



AUTOMATOR MARKING SYSTEMS SRL
Via Meucci n.8
Corsico Milan - Italy
Tel 02 48601445
Fax 02 48601503
e-mail automator@automator.com

AC500+ Marking Controller
Manual

Date:
21/05/2024

OPERATIONS MANUAL

INSTALLATION AND MAINTENANCE MANUAL

AC500 marking controller



Revision 18

Preface

Automator International srl reserves the right to change the information and specifications contained in this document without prior notice.

Under no circumstances shall Automator International be liable for any indirect damage, including but not limited to lost profits, in connection with this document.

Automator International expressly guarantees brand new equipment, for no other use than marking, as stated in the sales conditions in each official offer.

The information contained in this document is confidential and the property of Automator International.

INDEX

Summary

OPERATIONS MANUAL	1
Preface.....	2
CHAPTER 1	9
1.1 Using the Manual.....	9
1.2 Overview of the marking system.....	9
1.3 Terminology.....	11
1.4 Considerations.....	13
1.5 Surface to be marked.....	14
1.6 Plunger.....	14
CHAPTER 2	17
2.1 Using the controller.....	17
2.2 Starting the software.....	18
2.3 Setting the marking head status (ONLINE/OFFLINE).....	23
2.4 Loading a programme.....	25
2.5 Parking the marking head.....	27
2.6 Pre-positioning the marking head.....	28
2.7 Manual Data Input.....	29
2.8 Text with query.....	32
2.9 Vacuum marking (cycle simulation).....	34
2.10 Area check cycle.....	36
2.11 Marking a programme.....	38
2.12 Marking time.....	39
2.13 Stopping the execution of a marking cycle.....	39
2.14 Stopping and resuming the execution of a marking cycle.....	40
2.15 Full screen preview.....	40
2.16 Moving the marking head.....	41
2.17 Selecting the operating level.....	44
2.18 Display controller version.....	47
2.19 Switching off the controller.....	48
2.20 Operator login management.....	51
CHAPTER 3	53
3.1 Setting the operating level.....	53
3.2 Basic considerations of programmes and fields.....	53
3.3 Creating a programme.....	57
3.4 Programme editor screen.....	59
3.5 Setting Programme Parameters.....	62
3.6 Defining the anchor point of the programme.....	62
3.7 Defining the head parking point.....	64
3.8 Defining Default Text Field Parameters.....	65
3.9 Using Custom Fonts.....	66
3.10 Setting the programme serial number.....	68
3.11 Setting the configuration for data to be marked from CSV files.....	74
3.12 Defining the background image of the programme preview.....	77
3.13 Defining the background grid of the programme preview.....	79
3.14 Adding fields to the programme.....	80

3.14.1 Text Field Parameters	87
3.14.2 Line field parameters.....	91
3.14.3 Arc Field Parameters	92
3.14.4 Text field parameters on arc.....	95
COMMON PARAMETERS	96
GRADE METHOD	98
POINT METHOD	98
3.14.5 Datamatrix Field Parameters	99
3.14.6 Pause Field Parameters	106
3.14.7 Goto Field Parameters	107
3.14.8 Bitmap image field parameters	109
3.14.9 Comment field parameters	112
3.14.10 DXF field parameters	112
3.14.11 Automation field parameters	115
3.15 Flag Commands.....	116
3.16 Interaction and positioning via the jog environment.....	121
3.17 Editing programme fields from preview	121
3.18 Modifying a programme.....	123
3.18.1 Testing a programme.....	123
3.18.2 Saving a programme.....	123
3.18.3 Changing programme parameters	124
3.18.4 Changing field parameters	124
3.18.5 Changes in programme fields.....	124
3.19 File Management.....	124
3.20 Controller Network Management.....	126
3.21 Non-repeatability check.....	130
CHAPTER 4.....	131
4.1 Accessing System Parameters	131
4.2 Definition of Omni Serial Numbering	132
4.3 Definition of shift codes.....	137
4.4 Defining the codes of the day	139
4.5 Definition of month codes.....	141
4.6 Definition of year codes	143
4.7 Definition of time codes	145
4.8 Defining the Calendar	147
4.9 Defining a text query.....	149
4.10 Setting General Parameters	150
4.11 Setting axis parameters.....	164
4.12 Setting Machine Maintenance	168
4.13 Head stock setting	170
4.14 Setting the System Language	177
4.15 Configuring Software Graphics	180
4.16 Serial Configuration	181
4.17 USB barcode configuration	185
4.18 Configuring Network Settings.....	189
4.19 Configuring and Using the Machine Database	194
4.20 Date and time configuration.....	200
4.21 Window Header Configuration	202
4.22 Setting Modbus axis parameters	203
4.23 AC500 Store.....	209
4.24 Fieldbus card configuration.....	219

4.25	Managing programme uploads from the network	223
4.26	Alarm history display	228
4.27	Viewing System Statistics	230
4.28	Backing up system data.....	234
4.29	System Upgrade	239
4.30	Displaying Free Memory Space	241
4.31	Management of external mouse and keyboard via USB	243
4.32	Viewing Marking History	244
4.33	Virtual keyboard management	250
4.34	Board versions	251
4.35	Printer settings configuration	254
CHAPTER 5	258
5.1	System Testing	258
5.1.1	Testing of digital inputs.....	259
5.1.2	Testing of digital outputs.....	260
5.1.3	Frequency output testing	261
5.1.4	Testing the work area.....	263
5.2	Calibration of axles	265
CHAPTER 6	268
6.1	Remote Communications	268
6.2	I/O Signals.....	269
6.3	Remote Programme Selection.....	273
6.4	Control signal TTL Input	273
6.5	Host Communication.....	275
6.6	RS-232 interface	276
6.7	RS-485 interface	277
6.8	Extended Protocol	278
6.9	Programmable protocol	285
6.10	Programmable USB barcode protocol.....	288
6.11	Remote Controller Control	293
Appendix A	296
ASCII - Decimal - Hexadecimal cross-reference	296
Appendix B	297
Theta-axis configuration and use (rotational)	297
B.1	Enabling and configuring the Theta axis	297
B.2	Theta axis test	300
B.3	Calibrating the Theta Axis	301
B.4	Modifying programmes with Theta axis	304
B.5	Manual environment with Theta axis	309
B.6	Working environment with Theta axis	311
Appendix C	314
Remote programme selection table	314
Appendix D	317
Alarm description table	317
Appendix E	320
Touch Screen Calibration Procedure	320
Appendix F	321
Object Dictionary Modbus Slave function	321
Appendix G	323
EtherNet/IP Fieldbus Module	323
Appendix H	346
Profibus Fieldbus Module	346
Appendix I	365
Profinet IO Fieldbus Module	365

Document:

INSTRUCTIONS FOR USE OF THE SOFTWARE

AUTOMATOR MARKING SYSTEMS SRL - Via Meucci n.8-20094 Corsico (MI) Italy
Tel 02 48601445 Fax 02 48601503 e-mail info@automator.com www.automator.com

MANUAL REVISIONS**Number and date****Revision 15 of 14.09.2016****Revision matrix of the manual and its parts**

Manual parts		Revision				
Index		14				
Chapter	0	8.0				
	1	13.0				
	2	13.0				
	3	14.0				
	4	15.0				
	5	7.0				
	6	15.0				
	AppA	3.0				
	AppB	8.0				
	AppC	3.0				
	AppD	14.0				
	AppE	6.0				
	AppF	8.0				
	AppG	13.0				
AppH	13.0					
Appl	15.0					
Date		Sep 2016				
Signature		UT/rb				

Document:

INSTRUCTIONS FOR USE OF THE SOFTWARE

AUTOMATOR MARKING SYSTEMS SRL - Via Meucci n.8-20094 Corsico (MI) Italy
Tel 02 48601445 Fax 02 48601503 e-mail info@automator.com www.automator.com

CHAPTER 1

Introduction

1.1 Using the manual

This manual explains the use of the AC500 controller to operate the marking head. The information in this manual describes the use of the basic marking software to set the message to be marked, to mark it, and to store it.

The AC500 controller can be coupled with different types of marking heads. **While most software features are common to the various systems, some features are unique and typical of a specific marking head. Consequently, some features documented in this manual may not be applicable to your specific system.** Where variations exist, they will be clearly identified within the text to inform the operator of possible differences.

This manual is divided into the following sections:

- *Section 1-Introduction* - provides an overview of the system, explaining its terminology and concepts.
- *Section 2 - Operating System* - provides information on the controller and instructions for operating the marking system.

1.2 Overall view of the marking system

The single-point marking system allows the permanent marking of a wide range of materials such as steel, aluminium and plastic. The shape, size and position of the message to be marked are programmable by the operator using the basic software.

APD - DOT-DENSITY MARKING SYSTEMS

The dot system uses a pneumatically driven plunger that permanently marks a message by etching the surface with a sequence of dots whose density can be programmed. The software defines the characteristics of the message, including the marking coordinates within the working area.

ADP - VIBRATION MARKING SYSTEM

Marking is done by pressing the plunger head against the surface. While the software moves the plunger cartridge according to the set X/Y coordinates, the plunger tip itself moves the material to be marked, thereby forming continuous characters.

ADP - SMEAR MARKING SYSTEM

Marking is carried out by maintaining the pressure thrust of the plunger (sintered diamond tip) on the surface of the workpiece during movement along the axes. The marking pattern is called smear or scratch marking.

At the software level, the use of a smear head will be 100% comparable with the functionality made available with the vibration head. To operate a smear head, an AC500 Rev. C controller is required.

ALEA - LASER MARKING SYSTEM

The marking is carried out by means of a Yag laser beam with a wavelength of 1064nm. The software moves the cartridge, from which the beam exits, along the set XY coordinates.

A typical marking system includes the following components :

- Marking head (including head cable, cartridge and plunger)
- Filter/regulator unit (for pneumatic systems only)
- AC500 controller (including power cable)
- AC500 Controller Manual (including applicable supplements)

Marking head. The marking head consists of a guide mechanism that quickly positions the plunger cartridge/laser beam according to defined coordinates. The ADP marking head also includes electropneumatic components to move the plunger forward and backward. Please refer to the specific head supplement for further information.

Filter/Regulator for ADP. For pneumatic systems, the filter/regulator unit controls the air pressure at the marking head. For most pneumatic systems, the unit includes a regulator with a pressure level indicator to control the forward air to the marking head. The forward air is used to push the plunger out of the cartridge; the plunger uses a spring system to return into the cartridge. **However, this unit is insufficient if the customer's air supply does not guarantee clean, dry air.**

Controller. The controller contains the basic software and generates commands to move the marking head. The controller consists of a 7" widescreen touch screen display and provides the user interface to the marking system, while allowing the operator to select commands and make choices exposed by the basic software. The controller's back panel includes a *Marker* port for connection to the marking head, a *Host* port for connection to RS-232 or RS-485 (with a server computer or bar code analyser), an *I/O* port for connection to a logic controller (such as a PLC), and a *TTL I/O* port for connection to optional Automator (remote station). An *Aux axis* port is also present for systems that include an extra accessory, i.e. auxiliary axis (such as Zeta or Theta axis).

System Software. The basic software is permanently installed in the controller. It offers the user interface schemes. The software also offers memory for storing data, loading and compiling customised marking templates.

1.3 Terminology

Programmes and Fields

The marking system uses *programmes* to mark objects. *The programmes* are files stored in the controller's memory. These files (extension *.prg*) can be

imported or exported via a USB key. Each *programme* contains one or more *fields*. A *field* defines one object and how it will be marked. Where applicable, *fields* may define text sequences, geometric objects, symbols, Goto commands or Pause commands. Text sequences may include alphanumeric characters, symbols, and special commands called *flags*, which automatically insert data into the text sequence, such as serial numbers, times, and dates according to a user's own table.

Marking area

The marking area is the rectangular area where programmes can be marked by the system. The area is defined by the movement of the marking head along its X- and Y-axes. The size of the marking area is determined by the physical limitations of the head. If a programme contains a field that is outside the marking area, the system will display a communication when attempting to mark it, so it will be corrected. Please refer to the specific *System Head Supplement*.

Workpiece positioning

It is important to understand the arrangement of the part to be marked in relation to the marking head. Different *programmes* are designed to mark objects in precise positions within the marking area. If the part is incorrectly positioned in relation to the marking area, marking will not occur as expected. The position of each *field* is defined by the coordinates on the X and Y axes, relative to the 0.0 point of the area. Since the marking head uses this reference point to locate and mark objects, and since all objects must reside within the marking area, the operator can thus understand how critical the correct alignment of the part within the head area is for satisfactory marking.

Calibration position

The calibration position of the X/Y axes is set in production and cannot be changed. The marking head moves to the calibration position whenever the machine is brought online. This allows the system to zero and reset its position in relation to the marking area. Optionally, the system can be programmed to immediately move the marking head to the calibration position before or after marking.

Parking location

The parking position can be defined by the operator. Often the parking position is used to move the plunger out of the marking area so that the marked workpiece can be easily removed or secured to the base. Consequently, a parking position can be defined differently for each *programme*. The marking head will move to its park position when the system finishes marking a programme or when a park command is given to the operator.

Prepositioning

The pre-positioning command transports the plunger to the anchor point of the first field to be marked. The pre-positioning command can be used to set the marking head before marking.

Operational levels

The marking system can be used in four modes: User, Supervisor, Distributor and Manufacturer. Access to the Supervisor, Distributor or Manufacturer mode is limited by the introduction of a password protection, to restrict access to authorised personnel.

User Enables programmes to be loaded and marked. See Section 2 for details on how to operate the marking system.

Supervisor Allows all choices available to the user, with the addition of creating or editing programmes.

Distributor Allows you to make all the choices available to the supervisor, with the addition of changing general system parameters and controlling system performance. See Sections 3, 4, and 5 for details on Distributor features.

Manufacturer Allows all choices available to the distributor, with the addition of changing system axis and maintenance parameters. See Sections 3, 4, and 5 for details on Manufacturer features.

Operation Sequence

Typically, the marking system operates in the following sequence. Optionally, the equipment can remain switched on from shift to shift, so that start-up is not necessary at each change.

1. Switch on the software system;
2. Turn on the air pressure;
3. Put your head online;
4. Load the programme to be marked;
5. (optional) Park the marking head;
6. Position and secure the part to be marked;
7. (optional) Preposition the marking head;
8. (optional) Operate marking in simulation;
9. Mark the piece;
10. Remove the marked part;

11. Repeat the marking process on a new workpiece.

There is no operation to follow to produce a particular marking. The quality of the marking depends on many factors such as

- ADP - the marking depth, surface area and plunger type
- ALEA - the frequency setting and filling range

Usually, it is best to actually carry out a marking test on a sample to obtain the best combination of settings in response to individual needs.

1.4 Considerations

General

ADP - DOT-DENSITY MARKING HEADS The force of marking on a point is instantaneous and concentrated in a small area. This minimises the total impact force on the surface to be marked.

ADP - VIBRATION MARKING HEADS. Keeping the forward air pressure low, with minimal plunger force, will result in quality marking.

ADP - SMEAR MARKING HEADS. The thrust pressure setting of the diamond plunger must be set according to the type of workpiece material to be marked.

-
ALEA - LASER MARKING HEADS. There are two parameters of ALEA laser marking: tension, with a value between 0 and 10; and filling, with several choices that influence marking times accordingly

ADP - Plunger stroke

The distance between the plunger in the rest position and the surface to be marked is known as *plunger stroke*.

DOT OR DOT-DENSITY MARKING HEADS. The plunger stroke is a major factor in determining the depth of marking. The plunger accelerates in its stroke, realising the greatest speed at the moment of striking. Therefore, as the plunger stroke increases, so does the marking depth and dot diameter. The stroke of the plunger also has an effect on the marking speed: as the stroke increases, it will take longer to make the mark.

VIBRATION MARKING HEADS. By varying the plunger stroke, the marking depth can be increased or decreased. As the plunger stroke is increased, the marking depth at the initial point of impact will increase accordingly.

ALEA - Cartridge Fire

The focus is the distance between the exit of the laser beam from the cartridge and the surface to be marked. Its default value is 40mm with a tolerance of ± 0.5 .

ADP - Forward air pressure**DOT OR DOT-DENSITY MARKING HEADS.** For systems

tyres, by varying the forward air pressure and, where applicable, the return air pressure, the marking depth and dot diameter can be increased or decreased. Obviously, the depth increases with increasing forward air pressure. In general, increasing the forward air pressure for the same stroke will shorten the basic marking time, but will also increase the dot size and marking depth.

VIBRATION MARKING HEADS. The forward air pressure is the most important factor in determining the marking depth. Variation in forward air pressure can be used to increase or decrease the marking depth. High forward air pressure and a long plunger stroke will contribute to deeper characters at the initial point of impact.

1.5 Surface to be marked

The depth of marking varies in relation to the hardness of the material being marked.

Finishing

ADP - The surface finish of a material is probably the most important consideration when selecting the type of plunger. Even on very hard material (Rockwell 40C), a very legible marking can be obtained at 0.0025 mm depth if the material has a 'smooth as glass' surface finish

ALEA - Deep marking is achieved with an appropriate filling field setting: the denser the filling, the greater the depth with longer marking times.

Hardness

ADP - Material hardness influences plunger size, cone angle and plunger material.

1.6 Plunger

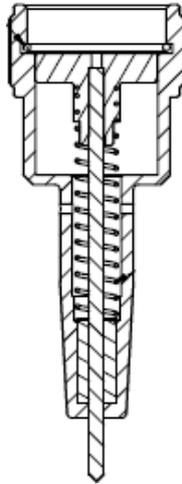
Different plungers are available in terms of material, size, length.

Size

At the same forward and return air pressure, on the same material, the heavier plunger will produce a deeper marking than the lighter plungers.

Conical angle

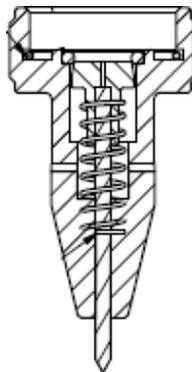
POINT MARKING HEADS. The cone angle is the angle at the plunger tip. The wider the cone angles (120°), the larger the diameter of the marked dots. Conversely, the smaller the cone angles (60°), the deeper and smaller in diameter are the dots. This marking head is used in the case of defined dot marking (5x7, 11x16, etc.).



Punch holder and punch for marking in dot mode

DOT DENSITY MARKING HEADS (CONTINUOUS). The cone angle is the angle at the tip of the plunger. The wider the cone angles (120°), the larger the diameter of the marked points. Conversely, the smaller the cone angles (60°), the deeper and smaller in diameter are the dots. This marking head is used in the case of 'continuous' marking, i.e. dot density.

VIBRATION MARKING HEADS. The cone angle is the angle at the tip of the plunger. The smaller the cone angles (30°), the wider the marking groove. Conversely, the larger the cone angles, the deeper the marking grooves.



SMEAR MARKING HEADS. The cone angle is only available at 90 degrees. The tip is made of sintered diamond.



CHAPTER 2

The operating system



2.1 Using the controller

The controller contains the basic software and generates commands to control the marking head. The controller consists of a 7" widescreen touch screen display.

There is a single button  on the controller that allows the display to be switched on or off.

NOTE

Pressing the control on/off button will disable all outputs controlled by the control. This will make it possible to forcibly interrupt

the execution of the machine cycle in an emergency.

There are also three LEDs on the controller to indicate the status of the machine and controller:

- **Blue LED:** indicates the power status of the controller. If the LED is lit, it means that the controller is correctly powered.
- **Green LED:** indicates the online status of the machine. If the LED is lit, it means that the system is ready to mark.
- **Red LED:** indicates any alarm or offline system status. If the LED is lit, it means that the system is in an offline or alarm state and therefore no marking can be made in this condition.

2.2 Start the software

NOTICE TO ADP

Ensure that the operator has connected a suitable filter to the marking system and the company supply system. Failure to do so may result in damage to the system

NOTICE

Make sure the Controller is switched off, before connecting or disconnecting to the Marking Head. This could result in damage to the controller itself.

1. Power the controller.
2. Wait for the system to boot. At this stage, a screen will appear indicating that the system is waiting to load.
3. At the end of booting, the software will be started.
The system will perform a test to check the compatibility of the software system. If the test is unsuccessful, an error message will appear on the display indicating the procedures to be performed to configure the system correctly.

After firmware/kernel/software compatibility is confirmed, the system will display the Main Operator Screen. Power will be supplied to the marking head.



Software main screen

The table at the top left of the main screen of the software shows the programme that is to be executed. Each line in the table corresponds to a field in the programme. The columns of the table contain information related to the field type, the reference position within the work area and any details for the current field information (these vary depending on the field type). Above the table is a field for displaying the name of the currently loaded programme.

At the side of the table is another field that allows the display of the anchor point linked to the programme (expressed in X and Y coordinates).

At the bottom right of the programme area, there are fields for predicting the execution time of the programme and the number of repeated executions of the programme.

At the bottom of the programme table, there is a scroll bar which allows you to change the speed at which the programme runs. According to the value set in the slider bar (value in percent and shown next to it), all speeds set by the parameter environment will be related by the set percentage value.

NOTE

It is not possible to delete programme fields from the table. Only certain types of fields can be modified from the table.

1. 2. 3. 4. 5. 6. 7.

The screenshot shows a software window titled "TEST 5090". It contains a table with the following data:

	TYPE	POSITION	
1	DXF	X=0.00; Y=50.00	PROGRAMMA-5090.dxf (NORMAL)
2	TEXT	X=5.51; Y=45.02	12345 (NORMAL)
3	DATAMATRIX	X=70.98; Y=43....	12345 (NORMAL)
4	TEXT	X=54.37; Y=9.34	ADP5090 (NORMAL)
5	TEXT	X=6.11; Y=8.68	ABCDE (NORMAL)

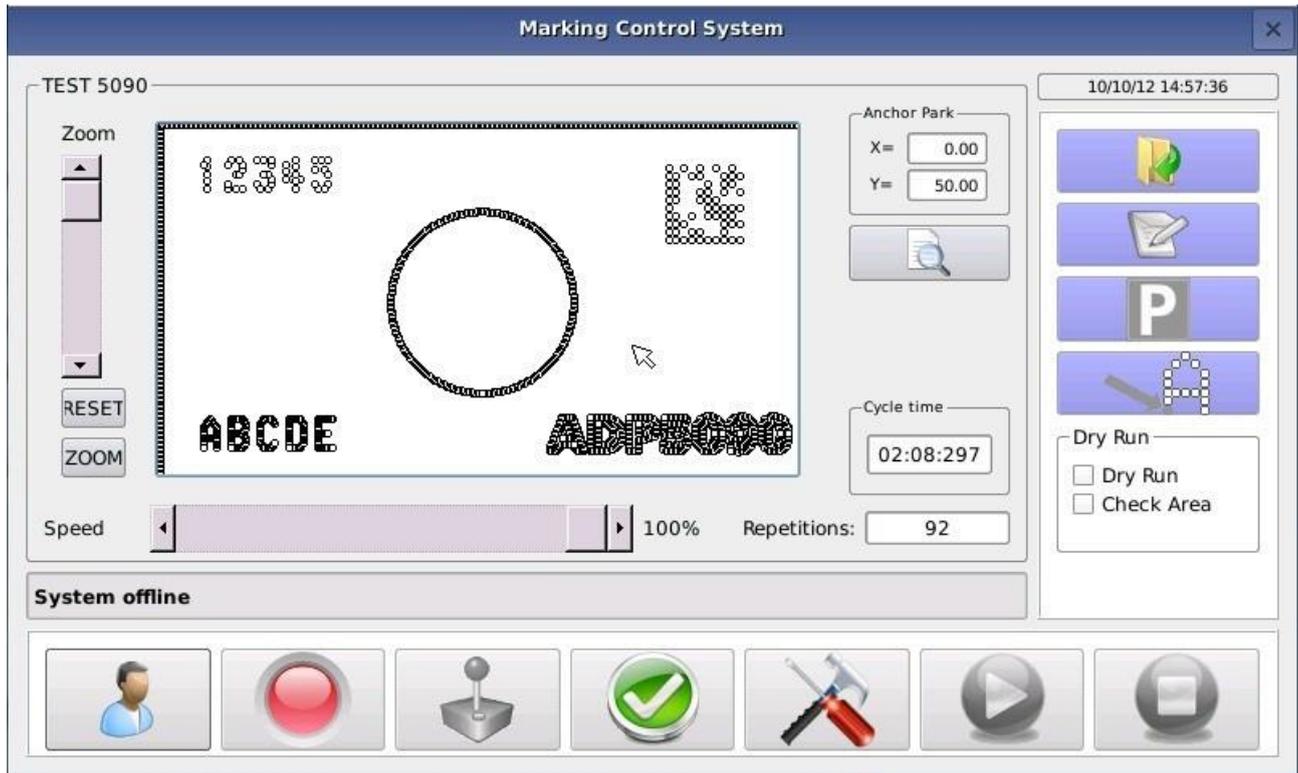
Below the table is a speed slider set to 100% and a "Repetitions:" field with the value 92. To the right, there is an "Anchor Park" section with X=0.00 and Y=50.00, a magnifying glass icon, and a "Cycle time" field with the value 02:08:297.

1. Name of current programme.
2. Table of the current programme.
3. Slider bar for setting working speed.
4. Coordinates of the anchorage point.
5. Button to recall preview of current programme.
6. Estimated time for the execution of the current programme.
7. Number of completed executions of the current programme.

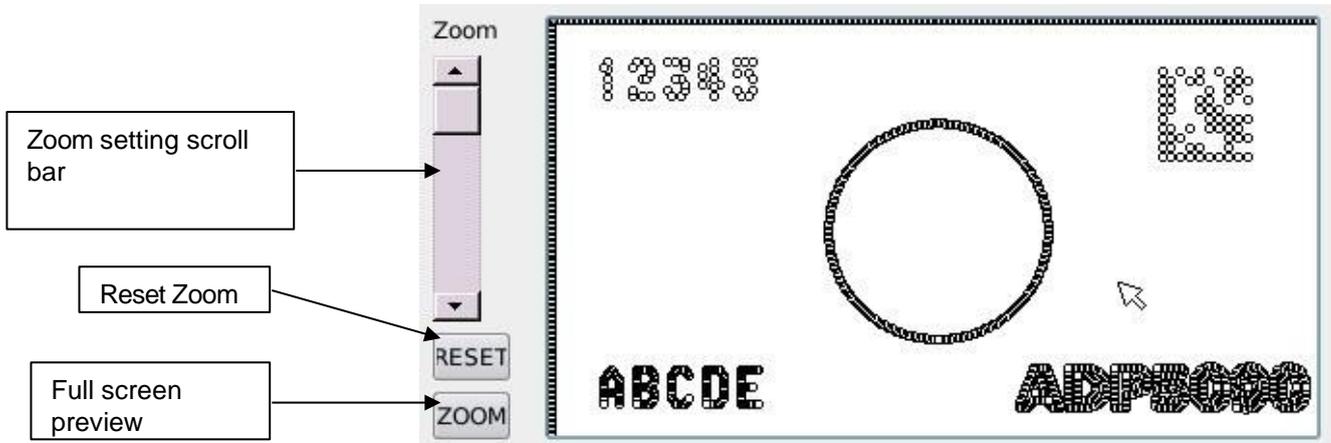


The  button allows you to call up a 2D preview of the currently loaded programme. When running the programme (in simulation or marking cycle) it will be updated with the fields being marked and those already completed.

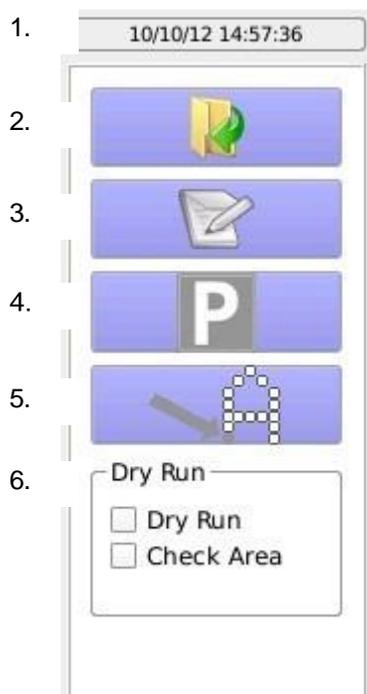
From the programme preview, it will be possible to *zoom into* the work area of the current programme. To restore the default zoom (i.e. over the entire work area), simply click on the reset button.



Main software screen with programme preview



The button bar on the right-hand side of the main screen contains some functionality for programme management and cyclic marking.



1. Current date and time of the operating system.
2. Button for loading a programme in the controller's memory.
3. Button to enter the programme editor environment.
4. Button to position the marking head at the parking level.
5. Button to position the marking head at the pre-positioning height (corresponding to the start of the programme).
6. Cycle management tab: possibility to enable cycle in simulation or area check.

NOTE

The cycle management and marking head positioning functions can be disabled if the system is *in offline mode*. If the system is correctly set up *online*, the functionalities are enabled.

NOTE

The programme editing functionality is only enabled if the user level is set as *supervisor*, *distributor* or *manufacturer*.

At the bottom of the main screen are buttons for managing the marking head and user protection level.



1. Label for displaying current status of marking head (*online/offline*).
2. Button for managing the user protection level.
3. Button for managing the status of the marking head.

4. Button to enter the jog environment.
5. Button for displaying system alarm status.
6. Button for access to auxiliary software functions.
7. Button to start/stop the marking cycle of the current programme.

NOTE

On the software screens, there is always a button with

this icon , with which you can exit the currently displayed screen.

2.3 Setting the marking head status (ONLINE/OFFLINE)

The current status of the head (online/offline) is shown both on the controller and on the display.



Offline head situation

**Online head situation**

In the case of an offline head, the following displays will be present on the controller and on the display:

- The *red LED* on the controller will be on and the *green LED* off;
- The message 'System offline' will be displayed on the machine's status label;
- The display will show the machine's current status button with the 'off' icon ;



- The cycle start, cycle stop and jog buttons will be disabled on the display;

To put your head in the *online* state:

- From the main menu, press on the machine's current status button ;
- The marking head will move towards the calibration position.



In an *online* head situation, the following displays will be present on the controller and on the display:

- The *green LED* on the controller will be on and the *red LED* off;
- The display will show the message 'System online' on the machine's status label;
- The display will show the machine's current status button with the 'on' icon ;



- The display will enable the cycle start, cycle stop (only if the cycle is running) and jog buttons;

While the system is online, programmes can be marked and the marking head can be moved using software commands.

To put the system in the *offline* state:

- From the main menu, press on the machine's current status button .



While the marking head is offline, all marking, pre-positioning and parking commands will be disabled.

2.4 Loading a programme

In order to run and mark a programme, it must be loaded from the controller's memory. If no programme is currently loaded in the software, a situation like this will arise:



Main screen with no programme loaded

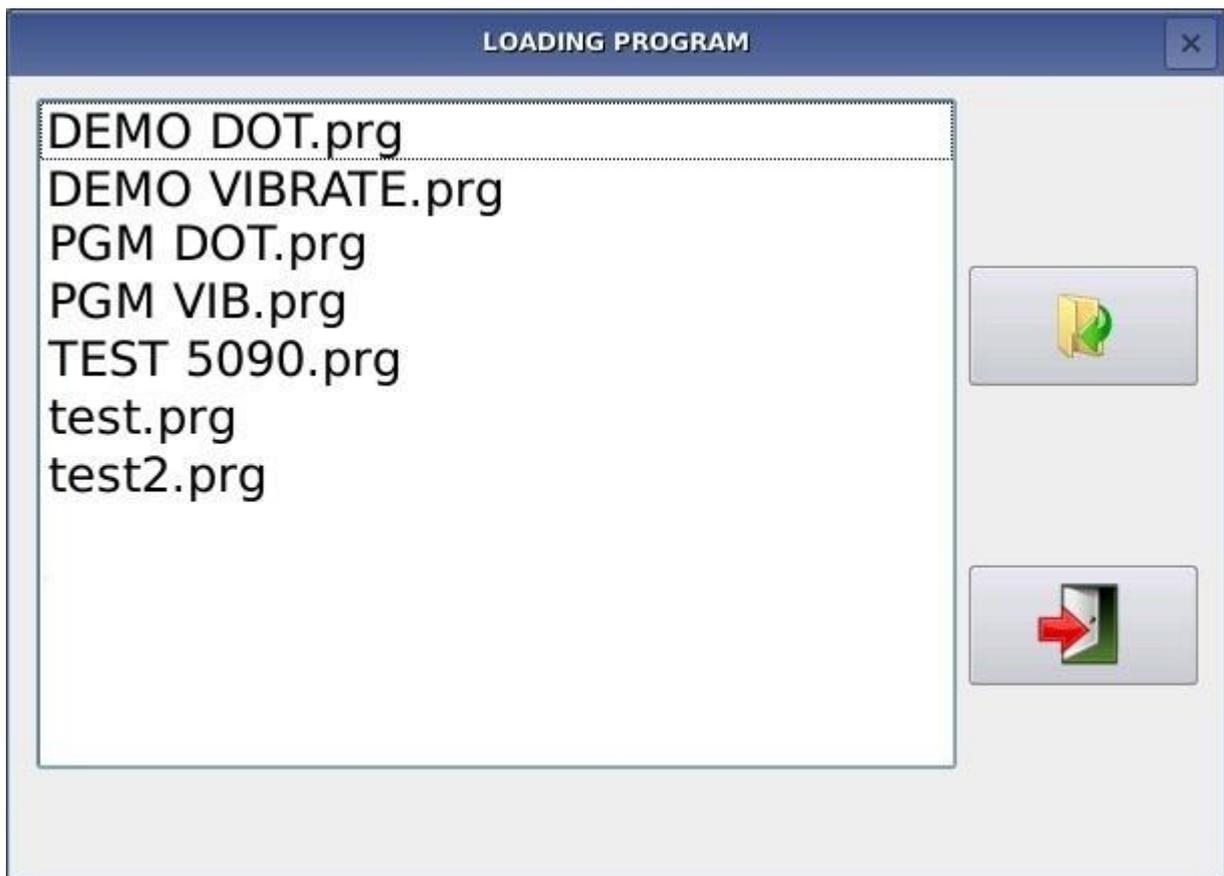
In this situation, it will be impossible to launch the execution of a marking cycle even if the head is in the *online* state.

To load a programme:

- From screen main screen press the button of load programmes



- A new window will open in which the programmes in the controller memory will be displayed



Programme loading screen

- Select the programme you want to load from the list. If there are many programmes in the memory, a scroll bar will appear to the right of the list.
- Once you have selected the programme you wish to load, press the load button to

the right of the screen . 

In the example shown, the programme 'test.prg' will be loaded.

The programme loading screen will close and the system will return to the main screen of the software with the programme loaded in the dedicated table.



Main screen with programme loaded

2.5 Park the marking head

Each programme has a parking position defined in the programme design. Often, the parking position is used to move the head out of the marking area so that the operator can easily remove or fix the marked part.

NOTE

The head must be Online and the message loaded for it works the command parking

To park the marking head:

1. Load a programme.
2. Set the marking head *online*.
3. From the main screen, press the parking button  to park your head according to the coordinates already defined.

The head will move towards the parking position.



If the *Automatic Parking* parameter has been enabled (see *general parameter* description in this manual), the system will automatically bring the marking head to a parking position whenever the system is brought online, or when a new programme is loaded (if the system is already in an online state).

2.6 Prepositioning the marking head

This command moves the plunger to the first field to be marked in the loaded programme. This command is mainly used to align the marking head to the material to be marked, before the start of the marking cycle.

To pre-position the marking head:

1. Load a programme.
2. Set the marking head *online*.
3. From the main screen, press the pre-positioning button  to position the head at the pre-positioning point.

ATTENTION



With the *pre-positioning* command, the marking head performs a movement. Before confirming the command, check that all safety precautions have been taken in terms of clothing and personal hearing and sight protection for the operator.

2.7 Manual data entry

In some programmes, manual data entry is required before marking. These fields are marked with a coloured line in the table.



Example of variable text in the second line of the programme

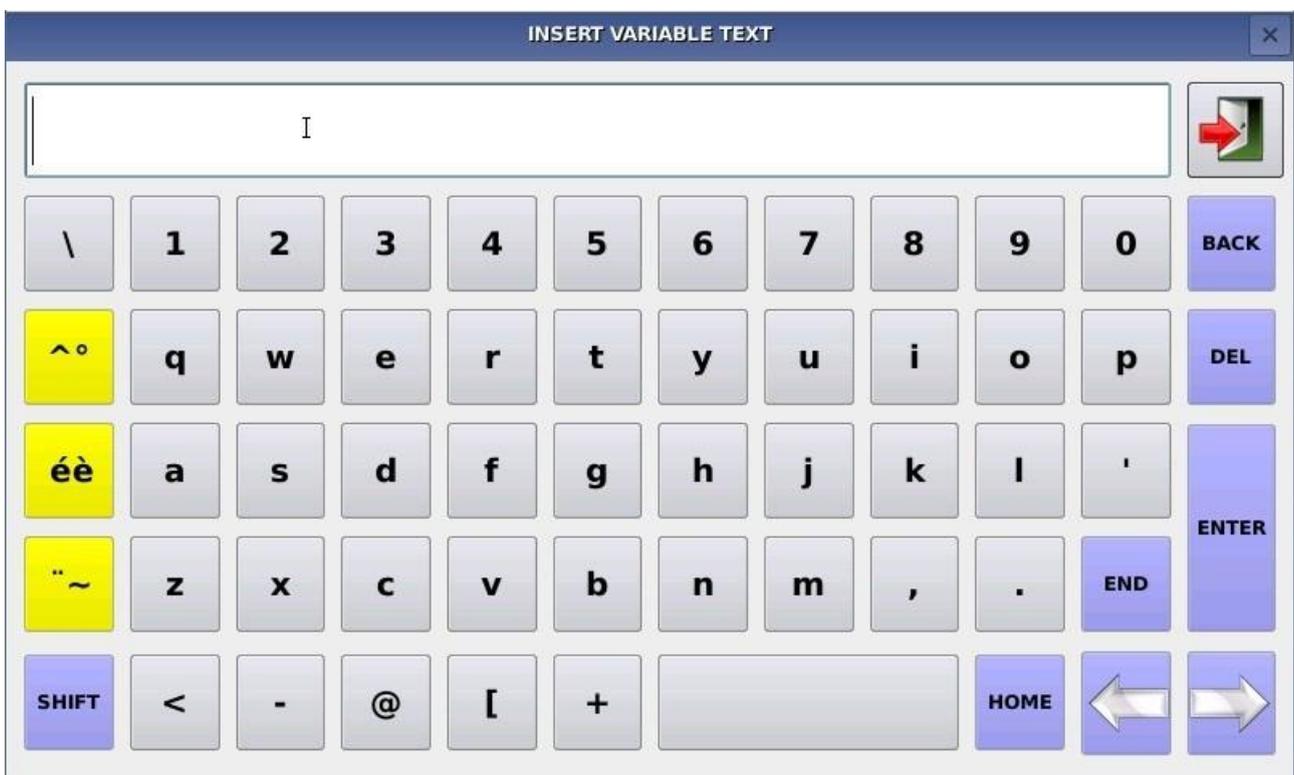
Variable fields are identified with the following syntax within a text field:

%V

When compiling the programme, the text '%V' will be replaced by the text entered in the variable.

To insert variable text into the message:

1. Press the line containing the text variable.
2. The system will open the alphanumeric keyboard to allow the desired data to be entered



The alphanumeric keyboard

3. *Enter* the desired data and press the *Enter* button on the keyboard. For example, enter the text 'var'.



The alphanumeric keyboard

4. From the main screen we can see the variable field updated with the data we have just entered. In particular, the data will be shown in the third column of the table at the changed variable row.



Main screen with updated variable field

NOTE

If a variable field has been set up with automatic variable cleaning management, at the end of a cycle the set value will be cleared and then the variable will assume a null value.

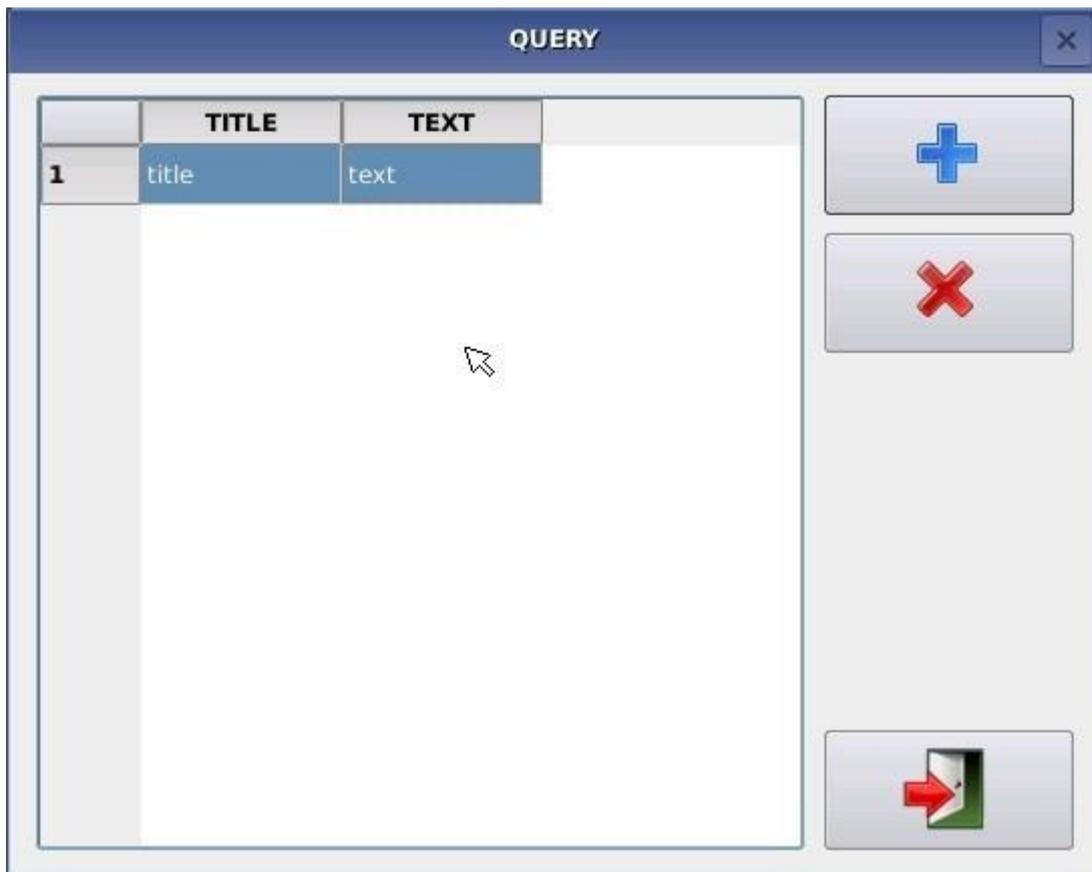
2.8 Text with query

The programmes can be composed of special flag commands in order to define several texts with one common base. Query text is stored in a dedicated table that can be created and edited by the *supervisor*, *distributor* or *manufacturer* user. Queries are formed by a title and a text. To edit the query table, it is necessary to go to the auxiliary functions menu as



supervisor, *distributor* or *manufacturer* user and press the *query* button .

The software will display the query setting table on the screen.



Query setting table

Each row in the table represents a query with two values associated with it: the *title* and the *text*. By clicking on one of the two table cells, you will be able to change the corresponding

query value. It is also possible to insert a new query using the  button and delete the selected query with the  button.

If there is text in the programme that uses query command flags, a new button  will appear on the main screen.

Pressing this button will display the query table. In this table, only the query *text* field may be edited, and no queries may be inserted or deleted from the table.



Main screen with a programme loaded with a query in the second field

Queries are identified with the following syntax within a text field:

%[number]q or *%[number]Q*

where *[number]* corresponds to the query number, *q* to the query title and *Q* to the query text.

In the example shown, there is a text field with the message *'%1q_%1Q'*. This means that when compiling the programme, the flags *'%1q'* and *'%1Q'*,

will be replaced by the *title of the first query* ("%1q") and the *text of the first query* ("%1Q").

The result you will get, based on the values set in the query, is as follows:



Main screen with preview of the query of the second field

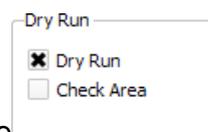
2.9 Vacuum marking (cycle simulation)

It is possible to simulate the marking of the loaded programme without actually marking the workpiece. This method is called '*empty marking*' or '*cycle simulation*'. With this command, it is possible to check whether the part to be marked is correctly positioned in relation to the message to be marked.

