Sommario

DEVICE HMI

1. Introduction

The internal HMI device is a tool that resides in the CPU that allows to manage the shown of alphanumeric display with 7 segments present on MicroQMove series D9 and series D2 instruments.
Creates a simple interface in few time and with just a few lines QCL is using a collection of prebuilt functionality that the programmer needs only to configure, either using a range of specializations which are useful to solve those cases where the features configured fail to meet the needs of your application.
Provides 8 recursive numerical views with scrool up/down keys. Each view has a programmable letter. By setting zero in the programmable letter you have an extra digit.

1.1 Installation

1.1.1 Device declaration in the configuration file (.CNF)

In the configuration file (.CNF), the BUS section must be declared so that you have the hardware resources needed to implement the HMI device.
In the INTDEVICE section of the .CNF file must be to add the following definition:

```
;---------------------------------
; Internal devices declaration
;---------------------------------
INTDEVICE
..
<device_name> HMI TCamp
```

where:

<device_name>	The name assigned to the device.
HMI	Keyword that identifies the device display management.
TCamp	Sample time device (1÷250 ms).

1.1.1.1 Example

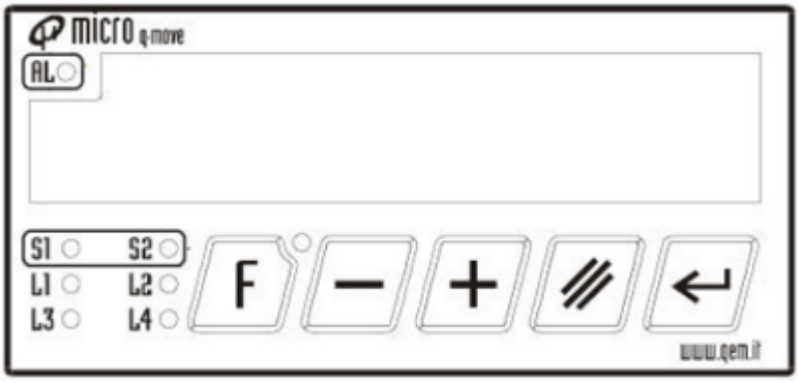
```
;---------------------------------
; Internal device declaration
;---------------------------------
INTDEVICE
hmDisplay    HMI   1
```

1.2 Operation

1.2.1 D2 series

The HMI device manages:

- an interface consisting of a 7-digit seven-segment line;
- an alphanumeric keyboard.



1.2.1.1 Constant keys assignment

Key	Constant code
F	16
-	32

Key	Constant code
+	4
CLEAR	8
ENTER	1

1.2.1.2 Constant led assignment

Led	Constant code
L1	2
L2	4
L3	8
L4	16
F	512
AL	1

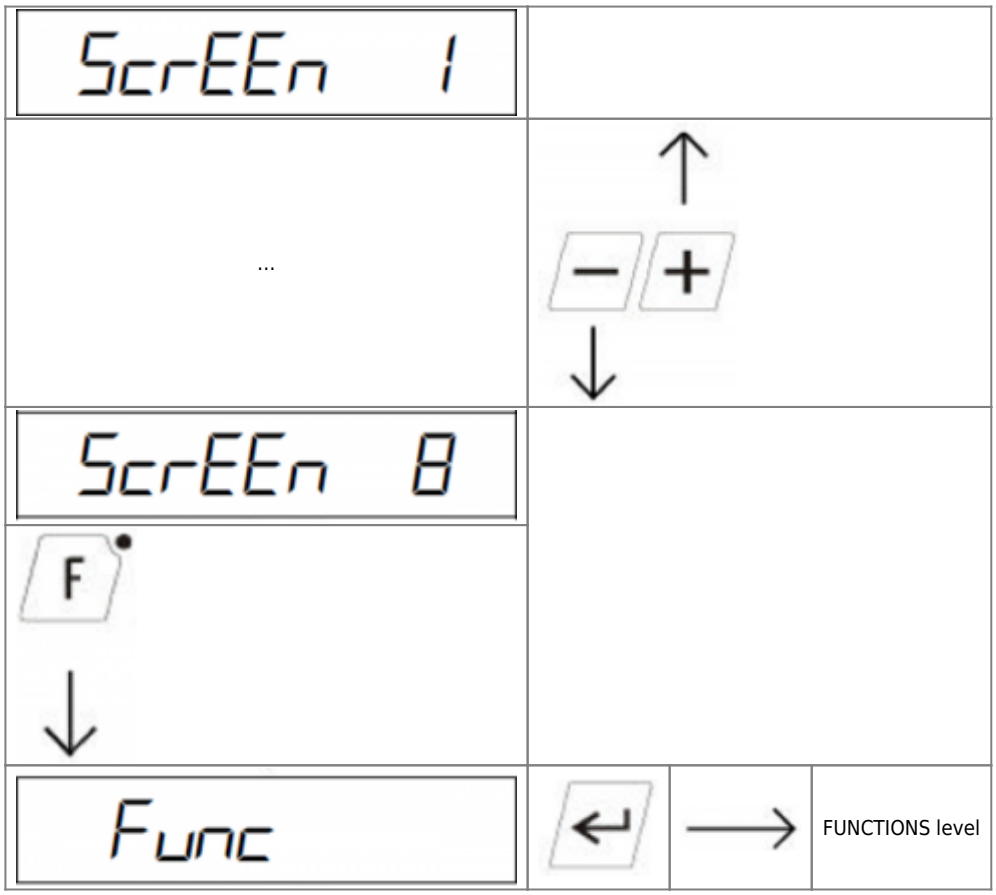
1.2.1.3 Navigation diagram between functions

The HMI device by default shown the SCREEN level (is the recursive view management).

To move to the other functions, you must scroll through the various levels with F key (must be enabled by setting to 1 the bit 14 and the bit 15 of the “enable” parameter).

Confirm the level reached by pressing the ENTER key the following navigation diagram.

1.2.1.4 SCREEN level



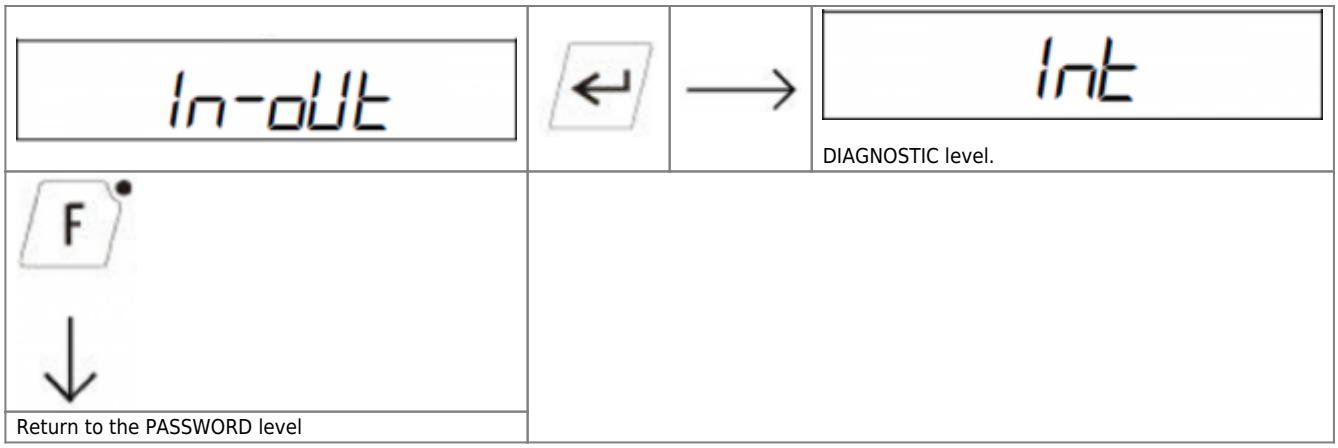
F ↓			
MEnu	↵	→	MENU level
F ↓			
MAn	↵	→	MANUAL level
F ↓			
Return to the FUNCTIONS level			

To exit by levels and go back to the recursive views press F for 2 seconds about.

