-style-type: none"> <li>Select the program number on which the functions bits have to be read.</li> <li>Write the functions bits in the corresponding area depending on the configuration mode of the slave.</li> <li>Activate the "Write functions bits" command: Write at the address 00(h), the value <b>1000(h)</b> Byte 0 = 00(h) Byte 1 = 10(h) (Y00C(h) = 1)</li> </ul>	
	<p align="center"><u>Acknowledgement</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>Byte 0 = 00(h)</li> <li>Byte 1 = 10(h) (X00C(h) = 1)</li> </ul> <p>Command error code:</p> <ul style="list-style-type: none"> <li>Byte 2 = FF(h)</li> <li>Byte 3 = FF(h)</li> </ul> <p>(if command error code = FFFF(h), command is in progress)</p>
	Running "Write functions bits" command
	<p align="center"><u>Command finished</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>Byte 0 = 00(h)</li> <li>Byte 1 = 10(h) (X00C(h) = 1)</li> </ul> <p>Command error code if the command is correctly carried out:</p> <ul style="list-style-type: none"> <li>Byte 2 = 00(h)</li> <li>Byte 3 = 00(h)</li> </ul> <p>OR if an error occurred during the command:</p> <ul style="list-style-type: none"> <li>Byte 2 = 00(h)</li> <li>Byte 3 = 10(h) (X01C(h) = 1)</li> </ul>
<ul style="list-style-type: none"> <li>Wait the end of the command: command echo = 1000(h) (X00C(h) = 1) command error code ≠ FFFF(h) (end of command)</li> </ul>	
<ul style="list-style-type: none"> <li>Deactivate the "Write functions bits" command: Write at the address 00(h) the value <b>0000(h)</b> Byte 0 = 00(h) Byte 1 = 00(h) (Y00C(h) = 0)</li> </ul>	



The functions bits are dependents of the program number.  
A program selection has to be realised before executing command.



The master instrument must always set to zero the command bit. If it is not done, the slave instrument will not detect the following command on this bit. It has detection on the rising edge (when the bit state goes from 0 to 1).



## Parameters

### Downloading of the parameters



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

Identifier N°		Meaning	Value	
Dec	Hexa			
01	0001	"FILL TIME" Fill time	0 > 650 seconds	
02	0002	"STAB TIME": Stabilization time	0 > 650 seconds	
03	0003	"TEST TIME" Test time	0 > 650 seconds	
06	0006	"PRE FILL" Pre fill time	0 > 650 seconds	
07	0007	"PRE DUMP" Pre dump time	0 > 650 seconds	
09	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	
17	0011	"Min Vol." Minimum volume reject level (volume test type measure)	0 > 9999	
18	0012	"Max. Vol." Maximum volume reject level (volume test type measure).	0 > 9999	
20	0014	"VOLUME" Part volume.	0 > 9999	
21	0015	"TYPE": Test type	Invalid Leak Desensitized Blockage Operator	0000 1000 2000 3000 4000
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	







Identifier N°		Meaning	Value	
Dec	Hexa			
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 ref. part in recovery	0 > 9999	
66	0042	“Set FILL” Fill instruction value:	- 9999 > 9999	
67	0043	“Set PreFILL” Pre-fill instruction value:	- 9999 > 9999	
68	0044	“SEALED PART” Choice of the sealed component	Standard Large Leak	0000 1000
72	0048	“Drift Unit” Calibration drifts percent.	0 > 100%	
80	0050	“Diff A-Z” Differential auto reset time.	0 > 650 seconds	
102	0066	“BLOW MODE” Type of permanent blowing	Regulator 2 Regulator 1	0000 1000
103	0067	“FILL MODE” Type of fill.	Standard Instruction Ballistic EASY EASY Auto	0000 1000 2000 7000 8000
104	0068	“PreFILL” Type of pre-fill.	Standard Instruction Ballistic EASY EASY Auto	0000 1000 2000 4000 5000
106	006A	“CheckTime” Commutation time of the equalization valve calibration check.	0 > 650 seconds	
107	006B	“% Drift” ATR absorption tolerance.	0 > 100%	
108	006C	“Start” Start value of the transient (ATR).	- 9999 > 9999	
110	006E	“EXT. DUMP” Type of external dump.	Normally close Normally open	0000 1000
111	006F	Reserved.		



Identifier N°		Meaning		
Dec	Hexa			
112	0070	'IN7:' Function attributed to the entry of the special cycles (input 7)	Refer to the "Configurable input values" table at the end of this chapter	
117	0075	"Set Blow" Permanent blowing pressure instruction.	- 9999 > 9999	
118	0076	"REJECT CALC." Original unit for the calculation of rejects in cm <sup>3</sup> / min (Pa, Pa/s...).	Refer to Unit table.	
119	0077	Reserved.		
120	0078	Reserved.		
121	0079	Reserved.		
122	007A	Reserved.		
123	007B	"LANGUAGE" Choice of the language.	Default language 2 <sup>nd</sup> predefined language	0000 1000
124	007C	"Max Value" Reject in calibration check.	0 > 9999	
125	007D	"% Drift" Percentage of the calibration check.	0 > 100%	
126	007E	"Max PreFILL" Maximum pressure value in pre-fill.	- 9999 > 9999	
127	007F	"LeakUnit" Reject unit.	Refer to Unit table	
128	0080	"Leak Rate" Instruction value during a calibration.	0 > 9999	
135	0087	"% of T FAIL" Reject level percent of the auto parameter function	0 > 100%	
138	008A	"FILL REG" Regulator number selection for the fill.	Regulator 1 Regulator 2	0000 1000
139	008B	"PRE FILL REG" Regulator number selection for the pre-fill.	Regulator 1 Regulator 2	0000 1000
140	008C	Reserved.		
141	008D	Reserved.		
142	008E	Reserved.		
143	008F	Reserved.		
144	0090	"OUTPUTS CONFIG." Setup of the outputs (standard or compact).	Standard Compact	0000 1000
148	0094	"FILTER" Filtering.	0 > 650 seconds	
149	0095	"UNITS" Unit type	SI SAE CUSTOM	0000 1000 2000



