ALTER

ALTER ELETTRONICA s.r.l
15033 Casale Monferrato (AL) – ITALY

13/015

Converter module Fanuc Pulsecoder / Encoder TTL
+ Linear positioner

Instruction manual: 91/117 - Version 1.0 - Date: 04/19/2018
Compatible with Firmware V1.x

Chapter1 - Index

General index

Chapter2 - Safety Information

- Read this manual carefully before using the module 13/015.
- Store the manual carefully and in a place of easy access for future reference if needed.
- Make sure that this manual is delivered to the end user.

The safety symbols used in this manual are described below:

	**DANGER:** This symbol indicates the possibility of serious injury to persons due to electrical or mechanical shocks.
	**CAUTION:** This symbol indicates the possibility of damage to objects or to the module itself.
	**WARNINGS**: Additional information for the correct use of the module.

- ✔ Make sure the module power supply voltage corresponds to the data plate.
- ✔ Never supply power to the module without the cover and do not remove the cover while power is present.
- ✔ Do not perform manipulations on the module with wet hands. There is a danger of electrical shock.
- ✔ Before you start wiring make sure there is no power supply.
- ✔ Before performing any maintenance must be disconnected all power sources.
- ✔ Maintenance, inspection and replacement must be performed by a designated person.

- ✔ Always clamp the module before wiring.
- ✔ The installation must be performed by qualified technical personnel.
- ✔ For the respect of the rules on electrical safety, make the ground connections according to the standard of the country where the module is installed.
- ✔ Install a protection circuit (fuse or magnetic circuit breaker) feeding module.
- ✔ Never modify the form.
- ✔ Clean the module with a vacuum cleaner. Do not use organic solvents. There is a danger of damaging the module.
- ✔ It 'important for your safety that a possible revision of the module is carried out by our company.
- ✔ In case of disposing, the module is considered an industrial waste, therefore comply with the rules imposed by the laws of the country where it is installed.

The module 13/015 meets the following industry standards:

Standard/Marking	Description
CEI EN 60204-1	Safety Low Voltage Directive, 73/23/CEE.
CEI EN 61800-3	Product standard referred to the directive EMC 89/336/CEE.
CEI EN 60529	Protection level IP20.
CE	Marking CE.

Chapter3 - Technical features

3.1 Generality

The module 13/015 has two functions: the first is to convert the signal of a Fanuc PULSECODER serial encoder into an incremental TTL Line Driver with Hall sectors, the second is a two position linear positioner customized to customer requests. Fanuc serial encoders supported:

- αi64 (Series: A860-0365).
- αiAR128 (Series: A860-2010).
- αiA1000 (Series: A860-2000).

The module reports alarms and warnings present in the encoder and self-configures reading the encoder resolution stored within it.

Furthermore, the module provides a digital output + 24V to report faults and lock the operation of the axle or other accessories.

All controls are opto-isolated and operate at 24VDC positive logic and can be generated by: buttons, relay contacts, PLC outputs, etc. and come from one or more points.

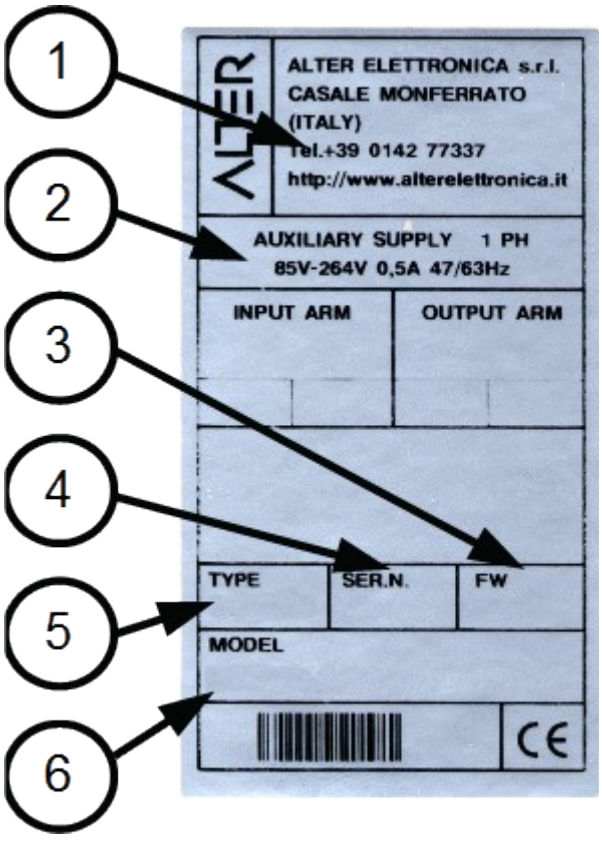
The digital outputs are opto-isolated, work at 24VDC positive logic and are electronically protected against overload and short circuit. The state command and outputs is displayed with LEDs.

All settings are made with a PC connected to the USB port of the module using the software provided by ALTER, and are stored internally in the module.

The electronic circuits and the I / O connectors are on a printed circuit board placed inside a metal container to obtain the best shielding against interference.

Alarms are stored in the module, you can be viewed via the PC and reset with a corresponding digital input.

3.2 Name plate



Explanation of the various fields of the label:

1. Manufacturer's name, address, contacts.
2. Supply voltage auxiliary services.
3. Version of firmware loaded into the module.
4. Serial number of the module.
5. Type of module.
6. Model of the module.

All other non-specified areas, are not used in this product.

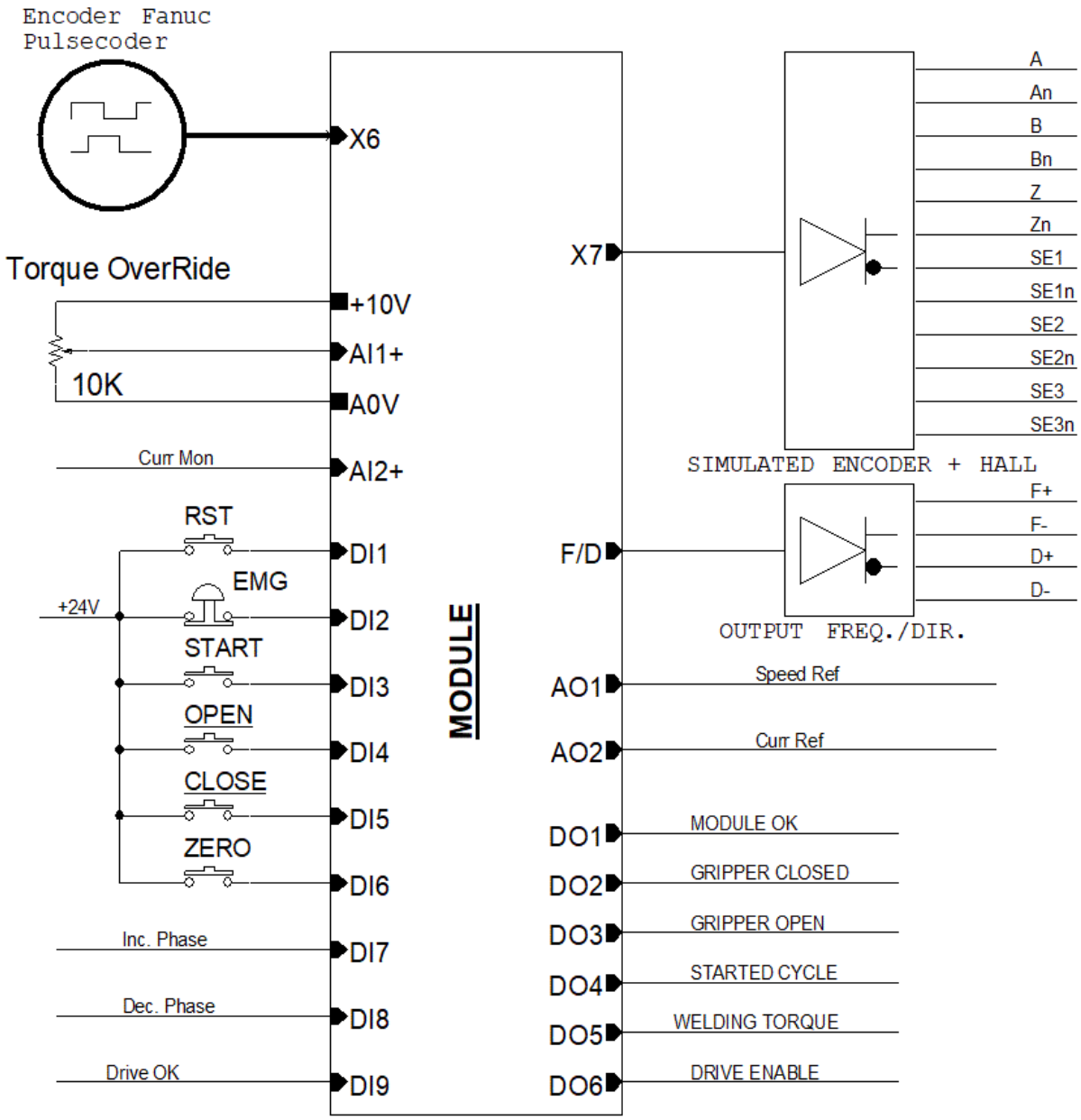
3.3 Technical data

- Implementation for mounting on panel. IP20 protection.
- Operating ambient temperature: from 0 °C to + 40 °C.
- Storage temperature: -10 °C to + 70 °C
- Maximum relative humidity: 95% non-condensing.
- Maximum altitude: 1000 m. a.s.l.
- Single-phase power supply: 85 ÷ 264VAC (47 ÷ 63Hz), 120 ÷ 370Vcc / 500mA max (protect with delayed fuses 250V / 1A).
- Over-voltage protection of:
  - Inputs and signal outputs.
  - Service power supplies.
- Service connections and signals on removable connectors
- Opto-isolated logic inputs (command 15 to 30VDC / 10mA max).
- Opto-isolated digital outputs (24VDC / 100mA max) protected against overload and short circuit.
- Analog voltage outputs, with resolution 14 bits + sign (± 10V max. - 100Ω output resistance).
  Power supply outputs for references:
  - + 24V ± 1% - 100mA max.
  - + 10V ± 5% - 5mA max.
  - -10V ± 5% - 5mA max.
- LED display of the logic states of digital I / O, alarms, functional module.
- Diagnostics and programming with PC software (Windows), with the ability to copy the settings from the PC to the module, and vice versa.
- Malfunction indication and alarms on a digital output.