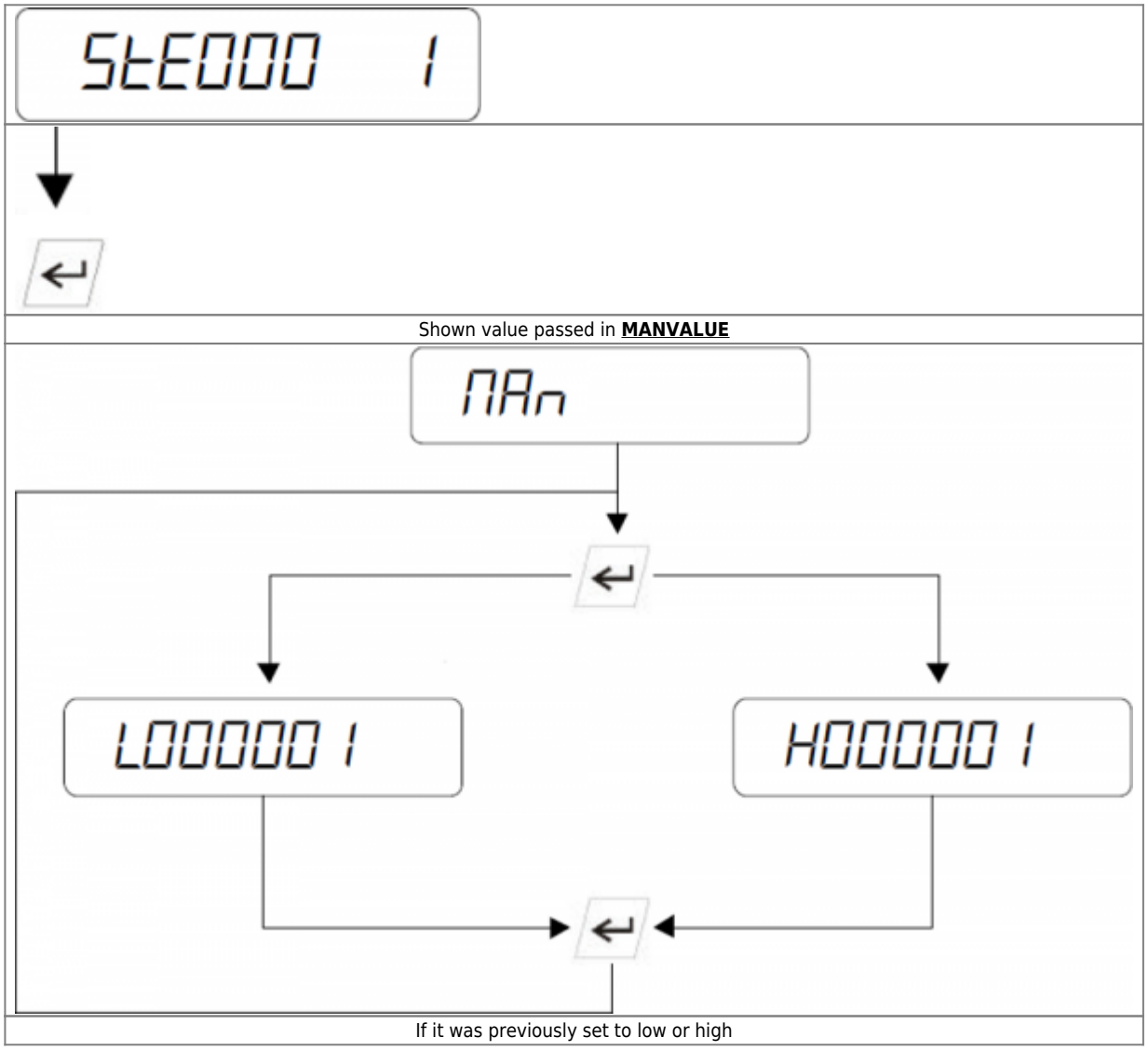
1.2.1.5 FUNCTIONS level

PASS	↵	→	H . . .000 → 100, 123, XXX PASSWORD level
F ↓			
ProG E	↵	→	PrE0001 Execution program setup

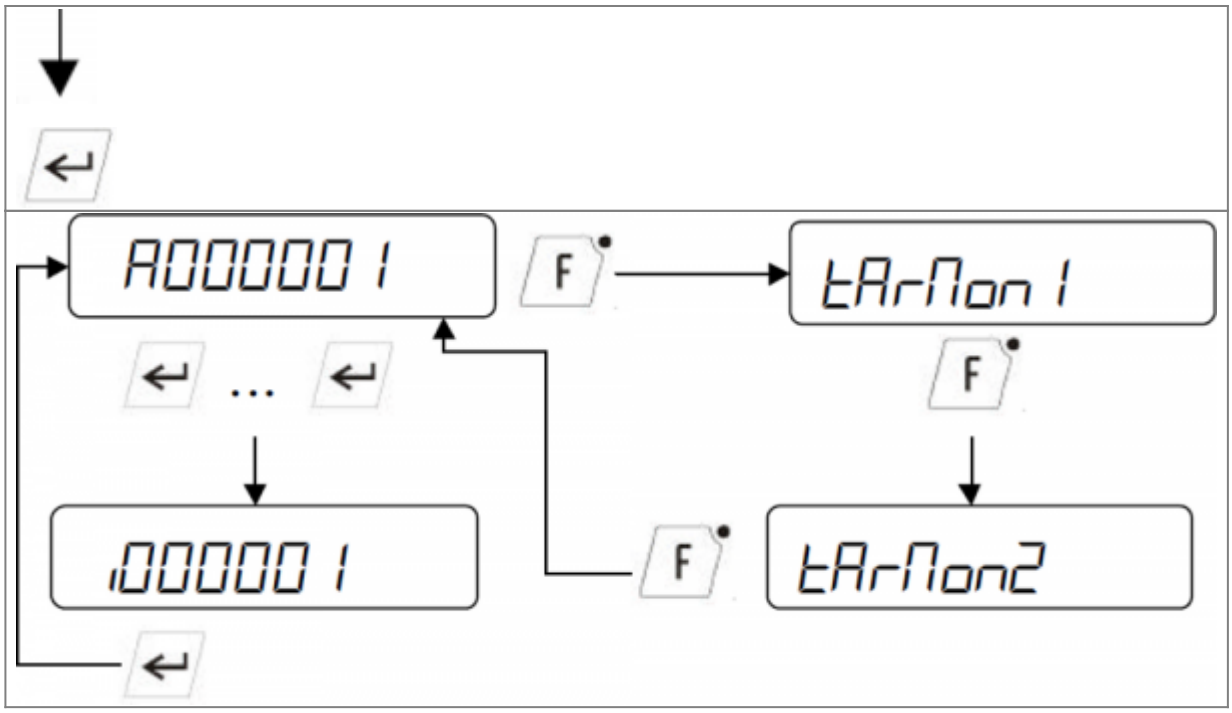
F ↓			
StEP E	↵	→	StE0001 Execution step setting
F ↓			
	↵	→	 Generic long setting
F ↓			
PAr03	↵	→	C000001 PAR03 setting
F ↓			
PAr04	↵	→	d000001 PAR04 setting
F ↓			



1.2.1.6 MANUAL level

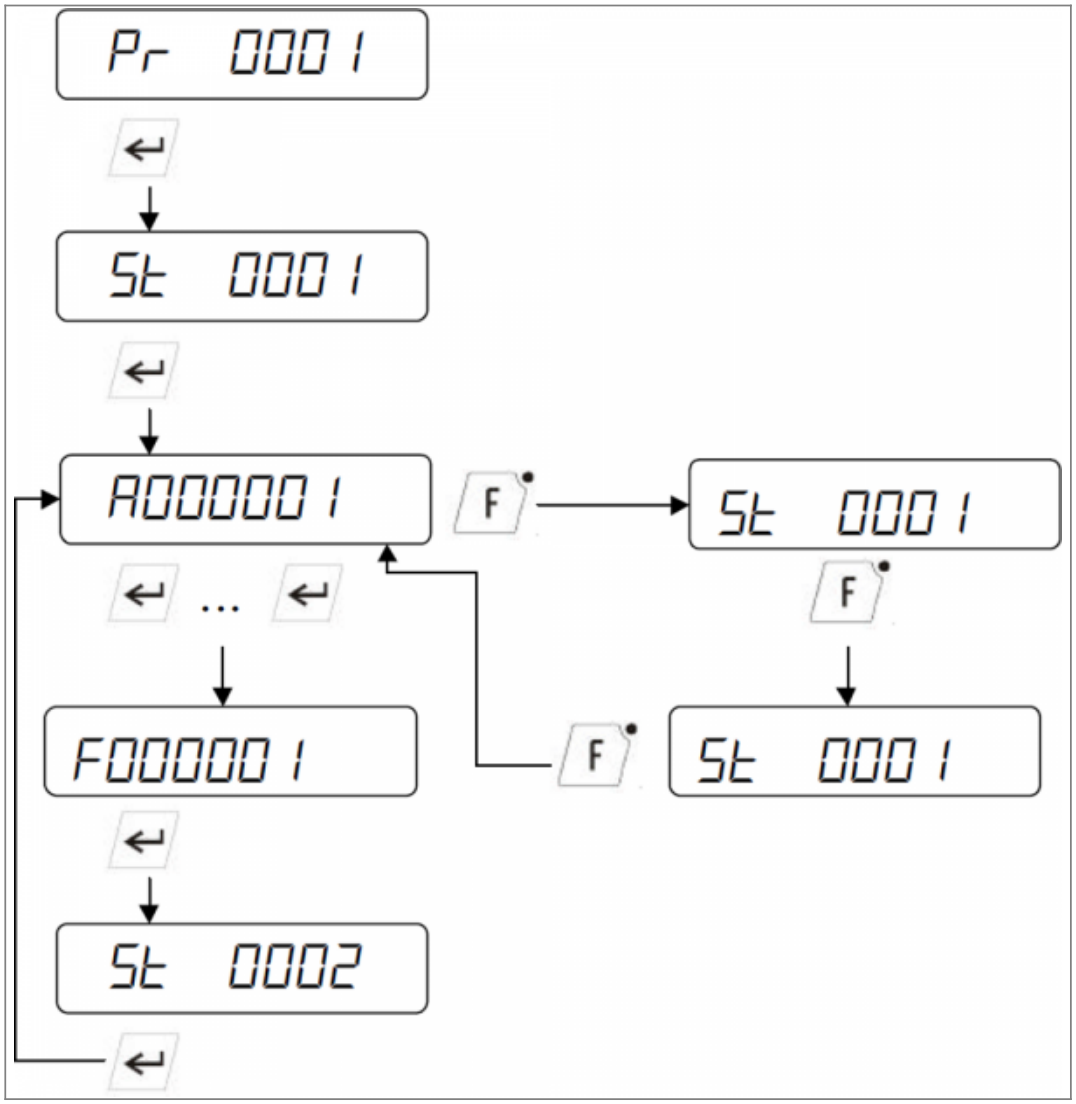


1.2.1.7 ADJUSTMENT level



To exit from levels and return to recursive views press the  key for 2 seconds about.

1.2.1.8 MENU level



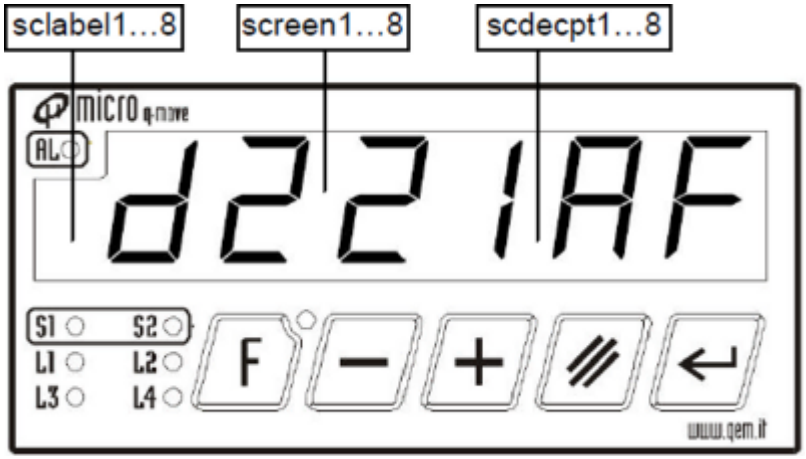
1.2.2 Recursive views

There are up to 8 recursive views that are changed using the keys  .
The number of recursive views can be set using **scnum** parameter (each bit enables a view).
All these are only in view, are not entered from keyboard.