Identifier N°		Meaning	Value	
Dec	Hexa			
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	
175	00AF	“REGUL. CTRL.” Regulator check during its learning.	Automatic Ext	0000 1000
203	00CB	“ELEC. REG.” Activation or not of the built in electronics regulators.	None Reg 1 Reg 2 ALL Reg	0000 1000 2000 3000
232	00E8	“ATR DRIFT” Drift transient (ATR).	0 > 100%	
233	00E9	“AZ SHORT” Quick auto-zero time.	0 > 650 seconds	
273	0111	“DUMP” Dump time in calibration check mode	0 > 650 seconds	
291	0123	“T.ATR2” Stabilization time for the ATR 2 function	0 > 650 seconds	
295	0127	“DUMP LEVEL” Minimum dump pressure level to reach	- 9999 > 9999	
297	0129	“MAX BLOW” Blowing maximum pressure level	- 9999 > 9999	
298	012A	“MIN BLOW” Blowing minimum pressure level	- 9999 > 9999	
315	013B	Reserved.		
334	014E	Reserved.		
335	014F	Reserved.		
336	0150	Reserved.		
340	0154	“Transient” ATR transient value.	- 9999 > 9999	
349	015D	Reserved.		
353	0161	“Press. UNIT” (configuration/pneumatic menu) General pressure unit	Refer to Unit table.	
354	0162	“LINE P. MIN” Minimum line pressure level	- 9999 > 9999	
355	0163	Reserved.		
356	0164	Reserved.		



Identifier N°		Meaning	Value	
Dec	Hexa			
357	0165	Reserved.		
358	0166	Reserved.		
359	0167	Reserved.		
360	0168	Reserved.		
361	0169	Reserved.		
362	016A	Reserved.		
363	016B	Reserved.		
364	016C	“DISPLAY MODE” Leak display management	xxxx xxx.x xx.xx x.xxx	0000 1000 2000 3000
366	016E	“MODE” (EXT DUMP menu) Dump mode	Continuous Time	0000 1000
367	016F	“Program” (DUMP OFF menu) Program number of the dump of function	0 > 128	
368	0170	“Tolerance A” Tolerance level A for ntest cycle	0 > 100%	
369	0171	“Tolerance B” Tolerance level B for ntest cycle	0 > 100%	
370	0172	Reserved.		
371	0173	“NAME:”(Units menu) CAL unit personalization	CHAR[5]	
372	0174	“BYPASS” Bypass valve mode selection	Pre-Fill + Fill Pre-Fill Fill	0000 1000 2000
373	0175	“% Cut OFF” Cut off function Percent	0 > 100%	
374	0176	“ATF TIME” Divisor time of ATF function	0 > 650 seconds	
375	0177	‘IN8:’ Function attributed to the entry of the special cycles (input 8)	Refer to the “Configurable input values” table at the end of this chapter	
376	0178	‘IN9:’ Function attributed to the entry of the special cycles (input 9)	Refer to the “Configurable input values” table at the end of this chapter	
377	0179	“MEAS. START” Waiting time for starting the measurement in burst test	0 > 650 seconds	
378	017A	“Time Adj” Adjusting fill time (electronic regulator)	0 > 650 seconds	



Identifier N°		Meaning	Value	
Dec	Hexa			
379	017B	“USB:” USB mode (printer or supervision)	Supervision Printer Bar code Auto None	0000 1000 2000 3000 4000
380	017C	Reserved.		
405	0195	Reserved.		
406	0196	Reserved.		
407	0197	Reserved.		
408	0198	Reserved.		
409	0199	Reserved.		
410	019A	Reserved.		
455	01C7	“DROP PRESS.%” Drop Press function Percent	0 > 100%	
456	01C8	“ATM PRESS.” Atmospheric Pressure	900 > 1100	
457	01C9	“TEMP.” Temperature	0 > 800	
458	01CA	“DISP. OPT.” Display Option in flow reject	None Pa Display Ambient Temp. Object Temp. Test check ATR Temp. correction Leak offset learning PATM correction	0000 1000 2000 3000 4000 5000 6000 7000 8000
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	



Identifier N°		Meaning	Value	
Dec	Hexa			
485	01 E5	"EXT. ACCES" Security by external access (Fieldbus/Modbus) Reset value with Modbus: → Writing at address 0xC1E5 Reset value with Fieldbus: → Writing one word with ID = 0xC1E5	Read/Write Read Only No Access	0000 1000 2000
486	01 E6	"OFFSET" Offset Learning	- 9999 > 9999	

### Regulator selection

Regulator selection for fill and pre-fill (word 1, bit n°0 and 1) in the table of the functions bits.

	Fill regulator	Pre-fill regulator
Regulator 1	0	0
Regulator 2	1	1

### Configurable input values

F6 V1.3XX			
Input value	Value code	Input value	Value code
Program Selection	0000	Atr Learning Cycle	17000
P1 Sensor Check (*)	4000	Sd Prt Pass Learn	18000
P1 Reg1 Check (*)	5000	Sd Prt Fail Learn	19000
P2 Sensor Check (*)	6000	Volume Comp.	20000
Leak Sensor Check (*)	7000	Test Check Result	21000
Auto Test (*)	8000	Step By Step	22000
Regul. 2 Adjust	9000	Auto Setup	23000
Regul. 1 Adjust	10000	Atr+Custom Learn.	24000
Part Regulator	11000	Code Reader	25000
Infinite Fill	12000	Auto Vol	26000
Piezo Auto Zero	13000	Test On Caps	27000
Custom Unit Learn	14000	Temp.2 Corr. Learn	30000
Custom Unit Check	15000	Temp.2 Sensor Read	31000
Chck+Lrn Cust. Unit	16000		

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



F6 V2.XXX			
Input value	Value code	Input value	Value code
Program Selection	0000	Chck+Lrn Cust. Unit	24000
Diff Temp. Check (*)	8000	Atr Learning Cycle	25000
Direct P. Check (*)	9000	Sd Prt Pass Learn	26000
P1 Reg1 Check (*)	10000	Sd Prt Fail Learn	27000
Indirect P. Check (*)	11000	Volume Comp.	28000
Leak Sensor Check (*)	12000	Test Check Result	29000
Line P. Sensor Check (*)	13000	Atr+Custom Learn.	30000
Temperature Check (*)	14000	Code Reader	31000
Atm Pressure Check (*)	15000	Auto Vol	32000
Auto Test (*)	16000	Test On Caps	33000
Regul. 2 Adjust	17000	Temp.2 Corr. Learn	36000
Regulator Adj.	18000	Temp.2 Sensor Read	37000
Part. Regulator Adj.	19000	Leak Offset Learn	38000
Infinite Fill	20000	Offset+Vol. Learn	39000
Piezo Auto Zero	21000	N Start	40000
Custom Unit Learn	22000	Sync Test	41000
Custom Unit Check	23000		