3.4 Functional scheme

The following figure shows the functional diagram of the module that represents all inputs, outputs available, with related commands and signals, as in the standard factory configuration.

Some inputs and outputs can be modified by the customer according to their needs.



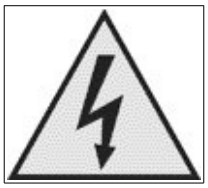
Drawing 1: Module functional diagram

Chapter4 - Installation

4.1 Preliminary operations

- Check that the module has not been damaged during transportation.
- Mount vertically the module, away from heat sources.
- Use mounting panels in non-painted sheet grounded.
- Connect to a good ground one of the ground terminals placed on the module side.
- Follow the connection diagrams shown in the manual.
- Use shielded cables for signal connections.
- Connect ground on the carpentry or to the provided terminals both ends of the screens.
- Do not use terminal blocks but only shielded connectors for the joining of shielded cable signal.
- Mount suppressors (spark suppressor for a.c. / diodes for d.c.) in parallel to the coils of all contactors, relays, solenoid valves, single phase motors, three phase motors, etc.

4.2 Power connection services (X1)



The service supply is connected to the removable connector identified with the written **ACL** and **ACN** that is located in the upper part of the module; this supply voltage can be supplied by a network with alternating current or direct current without any particular setting.

In the case of AC mains supply voltage must be between 85 and 264Vac (frequency from 47 to 63Hz); instead in from the DC voltage power supply network of the case must be between 120 and 370Vcc.

In both cases it is mandatory to protect the module with a pair of suitable voltage fuses used to, with a current size retarded 1A.

4.3 Signals connections

With reference to drawing 15 on page 25, from the top side of the module are the signals of connectors that are described in the following paragraphs.

4.3.1 Analog input connector (X2)

NAME	DESCRIPTION	
+10V	Output +10Vdc ±5% - 5mA max.	
-10V	Output -10Vdc ±5% - 5mA max.	
A0V	0V analog. The analog 0V is connected to the module container.	
AI1+	Warm pole of the analog input 1.	"Torque OverRide" adjustment potentiometer.
AI1-	Cold pole of the analog input 1.	
AI2+	Warm pole of the analog input 2.	Current supplied by the drive.
AI2-	Cold pole of the analog input 2.	
AI3+	Warm pole of the analog input 3.	Not used.
AI3-	Cold pole of the analog input 3.	
A0V	0V analog. The analog 0V is connected to the module container.	

Features common to all analog inputs:

- Maximum voltage: +/- 10V between the pole + and the pole – or than A0V.
- Input resistance: 110KΩ.
- Resolution: 11 bits + sign or 15 bits + sign.

Always use good quality shielded cables and connect the two ends of the shield grounded. On the frame of the module near the connectors, the fasteners for screens are available.

4.3.2 Analog output connector (X3)

NAME	DESCRIPTION
AO1	Analog output 1. speed reference
A0V	0V analog. The analog 0V is connected to the module box.
AO2	Analog output 2. Not used.
A0V	0V analog. The analog 0V is connected to the module box.
AO3	Analog output 3. Not used.
A0V	0V analog. The analog 0V is connected to the module box.

Features common to all analog outputs:

- Maximum voltage: +/- 10V (or 0 to 10V) between the output terminal and A0V.
- output resistance: 100Ω.
- Resolution: 14 bits + sign.

Always use good quality shielded cables and connect the two ends of the shield grounded. On the frame of the module near the connectors, the fasteners for screens are available.

> ***NOTE:*** *due to the 100Ω output resistance, you have to consider that you may need to adjust the analog output gain to reach the value indicated in the 10V features. For example: if the analog output is connected to an analog input of a drive having input resistance of 10K, it must consider that from empty to load the signal will decrease by about 1%, so we will have 9,9V instead of 10V.*

4.3.3 Can Bus Connector (X4)

NAME	DESCRIPTION
TRM	Inserting the bus terminating resistor.
H	Can bus wire H.
L	Can bus wire L.
A0V	0V analog. The analog 0V is connected to the module box.

> ***NOTE:*** *in this module the "Can Bus" connector is not used.*

4.3.4 USB connector (X5)

This connector is for connecting a USB type B cable to your PC for programming, diagnostics, parameter save. For more information see section 5.2 on page 13.

4.3.5 Fanuc serial encoder input connector (X6)

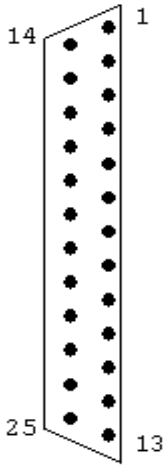
This connector is used to connect the encoder to the module: it is compulsory to use a shielded cable with twisted pair conductors to have a cleaner signal and immune to any disturbances and the shield must be connected to ground at both ends.

At module 13/015 can be connected a Fanuc serial encoder from the following models:

- αi64 (Series: A860-0365).
- αiAR128 (Series: A860-2010).
- αiA1000 (Series: A860-2000).

To use other models, contact our technical department.

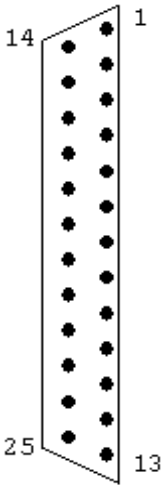
4.3.5.1 Serial encoder Fanuc αI64



INPUT SIGNAL ENCODER (X6)		CONNECTION	ENCODER		
SIGNAL	PIN N°		PIN N°	SIGNAL	CONNECTOR
+5V (encoder positive supply)	1		K	+5V	
0V (Supply 0V encoder)	2		T	0V	
REQ pair screen	8				
REQ+ serial interface	9		F	REQ	
REQ- serial interface	10		G	*REQ	
SD pair screen	16				
SD+ serial interface	17		A	SD	
SD- serial interface	18		D	*SD	
0V (cable shield) – connector housing					

View from solder side of the mobile connector "D" type 25 pin female.

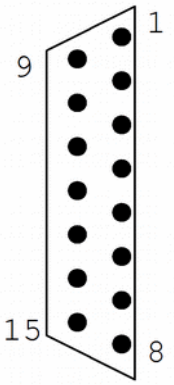
4.3.5.2 Serial encoder αiAR128 e αiA1000



INPUT SIGNAL ENCODER (X6)		CONNECTION	ENCODER		
SIGNAL	PIN N°		PIN N°	SIGNAL	CONNECTOR
+5V (encoder positive supply)	1		8	+5V	
0V (Supply 0V encoder)	2		7	0V	
REQ pair screen	8				
REQ+ serial interface	9		6	REQ	
REQ- serial interface	10		5	*REQ	
SD pair screen	16				
SD+ serial interface	17		2	SD	
SD- serial interface	18		1	*SD	
0V (cable shield) – connector housing					

View from solder side of the mobile connector "D" type 25 pin female.

4.3.6 Connector emulated encoder output (X7)



EMULATED ENCODER OUTPUT (X7)		CONNESSIONE	CNC or DRIVE	
SIGNAL	PIN N°		PIN N°	SIGNAL
Channel “A” line-driver 5V	1			
Channel “$\overline{A}$” line-driver 5V	2			
Channel “B” line-driver 5V	3			
Channel “$\overline{B}$” line-driver 5V	4			
Channel “Z” line-driver 5V	5			
Channel “$\overline{Z}$” line-driver 5V	6			
0V	9	----------		
Sector HALL “SE1” line-driver 5V	10			
Sector HALL “$\overline{SE1}$” line-driver 5V	11			
Sector HALL “SE2” line-driver 5V	12			
Sector HALL “$\overline{SE2}$” line-driver 5V	13			
Sector HALL “SE3” line-driver 5V	14			
Sector HALL “$\overline{SE3}$” line-driver 5V	15			
0V (schermo cavo) – Carcassa connettore				

View movable connector from solder side (Connector type “D” 15 pin female).