To make a vacuum marking:

1. Load a programme;
2. Putting your head online;

3. Enable cycle execution check *in idle marking mode*





4. Launch execution via the cycle start button on the display or from the external Start.



Main screen with vacuum marking cycle in progress

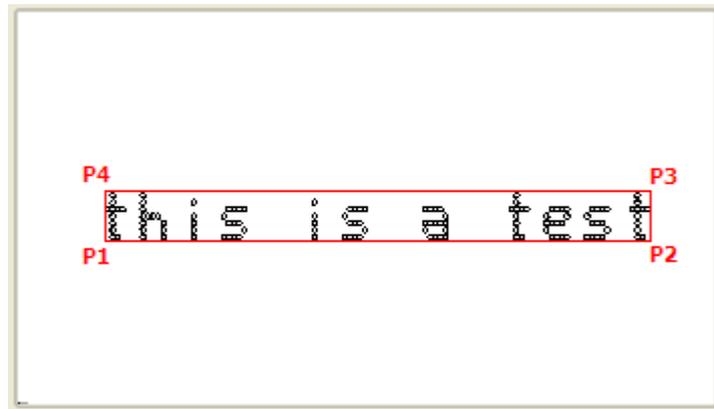
ATTENTION



With the *marking* command *empty*, the marking head performs a movement. Before confirming the command, check that all safety precautions for the operator have been observed, who must leave the working range of the head unobstructed.

2.10 Area check cycle

The *area check* cycle is a similar functionality to the *vacuum marking* cycle and allows the operator to check the limits within which the programme will run. With the *area check* cycle, nothing will be marked on the part as with the *vacuum marking* cycle.



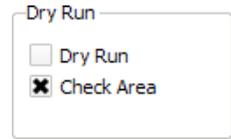
Example of an area check cycle

In this cycle the marking head will position itself in the four limits of the area occupied by the programme and execute them in clockwise direction starting from the lower left limit. In the example shown, the cycle will be:

1. Positioning the head on point P1;
2. Positioning the head on point P2;
3. Positioning the head on point P3;
4. Positioning the head on point P4;
5. Positioning at the parking co-ordinate of the programme.

To follow an area check cycle:

1. Load a programme;
2. Putting your head online;



3. Enable cycle execution *area check*



4. Launch execution via the cycle start button on the display or from the external Start.



Main screen with area check cycle in progress

ATTENTION



With the *area check* command, the marking head performs a movement. Before confirming the command, check that all safety precautions for the operator have been observed, who must leave the

NOTE

The cycle setting in the check area is exclusive to the empty marking cycle setting. This means that only one of the two modes may be selected at a time.

2.11 Marking a programme

Before marking a message, check the following points :

- Exact adjustment of supply and return air
- correct positioning of the workpiece
- plunger stroke adjustment

ATTENTION

When the marking head is placed online, it will move to the calibration position.

To mark a programme:

1. Load a programme;
2. Putting your head online;
3. Ensure that the material to be marked and the programme are ready for marking:
 - Position and secure the material to be marked.
 - Enter variable text, if required.
 - Enter the query text, if required.



4. Launch execution via the cycle start button on the display or from the external Start.

ATTENTION



With the marking command, the marking head is set in motion. Before confirming the command, check that you have complied with all safety precautions in terms of clothing and personal hearing and sight protection for the operator. Do not obstruct the working area of the head by inserting hands or anything else under the plunger.



Main screen with marking cycle in progress

2.12 Marking time

On the main screen, the execution time of the loaded programme cycle can be displayed. The execution time is expressed in seconds with a display down to 0.001 seconds (milliseconds). The marking time is calculated from the machine data of the software, and depends on the parameters of the set handling and marking times and speeds.

2.13 Stopping the execution of a marking cycle

During the execution of a marking cycle, it is possible to force a cycle stop by pressing the



button. The button is normally disabled and is only enabled at the start of a marking cycle execution. Pressing the button will abort the cycle and the marking head will stop in the position it was in. If the parameter "Offline marking head in case of ABORT" is enabled, the system will go offline and it will then be necessary to put the marking head back online to start another marking cycle. Otherwise, if the parameter is disabled, the head will remain in the online state.

This functionality can be used in both marking and cycle simulation.

NOTE

Interrupting a running cycle will generate a *STOP BY USER* alarm and will be visible in the machine alarm report screen.

2.14 Stopping and resuming the execution of a marking cycle

During the execution of a marking cycle, it will be possible to interrupt and later resume execution.



To interrupt the execution of a cycle, simply press the  button while the cycle is running. In the aborted cycle situation, the system will be left in the online state and the marking head will remain in the position where the cycle was aborted. Unlike the cycle abort command, it will be possible to resume execution from the point where the cycle was



aborted. To resume execution of the cycle, simply press the  button.

This functionality can be used in both marking and cycle simulation.

ATTENTION

In the state of an interrupted marking cycle, the material to be marked must not be moved as this could result in the loss of references with the marking head. The result would be incorrect marking when the cycle resumes.

2.15 Full screen preview

From the work environment, it is possible to display the programme preview in full screen.

From the preview screen, simply press the  button to display a new window with the enlarged programme preview on the screen.



Full-screen preview screen

2.16 Moving the marking head

From the jog environment, it is possible to move the marking head within its maximum X- and Y-axis coordinates.

To enter the jog environment:

ATTENTION



With the Jog command, the marking head is set in motion. Before confirming the command, ensure that all safety precautions have been taken in terms of clothing and personal hearing and vision protection for the operator. Do not obstruct the working area of the head by inserting hands or anything else under the plunger.

1. Putting your head online;



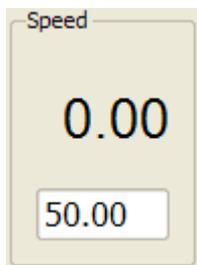
2. Press the jog button on the main screen ;

3. The jog screen will be displayed



Jog screen

4. Enter the axis movement speed in the text box provided



- 1.
- 2.

1. Current speed of the reference axis
2. User-set speed

5. To move the axes simply press the movement buttons

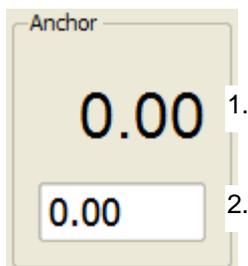


1. Movement of the X-axis to the right;

- 2.  Movement of the X-axis to the left;
- 3.  X-axis movement to the right with step;
- 4.  X-axis movement to the left with step;
- 5.  Movement of the Y-axis upwards;
- 6.  Movement of the Y-axis downwards;
- 7.  Moving the Y-axis upwards in steps;
- 8.  Moving the Y-axis downwards in steps;

6. To move the axis to a specific axis co-ordinate

a. enter the value at which you want to position the axis in the text box provided



- 1. Current share of the reference axis
- 2. User-set quota

b. press the Start movement key



The marking head will move to the set axis dimension.

It is possible to set a movement step value for step movements. To set the movement step value, simply enter a value in the appropriate header box . STEP JOG

It is also possible to stop a movement at a specific axis position by pressing the movement stop button . 

Finally, there is an additional PIN DROP button, which enables the descent of the marking tip within the environment  . WARNING: Make sure that the head is in a safe position before pressing this button.

2.17 Selecting the operating level

To select the operator operating level with which you want to control the machine, simply press the operating level management button on the main screen:

-  User mode;
-  Supervisor mode;
-  Distributor mode;
-  Manufacturer mode.

By pressing the button, the operating level mode can be changed. Supervisor mode, distributor mode and manufacturer mode can be password-protected to restrict access.

Pressing the button will display a new screen with the possible user levels to be selected on it. If one of these is password-protected, it will be necessary to enter the password in the appropriate screen in order to change the current level.



User level selection screen

User mode privileges

In user mode, it is possible:

- open a new programme in the working environment;
- insert variable text/query from work environment;
- change the marking speed from work environment;
- Place the marking head in online/offline state;
- moving the marking head from jog environment;
- view alarm history and programme statistics;
- view the scoring history;
- display memory occupancy information on controller;
- switch off the control unit from the environment.

Supervisor mode privileges

In supervisor mode it is possible:

- all the functionalities of user mode;
- access the programme editing environment;
- reset the counter for the number of repetitions;
- set operator settings;
- access programme management;
- set some general machine parameters;
- select the current language of the software;
- save system backup data;
- select the current head from the head magazine;
- configure serial port settings;
- configure barcode reader;
- apply filters in the marking history;
- exporting historical marking data;
- configure the controller's network settings.

Privileges of Distributor Mode

In distributor mode it is possible:

- all the functionality of supervisor mode;
- set all general machine parameters;
- access the calibration environment;
- access the machine's test environment;
- set maintenance data;

- reset statistics;
- carry out the system update;
- delete or restore system backup data;
- enable or disable system languages;
- configure controller network settings.

Privileges of manufacturer mode

In manufacturer mode it is possible:

- all the features of distributor mode;
- set axis data;
- set the marking head data in the head store;
- reset entries in the alarm history;
- add, delete or modify system languages;
- reset the entries in the marking history.

2.18 Display controller version

It is possible to view the software, firmware and kernel version installed on the controller.

To view the software versions installed on the controller, go to the auxiliary functionality screen and press the *about* button.



About button

A new window will be displayed with software information on it. Among the information displayed will be

- the type of machine set;
- the loaded configuration: if a valid configuration is not recognised, it will be reported as 'unknown', and if the configuration has been changed, a '*' character will be inserted at the bottom.



Controller version display screen

2.19 Switching off the controller

To switch off the controller, it is necessary to press the appropriate button located in the software information environment. This operation is particularly important because it is necessary in order to save all machine status information before switching off the power supply.

To switch off the controller, go to the auxiliary functions screen and press the *about* button.



About button

A new window will appear with software information on it.



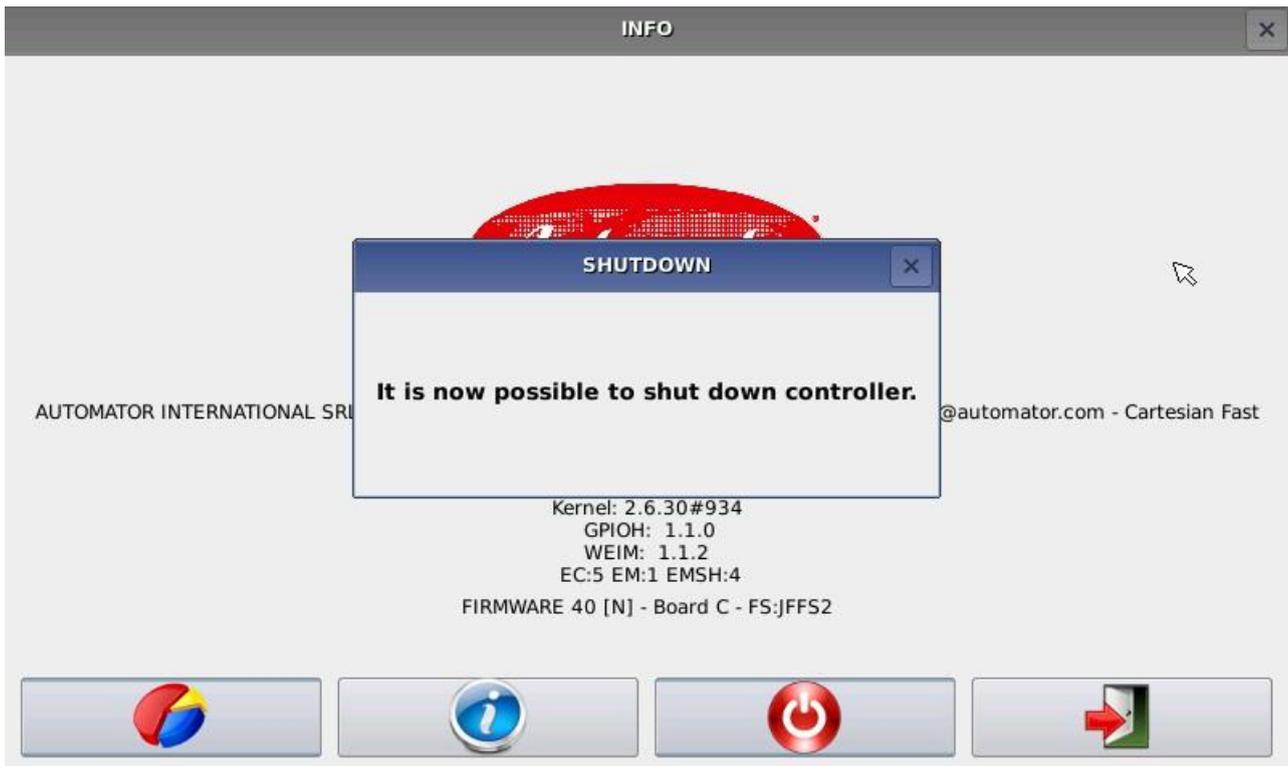
Controller version display screen

From this environment a button will be visible to proceed with the shutdown of the controller.



Controller off button

When the button is pressed, a confirmation will be requested to proceed with the shutdown operation. If you wish to continue, confirm the request. At the end of the operation, a message will be displayed indicating that power can be removed from the control unit.



Screen with completion of shutdown operation

ATTENTION

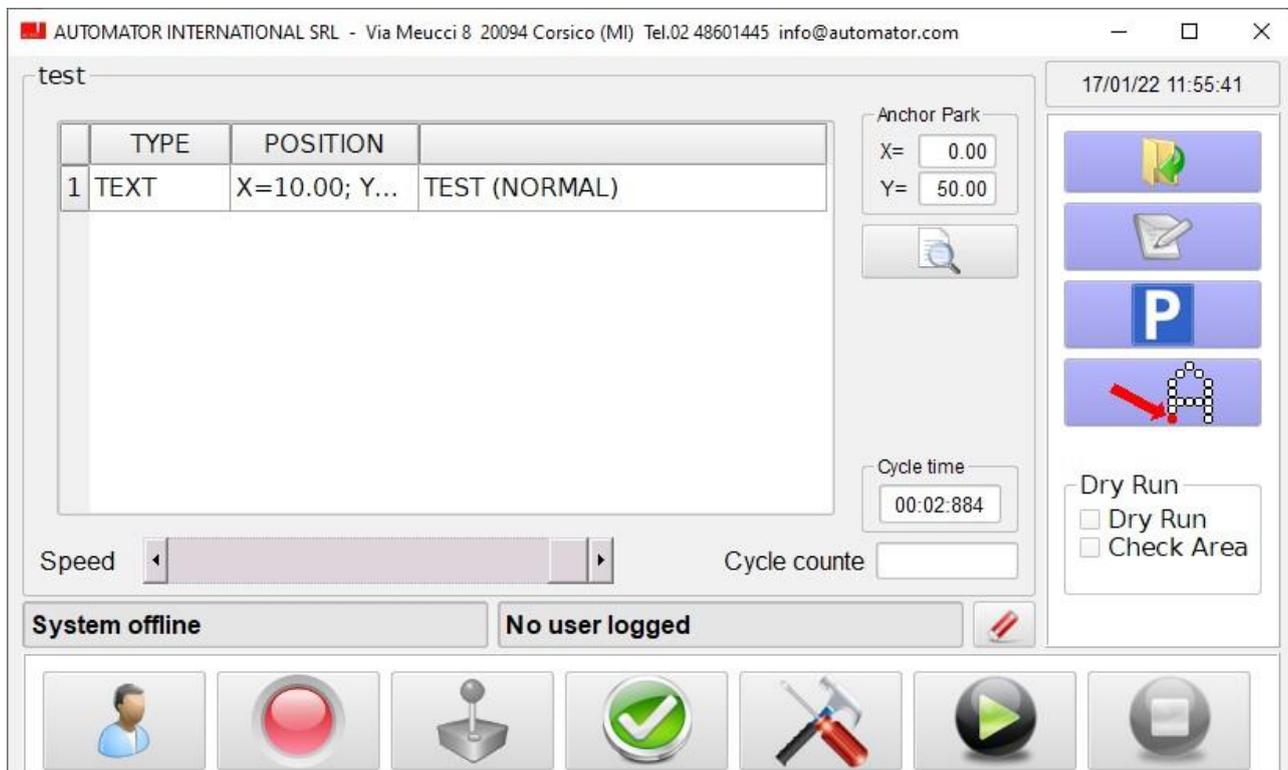
Once the shutdown procedure has been started, it will no longer be possible to navigate through the software environments.

ATTENTION

If power is removed from the controller before completion, data loss is possible.

2.20 Operator login management

If operator login management from the machine parameter is enabled, the work page will appear as follows.



A new bar will appear at the bottom of the work screen, next to the machine status bar, showing the code of the last user who logged in to the control unit.

In the event that no operator has logged in, the message "NO USER LOGGED" will be displayed as shown in the image.

The operator can access the machine as follows:

- via USB barcode: if properly configured to read the operator's code;
- manually by pressing in the status bar and entering his personal code.

The operator code will later be recorded in the marking history.

Operator access to the machine is MANDATORY in order to start marking. If an attempt is made to start a cycle WITHOUT having logged in, an error message will be displayed.



CHAPTER 3

Creating a programme

A programme is nothing more than a file containing a list of fields. Each field has parameters associated with it that are used to describe its execution and marking on the material to be marked.

There is a dedicated memory in the controller for storing programmes. These can be loaded and executed by the machine at any time. This chapter provides information on how to create, modify and manage a memory of programmes that can be marked by the machine. In particular:

- Some considerations on the programmes and the fields that form them;
- Creating new programmes;
- Modifying already existing programmes;
- Importing or exporting programmes from a USB key.

3.1 Setting the operating level

In order to be able to access the creation, modification or deletion functions of programmes stored on the controller, the operating level must be set to *supervisor*, *distributor* or *manufacturer* mode.

3.2 Basic programme considerations and fields

Creating a programme requires knowledge of a series of parameters that define the content and appearance of each field in the programme. In order to create a programme, the size of the marking area, the type of field, the font size, the font selection, the marking mode and the order in which the fields are to be marked must all be taken into account.

Dimensions of the marking area

The size of the marking area is a primary consideration in the creation of a programme. Each field that will be marked must lie, in its entirety, within the physical boundaries of the area. Otherwise, the system will display an error message when an attempt is made to mark it. You may redefine the location of the field or make slight adjustments to its dimensions or alignment parameters to reposition it

within the area. The size of the marking area depends on the parameters set in the software during configuration.

Field types

A programme may consist of one or more fields. There are different types of fields and, depending on the type, the parameters to be set when creating a field change.

Fields may be of the following types:

- Text;
- Line;
- Arch;
- Text on arch;
- BMP image;
- Datamatrix;
- DXF files;
- Pause;
- GoTo;
- Comment.

Defining field co-ordinates

The system uses X and Y co-ordinates to identify the location of each field within the marking area. You can define these values by entering the specific coordinates into the appropriate fields by hand. Or, you can move the laser plunger/cartridge to the desired position and set the coordinates of the current position of the marking head as field coordinates. Whenever you have to define an axis co-ordinate, you can interactively position the marking head at the desired location.

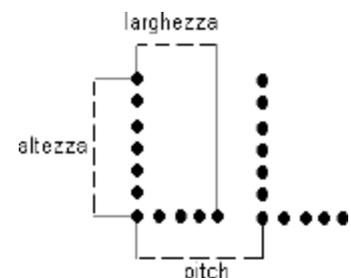
Font size

The three parameters that control the size and shape of the font are: height, width and pitch:

Height. Height is measured from the centre of the dot to the top of the character to the centre of the dot to the bottom of the character.

Width. Width is measured from the centre of the dot to the left side of the character to the centre of the dot to the right side of the character. Unless there is a specific width requirement by the operator, for a graphic proportion, it should be 66% of the character height.

Pitch or Pitch. Pitch defines the distance from the start of one character to the start of the next character. It includes the width of the character and the space between two characters. Unless there is a specific operator requirement, pitch should equal the height of the character.



Font Selection

The system offers many font selections for the text fields to be marked. The font of the character determines the marked appearance of the text. Each font selection offers a specific character set. Standard fonts for Automator marking systems are illustrated in Appendix A of this manual. Custom fonts can be loaded and are called TTF fonts. These fonts can be imported to the controller and used in the creation of a text field.

Some factors to consider when selecting a font are font size, marking speed, character identification or recognition.

Size

The desired font size must be a primary consideration when selecting a font.

DOT MARKING SYSTEMS. Very small characters may look better with a font containing a lower dot definition (e.g. 5x7). Very large characters will look better with fonts with higher dot densities (e.g. 11x16). Fonts with a higher dot density are suitable for graphic forms requiring greater definition.

VIBRATION MARKING SYSTEMS. Very small characters may look better with a font that contains fewer endings (e.g. SIMPLEX). Large characters may look better with a font that contains more endings (e.g. Complex or DIN). Fonts with more endings also give way to more polished shapes.

SMEAR MARKING SYSTEMS. Rounded fonts can have a better rendering and result in better handling along the axes. Custom fonts can be created via the Font Creator application.

LASER MARKING SYSTEMS. Very small characters can look great due to the micrometric size of the laser beam.

Speed

Marking speed is a further consideration when selecting a font. Some fonts may require a longer marking time. Font size also affects marking speed. Smaller fonts may be marked faster; larger fonts, fonts with higher dot densities and fonts with multiple endings may require longer marking times.

Character recognition

The OCR font and the PS-OCR font are specially designed to be used with the AUTOMATOR PS-OCR® Optical Character Reader. For the font to be reliably read by the Reader, a special relationship must exist between character height, width and pitch.

To achieve this, follow the instructions for setting the text size to be read by the PS-OCR Reader:

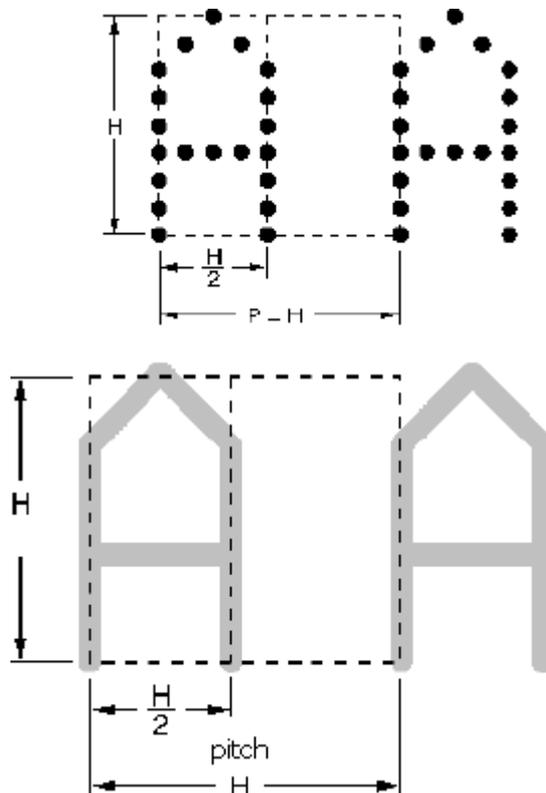
Decide the height of the characters to be marked. The height parameter will be used as the basis for establishing the height and pitch parameters.

1. Set the character width equal to $\frac{1}{2}$ the height;
2. Set the pitch equal to the height.

These settings will create a square character block (the height x the pitch) with the character occupying the first half of the pitch block. This relationship provides a height/width ratio that allows the Reader to recognise characters and offers enough space between characters to give reliable readings.

For example, if the character is to be 0.5mm high :

- Set the width to 0.25 mm
- Set the pitch to 0.5 mm



Marking Order

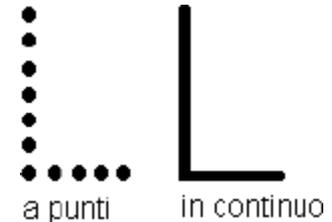
The system marks the fields of a message in the order in which they were defined in the message. That is, the head will mark the first field in its entirety before moving on to subsequent fields. Normally, this is not a problem when creating a template; however, there may be an advantage in marking fields in a particular sequence, for example to improve the efficiency of the marking cycle. If this is the case, it is still possible to reorder the sequence of fields. See below: *Defining a programme*

Marking Modes

The software offers several ways of marking the fields of its programmes. As the parameter is defined with the field, the marking mode can be specified from field to field.

Dot-matrix marking This mode marks dot-matrix characters with a style defined by the selected font.

Continuous marking With this mode it is possible to define the dot density for marking fully formed characters that produce an engraving appearance. The higher the dot density, the longer the marking time and the faster the plunger wear.



3.3 Creating a programme

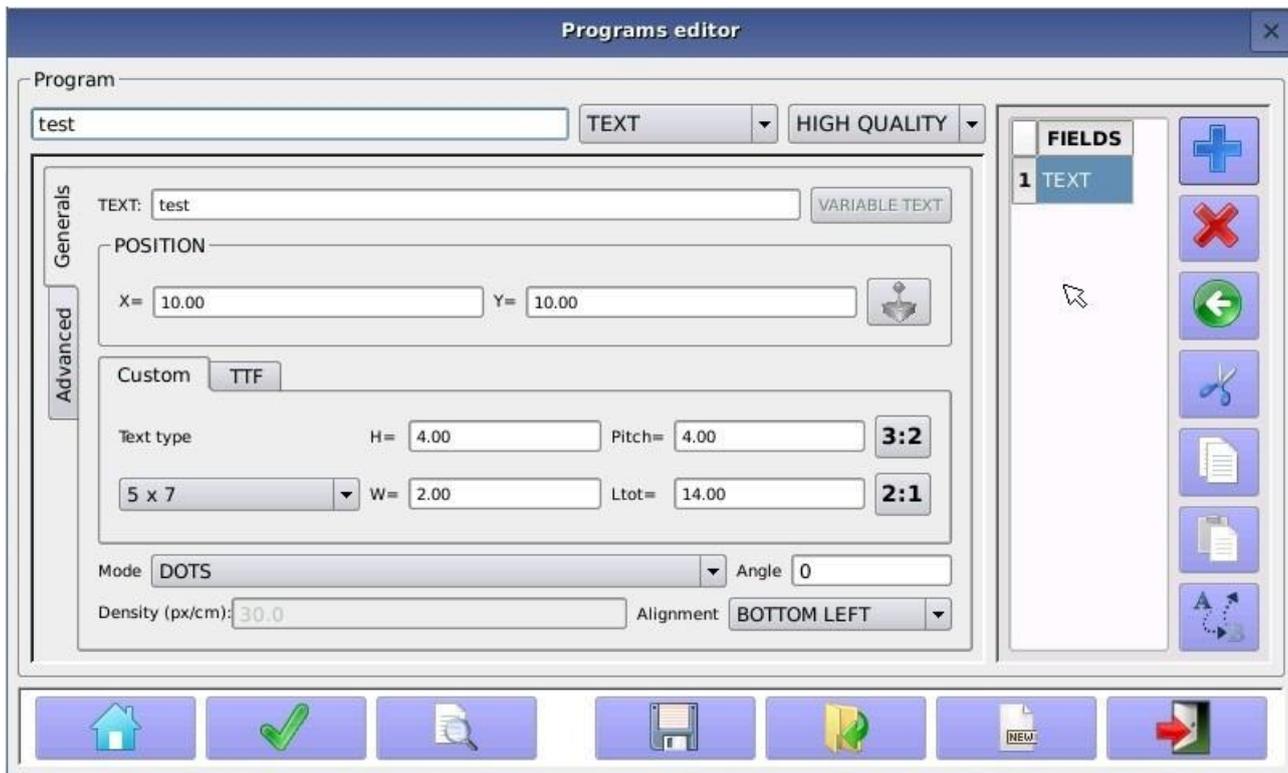
When a new programme is created, the system stores it in the controller's permanent memory. Each programme must be identifiable by a name. Once saved in memory, it can be loaded to be marked or modified whenever necessary.

To create a new programme, it is necessary to

1. Set the operating level to *Supervisor*, *Distributor* or *Manufacturer* mode;
2. From the main screen of the software, press the programme editor access button ;



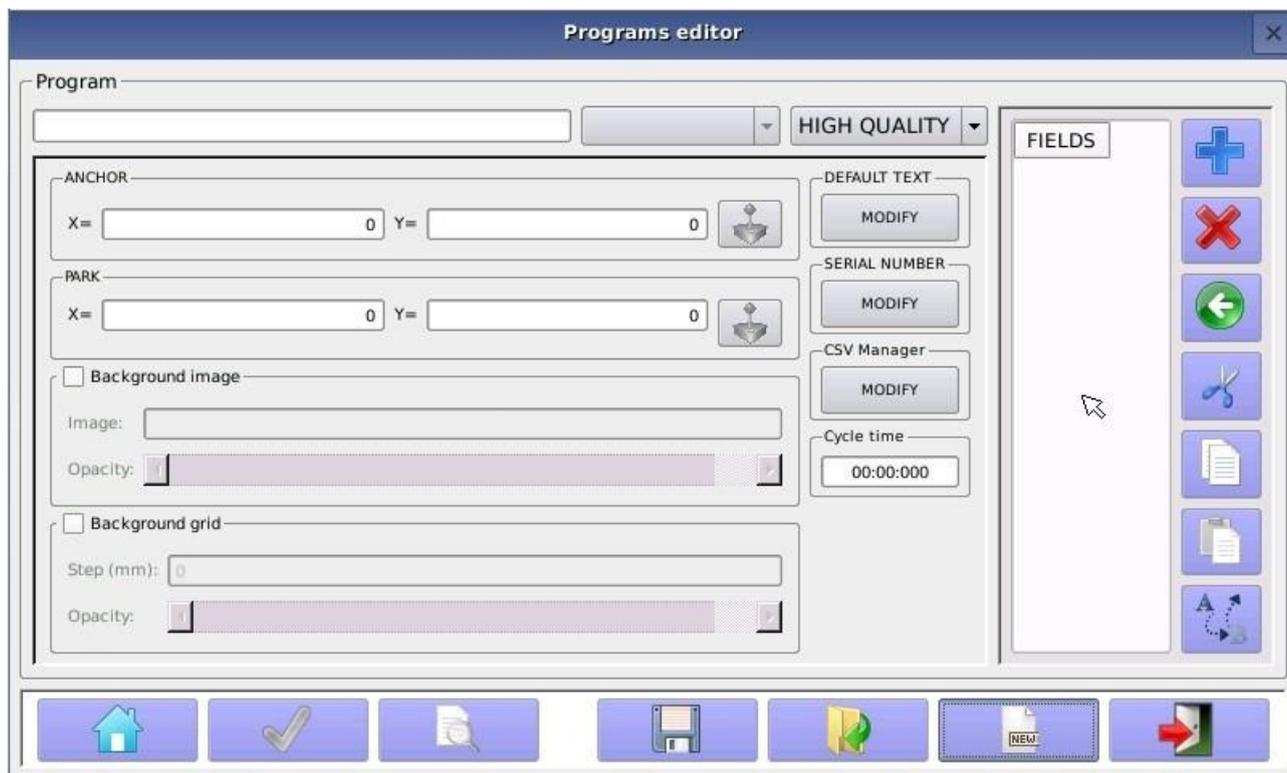
3. The software will display the programme editor environment on the screen



Programme editor screen

From the environment of editor press on button of creation new programme  ;

4. If another programme is already loaded into memory, a message window will be displayed asking whether to save the current programme or not. If you have made changes to the programme and want to save, press the confirm button ("Yes") otherwise to continue without saving press the cancel button ("No");
5. At this point, the editor environment will display a window with an empty programme.



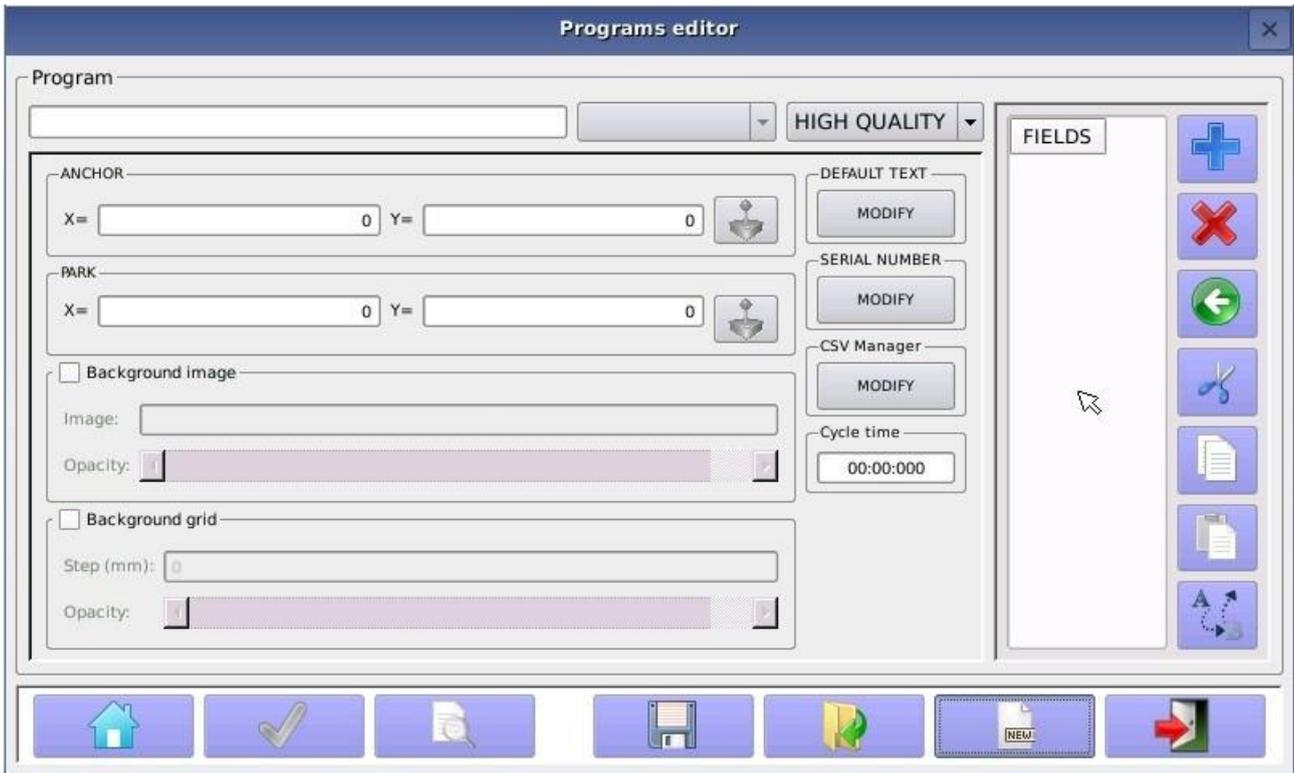
Programme editor screen with empty programme

6. Enter the name of the programme and the fields you wish to mark (see section *Setting up fields in the programme*);

7. Save the programme by pressing the save button . If a programme with the name of the current programme exists in the memory of the control, a message window will appear asking whether to overwrite the programme in the memory or not.

3.4 Programme editor screen

The programme editor environment allows programmes to be created, loaded and edited on the controller. The editor environment can only be accessed by users with *supervisor*, *distributor* or *manufacturer* operating level. The basic screen of the environment is as follows:

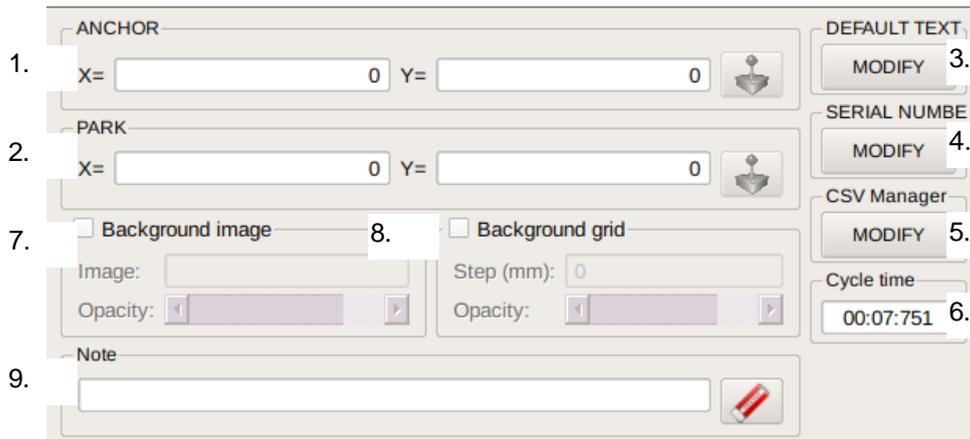


Programme editor screen

The programme editor screen consists of several parts: the general programme settings, the fields that make up the programme and the settings of each individual field.

In particular, the window consists of the following controls:

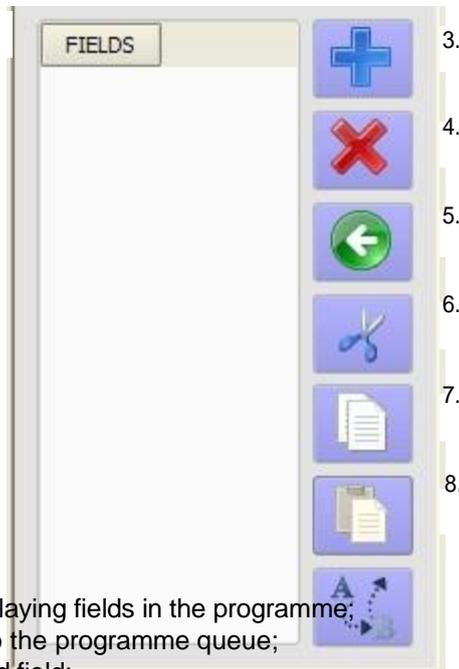
1. Text box for setting and displaying the *programme name*;
2. Drop-down menu for setting and displaying the selected *field type*;
3. Drop-down menu for setting and displaying *the marking quality* of the selected field;
4. Area dedicated to setting and displaying data related to the selected programme and field.



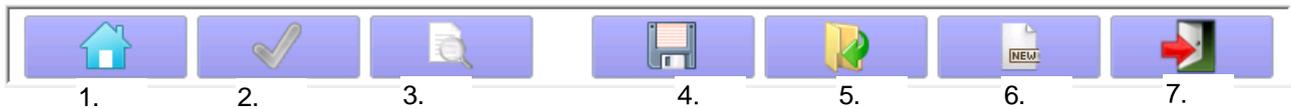
1. Text boxes for setting and displaying the *anchor point* of the programme;
2. Text boxes for setting and displaying the programme's *parking point*;
3. Button to enter the setting and display environment of the default text field data;
4. Button to enter the setting and data display environment for the serial number of the programme.
5. Button to enter the setting and display environment for CSV field data.
6. Estimated execution time for the current programme.
7. Data for setting the background image of the programme preview.
8. Data for setting the background grid of the programme preview.
9. Programme notes setting box displayed each time the programme is loaded.

1.

2.



1. Table for selecting and displaying fields in the programme;
2. Button to add a new field to the programme queue;
3. Button to delete the selected field;
4. Button to insert a new field at the current position in the programme table;
5. Button to cut the selected field;
7. Button to paste a previously cut or copied field;
8. Button to find and replace a particular text with another one set by the user.



1. Button for displaying programme data;
2. Button to launch a compilation of the current programme;
3. Button to preview the current programme;
4. Button to save the programme in the controller's memory;
5. Button to load a programme from the controller's memory;
6. Button to create a new programme;
7. Button to exit the editor environment.

The following sections will explain in detail all the possible functionalities that the programme editor environment offers.

3.5 Setting programme parameters

Certain parameters and choices can be defined and can be stored in the programme. Unlike system parameters that affect all models and operations, programme parameters are only specific to the currently loaded programme. Programme parameters can be defined when the programme is created or can be changed at a later date.

Message parameters can be used for :

- define the anchor point of the programme;
- define the parking position of the programme;
- set the serial number of the programme;
- configure settings for data to be marked from CSV files;
- define the default parameters of the text field;
- set a background image for the programme preview;
- set the background grid for the programme preview.

3.6 Defining the anchor point of the programme

The programme fields are defined by co-ordinates referring to the bottom left corner of the marking area. However, when setting the programme to be marked, difficulties may arise if it proves necessary to move the fields at a later date.

The software offers the possibility of setting the coordinates of the anchor point, to which the X and Y definition of the individual fields refer, in such a way that by varying its values on the axes, the fields referring to it also automatically move accordingly.

Suppose the fields have already been defined in the programme and it turns out that they all need to be rearranged. Without the anchor point, all fields must be moved, one by one, according to the new X and Y co-ordinates; otherwise, the

several fields in a simpler way, by hooking them to a single anchor point, varying the coordinates of the latter.

Thus, when the anchor point of a programme is changed, the system will transfer the first field of the programme to the new position and consequently all other fields of the same programme.



To set the anchor point:

1. From the programme editor screen, set the coordinates of the anchor point in the dedicated text boxes

ANCHOR

X= Y= 

The coordinates must refer to the 0.0 point of the marking area and must lie within it.

It is also possible to acquire the anchor point via the jog environment. To proceed

with the acquisition, simply press the  button.

NOTE

The marking head must be set to online status in order to position the fields with the jog.

2. Save the programme.

NOTE

After adjusting the X and Y co-ordinates of the anchor point, check with the *compile* button that the changes have not taken the fields out of the marking area.

3.7 Define head parking point

Each programme has the option of defining a parking zone, where the head will move once the marking cycle is complete, for example to facilitate loading and unloading of the material to be marked. The parking command may differ from program to program and must include the co-ordinates of the 2 axes, X and Y.

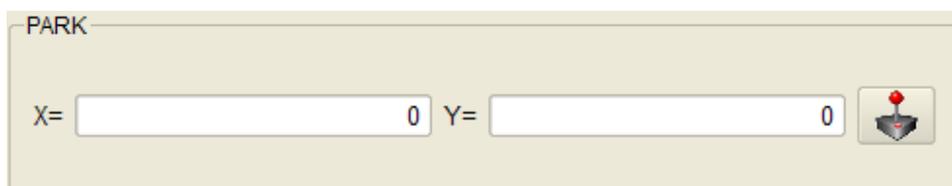
If the auxiliary axis (Z-axis or Theta-axis) is enabled, it will be possible to set the parking quota for this axis as well. With the Z-axis, it will also be possible to define the logic between the XY-axis and Z-axis movements. so that parking movements can be customised. The setting of this data influences which axis will be given priority (whether Z or XY) in the parking movement. The logic set for moving Z will also be used for the pre-positioning movement.

NOTE

The head must be online and the programme loaded for the parking command to work.

To set the parking point:

1. From the programme editor screen, set the coordinates of the parking point in the dedicated text boxes



The coordinates must refer to the 0.0 point of the marking area and must lie within it.

It is also possible to acquire the parking point via the jog environment. To proceed with the acquisition, simply press the  button.

NOTE

The marking head must be set to online status in order to position the fields with the jog.

2. Save the programme.

ATTENTION



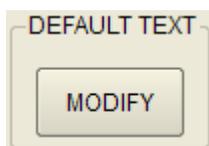
With the *park* command, the marking head is set in motion. Before confirming the command, check that you have complied with all safety precautions in terms of clothing and personal hearing and sight protection for the operator. Do not obstruct the working area of the head by inserting hands or anything else under the plunger

3.8 Define default text field parameters

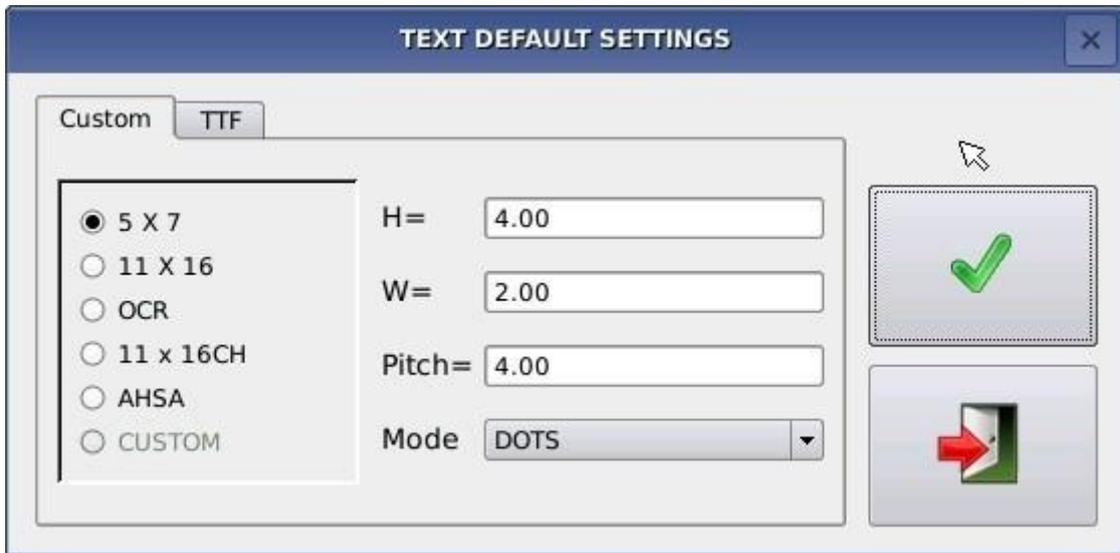
When creating a text field in a programme, the system automatically provides certain default values. These values are used to set the size, shape and graphics of the text to be marked. It is of course possible to vary these values according to the operator's needs.