(\*) 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 <sup>3</sup> /s
1000	03E8	cm <sup>3</sup> /min
2000	07D0	cm <sup>3</sup> /h
3000	0BB8	mm <sup>3</sup> /s
4000	0FA0	Calibrated Pascal (Pa)
5000	1388	Calibrated Pascal/second (Pa/s)
6000	1770	Pascal (Pa)
7000	1B58	High resolution Pascal (Pa HR)
8000	1F40	Pascal/second (Pa/s)
9000	2328	High resolution Pascal/second (Pa/s HR)
10000	2710	Second (s)
11000	2AF8	Bar
12000	2EE0	KiloPascal (kPa)
13000	32C8	PSI
14000	36B0	Millibar (mbar)
15000	3A98	Mega Pascal (MPa)
16000	3E80	Liter (l)
17000	4268	Calibration check unit
18000	4650	KiloPascal/second (kPa/s)
19000	4A38	Millimeter (mm)
30000	7530	Liter/hour (l/h)
43000	A7F8	D mode Pascal (Pa)
44000	ABE0	Low resolution Pascal (Pa LR)
45000	AFC8	Low resolution Pascal/second (Pa/s LR)
46000	B3B0	Inch <sup>3</sup> /s
47000	B798	Inch <sup>3</sup> /min
48000	BB80	Inch <sup>3</sup> /hour
49000	BF68	Feet <sup>3</sup> /hour
50000	C350	Milliliter/second (mm/s)
51000	C738	Milliliter/minute (mm/min)
52000	CB20	Milliliter/hour (mm/h)
53000	CF08	Liter/minute (l/min)
54000	D2F0	Meter <sup>3</sup> /hour (m <sup>3</sup> /h)
55000	D6D8	Millimeter <sup>3</sup> (mm <sup>3</sup> )
56000	DAC0	Centimeter <sup>3</sup> (cm <sup>3</sup> )
57000	DEA8	Microsecond (μs)
58000	E290	USA cm <sup>3</sup> /s same as the cm <sup>3</sup> /s
59000	E678	USA cm <sup>3</sup> /min same as the cm <sup>3</sup> /min
60000	EA60	USA cm <sup>3</sup> /h same as the cm <sup>3</sup> /h
61000	EE48	Milliliter (ml)





Unit code		Unit
Decimal	Hexadecimal	
62000	F230	Liter (l)
63000	F618	Inch <sup>3</sup>
64000	FA00	Feet <sup>3</sup>
68000	01 09A0	oz(US)/s
69000	01 0D88	oz(US)/mn
70000	01 1170	oz(US)/h
71000	01 1558	oz(UK)/s
72000	01 1940	oz(UK)/mn
73000	01 1D28	oz(UK)/h
74000	01 2110	US gallon
75000	01 24F8	UK gallon
76000	01 28E0	PPM
77000	01 2CC8	PPM HR
78000	01 30B0	Calibrated PPM
80000	01 3880	mmCE
81000	01 3C68	mmCE/s
84000	01 4820	SCCM
92000	01 6760	Points
93000	01 6B48	Feet <sup>3</sup> /s F620
94000	01 6F30	Feet <sup>3</sup> /min F620
95000	01 7318	ACCM MF
96000	01 7700	Inch Mercure (inHg)
99000	01 82B8	Millimeter Mercure (mmHg)
100000	01 86A0	µg H <sub>2</sub> O/min
102000	01 8E70	No unit



## Reading of the parameters

The reading of the parameters is carried out by data exchange in the corresponding area depending on the configuration mode of the slave. Each parameter is identified by one identifier. See identifiers tables.

This table is an example based on the reading of two parameters:

- **Test time** (identifier number 3)
- **Stabilization time** (identifier number 2)

Master	Slave						
<ul style="list-style-type: none"> <li>— Select the program on which parameters has to be read</li> <li>— Write in the parameter area depending on the configuration mode of the slave, the number of parameters followed by their identifiers: On network:  <table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">02</td> <td style="padding: 2px 5px;">00</td> <td style="padding: 2px 5px;">03</td> <td style="padding: 2px 5px;">00</td> <td style="padding: 2px 5px;">02</td> <td style="padding: 2px 5px;">00</td> </tr> </table> </li> </ul> <p>0002(h) 0003(h) 0002(h)            0002(h) = two parameters            0003(h) = test time identifier            0002(h) = stabilization time identifier</p> <ul style="list-style-type: none"> <li>— Activate the “Read parameters” command: Write at the address 00(h), the value <b>0020(h)</b>            Byte 0 = 20(h) (Y005(h) = 1)            Byte 1 = 00(h)</li> </ul>	02	00	03	00	02	00	
02	00	03	00	02	00		
	<p style="text-align: center;"><u>Acknowledgement</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 20(h) (X005(h) = 1)</li> <li>— Byte 1 = 00(h)</li> </ul> <p>Command error code:</p> <ul style="list-style-type: none"> <li>— Byte 2 = FF(h)</li> <li>— Byte 3 = FF(h)</li> </ul> <p>(if command error code = FFFF(h), command is in progress)</p>						
	<p style="text-align: center;">Running “Read parameters” command</p> <p style="text-align: center;"><u>Command finished</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 20(h) (X005(h) = 1)</li> <li>— Byte 1 = 00(h)</li> </ul> <p>Command error code if the command is correctly carried out:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 00(h)</li> <li>— Byte 3 = 00(h)</li> </ul> <p>OR if an error occurred during the command:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 20(h) (X015(h) = 1)</li> <li>— Byte 3 = 00(h)</li> </ul>						
<ul style="list-style-type: none"> <li>— Wait the end of the command: command echo = 0020(h) (X005(h) = 1) command error code ≠ FFFF(h) (end of command)</li> </ul>							
<ul style="list-style-type: none"> <li>— Deactivate the “Read parameters” command: Write at the address 00(h) the value <b>0000(h)</b>            Byte 0 = 00(h) (Y005(h) = 0)            Byte 1 = 00(h)</li> </ul>							



Master	Slave
<p>— Read the parameters in the corresponding area depending on the configuration mode of the slave: 0003(h) 03E8(h) 0000(h) 0002(h) 01F4(h) 0000(h) 0003(h) = test time identifier 00003E8(h) = 1000 =&gt; test time value = 1000/1000 = 1 second 0002(h) = stabilization time identifier 00001F4(h) = 500 =&gt; stabilization time = 500/1000 = 0,5 second</p>	



The master instrument must always set to zero the command bit. If it is not done, the slave instrument will not detect the following command on this bit. It has detection on the rising edge (when the bit state goes from 0 to 1).