This connector is used to send to the CNC (or other users) the emulated encoder signal: it is identical to the signal provided by a classical incremental encoder TTL line driver square waves.

If necessary, you can change the setting of the resolution to reduce the maximum pulse frequency: see explanation for the configuration in the section 5.5.1 on page 14.

Also on this connector it is recommended to use shielded cable with paired braided wires.

4.3.7 Connector signals Output frequency / direction (X8)

NAME	DESCRIPTION
F+	Frequency signal (Line Driver 5V – Direct pole)
F-	Frequency signal (Line Driver 5V – Negate pole)
D+	Direction signal (Line Driver 5V – Direct pole)
D-	Direction signal (Line Driver 5V – Negate pole)

NOTE: *in this module the X8 connector is not used.*

4.3.8 Digital input connector (X9)

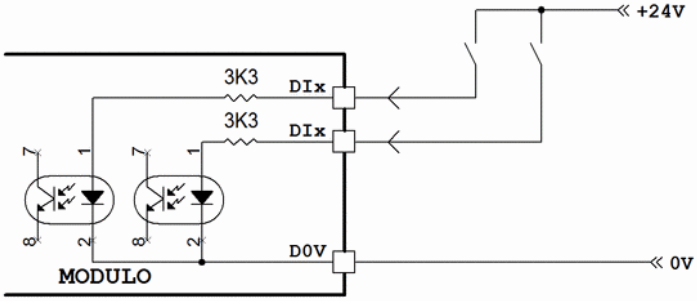
NAME	DESCRIPTION
DI1	Digital input 1: alarm reset
DI2	Digital input 2: emergency button.
DI3	Digital input 3: "Start Cycle" button.
DI4	Digital input 4: "Open gripper" button.
DI5	Digital input 5: "Close gripper" button.
DI6	Digital input 6: "Execute zero" button.
DI7	Digital input 7: "INCREASE offset electric angle" command
DI8	Digital input 8: "DECREASE offset electric angle" command

NAME	DESCRIPTION
DI9	Digital input 9: "Drive OK" drive signal.
D0V	0V digital inputs.
A0V	0V analog. The analog 0V is connected to the module box.
+24V	Output power supply + 24V – 100mA max.

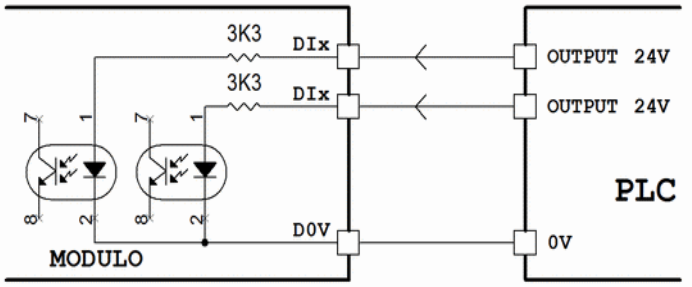
Supply voltage from 18Vdc to 30Vdc (nominal 24VDC). The 24VDC power can be supplied from the same module (if the total current drawn by the loads connected to the outputs does not exceed 100mA): D24 connected to terminal + 24V (see paragraph 4.3.9) and D0V with terminal A0V. If you can not use the internal power supply, you must use an external power supply.

The status of each digital input is displayed by the corresponding LED which indicates that the command is valid (see paragraph 6.1 on page 24).

Examples of the digital inputs connections:



Drawing 2: Connection of contacts to digital inputs



Drawing 3: Connecting PLC outputs to the digital inputs

4.3.9 Digital output connector (X10)

NAME	DESCRIPTION
+24V	Output power supply + 24V – 100mA max.
D24	Common to be connected to + 24V for the digital outputs.
DO1	Digital output 1: Module OK.
DO2	Digital output 2: "Closed gripper" signal.
DO3	Digital output 3: "Open clamp" signal
DO4	Digital output 4: "Automatic cycle started" signal
DO5	Digital output 5: "Welding torque reached" signal.
DO6	Digital output: drive enable.

Supply voltage from 18Vdc to 30Vdc (nominal 24VDC). The 24VDC power supply can be provided by the module itself (if the total current drawn by the loads connected to the outputs does not exceed 100mA): D24 connected to terminal + 24V and D0V with terminal A0V (see paragraph 4.3.8). If you can not use the internal power supply, you must use an external power supply.

Output states:

OFF = floating

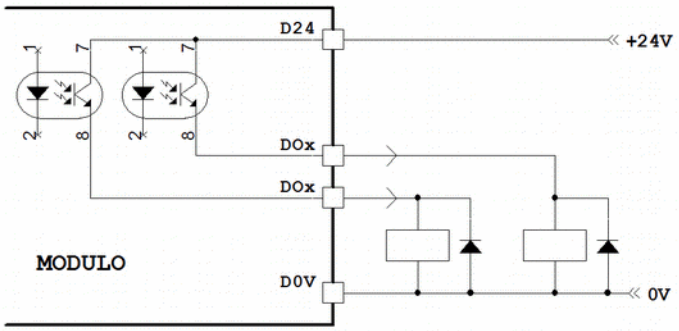
ON = Connected to the + 24V power supply (D24) (signaled by the corresponding LED)

Without service supply all outputs are OFF.

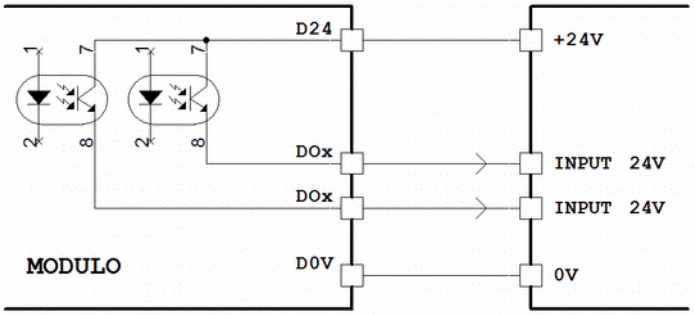
Maximum current for each output 100 mA, internal voltage drop at maximum current 2V. In the event of overload or short circuit on one or more outputs all outputs are forced OFF state permanently and the module signals the anomaly.

The status of each digital output is displayed by the corresponding LED which indicates that the output is commanded (see paragraph 6.1 on page 24).

Examples of connections to the digital outputs:



Drawing 4: Connecting relay to the digital outputs



Drawing 5: Connecting PLC inputs to the digital outputs

4.4 Restart after an alarm

When the module enters the alarm state (DO1 = OFF and flashing red led FLT) it is not guaranteed the accuracy of the output signals (emulated encoder), so the user has to take measures to prevent damage or danger to property or people.

After detecting the cause of the alarm, you can reset the module with one of the ways specified in paragraph 5.13.1 on page 21.

Chapter5 - Commissioning

To configure the module you need to be equipped with:

1. A PC with Windows operating system.
2. One free USB port on your PC (you can also use a USB Hub).
3. A USB connection cable type B (the one used for USB printers).
4. The software to be loaded on the PC to interface to the module (provided by Alter on request).
5. The driver for the USB connection (if there is an Internet connection, this is not necessary because the module is Plug & Play and the driver is automatically downloaded).

In the absence of any of the above points you can not configure or take a module diagnostics.

> ***NOTE:*** *This book does not cover the topic of software, drivers, or other problems relating to compatibility with PC supplied to the customer. In case of need you can contact the technical department ALTER. The commissioning requires that the client PC is configured and ready to use.*

5.1 Predispositions

Before setting the parameters of the module is mandatory to follow these steps:

- Connect the cable between EnDat encoder and X6 connector as shown in section 4.3.5 on page 8.
- Connect the cable between the CNC (or other user of the simulated encoder) to the connector X7 as described in paragraph 4.3.6 on page 10.
- Provide auxiliary power to the appropriate terminals (see paragraph 4.2 on page 7).
- Will illuminate all LEDs for 3 seconds (LED test), then most will go out.
- Check that the green LED "ON" is flashing. For the moment the other LEDs do not matter.
- Connect one end of the USB cable to the X5 connector on the module and the other end to a free USB port on the PC.
- Eventually wait for the time necessary to the PC to install the drivers for the module.
- Start the programming software on the PC.

5.2 Introduction to the software on the PC

After starting the application on the PC, go to the top menu and click "File → Open Project", select the project "13-015_V0100_EN.pmp". At this point we are in front of four areas where you see different data:

1. At the top we find the "**Toolbar**" with various buttons to perform certain functions.
2. On the left side we find the "**Project Tree**" where you can select the various groups of parameters that have been brought together for simplicity, the various scopes to analyze the low-rate signals or the recorder for analyzing fast signals.
3. At the bottom we find the "**Variable Watch**" in which the variables are displayed with their current value in real time, the parameters to be changed and any commands (alarm reset, save parameters, etc.).
4. In the central part we find an area that can change operation according to the context. In this part we can find:
    1. "**Algorithm block description**" in which they appear drawings or instructions to facilitate the setting or to better clarify the meaning of the variables listed in the "Variable Watch".
    2. "**Oscilloscope**" where you see some variables (up to 8) displayed graphically with respect to a time base, or with respect to another variable (Graph X-Y). Updating this variable is related to the bit-rate of the communication between PC and module, so the variations of fast signals can not be represented.
    3. "**Recorder**" where you see some variables (up to 8) displayed graphically with respect to a time base, or with respect to another variable (Graph X-Y). Updating this variable is related to the speed of the faster cycle (which can be seen in the menu "Diagnostics" parameter "Fast Cycle: Period"), so it can also represent variables that change in the order of micro seconds.