To set the default parameters of the text field:

1. From the programme editor screen, press on the text default parameter configuration button



- The software will display a window on the screen allowing the selection of default parameters to be set in the text field



Text field default parameter configuration window

The configuration parameters are divided according to the font style of the text field:

- **Standard fonts**

It will be possible to set the default characteristics of text fields with standard fonts. The characteristics vary from the point density of the font, to the height, width and pitch dimensions of the font, to the way the text is marked.

- **TTF**

It will be possible to set the default characteristics of text fields with TTF fonts. The characteristics include the font type (selectable from those in the controller memory) and the font height.

- Once you have set the default values you want to use for the text fields, simply save them. If you wish to keep the changed data as default parameters, press the



button. Otherwise, if you wish to keep the

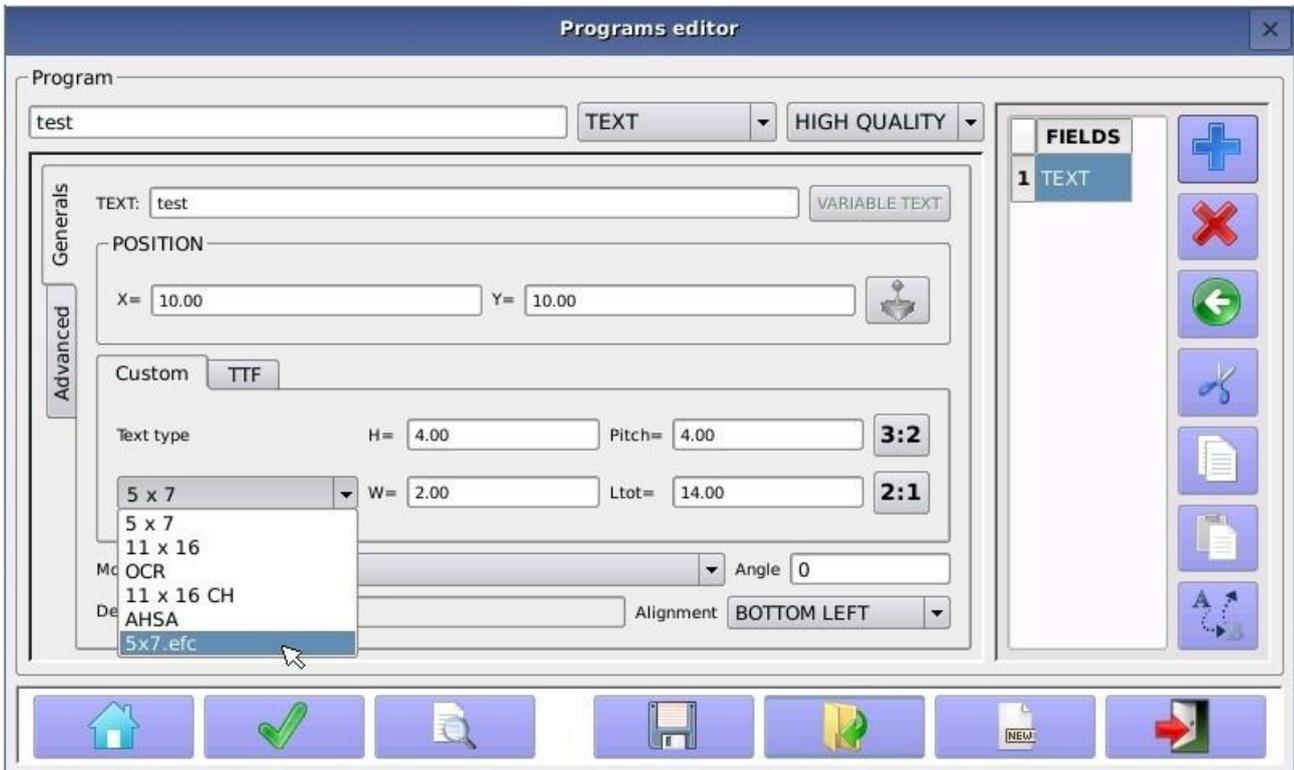


previous settings, press the button (following this operation the modified data will be lost).

3.9 Using Custom Fonts

The software also allows the import of customised font sets that can be used to mark fields with text. Custom fonts are created via the *Font Creator* software and can be inserted from the *File Management* environment. Once uploaded to the controller, they can be selected from the drop-down menu of proprietary fonts. I

Custom fonts will be distinguishable because they will include the extension *efc* in the font name.



Custom Font Selection Screen



Preview screen of a text with custom fonts

NOTE

Custom fonts may not contain all characters that can be entered. If the characters inserted in the text to be marked contain characters that are not present in the custom font, they will be filled in as spacing characters (*blank*).

NOTE

Custom fonts may not be executed in all working modes (dot, continuous and vibrating or smearing). This depends on how the font was created.

3.10 Set the programme serial number

The serial number of a programme is stored within the programme and is only specific to the programme in which it resides. Therefore, the serial number of a programme is independent of other serial numbers of other programmes.

The serial number of the programme is marked whenever a field containing the %S flag is marked.

The serial number can be set to increment:

- whenever the programme is marked;

- each time every field in the programme containing that command is marked.

According to the set increment value:

- Positive (increment > 0)

When the serial number reaches the upper limit, the number can:

- reset to the lower limit set in a numbering table;
- stop the marking operation until the numbering setting is changed.

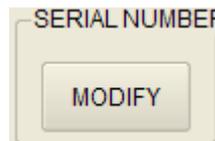
- Negative (increment < 0)

When the serial number reaches the lower limit, the number can:

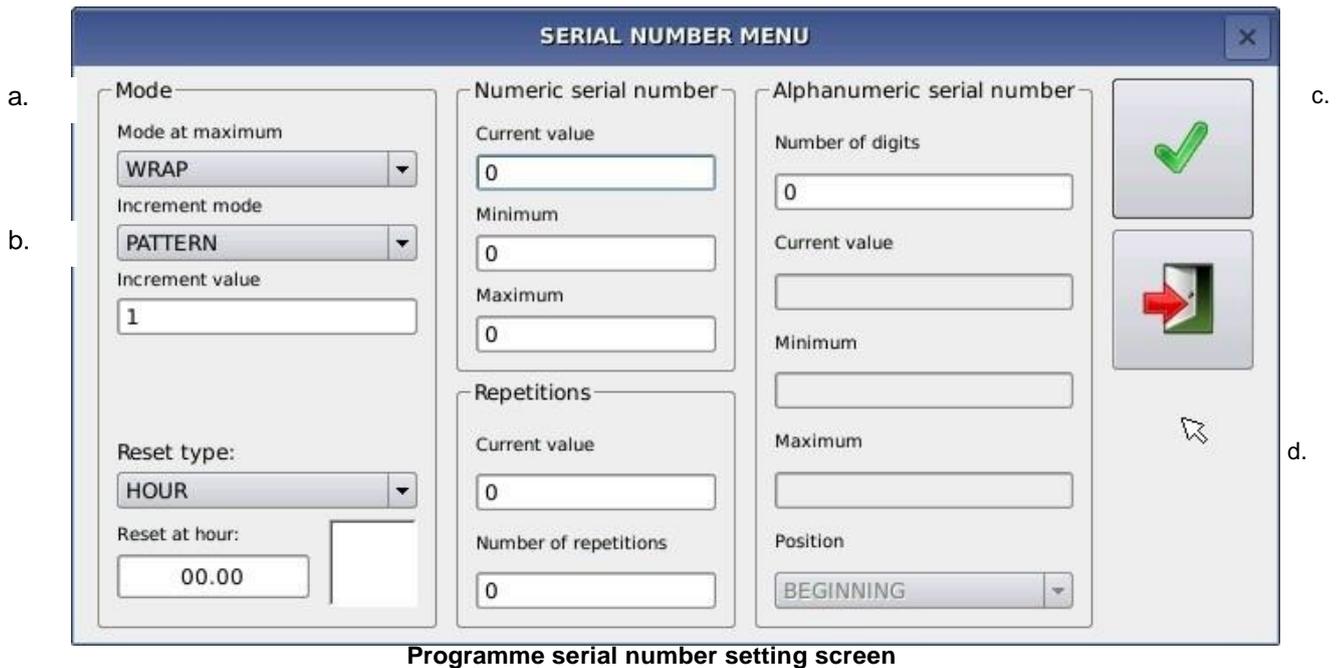
- reset at limit upper set at a appropriate table of numbering;
- stop the marking operation until the numbering setting is changed.

To set the serial number data for the programme:

1. From the programme editor screen, press on the serial number parameter configuration button



2. The software will display a window on the screen allowing you to set the configuration data for serial number management



Programme serial number setting screen

a. In this screen you can set how the serial number is to be used:

- **Limit modes**

Parameter indicating how the serial number will be handled if the value reaches one of two limits (based on the increment set).

• **WRAP**

Serial number management mode that allows the serial number to return to the minimum value (or maximum, depending on the increment set) if the maximum value (or minimum, depending on the increment set) is reached. This effectively resets the serial number to an initial value and allows marking to continue without interruption by the user.

• **STOP**

Serial number management mode that allows execution to be stopped if the maximum value of the serial number is reached (or minimum, depending on the increment set).

- **Increasing modes**

Parameter indicating how the serial number is incremented.

• **PATTERN**

Increment mode according to programme. The serial number will only be incremented, by the value set as increment, at the end of the programme execution.

• **FIELD**

Increment mode based on the programme field. The serial number will be incremented, by the value set as increment, at the end of the execution of a field using the serial number.

- **Increment value**

Parameter indicating by which number the serial number is to be incremented, depending on the increment mode set. The increment value can assume positive or negative values.

- **Serial number reset**

Parameters indicating the reset mode to be used and its enabling. The software also allows the type of reset to be used to be configured (by time, by day, etc.) and according to the different mode selected, the relevant values can be entered.

- **NOW**

Allows you to set the reset time for the current programme serial number. For example, if 02:00 is set as the reset time, the software will reset the serial number of the programme as soon as it recognises that this time has passed.

- **DAY**

Allows you to set the day of the week to reset the serial number of the current programme. For example, if you set Monday as the reset day, the software will reset the serial number of the programme as soon as it recognises that that day has passed.

- **DATA**

Allows you to set the day of a month to reset the serial number of the current programme. For example, if you set the 15th of the month as the reset date, the software will reset the serial number of the programme as soon as it recognises that this date has passed. If you set the 31st of the month as the date, the last valid day of the month will be taken in months with fewer days.

- **TURNO**

Allows you to set the reset shift of the current programme serial number. For example, if you set the second shift as the reset shift, the software will reset the serial number of the programme as soon as it recognises that this shift has passed.

If the serial number increment is positive (>0), the serial number will be reset to the set minimum limit. If the serial number increment is negative, the serial number will be reset to the maximum limit set.

- b. In this window, it will be possible to set the current value of the serial number and the minimum and maximum value within which the serial value will be set. The value

current must be between the set minimum and maximum value. In addition, the minimum set value must be less than the maximum value.

- c. In this window, it will be possible to set the parameters required to manage the alphanumeric values within the serial number.

- **Number of digits**

Parameter indicating how many alphanumerical digits are to be managed within the serial number. If the number of digits is 0 then the alphanumeric value parameters will not be enabled.

- **Positioning in the serial number**

Parameter indicating where to place the alphanumeric digits within the serial number.

• **START**

Placement of alphanumeric digits at the top of the serial number field.

In this case, the serial value increment management, and the checks on the maximum values, will also be modified and referred to the values of the alphanumerical digits. In this case:

if we have the serial value "A101", where "A" is the alphanumeric value, at the time of the increment, if the numeric value reaches the value

maximum set in the serial value parameter, the numeric value will be set equal to the minimum value and the alphanumeric value will be incremented (or decremented) by one digit. For example:

Minimum value	=	10
Maximum numerical value	=	101
Numerical current value	=	101
Increase	=	+1
Alphanumeric value	=	"A"

The result of the serial value 'A101' incremented will be: 'B10'.

In this case, checks on the mode of the serial value limit will refer to the maximum values of the alphanumeric value.

• **END**

Placement of alphanumeric digits at the end of the serial number field.

In this case, the serial value increment management, and the checks on the maximum values, will also be modified and referred to the values of the alphanumerical digits. In this case:

if we have the serial value '101A', where 'A' is the alphanumeric value, when incrementing, the alphanumeric value will be incremented (or decremented) by one digit. For example:

Minimum value	=	10
Maximum numerical value	=	101
Numerical current value	=	101
Increase	=	+1
Alphanumeric value	=	"A"

The result of the serial value '101A' incremented will be: '101B'.

If the alphanumeric value reaches the set maximum alphanumeric value, the numeric value will be incremented by the set increment value and the alphanumeric value will be increased to its minimum value

- **General alphanumeric digit parameters**

General parameters for setting the current value, minimum value and maximum value of alphanumerical digits.

d. In this window, it will be possible to set the parameters required to manage the number of repetitions of the serial number.

- **Current value**

Parameter indicating the current number of repetitions performed

- **Number of repetitions**

Parameter indicating how many repetitions are to be performed of the current serial number.

These two values are used to determine the number of times the system should mark the current serial number. The current value of the number of repetitions indicates how many times it has already been executed. When this value reaches the set number of repetitions then the serial number will be incremented (according to the set setting).

If the number of repetitions is left at 0 (zero), no such management will be used.

If the number of repetitions of the serial number is used, the handling specified in the general parameter '*Handling*' will be used in the event of a cycle abort.

abort for serial number repetitions' (see *General parameter setting* section in the software).

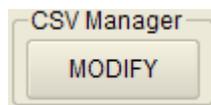
3.11 Set configuration for data to be marked from CSV file

The software also allows you to mark up data from a worksheet in CSV format. It will then be possible to take data from the table of the CSV file and mark it in a programme. This management is very flexible on the part of the operator, as the way in which the data is taken from the file is fully settable. These parameters will be explained in detail in this section. The settings of the CSV management are program-related, so it is possible to set up the same file in several programmes and mark it with totally different modes between the two.

As with serial numbers, it is possible to use the data in the CSV file via the *%t* flag command. See the flag command section for further details.

To set up CSV management data:

1. From the programme editor screen, press on the CSV management parameters configuration button



2. The software will display a window on the screen allowing you to set the configuration data for CSV management

CSV MANAGER

Filename: 

	1	2	3	4
1	City	Nation	Population	Language
2	Rome	Italy	2000000	Italian
3	Paris	France	5000000	French
4	London	United Kingd...	8000000	English

Current cell:  Empty cell mode:

Scan type: Custom mode cell: 

Increment: Separator: File deletion after execution




Data configuration screen for CSV management

From this screen you can set all possible settings for customising the use of CSV management. There is also a preview table of the selected CSV file from which it will be possible to select the various reference cells. All the parameters on this screen are described below:

- **File name**

Specifies the file to be used for current management. The file can be loaded from the controller memory or directly from a USB stick. If the set file will be taken from the USB stick, "[USB]" will be displayed following the file name. In order to select the file to be used, you must press the appropriate button on the right of the text box.

**CSV file upload button**

- **Current cell**

Defines the cell from which the first data item to be marked will be taken. The data displayed refer to the rows and columns of the table. The values refer to the position in rows and columns of the table. For example: if the field is set to 1;2 the current cell will be in the first row/second column of the table.

It is possible to select the current cell directly from the CSV preview using the button to the right of the text box.

**Cell selection button**

Pressing this button will display the currently selected cell as the current cell. Selecting any cell on the table will change the value of the current cell according to the selected cell.

To remove the cell selection mode, simply press the cell selection button once more. Finally, the current cell can be set manually by pressing on the reference text box.

- **Scan type**

Defines the type of table scanning based on the selected *increment* parameter. The selectable types of table scanning are: scan by *rows* or scan by *columns*.

Scan by rows

The programme will increment rows by rows starting from the current cell. When the last row is reached, the programme will automatically move on to the next column.

Scanning by columns

The programme will increment the columns step by step starting from the current cell. When the last column is reached, the programme will automatically move on to the next row.

When the programme finishes the last executable cell, the CSV is considered finished.

- **Increase**

Defines the number of cell increments each time a CSV command is used. The increment value works in conjunction with the scan type to define how to extract information from the CSV table. The increment will be done whenever a %t flag field is used.

- **Empty cell mode**

Defines the mode to be followed if an empty cell is found when scanning the table. The values that can be set are: *ignore*, *next row*, *next column*, *stop* and *custom*.

Ignore

If an empty cell is detected during scanning, it will be placed normally within the field. This will allow an empty cell to be treated just like a normal table cell.

Next line

If an empty cell is detected during scanning, it will be skipped by taking the cell in the next row. If several consecutive empty cells are present, they will be skipped until a non-empty cell is reached.

Next column

If an empty cell is detected during scanning, it will be skipped by taking the cell in the next column. If several consecutive empty cells are present, they will be skipped until a non-empty cell is reached.

Stop

If an empty cell is detected during scanning, marking will be prevented. This case will be handled as an end of the csv.

Custom

If an empty cell is detected during the scan, the cell set in the custom cell field is taken as the current cell.

- Custom mode cell

Defines the table cell to which the programme should position itself in the case of an empty cell and empty cell mode set to custom cell. The displayed data refer to the rows and columns of the table. The values refer to the position in rows and columns of the table. For example: if the field is set to 1;2 the custom cell will be in the first row/second column of the table.

It is possible to select the custom cell directly from the CSV preview using the button to the right of the text box.



Cell selection button

Pressing this button will display the currently selected cell as a custom cell. Selecting any cell on the table will change the value of the custom cell according to the selected cell.

To remove the cell selection mode, simply press the cell selection button once more. Finally, the custom cell can be set manually by pressing on the reference text box.

- Deleting files after execution

This parameter can be used to enable deletion of the file once it has finished execution. Both the execution of the last table cell and the positioning on empty cell with empty cell mode set to stop are considered as end-of-execution conditions.

3.12 Defining the background image of the programme preview

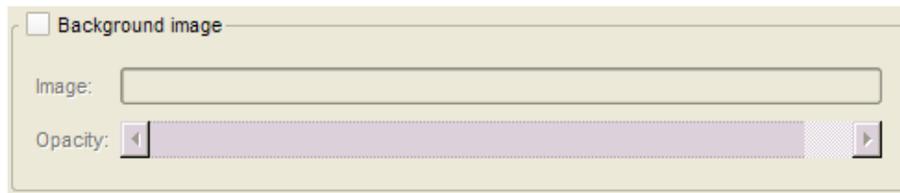
The software allows you to associate, for each programme in memory, a PNG image to be displayed in the programme preview. This image can be useful for

represent certain encumbrances in the part to be marked (e.g. nameplates) and can therefore help the positioning of the fields to be previewed.

The image to be set in the programme preview must be present in the controller's memory. In addition, there is a further parameter that allows you to set the degree of opacity (and thus relative transparency) of the image to be displayed.

To set programme preview data:

1. From the programme editor screen, enable background management by enabling the appropriate check;



Window for enabling background management for preview

2. Select the image to be associated with the programme;
3. Set the degree of opacity of the image;



Example of preview with background image

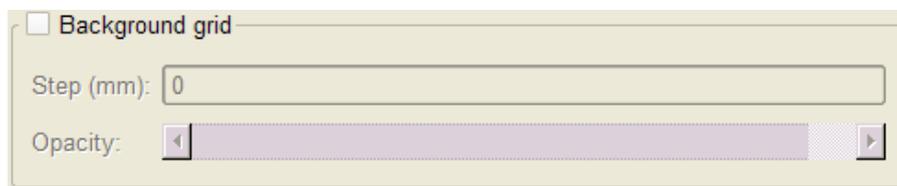
3.13 Defining the background grid of the programme preview

The software allows you to associate, for each programme in memory, a grid to be displayed in the programme preview. This grid can be useful for displaying more precisely the position of fields within the work area and can therefore help the positioning of fields to be previewed. The software also allows you to set the pitch to be used for the grid representation in the programme preview.

In addition, there is a further parameter that allows you to set the degree of opacity (and thus relative transparency) of the image to be displayed.

To set the data of the programme preview:

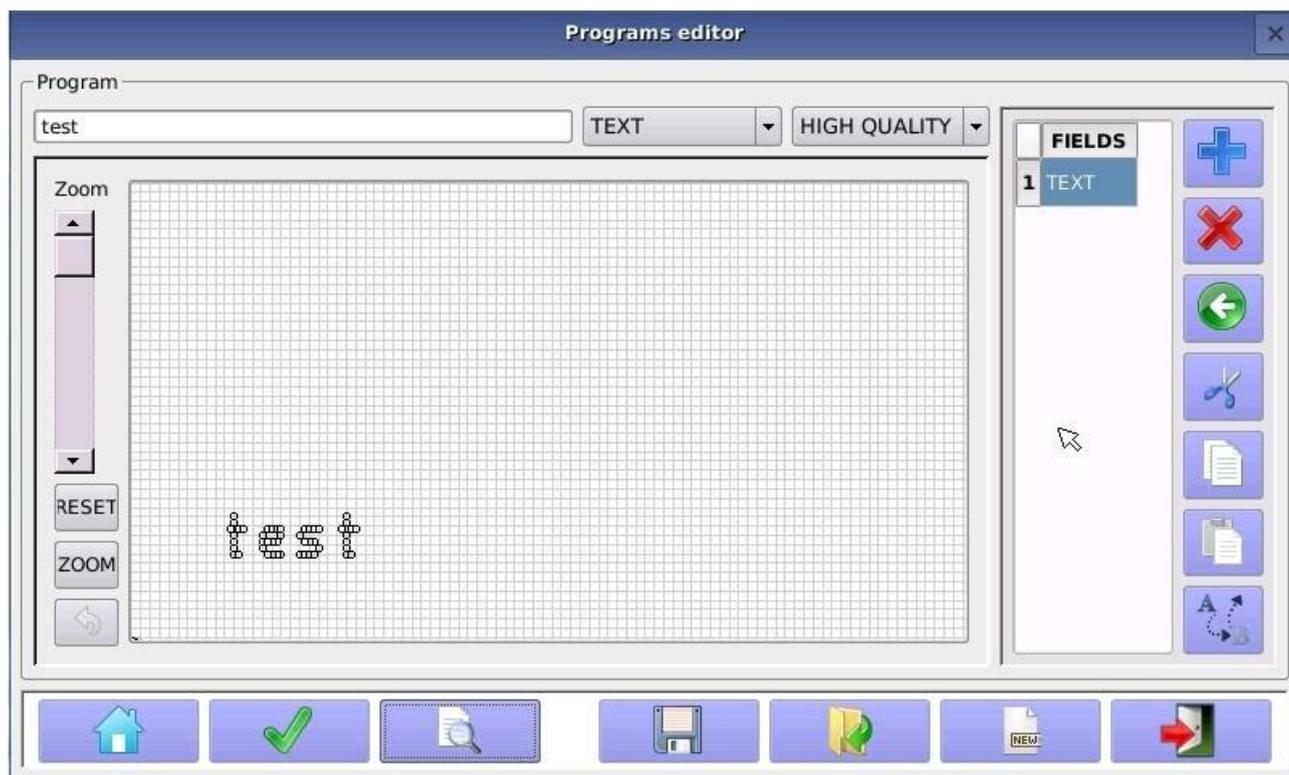
1. From the programme editor screen, enable background management by enabling the appropriate check;



Window for enabling background management for preview

2. Set the pitch to be used in the grid representation;
3. Set the degree of opacity of the grid;





Example of preview with background grid

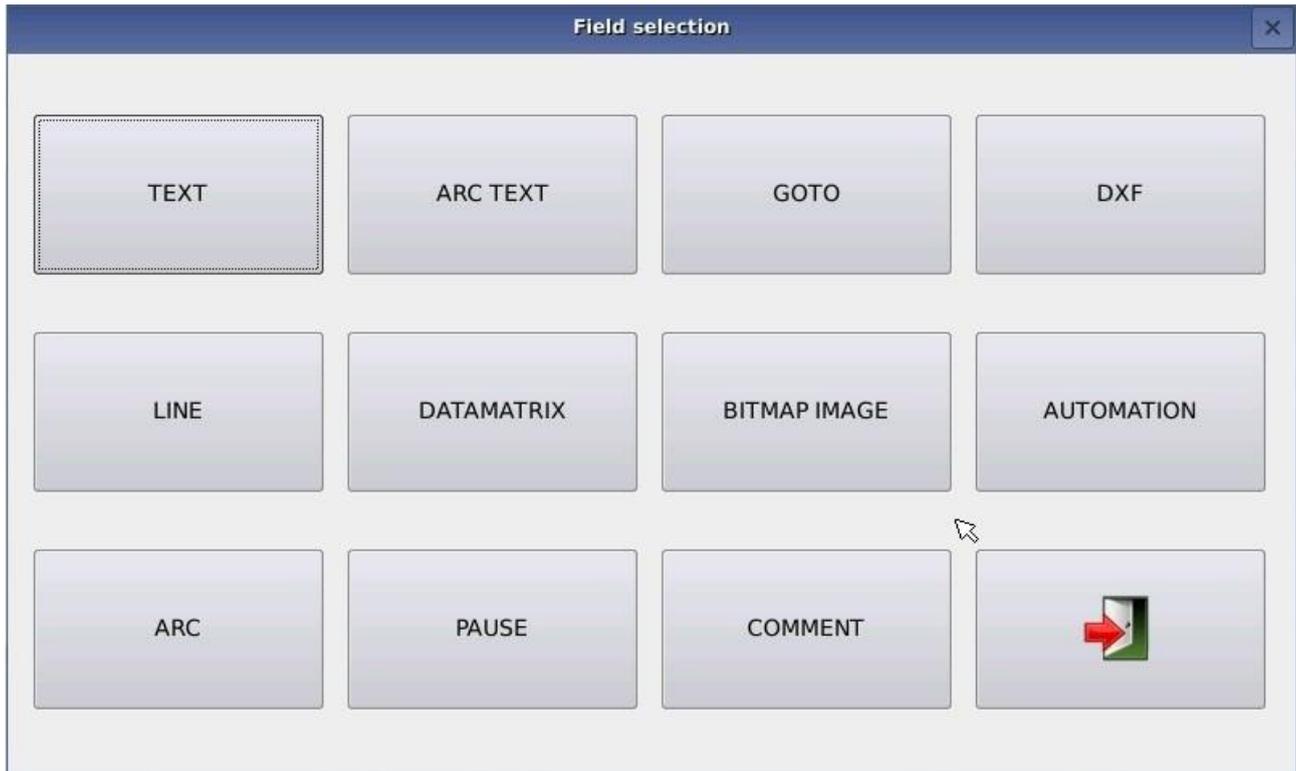
3.14 Adding fields to the programme

After the creation of a programme, fields must be added to define the contents, format and position of the data to be marked. There are different types of fields. Each message can contain as many fields as desired, which are marked in the order in which they were entered. However, it is possible to change the sequence of fields to optimise the marking process by means of cut/copy/paste operations.

To set one or more fields in the programme:

1. *[optional]* Place your head online. (this can help to define the position of the field later);
2. Load the desired programme or create a new one;
3. Enter the programme editor environment and press the button to add a new field to the programme queue . 

The software will display an on-screen screen for selecting the type of the new field.



Field type selection screen

4. Select the new field to be added based on the fields proposed in the selection screen. The selectable fields are:

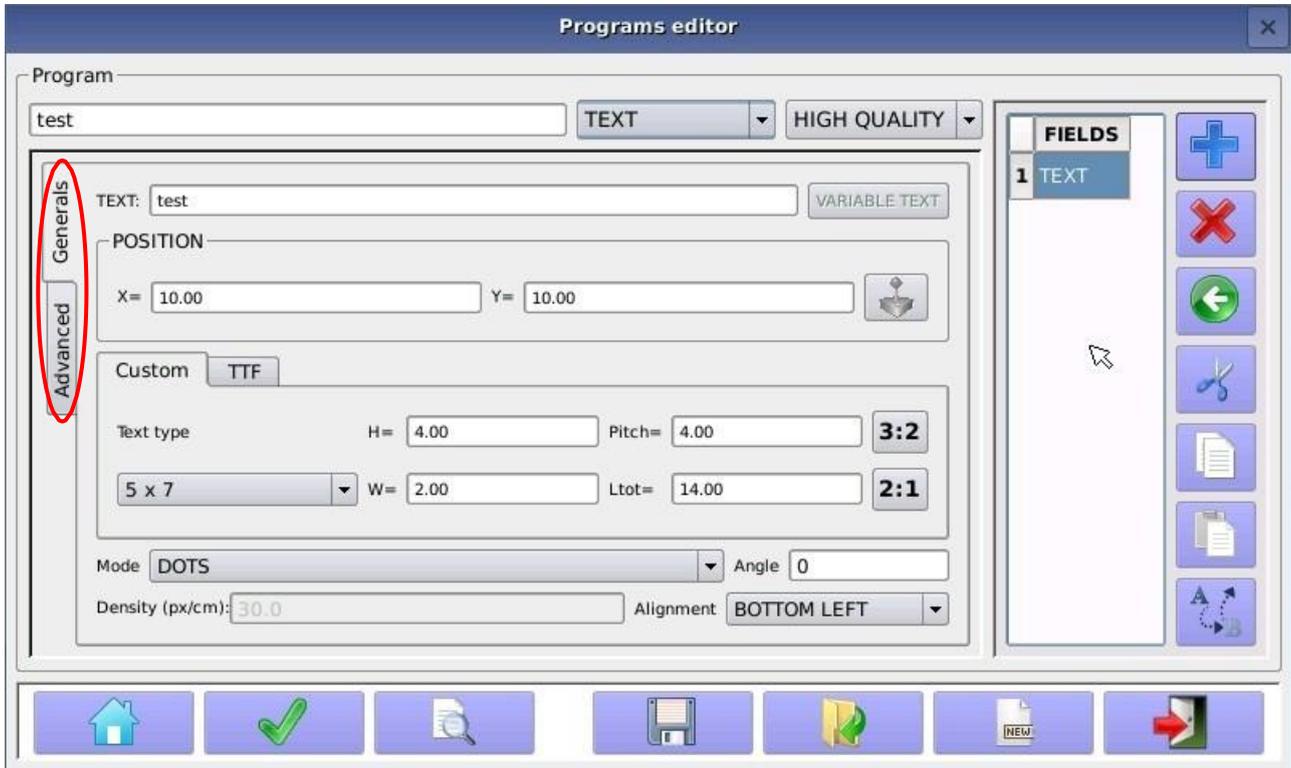
- *Text*
- *Line*
- *Arch*
- *Text on arch*
- *Datamatrix*
- *Break*
- *Goto*
- *Bitmap image*
- *Comment*
- *DXF*

The system will open a page for setting the parameters of the selected field.

5. Set the parameters of the selected field;

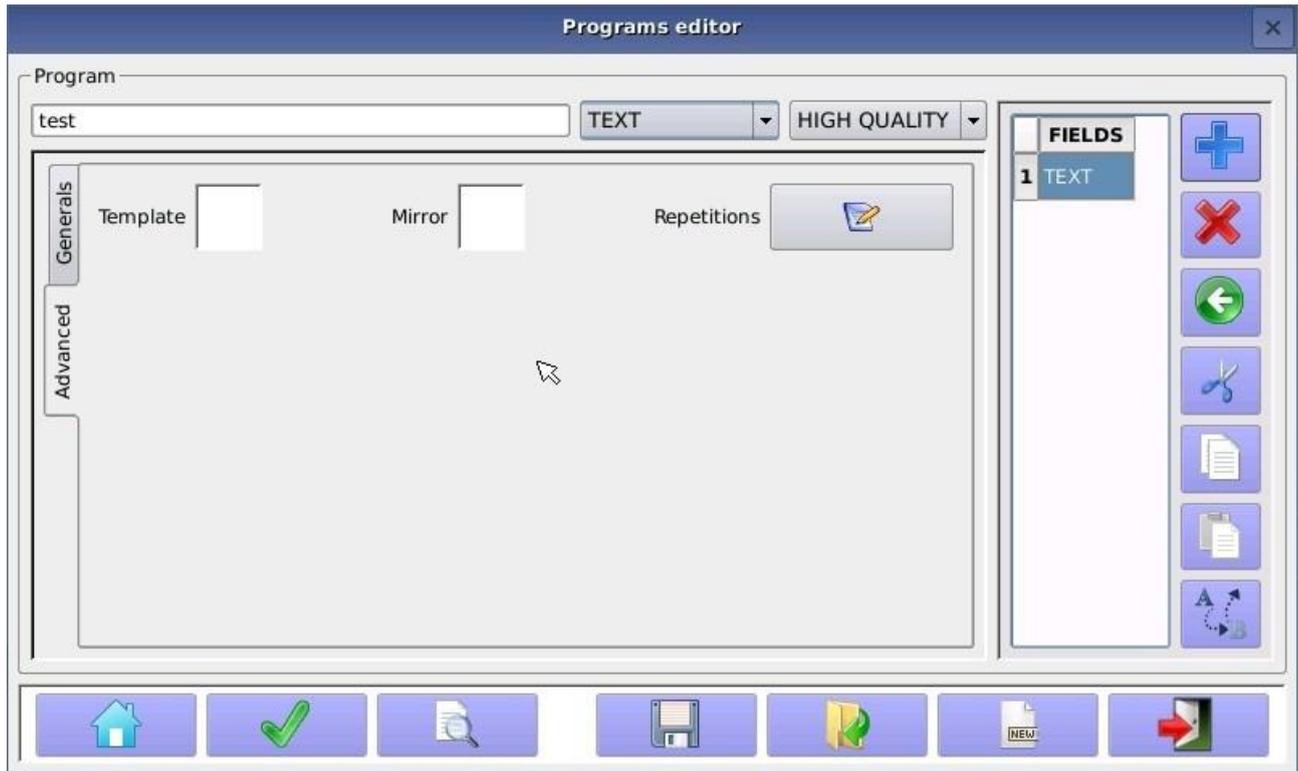
6. When you have finished entering the parameters, save the programme using the button provided.

The field parameters are divided into *general* and *advanced* ones. In the display, these parameters are clearly distinguishable because they are divided into two different tabs on the left-hand side of the window dedicated to setting field parameters.



Editor screen with text field (highlighted data tabs)

While the general field parameters depend on the type of field, the advanced data are linked to further functionality that can be used when editing or running the programme.

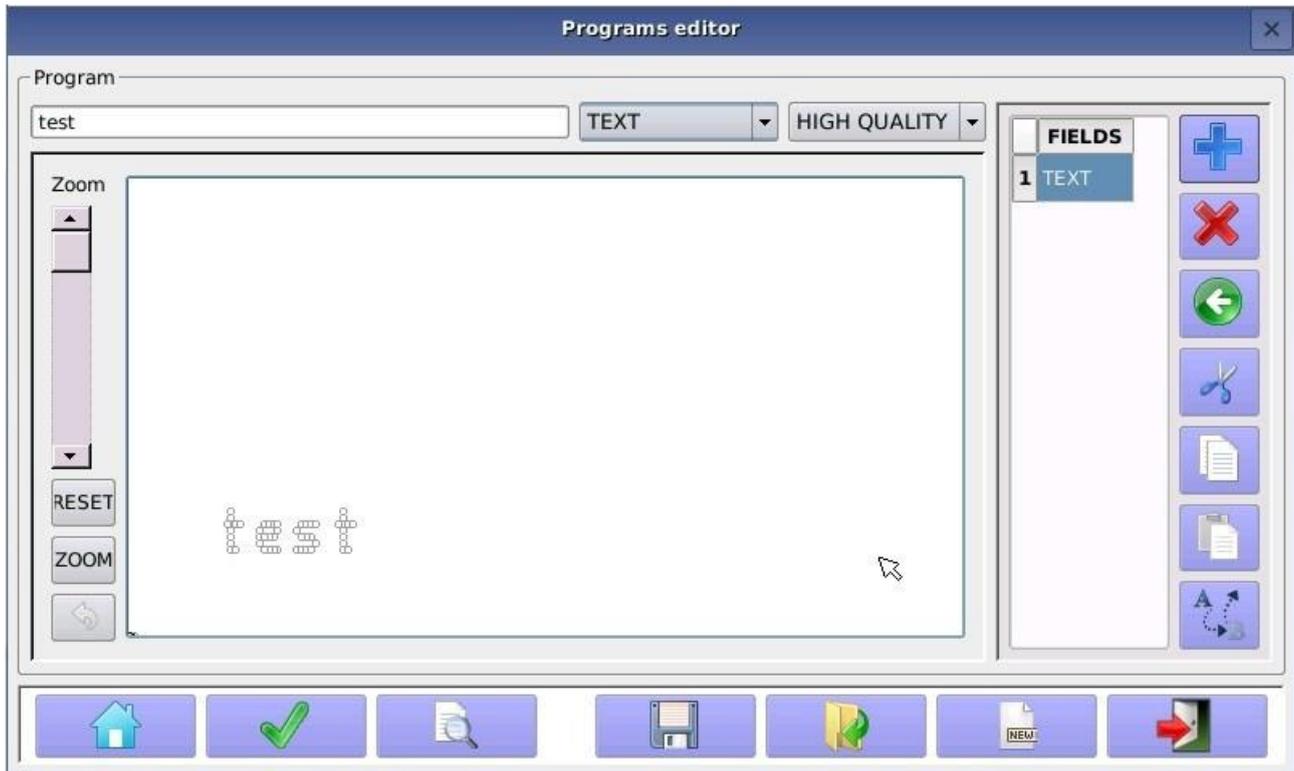


Screen displaying advanced field data

Advanced data can be:

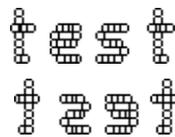
- Enabling '*template*' mode;
- *Mirror* mode enabled;
- Setting the number of field repetitions.

Template mode allows the current field to be displayed but not marked. This property is used to simplify an alignment between fields, one of which you do not wish to mark. Of course, template mode will only be usable on executable fields and therefore not on fields of type GoTo, Pause, etc.



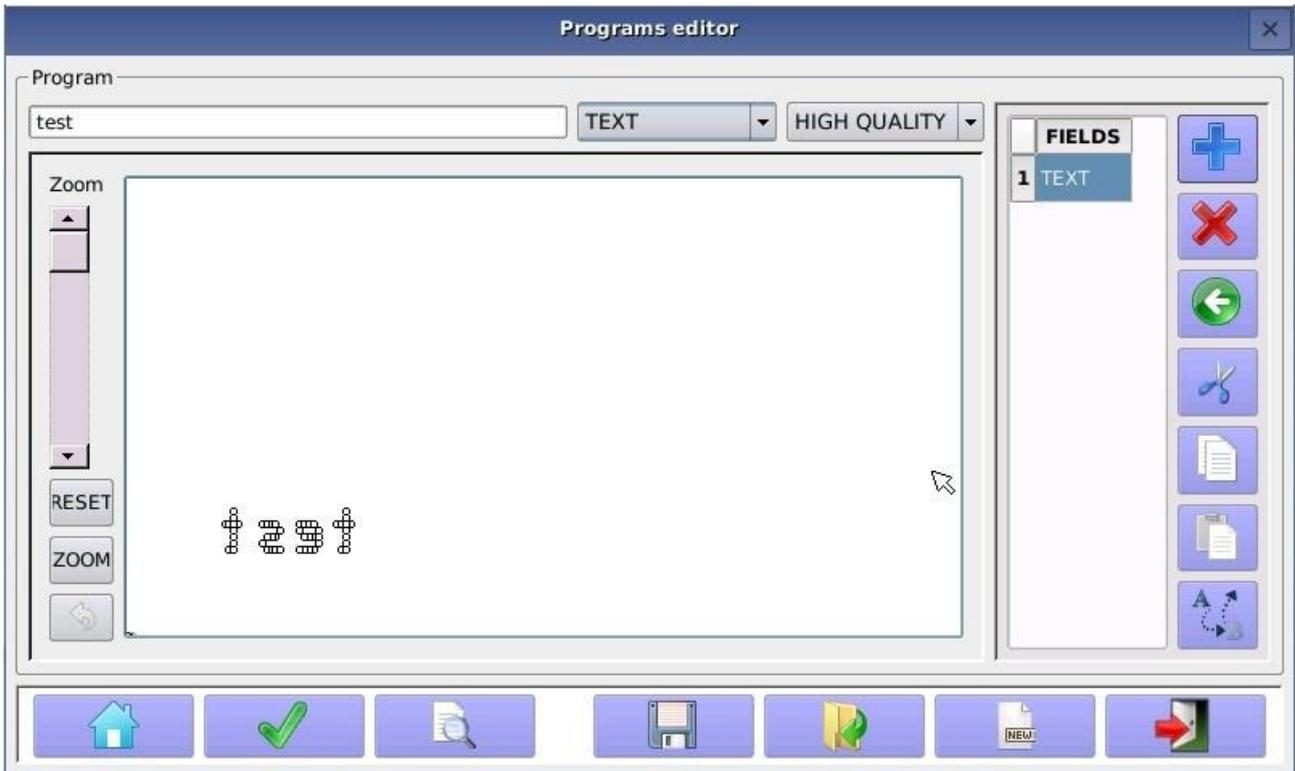
Example of a template text field preview

Mirror mode allows the current field to be mirrored in both display and execution. Mirror mode allows the field to be run in mirrored mode:



Example of mirrored field

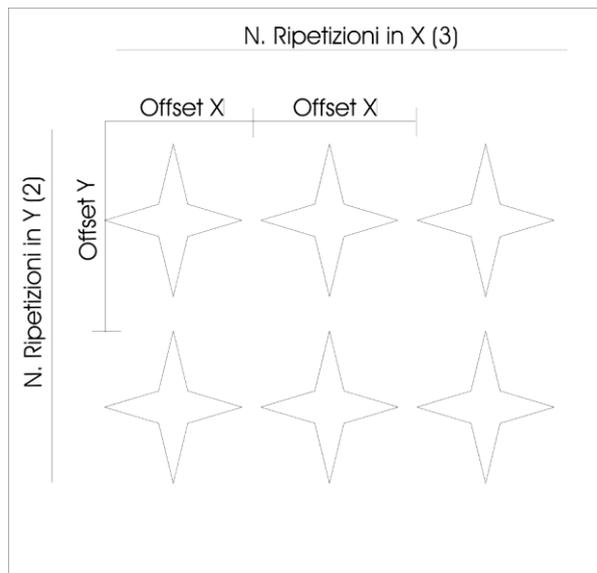
Of course, mirror mode will only be usable on executable fields and therefore not on GoTo, Pause, etc. fields.



Example of a preview of a mirrored text field

The number of repetitions allows you to set a number of repetitions, either by rows or columns, of the current field. For both repeat modes, an offset can be set between one field and the next.

Of course, the template mode will only be usable on the executable fields and therefore not on the GoTo, Pause, etc. type fields.