## Writing of the parameters

The writing of the parameters is carried out by data exchange in the corresponding area depending on the configuration mode of the slave. Each parameter is identified by one identifier. See identifiers tables.

This table is an example based on the reading of two parameters:

- **Test time** (identifier number 3)
- **Stabilization time** (identifier number 2)

Master	Slave														
<ul style="list-style-type: none"> <li>— Select the program on which the parameters have to be modified</li> <li>— Write in the parameter area depending on the configuration mode of the slave, the number of parameters followed by their identifiers and their wanted value:</li> </ul> <p>On network:</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td>02</td><td>00</td><td>03</td><td>00</td><td>E8</td><td>03</td><td>00</td><td>00</td><td>02</td><td>00</td><td>D0</td><td>07</td><td>00</td><td>00</td> </tr> </table> <p>0002(h) 0003(h) 03E8(h) 0000(h) 0002(h) 07D0(h) 0000(h)</p> <p>0002(h) = two parameters 0003(h) = test time identifier 000003E8(h) = 1000 =&gt; 1 second 0002(h) = stabilization time identifier 000007D0(h) = 2000 =&gt; 2 second</p> <ul style="list-style-type: none"> <li>— Activate the "Write parameters" command:</li> </ul> <p>Write at the address 00(h), the value <b>0040(h)</b> Byte 0 = 20(h) (Y006(h) = 1) Byte 1 = 00(h)</p>	02	00	03	00	E8	03	00	00	02	00	D0	07	00	00	<p style="text-align: center;"><u>Acknowledgement</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 40(h) (X006(h) = 1)</li> <li>— Byte 1 = 00(h)</li> </ul> <p>Command error code:</p> <ul style="list-style-type: none"> <li>— Byte 2 = FF(h)</li> <li>— Byte 3 = FF(h)</li> </ul> <p>(if command error code = FFFF(h), command is in progress)</p> <p>Running "Write parameters" command</p> <p style="text-align: center;"><u>Command finished</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 40(h) (X006(h) = 1)</li> <li>— Byte 1 = 00(h)</li> </ul> <p>Command error code if the command is correctly carried out:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 00(h)</li> <li>— Byte 3 = 00(h)</li> </ul> <p>OR if an error occurred during the command:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 40(h) (X016(h) = 1)</li> <li>— Byte 3 = 00(h)</li> </ul>
02	00	03	00	E8	03	00	00	02	00	D0	07	00	00		



Master	Slave
<ul style="list-style-type: none"><li>— Wait the end of the command: command echo = 0040(h) (X006(h) = 1) command error code ≠ FFFF(h) (end of command)</li></ul>	
<ul style="list-style-type: none"><li>— Deactivate the “Write parameters” command: Write at the address 00(h) the value <b>0000(h)</b> Byte 0 = 00(h) (Y006(h) = 0) Byte 1 = 00(h)</li></ul>	



The master instrument must always set to zero the command bit. If it is not done, the slave instrument will not detect the following command on this bit. It has detection on the rising edge (when the bit state goes from 0 to 1).



## Reading of the program name

77 / 95

Master	Slave
<ul style="list-style-type: none"> <li>— Select the program whose name you want to read</li> <li>— Activate the “Read program name” command: Write at the address 00(h), the value <b>2000(h)</b> Byte 0 = 00(h) Byte 1 = 20(h) (Y00D(h) = 1)</li> </ul>	
	<p style="text-align: center;"><u>Acknowledgement</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 00(h)</li> <li>— Byte 1 = 20(h) (X00D(h) = 1)</li> </ul> <p>Command error code:</p> <ul style="list-style-type: none"> <li>— Byte 2 = FF(h)</li> <li>— Byte 3 = FF(h)</li> </ul> <p>(if command error code = FFFF(h), command is in progress)</p>
	<p>Running “Read program name” command</p> <p style="text-align: center;"><u>Command finished</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 00(h)</li> <li>— Byte 1 = 20(h) (X00D(h) = 1)</li> </ul> <p>Command error code if the command is correctly carried out:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 00(h)</li> <li>— Byte 3 = 00(h)</li> </ul> <p>OR if an error occurred during the command:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 00(h)</li> <li>— Byte 3 = 20(h) (X01D(h) = 1)</li> </ul>
<ul style="list-style-type: none"> <li>— Wait the end of the command: command echo = 2000(h) (X006(h) = 1) command error code ≠ FFFF(h) (end of command)</li> </ul>	
<ul style="list-style-type: none"> <li>— Deactivate the “Read program name” command: Write at the address 00(h) the value <b>0000(h)</b> Byte 0 = 00(h) Byte 1 = 00(h) (Y00D(h) = 0)</li> </ul>	
<ul style="list-style-type: none"> <li>— Read the program name of 12 characters/bytes maximum in the corresponding area depending on the configuration mode of the slave:</li> </ul>	



The program name is dependant of the program number in edition, a program selection has to be realized.



The master instrument must always set to zero the command bit. If it is not done, the slave instrument will not detect the following command on this bit. It has detection on the rising edge (when the bit state goes from 0 to 1).



## Writing of the program name

Master	Slave
<ul style="list-style-type: none"> <li>— Select the program whose name you want to modify</li> <li>— Write the program name of 12 characters/bytes maximum in the corresponding area depending on the configuration mode of the slave.</li> <li>— Activate the “Write program name” command: Write at the address 00(h), the value <b>4000(h)</b> Byte 0 = 00(h) Byte 1 = 40(h) (Y00E(h) = 1)</li> </ul>	
	<p style="text-align: center;"><u>Acknowledgement</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 00(h)</li> <li>— Byte 1 = 40(h) (X00E(h) = 1)</li> </ul> <p>Command error code:</p> <ul style="list-style-type: none"> <li>— Byte 2 = FF(h)</li> <li>— Byte 3 = FF(h)</li> </ul> <p>(if command error code = FFFF(h), command is in progress)</p>
	<p style="text-align: center;">Running “Write program name” command</p>
	<p style="text-align: center;"><u>Command finished</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 00(h)</li> <li>— Byte 1 = 40(h) (X00E(h) = 1)</li> </ul> <p>Command error code if the command is correctly carried out:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 00(h)</li> <li>— Byte 3 = 00(h)</li> </ul> <p>OR if an error occurred during the command:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 00(h)</li> <li>— Byte 3 = 40(h) (X01E(h) = 1)</li> </ul>
<ul style="list-style-type: none"> <li>— Wait the end of the command: command echo = 4000(h) (X00E(h) = 1) command error code ≠ FFFF(h) (end of command)</li> </ul>	
<ul style="list-style-type: none"> <li>— Deactivate the “Write program name” command: Write at the address 00(h) the value <b>0000(h)</b> Byte 0 = 00(h) Byte 1 = 00(h) (Y00E(h) = 0)</li> </ul>	