The device allows you to set the leftmost display using with **sclabel1...8** parameter. By setting this value to 0, the display shows the value contained in the screen1...8 parameter to 6 digits and sign; if **sclabel1...8** take the value different from 0, the visualization will be 5 digits and the sign or 6 digits without sign.

For each shown variables you can set the number of decimal digits with the **scdecpt1...8** parameter.
Also with the **scalpha** parameter you can change the visualization in alphanumeric characters.

The alphanumeric characters are set through **scdis1...7** parameters.
The **scactual** parameter provides the current view (each bit has a view).



N.B.: When, in the device you try to confirm a minimum or maximum data out of bounds, the “Error” message appears for one second, then returns the introduction with the old value.
When the device, shown a exceeds data with the maximum number of shown digits, the display shows a character. The character will be:

“=” If positive visualization overflow and “_” if negative visualization overflow.

1.2.3 Alarms management

The device allows a complete alarm management.
You can force an alarm by QCL entering the following into a suitable list active alarms inside the device. You must specify which alarm you want to insert in the list with the **alvalue** parameter and his priority with **alprior**, then insert it with the **SETALARM** command.
The list is composed of a maximum of 20 items.
For the alarms priority is adjustable from 0÷19. The zero level is reserved to te message.
When an alarm is active, recursive visualization is overwritten with the alarm message type: A-01 and the relative red led will light up AL.

You can change views but after 7 seconds, the display will revert to show the alarm. The **alarm** led remain always on.

By pressing the  button for 3 sec. during the alarm message visualization, the operator can cancel the alarm. If there are more active alarms at the same time will receive the first forced to higher priority. In case of equal priority will receive the last intervened. The cancellation is always relative only to the visualized alarm.

The bit 0 of **alsetting** parameter, will allow to cancel with  key all alarms with one click.
The bit 1 of **alsetting** parameter, will allow to cancel also the messages.

The same alarm management is also used to display messages. A message behaves like a alarm only that the message consists of: M-01

A message does not active the ALARM led, stays on for 5 sec and then disappears without pressing  key.
A message is set as an alarm but **must always have zero priority**.

Alarms are displayed only in recursive views. If intervene an alarm in: F1, F2, MENÙ, MAN mode or other the visualization will appear as soon as I get out of this function. Instead, the ALARM led lights immediately.

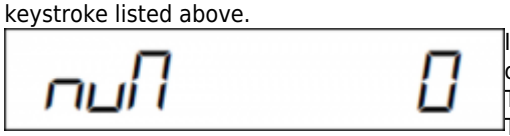
1.2.4 I/O diagnostics

The I/O diagnostics it's accessible from the FUn mene described above.

1.2.4.1 Interrupt signals visualizing

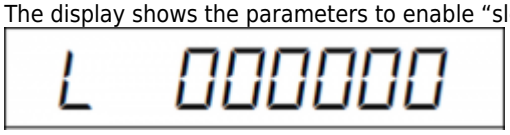
1.2.5 Manual state management

If the bit 0 of the **enable** parameter is to 1 enables the ability to access the manual state of the device through the appropriate keystroke listed above.

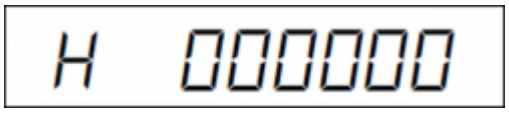


If the bit 0 of the **mansetting** parameter is to 1 is shown the written: where the operator can specify which axis intends to move.
The number that the operator writes is reported in **axisnum** parameter.
The **manvalue** and **mandecpt** parameters, allow you to specify the value to display in this state and the number of decimal places with which access respectively.

The display shows the parameters to enable “slow” movement of the axis:



or “fast” mode.



The passage and the confirmation of the parameters entered by the  key.

Jog forward and backward movements of the axis are carried out by the  buttons.

1.2.5.1 Setup parameters and generic

The HMI device provides 12 SETUP variables (**setup1…12**) and 3 generic (**par03,par04,par07**) parameters.

The system also comes with 2 other generic **par01** and **par02** parameters that may be protected by a password chosen by the programmer and can be set with the **pass01** parameter. This password can not take reserved values 100 and 123.
The first are protected with password 100 while the seconds are freely usable.

If the bit 2 of the **enable** parameter is to 1 enables the ability to go into SETUP mode.
After the introduction of the password (100), the device prompts you to enter the first setup parameter marked with the A letter.

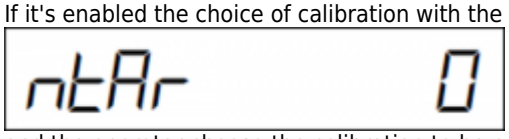
With the  keys You can cycle through the twelve variables and with the {:software:devices:hmi:hmi_03.png?nolink50|}} key to confirm the entry of a value.

If the bit 3 of the **enable** parameter is to 1 enables the ability to set the protected **par01** and **par02** parameters.
With the pressure in the sequence described in the specific paragraph, the device prompts you to enter the first parameter.
To introduce the par03…04 parameters You must enable the option setting respectively the bit 7 and 8 of the **enable** parameter.

If the bit 12 of the **enable** parameter is to 1 enables the ability to set the **par07** parameter.
The **par07** parameter is generic with the feature of being adjustable in any mode. For this parameter, you can also set the number of characters with **nchar07**, the number of decimal places with **decpt07**, the offset value with **off07** and the following configurations with the bits of the **set07** parameter:
bit 0: enables data input;
bit 1: enable the completion of the data with leading zeros (only if bit0 = 0);
bit 2: reserved;
bit 3: enables alphanumeric display;
bit 4: reserved;
bit 5: enables introduction given with exponential increase/decrease;
bit 6: disable introducing sign.
To start the introduction or the simple visualization of the **par07** parameter you use the **ENPAR07** command.

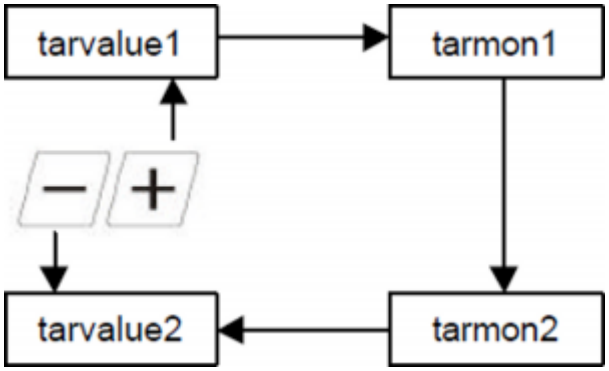
1.2.6 State management in calibration

If the bit 13 of the **enable** parameter is to 1 the device provides a structure of introductions and views in order to build a calibration sequence.
The adjustment is accessible through the sequence described above.
If it's enabled the choice of calibration with the bit 0 of the **tarsetting** parameter to 1 shown the written:



and the operator choose the calibration to be executed. The input value is shown in the **tartype** parameter.
For each calibration page the system provides up to 8 long (**tarvalue1…8**) enabled by **tarnum** parameter manage to bit (one bit for each **tarvalue**). These values can be read and modified.

You can scroll the **tarvalue** parameters by using the  keys, while it is possible to display the **tarmon1** and **tarmon2** parameters with the  key.

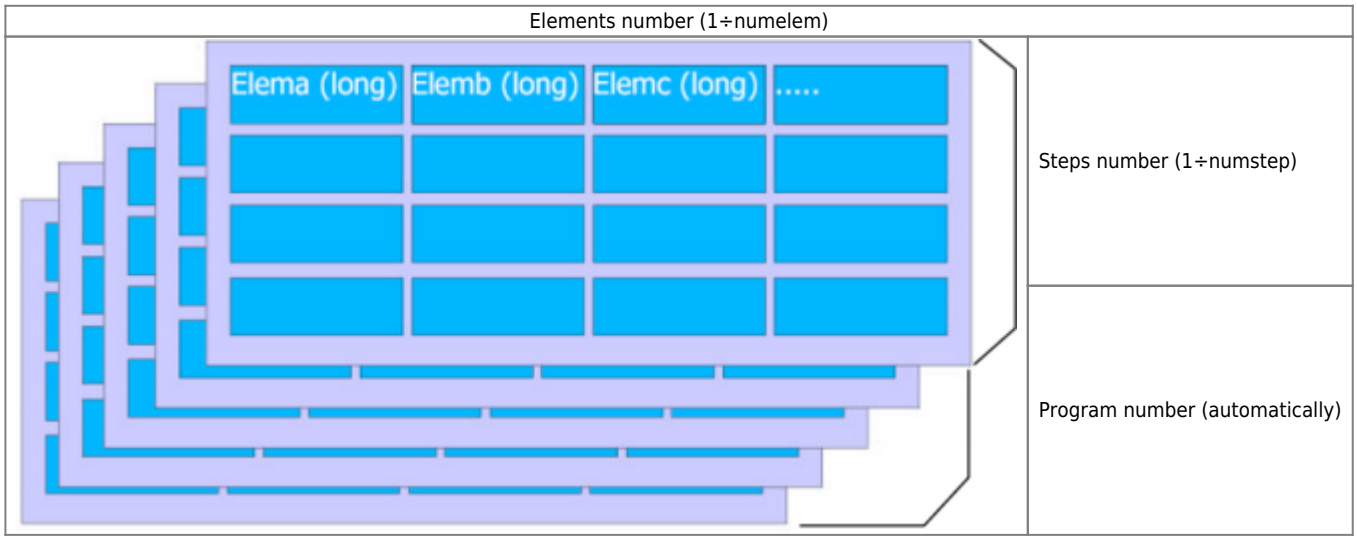


The **tarmon1…2** parameters allow you to associate two read-only variables to display during calibration.
The device provides the number of introduction current calibration, both read and write access, in the **taractual** parameter as the following table:
0 = In introduction tartype parameter
1 = In introduction tarvalue 1 parameter
2 = In introduction tarvalue 2 parameter
3 = In introduction tarvalue 3 parameter
4 = In introduction tarvalue 4 parameter
5 = In introduction tarvalue 5 parameter
6 = In introduction tarvalue 6 parameter
7 = In introduction tarvalue 7 parameter
8 = In introduction tarvalue 8 parameter

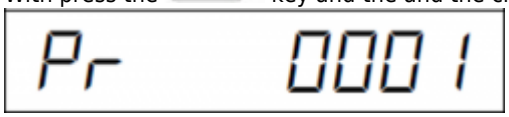
The setting of the taractual less than 0 or greater than 8 is not allowed and the default value 1 is enforced.
You can determine whether you get one of the **tarmon1** and **tarmon2** parameters through the bit 1 and 2 of the tarsetting parameter.
Normally the parameters displayed on the calibration are unlabelled to distinguish them, but you can set it through **dis1...7** parameters depending on the current view.