Without going into the details of all the functions of the various menus and buttons, in the next few paragraphs will explain how to configure the module using the software on the PC to obtain a quick start user service.

5.3 Communication port activation

- In the top menu select "Project → Options".
- From the window that appears, select the tab "Comm" and set the following values:
    - Direct RS232 Port: COMM_ALL.
    - Direct RS232 Speed: 57600.
- Press "OK" to save the changes.
- Press the "SAVE" button on the "Toolbar" to update the project.
- Press red button "STOP" in the "Toolbar" so as to clear the blue outline.

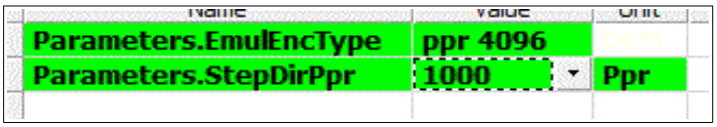
- If communication between the PC and module is done correctly, should not appear in windows alarm on the PC and in the lower right edge should appear the written “RS232; COMx; Speed=57600”.
- At this point you can continue with the other paragraphs.

5.4 How to change the values

Generally, the parameters that can be changed are highlighted with a certain color.

To change the value, proceed as follows:

- With the Windows pointer, click once on the value to be changed.
- To the right of the value it will see a gray square with a lower arrow: click once on it (see Drawing 6).
- At this point there may be two situations:
  1. The value to be changed is highlighted: in this case you can write with the numeric keypad a numeric value.
  2. A small window appears with the written values: in this case it is mandatory to choose between the listed values.
- After the selection, press ENTER.
- If the value is written and if they do not appear alarm messages on the lower left, then the parameter has been accepted and is already operational.



Drawing 6: Example of value modification

5.5 Fast commissioning

First of all, check that the status of the front LEDs is as indicated in these points:

- The red LED FLT is OFF.
- The green LED DO1 is ON steady.
- The yellow LED DI2 is ON steady.
- The yellow LED DI9 is ON steady.
- The green LED ON is flashing.

If this result is not achieved, proceed to the paragraph 5.13 on page 20 to detect the cause of the problem.

1. In the "Project Tree" select the "Module parameters setting" block. In the lower part "Variable Watch" some parameters will appear that can be modified. Parameters to insert:

   1.1. **Parameters.EncoderType**: select the Fanuc Pulsecoder model to use.

   1.2. **Parameters.EmulEncType**: select the simulated TTL encoder resolution (usually 4096 PPR).
2. In the "Project Tree" select the "Encoder phasing" block. In the lower part "Variable Watch" some parameters will appear that can be modified. Parameters to insert:

   2.1. **Parameters.MotorPoleCoup**: set the number of polar pairs of the motor used.

At this point it is possible to start the commissioning which differs according to the type of user connected to the X7 connector. So starting from the simplest to the most difficult we can distinguish 4 different procedures:

1. Connection to an ALTER PWM3D model (go to page 14).
2. Connection to an ALTER PWM3A model (go to page 15).
3. Connection to a CNC, PLC or other user who does NOT use the Hall sectors (pins 10 to 15 of X7 are not used) (go to page 15).
4. Connection to a generic drive that does not provide automatic phasing (for example the ALTER BTD1 models) (go to page 15).

Follow the appropriate paragraph according to your configuration.

5.5.1 Connection to an ALTER PWM3D model drive

In this case the phasing procedure is totally automatic and is performed by the drive (see instruction manual of the PWM3D). The only operations to be performed in the module 13/015 are the following:

1. Read the parameters entered in the paragraph 5.5.
2. In the "Project Tree" select the "Save / Restore parameters" block. In the first line highlighted in GREEN, click on the writing on the right to save the modified parameters. In case of further information see the paragraph 5.12 a pag.19.
3. Follow the commissioning explained in the instruction manual of the PWM3D drive, remembering that the parameters to be inserted in its display are:

   3.1. Motor pole (Motor Pole): the number of motor poles as inserted in the parameter "*Parameters.MotorPoleCoup*" x 2

3.2. Transducer Type (Feedback Type): TTL encoder.

3.3. Encoder Lines resolution: 4096 or the value read in the parameter "*Parameters.EmulEncType*" some points above.

Commissioning for this type of use ends here. If you want more information on the various menus available in the programming SW, you can consult from the paragraph 5.6 from page 16 onward.

5.5.2 Connection to an ALTER drive PWM3A model

In this case it is possible to perform the transducer phasing directly with the drive, but NOT automatically (see instruction manual of the PWM3A). The only operations to be performed in the module 13/015 are the following:

1. Read the parameters entered in the paragraph 5.5.
2. In the "Project Tree" select the "Save / Restore parameters" block. In the first line highlighted in GREEN, click on the writing on the right to save the modified parameters. In case of further information see the paragraph 5.12 a pag.19.
3. Follow the commissioning explained in the instruction manual of the PWM3A drive, (section "Commissioning - brushless motor + TTL encoder").
   3.1. Set the SW4 (PPR Encoder) on the column equal to the value read in the parameter "*Parameters.EmulEncType*" some points above.
   3.2. Carry out the phasing of the transducer as indicated in the paragraph "Testing the motor connection and possible phasing".
4. When the motor passes the test with the PWM3A drive and can run correctly in both directions, a fine phasing adjustment can be made in this way:
   4.1. Connect two buttons to the DI7 and DI8 inputs of the module 13/015 (for convenience).
   4.2. In the "Project Tree" select the "Encoder phasing" block.
   4.3. Change the "**CmdModule.PhaseReg**" parameter to ACTIVE.
   4.4. Enable motor operation with a very low speed reference (around 10 RPM).
   4.5. With your hand you must readjustly brake the motor shaft until you feel the blows on it due to the windings switching.
   4.6. Try to command input DI7 (or DI8) until you feel that the blows are reduced. Vice versa, if the blows increase, the other digital input must be activated.
   4.7. As an alternative to the use of DI7 and DI8 inputs, it is possible to manually change the offset of the electrical angle by modifying the parameter "**Parameters.AngleEleOffs**" in small steps.
   4.8. The points from 4.5 to 4.7 they must be repeated until satisfactory operation can be achieved.
5. At the end of this procedure the modified parameters must be saved again: in the "Project Tree" select the "Save / Restore parameters" block. In the first line highlighted in GREEN, click on the writing on the right to save the modified parameters. In case of further information see the paragraph 5.12 a pag.19.

Commissioning for this type of use ends here. If you want more information on the various menus available in the programming SW, you can consult from the paragraph 5.6 from page 16 onward.

5.5.3 Connection to equipment that does not use the hall sectors

The only operations to be performed in the module 13/015 are the following:

1. Switch on the equipment and make sure everything is working properly. In case the encoder reading direction is opposite to the desired one, it is necessary to modify:
   1.1. In the "Project Tree" select the "Encoder phasing" block and change the value of the "**Parameters.EmulEncDir**" parameter from NORMAL to REVERSE.
2. Try the operation again.
3. In the "Project Tree" select the "Save / Restore parameters" block. In the first line highlighted in GREEN, click on the writing on the right to save the modified parameters. In case of further information see the paragraph 5.12 a pag.19.

Commissioning for this type of use ends here. If you want more information on the various menus available in the programming SW, you can consult from the paragraph 5.6 from page 16 onward.

5.5.4 Connection to a generic drive

In this case the transducer phasing is done completely with the module 13/015.

The operations to be performed in the module 13/015 are the following:

1. Read the parameters entered in the paragraph 5.5.
2. In the "Project Tree" select the "Save / Restore parameters" block. In the first line highlighted in GREEN, click on the writing on the right to save the modified parameters. In case of further information see the paragraph 5.12 a pag.19.
3. Connect two buttons to the DI7 and DI8 inputs of the 13/015 module (for convenience).
4. It is absolutely essential to avoid damage to the motor, connect the DO1 output of the 13/015 module to the drive so that when it goes to logic level 0 the drive must disable and stop the motor.
5. Disconnect the motor from the mechanics, so that it can rotate freely without moving any part of the machine.