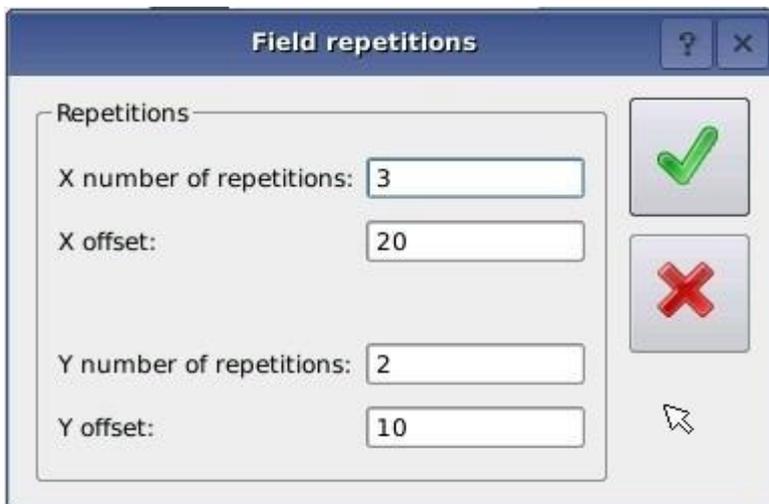
Example of repetitions

In order to set the number of repetitions, it is necessary to press the repeat configuration button:



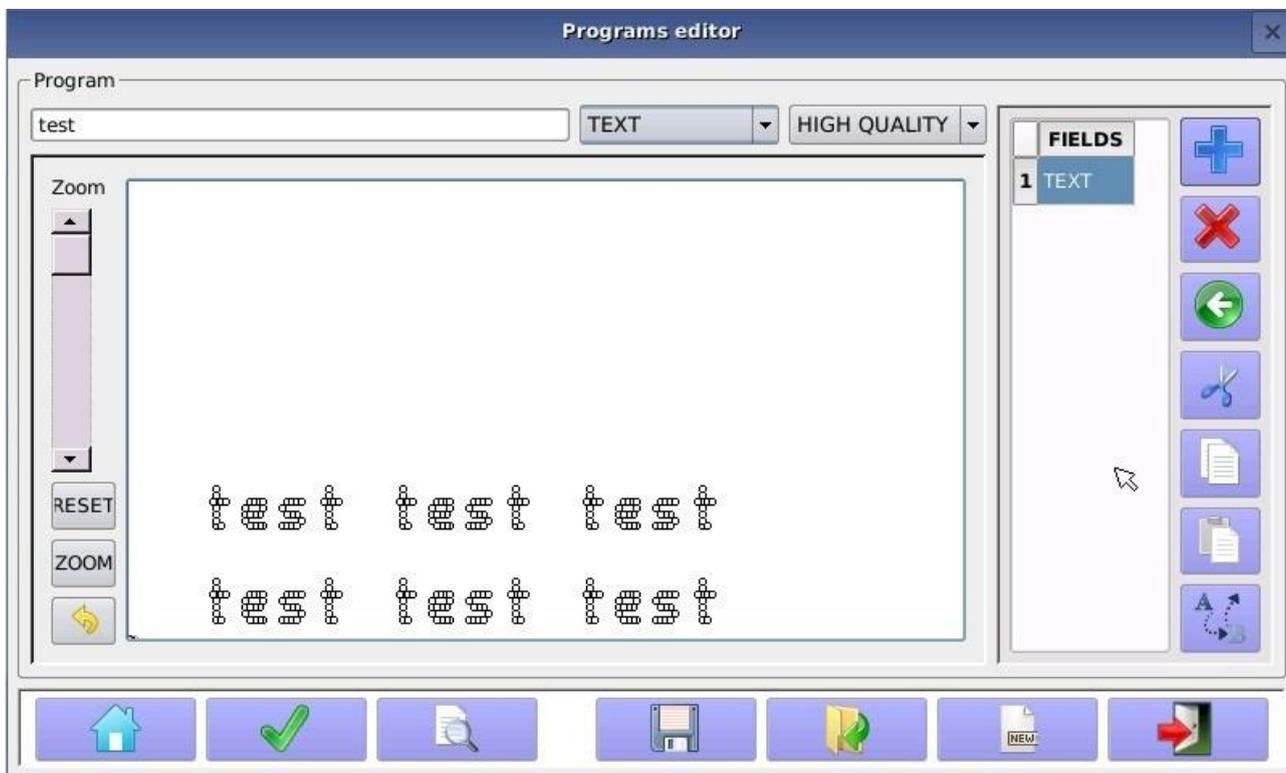
Repeat configuration button

And set the appropriate values in the repeat configuration screen:



Repeat configuration screen

The number of repetitions of the field will be calculated and it will be possible to execute the field in this way:



Programme preview screen with repeating field

3.14.1 Text field parameters

The text field is defined by the size, shape, content and position of the character string. Below are the parameters to be defined in the text field:

Text field parameter setting page (Default font)

Text field parameter setting page (TTF font)

Text

Defines the content of the string, which may contain upper or lower case characters, keyboard symbols, spacing and special commands called FLAG. These commands are introduced by the percentage sign % and followed by letters each corresponding to a command. They

are used to insert in the text content values to be updated such as date, time, shift, serial number.

Font

Defines the font to be used when displaying the string. The fonts vary from the control's default font to custom fonts (TTF fonts).

Default font

Type defines the type of proprietary character to be used in the field. The proprietary fonts are *5x7*, *11x16*, *OCR*, *11x16 CH* and *AHSA*. Furthermore, if custom fonts have been entered in the controller, they will be displayed following these default fonts.

NOTE

Some proprietary fonts may not contain all the characters of the ASCII code.

Character Height the height of the character is measured from the centre of the top point to the centre of the bottom point of the character to be marked.

Character Width The character width is measured from the centre of the leftmost point to the centre of the rightmost point of the character to be marked.

Pitch is measured from the centre of the leftmost point of one character to the centre of the starting point of the next character, including the width of the character and the spacing between two consecutive characters. Unless there is a special requirement, the pitch value is generally identical to the character height value.

Total Text Length the total text length indicates how long the text is based on the number of characters present and the character width and pitch parameters.

Ratio indicates the ratio to be used between the height and width of the characters. If selected, the two values will be 'locked' so as to maintain the same ratio and when one is changed, the other will be recalculated.

NOTE

automatic calculation of character width, pitch and total length parameters if you change one of these parameters, the software will automatically recalculate the updated values of the other fields.

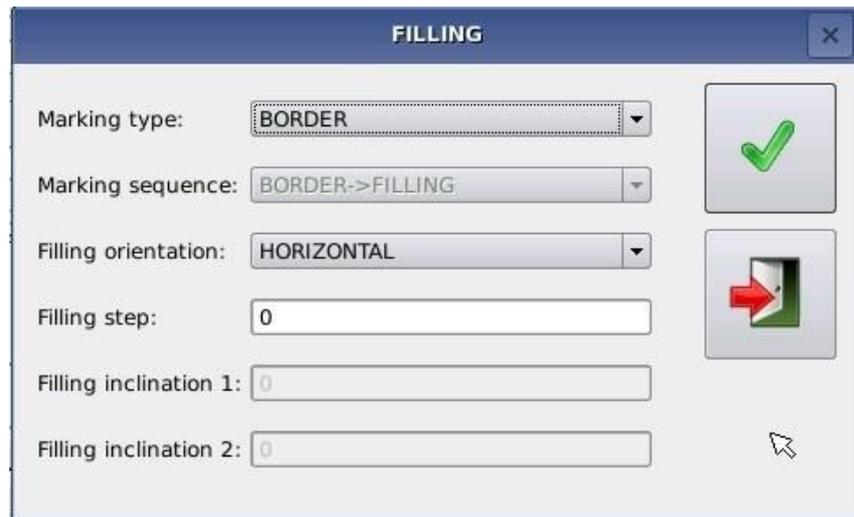
TTF

TTF font type selectable

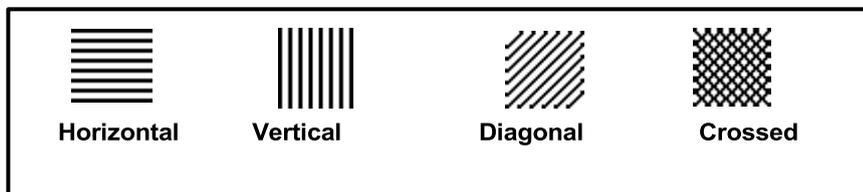
Character Height the height of the character is measured from the centre of the top point to the centre of the bottom point of the character to be marked.



Filling parameter configuration button



Filling parameter configuration screen



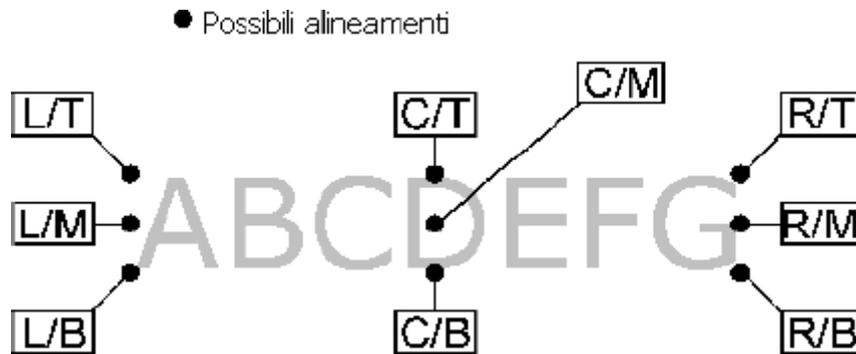
Vertical

Example of field with high fill pitch

Filling pitch indicates the distance between one filling line and the next. A low pitch value indicates a higher concentration of fill lines in the field, conversely, a high pitch value indicates a lower concentration of fill lines in the field.

X, Y Defines the coordinates of the field within the marking area. The coordinates refer to the justification of the field.

Justification Defines the point to which the co-ordinates refer in relation to the field.



- L/T (Left/Top) - Justification top left**
- L/M (Left/Middle) - Left centre justification**
- L/B (Left/Bottom) - Left bottom justification**
- C/T (Centre/Top) - Top centre justification**
- C/M (Center/Middle) - Justification in centre**
- C/B (Center/Bottom) - Justification at bottom**
- R/T (Right/Top) - Justification at top right**
- R/M (Right/Middle) - Justification in centre right**
- R/B (Right/Bottom) - Justification in bottom right**

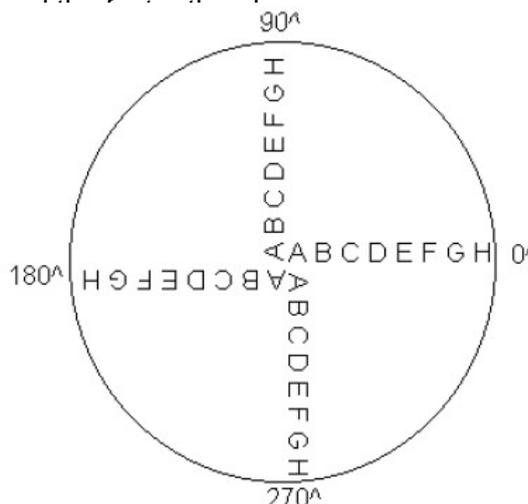
If uppercase or lowercase characters are marked with the justification BELOW, the marking will look like the following:



Mode Defines the marking mode with which the current field will be executed.

Density If the continuous mode is set, the dot density can be defined (density of dots per inch or centimetre). The higher the dot density, the longer

Corner



Document:

INSTRUCTIONS FOR USE OF THE SOFTWARE

AUTOMATOR MARKING SYSTEMS SRL - Via Meucci n.8-20094 Corsico (Mi) Italy
Tel 02 48601445 Fax 02 48601503 e-mail info@automator.com www.automator.com

Quality Defines the marking quality according to the set marking head. Depending on the selected quality, a more or less defined marking can be achieved. This will affect the performance of the machine.

3.14.2 Line field parameters

The line field is defined by the co-ordinates of the two points between which the line lies.

The screenshot shows a software interface for setting line field parameters. It is divided into two tabs: 'Generals' and 'Advanced'. The 'Advanced' tab is active. Under 'P1 point', there are input fields for X= (0.00) and Y= (0.00) with a small icon to the right. Below that, 'P2 point' has similar X= (0.00) and Y= (0.00) fields with an icon. The 'P2 Reference' is set to 'ABSOLUTE POSITION' in a dropdown menu. The 'Mode' is set to 'CONTINUOUS' in another dropdown menu. At the bottom, the 'Density (px/cm):' is set to '30.0'.

Text field parameter setting page

X, Y (P1) They define the co-ordinates of the first point. This is taken as the starting point of the line.

X, Y (P2) They define the co-ordinates of the second point. This is taken as the end point of the line.

Reference P2 Defines what point P2 is referred to. If it is set to absolute position, it means that it will use the origin point of the marking area as a reference. Otherwise, if it is set to relative position, the co-ordinates will refer to point P1 on the line.

Defines the marking mode with which the current field will be executed.

Mode

If the continuous mode is set, the dot density (dot density per inch or per centimetre) can be defined. The higher the dot density, the longer the marking time and faster the plunger wear.

Density

Defines the marking quality according to the set marking head. Depending on the selected quality, a more or less defined marking can be achieved. This will affect the performance of the machine.

Quality

3.14.3 Arc field parameters

The system software provides two ways of creating arc fields: the *degree method* and the *point method*. To implement an arc field, a set of parameters must be set. The categories of parameters to be set are: *common arc parameters*, *degree method parameters* and *point method parameters*.

Text field parameter setting page (degree method)

COMMON PARAMETERS

Mode Defines the marking mode with which the current field will be executed.

Density If the continuous mode is set, the dot density (dot density per inch or per centimetre) can be defined. The higher the dot density, the longer the marking time and faster the plunger wear.

Quality Defines the marking quality according to the set marking head. Depending on the selected quality, a more or less defined marking can be achieved. This will affect the performance of the machine.

GRADE METHOD

Degrees		DOTS	
Center			
X=	<input type="text" value="0.00"/>	Y=	<input type="text" value="0.00"/>
Radius	<input type="text" value="0.00"/>		
Start Angle	<input type="text" value="0"/>		
End Angle	<input type="text" value="0"/>		
Prepositioning	START POINT		▼

Settable parameters for the degree method**Centre**

The arc centre parameters in X and Y identify the position of the arc centre within the marking area. The coordinates are calculated with reference to the origin point of the marking area (0,0). Set the desired coordinates or use the Jog key.

Radius

Parameter indicating the length of the circle radius (in inches or millimetres, depending on the unit of measurement chosen). The radius value mathematically defines the measurement of the arc. Check that the arc does not protrude from the marking area.

Corners

The values of the initial angle and final angle define where the arc begins and ends. The starting point of the arc is the intersection of the initial angle and the circumference described by the set radius measure and with centre the centre of the set angle. The end point of the arc is the intersection of the circumference itself with the final angle set. The reference point 0° corresponds to the 3 o'clock position on the dial and is incremented anti-clockwise, up to the maximum value of 360°. Set start angle and end angle values in degrees.

Prepositioning

The pre-positioning value is used to indicate how to perform the pre-positioning movement on this field. If set to initial point, pre-positioning will be done at the justification position of the field otherwise, if set to centre, pre-positioning will be done at the centre of the arc.

**NOTE**

The initial angle and the final angle determine whether the angle is concave or convex, since the software ALWAYS handles arcs with anticlockwise direction from the initial point.

POINT METHOD

Parameters that can be set for the point method

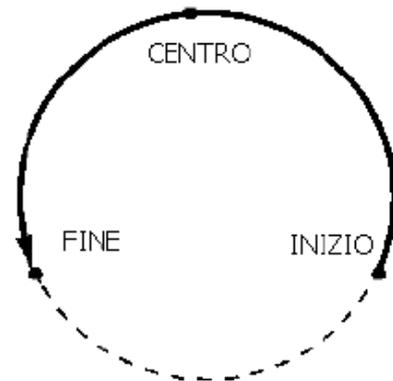
It is advisable to first define the start and end points of the arc, before defining the opening of the arc. The system will also create a more accurate arc if the operator sets the centre point as close to reality as possible.

NOTE

The start, middle and end points must NOT fall on the same horizontal or vertical line

Initial

The initial point parameters in X and Y identify the co-ordinates of the initial point of the arc. The co-ordinates refer to the origin point of the marking area (0,0). Set the desired coordinates or use the Jog key.

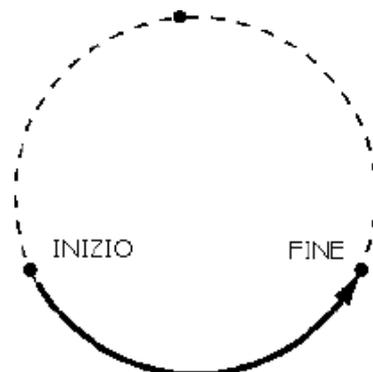


NOTE

The system constructs the arcs anticlockwise from the starting point of the arc.

Centre

The central point parameters in X and Y identify the path definition co-ordinates at arc within the area marking area. The coordinates are referenced to the origin point of the marking area (0,0). Set the coordinates directly or use the Jog key.



Final The end point parameters in X and Y identify the end point of the arc. The co-ordinates refer to the origin point of the marking area (0,0).
Enter coordinates directly or use the Jog key

NOTE

The start angle (start angle) and the end angle (end angle) determine whether the angle is concave or convex, as the software ALWAYS handles arcs with anticlockwise direction from the start point.

3.14.4 Text field parameters on arc

This type of field allows a text string to be constructed along an arc of a circumference by combining the characteristics of the text and the arc. As with the definition of the arc, there are two reference methods for constructing arc text.

Arc text parameter setting page

COMMON PARAMETERS

Mode Defines the marking mode with which the current field will be executed.

Density	If the continuous mode is set, the dot density (dot density per inch or per centimetre) can be defined. The higher the dot density, the longer the marking time and faster the plunger wear.
Quality	Defines the marking quality according to the set marking head. Depending on the selected quality, a more or less defined marking can be achieved. This will affect the performance of the machine.
Text	Defines the content of the string, which may contain upper or lower case characters, keyboard symbols, spacing and special commands called FLAG. These commands are introduced by the percentage sign % and followed by letters each corresponding to a command. They are used to insert values into the text content to be updated such as date, time, shift, serial number.
Font	Defines the font to be used when displaying the string. The fonts vary from the control's default font to custom fonts (TTF fonts). <i>See settable text field parameters.</i> Defines the point to which the co-ordinates refer in relation to the field.

Justification**R/T (Right/Top) - Justification at top right****R/M (Right/Middle) - Justification centre right R/B****(Right/Bottom) - Justification bottom right C/T****(Centre/Top) - Justification top centre****C/M (Center/Middle) - Justification in the centre C/B****(Center/Bottom) - Justification at the bottom in the centre****L/T (Left/Top) - Justification at the top left****L/M (Left/Middle) - Justification in centre left L/B****(Left/Bottom) - Justification in bottom left**

If T (high) alignment is selected and characters with a special phoneme are marked, (e.g. 'É'), the marking head will use the top of these characters as the upper limit and align the other characters accordingly. That is, it will not use the top of the characters, but the mark above them.

If alignment B (low) is selected and lower case characters are marked, the lower alignment limit will be the lowest point of the characters (e.g. 'g').

Shape

Parameter for selecting the shape of the text on the current arc. Parameter can be set to Concave or Convex. Unlike arcs, which are always drawn counterclockwise, the software handles arc text both counterclockwise and clockwise, depending on the Shape parameter and the text alignment parameters.



ALLINEAMENTO A SINISTRA/IN BASSO

GRADE METHOD

Centre

The arc centre parameters in X and Y identify the position of the arc centre within the marking area. The coordinates are calculated with reference to the origin point of the marking area (0,0).

Corner

The initial angle S defines where the initial point of the arc will begin the circular path. The initial point of the arc will be the point of intersection of S / initial angle and the circumference. on which the arc text resides. The reference point 0° corresponds to 3 o'clock and increases counterclockwise until it reaches a maximum value of 360. enter the value of the starting angle.



Radius

Enter the radius value of the circle in inches or millimetres, depending on the unit of measurement chosen. The radius value defines the size of the arc. Ensure that the radius measurement does not create an arc that protrudes from the marking area.

POINT METHOD

It is advisable to first define the start and end points of the arc, before defining the opening of the arc. The system will also create a more accurate arc if the operator sets the centre point as close to reality as possible.

NOTE

The start, middle and end points must NOT fall on the same horizontal or vertical line

The start point defines the positioning of the text string. The middle and end points help define the circular or arched path, but the text may or may not pass through the three points. In fact, it depends on whether the text is concave or convex. The text may pass the defined end point if the text string is longer than the defined arc.

Initial The initial point parameters in X and Y identify the co-ordinates of the initial point of the arc. The co-ordinates refer to the origin point of the marking area (0,0). The system will align the field to the X/Y co-ordinates, depending on the type of alignment chosen.
Set the desired coordinates or use the Jog key.

Centre The central point parameters in X and Y identify the path definition co-ordinates at arc within the area marking area. The coordinates are referenced to the origin point of the marking area (0,0). Set the coordinates directly or use the Jog key.



Final The end point parameters in X and Y identify the end point of the arc. The co-ordinates refer to the origin point of the marking area (0,0).
Enter coordinates directly or use the Jog key

Arc text types

Consider the following factors when defining fields with text on arc:

- Increasing the radius of the arc decreases the curvature of the text;
- Reducing the radius of the arc increases the curvature of the text;

- Small adjustments in the alignment parameters can reposition the text along the path of the arc. It is a good idea to test for changes;
- If an arched text is slightly outside the marking area, it is advisable to change the alignment of the text;
- If the arc text is *convex*, try the alignment at the top or in the middle;
- If the arch text is *concave*, try alignment down or centre.

3.14.5 Parameters of the datamatrix field

The datamatrix field allows a text message to be encoded according to a two-dimensional matrix symbol, readable by a special device, and marked as part of the message. The datamatrix field uses an algorithm based on the Data Matrix Library, which is present in the system. There are several parameters for defining and formatting datamatrix messages.

- The successful reading of Matrix symbols is highly dependent on the marking. The optimum situation is that the points of the matrix symbol almost

touch. Therefore, a special relationship is required between the diameter of the marked dot and the spacing between the marked dots.

The screenshot shows a software interface for configuring a datamatrix field. It is organized into two main sections: 'Generals' and 'Advanced'.
 In the 'Generals' section:
 - 'Type' is a dropdown menu set to 'DATAMATRIX'.
 - 'GS1 mode' is an unchecked checkbox.
 - 'Format' is a dropdown menu set to 'Auto'.
 In the 'Advanced' section:
 - 'TEXT' is a text input field, currently empty, with a 'VARIABLE TEXT' button to its right.
 - 'X=' and 'Y=' are numerical input fields, both set to '0.00'.
 - 'Vertical dimension:' is a numerical input field set to '0.00', accompanied by a green checkmark icon.
 - 'Spacing:' is a numerical input field set to '0.00'.
 - 'Angle' is a numerical input field set to '0'.
 - 'Alignment' is a dropdown menu set to 'BOTTOM LEFT'.
 - 'Speed' is a slider control set to '100%'.
 - 'Modalita' di esecuzione:' is a dropdown menu set to 'SPIRALE'.

Datamatrix parameter setting page

Text

The text parameter defines which information is to be included in the matrix symbol. It may include upper case, lower case, keyboard symbols, spacing and flag commands. Flag commands are used to include the current date, time, serial numbering in the message to be marked.

The content of the text string will determine the format selection (see below). In addition, the combination of the format and length of the text string will determine the geometric shape of the matrix symbol (square or rectangular).

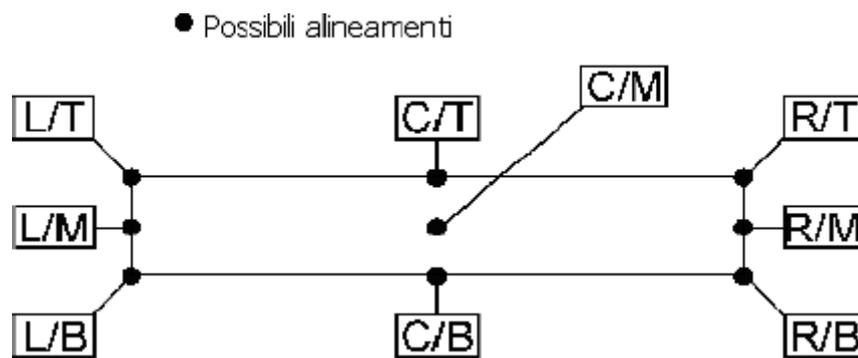
X, Y

The X and Y co-ordinate values define the position where the field will be placed within the marking area. The co-ordinates are calculated in relation to the origin point of the area (0,0) and the field will be aligned depending on how the relevant parameter has been set.

Set coordinates directly in the fields or use the Jog key. Defines the point

Justification

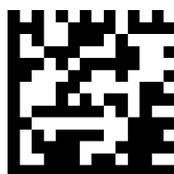
to which the coordinates are referenced with respect to the field.



- L/T (Left/Top) - Justification top left**
- L/M (Left/Middle) - Left centre justification L/B (Left/Bottom) - Left bottom justification**
- C/T (Centre/Top) - Top centre justification**
- C/M (Center/Middle) - Justification in centre C/B (Center/Bottom) - Justification at bottom**
- R/T (Right/Top) - Justification at top right**
- R/M (Right/Middle) - Right centre justification R/B (Right/Bottom) - Justification bottom right**

Type

The type parameter defines the datamatrix type to be used. It is possible to set the standard datamatrix type or the QR type.



Datamatrix

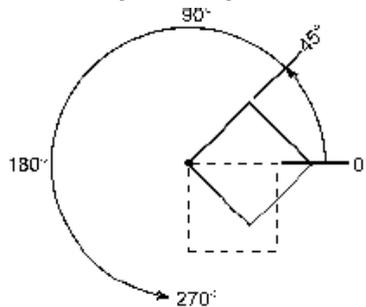


QR

Format The format parameter defines the geometric form of the character a **SQUARE** equal number of columns and rows

- 8 X 18** 8 rows , 18 columns
- 8 X 32** 8 rows , 32 columns
- 12 X 26** 12 rows, 26 columns
- 12 X 36** 12 rows, 36 columns
- 16 X 36** 16 rows, 36 columns
- 16 X 48** 16 rows, 48 columns

Corner Specify the rotation angle of the symbol, relative to the anchor point. The 0° reference point (no rotation) corresponds to at 3 o'clock on the dial. Increasing the angle will rotate the dial anti-clockwise



Allineamento a sinistra/in alto

Rotation also depends on the alignment parameters.

Vertical dimension

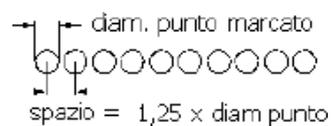
Enter the dimension (in inches or millimetres, depending on the unit of measurement chosen). The vertical dimension parameter allows you to set the desired height of the marked matrix. The system will automatically calculate the spacing and width of the symbols.

Note the dimension parameter measures the distance between the centre of the highest point and the centre of the lowest point.



Space

Enter the space dimension (in inches or millimetres, depending on the unit of measurement chosen). The space parameter allows the operator to control the specific spacing of the points made by the marking head. If the diameter of the marked dot is known, set the space to a value equal to 125% of the dot diameter.



Quality

Defines the marking quality according to the set marking head. Depending on the selected quality, a more or less defined marking can be achieved. This will affect the performance of the machine.

Speed

The parameter controls the marking speed of the datamatrix field only. The speed set is relative to the speed of the entire marking of the field. For example, setting the speed of the datamatrix field to 100% will result in marking the datamatrix field at the same speed as marking all other fields in the programme. Setting the speed to 50% means marking the datamatrix field at half the marking speed of the programme. Higher percentages result in faster marking cycles.

Modalities Execution

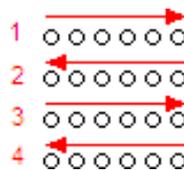
The execution mode parameter allows the execution of marking the datamatrix field to be changed.

In particular, the modalities of execution are as follows:

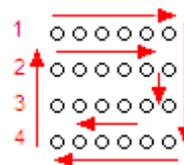
- **STANDARD**: the field is always executed line by line from left to right, like this:



- **INVERSE**: the datamatrix field will be executed in a more continuous manner, i.e. always line by line but in the opposite direction to the direction of execution of the previous line, like this:

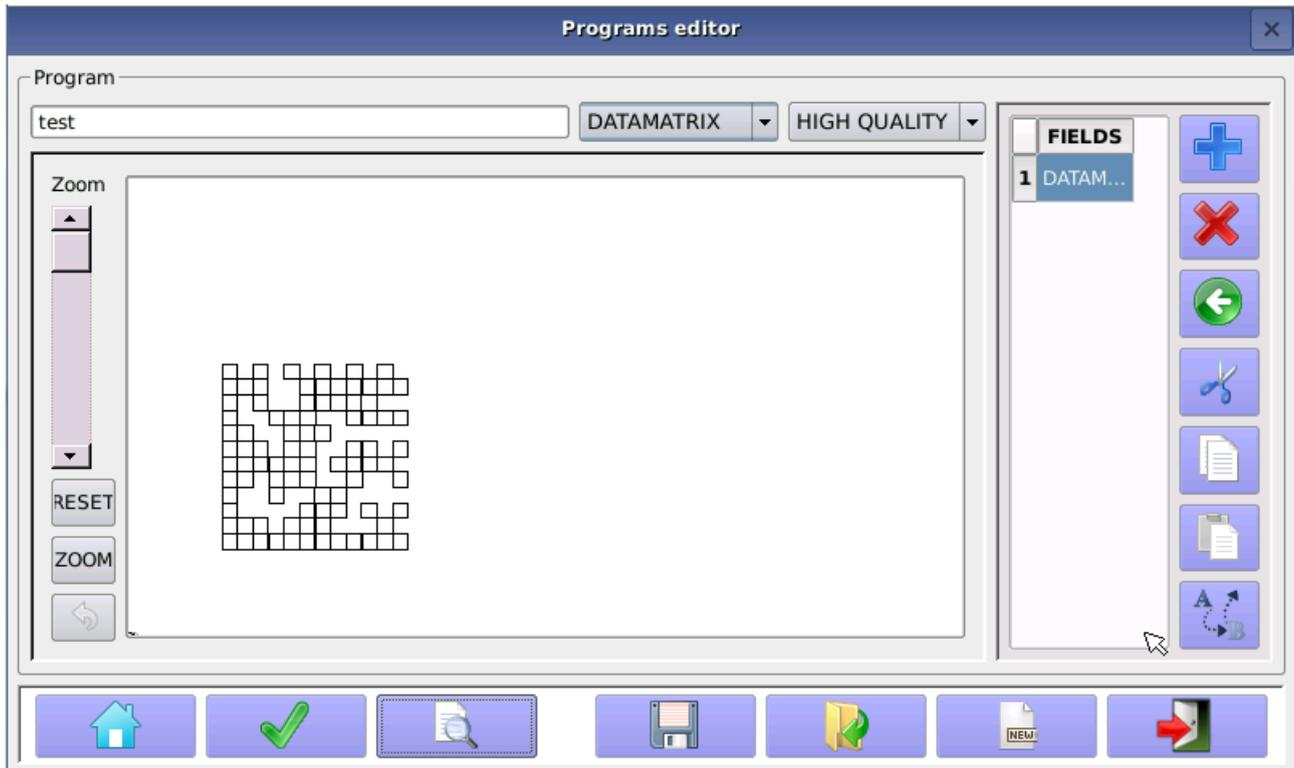


- **SPIRAL**: The datamatrix field will be executed in such a way that the points of the datamatrix will be marked with a spiral path originating at the top left point of the grid and rotating in a CLOCKWISE direction until the points are completed, like this



The execution of the datamatrix field also depends on a parameter in the general software data. This parameter will indicate a datamatrix execution offset, which will represent how far the marking head must move for the execution of a new datamatrix line. If this parameter is 0, the head will move directly to the execution position of the new line of the datamatrix field.

The datamatrix field can also be used on a laser machine. The execution of this field is done in a different way on the laser machine. The execution points will be replaced by squares as shown in the image below:



Screenshot with datamatrix on laser machine

For the laser machine, it is possible to select the datamatrix filling functionality. If font filling management is enabled (see the *general parameters* section in the manual), a button will be displayed allowing you to set certain parameters for the filling function.



Filling parameter configuration button

Pressing the fill parameter configuration button will open a screen where you can set the values of these parameters.



Filling parameter configuration screen

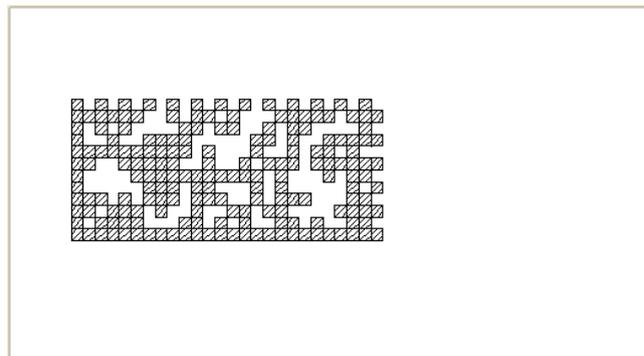
Marking type indicates the type of marking to be used for the following field. The values that can be set are: *border*, *fill* or *both*.

Marking sequence can only be set if the marking type is set to *both*. Indicates the order in which the marking type is executed. Settable values are: *edge->fill* (indicates that the border is executed first and only then the fill) and *fill->edge* (indicates that the fill is executed first and only then the border).

Orientation indicates the orientation to be performed for the filling function. The values that can be set are: *horizontal*, *vertical*, *diagonal* and *crossed*.



Filling pitch indicates the distance between one filling line and the next. A low pitch value indicates a higher concentration of fill lines in the field, conversely, a high pitch value indicates a lower concentration of fill lines in the field.



Example of field with high fill pitch



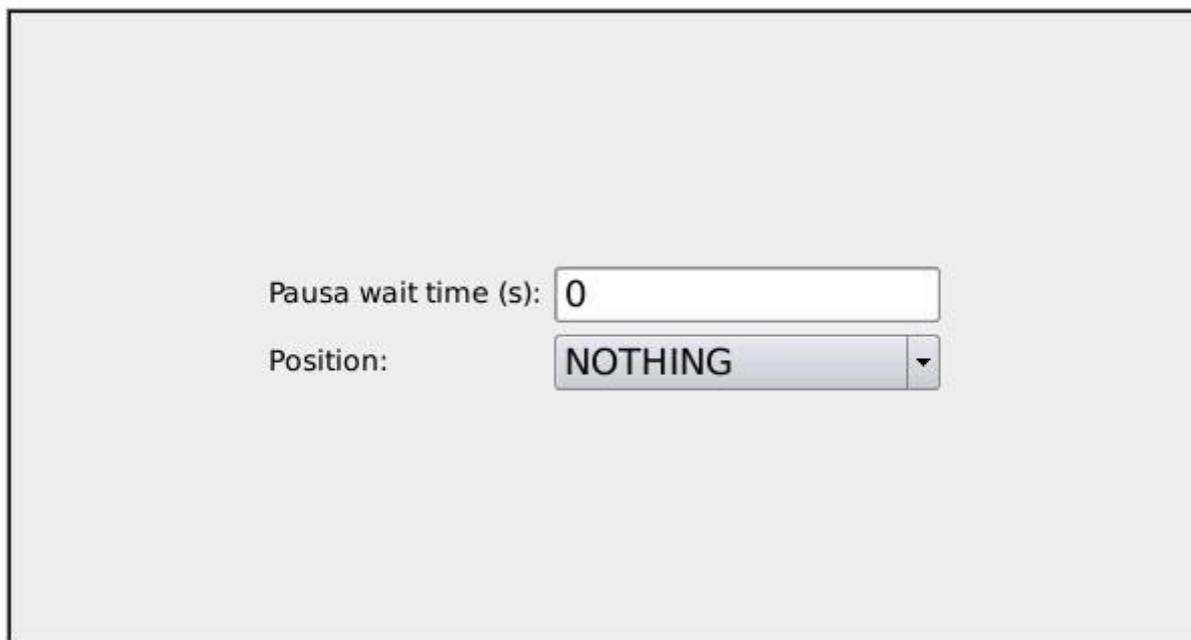
Example of field with low fill pitch

Tilt 1 fill can only be set if the orientation type is set to *diagonal* or *crossed*. Indicates the angle of inclination of the filling lines.

Tilt 2 filling can only be set if the orientation type is set to *cross-fill*. Indicates the angle of inclination of the secondary lines in the cross-fill.

3.14.6 Pause Field Parameters

A pause can be programmed during the marking cycle by including a corresponding field in the programme. The system will suspend the marking operation until a new marking command is received or once the pause waiting time set in the pause field parameters has passed.



Pause field parameter setting page

If the wait time is 0 (zero), the system will wait for a new marking command to resume the cycle. Otherwise, if the wait time is a value other than 0, the marking system to resume the cycle will wait either for a new marking command or for the passage of the set time.

It is also possible to set the position to which the marking head must be brought when executing a pause field. This allows you to customise the behaviour during a pause field at will and can be useful depending on the markings to be made. The pause-field position can be set as follows:

- **NONE:** If no position is chosen, the marking head will remain in the position it was in when the break camp was executed;
- **NEXT FIELD:** If you choose to place the head on the next field, the marking head will be placed on the starting position of the field following the break field.
- **PARKING:** If you choose to position the head on the parking position, the marking head will position itself on the parking position set for the current programme.

3.14.7 Goto field parameters

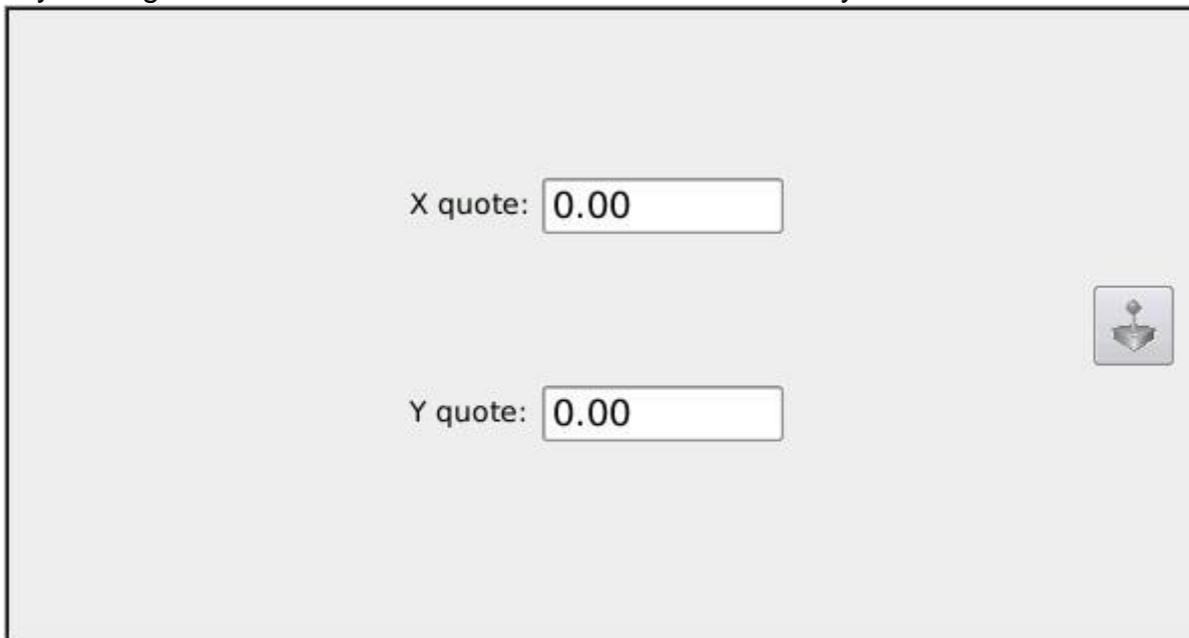


In the message you can a Goto camp,

the marking head will move from the current position to the new position whose co-ordinates are defined in the Goto field. This is particularly useful when obstacles have to be overcome in the marking area. No marking is carried out with the Goto field.

If auxiliary axes are set, it will be possible to move all axes with the Goto field. The only note concerns the possibility of moving the set auxiliary axis at the same time

with the Y-axis, which is not possible as the auxiliary axis is switched with the Y-axis. However, it is possible to set movement priorities or disable an axis in the Goto field. All this makes the movements to be made during a marking cycle totally customisable. Movement priority management cannot be set without an enabled auxiliary axis.



GoTo field data setting page

- X** The X parameter defines the co-ordinate on the X-axis where the head will position itself when it receives the Goto command. The co-ordinate is calculated relative to the origin point of the area (0,0). Set the X-axis co-ordinate or use the Jog key.
- Y** The Y parameter defines the co-ordinate on the Y axis where the head will position itself when it receives the Goto command. The co-ordinate is calculated relative to the origin point of the area (0,0). Set the Y-axis co-ordinate or use the Jog key.
- Z** The Z parameter defines the co-ordinate on the Z axis where the head will position itself when it receives the Goto command. The co-ordinate is calculated relative to the axis origin point. Set the Z-axis co-ordinate or use the Jog key. *Only set if Z is enabled.*
- T** The T parameter defines the co-ordinate on the Theta axis where the head will position itself when it receives the Goto command. The co-ordinate is calculated relative to the axis origin point. Set the Theta-axis co-ordinate or use the Jog key. *Only set if Theta is enabled.*
- X1** The X1 parameter defines the co-ordinate on the X1 axis where the axis will position itself when it receives the Goto command. The co-ordinate is calculated relative to the axis origin point. Set the X1 axis co-ordinate or use the Jog key. *Only set if X1 is enabled.*
- Y1** The Y1 parameter defines the co-ordinate on the Y1 axis where the axis will position itself when it receives the Goto command. The co-ordinate is calculated relative to the axis origin point. Set the Y1 axis co-ordinate or use the Jog key. *Only set if Y1 is enabled.*

Priorities

The Priority parameter defines the priority of moves in the Goto field. Lower values indicate a higher priority for movements. If two axes are set with the same priority, they will be commanded simultaneously. *It can only be set if an auxiliary axis is enabled.*

3.14.8 Bitmap image field parameters

The bitmap image field allows you to insert an image in bitmap format (file extension *.bmp*) into the programme. From the editor environment, it is possible to select one of the images in the controller memory (to import image files, see next chapter) and set some parameters for the representation of the image in the marking area.

When inserting a bitmap image field, some important considerations must be taken into account.

First of all, bitmap images must be saved in monochrome format. Of each *.bmp* file, all and only the 'coloured' pixels are marked by rows. For example, let us consider a bitmap of 640x480 pixels with, of the total 307200, 50000 'coloured' pixels. Regardless of the size in mm set for marking the image in the programme on the control, 50000 markings will be made. Changing the size of the image will only change the density of the marked points. It is therefore evident that already during the PC construction of the image, a bitmap must be created consciously, i.e. there is no point in marking a 640x480pixel bitmap in 2x1mm.

It should also be borne in mind that if the density is too high, the marking speed is automatically reduced to ensure the correct timing of the solenoid valve, with the effect of making the execution of the cycle even slower.