1.2.7 Program memory management

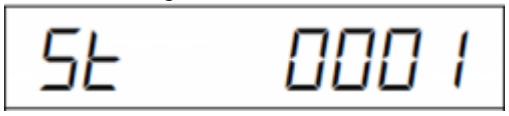
If the bit 1 of the **enable** parameter is to 1 you can access the memory management for the work programmes. This memory is located in serial flash. The device manage all operations to introduce the values.
Program memory is fully configurable by selecting the number of internal elements at every step (**numelem** from 1 to 6), and the number of steps for each program (**numstep** from 1 to 4096).
Automatically calculates the number of programs (**numprog**) (see at pag. 32 **numprog** parameter)



With press the  key and the and the choice of  menu you are prompted for dialing program.

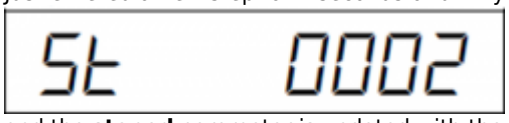


The value entered by the operator you can read it in the **proged** parameter.
If the bit 0 of the **prgsetting** parameter is to 0 is required the introduction of step number to edit.
Shown following view:



The value entered by the operator you can read it in the **steped** parameter.
If the bit 0 of the **prgsetting** parameter ia to 1 you jump right into the introduction of the first step. Any confirmation of the data introduced will go to next item. To the last item you pass to the next step. WIth the  key You can iterate over every step.

Just entered a new step for 2 seconds or until you press the  key shown following view:



and the **steped** parameter is updated with the new value of step.

If you get to the last step, you will return to first without changing program.
The **elemactual** parameter allows you to know which element of the step you are inserting. The **elemtypef** parameter lets you specify how to insert the f element.
If is 0 the f element is inserted as a single long value, While if it is including between 1 and 31 are inserted, one to one, the number of specified bits.
The **elema...f** elemnts are indicated with A...F letters.

If the bit 1 of the **prgsetting** parameter is to 1 enables the introduction of program end with the F3 key.

The introduction of the program parameter lets you specify the step number where you want to end the program. The calculation of program memory available is reduced to 4096-N programs, then:
numprog = 4096 / ((numstep * numelem * 4) + 5)

The elements of each step (**elema...f**) and the **elemend** paramenter can be read or written by selecting the program number **progin** and the step **stepin** and giving alternately the **WRITESTEP** and **READSTEP** commands.

1.2.7.1 Example:

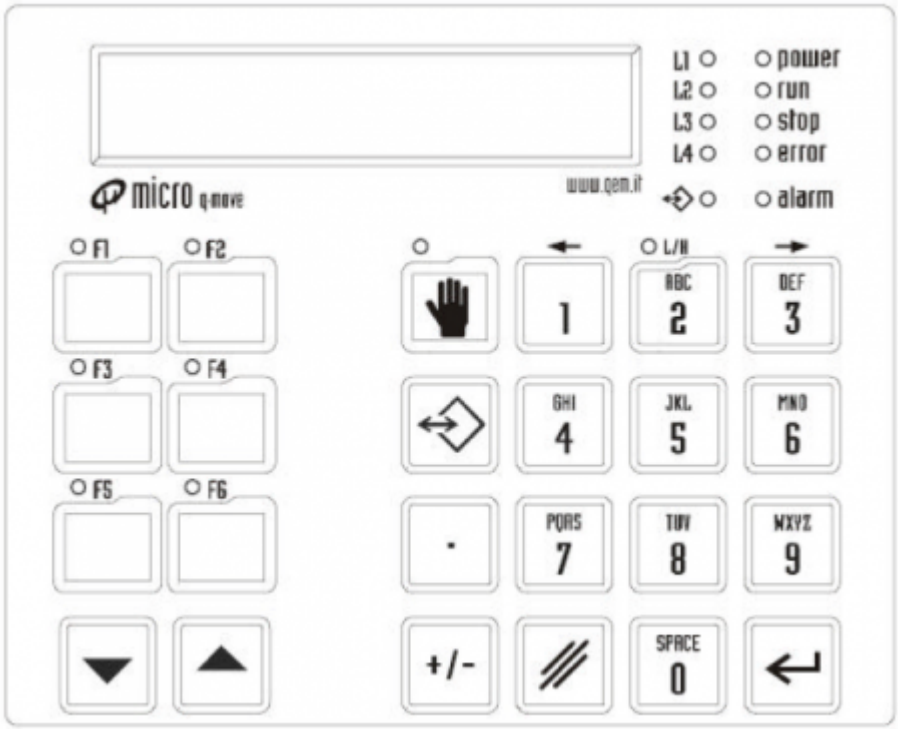
```
; ;---------------------------------
    ;Program call command

    IF gwComDisplay EQ 10
      gbI = 1
      gwComDisplay = 20
    IF gwComDisplay EQ 20
      hmi:progin = swPrgEx
      hmi:stepin = gbI
      hmi:stepout = 0
      READSTEP hmi                      ;Read device command
      gwComDisplay = 30
    ENDIF
    IF gwComDisplay EQ 30
      IF hmi:stepin EQ hmi:stepout      ;Wait reading executed

            aslArray1[gbI] = hmi:elema
            aslArray2[gbI] = hmi:elemb
            aslArray3[gbI] = hmi:elemc

            gbI = gbI + 1
            IF gbI LE NUM_STEP
                    gwComDisplay = 20
            ELSE
                    gwComDisplay = 0
            ENDIF
      ENDIF
    ENDIF
     ;---------------------------------
```

P.S.
When, from Qview, you make a “Save data” saves all the retentive parameters of the application in a file. In the microQMove, if is present the HMI device, program memory is saved (16kbytes).

1.2.8 D9 series

The HMI device manage:

- an interface consisting of a 7-digit to seven-segment line;
- an alfanumeric cable.



1.2.8.1 Constant key allocation

Key	Constant Code
F1	256
F2	512
F3	16384
F4	32768
F5	1048576
F6	2097152
UP	8
DOWN	4
MAN	1024
MENU	65536
.	4194304
+/-	16
1	2048
2	4096
3	8192
4	131072
5	262144
6	524288
7	8388608
8	1
9	2
0	64
CLEAR	32
ENTER	12

1.2.8.2 Constant leds assignment

Led	COnstant Code
L1	1
L2	2
L3	4
L4	8
F1	65536
F2	262144
F3	524288
F4	16384
F5	32768
F6	33554432
MAN	1024
MENU	16
L/H	409

1.2.9 Recorsive views

There are up to 8 recursive views that are changed using the  keys.
The number of recursive views can be set using **scnum** parameter (any bit enable a view).
All these are only in viewing, are not entered from keyboard.
The device allows you to set the leftmost display using **sclabel1...8** parameter. By setting to 0 this value, the display shows the value contained in the **screen1...8** parameter to 6 digits with sign; if **sclabel1...8** take the value different from 0 the representation will be at 5 digits with sign or 6 digits without sign.
For each of the shown you can set the number of decimal digits with the **scdecpt1...8** parameter.
Also with the scalpha parameter you can change a display at alphanumeric characters.
The alphanumeric characters are set using the **scdis1...7** parameters.
The **scactual** parameter provides the current view number (each bit has a view).

P.S. When, the device you try to confirm a given minimum or maximum limits allowed by the introduction, the message “Error” appears for one second, then returns the introduction with the old value.
When the device shown a data exceeds the maximum number of viewable digits, the display shows, for all the displays concerned, a character. The character will be:

“=” If positive shows overflow and “_” if negative shows overflow.

1.2.10 Alarms management

The device has a complete alarm management.
From QCL You can force an alarm by entering this in a separate list active alarms inside the device. You must specify which alarm you want to insert in the list with the **alvalue** parameter and the priority with **alprior**, then insert it with the **SETALARM** command.
The list is composed of up to 20 items.
For the alarms the priority is adjustable from 0÷19. The zero level is reserved for messages.
When an alarm intervenes the recursive visualization is overwritten with the alarm message type: A-01 and the relative red led will light up alarm.
You will can change the view but after a time of 7 sec., the display will return to show the alarm. The **alarm** led will remain ON in any case.

Press the key for 3 sec. during viewing of the alarm message, the operator can cancel the alarm. If there are multiple active alarms at the same time will receive the first forced to higher priority. In case of equal priority will be shown the last intervened. Deletion in any case is always relative only to the shown alarm.\ The bit 0 of the **alsetting** parameter will allow cancel with key all alarms with one click instead of one alarm at a time.
The bit 1 of the **alsetting** parameter will allow cancel even the messages.
The same alarm management is also used to viewing of the messages. A message behaves like an alarm only that the message consists of: M-01

A message does not activate the ALARM led, lasts for 5 seconds and then disappears without pressing the key.
A message can be set as an alarm but must have always zero priority.
Alarms are shown only in recursive views. If an alarm when it is: F1, F2, MENÙ, MAN or other, the viewing will appear as soon as you exit this function. The ALARM led instead turn on right away.

1.2.11 I/O diagnostics

The I/O diagnostics is accessible through the F1+6 sequence.

In this mode there are always four views are selectable using the  keys.

1.2.11.1 Interrupt signals viewing

1.2.11.2 Slot 03 inputs viewing

1.2.11.3 Slot 04 inputs viewing

1.2.11.4 Slot 04 outputs viewing

Note
For the n°2 viewing the card L3-I17 must be present in the slot 03.
For the n°3 viewing the card L3-I17 must be present in the slot 04.
For the n°4 viewing the card H3-RV0 must be present in the slot 04.

1.2.12 Manual state management

If the bit 0 of the enable parameter is to 1 enables the access the State of the device manually by  pressing. Pressing this button lights the corresponding led.
If the bit 0 of the **mansetting** parameter is to 1 is shown the written:

where the operator can specify which axis intends to move.