6. In the drive, adjust the current limit to the minimum value.
7. In the drive, set the "Number of motor poles" (if required) = *Parameters.MotorPoleCoup* x 2, the transducer type (TTL encoder) and the encoder resolution as set in the *Parameters.EmulEncType* parameter.
8. In the "Project Tree" select the "Encoder phasing" block:
   8.1. Change the "**CmdModule.PhaseReg**" parameter to ACTIVE.
   8.2. Change the "**SpeedIncAngleOffs**" parameter in 10.0 °/sec. In this way the offset variation will be fast.
   8.3. Enable motor operation with a very low speed reference (1 Volt).
   8.4. If the motor goes off and exceeds the speed set in the "**PhaseRegSpeed**" parameter (usually 500 RPM), the module goes into alarm: flashes red LED FLT and the output DO1 goes to 0. The drive should have blocked the motor.
   8.5. In this case it is necessary to invert the encoder direction by modifying the parameter "**Parameters.EncoderDir**" from NORMAL to REVERSE.
   8.6. Re-enable the drive: the motor may stop or turn incorrectly.
   8.7. Command input DI7 or DI8 to change the parameter "**Parameters.AngleEleOffs**" until you find the right one that makes the motor run correctly.
   8.8. If the above parameter has made a complete variation of 360 ° and the right value has not been found, it is still necessary to modify "**Parameters.EncoderDir**", add 180° to the angle of "**AngleEleOffs**" and repeat from the previous point.
   8.9. When the correct OFFSET value has been found that allows a correct and controlled rotation of the motor, a finer adjustment can be made with the following points.
   8.10. Change the "**SpeedIncAngleOffs**" parameter in 1.0 °/sec. In this way the offset variation will be slower.
   8.11. Enable motor operation with a very low speed reference (around 10 RPM).
   8.12. With your hand you must readjustly brake the motor shaft until you feel a blast on it due to the windings switching.
   8.13. Try to command input DI7 (or DI8) until you hear that the blows are reduced. On the other hand, if the shots increase, one must act on the other digital input.
   8.14. As an alternative to the use of DI7 and DI8 inputs, it is possible to manually change the offset of the electrical angle by modifying the parameter "**Parameters.AngleEleOffs**" in small steps.
   8.15. The points from 8.12 to 8.14 must be repeated until satisfactory operation can be achieved.
   8.16. When finished, return the "**CmdModule.PhaseReg**" parameter to "NOT ACTIVE" mode.
   8.17. If the motor runs the opposite of the desired one, the parameter "**Parameters.EmulEncDir**" must be changed from NORMAL to REVERSE.
9. At the end of this procedure the modified parameters must be saved again: in the "Project Tree" select the "Save / Restore parameters" block. In the first line highlighted in GREEN, click on the writing on the right to save the modified parameters. In case of further information see the paragraph 5.12 a pag.19.

Commissioning for this type of use ends here. If you want more information on the various menus available in the programming SW, you can consult from the paragraph 5.6 from page 16 onward.

5.6 Setting the module parameters

In the "Project Tree" select the "Module parameters setting" block. In the lower part "Variable Watch" the parameters will appear to adapt the module to the equipment connected to the X7 connector and to the electric motor:

- **EncoderType**: select the Fanuc encoder model connected to the X6 connector.
- **EmulEncType**: select the type of encoder you want to emulate, among the following PPR: 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192. The higher the number of PPR and the higher the frequency of signals A, B on connector X7. From factory setting this parameter is 4096 PPR which offers an excellent ratio between precision and maximum signal frequency. The maximum frequency of the signals on A and B on connector X7 is calculated with:

$F_{MAX} = \frac{PPR * RPM_{MAX}}{60}$	**F_{MAX}**: maximum frequency (Hz) on the conductors A, $\overline{A}$, B, $\overline{B}$. **PPR**: number of PPR set in the parameter "*EmulEncType*". **RPM_{MAX}**: maximum encoder rotation speed (RPM).

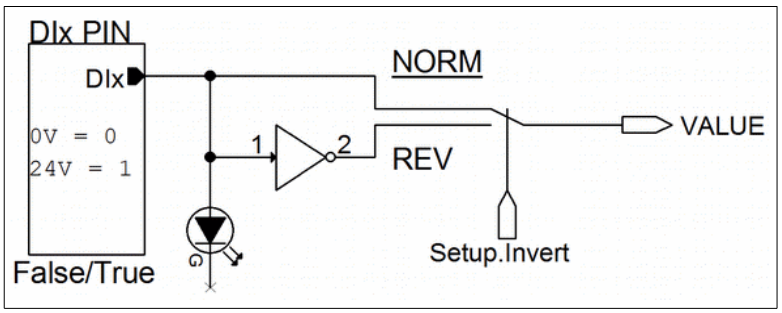
- **StepDirPpr:** [3 ÷ 65000]. Set the number of pulses per encoder revolution to be obtained on the frequency output (connector X8). The higher the number of PPRs, the greater the frequency of the signal on terminals F + and F-.

NOTE: *the "StepDirPpr" parameter must be set in conjunction with what is required by the card connected downstream of the module. The F + and F- outputs guarantee operation up to 4000KHz, but both the connection cable and the receiver card must be able to guarantee this signal frequency. The frequency on terminals F + and F- is as follows:*

$F_{MAX} = \frac{PPR * RPM_{MAX}}{60}$	**F_{MAX}**: maximum frequency (Hz) on terminals F + and F-. **PPR**: number of pulses / revolution set in the "*StepDirPpr*" parameter. **RPM_{MAX}**: maximum speed of rotation of the encoder (RPM).

5.7 Digital inputs setup

In the "Project Tree" select the block "Digital Input setup": here you can change the digital inputs settings and verify the current logic state of each input.



Drawing 7: digital input stage

Setup.Invert: with this parameter you can reverse the digital input logic state associated, whereas it if the input terminal is floating corresponds a state 0 (FALSE) instead if it is connected to + 24Vdc the status is 1 (TRUE): This status is displayed with its yellow LED on the front.

IN THIS MODULE THE PARAMETER IS FIXED.

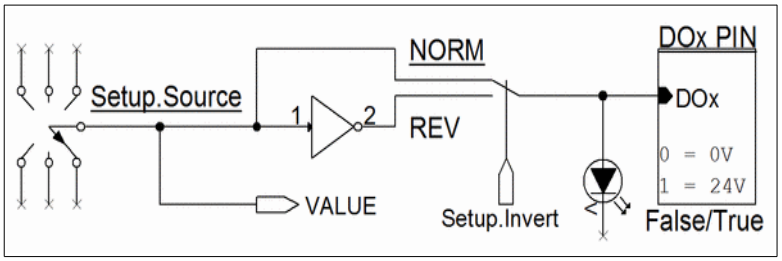
Value: This read-only parameter indicates the logic state provided to the blocks connected to that input digital.

Please remember that the digital inputs have a fixed function and can not be modified by the customer. See section 4.3.8 on page 10 to associate the function to the digital input used.

> ***NOTE**: the yellow LEDs on the front indicate the digital input logic state BEFORE the possible sign inversion, so they indicate the logic state of the input pin. With reference to Drawing 7, the LED will show the logical state of the point marked "DIx".*

5.8 Digital outputs setup

In the "Project Tree" select the block "Digital Output setup": here you can change the digital output settings and check the current logic state of each output.



Drawing 8: digital output stage

Setup.Source: with this parameter, you can view and change the signal source that will be used to control the digital output. (The source can not be changed).

Value: This read-only parameter indicates the logical state of the selected source with "*Setup.Source*".

Setup.Invert: with this parameter you can invert the logic state associated with the digital output, whereas it the status 0 (FALSE) keeps the floating digital output instead of the status 1 (TRUE) controls the output to + 24V.

Digital outputs already have established functions (see the paragraph 4.3.9 on page 11) and can not be changed.

> ***NOTE:** the green LEDs on the front indicate the digital output logic state AFTER any reversal in sign, so they indicate the logic state of the output pin. With reference to Drawing 8, the LED will show the logical state of the point marked "DOx".*

5.8.1 Signal sources for digital outputs

As indicated in the previous paragraph, in this module it is NOT possible to change the signal source that will be sent to the digital output by modifying the "*Setup.Source*" parameter. Below is a table that indicates the possible sources and the meaning of the logical states to understand the function of a certain output:

OUTPUT	SOURCE	DESCRIPTION	Status "FALSE"	Status "TRUE"
DO1	Module OK	Indicates if there are any alarms in the module.	Alarms present	Module OK
DO2	Gripper closed	Indicates if the gripper is in the closed position.	Gripper not closed	Gripper closed
DO3	Gripper open	Indicates if the gripper is in the open position.	Gripper not open	Gripper open
DO4	Cycle started	Indicates that the automatic cycle has been started	Cycle not started	Cycle started
DO5	Closing torq reac	Signals that the clamp has reached the welding torque	Torque not reached	Torque reached
DO6	Enable drive	Drive enable	Drive disabled	Drive enabled

Table 1: Signal sources for digital outputs

5.9 Verification encoder functioning

In the "Project Tree" select the "Pulsecoder Fanuc" block: here you can check the operation of the Fanuc encoder connected to the X6 connector:

- **ComInitOk**: indicates whether the communication with the encoder has been initialized correctly.

- **AngleMec**: indicates the current mechanical angle of the encoder in degrees.
- **AngleEle**: indicates the current electrical angle of the encoder in degrees.
- **SpeedRpm**: indicates the current speed of the encoder in RPM.
- **PosIniOk**: indicates if the position is valid. This means that the encoder has moved to the zero position.
- **Flags**: indicates the logic status of some flags inside the encoder.
- **PosMec**: indicates the mechanical position of the encoder, also considering the laps performed.
- **RevMec**: indicates the revolutions of the encoder after switching on.