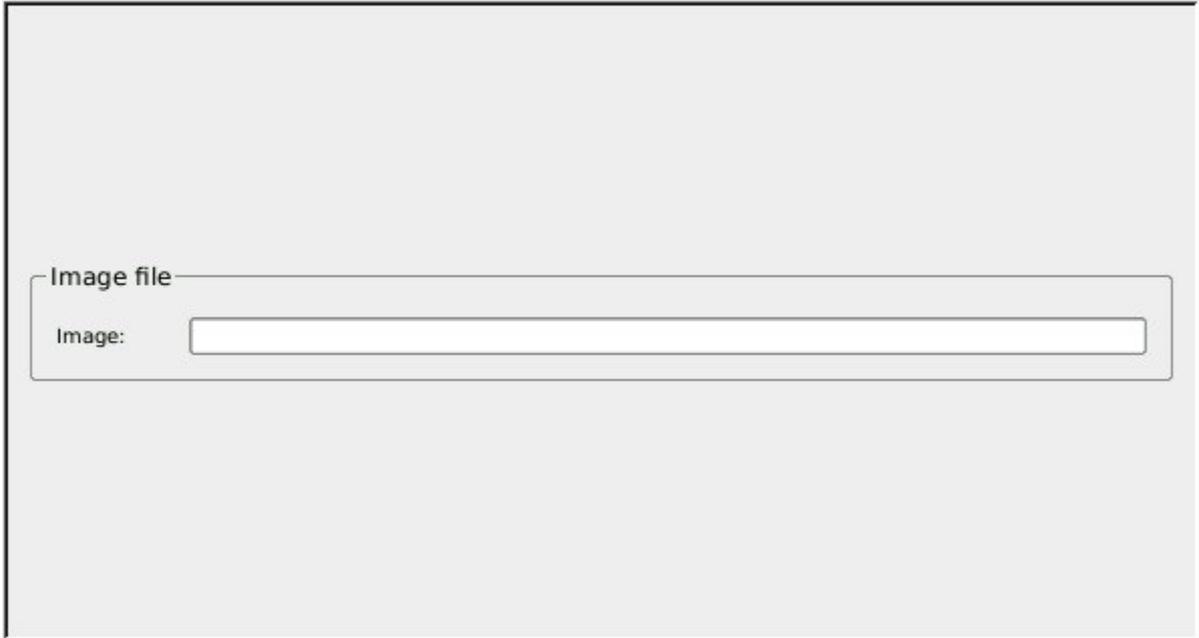
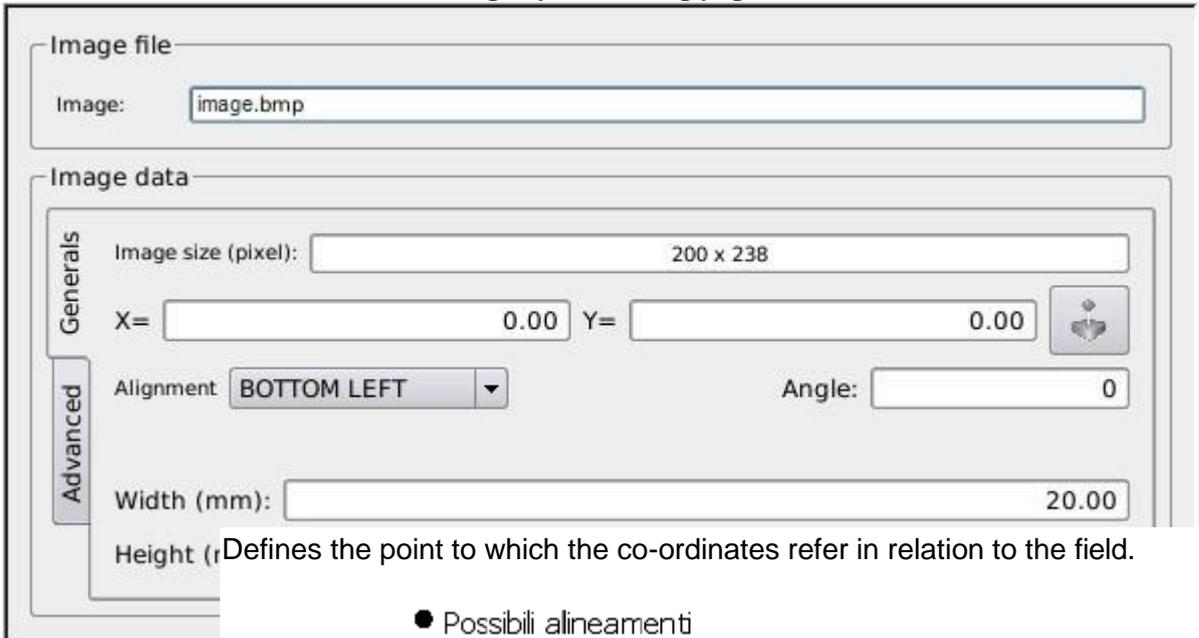
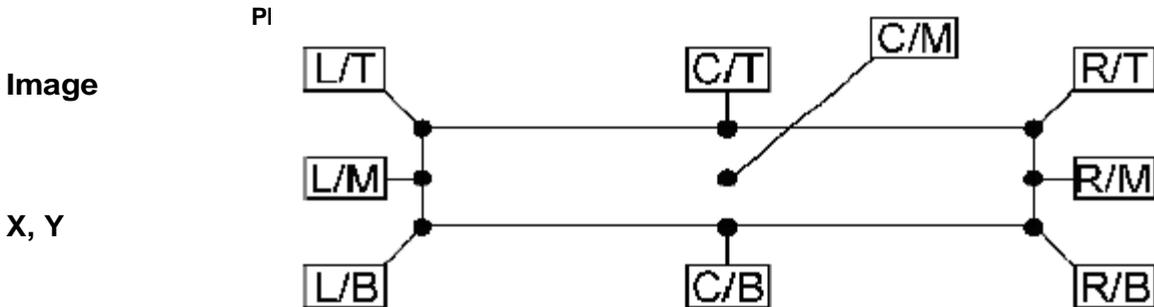


Image upload setting page



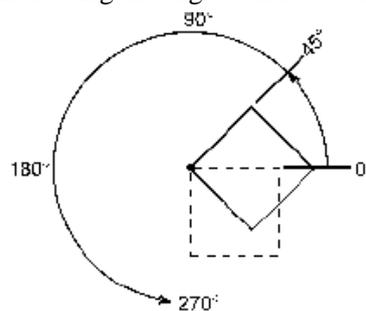
● Possibili allineamenti



- L/T (Left/Top) - Justification top left**
- L/M (Left/Middle) - Left centre justification**
- L/B (Left/Bottom) - Left bottom justification**
- C/T (Centre/Top) - Top centre justification**
- C/M (Center/Middle) - Justification in centre**
- C/B (Center/Bottom) - Justification at bottom**
- R/T (Right/Top) - Justification at top right**
- R/M (Right/Middle) - Justification in centre right**
- R/B (Right/Bottom) - Justification in bottom right**

Corner

Specify the angle of rotation of the image, relative to the anchor point. The 0° reference point (no rotation) corresponds to at 3 o'clock on the dial. Increasing the angle will rotate the dial anti-clockwise



Allineamento a sinistra/in alto

Rotation also depends on the alignment parameters.

Width

The width parameter defines the width in millimetres of the image that will be represented on the marking area. Changing this parameter will change the density of the points to be marked on the workpiece.

Height

The height parameter defines the height in millimetres of the image that will be represented on the marking area. Changing this parameter will change the density of the points to be marked on the workpiece.

Quality

Defines the marking quality according to the set marking head. Depending on the selected quality, a more or less defined marking can be achieved. This will affect the performance of the machine.

3.14.9 Comment field parameters

A comment can be programmed during the marking cycle by including a corresponding field in the programme. The system will pause the marking operation, as in the pause field, but the cycle will only resume when the OK button in the message window with the comment field is pressed.



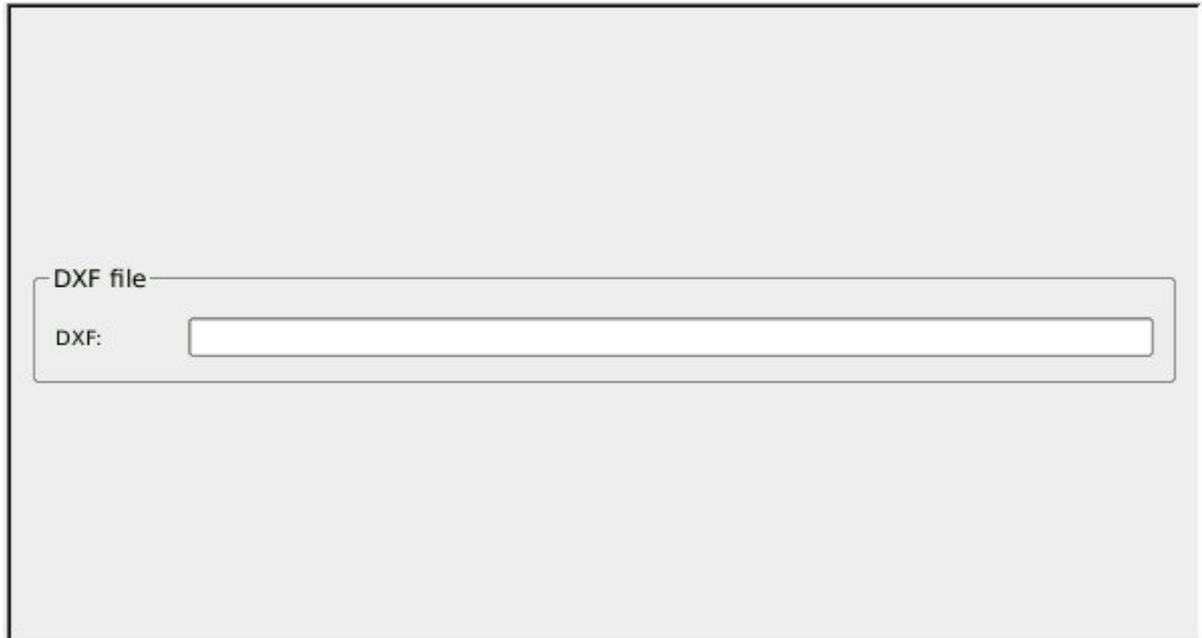
Page for setting the comment field parameters

Comment

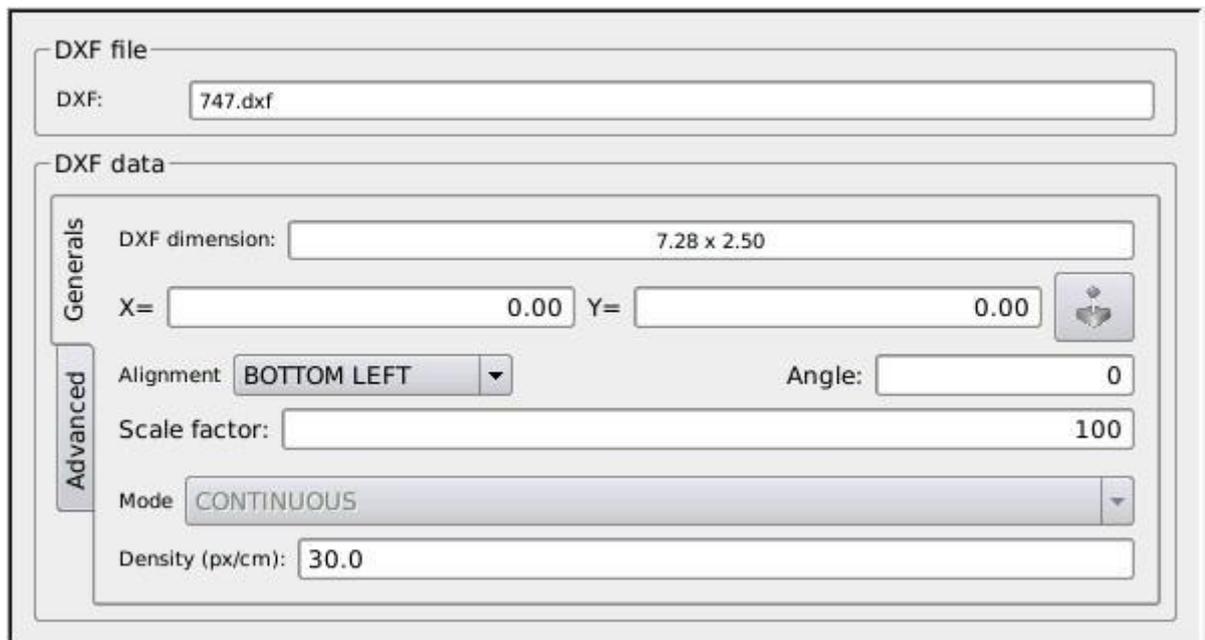
The parameter defines the text to be displayed in the message window that will appear when the comment field is executed during the marking cycle.

3.14.10 DXF field parameters

The DXF field allows you to insert a drawing in DXF format (file extension *.dxf*) into the programme. From the editor environment, it is possible to select one of the drawings in the controller memory (to import drawing files, see next chapter) and set some parameters for the representation of the drawing in the marking area.



Setup page of the DXF drawing to be represented



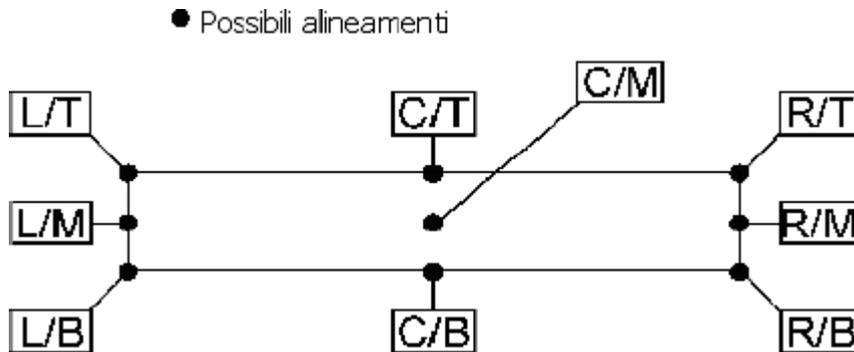
DXF drawing parameter setting page

DXF The dxf parameter defines the drawing file to be represented in the marking area. The drawing file is selectable from the files in the controller's memory.

X, Y The X and Y co-ordinate values define the position where the field will be placed within the marking area. The co-ordinates are calculated in relation to the origin point of the area (0,0) and the field will be aligned depending on how the relevant parameter has been set. Set coordinates directly in the fields or use the Jog key.

Justification

Defines the point to which the co-ordinates are referenced with respect to the field.



L/T (Left/Top) - Justification top left

L/M (Left/Middle) - Left centre justification L/B

(Left/Bottom) - Left bottom justification C/T (Centre/Top)

- Top centre justification

C/M (Center/Middle) - Justification in centre in centre C/B

(Center/Bottom) - Justification at bottom in centre

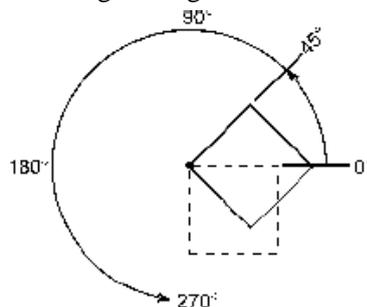
R/T (Right/Top) - Justification at top right

R/M (Right/Middle) - Justification in centre right R/B

(Right/Bottom) - Justification in bottom right

Corner

Specify the angle of rotation of the DXF, relative to the anchor point. The 0° reference point (no rotation) corresponds to at 3 o'clock on the dial. Increasing the angle will rotate the dial anti-clockwise



Allineamento a sinistra/in alto

Rotation also depends on the alignment parameters.

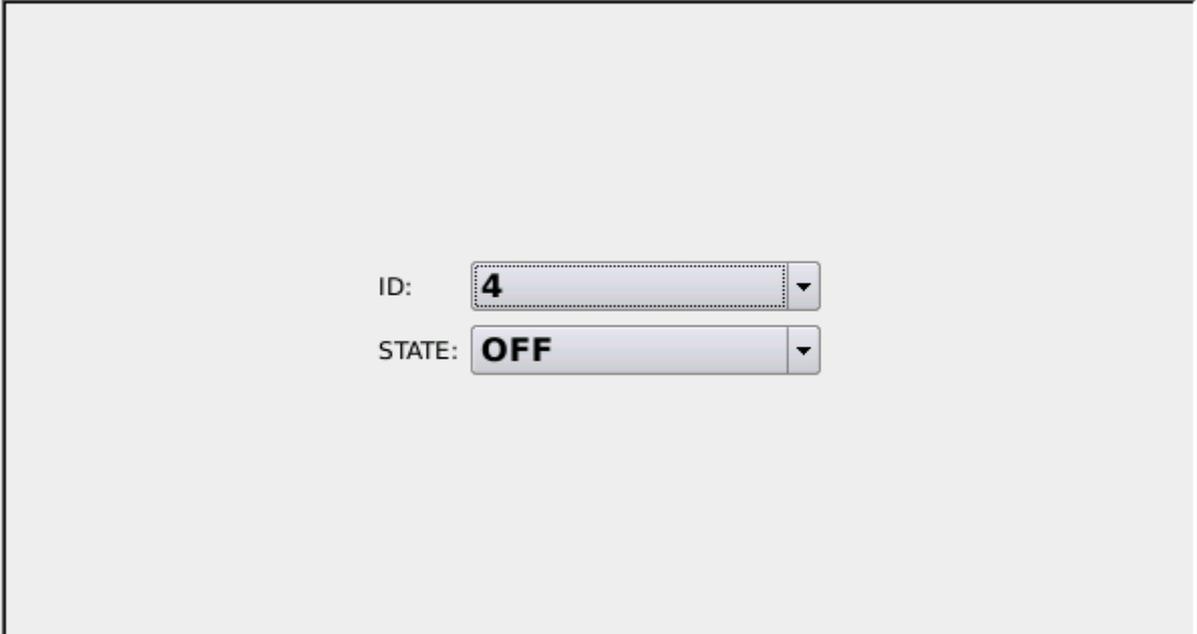
Scale factor	Defines the scale factor (in percent) with which the design will be represented in the marking area. A scale factor of 100% will result in the representation of the design in its actual scale with which it was created.
Mode	Defines the marking mode with which the current field will be executed.
Density	If the continuous mode is set, the dot density (dot density per inch or per centimetre) can be defined. The higher the dot density, the longer the marking time and faster the plunger wear.
Quality	Defines the marking quality according to the set marking head. Depending on the selected quality, a more or less defined marking can be achieved. This will affect the performance of the machine.

3.14.11 Automation field parameters

The automation field is used to make it possible to create external, customised logic during the execution of a marking cycle. It is possible to consider the automation field as a kind of programme field which allows logic external to the machining done by the AC500 control unit to be managed. At the moment, the pause field allows you to set the value of a digital output taken outside the control unit (see chapter on I/O signals in this manual) obviously only if this output is free and therefore not used by the software.

In the case of a normal machine configuration, three digital outputs, number 4, number 5 and number 6 (which are the output signals on PINs 18, 19 and 20 respectively) can be used to take signals out of the control unit during a marking cycle. The operation that can be done on an output is simply to set the status, i.e. whether to enable (ON) or disable (OFF) the output.

As mentioned above, the automation field only allows the use of free outputs, which are therefore not used by the control unit software. Precisely for this reason, it is important to bear in mind that outputs 5 and 6 are already used by the software in the case of the ALEA machine and that it will therefore not be possible to use them in the automation field in this configuration.



Automation field parameter setting page

ID Defines the index numeric index of the output from use for the field of automation. In this window, outputs that cannot be used will be filtered out.

Status Defines the ON/OFF status to be set at the selected output for the automation field.

In the marking phase, when the software has to manage an automation field, the logic entered during programming of the field will be processed.

NOTICE

The software does not manage any kind of control over these outputs, which means that it will be up to the operator to set the correct values for the states at the end or beginning of the machining cycle.

3.15 Flag Commands

The system allows 'flag' commands to be inserted into the text string, so that dynamic data such as the day and time can be marked in different formats. There are also 'flag' commands to insert serial numbers and coded data into the message to be marked, according to a table defined by the operator.

The 'flag' commands consist of:

% - percentage -

Letter - for each particular command - upper or lower case

Table 3-1. Flag' commands

FLAG	DATA ENTERED IN MARKING
%A	Shortened day of the week (Mon, Tue, Wed, Thu, Fri, Sat, Sun)
%B	Shortened month (Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec)
%C	Date and time in MM/DD/YYYY format
%D	Day of the month (01 - 31)
%d	Code of the Day
%E	Code of the Year
%F	Single Character Check Digit for PS-OCR
%G	Double Character Check Digit for PS-OCR
%H	Time of day (00 - 23)
%h	Time Code
%I	Time of day (00 - 12)
%J	Julian Day of the Year (001 - 366)
%j	Special Julian Day of the Year (see features below Store)
%K	Code of the month (1 - 9, O, N, D)
%L	Last number of the year (0 - 9)
%M	Month (01 - 12)
%N	Minutes of the hour (00 - 59)
%n	Current seconds (00 - 59)
%#O	Serial number
%#o	Serial number
%P	AM or PM
%#Q	Inserts text from the query table (# query number)
%#q	Inserts title from query table (# query number)
%R	Number of the week (01 - 53) [week 01 is the week containing 01 January].
%r	Number of the week (01 - 53) [week 01 contains the first Thursday of the year].
%#S	Programme serial number
%#s	Programme serial number
%T	Time (HH:MM or HH:MM:SS) [mode settable from general software parameter].
%t	Field from CSV file
%U	Code of the month
%#V	Variable text
%#v	Variable text
%W	Day of the week (where: 1 = Sunday ... 7 = Saturday)
%w	Day of the week (where: 1 = Monday... 7 = Sunday)
%X	Date (MM/DD/YY)
%Y	Year r (00 - 99)
%Z	Shift code
%%	Inserts the character '%' in the string
%#=#	Inserts the text of the specified field into the current string (#)
%#@	Inserts the specified ASCII character (#) into the current string

General Information

The total number of characters in a string is influenced by the characters entered by the 'flag' commands.

Entering the date and time

There are several 'flag' commands that correspond to different date and time formats. The system clock is the source of date and time information.

Error Checking

Flag' commands can be used to perform an error check operation when the marked programme is read by an OCR reader. The control 'flag' command may refer to one or two characters at the end of the field string. The characters of

control are calculated by a system based on the hexadecimal sum of the characters entered in the string. The final check character is entered into the programme. The PS-OCR reader can then read the string, perform its check by calculation and verify that the check characters in the programme match the result of the sum performed. The %F and %G 'flag' commands, if used, must be placed at the end of the text string . The command

%F inserts a single control character at the end of the programme. The %G command inserts two control characters. The single control character command must be used when space prevents the use of two control characters.

Coded information

Some 'flag' commands allow you to define codes to represent the current year, month or shift.

Text entered by the operator

The system offers the operator two ways of entering data into a message: query and variable text:

- **Queries** to enter data that are common to several programmes;
- **Variable text** To insert data proper to a field.

Flag' commands referring to text variables may be part of a field or constitute a field in its own right.

Settable commands can be of two formats: **%#V** or **%#v**. In particular:

is a value defining the maximum number of characters attributable variable text fields may have up to 50 characters. (e.g., %50V or %50v) If this data element is omitted, the field will assume a variable length to the set data element.

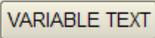
V (upper case). The field will be filled with spaces if the operator enters fewer fields than the variable.

v (lower case). The field will NOT be filled with spaces if the operator enters fewer fields than the variable.

Example:

Campo	Dati dell'Operatore	Marcatura
SAMPLE_%5VABC	12345	SAMPLE_12345ABC
SAMPLE_%5VABC	123	SAMPLE_123 ABC
SAMPLE_%5vABC	12345	SAMPLE_12345ABC
SAMPLE_%5vABC	123	SAMPLE_123ABC

To insert a variable text into a text, it is necessary:

1. Create a field in the programme that allows text to be entered;
2. Enter in the field text parameter the flag command that calls up the variable text;
3. This will enable a 'Variable Text' button  next to the text box. Pressing this button allows you to access the variable text data setting window.



Variable text data setting window

4. Set the variable text and select whether to enable automatic cleaning. If auto-cleaning is enabled, the variable text will be deleted automatically at the end of marking the field. If auto-cleaning is disabled, the variable text will not be modified until the operator makes a change to the variable.
5. Save the programme using the appropriate button in the editor environment.

Serial Numbers

The system allows two types of serial numbers to be entered:

- **Omni Serial Numbers** Serial numbers common to several programmes;
- **Serial Numbers Pattern** Serial numbers proper to a single programme;

The serial number flag can be placed within a text string or it can be part of a separate field.

Omni Serial Number can be given by one of the following formats: **%#O**, **%#o**, or **%o**

Serial Number Pattern can be given by one of the following formats: **%#S**, **%S**, **%#s**, or **%s**

In particular:

- %** introduces the flag command.
- #** 1 to 9 to define how many digits the serial number consists of, which the system will fill with spaces or zeros in front of the unit, depending on the flag command chosen.
- S / O** upper case - defines the type of serial numbering. The system will mark zeros in front of the unit if an integer from 1 to 9 is specified.
- s / o** lower case - defines the type of serial numbering. The system will leave spaces in front of the unit if an integer from 1 to 9 is specified.

For example, if the text string is set as follows:

ABC%4SABC

in marking you will have:

ABC0001ABC

where the 4 digits of the serial number are set at the flag command level. Both the lower and upper levels of the serial number and the flag command should be set correctly. Otherwise, it could happen that at parameter level the number 99999 is expected to be reached, while the serial number only has 4 digits marked. Therefore, once marked:

ABC9999ABC

you will have marking:

ACC0000ABC

because the serial number was set to 4 digits and not 5 digits. The number 1 of 10000 is thus truncated.

The following table allows you to consider the different cases of serial number marking depending on the flag command set:

To insert a variable text into a text, it is necessary:

1. Create a field in the programme that allows text to be entered;
2. Enter the flag command recalling serial number in the field text parameter;
3. Save the programme using the appropriate button in the editor environment.

3.16 Interaction and positioning via the jog environment

It is possible to use the jog environment to define the position of a field within the marking area. Whenever the system displays a screen where coordinates can be set, the jog environment recall button will be activated.



Button to call up the jog environment

NOTE

The marking head must be set to online status in order to position the fields with the jog.

To position a field via the jog environment:

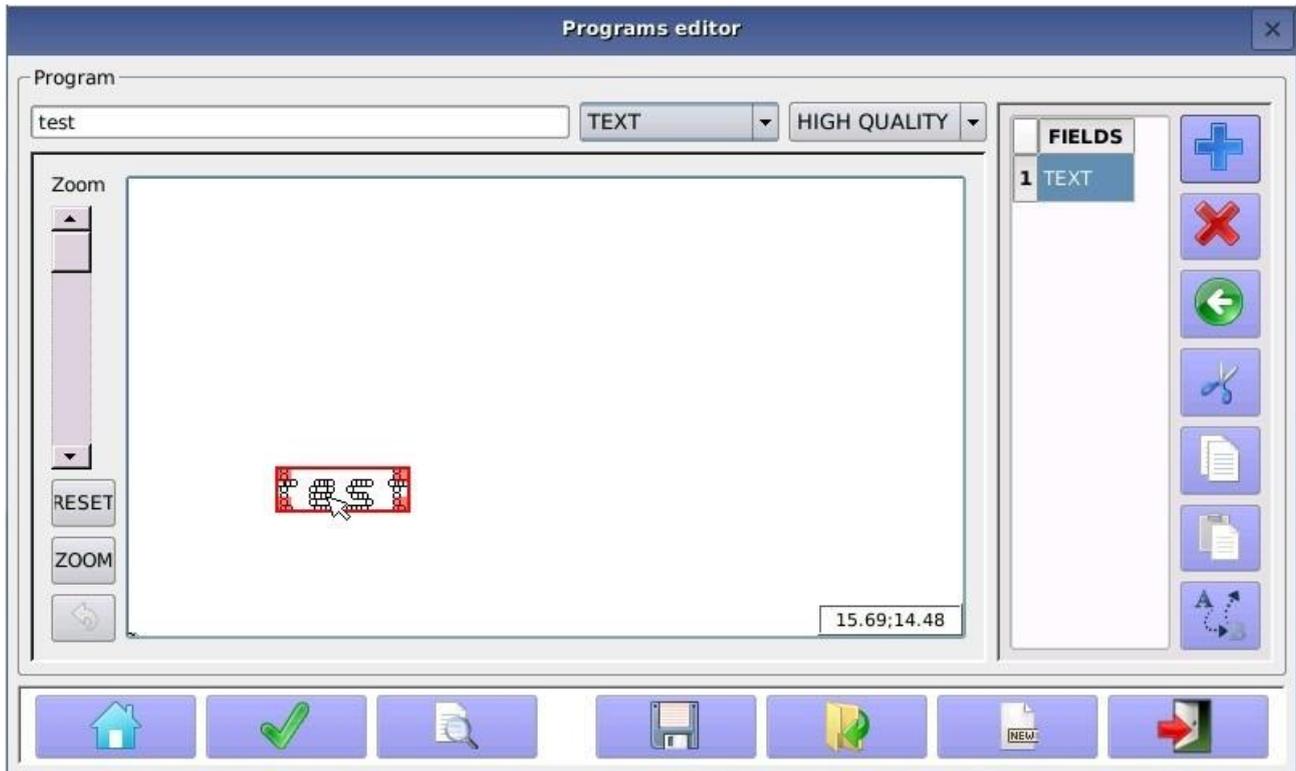
1. Press on the button to call up the jog environment;
2. From the jog environment, position the marking head to the desired position;



3. Press the capture button to confirm .

3.17 Edit programme fields from preview

From the programme preview in the editor environment, it is possible to reposition fields or change their size. If you select a field in the preview, by simply pressing on the screen in the vicinity of the displayed field, you will be able to move the field to the marking area to facilitate the positioning of fields in the programme.



Editor environment screen with preview view

Once you have selected the field you wish to edit, the permitted operations are:

- Moving the field within the working area;
- Resizing the field with reference to a corner of the rectangle in which the field is inscribed.

During a field movement operation, the current actual altitude at which the field is located is displayed.

It is also possible to undo a move or resize operation of one or more fields by pressing



NOTE

After an operation to move or resize the preview, if you return to the programming environment or perform any other operation that does not concern editing from the preview, you will no longer be able to undo the operations previously performed.

NOTE

The resize operation of a field is not possible on fields of type arc, text on arc, fields containing a rotation angle and fields that cannot be displayed (GOTO, Comment, etc.).

3.18 Modifying a programme

The system offers several ways of modifying programmes once they have been defined and stored in the controller. In particular, it is possible to load a programme, test it, create a duplicate or delete it.

3.18.1 Testing a programme

It is possible to test a programme to verify that all fields are within the marking area. The system examines each field, in the order in which they were defined. If there are any errors in the test, the system will warn of the error with a message window.

To test a programme, it is necessary:

1. Load the programme you want to test;
2. From the programme editor environment, press the compile button ; 
3. It will be displayed a screen a window of messages with the outcome of the compilation;
4. If the compilation returns an error, check and edit the incorrect field so that it is in the marking area and relaunch the compilation.

3.18.2 Saving a programme

New programmes or changes to existing ones are not stored in the permanent memory until they are saved. The save function allows programmes to be copied and saved under a different name.

NOTE

One must save a programme before loading another. Otherwise, you risk losing the changes you have made. It is a good idea to save your work from time to time to avoid losing what you have entered.

To save a programme, it is necessary:

1. From the programme editor environment, make sure you have entered a name for the current programme;
2. Press the programme save button  ;

3. If a programme with the set name already exists in the controller memory, you will be asked whether to overwrite the existing programme.

3.18.3 Changing programme parameters

Certain parameters and options are stored within the programme. Programme parameters only affect the current programme. Parameters can be defined both when creating a programme and when editing the programme.

3.18.4 Variation of field parameters

Each field in a programme is defined by a variety of parameters that control its positioning, appearance and content. It is possible to make changes whenever necessary.

3.18.5 Changes in programme fields

It is possible to insert, edit, delete operations on any field in the programme. It is also possible to cut, copy and paste a field into the programme. These operations are accessible via the buttons next to the field table in the editor environment.

3.19 File Management

For the user with the operating level set to *supervisor*, *distributor* or *manufacturer*, it is possible to access the management environment of files (programs, bitmap images, dxf drawings, TTF fonts, custom fonts, CSV files and virtual keyboards) present and stored on the controller memory. From this environment, it is possible to import, export, delete, copy or rename files on the controller memory.

It is possible to enter the file management environment in this way:

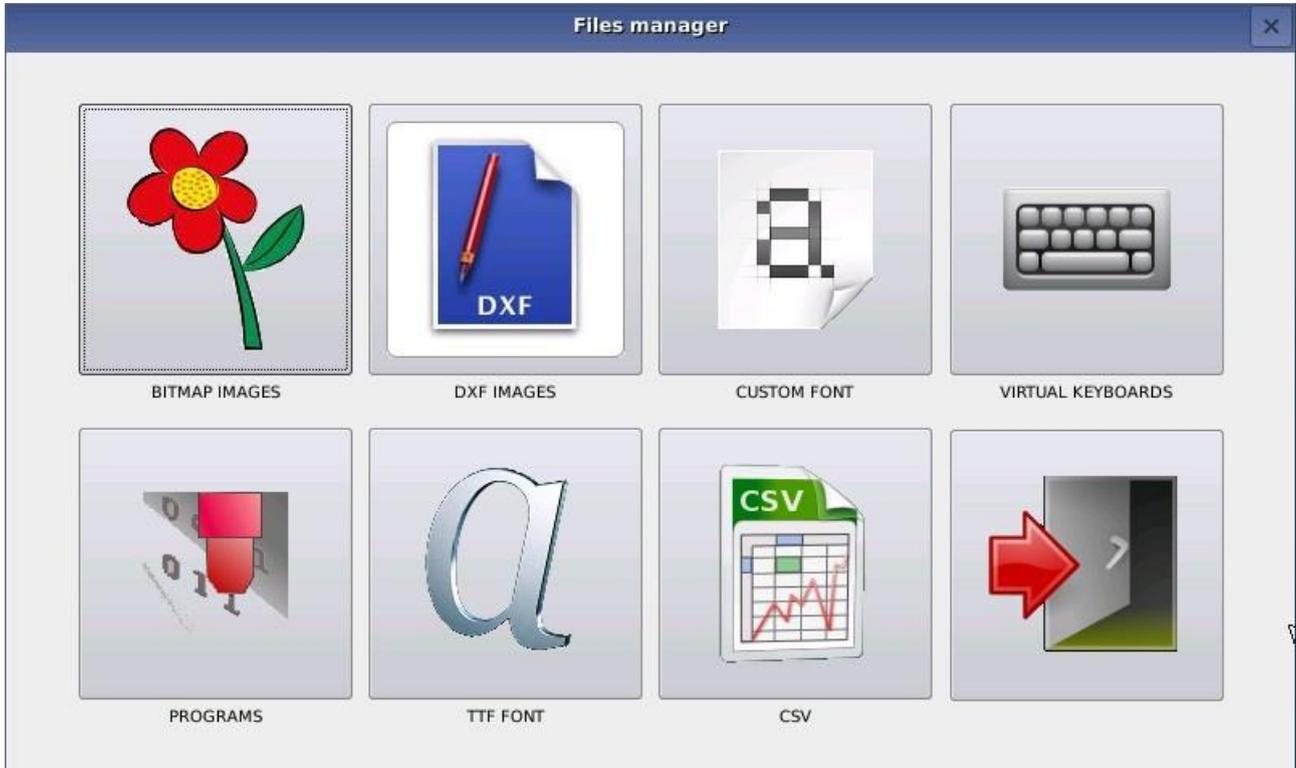
1. From the main screen press the button to access the auxiliary functions of the



2. From menu of functions auxiliary functions press on button of *management file*



In this environment, it will be possible to enter into the management of the various types of files managed by the software.



File Management Environment Screen

For each type of file, it will be possible to perform the following operations.



1. Table of files in memory.
2. Filter that can be set on the display of files in the table.
3. Buttons for selecting/deselecting all the files in the table.
4. Button for deleting selected files in the table.
5. Button for copying the selected file.
6. Button to rename the selected file.
7. Button to export the selected file to the USB stick.
8. Button to import one or more files from the USB stick.
9. Button to exit the programme management environment.

NOTICE

Deleting or renaming a bitmap image, a dxf drawing or a TTF font can cause problems in the execution of programmes if there was a reference to one of these files in them.

3.20 Controller network management

The software allows you to configure and use a network of control units so that they can share marking data information via Ethernet. In particular, these control units can mark serial numbers of the programme or omni sequentially, as if these data were managed by a single control unit. By

As a matter of fact, the controller network works in such a way that the information on the serial numbers to be marked is stored and managed by a single controller (called *master*) and all the other controllers in the network will simply request the serial number to be marked at the start of the cycle (*slave* controllers). To configure the control unit's network settings, please refer to the appropriate section in the manual.

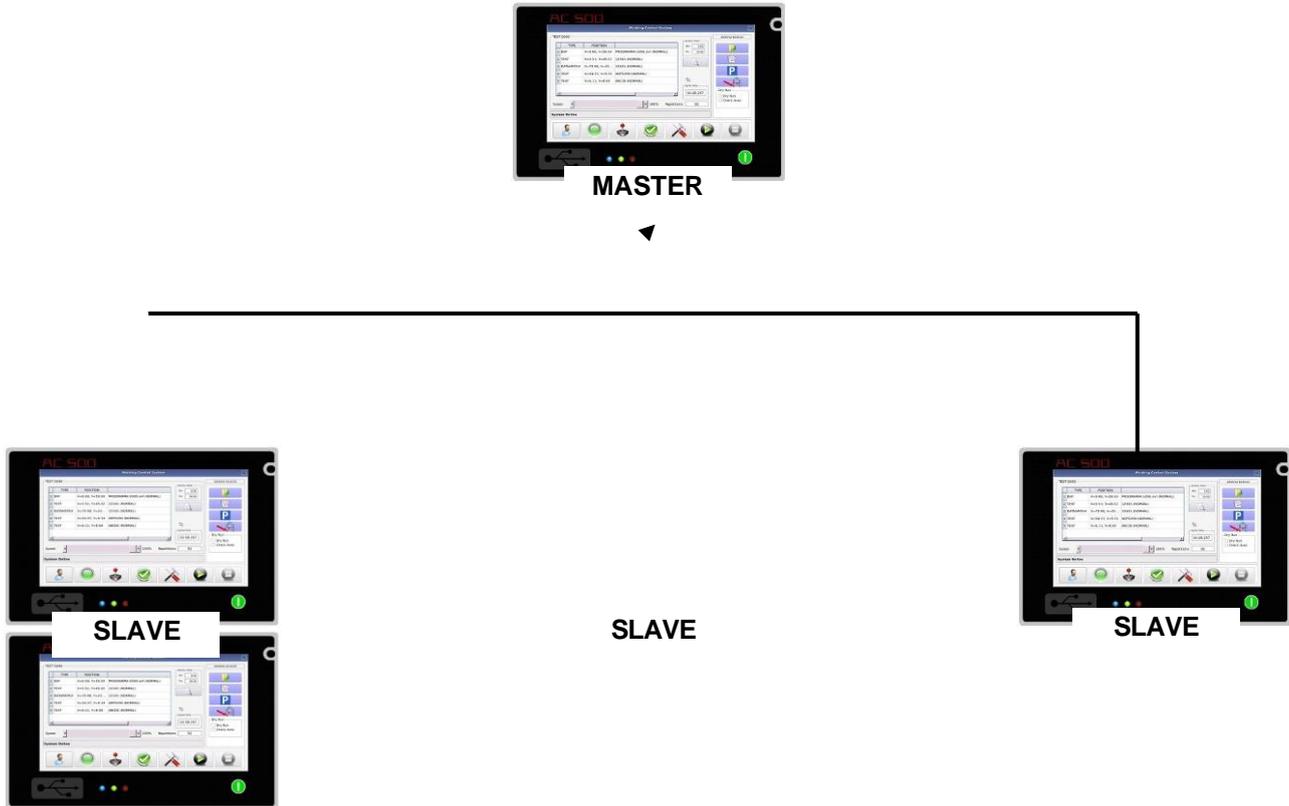
A controller network absolutely must have one master controller, while it can have an indefinite number of slave controllers. The same network can also have several masters (which cannot communicate with each other) and each master can have one or more connected control units. This can allow a certain flexibility on the management of a network as one group of control units can be set up to work on a certain group of data and another group to work on others. The possible configurations and uses are varied and totally customisable according to requirements.

Looking at the communication between master and slave in more detail, it can be said that the master, by itself, does not communicate with any control unit, but waits to receive communication from outside (*server* function). Conversely, the slave control unit will attempt to connect to the set master control unit at start-up, until the connection is successful (*client* function). Once the connection request has been received by the master control unit, the connection between the two control units will be opened and the exchange of information between the two will begin. The slave controller will be in charge of making the requests and the master controller will respond accordingly.

The slave control unit will continuously send an *ALIVE* packet to inform the master control unit that the control unit is alive and that communication should therefore be kept open. If the master does not receive the *ALIVE* packet within a certain *time (timeout)*

this will consider the slave controller as not connected and will therefore close the connection. Since all communication is based on a request and response system between the slave and the master, if the slave controller does not receive confirmation of the *ALIVE* packet received from the master, it will timeout and close the communication with the master.

When the slave control unit has to execute a programme containing serial numbers, it will process a request to the master control unit. Upon receiving this request, the master control unit will analyse the data to be marked and send them to the slave control unit which processed this request. At the end of the execution cycle the slave control unit will send the result of the cycle just executed to the master: if it was successful then the data sent by the master will be considered as executed and therefore the relative serial counter will be increased, whereas if the end-of-cycle result was negative (abort, error, etc.) the serial numbers sent to the slave control unit will be reused for a new request. This behaviour will be carried out even if a slave control unit in possession of the data disconnects.



To run a network cycle on a slave controller, it is necessary that

- The control unit is connected to the master;
- The loaded programme contains programme serial numbers or omni;
- The loaded programme is the same as the one loaded on the master control unit.

If a slave is configured and connected with a master, it can still continue to execute programmes that do not contain serial numbers without passing through the network. In this mode on the slave control unit, access to the settings of the programme and omni serial numbers is not permitted, as they must necessarily be managed by the master control unit.



Work screen with serial number management from network to slave

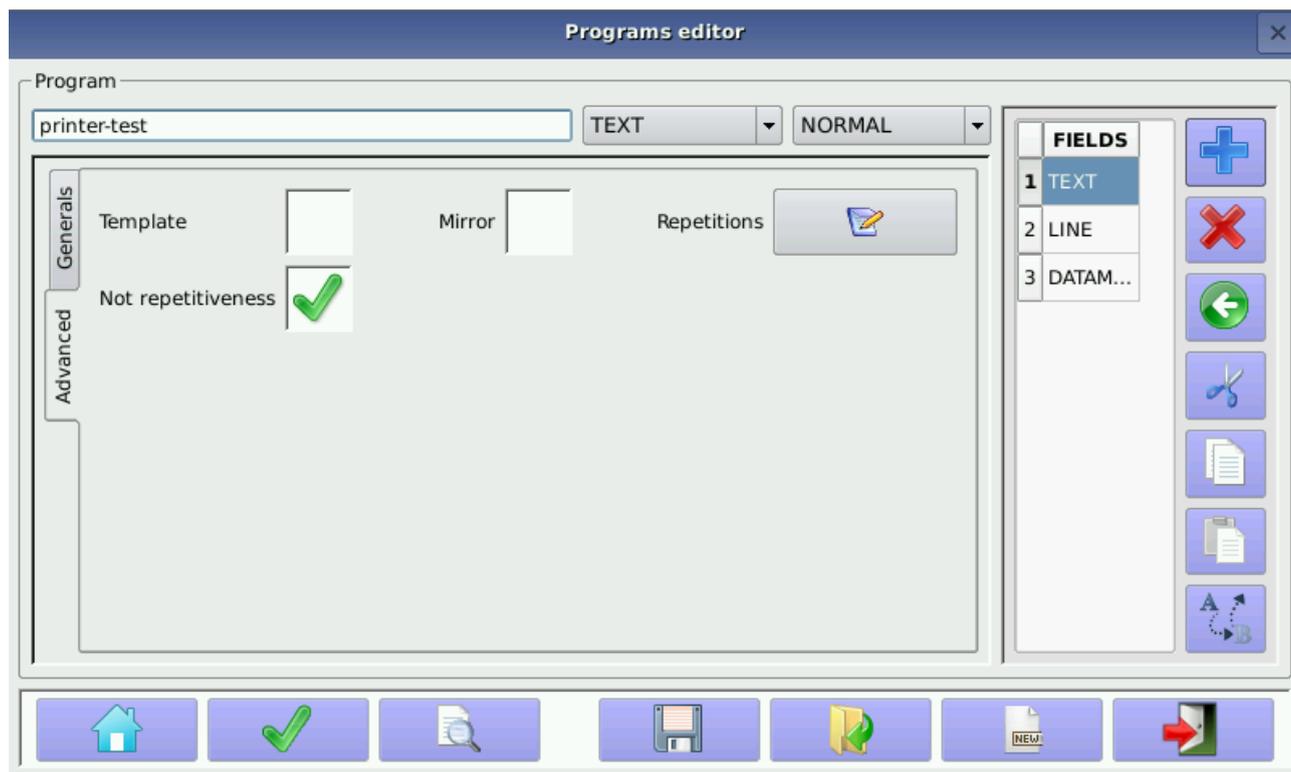
In the programme preview, the flag for the field to be marked will be displayed and not the actual value, as the value of the serial to be marked will only be requested at the start of the cycle (same handling on the master side).



Work screen with cycle execution with serial number from network

3.21 Non-repeatability check

In the programme editor, if an element of type TEXT, TEXT ARC or DATAMATRIX is selected, it will be possible to display the choice for the non-repeatability marking check. This function will only be enabled if it is purchased from the Store and the relevant general parameter is enabled. For further information, please refer to the chapter on general software parameters.