The number that the operator writes is shown in the **axisnum** parameter.
The **manvalue** and **mandecpt** parameters, allow you to specify the value to display in this state and the number of decimal digits with which viewing.
If the bit 1 of the **mansetting** parameter is set to 1 you log on to the state manual without key pressing.

1.2.12.1 Setup parameters and generics

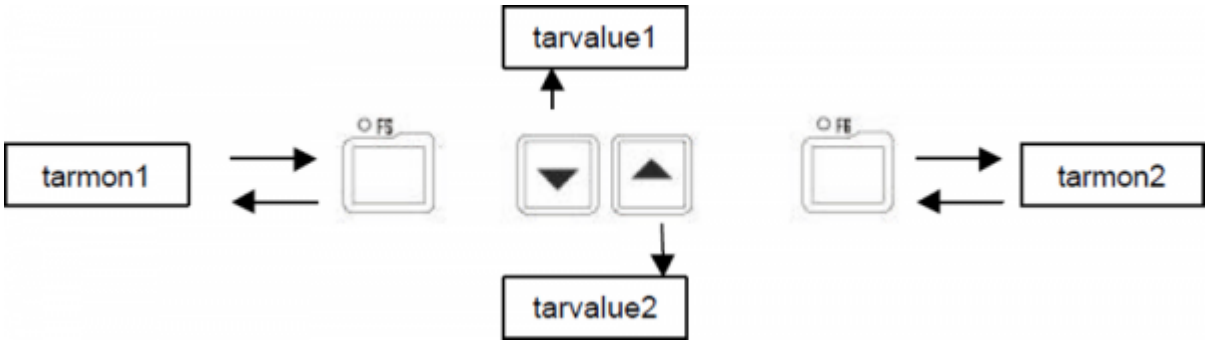
The HMI device provides 12 SETUP variables (**setup1...1**2) and 5 generic parameters (**par03...07**). The system also comes with 2 other generic **par01** and **par02** parameters that may be protected by a password chosen by the programmer and can be set with the **pass01** parameter. This password cannot take reserved 100 and 123 values.
The first are password protected 100 while the latter are usable freely.
If the bit 2 of the **enable** parameter is to 1 is enabled the SETUP input mode.
By pressing the keys in sequence F1 + 0 + 100 the device requires you to enter the first setup parameter marked by letter A.
With the arrow keys you can cycle through the twelve setup variables and with the ENTER key confirm the entry of a value.
If the bit 3 of the **enable** parameter is to 1 is enabled the setting of protected **par01** and **par02** parameters.
By pressing the F1 + 0 + XXX keys in sequence, where XXX is the password chosen by the operator, the device requires you to enter the first parameter.
To introduce the par03...06 parameters you must enable the option setting respectively the bit 7, 8, 10, 11 of the **enable** parameter.
The **par03** parameter is inserted by the F1 + 4 combination.
The **par04** parameter is inserted by the F1 + 5 combination.
The **par05** parameter is inserted by the F2 + 4 combination.
The **par06** parameter is inserted by the F2 + 5 combination.
If the bit 12 of the enable parameter is to 1 is enabled the settingof the **par07** parameter.
The **par07** is a generic parameter with the characteristic of being adjustable in any mode. For this parameter, you can also set the number of characters with **nchar07**, the number of decimal digits with **decpt07**, the offset value with **off07** and the following configurations with the bits of the **set07** parameter:
bit 0: enables data input;
bit 1: enable the completion of the data with leading zeros (only if bit 0 = 0);
bit 3: enables alphanumeric display;
bit 4: disable the hold button release.
To start the introduction or the simple viewing of **par07** parameter you use the **ENPAR07** command.

1.2.13 Adjustment state management

If the bit 13 of the **enable** parameter is to 1 the device provides a structure of introductions and views in order to build a calibration sequence. The adjustment is accessible through the sequence F1 + 0 + 123.
When enabled the choice of calibration with the bit 0 of the **tarsetting** parameter to 1 shown the written:

and the operator choose the calibration to be execute. The input value is shown in the **tartype** parameter.
For each calibration page the system provides up to 8 long (**tarvalue1...8**) enabling from the **tarnum** parameter manage to bit (one bit for each **tarvalue**). These values can be read and modified. You can scroll the tarvalue parameters by using the

keys while it is possible to display the **tarmon1** parameter with F5 key and the **tarmon2** paramter with F6 key.

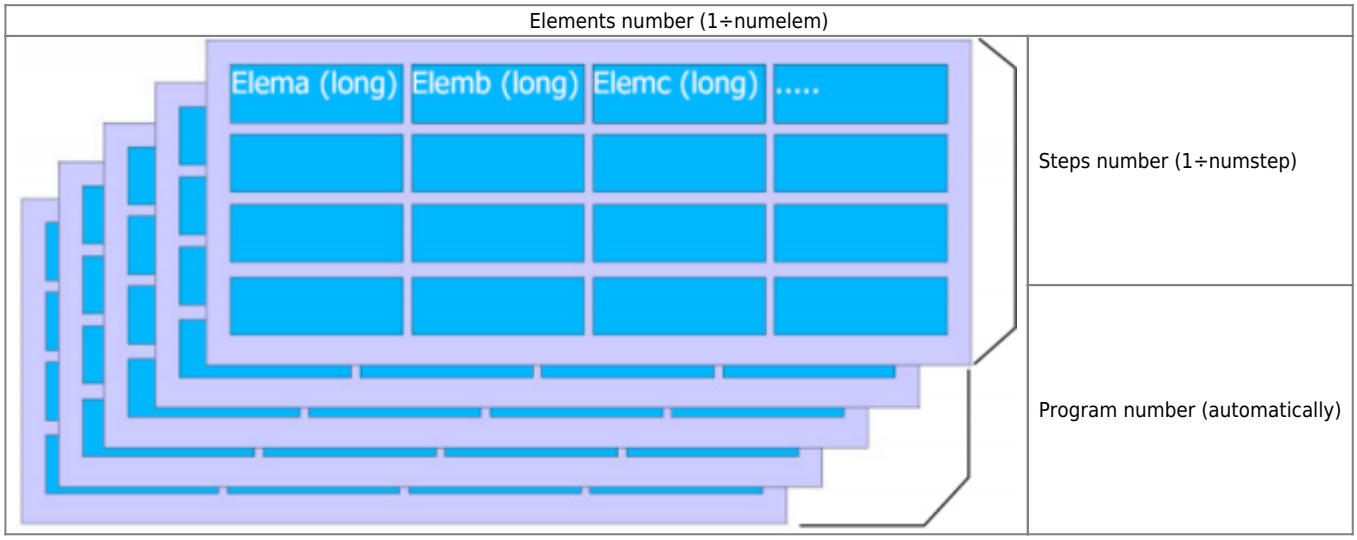


Also you can attach two variables to watch during the calibration stage with two **tarmon1...2** parameters only in reading mode.
The device provides the number of introduction current calibration, both reading and writing, in the **taractual** parameter as to the following table:
0 = In introduction tartype parameter

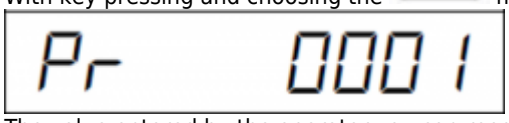
1 = In introduction tarvalue 1 parameter
2 = In introduction tarvalue 2 parameter
3 = In introduction tarvalue 3 parameter
4 = In introduction tarvalue 4 parameter
5 = In introduction tarvalue 5 parameter
6 = In introduction tarvalue 6 parameter
7 = In introduction tarvalue 7 parameter
8 = In introduction tarvalue 8 parameter
The setting of **taractual** minor to 0 or greater of 8 is not allowed and the default value is enforced to 1.
You can determine whether you get one of the two **tarmon1** and **tarmon2** parameters through the bit 1 and 2 of the **tarsetting** paramter.
Normally the parameters viewing on the calibration are unlabelled to distinguish them from each other, but you can set it through the **dis1...7** parameters depending on the current view.

1.2.14 Programs memory management

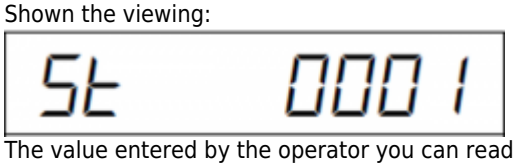
If the bit 1 of the **enable** paramteris to 1 you can access the memory management for the work programmes. This memory is located in serial flash. The device manage all operations to introduce the values.
Program memory is fully configurable by selecting the number of internal elements at every step (**numelem** from 1 to 6), and the number of steps for each program (**numstep** from 1 to 4096).
Automatically calculates the number of programs (**numprog**) (see to page 32 **numprog** parameter)



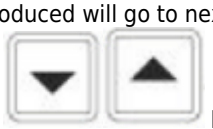
With key pressing and choosing the  menù are requested the selection of the program number.

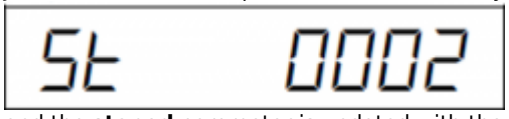


The value entered by the operator you can read it in the **proged** parameter.
If the bit 0 of the **prgsetting** parameter is to 0 is request the introduction of step number to edit.
Shown the viewing:



The value entered by the operator you can read it in the **steped** parameter.
If the bit 0 of the **prgsetting** parameter is to 1 you jump right into the introduction of the first step. Any confirmation of the data introduced will go to next item. The last item you pass to the next step.

With the  keys you can choose over every step.
Just entered a new step for 2 seconds or until you press the **ENTER** key to compare the viewing:



and the **steped** parameter is updated with the new value of step.
When I get to the end of the last step, return to first step without changing program.

The **elemactual** parameter allows you to know which element of the step you are inserting. The elemtypef parameter lets you specify how to insert the f element.
If is 0 the f element is placed as the only long value, while if it is between 1 and 31 are placed, one by one, the number of bits specified. The **elema...f** elements are indicated with A…F letters.
If the bit 1 of the **prgsetting** parameter is to 1 enables the introduction of program end with F3 key.
The introduction of the end program allows you to specify the step number to which you want to end the program. The calculation of program memory available is reduced to 4096-N programs, then:\\numprog = 4096 / ((numstep * numelem * 4) + 5)
The elements of each step (**elema...f**) and the **elemend** parameter can be read or written by selecting the program number **progin** and the step **stepin** and giving alternately **WRITESTEP** and **READSTEP** commands.