Furthermore, by selecting the "Rx encoder & Speed data" or "Mechanical & electric angle (high speed)" oscilloscope, the angle, the encoder speed and other data useful for performing a function diagnostics can be displayed.

5.10 Verification frequency / direction output

In the "Project Tree" select the "Step / Dir Output" block: here you can check the frequency / direction exit point (see paragraph 4.3.7 on page 10) and change the parameter of PPR. Furthermore, the comparison of the pulse / rev counter of this function compared to the mechanical position of the encoder can be displayed on the oscilloscope.

The available data are as follows:

- **Parameters.StepDirPpr**: this parameter is the same that can be found in the module parameter setting menu (see paragraph 5.6 on page 16) and is used to set the number of pulses / revolution that will be generated on terminals F + and F-.
- **FioCounter**: displays a pulse counter generated by the frequency / direction output. The value will always be between 0 and the number set in "Parameters.StepDirPpr" and is used to check exactly how many pulses were generated by the module.

***NOTE**: it is not guaranteed that when the encoder is in the 0° position the FioCounter counter is at the count value 0: it is a simple pulse counter.*

5.11 Encoder phasing

In the "Project Tree" select the "Encoder phasing" block: this menu is used to adjust the encoder timing with respect to the motor in which it is mounted. For its use see the paragraphs from 5.5.2 to 5.5.4 (on page 15).

In this menu we find the following parameters:

- **CmdModule.PhaseReg**: this parameter activates or deactivates the "Encoder offset adjustment" mode, which means:
  1. A maximum speed threshold is enabled which sends the module to "Fault" if it is exceeded. This is to protect the motor in case of "escape" due to the inverted transducer.
  2. With the DI7 and DI8 inputs it is possible to increase or decrease the offset of the electrical angle and therefore the timing. It is not essential to use the digital inputs because the offset parameter can also be changed with the PC SW; this possibility has been inserted to facilitate the operation.
- **SpeedIncAngleOffs**: [0.1 ÷ 25.5 °/sec]. This parameter adjusts the rate of increase (or decrement) in degrees/sec of the parameter "*Parameters.AngleEleOffs*" which will occur when the DI7 or DI8 inputs are controlled.
- **Parameters.MotorPoleCoup**: [1 ÷ 10]. Setting the number of POLAR COUPLES of the motor in which the encoder connected to X6 is mounted.
- **PhaseRegSpeed**: [0 ÷ 30000 RPM]. Sets the maximum speed threshold during the phasing mode. If for any reason the motor speed exceeds this threshold, the module goes into "Fault" signaling the "OverSpeed" alarm.
- **Parameters.AngleEleOffs**: [-180 ° ÷ 180 °]. Sets an Offset value that will be added to the true electrical angle of the motor to create a phase shift and compensate for the position of the encoder relative to the motor rotor. This parameter is automatically changed with the DI7 and DI8 inputs.
- **Parameters.EncoderDir**: this parameter is used to invert the direction of the encoder and therefore of the sequence of the Hall sectors with respect to the rotation of the motor.
- **Parameters.EmulEncDir**: this parameter is used to invert the direction of the "Emulated Encoder" output if the motor does not rotate in the required direction.
- **StsFanuc.SpeedRpm**: this read-only parameter indicates the current speed of the encoder in RPM.
- **StsFanuc.AngleEle**: this read-only parameter indicates the electrical position received from the encoder or calculated using the mechanical angle (*AngleMec*) and the number of motor polar pairs (*MotorPoleCoup*).
- **StsModule.AngleEle**: This read-only parameter indicates the current electrical angle of the motor in degrees (including the phase shift due to the *AngleEleOffs* parameter).

By selecting one of the oscilloscopes indicated by "Hall sectors", you can see the logic states of the outputs matched to the Hall sectors (SE1, SE2, SE3) in comparison with the mechanical and electrical angle of the motor.

Remember that: $Electrical\ angle = mechanical\ angle \times pole\ pairs$ with pole pairs = Parameters.MotorPoleCoup.

5.12 Backup / Restore parameters

All changes that are made to the parameters remain valid until it is lacking the power to auxiliary services; if those changes have not been saved (stored) you will be lost and the next time you will find the old data. This feature has the advantage that, in case of accidental modification of one or more parameters, it is sufficient to remove the power supply for a few seconds and then switch it on again to return to the situation of the last backup.

In this section we will see how to store the parameters in order to find them again at the next start.

Name	Value
Backup start	**press to start -->**
Backup status:	
Memory full?	NO
Erase code error:	0
Read code error:	0
Write code error:	0
Factory data reset:	**NOT ENABLED**
Firmware download:	**NOT ENABLED**

Drawing 9: Backup / Restore parameters

In the "Project Tree" select the block "Backup / Restore parameters" in the "Variable Watch" area will appear as parameters in Drawing 9.

Backup parameters: in 1st line we find the button to start the "backup procedure", follow these steps:

- With the mouse pointer press once on the word "press to start."
- You will see a gray square (see Drawing 10). Press with the mouse pointer on the square.
- Will show "START". Press with mouse pointer over.
- After a few moments in the 2nd line will show the written "BACKUP OK" if the copy is completed correctly (see Drawing 10); otherwise it will display "BACKUP ERROR" and in the following lines there will be Error codes. If necessary, these codes can be communicated to ALTER to verify the malfunction.
- If the copy is finished successfully you can also turn off the module without the risk of losing the entered values.

Name	Value
Backup start	**press to start -->**
Backup status:	**BACKUP OK**
Memory full?	NO
Erase code error:	0

Drawing 10: backup finished

Restore parameters: if necessary you can restore the factory settings. Of course you will lose all changes made during commissioning service. To avoid that a recovery can take place accidentally, the procedure to be carried out is more complex:

- With the mouse pointer press once on the word "NOT ENABLED", the orange row with label "Factory data reset".
- You will see a gray square(see Drawing 11). Press with the mouse pointer on the square.
- You will see a menu with two entries: NOT ENABLED, and ENABLED. Select "ENABLED".
- At this point you have to obtain a situation similar to the Drawing 11.
- Remove the power supply of services for a few seconds and then restore it.
- After the restart will be loaded the original parameters, but to make them definitive must overwrite those earlier, following the "Backup parameters" procedure.

Write code error:	0
Factory data reset:	**ENABLED**
Firmware download:	**NOT ENABLED**
BootLoader version:	2.00

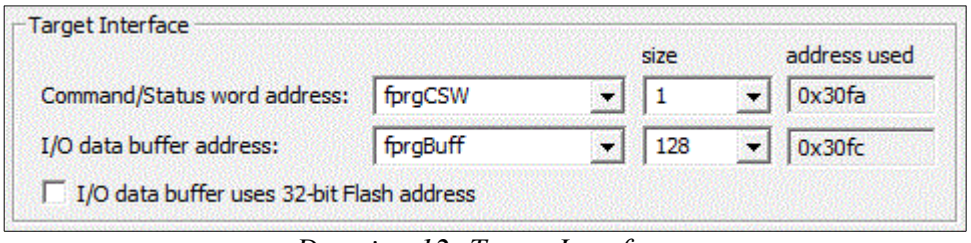
Drawing 11: restore factory parameters

> ***NOTE:*** *forcing the user to follow this procedure to restore parameters, if the command does not want we made sure that previous data is not lost. In fact, even if the user accidentally made a restore, there is still a chance to recover the mistake made: simply NOT save the restored parameters, power cycle the module to still be the previous parameters.*

5.12.1 Transfer parameters from the module to the PC

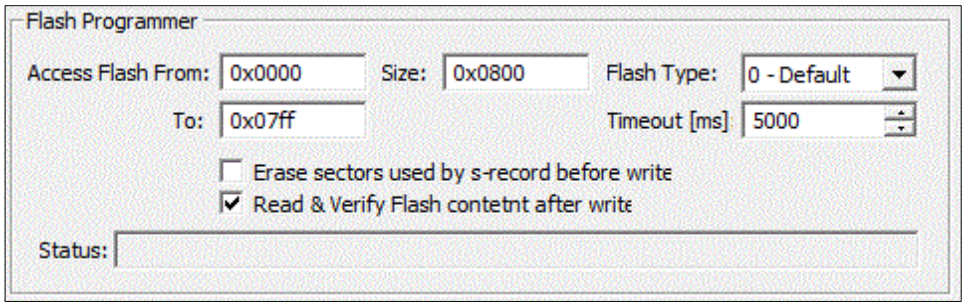
You can transfer the parameters from the module to the PC and save them to the HD storage or to restore them in the module for replacement. The following procedure will transfer all the parameters currently used in the module (IE the ones displayed in the various menus) that may also be different from those saved in the internal memory:

1. In the programming software, click on the top menu "Tools → S-Record Transfer...". A window divided into four areas with the set values or buttons to push.
2. Check that in the upper part may be all set as shown below (except "address used"):



Drawing 12: Target Interface

3. Make sure that in the lower part is all set as in the following figure:



Drawing 13: Flash Programmer

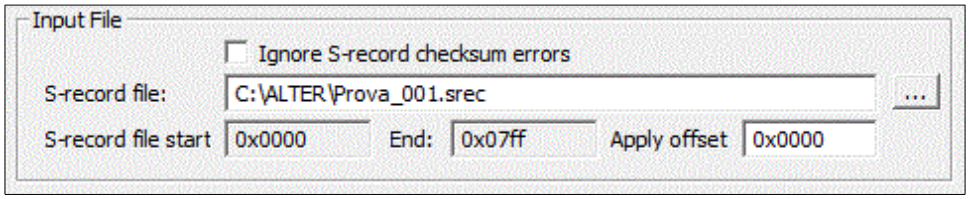
4. Press the button "Read Flash..." in the lower left. A window will open that displays the phase of the data download.
5. After a few moments another window will appear with a request to specify where to save the file.
6. It is advisable to create a "ALTER" folder in "C:" and name the data-set that can then be easily identified. In this example we will call it "Prova_001.srec".
7. Press "Close & Save Settings" at the bottom right to close the window.