**i** The program name is dependant of the program number in edition, a program selection has to be realized.

**!** The master instrument must always set to zero the command bit. If it is not done, the slave instrument will not detect the following command on this bit. It has detection on the rising edge (when the bit state goes from 0 to 1).



## CYCLE

### Standard command cycle

#### Start cycle command on the ATEQ device

79 / 95

Master	Slave
<ul style="list-style-type: none"> <li>— Select the program you want to start</li> <li>— Activate the “Start” command: Write at the address 00(h), the value <b>0002(h)</b> Byte 0 = 02(h) (Y001(h) = 1) Byte 1 = 00(h)</li> </ul>	
	<p style="text-align: center;"><u>Acknowledgement</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 02(h) (X001(h) = 1)</li> <li>— Byte 1 = 00(h)</li> </ul> <p>Command error code:</p> <ul style="list-style-type: none"> <li>— Byte 2 = FF(h)</li> <li>— Byte 3 = FF(h)</li> </ul> <p>(if command error code = FFFF(h), command is in progress)</p>
	<p>Running “Start” command</p> <p style="text-align: center;"><u>Command finished</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 02(h) (X001(h) = 1)</li> <li>— Byte 1 = 00(h)</li> </ul> <p>Command error code if the command is correctly carried out:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 00(h)</li> <li>— Byte 3 = 00(h)</li> </ul> <p>OR if an error occurred during the command:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 02(h) (X011(h) = 1)</li> <li>— Byte 3 = 00(h)</li> </ul>
<ul style="list-style-type: none"> <li>— Wait the end of the command: command echo = 0002(h) (X001(h) = 1) command error code ≠ FFFF(h) (end of command)</li> </ul>	
<ul style="list-style-type: none"> <li>— Deactivate the “Start” command: Write at the address 00(h) the value <b>0000(h)</b> Byte 0 = 00(h) (Y001(h) = 0) Byte 1 = 00(h)</li> </ul>	

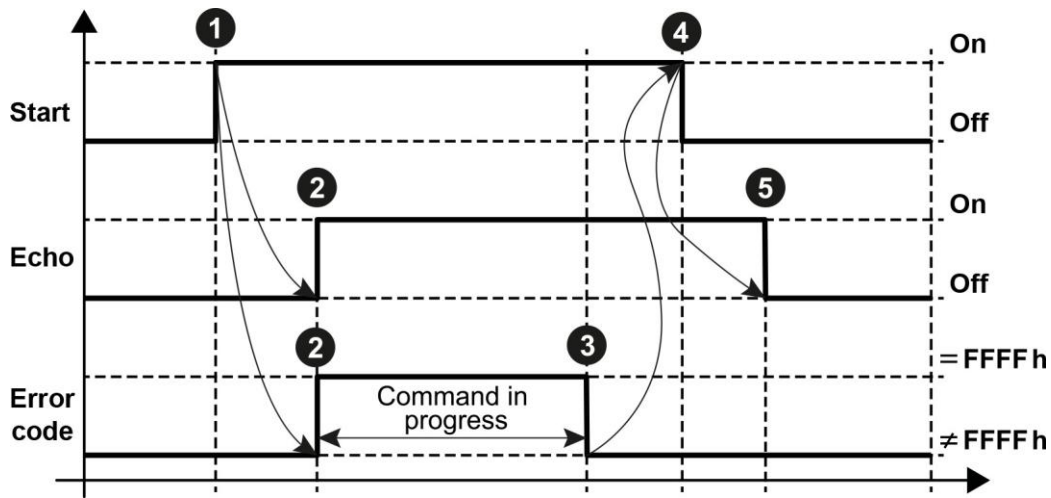


The master instrument must always set to zero the command bit. If it is not done, the slave instrument will not detect the following command on this bit. It has detection on the rising edge (when the bit state goes from 0 to 1).





## Start command diagram



1	Start command = On
2	Acknowledge by ATEQ = (Echo command = On) and (Error code command = FFFFh)
3	Wait end of Start command = (Echo command = On) and (Error code command ≠ FFFFh)
4	Start command = Off
5	Acknowledge by ATEQ = (Echo command = Off) and (Error code command ≠ FFFFh)



The **Echo** command is a copy of the **Start** command. The **Start** signal must be maintained (ON) till the end of the **Start** command condition is reached.



## Reset command on the ATEQ device

81 / 95

Master	Slave
<ul style="list-style-type: none"><li>— Activate the “Reset” command:</li><li>— Write at the address 00(h), the value <b>0001(h)</b> Byte 0 = 01(h) (Y000(h) = 1) Byte 1 = 00(h)</li></ul>	
	<p style="text-align: center;"><u>Acknowledgement</u></p> <p>Command echo:</p> <ul style="list-style-type: none"><li>— Byte 0 = 01(h) (X000(h) = 1)</li><li>— Byte 1 = 00(h)</li></ul> <p>Command error code:</p> <ul style="list-style-type: none"><li>— Byte 2 = FF(h)</li><li>— Byte 3 = FF(h)</li></ul> <p>(if command error code = FFFF(h), command is in progress)</p>
	<p>Running “Reset” command</p> <p style="text-align: center;"><u>Command finished</u></p> <p>Command echo:</p> <ul style="list-style-type: none"><li>— Byte 0 = 01(h) (X000(h) = 1)</li><li>— Byte 1 = 00(h)</li></ul> <p>Command error code if the command is correctly carried out:</p> <ul style="list-style-type: none"><li>— Byte 2 = 00(h)</li><li>— Byte 3 = 00(h)</li></ul> <p>OR if an error occurred during the command:</p> <ul style="list-style-type: none"><li>— Byte 2 = 01(h) (X010(h) = 1)</li><li>— Byte 3 = 00(h)</li></ul>
<ul style="list-style-type: none"><li>— Wait the end of the command: command echo = 0001(h) (X000(h) = 1) command error code ≠ FFFF(h) (end of command)</li></ul>	
<ul style="list-style-type: none"><li>— Deactivate the “Reset” command: Write at the address 00(h) the value <b>0000(h)</b> Byte 0 = 00(h) (Y000(h) = 0) Byte 1 = 00(h)</li></ul>	



The master instrument must always set to zero the command bit. If it is not done, the slave instrument will not detect the following command on this bit. It has detection on the rising edge (when the bit state goes from 0 to 1).