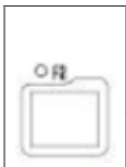
1.2.14.1 Example:

```
;-----------------------------------
    ;Program call command

    IF gwComDisplay EQ 10
      gbI = 1
      gwComDisplay = 20
    IF gwComDisplay EQ 20
      hmi:progin = swPrgEx
      hmi:stepin = gbI
      hmi:stepout = 0
      READSTEP hmi ;Reading device command
      gwComDisplay = 30
    ENDIF
    IF gwComDisplay EQ 30
      IF hmi:stepin EQ hmi:stepout ;Wait for read executing

        aslArray1[gbI] = hmi:elema
        aslArray2[gbI] = hmi:elemb
        aslArray3[gbI] = hmi:elemc

        gbI = gbI + 1
        IF gbI LE NUM_STEP
              gwComDisplay = 20
        ELSE
              gwComDisplay = 0
      ENDIF
    ENDIF
    ENDIF
;-----------------------------------
```

Note When, from Qview, making a “Save data” saves all the retentive parameters of the application in a file. In the microQMove, if there is HMI device, it also saves the program memory (16kbytes).

1.2.15 F functions management

	0	**100**	Introduction of 12 setup parameters. The values are shown on the setup01..12 parameters	
		123	Calibration	
		XXX	Introduction of 2 generic par01 and par02 parameters. The password can be set with the pass01 parameter.	
	1		Choice number of running program. "PrE 1234"	
	2		Choice number of step running. "StE 1234"	
	3		Empty introduction, you can print a string writing dis1..7 parameter “ “	
	4		Introduction of a generic long on the par03 parameter. "C 123456"	
	5		Introduction of a generic long on the par04 parameter. "d 123456"	
	6		I/O diagnostics.	" InP 788" "out 88"

		Introduction of a generic long on the par05 parameter. "E 123456"
		Introduction of a generic long on the par05 parameter. "F 123456"

The following block diagram summarizes the steps that you can run directly from the keyboard instrument. These steps, however, can be implemented also by QCL.

N.B. To exit from each of these steps and return to the views you need to use the recursive EXIT command.

1.2.16 Mappa dei caratteri

Decimal value to be introduced in: scdis1...7 sclabel1...7 dis1...7	7 segments character	Decimal value to be introduced in: scdis1...7 sclabel1...7 dis1...7	7 segments character	Decimal value to be introduced in: scdis1...7 sclabel1...7 dis1...7	7 segments character	Decimal value to be introduced in: scdis1...7 sclabel1...7 dis1...7	7 segments character
0	0	14	E	28	J	42	B
1	1	15	F	29	c	43	n
2	2	16	G	30	h	44	c
3	3	17	H	31	i	45	o
4	4	18	I	32	L	46	r
5	5	19	L	33	-	47	n
6	6	20	n	34	u	48	u
7	7	21	o	35		49	L
8	8	22	P	36		50	.
9	9	23	q	37		51	8 .
10	A	24	r	38		52	
11	b	25	t	39		53	
12	C	26	U	40		54	
13	d	27	y	41	o	55	

1.3 Commands and parameters

1.3.1 Symbols used

The **name** of the parameter, state or comnand is shown on the left of the table.

R
Indicates if it's parameter or state is retentive (upon initialization of the device maintains the previously defined), or the state assumes upon initialization of the device.
If the device does not need to initialize the “R” field indicates the value that the parameter or state take at the power on of the card.
R = Retentive
0 = Upon initialization of the device the value is forced to zero.
1 = Upon initialization of the device the value is forced to one.
- = Upon initialization of the device is presented significant value.

D
Indicates the **size of the parameter**.
F = Flag
B = Byte
W = Word

L = Long
S = Single Float

1.3.1.1 Conditions

Describes all **conditions that is considered correct or because the command is accepted**.
In some cases, limit values are specified for the acceptance of the parameter: If there are any values outside the limits you set, the data is in any case accepted; therefore appropriate controls of the application must be provided in order to ensure the proper functioning.
To run a command, ll conditions must be met; otherwise the command is not sent.

A
indicates the **access mode**.
R = Read.
W = Write.
RW = Read / Write.

1.3.2 Commands

Name	Conditions	Description
SETALARM	-	**SETTING a new ALARM** With this command you can force a new alarm. If the alarm is already on the list with the same priority it will not be forced. If the priority is different it will be updated.
CLRALARM	-	**CLEAR ALARM** As CLEAR key pressing for 3 sec. Reset the alarms.
READSTEP	-	**ReadStep** Reads the selected step in stepin.
WRITESTEP	-	**WriteStep** Writes the selected step in stepin.
ENPAR07	status = 0	**Enter on par07** Used to force the introduction or shown of the par07 parameter. In the case of the introduction of the parameter appears always the label “i”. " , 123456 "..
EXIT	-	**Exit from procedure** Allows the user to exit any procedure and return to recursive views.
CMD01	-	**Available command for future implementations** Command available for future implementations.
CMD02	-	**Available command for future implementations** Command available for future implementations.

1.3.3 Parameters

Name	D	R	A	Conditions	Description
key	L	-	R	-	**Key** Represents at all times the state of the keys. Each key is represented by one bit. For the bit assignments refer to the table in the dedicated chapter.
leds	L	0	R/W	-	**Leds status** Represents at all times the state of the keyboard leds. This variable can also be changed from the device when actions are performed in the keyboard. For the bit assignments refer to the table in the dedicated chapter.
blinkleds	L	0	R/W	-	**Blink leds status** Represents at all times the state of the blink on the keyboard led. This variable can also be changed from the device when actions are performed in the keyboard. Bit mapping reflects that of the parameter leds. To flash an led must be enabled by leds variable. The blink time is set at 300 ms ON and 700 ms OFF.
dis1…7	B	-	R/W	-	**Display position 1…7** Represents the current contents of the display at the position 1…7. Dis1 is the rightmost display and dis7 is the leftmost.
blinkdis	B	0	R/W	-	**Blink display** It's a bit variable to enable blink on a character. Each bit is a character. The least significant bit is associated with the rightmost display.
screen1…8	L	0	R/W	-	**Screen 1…8 value** It is the value of the recursive view n. 1…8
sclabel1…8	B	0	R/W	-	**Label for screen 1…8** It is the leftmost digit value of recursive view n. 1…8. If set to zero, the entire display is used for the numeric value by printing a value to 6 digits and sign. If the value is non-zero, the representation will be 5 digits and sign or 6 digits without sign.
scdecpt1…8	B	0	R/W	-	**Decimal point for screen 1…8** It is the value of the decimal point for viewing n. 1…8. Valid range: 0 ÷ 3

Name	D	R	A	Conditions	Description
scnum	B	R	R/W	-	**Number of screen** It is the number of views enabled. It's manage to bit. Bit 0 = screen1, bit 1 = screen2... ecc.
scactual	B	0	R/W	-	**Actual screen** The read provide the actual view number, in writing to set the number of the current view. **0** = screen1, **1** = screen2... ecc. Valid range: 0 ÷ 7
scalpha	B	0	R/W	-	**Screen in alpha mode** Enables the alpha mode viewing. In this mode the display shows the information contained in the scdis1...8 parameters. **Bit 0** = screen1, **bit 1** = screen2... ecc.”
scdis1...7	B	0	R/W	-	**Screen display 1...7** Represents the contents of the display while recursive viewing in alpha mode.
alvalue	B	0	R/W	-	**Alarm value (1 ÷ 99)** It's the alarm value to be included with the SETALARM command.
alprior	B	0	R/W	-	**Alarm priority (1 ÷ 99)** It's the value of the alarm priority to be included with the SETALARM command.
alsetting	B	0	R/W	-	**Alarm setting** The zero bit, if active, will be deleted with the CLEAR key (or with the CLRALARM comand) all alarms with one press instead of one at a time. The bit 1 it is used to choose how the CLEAR button works when delete all alarms: **0**: delete only the alarms and not the messages; **1**: delete everything.
mansetting	B	0	R/W	-	**Setting manual** Il bit 0, if setting, enables the axis selection in movement. The bit 1, if setting, controls the input manually without MAN keypress.
axisnum	B	0	R/W	-	**Axe number** If the axis selection is enabled indicates the axis to be moved.
manvalue	L	0	R/W	-	**Manual value** It's the value shown during the movement.
mandecpt	B	0	R/W	-	**Manual decimal point (0 ÷ 3)** E' il numero di cifre decimali durante la visualizzazione di manvalue.
taractual	B	0	R/W	-	**Actual tarature** Indicates the number of the current introduction.
tardecpt	B	0	R/W	-	**Tarature decimal point. (0 ÷ 3)** It's the number of decimal places when displaying tarvalue viewing. You can change it depending on the value of taractual through QCL. The range is between 0 and 3.
tarsetting	B	0	R/W	-	**Setting tarature** The bit 0 enables the selection of calibration. The bit 1 It indicates that the calibration is viewing the monitor 1. The bit 2 It indicates that the calibration is viewing the monitor 2.
tartype	B	0	R/W	-	**Tarature type** Indicates the setting chosen by the operator in the first display (if enabled).
tarnum	B	0	R/W	-	**Tarature number** Indicates the number of introductions enabled managed at bit. **Bit 0** = tarvalue1, **bit 1** = tarvalue2... ecc.
tarvalue1...8	L	0	R/W	-	**Tarature value 1...8** It's one of 8 values shown during calibration.
tarmon1...2	L	0	R/W	-	**Tarature monitor 1...2** <u>Only for D9 series</u> These two long contain the value shown by the procedure when F5 and F6 keys are pressed. To the F5 key is associated with the “tarmon1” variable. To the F6 key is associated with the “tarmon2” variable. The led associated to the F5 key turn on when you are viewing tarmon1, idem for the F6 led.
taractual	B	0	R/W	-	**Actual tarature** Indicates the number of the current introduction.
numelem	B	R	R/W	-	**Element number (1 ÷ 6)** Indicates the number of elements within a step.
numstep	W	R	R/W	-	**Step number (1 ÷ 4096)** Indicates the number of steps in each program.
numprog	W	-	R	-	**Program number** Indicates the number of the available programs. The value is taken from the number of long programs in memory, by “numelem” parameter and from “numstep”. If you enable the introduction of program end, by setting the bit 1 of the “prgsetting” variable to 1, the number of programs available is calculated: numprog = 4096 /(numstep * numelem + 1). If you do not enable the introduction of program end, by setting the bit 1 of the “prgsetting” variable to 0, the number of programs available is calculated: numprog = 4096 /(numstep * numelem).
proged	W	0	R	-	**Program edit** Introduction program of the program memory.
steped	W	0	R	-	**Step edit** Instroduction step into program memory.