NOTE: *the parameters downloaded and stored on the HD of the PC may be used only to be transferred within the same module type with the procedure explained in the next paragraph. It is forbidden to edit the file or transfer the parameters of another product: the module notices this error and block the transfer.*

5.12.2 Transferring parameters from the PC to the module

The parameters that were stored on the PC with the procedure of the preceding paragraph, may be transferred in the module with the following points:

1. In the programming software, click on the top menu "Tools → S-Record Transfer...". A window divided into four areas with the set values or buttons to push.
2. Check that in the upper part may be all set as shown in Drawing 12 and in the lower part is all set as in the Drawing 13.
3. In the central part "Input File" press the right button "..." and select the file to be transferred in the module: for example we transfer the data set stored in the preceding paragraph. You should get a situation similar to the following figure:



Drawing 14: Input file

4. Press the "Write Flash" button in the bottom middle: a window will appear showing the progress of the data transfer phase.
5. If the transfer is completed without errors, you will appear the message "Flash Write operation finished successfully" in the line "Status".
6. Press "Close & Save Settings" at the bottom right to close the window.
7. New parameters are available in the module and can be verified by selecting the various menu of the "Project Tree". To make definitive you must save them to the internal memory module by following the procedure in section 5.12 on page 19, else at the next restart of the module will return the last parameters that had been stored internally.

NOTE*: the parameters downloaded and stored on the HD of the PC may be used only to be transferred within the same module type with the procedure explained in the next paragraph. It is forbidden to edit the file or transfer the parameters of another product: the module notices this error and block the transfer.*

5.13 Module alarms

In the "Project Tree" select the "Alarms module" block: here you can see the states of all possible alarms in the module.

When the red LED "FLT" blinks or the "Module OK" output goes to level 0, it means there is an alarm present. To understand what the cause must go to this menu you see which of these alarms has the word "ALLARME".

List of alarms and possible resolution:

ALARM	REASON	RESOLUTION
StsFanuc.TimeOutFlt	Exceeded the maximum response time of encoder Fanuc	Check the connection cable, serial line connection, encoder conditions.
StsFanuc.DataRxFlt	Errors in the data received from the Fanuc encoder.	Use appropriate connection cable, arrange cable away from sources of interference, check encoder conditions. Optionally, you can increase the "ErrCntLimit" parameter to tolerate a greater number of data errors before reporting.

ALARM	REASON	RESOLUTION
StsFanuc.MotorOverTemp	The motor is overheated.	Reduce the load on the motor or control the fan.
Stsmodule.EncInternalFlt	Presence of errors inside the encoder.	Replace encoder.
Stsmodule.OverSpeed	The speed threshold has been exceeded during "phasing" mode.	If finished timing, deactivate the command. If the motor has runaway during phasing, the direction of the encoder must be reversed.
CmdModule.DriveFlt	Drive in FAULT.	Check the alarm signals on the drive
CmdModule.Emergency	Emergency button pressed.	Release the emergency button and / or check the yellow DI2 LED input on.
StsDriver.AdcLim	Saturation of the A/D converter inside the module.	Verify that the above signals are within the prescribed range: AI1, AI2, AI3.
StsDriver.I2cFlt	Internal communication problem.	Restart the module and check if it appears again. Notify the ALTER technical service.
StsDriver.OutFlt	Overload on one or more digital outputs.	Disconnect the wires connected to the digital outputs and after having reset the alarm, reconnect one by one to check what is causing the fault. In the case of capacitive loads driven by the digital outputs, it may be necessary to connect a 100Ω ½Watt resistor in series with the wire.
StsDriver.SupFlt	Auxiliary supplies out of tolerance.	Check the supply voltage of the services that is in the allowed range. Select the "Auxiliary power supply voltages" menu, check which is wrong and notify ALTER technical service.
StsDriver.WdogFlt	Cycle time out of tolerance.	Notify the ALTER technical service.

Table 2: Module alarms

Editable parameters:

- **Parameters.ErrCntLimit**: [1 ÷ 255]. This parameter indicates the number of communication-error cycles (*DataRxFlt*) which are tolerated before generating an error and lock the module (Fault). In case of disturbances, poor quality cable or other problems that make the "*StsModule.ComFlt*" alarm appear, you can try to increase this value.
- **CmdModule.ByPassMOT:** [Disabled / Enabled]. In case of over-temperature motor, this parameter can be modified to temporarily release the motor operation (MOT alarm by-pass).

5.13.1 Alarm reset

After eliminating the cause which produced the alarm, you can delete the message and restore normal operation of the module. To do this you can proceed in three different ways:

1. Remove a service power supply for a few seconds and then restore it.
2. Control the digital input 1 (DI1) with a pulse from 0V to + 24V for one second: this serves to reset from a PLC or CNC.
3. In the "Project Tree" select the block "module Alarms" and press the line where it says "CmdModule.ClrFlt"; select the word RESET.

If the FLT LED continues to flash even after making one of the points listed above, then it means that the cause of the alarm has not been resolved: refer to the "Module alarms" menu (see section 5.13 on page 20).

5.14 Diagnostics

In the "Project Tree" select the "Diagnostics" block: here you will find some data that can be useful to talk to the ALTER technical service.

5.15 Positioner

In the "Project Tree" select the "Positioner" block: here are all the parameters to adjust the positioning and the various established cycles:

- **Parameters.AxialPitch**: [0,1 mm ÷ 1000 mm]. Set the screw pitch of the mechanics. A mechanical revolution of the motor will cause the linear movement of the axis set in this parameter.
- **Parameters.AccelPos**: [0 m/s^2 ÷ 100 m/s^2]. Set the acceleration to be used during positioning movements. A larger number will result in increased stress on the mechanics and an increase in tracking error if the drive and motor are under-dimensioned.

- **Parameters.SpeedMax**: [0.001 m/s to 100 m/s]. Set the maximum motor speed as indicated on the plate mounted on it. This speed value corresponds to a 10V speed reference at the drive. Therefore the drive must be able to guarantee that set speed, otherwise the speed value scale will not be correct.
- **Parameters.SpeedPos**: [0.001 m / s ÷ SpeedMax]. Set the maximum speed that the motor will do during positioning.
- **Parameters.SpeedManF16**: [1% SpeedMax ÷ 50% SpeedMax]. Set the maximum speed of the motor that will do during manual movements or in "zero search".
- **Parameters.TorquePosF16**: [1% ÷ 100%]. Set the maximum torque that the motor can give when in positioning mode. When the value is 100% it means that the analog output AO2 "Current Reference" will be at the value 10V (see paragraph 4.3.2 on page 8), so the torque will depend on the size of the drive and the motor. This torque can be reduced in real time using the "Torque OverRide" potentiometer up to 10% of this value.
- **Parameters.TorqueManF16**: [1% ÷ 100%]. Set the maximum torque that the motor can give when it is in manual or "zero search" mode. When the value is 100% it means that the analog output AO2 "Current Reference" will be at the value 10V (see paragraph 4.3.2 on page 8), so the torque will depend on the size of the drive and the motor. This torque can be reduced in real time using the "Torque OverRide" potentiometer up to 10% of this value.
- **Parameters.TorqueWeldF16**: [1% ÷ 100%]. Set the maximum torque that the motor can give when the gripper is in the welding position. When the value is 100% it means that the analog output AO2 "Current Reference" will be at the value 10V (see paragraph 4.3.2 on page 8), so the torque will depend on the size of the drive and the motor. This torque can be reduced in real time using the "Torque OverRide" potentiometer up to 10% of this value.
- **Parameters.TorqueWindowF16**: [0% ÷ 10%]. Set the window for the welding torque signal reached. The 100% value corresponds to the peak current of the drive.
- **Parameters.TimeCycleClosed**: [0 sec ÷ 25.0 sec]. Set the time that must elapse with the gripper in the closed position in the automatic cycle. At the end of this time the gripper will go into the open position.
- **Parameters.TimeCycleOpen**: [0 sec ÷ 25.0 sec]. Set the time that must elapse with the gripper in the open position in the automatic cycle. At the end of this time the gripper will go into the closed position.
- **Parameters.PosFbkDir**: inverts the count direction of the measured mechanical position. This parameter is used when the "open gripper" position is negative and should be positive.
- **PosClosedF32**: [mm]. Set the quote for the position of "Closed gripper" (generally it is 0).
- **PosOpenF32**: [mm]. Set the quote for the position of "Open gripper", it MUST BE a positive value; if the direction is against (therefore the open gripper position is negative), invert the **PosFbkDir** parameter to obtain the quota with the positive sign.
- **CmdPositioner.PosFbkF32**: [mm]. Indicates the current position of the clamp with respect to zero, if it has been carried out. This value can be used to set the dimensions for "open gripper" and "closed gripper", only if **HomingOk** = YES.
- **CmdModule.TorqueOverRide**: [%]. Indicates the percentage of "Torque OverRide" set with the potentiometer: turning it in the directions you will see change this value and you can calculate how much the torque of the gripper is reduced.
- **StsModule.HomingOk**: indicates whether the clamp zero was made. This operation is carried out keeping the input DI6 active until the end of the operation.