## Special cycles

### Special cycle table

Write the identifier number of the wanted special cycle at the address 04(h) and its instruction if necessary.

Y040(h) → Y047(h) = identifier number of the special cycle

Y048(h) → Y04F(h) = instruction for the special cycle

Numb	Special cycle	Numb	Special cycle
1	ATR Learning Cycle	17	Sd Prt FAIL Learn
2	Test Check Result	18	Direct P. Check (**)
3	AUTO TEST (**)	19	Leak Sensor Check (**)
4	Custom Unit Learn	20	Reserved
5	Custom Unit Check	21	Reserved
6	ATR+Custom Learn	22	Reserved
7	Piezo auto zero Reg 1	23	No special cycle
8	Piezo auto zero Reg 2	24	Reserved
9	Regul. 2 adjust	25	Line P. Sensor Check (*) (**)
10	Regulator Adj	26	No special cycle
11	Infinite fill	27	Reserved
12	Volume Comp	28	Reserved
13	Auto Vol	29	Temperature check (*) (**)
14	No special cycle	30	Atm Pressure Check (*) (**)
15	No special cycle	31	No special cycle
16	Sd Prt PASS Learn		

To activate a special cycle, you must send a **Start** command (Y001(h)) and a **Start special cycle** command (Y002(h)).

(\*) For version ≥ 1.400 only.

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



## Auto-zero on the ATEQ device

Master	Slave
<ul style="list-style-type: none"> <li>— Select the program on which you want to make the auto zero</li> <li>— Write at the address 04(h) the identifier number of the special cycle for an auto zero</li> <li>— Activate the “Start” and the “Start special cycle” commands:</li> </ul> <p>Write at the address 00(h), the value <b>0006(h)</b>            Byte 0 = 06(h) (Y001(h) = 1 and Y002(h) = 1)            Byte 1 = 00(h)</p>	
	<p style="text-align: center;"><u>Acknowledgement</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 06(h) (X001(h) = 1 and X002(h) = 1)</li> <li>— Byte 1 = 00(h)</li> </ul> <p>Command error code:</p> <ul style="list-style-type: none"> <li>— Byte 2 = FF(h)</li> <li>— Byte 3 = FF(h)</li> </ul> <p>(if command error code = FFFF(h), command is in progress)</p>
	<p style="text-align: center;">Running “Start” and “Start special cycle” commands</p> <p style="text-align: center;"><u>Command finished</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 06(h) (X001(h) = 1 and X002(h) = 1)</li> <li>— Byte 1 = 00(h)</li> </ul> <p>Command error code if the command is correctly carried out:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 00(h)</li> <li>— Byte 3 = 00(h)</li> </ul> <p>OR if an error occurred during the command:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 06(h) (X011(h) = 1 and X012(h) = 1)</li> <li>— Byte 3 = 00(h)</li> </ul>
<ul style="list-style-type: none"> <li>— Wait the end of the command: command echo = 0006(h) (X001(h) = 1 and X002(h) = 1) command error code ≠ FFFF(h) (end of command)</li> </ul>	
<ul style="list-style-type: none"> <li>— Deactivate the “Start” and “Start special cycle” commands:</li> </ul> <p>Write at the address 00(h) the value <b>0000(h)</b>            Byte 0 = 00(h) (Y001(h) = 0 and Y002(h) = 0)            Byte 1 = 00(h)</p>	



The master instrument must always set to zero the command bit. If it is not done, the slave instrument will not detect the following command on this bit. It has detection on the rising edge (when the bit state goes from 0 to 1).



## RESULTS

### FIFO results

#### FIFO list results structure

At the end of each cycle, a result is stored as an array of 40 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 leak). The results are in the memory of the instrument. To obtain them, it is necessary to carry out a “Read FIFO results” request.

#### Standard results

The standard result structure contains **36 bytes**.

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	Leak low section word.	Long	4	x1000
10	Leak high section word.			
11	Leak unit code low part word (refer to. Units table).	Long	4	x1000
12	Leak unit code high part word (refer to. Units table).			
13	Pressure piezo 2 low part word.	Long	4	x1000
14	Pressure piezo 2 high part word.			
15	Pressure piezo 2 unit code low part word (refer to units table).	Long	4	x1000
16	Pressure piezo 2 unit code high part word (refer to units table).			
17	Test check result low part word.	Long	4	x1000
18	Test check result high part word.			



## Extended results

The extended result structure contains **80 bytes**.

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	Leak low section word.	Long	4	x1000
10	Leak high section word.			
11	Leak unit code low part word (refer to. Units table).	Long	4	x1000
12	Leak unit code high part word (refer to. Units table).			
13	Pressure piezo 2 low part word.	Long	4	x1000
14	Pressure piezo 2 high part word.			
15	Pressure piezo 2 unit code low part word (refer to units table).	Long	4	x1000
16	Pressure piezo 2 unit code high part word (refer to units table).			
17	Test check result low part word.	Long	4	x1000
18	Test check result high part word.			
19	Test check result unit code low part word (refer to units table).	Long	4	x1000
20	Test check result unit code high part word (refer to units table).			
21	Large Leak low part word.	Long	4	x1000
22	Large Leak high part word.			
23	Large Leak unit code low part word (refer to units table).	Long	4	x1000
24	Large Leak unit code high part word (refer to units table).			
<b>V2.xxx Only</b>				
25	Pa – Pa/s Leak result low part word	Long	4	x1000
26	Pa – Pa/s Leak result high part word			
27 - 36	<i>Unused</i>			
37	Atmospheric pressure in hPa low part word	Long	4	x1000
38	Atmospheric pressure in hPa high part word			
39	Temperature in °C low part word	Long	4	x1000
40	Temperature in °C high part word			