Screen for non-repeatability check

In the advanced section it will be possible to enable the control for the selected field. This control can only be enabled for a single field in the programme.

If multi-field management is enabled, an error message will be displayed at the start of a marking cycle.

By enabling this control, at the end of a marking, the software will save in memory the textual content information of the field just marked.

At the beginning of each marking, if the current programme contains a field with a non-repeatability check enabled, a check will be made to assess whether the contents of the field have already been marked previously; if so, a message will be displayed notifying the user whether or not to mark the field.

From the backup management environment, it will be possible to delete the history file of fields that have been marked as non-repeatable.

CHAPTER 4

System configuration

The parameters and options that control system configuration are valid universally and during all operations. Unlike the parameters of the fields contained in the programme definition and which are only specific to that particular programme, the system parameters are common to all programmes and all related system operations. This chapter presents the system parameters and the operations to act on them.

- Define Password
- Defining the text query
- Defining the Omni serial number
- Defining shift codes
- Defining the codes of the day
- Defining Month Codes
- Defining year codes
- Defining time codes
- Set *i* parameters of head of marking
- Creating and managing the machine database
- Set the day and time
- Set the unit of measurement
- Setting start-up parameters
- Setting up host communications
- Enabling remote pattern selection
- Creating system backups
- Updating the software
- View the marking history

4.1 Accessing System Parameters

The software must be placed at the *Manufacturer* level to change any parameter or configuration of the system. Some settings can also be changed at the *Distributor* operating level and to a lesser extent at the *Supervisor* level.

1. From the main screen of the software, set the user operating level to Manufacturer (or Distributor or Supervisor, but be aware that you cannot change all system parameters);

- If the operating level is password-protected, it must be entered in the appropriate screen that appears on the screen.

2. Enter the auxiliary functionality environment.

4.2 Definition of Omni serial numbering

Omni serial numbering allows the same numerical sequence to be continued in the marking of different programmes. Unlike the serial programme numbers, whose parameters are stored in the programme definition itself, the serial Omni numbering is stored in the *permanent memory*, making it possible to use this value in all programmes containing the flag command of '%O'. The serial number Omni is incremented each time the programme containing the %O flag command is marked. If you set the serial number increment greater than zero (>0), the serial number Omni automatically resets to the set lower limit of numbering when the upper limit has been reached. Conversely, if you set the serial number increment smaller than zero (<0), the serial number Omni automatically resets to the set upper limit of numbering when the lower limit has been reached. Optionally, the serial number Omni can be programmed to reset at a certain time of day set in the parameters. Serial number setting is also possible for *Supervisor* and *Distributor* users.

To set the Omni serial number:

1. From the auxiliary functionality screen, press the user setup button;



User setup button

2. A screen will open showing the possible user parameters that can be changed;



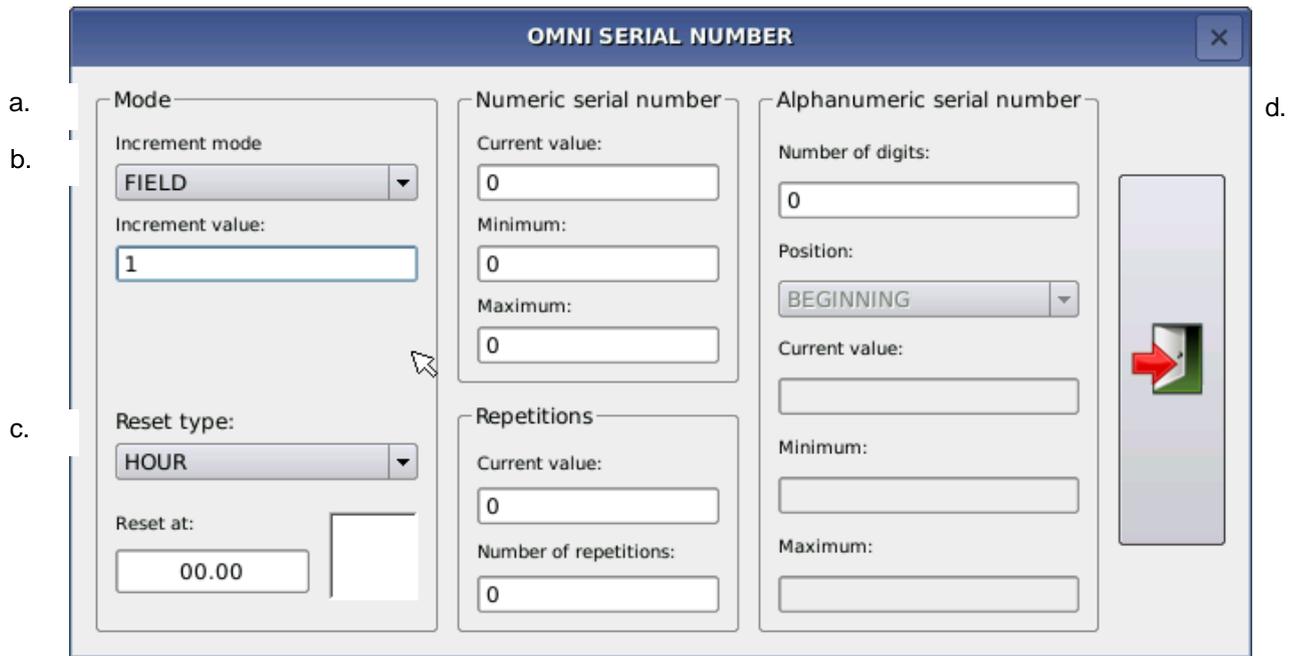
User parameter configuration screen

To change the serial number parameters, press the 'Omni serial number' button.



Omni serial number button

3. A new screen will open where you can set the parameters of the Omni serial number.



Omni serial number parameter configuration screen

a. In this screen you can set how the serial number is to be used:

- **Increasing modes**

Parameter indicating how the serial number is incremented.

- **PATTERN**

Increment mode according to programme. The serial number will only be incremented, by the value set as increment, at the end of the programme execution.

- **FIELD**

Increment mode based on the programme field. The serial number will be incremented, by the value set as increment, at the end of the execution of a field using the serial number.

- **Reset mode**

Parameters indicating the reset mode to be used and the relevant enabling for the Omni serial number. The software also allows the type of reset to be used to be configured (by time, by day, etc.) and according to the different mode selected, the relevant values can be entered.

- **NOW**

Allows you to set the reset time for the omni serial number. For example, if 02:00 is set as the reset time, the software will reset the omni serial number as soon as it recognises that this time has passed.

- **DAY**
Allows you to set the day of the week for resetting the omni serial number. For example, if you set Monday as the reset day, the software will reset the omni serial number as soon as it recognises that that day has passed.
 - **DATA**
Allows you to set the day of a month to reset the omni serial number. For example if you set the 15th of the month as the reset date, the software will reset the omni serial number as soon as it recognises that that date has passed. If 31 of the month is set as the date, the last valid day of the month will be taken in months with fewer days.
 - **TURNO**
Allows the omni serial number reset shift to be set. For example, if you set the second shift as the reset shift, the software will reset the omni serial number as soon as it recognises that this shift has passed.
- b. In this window it will be possible to set the current value of the serial number and the minimum and maximum value within which the serial value will be set. The current value must be between the set minimum and maximum value. In addition, the set minimum value must be less than the maximum value.
If the serial number increment is positive (>0), the serial number will be reset to the set minimum limit. If the serial number increment is negative, the serial number will be reset to the maximum limit set.
- c. In this window, it will be possible to set the parameters required to manage the number of repetitions of the serial number.
- **Current value**
Parameter indicating the current number of repetitions performed
 - **Number of repetitions**
Parameter indicating how many repetitions are to be performed of the current serial number.

These two values are used to determine the number of times the system should mark the current serial number. The current value of the number of repetitions indicates how many times it has already been executed. When this value reaches the set number of repetitions then the serial number will be incremented (according to the set setting).

If the number of repetitions is left at 0 (zero), no such management will be used.

If the number of repetitions of the serial number is used, the handling specified in the general parameter '*Abort handling for serial number repetitions*' will be used in the event of a cycle abort (see section *Setting General Parameters* in the software).

- d. In this window, it will be possible to set the parameters required to manage the alphanumeric values within the serial number.

Number of digits

Parameter indicating how many alphanumeric digits are to be managed within the serial number. If the number of digits is 0 then the alphanumeric value parameters will not be enabled.

- Positioning in the serial number

Parameter indicating where to place the alphanumeric digits within the serial number.

• START

Placement of alphanumeric digits at the top of the serial number field.

In this case, the serial value increment management, and the checks on the maximum values, will also be modified and referred to the values of the alphanumeric digits. In this case:

if we have the serial value 'A101', where 'A' is the alphanumeric value, when incrementing, if the numeric value reaches the maximum value set in the serial value parameter, the numeric value will be set equal to the minimum value and the alphanumeric value will be incremented (or decremented) by one digit. For example:

Minimum value	=	10
Maximum numerical value	=	101
Numerical current value	=	101
Increase	=	+1
Alphanumeric value	=	"A"

The result of the serial value 'A101' incremented will be: 'B10'.

In this case, checks on the mode of the serial value limit will refer to the maximum values of the alphanumeric value.

• END

Placement of alphanumeric digits at the end of the serial number field.

In this case, the serial value increment management, and the checks on the maximum values, will also be modified and referred to the values of the alphanumerical digits. In this case:

if we have the serial value '101A', where 'A' is the alphanumeric value, when incrementing, the alphanumeric value will be incremented (or decremented) by one digit. For example:

Minimum value	=	10
Maximum numerical value	=	101
Numeric current value	=	101
Increase	=	+1
Alphanumeric value	=	"A"

The result of the serial value '101A' incremented will be: '101B'.

If the alphanumeric value reaches the set maximum alphanumeric value, the numeric value will be incremented by the set increment value and the alphanumeric value will be increased to its minimum value

- **General alphanumeric digit parameters**
General parameters for setting the current value, minimum value and maximum value of alphanumerical digits.

4. Once the desired parameters have been set, exit the environment to save the changes made.

4.3 Definition of shift codes

The Flag %Z shift code command allows you to insert a code within a programme to be marked to designate which work shift is active at the time of marking. When the system encounters one of these commands, referring to its internal clock, the system will mark the character that has been defined in the shift table and that corresponds to one of the three work shifts. In the shift table, the operator can enter any single character corresponding to a work shift. In the user shift table, the start and end time of each shift is entered. Finally, it is possible to display a counter of the number of programmes executed in each work shift. For each finished programme, the corresponding counter of the current shift will be incremented.

Serial number setting is also possible for *Supervisor* and *Distributor* users.

Example. It is possible to define the shift table as 'PST' (i.e. First, Second, Third). If the programme containing the shift code flag is marked at the first shift, the system will automatically insert the letter P where %Z is.

Consideration No. 1 If the third shift starts at or after midnight (e.g. from 12 a.m. to 8 a.m.) the operator should set the TPS table as, starting at midnight, it is chronologically the first working shift.

Consideration No. 2 If fewer than three shifts are used, set the board to ignore missing shifts. For example, if there are two shifts, set the table to PSS (First, Second) so that it marks P for the first shift, and S for all others.

To configure shift codes:

1. From the auxiliary functionality screen, press the user setup button;



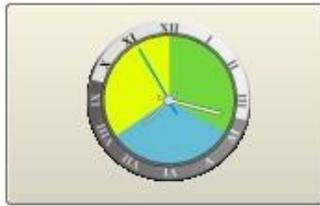
User setup button

2. A screen will open showing the possible user parameters that can be changed;



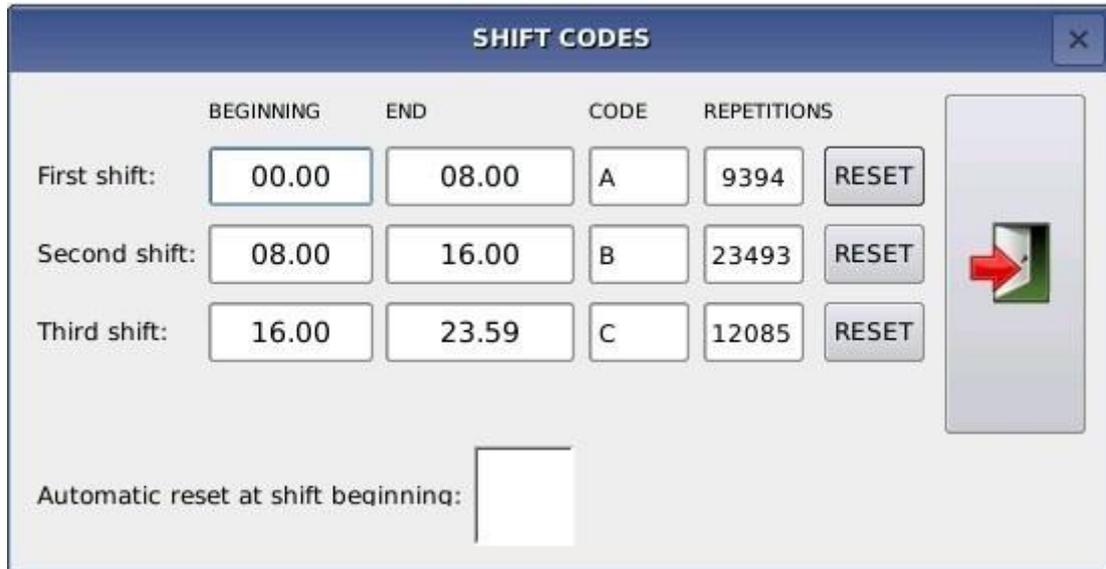
User parameter configuration screen

To change the shift code parameters, press the 'shift code' button.



Shift code button

3. A new screen will open where you can enter the shift code setting parameters



Shift code parameter setting screen

In this screen you can set the start and end times of the three shifts and the code associated with each of them. Shift start times cannot be changed as they are calculated according to the end time of the next shift set. Of course, changing the end time of the third shift will result in changing the start time of the first shift. It is possible to display next to each set shift the number of programmes executed in the time slot of that shift. The counters for the number of programme repetitions can be reset either manually, by pressing the *RESET* button located next to the relevant counter, or automatically at the beginning of each shift, by enabling the appropriate check at the bottom of the window.

4. Once the desired parameters have been set, exit the environment to save the changes made.

4.4 Defining the codes of the day

It is possible to enter 31 variables, corresponding to the 31 days of a month. If the flag %d is present in the field to be marked, the letter corresponding to the current day will be marked. For example, after setting the days table

with ABCDEFGHIJKLMNOPQESRUVWXYZ12345, if the current day is 3, marking the flag %d will result in the letter C being the third letter set as it is the third day of the month. Setting the day code is also possible for *Supervisor* and *Distributor* users. To configure the day code

1. From the auxiliary functionality screen, press the user setup button;



User setup button

2. A screen will open showing the possible user parameters that can be changed;



User parameter configuration screen

To change the month code parameters, press the 'month code' button.



Day code button

3. A new screen will open where you can enter the day code setting parameters

DAY CODES X

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
Q	R	S	T	U	V	W	X	Y	Z	1	2	3	4	5		

Day code parameter setting screen

On this screen you can enter the code associated with each day of the month.

Once the desired parameters have been set, exit the environment to save the changes made.

4.5 Definition of month codes

It is possible to enter 12 variables, corresponding to the 12 months of the year. If the %U flag is present in the field to be marked, the letter corresponding to the current month will be marked. For example, after setting the month table with ABCDEFGHIJKL, if the current month is March, in marking the %U flag you will have the letter C which is the third letter set as March is the third month of the year.

Setting the month code is also possible for *Supervisor* and *Distributor* users. To configure the month code

1. From the auxiliary functionality screen, press the user setup button;



User setup button

2. A screen will open showing the possible user parameters that can be changed;



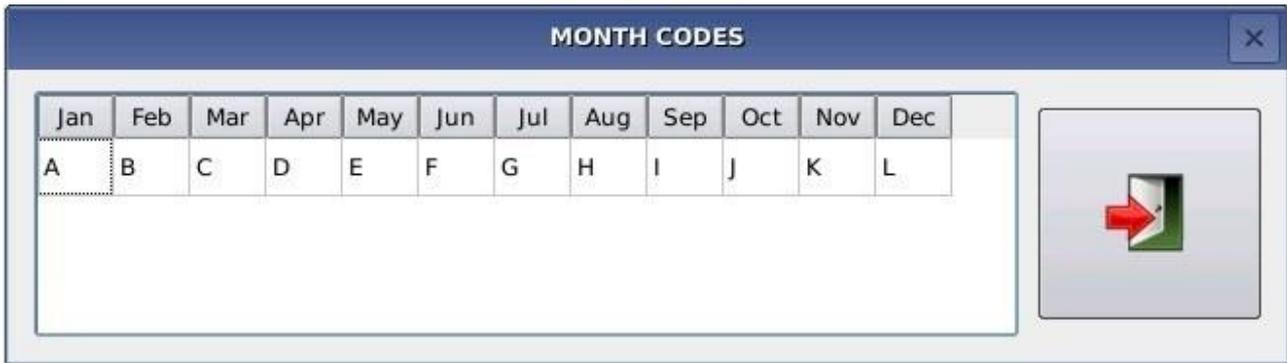
User parameter configuration screen

To change the month code parameters, press the 'month code' button.



Month code button

3. A new screen will open where you can enter the parameters for setting the month code

**Month code parameter setting screen**

In this screen you can enter the code associated with each month of the year.

- Once the desired parameters have been set, exit the environment to save the changes made.

4.6 Definition of year codes

It is possible to enter 10 variables, corresponding to the final digit of the year from 0 to 9. If the %E flag is present in the field to be marked, the letter corresponding to the current year will be marked. For example, after having set the year table with ABCDEFGHIJ, if the current year is 2010, in marking the %E flag you will have the letter A which is the first letter as 0 is the final digit of the year.

Setting the year code is also possible for *Supervisor* and *Distributor* users. To configure the year code

- From the auxiliary functionality screen, press the user setup button;

**User setup button**

- A screen will open showing the possible user parameters that can be changed;



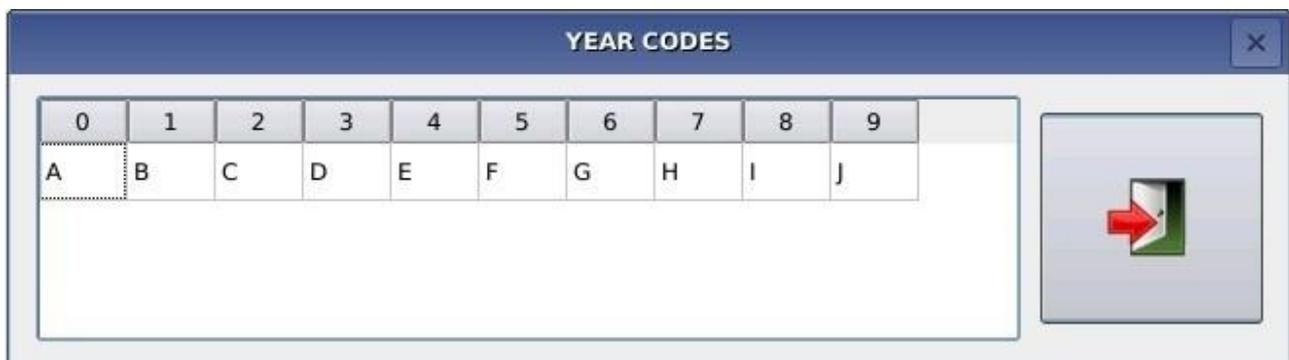
User parameter configuration screen

To change the year code parameters, press the 'year code' button.



Year code button

3. A new screen will open where you can enter the year code setting parameters



Year code parameter setting screen

On this screen you can enter the code associated with each year.

4. Once the desired parameters have been set, exit the environment to save the changes made.

NOTE

The first character set represents the first year of the decade (0).

4.7 Definition of time codes

It is possible to enter 24 variables, corresponding to the hours in a day from 0 a 23. If the flag %h is present in the field to be marked, the letter corresponding to the current time will be marked. For example, after setting the time table with ABCDEFGHIJKLMNOPQRSTUVWXYZ, if the current time is 08:30, marking the %h flag will result in the letter I.

Setting the time code is also possible for *Supervisor* and *Distributor* users. To

configure the time code

- From the auxiliary functionality screen, press the user setup button;



User setup button

- A screen will open showing the possible user parameters that can be changed;



User parameter configuration screen

To change the time code parameters, press the 'time code' button.



Time code button

- A new screen will open where you can enter the time code setting parameters

0	1	2	3	4	5	6	7	8	9	10	11	
A	B	C	D	E	F	G	H	I	J	K	L	
12	13	14	15	16	17	18	19	20	21	22	23	
M	N	O	P	Q	R	S	T	U	V	W	X	



Time code parameter setting screen

In this screen you can enter the code associated with each hour.

- Once the desired parameters have been set, exit the environment to save the changes made.

4.8 Defining the calendar

It is possible to change the calendar management to include holidays to be considered for the management of the special Julian code. To enable calendar management, the special Julian code functionality must be enabled in the Store and the calendar management parameter must be enabled. If a special Julian code is requested on a holiday, the Julian code will not be incremented (regardless of the increment set); the special Julian code will automatically be incremented to the next day of the holiday.

Code setting is now also possible for *Supervisor* and *Distributor* users. To

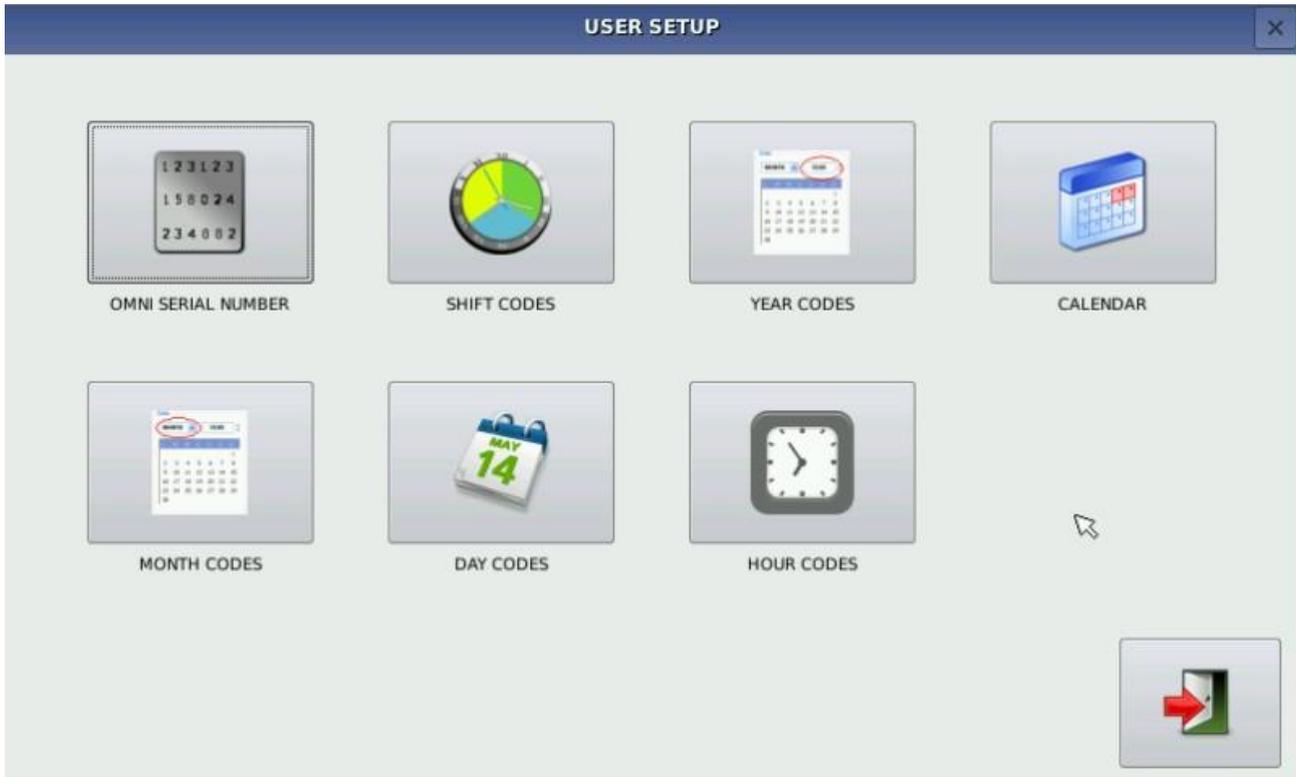
configure the calendar:

- From the auxiliary functionality screen, press the user setup button;



User setup button

- A screen will open showing the possible user parameters that can be changed;



User parameter configuration screen

To change the calendar settings, press the 'calendar' button.



Calendar button

- A new screen will open where you can enter your calendar configuration



Calendar configuration screen

With the buttons on the side, it will be possible to add holidays, remove them or insert them via macros so that, for example, a holiday can be added for all Sundays of the month or year.

4.9 Defining a text query

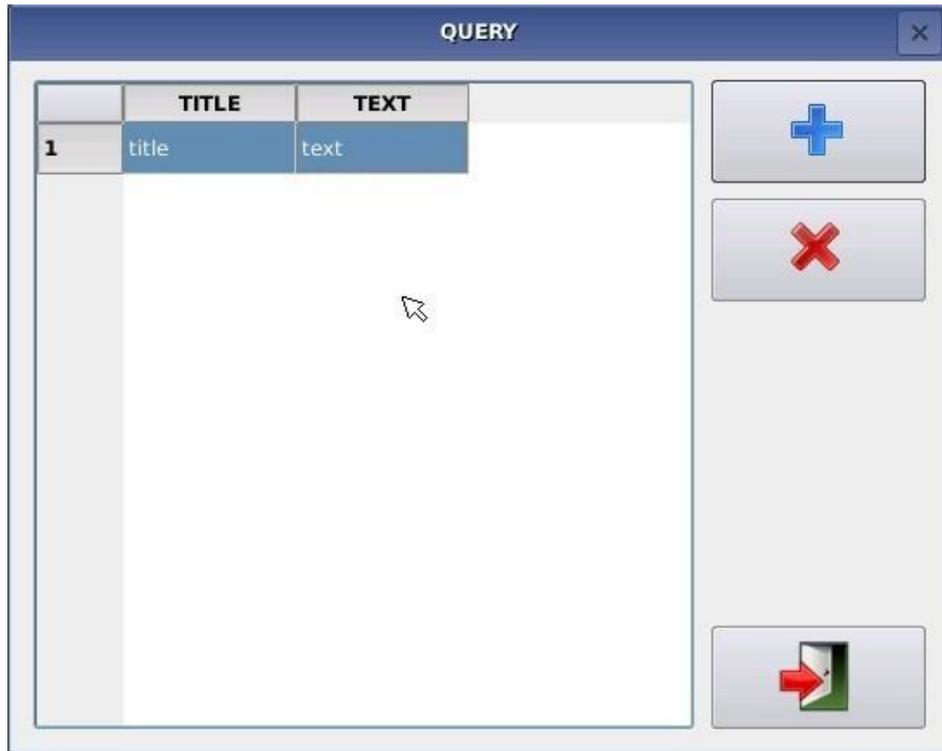
Text query parameters enable the user to operate on several strings of the fields of a programme simultaneously. To activate this possibility, one or more text strings must include one of the following flag commands %Q or %q. The flag command calls up a 'table' of data prepared by the operator. Two values can be set for each implemented text query: a title and a text. The two commands

%Q and %q exist precisely to retrieve one of the two specific query values

%Q title of text query

%q text query

Serial number setting is also possible for *Supervisor* and *Distributor* users. The software will display the query setting table on the screen.



Query setting table

Each row in the table represents a query to which the two values of *title* and *text* are associated. By clicking on one of the two table cells, you will be able to change the corresponding query value. In addition, it is possible to insert a new query with the



button and delete the selected query with the



button.

4.10 Setting General Parameters

The software allows the marking system to be configured according to the mechanical characteristics of the machine. In addition, there are general software configuration parameters that serve to configure the system according to particular requirements or user preferences.

This is why there is a configuration environment in the software for all these parameters that affect the system or the software in general. These include, for example, the size and positioning of the marking area, enabling machine simulation, password management for the *supervisor*, *distributor* and *manufacturer* operating level, and other parameters that will be explained in this section.

In order to be able to configure the general parameters of the system, it is necessary to be as a *distributor* or *manufacturer* user, as the data modification operations you will be carrying out are extremely delicate and require special attention at the time

of the value input phase. Some general parameters can also be set from the *supervisor* operator level.

To configure general parameters:

1. From the auxiliary functionality screen, press the system setup button;



System setup button

2. A screen will open showing the possible system parameters that can be changed;



System parameter configuration screen

To change general parameters, press the general data button.



General data button

3. A new screen will open where you can enter the general parameters of the software

GENERAL PARAMETERS		
	PARAMETER	VALUE
1	X position working area (mm):	0
2	X dimension working area (mm):	90
3	Y position working area (mm):	0
4	Y dimension working area (mm):	50
5	Cycle simulation:	NO
6	Repetition cycle number:	0
7	Start cycle handling:	NOTHING
8	End cycle handlind:	PARKING
9	Setting timeout (s):	20
10	Supervisor password:	0
11	Distributor password:	0
12	Measure unit:	MILLIMETERS

Table for entering general parameters

In this screen you can enter the general parameters of the software. The left column contains parameter descriptions, while the right column contains parameter values. Values can be of two types: numeric or multiple-choice. Numeric values require the entry of a number (which can be set using the numeric keypad), while multiple-choice values require the selection of one of the values proposed in the drop-down menu that appears when the cell with the value is pressed.

4. Once the desired parameters have been set, exit the environment to save the changes made.

Below is a brief description of the general parameters in the table:

	Parameter	Description
1	Position X work area	Indicates the position of the work area in relation to the X-axis. The value entered refers to the absolute minimum height at which the X-axis can be positioned. <i>Numerical value.</i>

2	Dimension X work area	Indicates the size of the work area in relation to the X-axis. <i>Numerical value.</i>
3	Position Y work area	Indicates the position of the working area in relation to the axis of the Y. The value entered refers to the absolute minimum height at which the Y-axis can be positioned. <i>Numerical value.</i>
4	Dimension X work area	Indicates the size of the work area with respect to the Y-axis. <i>Numerical value.</i>
5	Machine simulation	Indicates whether to enable machine simulation management. In this mode, it will be possible to move axes, perform marking cycles and see the behaviour of the software without controlling the machine. It is also possible to run the software in simulation mode by disconnecting the controller from the machine. <i>Choice value (NO - YES).</i>
6	Number of cycle repetitions	Indicates the number of repetitions to be performed in a marking cycle. If the number of repetitions is greater than zero, the same cycle will be repeated for the set number of repetitions at the end of the marking cycle. Data can also be set from the <i>supervisor</i> level. <i>Numeric value.</i>
7	Start-of-cycle management	Indicates the handling to be used at the start of a marking cycle. If set to none, the marking head will proceed with normal cycle execution at the start of the cycle. If set to homing, at the start of the cycle the marking head will first go to the homing position and then start with normal execution of the marking cycle. Data can also be set from <i>supervisor</i> level. <i>Optional value (NONE -HOMING).</i>
8	End-of-cycle management	Indicates the management to be used at the end of a marking cycle. If set to parking only, the marking head will return to the parking position at the end of the cycle. If it is set to parking and homing, at the end of the cycle the marking head will first go to the parking position and then to the homing position. If it is set to homing only, the head will go to the homing position at the end of the cycle. If set to none, the axes will remain in the position corresponding to the end of marking. Data can also be set from the <i>supervisor</i> level. <i>Value of your choice (PARKING - PARKING...HOMING - HOMING - NONE).</i>
9	Calibration timeout	Indicates the timeout time (expressed in seconds) within which

		<p>carry out axis calibration. If the calibration timeout is reached, the programme will display a warning window on the screen.</p> <p><i>Numerical value.</i></p>
10	Supervisor level password	<p>Indicates the password that will be required to access the supervisor operating level. If left at zero, the system will not request the password.</p> <p>This can also be set from the <i>supervisor</i> level.</p> <p><i>Numerical value.</i></p>
11	Distributor level password	<p>Indicates the password that will be required to access the distributor operating level. Only visible to the <i>manufacturer</i> user. If left at zero, the system will not ask for a password.</p> <p><i>Numeric value.</i></p>
12	Units of Measurement	<p>Indicates the unit of measurement that the system will use to define axis coordinates and all other linear measurements.</p> <p>Data can also be set from the <i>supervisor</i> level.</p> <p><i>Value of your choice (Millimetres - Inches).</i></p>
13	Offset datamatrix execution	<p>Indicates the datamatrix execution offset. The offset indicates how far the X-axis must move from the beginning of the datamatrix field.</p> <p><i>Numerical value.</i></p>
14	Enabling logic signals	<p>Indicates whether to enable logic signal handling. See the relevant section in this manual for further information.</p> <p><i>Choice value (NO - YES).</i></p>
15	Signal function SEL_3	<p>Indicates which functionality to associate with the logic signal SEL_3. See the relevant section in this manual for further information.</p> <p><i>Value of your choice (Selection - Online).</i></p>
16	Automatic switch-on	<p>Indicates whether to enable automatic controller start-up at software start-up. If it is disabled, the controller will remain switched off until the power button on the controller faceplate is pressed.</p> <p>Data can also be set from <i>supervisor</i> level.</p> <p><i>Optional value (NO - YES).</i></p>
17	Fieldbus Enabling	<p>Indicates whether Fieldbus protocol management is enabled. This parameter will be disabled if Fieldbus protocol management has not been purchased from the AC500 Store environment.</p> <p>Data can also be set from <i>supervisor</i> level.</p> <p><i>Optional value (NO - YES).</i></p>
18	Work area test speed	<p>Indicates the speed of axes from use in</p>

		test area movements in the test environment. <i>Numeric value.</i>
19	Machine type	Indicates the type of machine connected to the controller. <i>Value of your choice (Cartesian - SCARA - Laser - Cartesian Fast).</i>
20	Inner arm length SCARA	Indicates the length of the inner arm of the SCARA machine. <i>Numeric value.</i>
21	SCARA machine outer arm length	Indicates the length of the outer arm of the SCARA machine. <i>Numerical value.</i>
22	Position X fulcrum SCARA	Indicates the Cartesian position of the fulcrum of the inner arm relative to the working plane. <i>Numerical value.</i>
23	Y fulcrum position SCARA	Indicates the Cartesian position of the fulcrum of the inner arm relative to the working plane. <i>Numerical value.</i>
24	Position X online SCARA	Indicates the Cartesian position at which the marking head must be positioned, in the case of a SCARA machine, online. This position must be within the working area of the machine. <i>Numerical value.</i>
25	Y position online SCARA	Indicates the Cartesian position at which the marking head must be positioned, in the case of a SCARA machine, online. This position must be within the working area of the machine. <i>Numeric value.</i>
26	Maximum LASER frequency value	Indicates the maximum laser frequency value. Only for laser machines. <i>Numeric value.</i>
27	Maximum LASER voltage value	Indicates the maximum laser voltage value. For laser machines only. <i>Numerical value.</i>
28	LASER frequency duty cycle	Indicates the duty cycle value of the laser. For laser machines only. <i>Numeric value.</i>
29	LASER test security timeout	Indicates the laser safety time in the test environment. If the laser is switched on in the test environment, the laser will be forced to switch off once this time has elapsed. Only for Laser machine. Only for Laser machine. <i>Numeric value.</i>
30	LASER stanby voltage value	Indicates the voltage value of the laser in phase. The voltage value is commanded when the cycle is online but not still running. For laser machines only.

		<i>Numerical value.</i>
31	LASER power adjustment time	This indicates the waiting time that the software must observe when marking two fields with different laser voltage values. This time is necessary to allow the laser source to reach the new required power value. <i>Numerical value.</i>
32	Serial enabling	Indicates whether to enable serial port management for external controller control. If enabled, a new button will be displayed in the system parameter screen from where serial communication can be configured. Data can also be set from <i>supervisor</i> level. <i>Optional value (NO - YES).</i>
33	Maintenance password	Indicates the password to carry out maintenance management of the machine. <i>Numerical value.</i>
34	Maximum density in continuous mode	Indicates the maximum point density that can be set in the editor environment in fields with continuous working mode. <i>Numerical value.</i>
35	USB keyboard mapping	Indicates the type of mapping to be used on the USB keyboard connected to the controller. In versions prior to 3.1.0, the special 'Barcode' mapping can be used to enable control of a USB barcode reader. Setting the value to 'Barcode' will enable the button to access the barcode programming environment. From version 3.1.0 onwards, the barcode reader will no longer need to be enabled and the mapping selected in this parameter will be used for readings made by the reader. Data can also be set from <i>supervisor</i> level. <i>Value of your choice (Default - keypad list).</i>
36	Ramp utilisation threshold	Indicates the maximum axis speed at which a movement can be executed without ramps. If the speed of a movement is higher than this parameter, ramp management will be used. <i>Numerical value.</i>
37	Enabling laser filling	Indicates the possibility of enabling the management of filling fields with TTF text in the laser machine. If enabled, it will be possible to set the filling parameters in TTF text fields from the editor environment. <i>Choice value (NO - YES).</i>
38	Third-axis type	This indicates the possibility of selecting the type of use of the third axis managed by the controller. If the parameter is set to COMMUTE, the software will use the third axis, obviously if enabled, with phase switching via relay. Otherwise, if the parameter will be set

		<p>on STE/DIR the software will use the third axis by directly controlling the frequency on the appropriate channel. To use STEP/DIR mode, an FD1 motor with integrated drive and an AC500 board B or higher revision controller is required.</p> <p><i>Value of your choice (COMMUTATED - STEP/DIR).</i></p>
39	Third-axis management	<p>Indicates the possibility of enabling the management of the third axis. The third axis can be vertical (Z axis) or rotational (Theta axis). By enabling the management of the third axis, all software environments will be modified so that this axis can be used.</p> <p><i>Value of your choice (DISABLED - Z- AXIS - THETA AXIS).</i></p>
40	Paired axis at rest (rotary theta)	<p>Indicates the possibility of setting which axis, between Theta and Y, is to be maintained in pairs during a rest situation (end of cycle, pause command, etc.). Depending on the setting of this parameter, switching will be set to keep one axis in pair rather than another. The parameter is only taken into account for the third <i>Theta</i> axis in <i>rotary</i> working mode.</p> <p>If you set the <i>Theta</i>-axis as the axis to be held in torque, the axis calibration logic will also be changed. First the calibration of the <i>XY</i> axes will be commanded and only then the calibration of the <i>Theta</i> axis.</p> <p><i>Value of your choice (Y- AXIS - THETA AXIS).</i></p>
41	External Relay Management	<p>Indicates the possibility of setting the switching output of the third-axis management relay to a dedicated custom output. This allows relay management to be connected outside the controller. The dedicated output for such management is number 4 (corresponding to pin 18 of the I/O connector). The software also allows you to set the parameter to establish the output operating logic: if set to ON/ASSE Y it means that when the digital output is high the Y axis will be commanded (with low output the third axis will be commanded), vice versa if set to ON/THIRD AXIS it means that when the output is high the third axis will be commanded (with low output the Y axis will be commanded). Obviously if you set the value to OFF the normal internal relay management will be used.</p> <p><i>Value of your choice (OFF - ON/Y AXIS - ON/THIRD AXIS).</i></p>
42	Standard piece diameter theta axis	<p>Indicates the default value of the workpiece diameter to be used</p> <p>in the programmes to be marked with Theta axis.</p> <p><i>Numerical value.</i></p>
43	X/Y position machining theta axis	<p>Indicates the position to which the X or Y axis is to be brought when marking with the Theta axis (<i>perpendicular mode</i>).</p> <p><i>Numerical value.</i></p>
44	Axle drive diameter	Indicates the diameter of a possible drive roller

	theta	used in the rotation of the theta axis. <i>Numerical value.</i>
45	Theta axis calibration	If set to CURRENT POSITION, it is possible to disable the search for the calibration FC for the theta axis during control unit online operations. If set to CURRENT POSITION START CYCLE the theta axis will also be reset at each start of a marking job. <i>Value of your choice (FC - CURRENT POSITION - CURRENT POSITION START CYCLE).</i>
46	Axis aligned with rotation axis theta	Allows you to define which of the Cartesian axes of the control unit is aligned with the rotation axis of theta. <i>Value of your choice (AXIS X- AXIS Y).</i>
47	Flat surface tolerance theta axis	Allows you to define the default value of the tolerance in millimetres used when evaluating the feasibility of a marking field without moving theta axis in CHARACTER INDEX mode. <i>Numeric value.</i>
48	Maximum number of markings history files	Indicates the maximum number of files for the marking history that can be saved in the controller's memory. If the parameter is set to 0 (zero), the management of saving historical markings to files will be disabled. <i>Numerical value.</i>
49	Type of saving markings history file	Indicates the mode to be operated in the case of enabling management saving history file. The possible modes are: <ul style="list-style-type: none"> • <i>multiple files</i>: the saving of history files will be done by allowing the saving of markings from the same programme. • <i>Single File</i>: Saving of history files will be done by allowing the saving of a single file per programme. Multiple markings from the same programme will not be saved. • <i>Barcode</i>: saving history files will be done by barcode. It will be possible to use the barcode reader to enter the name of the file to be saved. <i>Value of your choice (MULTIPLE FILE - SINGLE FILE - BARCODE).</i>
50	Advanced Marking History Management	Indicates the possibility of enabling advanced management of the marking history. If enabled, more information will be stored in the marking history for more advanced data management. <i>Choice value (NO - YES).</i>
51	Automatic parking	Indicates the possibility of enabling automatic parking management. If enabled, the marking head will be brought to programme parking altitude after a machine online operation and each time a programme is loaded.