Selecting the oscilloscope "Current references" you can view in graphical form:

- **StsModule.SpeedRef**: speed reference sent to the drive. With a **positive signal**, the gripper must go to opening, but with a **negative signal** the gripper must go to closing.
- **StsModule.TorqueRef**: torque reference (current) sent to the drive (always positive).
- **CmdModule.TorqueFbk**: current torque delivered by the motor, signal received from the drive.

By selecting the "Position" oscilloscope you can view in graphic form:

- **PosOpenF32**: the set quota of "open gripper".
- **PosClosedF32**: the set quota of "closed gripper".
- **CmdPositioner.PosFbkF32**: current position of the gripper with respect to the zero position.
- **StsPositioner.PosErrorF32**: current position tracking error.

5.15.1 Position loop

In the "Project Tree" select the "Position loop" block: here are the settings of the gains and other parameters to optimize the position loop:

- **Parameters.PgainPosLp**: set the proportional gain of the position loop. A higher value will reduce the tracking error, but can lead to instability.
- **Parameters.IgainPosLp**: set the additional gain of the position loop. Increasing the value will result in a reduction in tracking error, especially when the engine has reached the position. But it can lead to oscillations and instability.
- **Parameters.PosWindow**: [0,1 mm ÷ 5 mm]. Set the tolerable position error to signal the position outputs reached.

By selecting the "Position loop" oscilloscope, you can view it in graphic form:

- **PosRefF32**: instant position reference. This is the position that the gripper must have moment by moment.
- **CmdPositioner.PosFbkF32**: indicates the current position of the clamp with respect to zero, if it has been carried out.
- **StsPositioner.PosErrorF32**: current position tracking error. If the two previous values are equal, then the latter will be zero.
- **StsModule.SpeedRef**: indicates the speed reference that is sent to the drive during positioning. With a **positive signal**, the gripper must go to opening, but with a **negative signal** the gripper must go to closing.

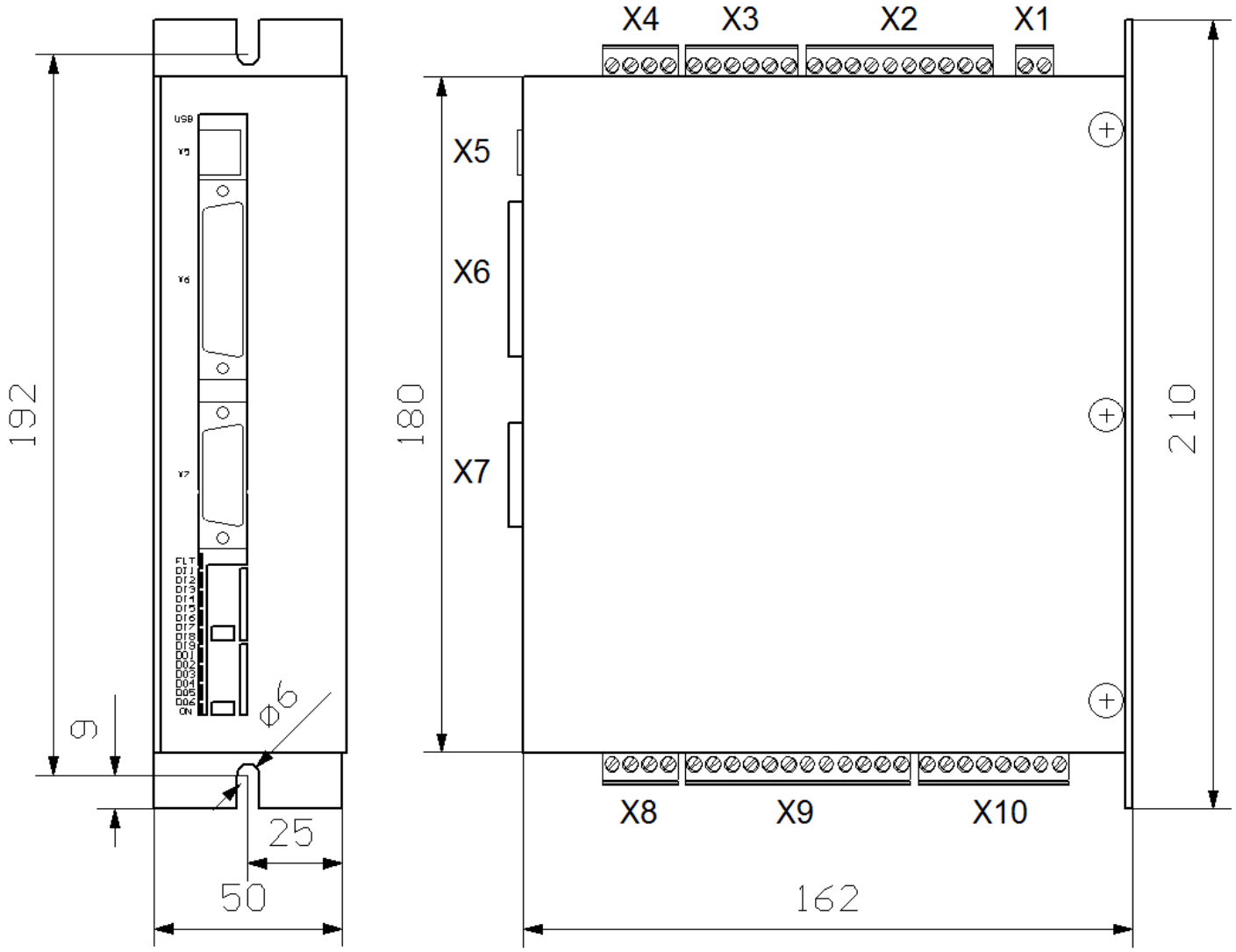
Chapter6 - Attachments

6.1 LED's summary table

In the following table, the components are listed as they appear on the front of the module, starting from the edge at the top left.

NAME		DESCRIPTION	Reference
USB	Usb	USB ongoing communication between module and PC	Paragr. 5.3 on page 13
FLT	Fault	Module in alarm condition	Paragr. 5.13 on page 20
DI1	Digital Input 1	Digital input control n°1 (Alarm reset)	Paragr. 4.3.8 on page 10
DI2	Digital Input 2	Digital input control n°2 (emergency button)	Paragr. 4.3.8 on page 10
DI3	Digital Input 3	Digital input control n°3 ("Start Cycle" button)	Paragr. 4.3.8 on page 10
DI4	Digital Input 4	Digital input control n°4 ("Open gripper" button)	Paragr. 4.3.8 on page 10
DI5	Digital Input 5	Digital input control n°5 ("Close clamp" button)	Paragr. 4.3.8 on page 10
DI6	Digital Input 6	Digital input control n°6 ("Execute zero" button)	Paragr. 4.3.8 on page 10
DI7	Digital Input 7	Digital input control n°7 ("INCREASE offset" command)	Paragr. 4.3.8 on page 10
DI8	Digital Input 8	Digital input control n°8 ("DECREASE offset" command)	Paragr. 4.3.8 on page 10
DI9	Digital Input 9	Digital input control n°9 (Drive OK)	Paragr. 4.3.8 on page 10
DO1	Digital Output 1	Digital output state n°1 (MODULE OK signal)	Paragr. 4.3.9 on page 11
DO2	Digital Output 2	Digital output state n°2 (GRIPPER CLOSED signal)	Paragr. 4.3.9 on page 11
DO3	Digital Output 3	Digital output state n°3 (GRIPPER OPEN signal)	Paragr. 4.3.9 on page 11
DO4	Digital Output 4	Digital output state n°4 (STARTED CYCLE signal)	Paragr. 4.3.9 on page 11
DO5	Digital Output 5	Digital output state n°5 (WELDING TORQUE Signal)	Paragr. 4.3.9 on page 11
DO6	Digital Output 6	Digital output state n°6 (drive enable)	Paragr. 4.3.9 on page 11
ON	Module ON	Powered and running Module (flashing).	Paragr. 5.1 on page 13

Chapter7 - Mechanical characteristics



Drawing 15: dimensions (mm)

Mass: 0,8 Kg

ALTER Elettronica s.r.l.

Via Ezio Tarantelli 7 (Z.I.)
15033 Casale Monferrato (AL)
ITALY

Tel. +39 0142 77337 (r.a.)

Fax. +39 0142 453960

Internet: http://www.alterelettronica.it

email: info@alterelettronica.it