Name	D	R	A	Conditions	Description
progin	W	0	R/W	-	**Program input** Indicates the program number to be stored with the WRITESTEP command or read with the READSTEP command.
stepin	W	0	R/W	-	**Step input** Indicates the number of the step to be stored with the WRITESTEP command or read with the READSTEP command.
stepout	W	0	R/W	-	**Step output** Indicates that the step was written, or that the step into reading is available. To verify that the command sent (WRITESTEP or READSTEP) has been executed it's should check that stepint is equal to stepout.
elema…f	L	0	R/W	-	**Element A…F** Are the values of the step used with the READSTEP and WRITESTEP commands.
elemtypef	B	0	R/W	-	**Type of element f** If set to 0 the element f is a long (as the other elements). If set to nonzero indicates the number of flags that are introduced on the elemf parameter. Valid range: 0 ÷ 31
elemend	B	0	R/W	-	**Elements for end program** It is the value of the program's end step, if enabled, read/write with the READSTEP and WRITESTEP commands.
elemdecpt	B	-	R/W	-	**Element decimal point (0 ÷ 3)** It is the number of decimal digits when showing items. You can modify it depending on the elemactual value throught QCL.
elemactual	B	0	R	-	**Actual element** Indicates the active introduction: **0**: out of menu; **1**: program introduction; **2**: step introduction; **3**: in elema introduction **4**: in elemb introduction **5**: in elemc introduction **6**: in elemd introduction **7**: in eleme introduction **8**: in elemf introduction If the elemtypef parameter is > 0 and < 32; elemactual, in the elemf parameter goes from 8 to 38. **39**: in introduction of program end (key F3 only D9). This introduction is only accessible if the bit 1 of the ""prgsetting"" parameter is equal to 1.
prgsetting	B	0	R/W	-	**Setting program data-entry** The bit 0 enables the selection of step when entering to the programs menu. Otherwise the introduction enters the step 1. The bit 1 enables the introduction of program end. When this bit is to 1, the number of programs available becomes "numprog = 4096 / (numstep * numelem + 1)". If the bit 1 is to 0 the number of programs available becomes "numprog = 4096 / (numstep * numelem)".
setup01…12	L	R	R/W	-	**Setup 01…12** Setup parameters value.
par01…07	L	R	R/W	-	**Parameter 01…07** Generic parameters value.
nchar07	B	0	R/W	-	**Char number for parameter 07** Indicates the number of characters for the 07 parameter. Valid range: 1 ÷ 7
off07	B	0	R/W	-	**Offset for parameter 07** Indicates the offset value for the 07 parameter. Valid range: 0 ÷ 6
decpt07	B	0	R/W	-	**Decimal point for parameter 07** Indicates the number of decimal digits for the 07 parameter. Valid range: 0 ÷ 3
set07	B	0	R/W	-	**Flags parameter 07** Bit 0: enables data input; **bit 1**: enables the leading zero blank (only if bit 0 = 0); **bit 2**: reserved; **bit 3**: enable alpha mode viewing; **bit 4**: disable the hold button release (only D9). Only for D2 **bit 5**: enable the introduction with exponential up/down; **bit 6**: disable the sign instroduction.
pass01	W	R	R/W	-	**Password for F + 0** Contains the value to be introduced in F1 + 0 + password to access par01 - par02 parameters. Cannot take reserved 123 and 100 values.
progex	W	R	R	-	**Program in execution** Indicates the running program (chosen with F1+1). Valid range: 1 ÷ numprog
stepex	W	R	R	-	**Step in execution** Indicates the execution step (chosen with F1+2). Valid range: 1 ÷ numstep

Name	D	R	A	Conditions	Description
status	B	0	R	-	**Status** Indicates the state of the view. It is s managed variable in bit. **Bit 0**: If 1 means it's down a number key in the keyboard and the instrument is located in the recursive views; **bit 1**: It means that showing an alarm; **bit 2**: It means that showing a message.
destatus	W	0	R	-	**Data-entry status** Indicates the status of the instrument: <u>For D2 series</u> **00**: in recursive views; **01**: in alarms views; **02**: in manual movements; **03**: in programs introduction; **04**: reserved; **05**: in password introduction; **06**: in setup; **07**: in par01 and par02 parameters introduction; **08**: in calibration (password 123); **09**: in chosing program to be execution; **10**: in chosing step to be execution; **11**: **12**: in par03 parameter introduction; **13**: in par04 parameter introduction; **14**: in I/O diagnostics; **15**: in level 1 choosing; **16**: in level 2 choosing; **17**: in par07 parameter introduction; <u>For D9 series</u> **00**: in recursive views; **01**: in alarms views; **02**: in manual movements; **03**: in programs introduction; **04**: in choosing F1 function; **05**: in password introduction; **06**: in setup; **07**: in par01 and par02 parameters introduction; **08**: in calibration (password 123); **09**: in chosing program to be execution; **10**: in chosing step to be execution; **11**: in empty introduction (F1 + 3); **12**: in par03 parameter introduction; **13**: in par04 parameter introduction; **14**: in I/O diagnostics; **15**: in choosing F2 function; **16**: in par05 parameter introduction; **17**: in par06 parameter introduction; **18**: in par07 parameter introduction; **19**: reserved.

Name	D	R	A	Conditions	Description
enable	W	0	R/W	-	**Enable** Enables the following functionality: For D2 series **bit 0**: manual movements; **bit 1**: menu; **bit 2**: setup; **bit 3**: par01 and par02 parameters introduction; **bit 4**: program in execution introduction; **bit 5**: step in execution introduction; **bit 6**: **bit 7**: par03 parameter introduction; **bit 8**: par04 parameter introduction; **bit 9**: diagnostics; **bit 10**: reserved; **bit 11**: reserved; **bit 12**: introduce with par07; **bit 13**: calibration; **bit 14**: F key; **bit 15**: level 1 enabling; For D9 series **bit 0**: manual movements; **bit 1**: menu; **bit 2**: setup; **bit 3**: par01 and par02 parameters introduction; **bit 4**: program in execution introduction; **bit 5**: step in execution introduction; **bit 6**: F1 + 3 introduction; **bit 7**: par03 (F1 + 4) parameter introduction; **bit 8**: par04 (F1 + 5) parameter introduction; **bit 9**: diagnostics (F1 + 6); **bit 10**: par05 (F2 + 4) parameter; **bit 11**: par06 (F2 + 5) parameter; **bit 12**: introduce with par07; **bit 13**: calibration; **bit 14**: F1 key; **bit 15**: F2 key.
modified	W	0	R/W	-	**Modified input** It is a managed variable in bit indicating whether during the introductions have changed some data. **Bit 0**: **Bit 1**: modified data in elema…f or elemend; **Bit 2**: modified data in setup01…setup12; **Bit 3**: modified data in par01 or par02; **Bit 4**: modified data in progex; **Bit 5**: modified data in stepex; **Bit 6**: **Bit 7**: modified data in par03; **Bit 8**: modified data in par04; **Bit 9:** **Bit 10**: modified data in par05 (only D9 series); **Bit 11**: modified data in par06 (only D9 series); **Bit 12**: modified data in par07; **Bit 13**: **Bit 14**: modified calibration data; **Bit 15**:
par01	L	-	R/W	-	**Available variable for future implementation** Variable available to future implementations.
par02	L	-	R/W	-	**Available variable for future implementation** Variable available to future implementations.

1.3.4 States

Name	D	R	A	Conditions	Description
st_alfull	B	0	R	-	**Buffer alarm full** Reporting of full alarm buffer. The status is updated as a result of a SETALARM command or when the operator press the CLEAR button. **0** = not full buffer. **1** = full buffer.
st_alactive	B	0	R	-	**Alarm active** Active alarm. **0** = there are no alarms. **1** = there is one alarm is active.
st_alset	B	0	R	-	**Alarm setted** Set to one when the alarm is set and reset with the SETALARM command.
st_alclear	B	0	R	-	**Alarm cleared** Set to one when the alarm is cleared and reset with the CLRALARM command.

Name	D	R	A	Conditions	Description
st_manfw	B	0	R	-	**Manual forward** Reporting of manual axis forward (pressing key 3 in manual): **0** = axis stopped. **1** = manual axis forward.
st_manbw	B	0	R	-	**Manual backward** Reporting of manual axis backward (pressing key 1 in manual): **0** = axis stopped. **1** = manual axis backward.
st_slow	B	0	R	-	**Slow** Reporting the speed of movement of the axis: **0** = is select the normal speed. **1** = is select the slow speed. To power up by default loads the value 1.
st_001	F	0	R	-	**Available status for future implementation** State available for future implementations.
st_002	F	0	R	-	**Available status for future implementation** State available for future implementations.

1.4 Limitations

The write operation via the WRITESTEP command must be executed bearing in mind that for the used component (Flash Eprom serial) this is costly in terms of time.
In fact the time used is variable from 512 to 1024 times the sampling time associated with HMI device. So this type of memory can be used to contain data that can be changed by the operator with relatively slow times. Definitely not a usable memory to contain data that must be written with a high frequency. In any case the write operation is executed with a background mode and will not affect the performance of the CPU to handle the rest of the device and the application.
For example, if the sampling time associated with the device is 6 ms, the time to execute a write to device can range from approximately 3 and 6 seconds. The stepout parameter becomes equal to stepin after this time.
Also the type of memory used guarantees a number of 100000 scriptures. Even so you should avoid writing programs which they write continuously on memory using the WRITESTEP command.