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	Pre-dump.
2	0002	Sealed component fill.
3	0003	Sealed component stabilization.
4	0004	Fill.
5	0005	Stabilization.
6	0006	Test.
7	0007	Dump.
65535	FFFF	No steps 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).
8	0008	ATR error.
9	0009	ATR drift.
10	000A	CAL error.
11	000B	Volume too small (sealed component).
12	000C	Volume too large (sealed component).
14	000E	Equalization valve switching error.
43	002B	Pressure too high.
44	002C	Pressure too low.
45	002D	Piezo sensor out of order.
46	002E	Dump error.
47	002F	CAL drift error.
48	0030	Calibration check error.
49	0031	Leak in calibration check too high.
50	0032	Leak in calibration check too low.
51	0033	Sealed component learning error.
64	0040	Piezo sensor 2 out of order.
65	0041	Pressure Piezo 2 too high.
66	0042	Pressure Piezo 2 too low.
68	0044	Pressure Piezo 2 switched alarm (test pressure too high).
69	0045	Pressure Piezo 2 switch (test pressure too small).
72	0048	Learning Electrical Regulator Default.





## Cycle results reading (last 8 results in FIFO)

Master	Slave
<p>— Read the number of available results in the FIFO at the address 04(h): 04(h) = 0000(h) → no results 04(h) &gt; 0000(h) → results available</p> <p>— Activate the “Read FIFO results” command: Write at the address 00(h), the value <b>0010(h)</b> Byte 0 = 10(h) (Y004(h) = 1) Byte 1 = 00(h)</p>	
	<p><u>Acknowledgement</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 10(h) (X004(h) = 1)</li> <li>— Byte 1 = 00(h)</li> </ul> <p>Command error code:</p> <ul style="list-style-type: none"> <li>— Byte 2 = FF(h)</li> <li>— Byte 3 = FF(h)</li> </ul> <p>(if command error code = FFFF(h), command is in progress)</p>
	Running “Read FIFO results” command
	<p><u>Command finished</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 10(h) (X004(h) = 1)</li> <li>— Byte 1 = 00(h)</li> </ul> <p>Command error code if the command is correctly carried out:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 00(h)</li> <li>— Byte 3 = 00(h)</li> </ul> <p>OR if an error occurred during the command:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 10(h) (X014(h) = 1)</li> <li>— Byte 3 = 00(h)</li> </ul>
<p>— Wait the end of the command: command echo = 0010(h) (X004(h) = 1) command error code ≠ FFFF(h) (end of command)</p>	
<p>— Deactivate the “Read FIFO results” command: Write at the address 00(h) the value <b>0000(h)</b> Byte 0 = 00(h) (Y004(h) = 0) Byte 1 = 00(h)</p>	



The master instrument must always set to zero the command bit. If it is not done, the slave instrument will not detect the following command on this bit. It has detection on the rising edge (when the bit state goes from 0 to 1).



## Reset FIFO results

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

Master	Slave
<ul style="list-style-type: none"> <li>— Activate the “Reset FIFO results” command: Write at the address 00(h), the value <b>0080(h)</b> Byte 0 = 80(h) (Y007(h) = 1) Byte 1 = 00(h)</li> </ul>	
	<p style="text-align: center;"><u>Acknowledgement</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 80(h) (X007(h) = 1)</li> <li>— Byte 1 = 00(h)</li> </ul> <p>Command error code:</p> <ul style="list-style-type: none"> <li>— Byte 2 = FF(h)</li> <li>— Byte 3 = FF(h)</li> </ul> <p>(if command error code = FFFF(h), command is in progress)</p>
	<p>Running “Reset FIFO results” command</p> <p style="text-align: center;"><u>Command finished</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 80(h) (X007(h) = 1)</li> <li>— Byte 1 = 00(h)</li> </ul> <p>Command error code if the command is correctly carried out:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 00(h)</li> <li>— Byte 3 = 00(h)</li> </ul> <p>OR if an error occurred during the command:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 80(h) (X017(h) = 1)</li> <li>— Byte 3 = 00(h)</li> </ul>
<ul style="list-style-type: none"> <li>— Wait the end of the command: command echo = 0080(h) (X007(h) = 1) command error code ≠ FFFF(h) (end of command)</li> </ul>	
<ul style="list-style-type: none"> <li>— Deactivate the “Reset FIFO results” command: Write at the address 00(h) the value <b>0000(h)</b> Byte 0 = 00(h) (Y007(h) = 0) Byte 1 = 00(h)</li> </ul>	



The master instrument must always set to zero the command bit. If it is not done, the slave instrument will not detect the following command on this bit. It has detection on the rising edge (when the bit state goes from 0 to 1).



## Last results

### Last results structure

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 leak).

The last result is in the memory of the instrument. To obtain them, it is necessary to carry out a “Read last results” request.

### Standard results

The standard result structure contains **36 bytes**.

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	Leak low section word.	Long	4	x1000
10	Leak high section word.			
11	Leak unit code low part word (refer to. Units table).	Long	4	x1000
12	Leak unit code high part word (refer to. Units table).			
13	Pressure piezo 2 low part word.	Long	4	x1000
14	Pressure piezo 2 high part word.			
15	Pressure piezo 2 unit code low part word (refer to units table).	Long	4	x1000
16	Pressure piezo 2 unit code high part word (refer to units table).			
17	Test check result low part word.	Long	4	x1000
18	Test check result high part word.			



## Extended results

The extended result structure contains **80 bytes**.

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	Leak low section word.	Long	4	x1000
10	Leak high section word.			
11	Leak unit code low part word (refer to. Units table).	Long	4	x1000
12	Leak unit code high part word (refer to. Units table).			
13	Pressure piezo 2 low part word.	Long	4	x1000
14	Pressure piezo 2 high part word.			
15	Pressure piezo 2 unit code low part word (refer to units table).	Long	4	x1000
16	Pressure piezo 2 unit code high part word (refer to units table).			
17	Test check result low part word.	Long	4	x1000
18	Test check result high part word.			
19	Test check result unit code low part word (refer to units table).	Long	4	x1000
20	Test check result unit code high part word (refer to units table).			
21	Large Leak low part word.	Long	4	x1000
22	Large Leak high part word.			
23	Large Leak unit code low part word (refer to units table).	Long	4	x1000
24	Large Leak unit code high part word (refer to units table).			
V2.xxx Only				
25	Pa – Pa/s Leak result low part word	Long	4	x1000
26	Pa – Pa/s Leak result high part word			
27 - 36	<i>Unused</i>			
37	Atmospheric pressure in hPa low part word	Long	4	x1000
38	Atmospheric pressure in hPa high part word			
39	Temperature in °C low part word	Long	4	x1000
40	Temperature in °C high part word			