		<i>Choice value (NO - YES).</i>
52	Command mode flag %X	Indicates the management mode of the %X (Date) flag command. The management can be either American (MM/DD/YYYY) or European (DD/MM/YYYY). <i>Value of your choice (AMERICAN - EUROPEAN)</i>
53	Abort management for serial number repetitions	Indicates the handling to be followed when cancelling the marking of a programme with repeated serial numbers. In the case of <i>NO</i> management, the software will not change any values and the number of repetitions will remain as the number calculated from the executions. In the case of <i>AUTOMATIC</i> management, the software will reset the repetition value to the number preceding the start of the cycle. In the case of <i>MANUAL</i> management, the software will display a message asking whether or not to reset the number of repetitions. <i>Value of your choice (NONE - AUTOMATIC - MANUAL)</i>
54	Enabling networked controllers	Indicates the possibility of enabling controller management in a network. If management is enabled, it will be possible to use the controller in communication with a network. Changing this parameter requires a restart of the control unit. <i>Choice value (NO - YES).</i>
55	Offline control unit in case of ABORT	Indicates whether to enable offline management of the control unit at the abort of a cycle. If enabled, the control unit will go offline in the event of a cycle abort. It will then be necessary to bring the machine online again. <i>Choice value (NO - YES).</i>
56	Customised shift code management	Indicates whether customised management of shift codes is enabled. If enabled, it will be possible to configure the advanced settings of shift codes in a customised manner. Using the flag command associated with shift codes in a text field will take the value indicated to the custom configuration. <i>Choice value (NO - YES).</i>
57	Number of Modbus axes FD1	Indicates how many Modbus axes the controller is to control. The maximum number of Modbus axes that can be set is four. If the parameter is set to zero, Modbus motor management will be disabled. FD1 type motors with integrated drive and RS485 serial wiring are required to use these motors. <i>Value of your choice (0 ÷ 4).</i>
58	Tip type	Indicates the tip type to be handled by the software. This parameter will then be used to check whether the selected stock head is compatible with the software settings. In fact, this parameter

		<p>serves to ensure greater safety controls as controlling an airhead with a control unit configured to control an electric drill can cause EV failures.</p> <p><i>This parameter can only be set on AC500 Board B or higher revision control units.</i></p> <p><i>Choice of value (PNEUMATIC - ELECTRIC).</i></p>
59	Electric tip modulation percentage	<p>Indicates the percentage of current modulation used to drive the electric tip. The percentage is applied to the maximum manageable current value on an electric tip.</p> <p><i>This parameter can only be set on AC500 Board B or higher revision control units.</i></p> <p><i>Numerical value (10 ÷ 100).</i></p>
60	Reverse logic enabling of DONE signal	<p>Indicates whether to enable reverse logic operation of the DONE signal. If enabled, it will be possible to invert the operating logic of the OUTPUT signal of DONE. With reverse logic the signal will be as follows:</p> <ul style="list-style-type: none"> - not in marking, end of programme = signal OFF - control unit in marking = signal ON <p><i>Choice value (NO - YES).</i></p>
61	Reverse logic enabling of READY signal	<p>Indicates whether to enable the management of the READY signal with reverse logic. If enabled, it will be possible to invert the operating logic of the OUTPUT signal of READY. With reverse logic the signal will be as follows:</p> <ul style="list-style-type: none"> - control unit OFFLINE, not marking = signal OFF - control unit in marking = signal ON <p><i>Choice value (NO - YES).</i></p>
62	Enabling file uploads from the network	<p>Indicates whether to enable the management of programme loading from a network server. If enabled, it will be possible to load any file from the set network server. To be able to use this management, the corresponding additional functionality must be purchased from the store.</p> <p><i>Choice value (NO - YES).</i></p>
63	Virtual keyboard mapping	<p>Indicates the keyboard mapping to be used in the virtual keyboard displayed for text entries. Selecting the Default mapping will present the software's standard alphanumeric keyboard.</p> <p><i>Value of your choice (Default - ...).</i></p>
64	Cycle Enable Password	<p>Indicates the password to be used to enable a marking cycle from touch. If the parameter is set to zero, no request will be made to the operator.</p> <p>To be able to use this management, it is necessary to purchase the corresponding additional functionality from the store.</p> <p><i>Numerical value.</i></p>
65	Cycle enable password request time	<p>Indicates the expiry time of a cycle enable by password. If the parameter is set to zero and the cycle has been enabled by password, an enable request will never be made until the system is restarted. Conversely, the password will be displayed again at</p>

		<p>expiry of the set time (referring to the last enable made/start of the control unit). To be able to use this management, it is necessary to purchase the corresponding additional functionality from the store.</p> <p><i>Numerical value.</i></p>
66	Enabling extended protocol over Ethernet	<p>Indicates the enabling of extended protocol management on the Ethernet network. In particular, it will be possible to use the extended protocol (see <i>chapter on software communications</i>) on a TCP-IP protocol.</p> <p><i>Choice value (NO - YES)</i></p>
67 (DEL)	Enabling Datamatrix compression	<p>Indicates the possibility of enabling datamatrix format compression. PARAMETER HAS BEEN REMOVED AS OF SOFTWARE VERSION 3.1.0.</p> <p><i>Choice value (NO - YES)</i></p>
67	Resetting serial numbers programmed in the background	<p>Indicates the possibility of enabling the reset associated with a time event of the serial numbers of programmes even for non-running programmes.</p> <p><i>Choice value (NO - YES)</i></p>
68	Enabling text field data control	<p>Indicates the possibility of enabling data validity check management for text fields. To be able to use this management, it is necessary to purchase the corresponding additional functionality from the store.</p> <p><i>Choice value (NO - YES)</i></p>
69	Enabling the label printer	<p>Indicates the type of label printer enablement. It will be possible to select: <i>END OF CYCLE</i> where the print will be executed at each end of marking, <i>REQUEST</i> where the print request will be executed at each end of cycle and it will be possible to make it or not, <i>BARCODE</i> where the print will be retrieved from barcode.</p> <p><i>Value of your choice (END OF CYCLE - REQUEST - BARCODE)</i></p>
70	Enabling non-repeatability check	<p>Indicates the enablement for the repeatability check at the start of marking. This parameter will only be enabled if the function is purchased in the store.</p> <p><i>Choice value (NO - YES)</i></p>
71	Automatic export of marking history	<p>Indicates enabling of the export of the marking history.</p> <p><i>Choice value (NO - YES)</i></p>
72	Command mode flag %T	<p>Indicates the mode which will be used for decoding the %T flag command. With this parameter it will be possible to select whether to mark the time with or without the seconds value.</p> <p><i>Value of your choice (HH:MM - HH:MM:SS)</i></p>

73	Enabling on-screen messages	Indicates the enabling of the display of dialogue messages in the event of particular events or errors that may occur in the software. An example of such possible visualisations is the error of failed programme loading from barcode reader in case the programme
-----------	------------------------------------	--

		required is not present on the control unit. <i>Choice value (NO - YES)</i>
74	Reset executions from operator level	Indicates the enablement of the reset execution counter management of marking programmes from the operator level. This parameter will only be enabled if the relevant function is purchased from the store. <i>Choice value (NO - YES)</i>
75	Flag command increment value %j (Julian)	Indicates the increment value to be used for the special management of the Julian day by flag command. The value used will be used to calculate the increment of the number at each 24 * hour value. To be able to use this management, it is necessary to purchase the corresponding additional functionality from the store. <i>Numeric value.</i>
76	Operator level reset timeout	Indicates the time value expressed in minutes to be used for managing the automatic reset of the operator level. If a value other than zero is set, the software will automatically modify the current level by setting it to base level, i.e. operator level. <i>Numeric value.</i>
77	Disabling cycle options	Indicates the enabling of inhibition management of marking cycle options (simulation, check area, etc.) for operator and supervisor user levels. <i>Choice value (NO - YES)</i>
78	Calendar management flag command %j (julian)	Indicates the enabling of calendar management for the calculation of the special Julian day by flag command. Enabling this parameter will allow access to the calendar configuration environment from the user setup environment. <i>Choice value (NO - YES)</i>
79	Turned logic enable for TTL signal ABORT	Indicates whether to enable ABORT signal management with reverse logic. If enabled, it will be possible to reverse the operating logic of the ABORT INPUT signal. With reverse logic the signal will be as follows: - signal OFF: ABORT command active - signal ON: ABORT command deactivated <i>Choice value (NO - YES).</i>
80	Minimal custom font microline in vibro	When marking text with custom fonts (efc) generated from DXF files, this parameter allows a minimum size to be defined for the microlines into which any arcs present are divided. This parameter has an impact on the execution speed and precision of the marking. Very small values result in high precision but low speed; conversely, larger values result in lower precision and higher speed. of execution.
81	Simplified programming	It allows to enable or not the management of simplified programming. <i>Choice value (NO - YES)</i>
82	Spacing programming	Defines the distance between one field and the next in the

	simplified	simplified programming.
83	Disabling operator level job loading	Enables the JOB loading function to be disabled for the 'user' access level. <i>Choice value (NO - YES).</i>
84	Enabling vector preview	Enables the rendering of the machining preview in vector format. The vector format allows for greater image quality after a ZOOM. <i>Choice value (NO - YES).</i>
85	OUT_4 signal function	Indicates the type of operation of the digital output OUT_4. It will be possible to select: AUTOMATIC (default) or ONLINE. If the output is set as ONLINE, this will be activated to digitally communicate to the outside world when the control unit is ONLINE or not. <i>Value of your choice (AUTOMATION - ONLINE)</i>
86	Compact response to extended protocol	If activated, it allows the response to extended protocol messages to be modified: <ul style="list-style-type: none"> • the BCC is always included in the reply message; • the text received is NOT reflected in the answer. <i>Choice value (NO - YES).</i>
87	Remote Protocol Encoding	Allows you to define the encoding type used for remote protocols. It will be possible to select: DEFAULT, ASCII and UTF8. <i>Value of your choice (DEFAULT - ASCII - UTF8).</i>
88	Operator login management	If enabled, it enables operator login management, which allows access to the control unit with a specific operator code. With this management enabled, milling is enabled ONLY with login. <i>Choice value (NO - YES).</i>
89	Speed check area	This parameter defines the execution speed of the AREA CHECK cycle. If the parameter is set to 0, the speed of the AREA CHECK is defined by the parameter "VEL. PREP" which can be set in the head store. <i>Numerical value.</i>
90	Signal function OUT_5	Indicates the type of operation of the digital output OUT_5. It will be possible to select: AUTOMATIC (default) or ERROR. If the output is set as ERROR, it will be activated to communicate digitally to the outside world when the control unit is in alarm. <i>Value of your choice (AUTOMATIC - ERROR)</i>
91	Error handling empty variable text	Allows you to define how to check an empty variable text field at the start of marking. It will be possible to select: DEFAULT, NO or YES. Below is the operation of the start command given by touch: <ul style="list-style-type: none"> • DEFAULT: message box with the possibility of continue or not with the marking cycle;

		<ul style="list-style-type: none"> • NO: no error message and variable text control bypassed; • SI: message box reporting the blocking error, NO possibility to continue marking. <p>Here is how the remote start command works:</p> <ul style="list-style-type: none"> • DEFAULT: no visual signal but abort of the programme; • NO: no error message and variable text control bypassed; • SI: message box reporting the blocking error, NO possibility to continue marking. <p><i>Value of your choice (DEFAULT - NO - YES)</i></p>
<p>92</p>	<p>Arc text management</p>	<p>Allows you to select the mode of operation and management of the arc text. Below are the two modes of operation:</p> <ul style="list-style-type: none"> • DEFAULT: The text is positioned on the arc by making the bottom left-hand corner of each character of the text tangent to the arc; • CENTRED: the text is positioned on the arc by making the central point of the base tangent to the arc each character of the text. <p><i>Value of your choice (DEFAULT - CENTRAL)</i></p>

4.11 Setting axis parameters

The software allows the axes parameters of the marking system to be configured according to the mechanical characteristics of the machine.

There is a configuration environment in the software for all these parameters concerning the X and Y axes. These include , for example, and other parameters that will be explained in this section.

In order to be able to configure the axis parameters, it is necessary to be a *manufacturer* user, as the data editing operations to be performed are extremely delicate and require special attention when entering values.

To configure axis parameters:

1. From the auxiliary functionality screen, press on the system setup button;



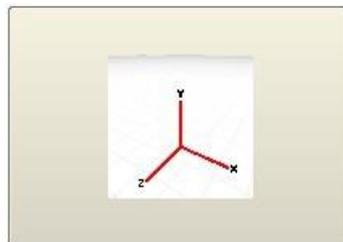
System setup button

2. A screen will open showing the possible system parameters that can be changed;



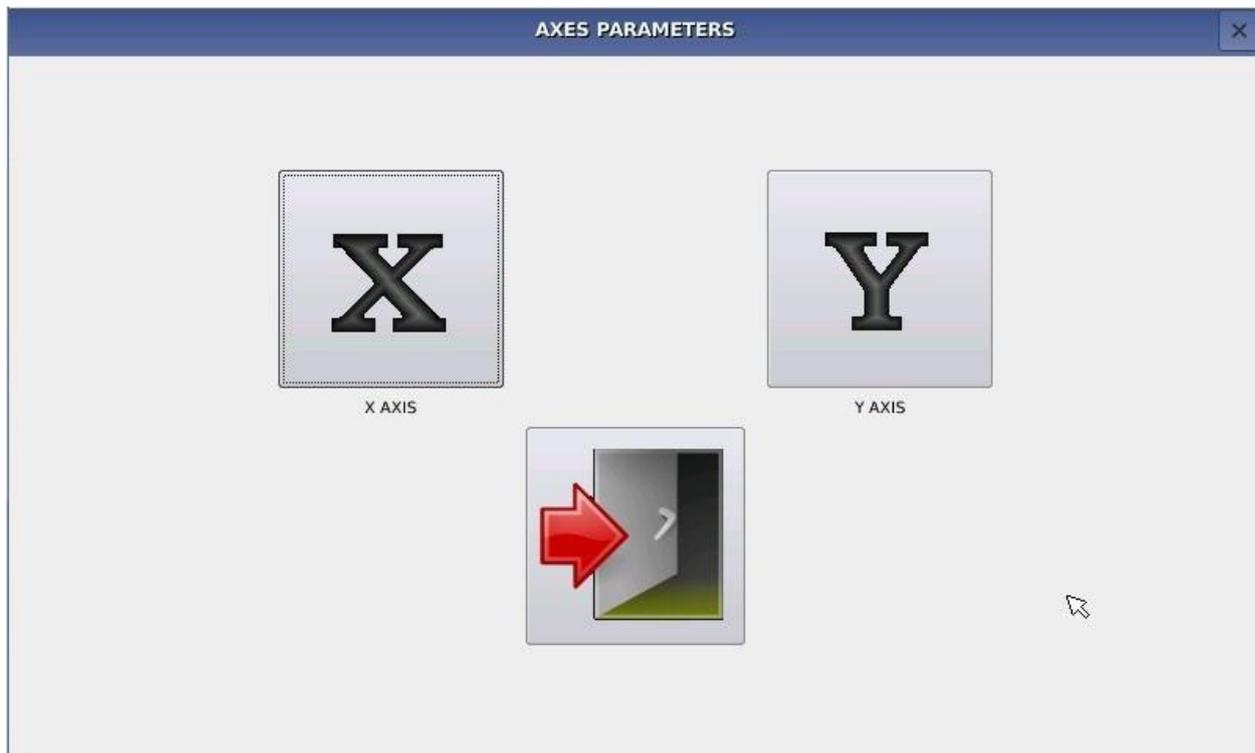
System parameter configuration screen

To change axis parameters, press the axis data button.



Axis data button

3. The software will display a screen where you can select which axis to configure.



Selection screen of the axis to be configured

To select the axis to be configured, press on one of the buttons relating to the axes controlled by the software.

4. A new screen will open where you can enter the axis parameters of the software

	PARAMETER	VALUE
1	Encoder pulses:	400
2	Quote / round encoder (mm):	5.6
3	Maximum quote (mm):	90
4	Minimum quote (mm):	0
5	Maximum speed (mm/s):	200
6	Microswitch speed (mm/s):	30
7	Acceleration time (s):	0
8	Deceleration time (s):	0
9	Mx. Frequency (HZ):	14285
10	Setting quote (mm):	0
11	Setting direction:	MINUS
12	Setting microswitch channel:	13

Table for entering axis parameters

5. Once the desired parameters have been set, exit the environment to save the changes made.

Below is a brief description of the axis parameters in the table:

	Parameter	Description
1	Number of encoder pulses	Indicates the number of encoder pulses. Data supplied by the encoder manufacturer. <i>Numeric value.</i>
2	Encoder ride share	Indicates the encoder revolution quota. The encoder revolution height is the space travelled in one encoder revolution. <i>Numeric value.</i>
3	Maximum quota	Indicates the maximum arrival altitude of the selected axis. <i>Numeric value.</i>
4	Minimum quota	Indicates the minimum arrival altitude of the selected axis. <i>Numeric value.</i>
5	Maximum speed	Indicates the maximum speed at which the axis can move. <i>Numeric value.</i>
6	Calibration FC search speed	Indicates the speed of the axis being calibrated. During calibration, the axis moves in search of the calibration FC (End of Travel). <i>Numeric value.</i>
7	Acceleration time	Indicates the time in <i>seconds</i> to be used in the axis acceleration ramps. More precisely, the parameter indicates how long it takes to reach the set maximum speed (parameter 5) starting from standstill. <i>Numeric value.</i>
8	Deceleration time	This indicates the time in <i>seconds</i> to be used in the axis deceleration ramps. More precisely, the parameter indicates how long it takes the axis to stop from the set maximum speed (parameter 5). <i>Numerical value.</i>
9	Maximum frequency	Indicates the maximum frequency at which the axis drive can go. Value expressed in Hz. The value cannot be set as it is calculated from the parameters <i>number of encoder pulses</i> , <i>encoder revolution rate</i> and <i>maximum speed</i> . <i>Numeric value.</i>
10	Calibration quota	Indicates the height at which the axis is calibrated. <i>Numerical value.</i>
11	Calibration direction	Indicates the calibration direction of the axis. This value depends on the direction of movement of the axis. FORWARD is towards large dimensions while REVERSE is towards small dimensions. <i>Value of your choice (BACK - FORWARD).</i>

12	Channel FC calibration	Indicates the number of the digital input set up for the axis calibration FC. <i>Numeric value.</i>
13	Q.output FC calibration	Indicates the displacement dimension of the axis if it is already on the limit switch during calibration. <i>Numerical value.</i>
14	Direction reversal	Indicates the direction of movement of the axis. <i>Choice value (NO - YES).</i>
15	Motor current level	Indicates the current level of the motors. In particular, the motor current level is selectable between: <ul style="list-style-type: none"> • level 0 (0.7A) • level 1 (0.9A) • level 2 (1.4A) • level 3 (1.7A) <i>Value of your choice (LEVEL 0 - LEVEL 1 - LEVEL 2 - LEVEL 3).</i>

NOTE

When setting the axis movement direction, it is important to also change the calibration direction and calibration altitude parameters.

4.12 Machine maintenance setting

From the parameter configuration environment, it is possible to set the data required for a machine maintenance request. In the machine maintenance data setting it is possible to enter checks to request machine maintenance after a certain amount of time the machine has been in use. When the software starts up, these checks will be performed and if one of them is checked, a screen showing the maintenance to be performed will be displayed. It is also possible to associate a maintenance with an image describing the maintenance in question.

From the screen displayed, it is possible to give the maintenance enabled.

In order to be able to change the maintenance data of the controller, it is necessary to be a user with distributor or manufacturer operating level.

To set recommended maintenance parameters:

1. From the auxiliary functionality screen, press the system setup button;



System setup button

2. A screen will open showing the possible system parameters that can be changed;



System parameter configuration screen

To change maintenance parameters, press the maintenance button.



Maintenance data button

3. In the maintenance environment, the maintenance counters (by number of marking strokes and by hours of controller use) and the table of managed maintenance on the controller are visible.

General

Hits counter

Counter:

Utilization hours

Counter:

Hours reference: ▼

Maintenance counter data

- The data of the maintenance counters are the absolute number of marked shots and the number of hours the controller has been in use. Both counters can be reset to zero at any time. In addition, the hour counter can be set to either the total hours of use of the controller or the hours of marking of the controller.

Machine recommended

	Description	Hits number	Hits counter	Hours number	Hours counter	Date	Enable	Image
--	-------------	-------------	--------------	--------------	---------------	------	--------	-------

Recommended maintenance data

- Recommended maintenance data can be entered in the table

4.13 Head stock setting

From the parameter configuration environment, it is possible to customise the marking head magazine. The head magazine serves to manage marking head-related parameters and offers the possibility of entering parameters for three different marking modes: *high quality*, *normal* and *fast*. These modes are used to set the marking quality of a program field. For each of these modes, it is possible to define marking speeds and times, which are used to configure the execution of the various programme fields as desired.

In order to be able to insert, modify or delete a head from the warehouse, it is necessary to be a user with a manufacturer operating level.

To configure the head store parameters:

1. From the auxiliary functionality screen, press the system setup button;



System setup button

2. A screen will open showing the possible system parameters that can be changed;



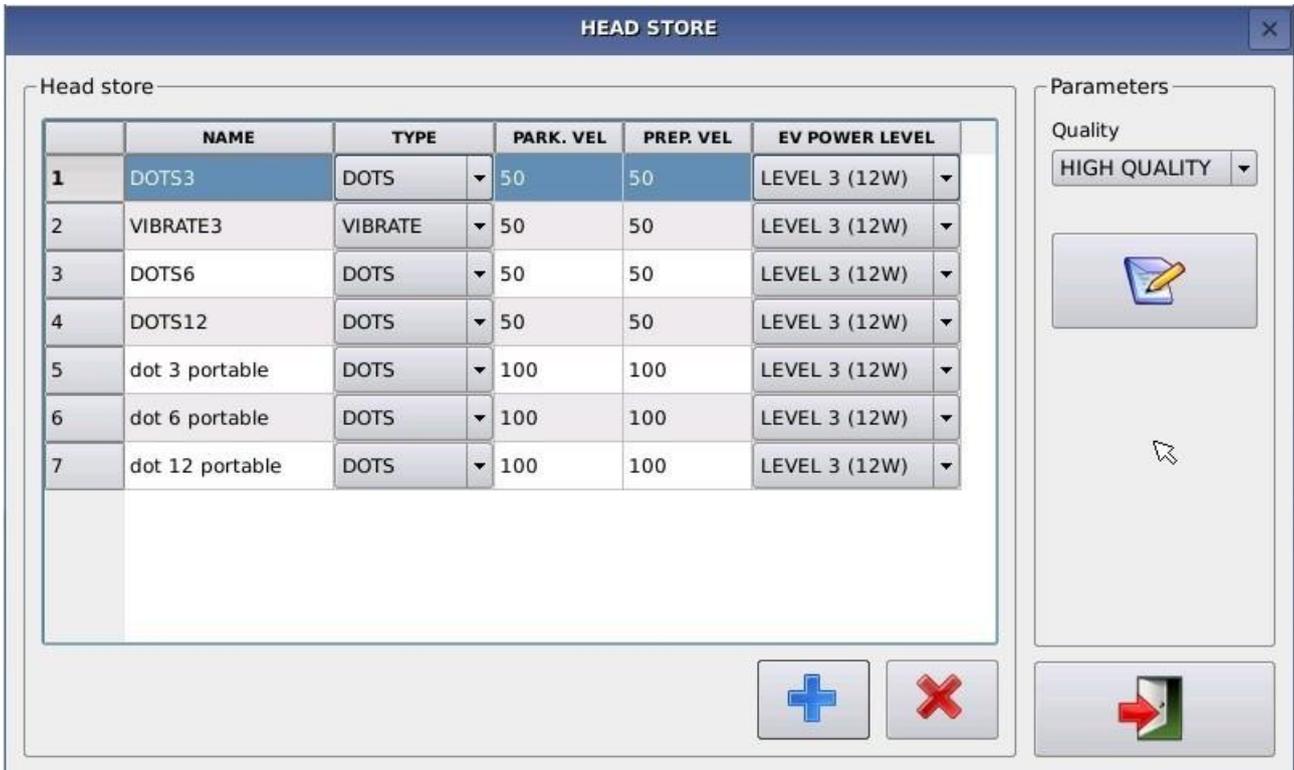
System parameter configuration screen

To change the head store parameters, press the head store button.



Head warehouse data button

3. Pressing the *head store* button will open the following window:



Head store setting management screen

The table on the left of the screen shows the magazine with the marking heads currently available on the AC500 controller.

It is possible to insert  or remove  a marking head from the buttons to the right of the screen.

The marking head has three mode-independent parameters: the name that describes the head and under which it will be displayed when selecting the head and running the programme, the parking speed, the pre-positioning speed, the current level of the solenoid valve and the type of marking to be associated with the head. All these values can be entered directly from the table.

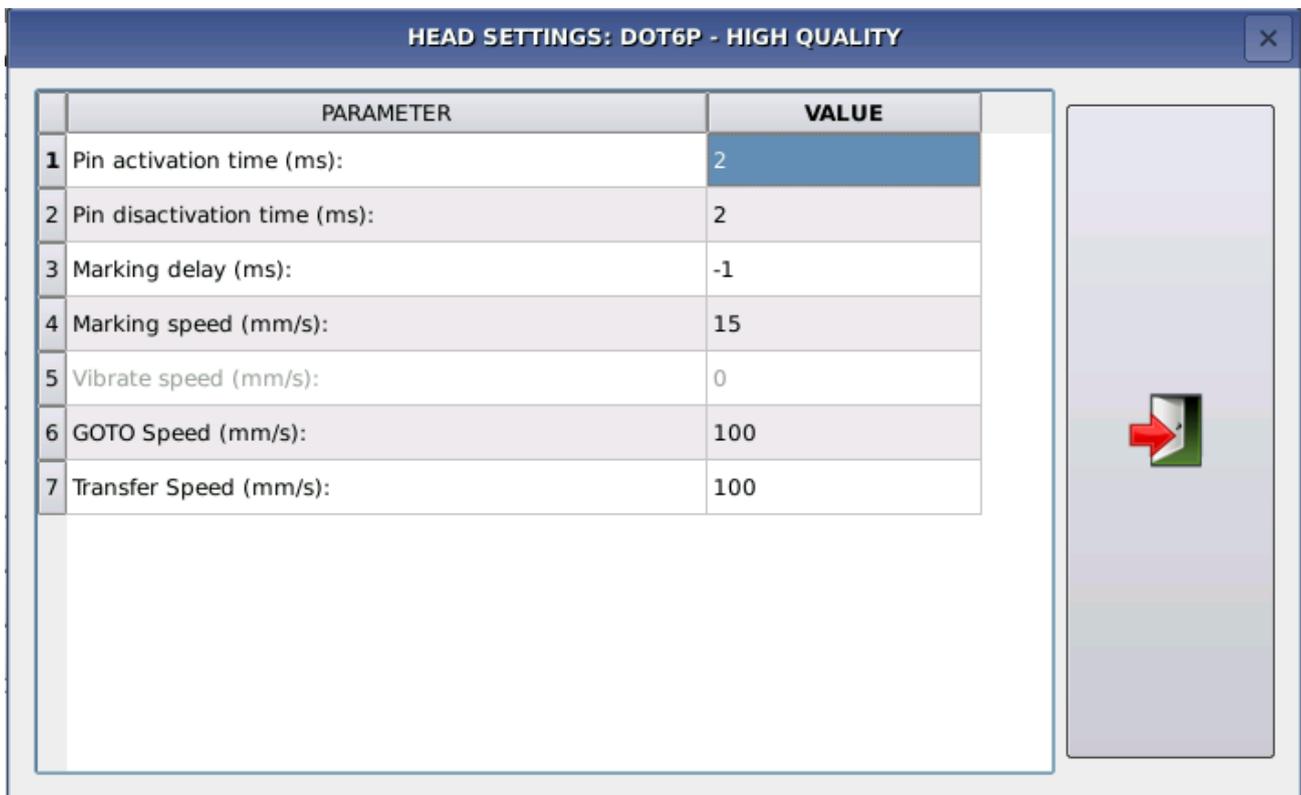
NOTE

Setting the type of marking to be associated with the head will affect the editing of the programme, as the mode of execution of each field will depend on the currently selected head.

For each marking head it is possible to enter the machining parameters for three different modes: HIGH QUALITY, NORMAL and QUICK.



To enter the parameters of a mode, press the edit key . Pressing this key will take you to the head parameter screen.



	PARAMETER	VALUE
1	Pin activation time (ms):	2
2	Pin disactivation time (ms):	2
3	Marking delay (ms):	-1
4	Marking speed (mm/s):	15
5	Vibrate speed (mm/s):	0
6	GOTO Speed (mm/s):	100
7	Transfer Speed (mm/s):	100

Head parameter setting screen

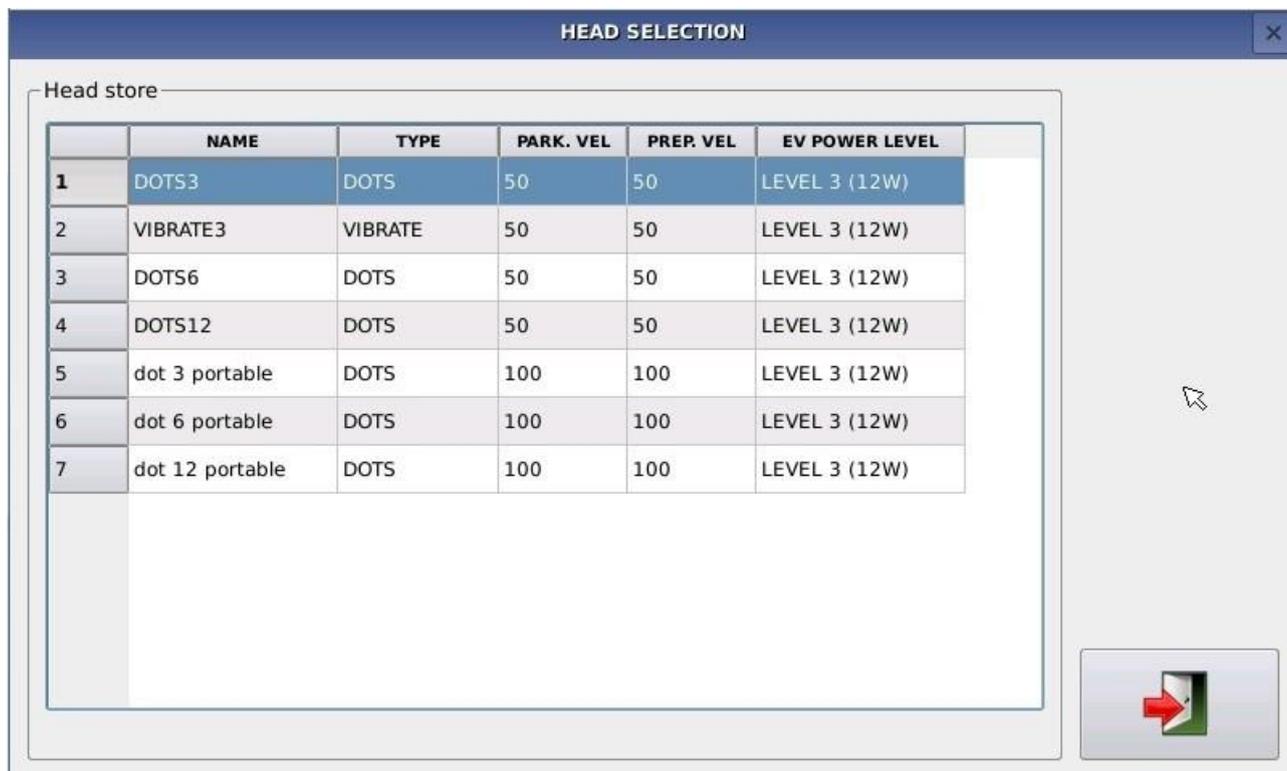
It will also be possible to select which machining head to work with.

For users with *supervisor* and *distributor* operating levels, it will be possible to select the machining head simply by entering the *system setup* environment of the auxiliary functions menu.

With *manufacturer* operating level, it will be necessary to press the button next to the *selected head* item in the system parameter menu.



The window that appears is as follows:



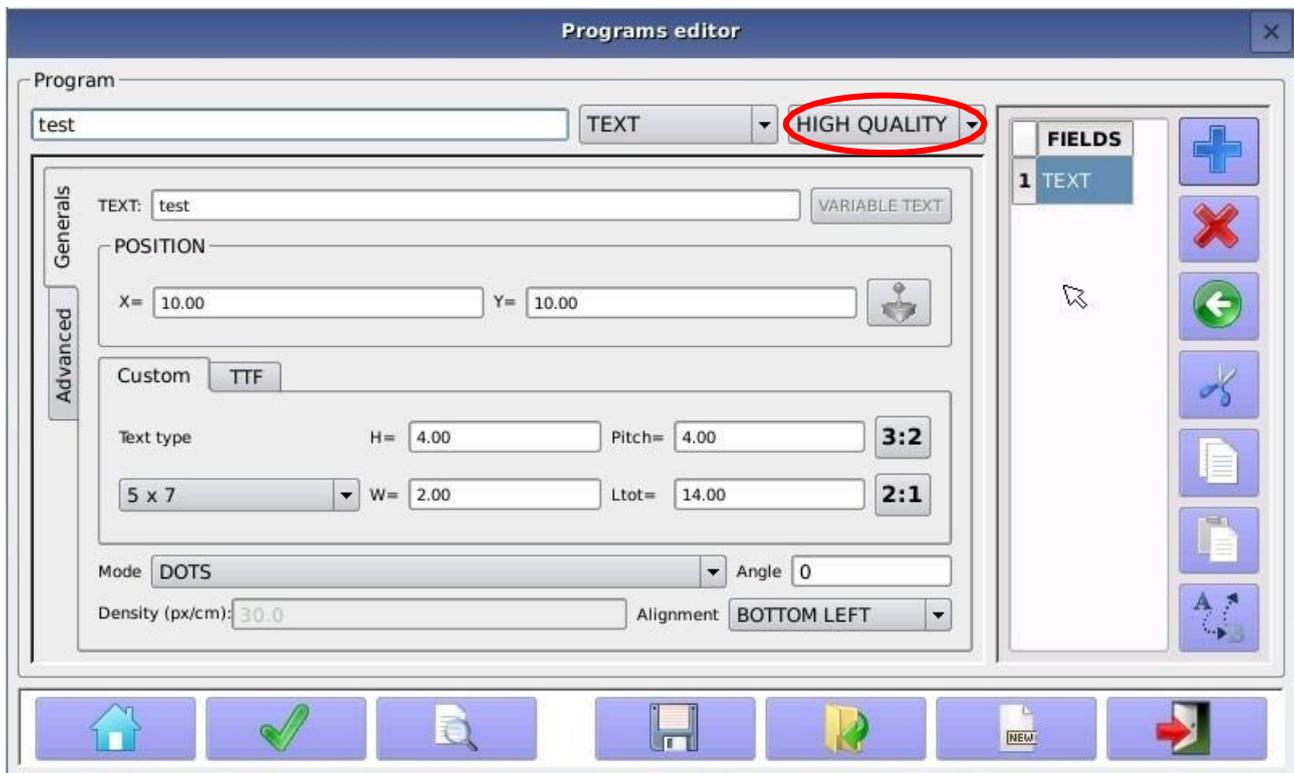
Marking head selection screen

The table will show all the marking heads in the magazine and you only have to select the head to be used and confirm the change when exiting the window.

Finally, when editing the programme, it will be possible to select for each field the mode in which the following marking will be performed. To choose the mode, simply select the item in the drop-down menu next to the field type.

Parameter	Description
Current EV level	<p>Indicates the current level of the solenoid valve. In particular, the current level for solenoid valves, which are current-controlled (with 48V alim) but differ in power level, is selectable between:</p> <p>AC500 - Board A:</p> <ul style="list-style-type: none"> • level 0 (4W) • level 1 (5W) • level 2 (6W) • level 3 (12W) <p>AC500 - Board B or higher revision:</p> <ul style="list-style-type: none"> • PNEUMATICS (5W) • ELECTRIC (12W) <p><i>Choice value:</i> <i>Board A (LEVEL 0 - LEVEL 1 - LEVEL 2 - LEVEL 3)</i></p>

Board B or higher (PNEUMATIC - ELECTRICAL)



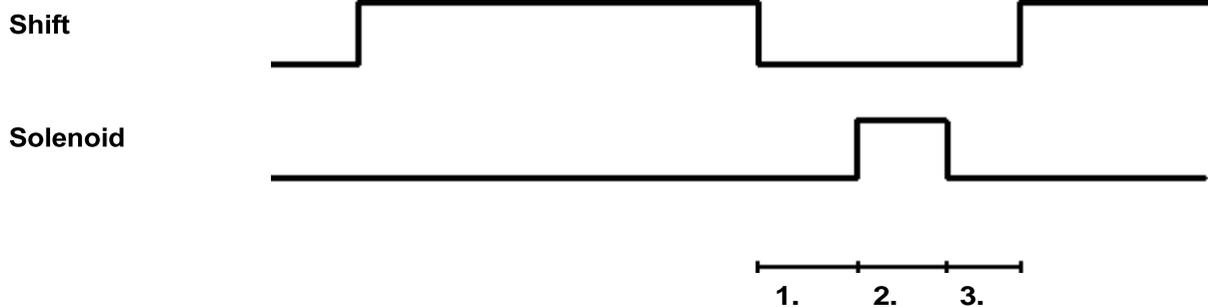
Screenshot of the programme editor with the marking mode/quality field highlighted

Below is a brief description of the marking head parameters in the head magazine table:

	Parameter	Description
1	Head descent time	Indicate the time of activation of the output of descent output of the solenoid valve. Expressed in milliseconds. <i>Numeric value.</i>
2	Head rise time	Indicates the time that elapses between the deactivation of the descent output of the solenoid valve following a marking and the start of axis movement for the execution of the next marking. Expressed in milliseconds. <i>Numeric value.</i>
3	Marking delay time	This indicates the time that elapses between the end of axis movement for the execution of a marking and the instant of activation of the EV descent output. This value can also assume negative values, allowing the activation instant to be anticipated with respect to arrival in position. In the case of advance marking, i.e. a negative value, an absolute value greater than the head descent time cannot be set. Expressed in milliseconds. ATTENTION: Marking advance management can ONLY be set for FAST CARTESIAN machines.

		<i>Numeric value.</i>
4	Pin recovery time	Indicates the minimum time between two consecutive markings. It intervenes mainly in continuous mode in general where a very high marking density is required. <i>This parameter can only be set on AC500 Board B or higher revision control units.</i> <i>Numerical value.</i>
5	Marking speed	Indicates the marking speed at which the marking head will move during marking. <i>Numerical value.</i>
6	Vibration speed / smear	Indicates the marking speed in vibration (or smear) mode at which the marking head will move during marking. <i>Numerical value.</i>
7	GOTO speed	Indicates the speed of movement when a Goto field is running in the programme. <i>Numeric value.</i>
8	Transfer speed	Indicates the speed of movement of the marking head when transferring between one field and another during execution. <i>Numerical value.</i>
9	Multi-point ascent time	Indicates the time that elapses between the deactivation of the descent output of the solenoid valve following a marking and the next marking in the case of multipoint marking. Expressed in milliseconds. <i>Numeric value.</i>
10	Point descent time in vibration	Indicates the descent time of a point mark on a vibro head. Expressed in milliseconds. <i>Numeric value.</i>

The head down time, head up time and marking delay time operate as depicted in the following diagram:



1. Marking delay time;
2. Marking head descent time;
3. Climbing time marking head.

4.14 Setting the system language

The software allows the language of the operating system to be configured. The language can be configured by selecting one of the languages on the software. Once the language has been selected, all messages in the software will be changed to the new selected language. *The manufacturer* user has the option of inserting, deleting or changing the software languages. The *distributor* user, on the other hand, can enable or disable the languages in the software.

To be able to configure the system language, you must be a *supervisor*, *distributor* or *manufacturer* user.

To configure the system language:

1. From the auxiliary functionality screen, press the system setup button;



System setup button

2. A screen will open showing the possible system parameters that can be changed;



System parameter configuration screen

To change the system language, press the language configuration button.



Language configuration button

3. The software will display an on-screen screen where you can select which language to configure



Language selection screen

From this screen you can select the language to be used in the software.

4. Once the desired language has been selected, exit the environment to save the changes made.

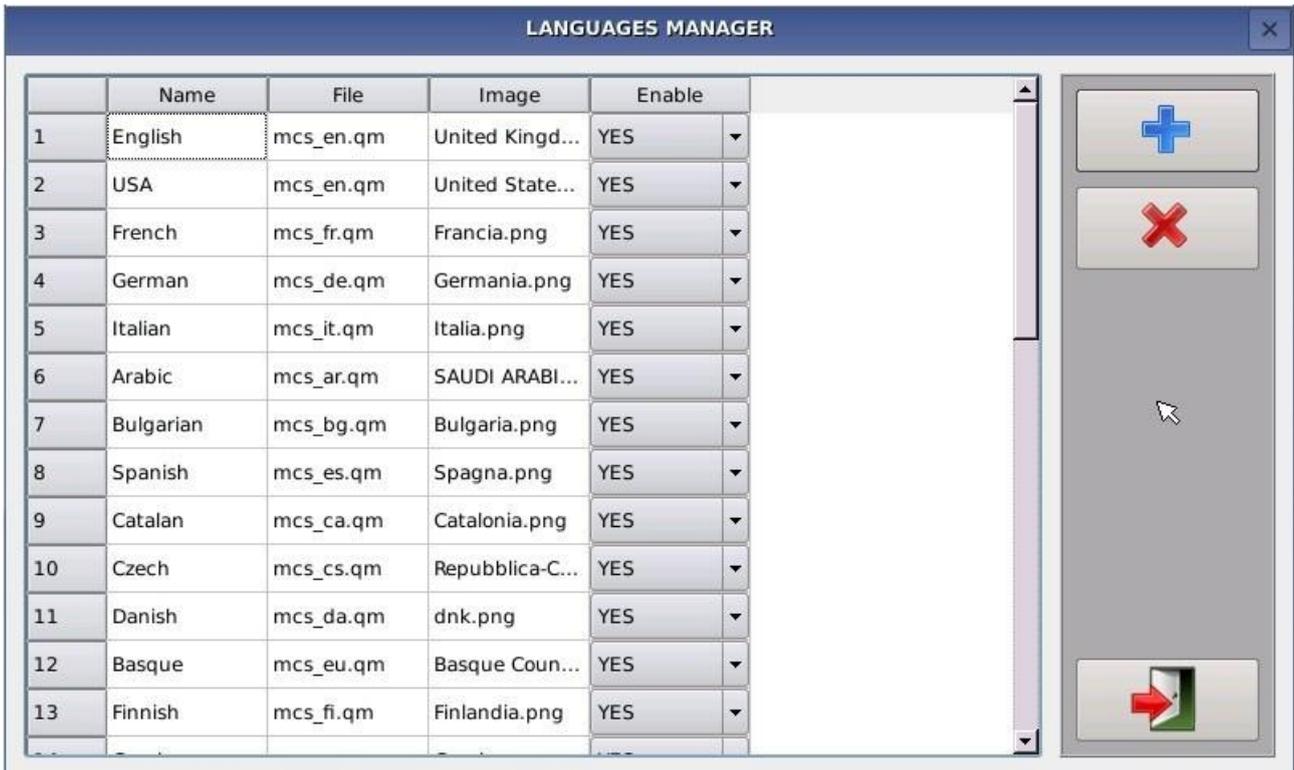
The manufacturer user also has the option of managing the software's languages in order to customise the messages and languages used by the software.

To customise language management, it is necessary to enter the language management environment via the button below:



Language management button

In the language management environment, the languages managed by the software can be inserted, deleted or modified. The environment presents itself in this way.



Language Management Screen



With the button  you can add a new language, and with the button  you can delete the language selected in the table.

The table shows all languages currently present on the controller. For each language, various information is given concerning the *name* of the language, the associated translation *file*, the *image* to be displayed on the button in the language configuration environment, and the language *enabling*.

Clicking on the cell relating to the translation *file* or the cell relating to the *image* to be associated will open the file management screen from which it will be possible to load the appropriate files from the controller's memory. From the file management screen, it will also be possible to load files from a USB key.

The *distributor* user can access this environment to choose the enabling status of each language. If a language is not enabled, it will not be selected in the language configuration environment.

4.15 Configuration of software graphics

In the software configuration environment, it is possible to customise certain graphical components of the software, such as the display of the splashscreen at software start-up.

To be able to configure the graphics components of the system, you must be as user *manufacturer*.

In order to configure the graphic settings of the software, it is necessary to:

1. From the auxiliary functionality screen, press the system setup button;



System setup button

2. A screen will open showing the possible system parameters that can be changed;



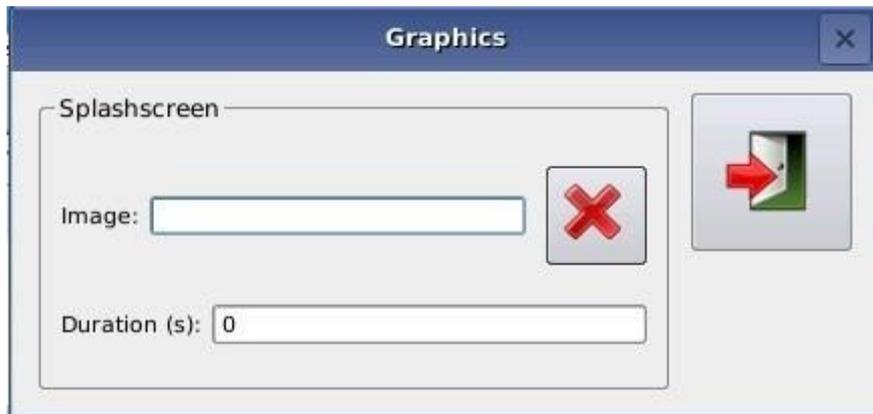
System parameter configuration screen

To change the graphics configuration of the software, press the graphics configuration button.



Graphic configuration button

5. The software will display a screen on the screen where you can set the graphics configuration parameters



Graphic selection screen

From this screen, it is possible to set an image to be displayed as a splashscreen at software start-up. It is also possible to set a time for the splash screen to be displayed. If this time is not set, the splash screen will be displayed for 3 seconds by default.

Once the software graphics configuration parameters have been set, exit the environment to save the changes made.

4.16 Serial Configuration

The software allows the serial port of the controller to be configured to communicate with a host computer. Configuring the serial port is necessary in order to successfully communicate with a host computer.

NOTE

Serial configuration is only available if serial management is enabled in the general software parameters. Otherwise, the serial configuration button will not be enabled in the system setting environment.

To be able to configure the system serial, you must be a *supervisor, distributor* or *manufacturer* user.