1.5 Application example

```
; Project :
; Module Name : DISPLAY
; Author :
; Date :
; Time :
; Description : Command manager to the display
;---------------------------------------------
;Hmi device initialization
hmi:numelem = 6                                                     ;number of items per step
hmi:numstep = 5                                                     ;number of steps
hmi:tarsetting = 1                                                  ;calibration selection enabled
hmi:alsetting = 1                                                   ;Deleting all warnings with CLEAR
hmi:enable = 1 + 2 + 4 + 16 + 512 + 4096 + 8192 + 16384             ;Various ratings

MAIN:
  WAIT gwComDisplay

  ;-----------------------------------
  ;Insert command 07 parameter
  IF gwComDisplay EQ INS_PAR_07
       hmi:nchar07 = 6                                              ;number of characters per entry
       hmi:off07 = 0                                                ;no offset on the position
       hmi:decpt07 = 0                                              ;number of decimal digits
       hmi:set07 = 1                                                ;enable the dataentry
       ENPAR07 hmi
       gwComDisplay = INS_PAR_07 + 1
  ENDIF
  IF gwComDisplay EQ (INS_PAR_07 + 1)
       IF hmi:destatus EQ 18                                        ;Waitinf ENPAR07 command executed
            gwComDisplay = INS_PAR_07 + 2
       ENDIF
  ENDIF
  IF gwComDisplay EQ (INS_PAR_07 + 2)
       IF NOT(hmi:destatus EQ 18)                                   ;Waiting out of par07 insertion
            gwComDisplay = 0                                        ;Inserted 07 parameter
       ENDIF
  ENDIF
  ;-----------------------------------
  ;Display control message with 07 parameter
  IF gwComDisplay EQ VIS_PAR_07
       hmi:nchar07 = 7                                              ;number of characters for insertion
       hmi:off07 = 0                                                ;no offset on the position
       hmi:set07 = 8                                                ;Alphanumeric viewing + read only
       ENPAR07 hmi
       gwComDisplay = VIS_PAR_07 + 1
  ENDIF
  IF gwComDisplay EQ (VIS_PAR_07 + 1)
       IF hmi:destatus EQ 18                                        ;Waiting ENPAR07 commend executed
          hmi:dis7 = 10                                             ;A
          hmi:dis6 = 26                                             ;U
          hmi:dis5 = 25                                             ;t
          hmi:dis4 = 21                                             ;o
          hmi:dis3 = 35                                             ;(space)
          hmi:dis2 = 35                                             ;(space)
          hmi:dis1 = 35                                             ;(space)
          gwComDisplay = VIS_PAR_07 + 2
          tmVisMsg = 100
       ENDIF
  ENDIF
  IF gwComDisplay EQ (VIS_PAR_07 + 2)
       IF tmVisMsg
          tmVisMsg = 1500
          gwComDisplay = VIS_PAR_07 + 3
       ENDIF
  ENDIF
```

```
IF gwComDisplay EQ (VIS_PAR_07 + 3)
     IF tmVisMsg OR (hmi:key EQ KEY_ENT)                          ;Time expired or ENTER pressing
        gwComDisplay = 0
        EXIT hmi                                                  ;exit from the par07 viewing
     ENDIF
ENDIF
;-----------------------------------
;Command program call
IF gwComDisplay EQ RIC_PRG
        gbI = 1
        gwComDisplay = RIC_PRG + 1
ENDIF
IF gwComDisplay EQ (RIC_PRG + 1)
        hmi:progin = swPrgEx
        hmi:stepin = gbI
        hmi:stepout = 0
        READSTEP hmi                                              ;Reading device command
        gwComDisplay = RIC_PRG + 2
ENDIF
IF gwComDisplay EQ (RIC_PRG + 2)
   IF hmi:stepin EQ hmi:stepout                                   ;Wait reading executed
        aslLungh[gbI] = hmi:elema
        aslRipet[gbI] = hmi:elemb
        asbVel[gbI] = hmi:elemc
        gbI = gbI + 1
          IF gbI LE NUM_STEP
             gwComDisplay = RIC_PRG + 1
          ELSE
             gwComDisplay = 0
          ENDIF
   ENDIF
ENDIF
;-----------------------------------
;Setting command of recursive views for automatic
IF gwComDisplay EQ VIS_AUTO
           ;HMI settings
           hmi:enable = hmi:enable ANDB (-1-8192)                 ;Disable calibration
           hmi:enable = hmi:enable ANDB (-1-16)                   ;Disable choice program
           hmi:enable = hmi:enable ANDB (-1-4)                    ;Disable setup
           hmi:enable = hmi:enable ANDB (-1-2)                    ;Disable programming
           hmi:enable = hmi:enable ANDB (-1-1)                    ;Disable manual
           hmi:leds = hmi:leds ORB LED_1                          ;Power on automatic led
           hmi:blinkleds = hmi:blinkleds ANDB (-1-LED_1)          ;Power off led blinking
           hmi:scnum = 127                                        ;one bit for each enabled view
           hmi:sclabel1 = CH_Q                                    ;Axis quota
           hmi:scdecpt1 = 1
           hmi:sclabel2 = CH_L                                    ;Step in execution (line)
           hmi:scdecpt2 = 0
           hmi:sclabel3 = CH_P                                    ;Program in execution
           hmi:scdecpt3 = 0
           hmi:sclabel4 = CH_S                                    ;Number of programmed pieces
           hmi:scdecpt4 = 0
           hmi:sclabel5 = CH_D                                    ;Number of produced pieces (Done)
           hmi:scdecpt5 = 0
           hmi:sclabel6 = CH_R                                    ;Number of remaining pieces (Remain)
           hmi:scdecpt6 = 0
           hmi:sclabel7 = CH_C                                    ;number of times to repeat the step in progress
           hmi:scdecpt7 = 0
                ;Message: "Auto"
           gwComDisplay = VIS_PAR_07
ENDIF
;-----------------------------------
;Setting command of recursive views for automatic
IF gwComDisplay EQ VIS_SEMIAUTO
           ;HMI settings
           hmi:enable = hmi:enable ANDB (-1-8192)                 ;Disable calibration
           hmi:enable = hmi:enable ANDB (-1-16)                   ;Disable choice program
           hmi:enable = hmi:enable ANDB (-1-4)                    ;Disable setup
           hmi:enable = hmi:enable ANDB (-1-2)                    ;Disable programming
           hmi:enable = hmi:enable ANDB (-1-1)                    ;Disable manual
           hmi:leds = hmi:leds ORB LED_1                          ;Power on automatic led
           hmi:blinkleds = hmi:blinkleds ORB LED_1                ;Automatic led flashing
           hmi:scnum = 127                                        ;one bit for each enabled view
           hmi:sclabel1 = CH_Q                                    ;Axis quota
           hmi:scdecpt1 = 1
           hmi:sclabel2 = CH_L                                    ;Step in execution (line)
           hmi:scdecpt2 = 0
           hmi:sclabel3 = CH_P                                    ;Program in execution
           hmi:scdecpt3 = 0
           hmi:sclabel4 = CH_S                                    ;Number of pieces programmed
           hmi:scdecpt4 = 0
           hmi:sclabel5 = CH_D                                    ;Number of produced pieces (Done)
           hmi:scdecpt5 = 0
           hmi:sclabel6 = CH_R                                    ;Number of remaining pieces (Remain)
           hmi:scdecpt6 = 0
           hmi:sclabel7 = CH_C                                    ;number of times to repeat the step in progress
           hmi:scdecpt7 = 0
ENDIF
;-----------------------------------
;Command setting manual views
IF gwComDisplay EQ VIS_MAN
           ;HMI settings
           hmi:leds = hmi:leds ANDB (-1 - LED_1)                  ;Automatic led off
           hmi:manvalue = anAvanz1:posit                          ;Showing value
           hmi:mandecpt = 1                                       ;Decimal digits
           hmi:mansetting = 0                                     ;None selecting axis (only)
           gwComDisplay = 0
ENDIF
;-----------------------------------
;Command set work schedules
IF gwComDisplay EQ VIS_PROG
           ;HMI settings
           hmi:elemtypef = 2                                      ;elemf bit number
           hmi:prgsetting = 1                                     ;enable step selection
           gwComDisplay = 0
ENDIF
;-----------------------------------
;Command setting recursive views of standby
IF gwComDisplay EQ VIS_STANDBY
           ;HMI settings
           hmi:enable = hmi:enable ORB 8192                       ;Enable calibration
           hmi:enable = hmi:enable ORB 16                         ;Enable choice program
           hmi:enable = hmi:enable ORB 4                          ;Enable setup
           hmi:enable = hmi:enable ORB 2                          ;Enable programs introduction
           hmi:enable = hmi:enable ORB 1                          ;Enable manual
           hmi:leds = hmi:leds ANDB (-1-LED_1-LED_2-LED_3-LED_4)  ;All leds OFF
           hmi:leds = hmi:leds ANDB (-1-LED_F3)
           hmi:scnum = 127                                        ;one bit for each enabled view
           hmi:sclabel1 = CH_Q                                    ;Axis quota
           hmi:scdecpt1 = 1
           hmi:sclabel2 = CH_L                                    ;Step in execution (line)
           hmi:scdecpt2 = 0
           hmi:sclabel3 = CH_P                                    ;Program in execution
           hmi:scdecpt3 = 0
           hmi:sclabel4 = CH_S                                    ;Number of pieces programmed
           hmi:scdecpt4 = 0
```

```
            hmi:sclabel5 = CH_D                                     ;Number of produced pieces (Done)
            hmi:scdecpt5 = 0
            hmi:sclabel6 = CH_R                                     ;Number of remaining pieces (Remain)
            hmi:scdecpt6 = 0
            hmi:sclabel7 = CH_C                                     ;number of times to repeat the step in progress
            hmi:scdecpt7 = 0
            EXIT hmi                                                ;Exit from any hmi state
            gwComDisplay = 0
  ENDIF
  ;-----------------------------------
  ;-----------------------------------
  ;Setting viewing for setup
  IF gwComDisplay EQ VIS_SETUP
            ;HMI settings
            hmi:leds = hmi:leds ORB LED_2
            hmi:setup01 = anAvanz1:measure
            hmi:setup02 = anAvanz1:pulse
            hmi:setup03 = anAvanz1:tacc
            hmi:setup04 = anAvanz1:tdec
            hmi:setup05 = aswModiPunz[1]
            hmi:setup06 = aswModiPunz[2]
            hmi:setup07 = aswModiPunz[3]
            hmi:setup08 = aswModiPunz[4]
            hmi:setup09 = swOutToll
            hmi:setup10 = swVelMinMan
            hmi:setup11 = swVelMaxMan
            hmi:setup12 = slRitSvolg
            gwComDisplay = 0
  ENDIF
  ;-----------------------------------
  JUMP MAIN
END
```