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

Master	Slave
<ul style="list-style-type: none"> <li>— Activate the “Read Last result” command: Write at the address 00(h), the value <b>8000(h)</b> Byte 0 = 00(h) Byte 1 = 80(h) (Y00F(h) = 1)</li> </ul>	
	<p><u>Acknowledgement</u></p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 00(h)</li> <li>— Byte 1 = 80(h) (X00F(h) = 1)</li> </ul> <p>Command error code:</p> <ul style="list-style-type: none"> <li>— Byte 2 = FF(h)</li> <li>— Byte 3 = FF(h)</li> </ul> <p>(if command error code = FFFF(h), command is in progress)</p>
	<p>Running “Read Last result” command</p> <p>Command finished</p> <p>Command echo:</p> <ul style="list-style-type: none"> <li>— Byte 0 = 00(h)</li> <li>— Byte 1 = 80(h) (X00F(h) = 1)</li> </ul> <p>Command error code if the command is correctly carried out:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 00(h)</li> <li>— Byte 3 = 00(h)</li> </ul> <p>OR if an error occurred during the command:</p> <ul style="list-style-type: none"> <li>— Byte 2 = 00(h)</li> <li>— Byte 3 = 80(h) (X01F(h) = 1)</li> </ul>
<ul style="list-style-type: none"> <li>— Wait the end of the command: command echo = 8000(h) (X00F(h) = 1) command error code ≠ FFFF(h) (end of command)</li> </ul>	
<ul style="list-style-type: none"> <li>— Deactivate the “Read Last result” command: Write at the address 00(h) the value <b>0000(h)</b> Byte 0 = 00(h) Byte 1 = 00(h) (Y00F(h) = 0)</li> </ul>	



The master instrument must always set to zero the command bit. If it is not done, the slave instrument will not detect the following command on this bit. It has detection on the rising edge (when the bit state goes from 0 to 1).



## Real time

### Status and real time measures

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	Meaning	Type	Bytes	Coeff
1	Program number.	Word	2	
2	Number of results waiting in the results FIFO memory.	Word	2	
3	Test type.	Word	2	
4	Status: bit 0 = 1: pass part. Bit 1 = 1: fail test part. Bit 2 = 1: fail reference part. 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	
4	Bit 6 = 1: recoverable part. Bit 7 = 1: CAL error or drift. Bit 8 = 1: Calibration check error. Bit 9 = 1: ATR error or drift. Bits 10 / 11 / 12 / 13 / 14 = 1: <i>Unused</i> . 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	Step code (refer to steps table).	Word	2
6	Low pressure section word.	Long	4	x1000
7	High pressure section word.			
8	Pressure unit code low part word (see units table).	Long	4	x1000
9	Pressure unit code high part word (see units table).			
10	Measurement result low section word (Leak, Volume...).	Long	4	x1000
11	Measurement result high section word (Leak, Volume...).			
12	Measurement result unit code low part word (Leak, Volume... Refer to units table).	Long	4	x1000
13	Measurement result unit code high part word (Leak, Volume... Refer to units table).			



### Examples

#### Pressure value = 207

Pressure: Words 6 and 7

On network:

98 28 03 00

00032898h → 207000(d)/1000 → 207

#### Leak value = -0.108

Leak: Words 10 and 11

On network:

94 FF FF FF

FFFFFF94h → -108(d)/1000 → -0.108





## SLMP PARAMETERS

### SLMP Command list

- SearchNode
- SetIpAddress
- CompareDeviceInfo
- GetParam
- SetParam
- StartSetParam
- EndSetParam
- CancelSetParam
- ReadStatus
- GetCommSetting
- ReadStatus2

### SLMP Error Code

Define	Code (hexa)
SLMP_ERR_COMMAND_SUBCOMMAND	0xC059
SLMP_ERR_WRONG_DATA	0xC05C
SLMP_ERR_DATA_LENGTH	0xC061
SLMP_ERR_UNDER_EXECUTION	0xCCE0
SLMP_ERR_REQ_DATA_SIZE	0xCCE1
SLMP_ERR_RES_DATA_SIZE	0xCCE2
SLMP_ERR_NO_EXIST_SERVER_NO	0xCF10
SLMP_ERR_CAN_NOT_COMMUNICATION_SETTING	0xCF20
SLMP_ERR_NO_EXIST_PARAM_ID	0xCF30
SLMP_ERR_CAN_NOT_PARAMETER_SET	0xCF31

### ATEQ parameter SET/GET error codes

Define	Code (hexa)
USER_ERR_PARAMETER_NUM_EXCEED	0x401
USER_ERR_DATA_SIZE_EXCEED	0x402
USER_ERR_PARAMETER_SET	0x403
USER_ERR_PARAMETER_GET	0x404



## SLMP ID for parameter SET/GET

Functionalities	R/W	ID	Length	Format	Comment
Parameters	R/W	001 - 499	4 bytes	Signed long, Coeff * 1000	"Parameters" section
Extended menu bits	R/W	500	20 bytes	Array of bytes	"General configuration" section
Select program	W	501	2 bytes	Unsigned short	
Functions bits	R/W	502	20 bytes	Array of bytes	"Function" section
Program Name	R/W	503	12 bytes	ASCII Array of 12 char	eg : « 123456789AB »
Number of FIFO results	R	504	2 bytes	short	
Result export frame	R	505	128 bytes max	ASCII frame	
Statistic cycle counter	R	506	16 bytes	(4 * DWORD)	Per program

## Export frame

The character "→" corresponds to a tab HT (09(h)).

The character "↵" corresponds to a carriage return CR (0D(h)).

1	→	2	→	3	→	4	→	5	→	6	→	7	→	8	8'	9	→	10	↵
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----	---	---	----	---

#	Item
1	Personalization
2	Program number
3	Test result message
4	Test numeric value
5	Test unit
6	Pressure numeric value
7	Pressure unit
8	Alarm message
8'	Bar code
9	Date
10	Hour

## Structure of statistic

Words	Meaning	Type	Bytes
1	Total number of parts	Unsigned long	4
2			
3	Number of good parts	Unsigned long	4
4			
5	Number of bad parts	Unsigned long	4
6			
7	Number of alarms	Unsigned long	4
8			