To configure the serial:

1. From the auxiliary functionality screen, press on the system setup button;



System setup button

2. A screen will open showing the possible system parameters that can be changed;



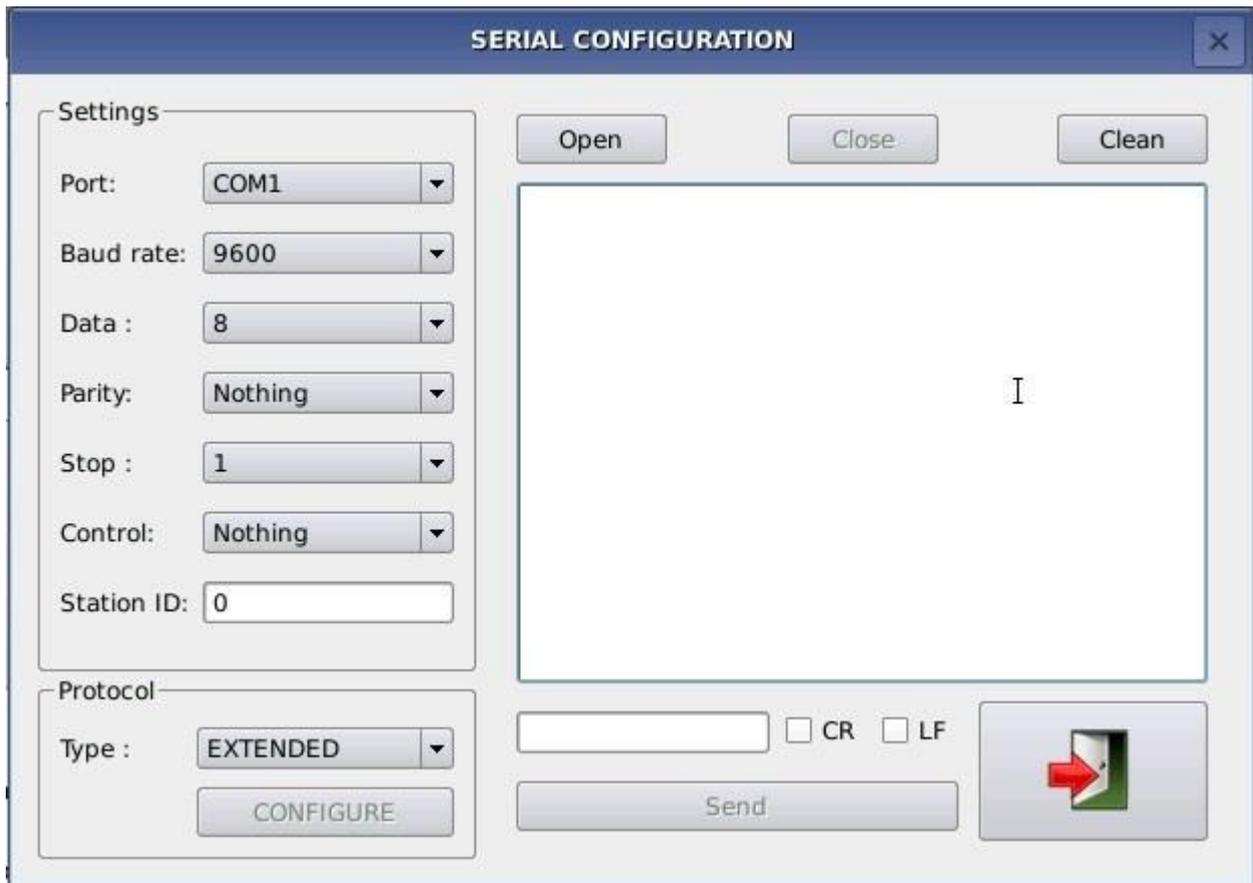
System parameter configuration screen

For modify the configuration of the serial, press on button serial configuration button.



Serial configuration button

3. The software will display a screen on the screen where you can set the serial data



Serial parameter configuration screen

In this screen, it will be possible to configure the serial and to test sending and receiving data with the settings set.

Door Parameter defining the communication port to be used. By default, the parameter is set to COM1.

Baud rate Parameter defining the baud rate to be used in communication. The baud rate indicates the data transmission and reception speed (data bits per second) on the serial. By default, the parameter is set to 9600.

Date Parameter defining the number of data bits to be managed on the serial. The value must coincide with the number of data bits set on the host computer.

- Parity** Parameter defining the value of the serial parity bit. This value is used to check for possible data transmission problems. The value must coincide with the parity bit set on the host computer.

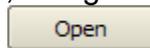
- Stop** Parameter defining the number of stop bits to be managed on the serial. The value must coincide with the stop bits set on the host computer.

- Control** Parameter defining the type of control to be managed on the serial. The value must coincide with the control value set on the host computer.

- Station ID** Parameter defining the number of the station with which to initiate serial communication.

To test the sending of the serial configuration, it is necessary:

- a. open the communication door, using the door opening button

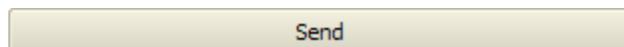


- b. enter a string command inside the header box at the bottom of the serial test page

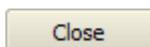
A rectangular text input field with a light beige background and a thin border, currently empty.

- c. (optional) enable special carriage return characters CR or end of line LF

- d. press the serial send button

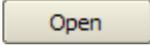


- e. close the communication port, using the appropriate button



To test the reception of the serial configuration, it is necessary:

- a. open the communication door, using the door opening button

Open

- b. if something is received from the serial, the data received will be displayed in the appropriate text area.
- c. *(optional) it is possible to delete the text contained in the text area by pressing the wipe button*

Clean

- d. close the communication port, using the appropriate button

Close

If neither receiving nor sending the serial does not work, check the data set in the configuration or check the cable connection between controller and host

- 4. Once the configuration is set, exit the environment to save the changes made.

It is also possible to select the type of protocol used in serial communication. The protocols that can be selected are: *extended protocol*, *programmable protocol* and *modbus protocol* (which can be enabled after purchasing the function from AC500 Store, see the appropriate section in the manual). For further information on the use and configuration of these protocols, consult the appropriate section in the manual.

NOTE

The setting of serial parameters can be inhibited by the software if Modbus axis control (occupying port 485) and Fieldbus protocol management (occupying port 232) are enabled.

4.17 USB barcode configuration

The software allows the configuration of a possible barcode reader connected to the controller via the USB port. The configuration of the barcode reader is necessary to customise the functionality you want to associate with the data read with the barcode. These settings only apply to USB barcodes.

NOTE

In software versions prior to 3.1.0, USB barcode configuration is only available if barcode management is enabled from the general software parameters. Otherwise, the barcode configuration button will not be enabled in the system setting environment.

To be able to configure the USB barcode, you must be a *supervisor, distributor* or *manufacturer* user.

To configure the barcode:

1. From the auxiliary functionality screen, press the system setup button;



System setup button

2. A screen will open showing the possible system parameters that can be changed;



System parameter configuration screen

To change the barcode configuration, press the barcode configuration button.

**Barcode configuration button**

3. The software will display a screen on the screen where you can set the barcode data

BARCODE CONFIGURATION

Programmable protocol

Start char: Terminator: or

Position: Length:

Ignore: Msg type:

Msg edit:

Enable field ID

Program saving before loading

Enable checksum to marking start

Reset barcode's fields at program ending

Reset barcode's status at program start

Auto NEXT Circular handling

Selection after program loading:

Buttons:

Barcode reader parameter configuration screen

In this screen you can configure the protocol used by the system to analyse the data received from the barcode reader. See the appropriate section of the manual for further information.

In addition to the protocol setting parameters, it is possible to enable or disable certain functions related to the use of barcodes.

- **Enable Field ID:** this parameter is used to indicate whether there is an ID within the message received from the barcode to identify a particular field in the programme or a specific query. If this parameter is enabled, the system will expect a message of the type:

Tnn<string>

where:

T = Indicates the type of message.

nn = two-digit number indicating the number of the field or query table where the data is to be entered. In the case of message type P (selection of a programme) this number is not used.

<string> = programme name (message type *Pattern*) or field data (message type *Update*, *Overwrite*, *Query*, *Append* or *Next*).

For further details on the fields, please refer to the appropriate section in the manual.

NOTE

Enabling this parameter only makes sense if message type 0 (zero) is used.

- **Saving of program on loading:** this parameter is used to indicate whether or not the program present in memory will be saved when loading a program from barcode. If this parameter is enabled, the currently loaded programme will be saved before loading the new programme requested by barcode, otherwise, if the parameter is disabled, the programme will not be saved and the programme requested by barcode will be loaded directly. This means that changes made and not saved to the programme replaced by the one requested by barcode may be lost.
- **Enable checksum check for start marking:** this parameter is used to determine whether the checksum check message must be received from barcode in order to start a marking cycle. If enabled, the programme loaded from barcode will not be executable until the checksum of the programme fields has been launched. If this is successful, it will be possible to proceed with the marking cycle. If disabled, the programme can be executed normally.
- **Clearing edited fields at the end of marking:** this parameter is used to determine whether or not to clear the fields edited by the barcode reader at the end of the execution of a programme. With this management enabled, it is possible to automatically reset a programme that is to be executed with fields read by the barcode reader.
- **Reset barcode handling at start of marking:** this parameter is used to determine whether to reset the barcode handling module at the start of marking to ensure that it starts from a clean and exact condition.

The reset performs, in detail: deletion of any characters present besides the terminator, reset to the first field of the index programme used by the 'Auto NEXT' function described later.

- **Auto NEXT:** this parameter enables the management of a NEXT command (i.e. movement between the text fields of the programme) in automatic mode after reading a Barcode of type OVERWRITE (Type 1) or type APPEND (Type A). Through this function, it will therefore be possible to read a Barcode to edit a text field and ensure that it is moved to the next one to edit it by means of another reading. This new management will have the advantage of handling the NEXT command without having a dedicated barcode to read.
- **Auto NEXT Circular Handling:** if enabled, automatically returns to the first field in the programme when the number of barcode reads equals the number of fields in the programme. If disabled, however, readings after the number of programme fields will be ignored.

4.18 Configuring Network Settings

The software allows you to set the network configurations of the control unit if you want to connect it via Ethernet to a LAN network or if you want to access the control unit remotely. In the network settings environment, you can configure the controller's IP address, subnet mask and gateway address. Furthermore, if controller network management is enabled, it is possible to configure the controller as *MASTER* or *SLAVE*.

To configure network settings, you must be as a *supervisor*, *distributor* or *manufacturer* user.

To configure network settings:

1. From the auxiliary functionality screen, press the system setup button;



System setup button

2. A screen will open showing the possible system parameters that can be changed;



System parameter configuration screen

To change the configuration of the network settings, press the network settings configuration button.



Network settings configuration button

3. The software will display a screen on the screen where you can set the network settings data

NETWORK CONFIGURATION					
IP address:	192	168	0	138	<input type="checkbox"/>  <input type="checkbox"/> 
Subnet mask:	255	255	255	0	
Gateway:	192	168	0	254	

Network settings configuration screen

4. To change any value, simply click on the appropriate text boxes and enter the new data. When a datum is modified, it is shown on the screen in red.

NETWORK CONFIGURATION					
IP address:	192	168	0	100	<input type="checkbox"/>  <input type="checkbox"/> 
Subnet mask:	255	255	255	0	
Gateway:	192	168	0	254	

Example of data modification

5. To change the data with the new settings entered, it is necessary to press the confirm button. After pressing the button, an information screen will appear asking whether to use the changed settings or not. Confirm the request if you want to change the settings.

If the controller network management parameter is enabled (see description of *general data* in the relevant section of the manual), an additional button will be displayed, allowing the relevant settings to be changed.

The screenshot shows a dialog box titled "NETWORK CONFIGURATION" with a close button (X) in the top right corner. The main area contains three rows of input fields:

- IP address:** Four input boxes containing the values 192, 168, 0, and 138.
- Subnet mask:** Four input boxes containing the values 255, 255, 255, and 0.
- Gateway:** Four input boxes containing the values 192, 168, 0, and 254.

On the right side of the dialog, there are three buttons: a green checkmark button, a red X button, and a button with a network diagram icon (a square connected to two smaller squares).

Network settings configuration screen

The controller network management settings serve to identify the controller's behaviour in the network. Above all, it is necessary to specify the role of the controller (MASTER or SLAVE) and the required settings. Further information on how controller network management works can be found in the appropriate section of the '*Controller Network Management*' manual.

MASTER control unit configuration

To configure a control unit as *master*, the type parameter must be set to MASTER and the port that will be used to receive slave connections. It is also possible to set a name to identify the control unit in the network.

The screenshot shows a dialog box titled "NETWORK CONFIGURATION" with a close button (X) in the top right corner. The main area contains three rows of input fields:

- Type:** A dropdown menu with "MASTER" selected.
- Name:** A text input box containing "AC500-1".
- Port:** A text input box containing "3750".

At the bottom center of the dialog, there is a button with a network diagram icon (a square connected to two smaller squares). On the right side, there are three buttons: a green checkmark button, a red X button, and a black left-pointing arrow button.

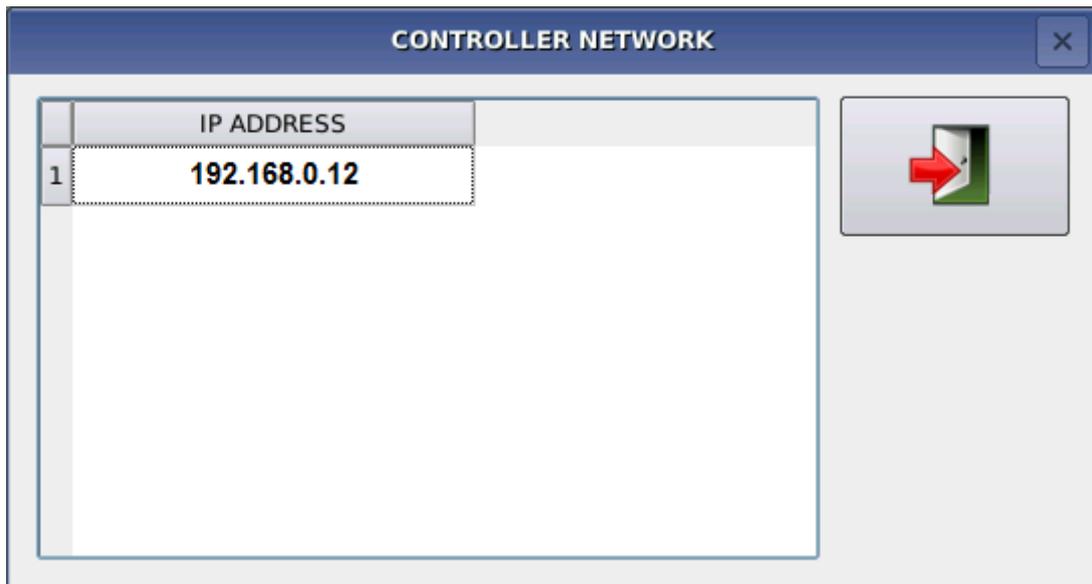
Master control unit configuration screen

For the master control unit, it is also possible to display all SLAVE control units currently connected. To display this information, simply press the button:



Network display button

The screen that opens will display in a table all the control units connected to the MASTER control unit showing their IP address.



SLAVE connected display screen

SLAVE control unit configuration

To configure a control unit as a *slave*, it is necessary to set the type parameter to SLAVE and the port and IP address that will be used to connect with the master. It is also possible to set a name to identify the control unit in the network.

NETWORK CONFIGURATION					
Type:	SLAVE				  
Name:	AC500-2				
Port:	3750				
Master settings					
IP address:	192	168	0	1	

Slave controller configuration screen

NOTICE

Changing the network configuration setting parameters will require a reboot of the control unit.

4.19 Configuration and use of the machine database

The software makes it possible to create and use a set of backup configurations for various types of machines in order to facilitate the configuration procedure of the controller when connected to a marking machine. Each configuration created in the machine database contains all the information required to configure the machine, i.e. *general data*, *axis data* and *head magazine data*. In the machine database it is possible to enter a new configuration with the current parameters set on the controller, modify a configuration, load a configuration and import or export a configuration from the database to a USB key.

To configure and use the machine database, you must be as user *distributor* or *manufacturer*.

To configure the machine database:

1. From the auxiliary functionality screen, press the system setup button;



System setup button

2. A screen will open showing the possible system parameters that can be changed;



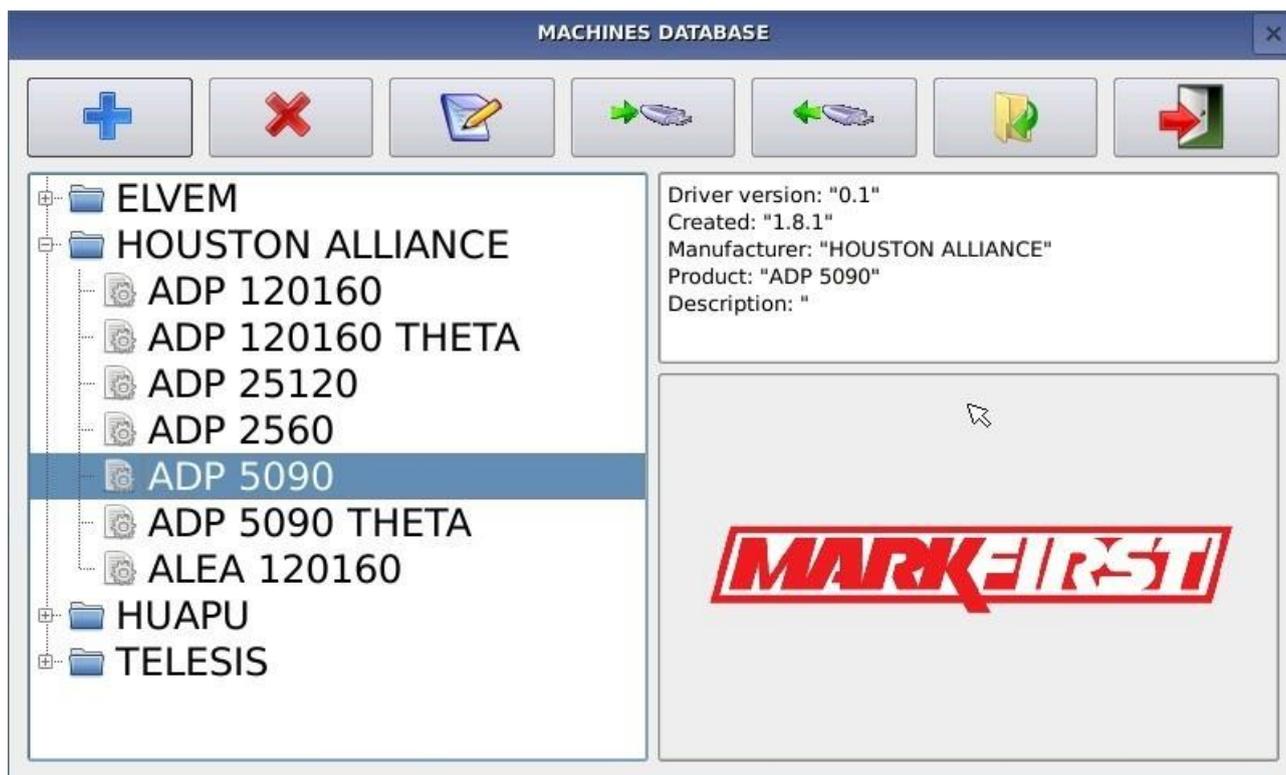
System parameter configuration screen

To change the machine database configuration, press the machine database configuration button.



Machine database configuration button

3. The software will display a screen where you can view and configure the machine database

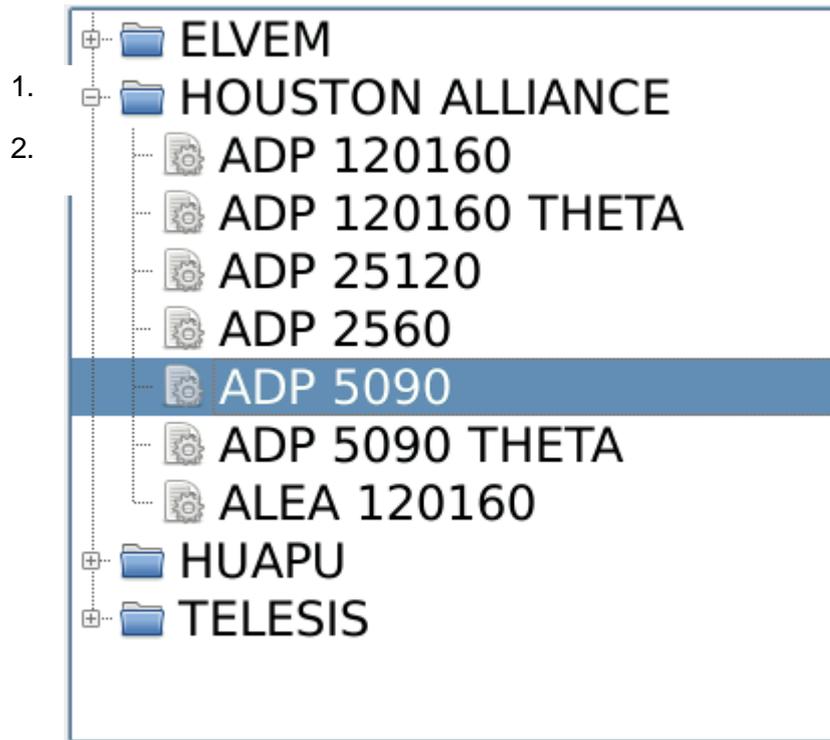


Machine database screen

The environment displayed will consist of a series of buttons for functionalities (add, edit, delete, etc. of a machine database configuration), a preview of the configurations present on the control unit (grouped by supplier) and a series of description information on the selected configuration.

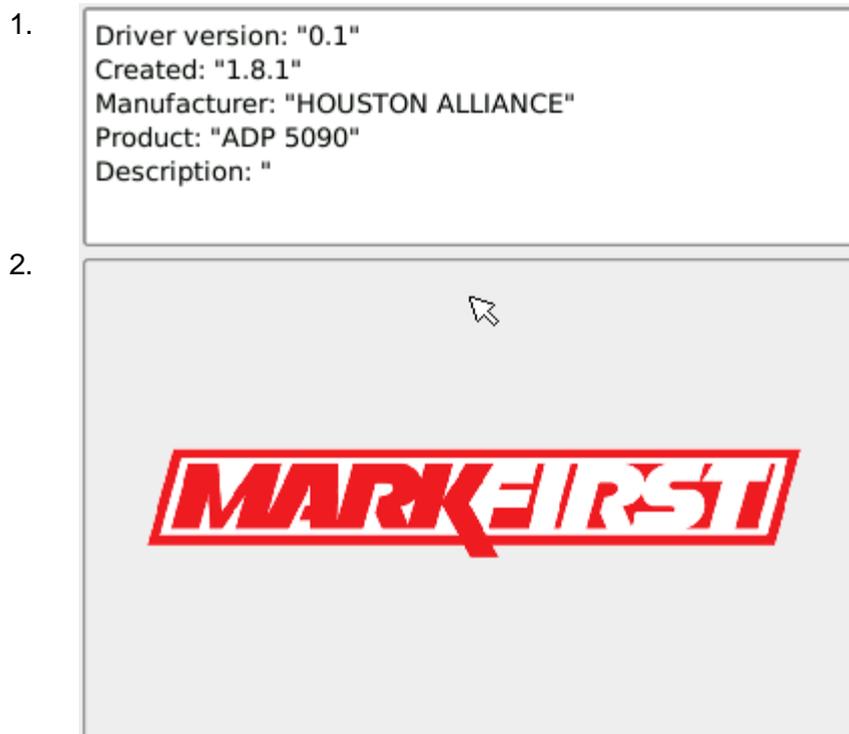


1. Button for entering a new configuration;
2. Button for deleting the selected configuration;
3. Button for changing the selected configuration;
4. Button for saving the selected configuration to USB;
5. Button for loading a configuration from USB;
6. Button for loading data of the selected configuration;
7. Room exit button.



Viewing database configurations by supplier

1. Displaying suppliers;
2. Display configurations for each supplier.



Display of selected configuration information

1. Information display of the selected configuration;
2. Optional image display set for selected configuration.

Adding a new configuration



Using the button for adding a new configuration, it will be possible to enter the configuration (general data, axis data and head stock) currently present on the control unit into the machine database. When the button is pressed, a new screen will be displayed, which will allow the descriptive data of the added configuration to be entered.

The screenshot shows a window titled "MACHINE INSERTION" with a close button (X) in the top right corner. The window contains the following elements:

- Manufacturer:** A text input field.
- Product:** A text input field.
- Description:** A large text area for entering a description.
- Image:** A text input field with a small red 'X' button next to it, likely for image selection or deletion.
- Preview:** A large empty rectangular area at the bottom left.
- Buttons:** On the right side, there are two buttons: a green checkmark button (OK) and a red 'X' button (Cancel).

Environment for entering a configuration

In this environment you will be able to enter descriptive data (supplier, product and description) and an optional image to be set for the configuration you wish to add. Using the two buttons on the right of the screen, you can decide whether or not to accept the changes made.

Deleting a Configuration

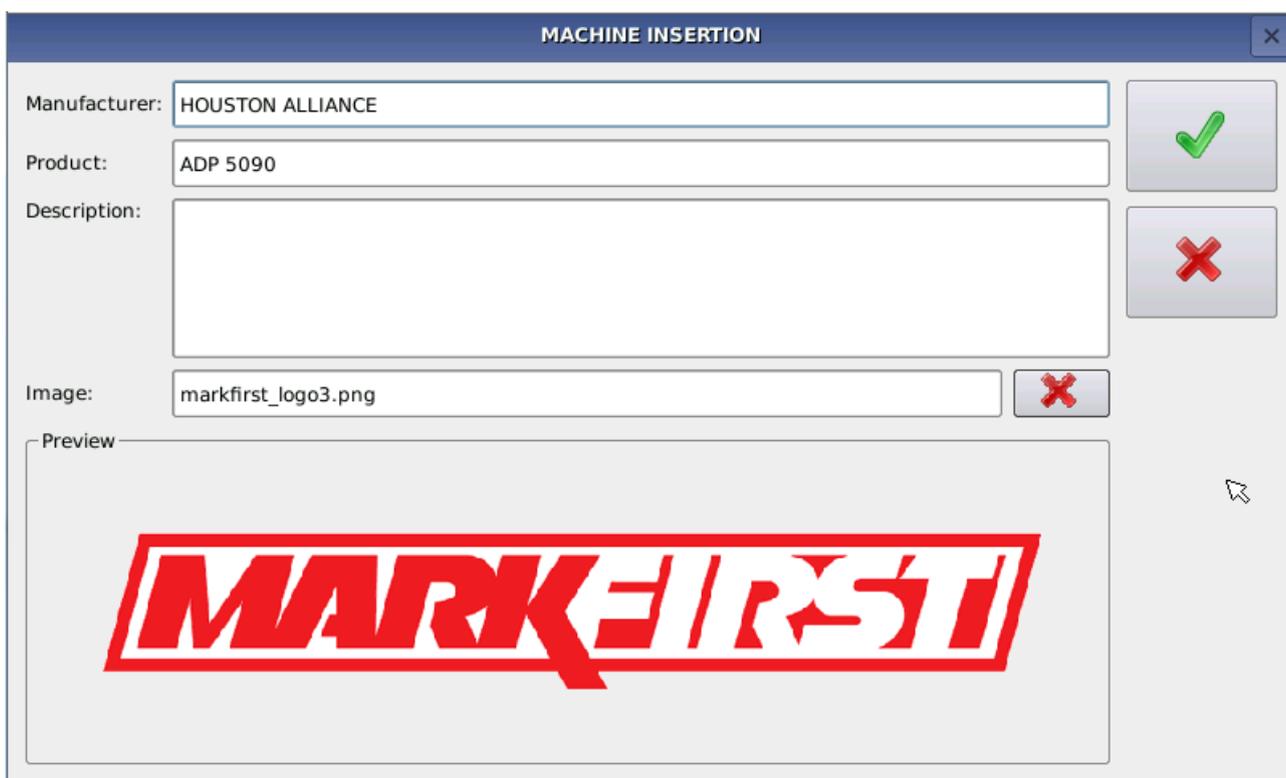


Using the delete configuration button, it will be possible to delete the configuration currently selected in the machine database list. When the button is pressed, a message will be displayed asking whether or not to confirm the deletion procedure. NOTE: If a supplier is selected and the delete button is pressed, it will be possible to delete all configurations associated with that supplier.

Modifying a configuration



Using the button for editing a configuration, it will be possible to edit (in terms of the configuration's descriptive data, not the data set) the configuration currently selected in the machine database list. When the button is pressed, a screen will be displayed where it will be possible to edit the descriptive fields and any image associated with the configuration.



Environment for entering a configuration

In this environment you can edit the descriptive data (supplier, product and description) and the optional image to be set for the configuration you want to add. Using the two buttons on the right of the screen, you can decide whether or not to accept the changes made.

Saving a configuration to USB



The button for saving a configuration to USB will save the currently selected configuration in the machine database list to the USB stick inserted in the controller. When the button is pressed, a message will be displayed

will ask whether or not to confirm the saving procedure. NOTE: if you select a supplier and press the save button, you will be able to save all configurations associated with that supplier.

USB loading of a configuration



Using the button for loading a configuration from USB, it will be possible to load a configuration from the USB stick inserted in the controller. Pressing the button will display a screen showing the configurations on the USB stick. NOTE: If you select a supplier and press the load button, you will be able to load all configurations associated with that supplier.

Loading a configuration



Using the button for loading a configuration it will be possible to load the data set of the currently selected configuration into the list of configurations on the control unit. When the button is pressed, a message will be displayed asking whether or not to confirm the configuration loading procedure. NOTE: If an attempt is made to load a dataset created with a later version of the software currently present on the control unit, a message warning of the operation will be displayed.

4.20 Date and time configuration

The software allows you to set the current date and time of the system. The date and time are used for flag commands, maintenance checks and other functions already described in this manual.

To configure the system date and time, you must be a *supervisor*, *distributor* or *manufacturer* user.

To configure the date and time:

1. From the main screen, press on the system date and time label;



Display of system date and time

2. A screen will open allowing you to configure the system date and time;



Date and time configuration screen

To change the system date or time, simply press the *set* button next to the field you wish to change.



Date configuration screen



Configuration screen now

To change a date or time value, press on the text box containing the field and enter the new value into the numeric keypad.

- Once the date and time has been changed, press the *edit* button on the right-hand side of the date and time configuration screen.

4.21 Window Header Configuration

The software also allows you to customise the header of the software windows. Window header means the title of the window bar.

To be able to set the window header, you must be as user *supervisor, distributor or manufacturer*.



Example of window bar with header

To set the window header:

- From the auxiliary functionality screen, press on the system setup button;



System setup button

2. A screen will open showing the possible system parameters that can be changed;



System parameter configuration screen

Enter the text to be displayed as a window header in the dedicated text box.

Windows header:

Box for setting the window header

3. Once the new value has been set, exit the environment to save the changes made.

4.22 Setting Modbus axis parameters

The software allows the Modbus axis parameters of the marking system to be configured according to the mechanical characteristics of the machine.

Modbus axes are optional axes that can be set up to a maximum of four from the general data environment. In addition to the optional Zeta and Theta axes, two custom axes called X1 and Y1 can also be configured. These axes, at the movement level, are completely managed by the operator via the marking programme; in fact, it will only be possible to control them via the GoTo field. These axes do not have a

internal logic managed by the software, such as Zeta and Theta axes, and provide further flexibility in the management of a marking system.

Modbus axes are special motors that can be controlled using the 485 serial port on the controller and via the Modbus protocol. Given the flexibility of the 485, which guarantees a true network (i.e. several devices connected at the same time), up to four Modbus axes can be controlled simultaneously.

In order to be able to configure the Modbus axis parameters, it is necessary to be a *manufacturer* user, as the data editing operations to be performed are extremely delicate and require special attention when entering values.

To configure Modbus axis parameters:

1. From the auxiliary functionality screen, press on the system setup button;



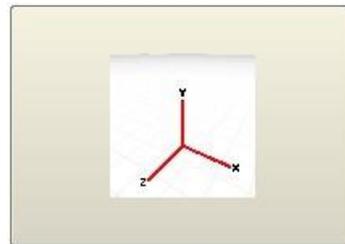
System setup button

2. A screen will open showing the possible system parameters that can be changed;



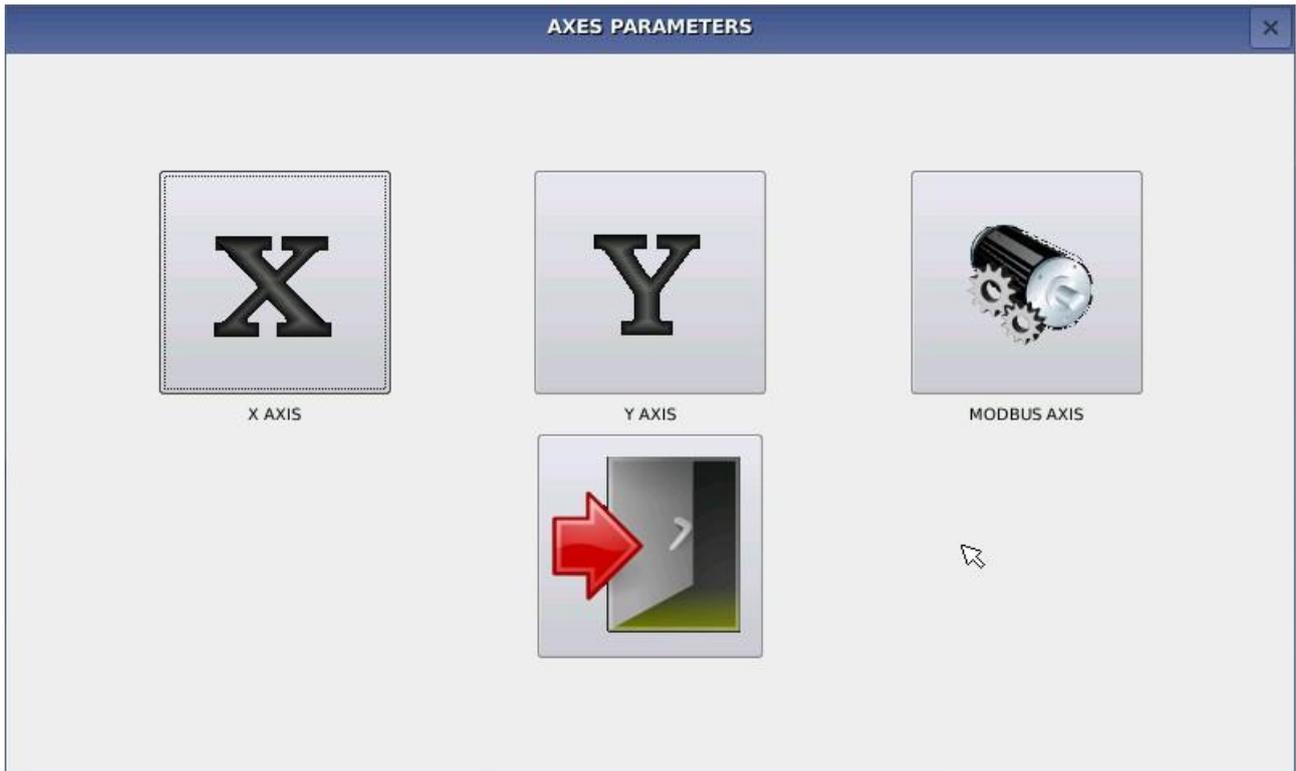
System parameter configuration screen

To change Modbus axis parameters, press the axis data button.



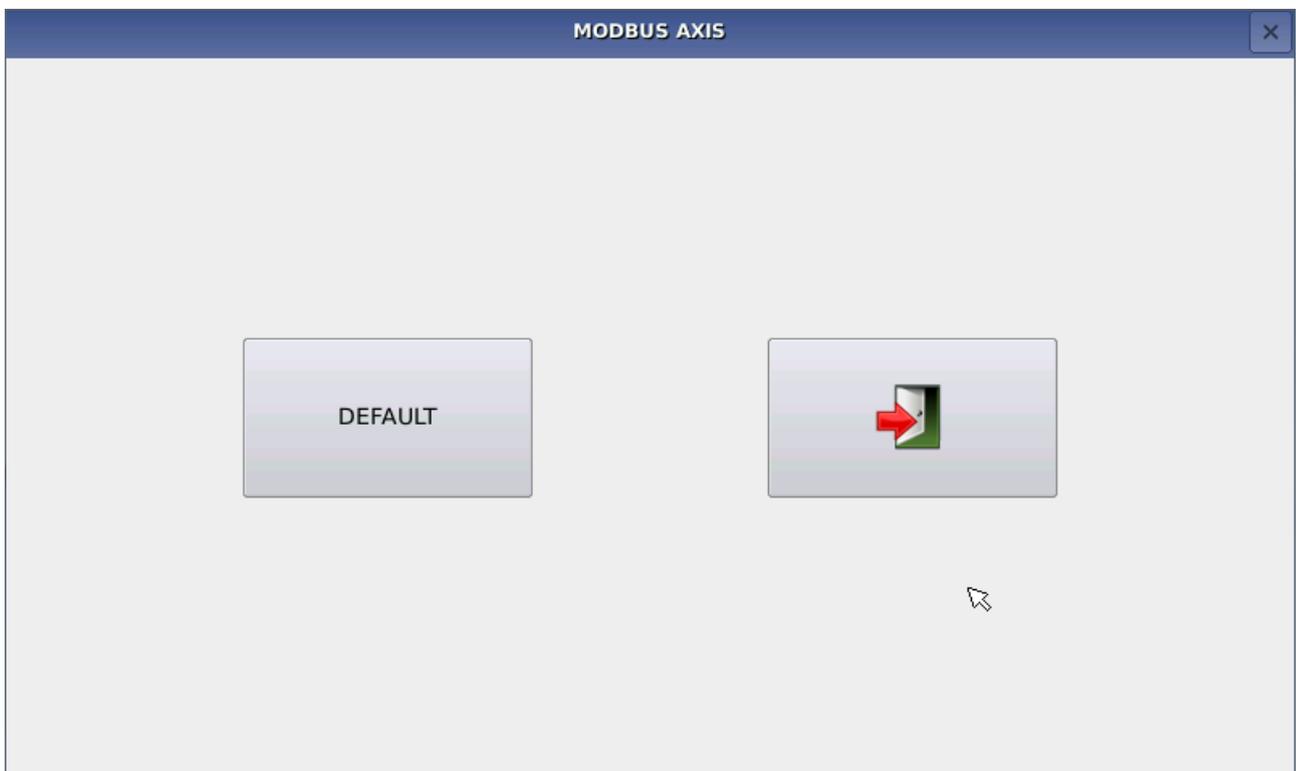
Axis data button

3. The software will display a screen where you can select which axis to configure.



Selection screen of the axis to be configured

4. The software will display a new screen with all configured Modbus axes listed. The list of axes displayed will show a description of how the axis was set up.



Modbus axis list screen

To select the Modbus axis to be configured, press the button for the axis you want to configure.

5. A new screen will open where you can enter the axis parameters of the software

MOTOR PARAMETERS: DEFAULT		
	PARAMETER	VALUE
1	Motor type:	DEFAULT
2	Slave ID:	0
3	Encoder pulses:	12800
4	Quote / round encoder (mm):	0
5	Maximum quote (mm):	0
6	Minimum quote (mm):	0
7	Maximum speed (mm/s):	0
8	Setting quote (mm):	0
9	Microswitch speed (mm/s):	0
10	Setting offset (mm):	0
11	Acceleration time (s):	0
12	Deceleration time (s):	0

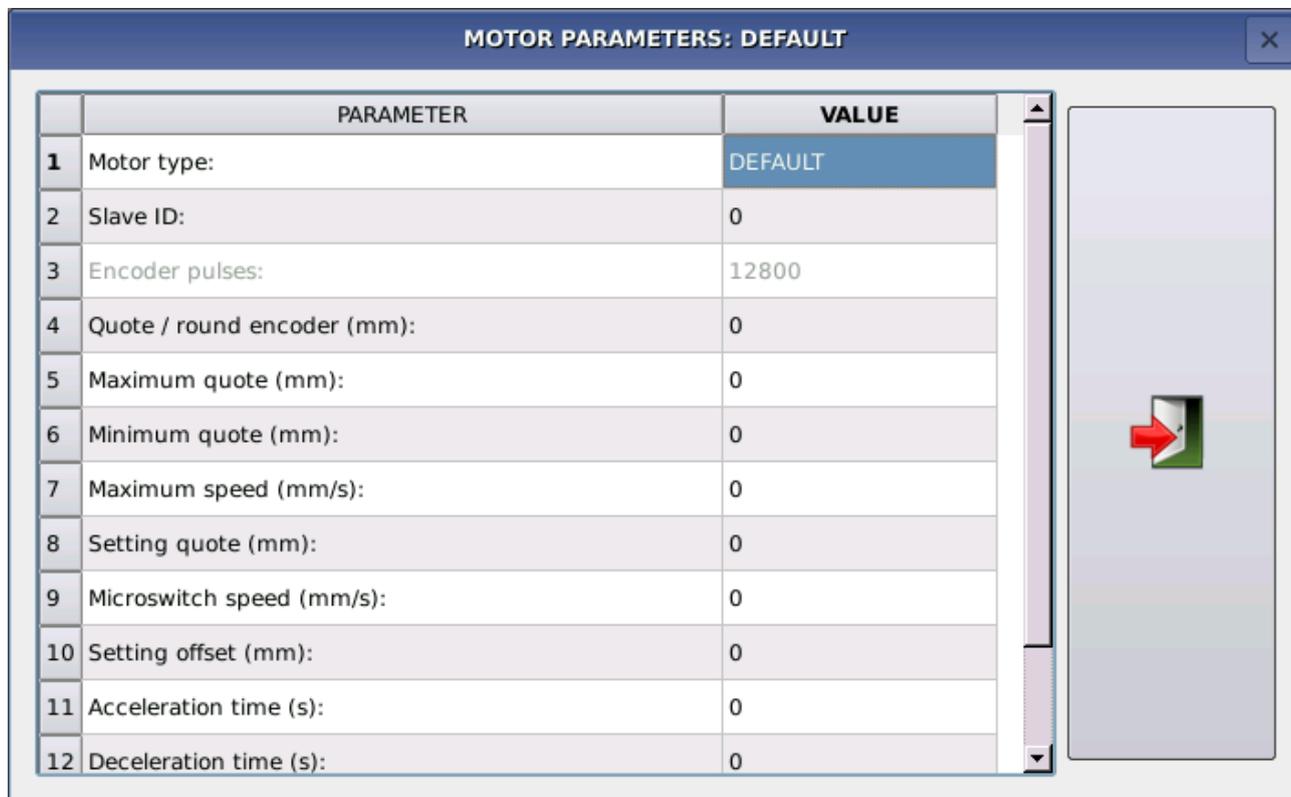


Table for entering axis parameters

6. Once the desired parameters have been set, exit the environment to save the changes made.

Below is a brief description of the Modbus axis parameters in the table:

	Parameter	Description
1	Engine Type	Indicates the type of axis to be associated with the motor being configured. This parameter has a strong influence on the motor configuration because the operating logic controlled by the software is associated with it. The default value represents an axis that is not set and therefore not used. <i>Value of your choice (DEFAULT - Z AXIS - THETA AXIS - X1 AXIS - Y1 AXIS).</i>
2	Slave ID	Indicates the identification index of the motor in the 485 serial network. This index must be unique for each Modbus motor connected to the control unit. <i>Numerical value.</i>
3	Number of encoder pulses	Indicates the number of encoder pulses. Data supplied by the encoder manufacturer. For FD1 motors used, the number of pulses is fixed at 12800.

		<i>Numerical value.</i>
4	Encoder ride share	Indicates the encoder revolution quota. The encoder revolution height is the space travelled in one encoder revolution. <i>Numerical value.</i>
5	Maximum quota	Indicates the maximum arrival altitude of the selected axis. <i>Numerical value.</i>
6	Minimum quota	Indicates the minimum arrival altitude of the selected axis. <i>Numerical value.</i>
7	Maximum speed	Indicates the maximum speed at which the axis can move. <i>Numerical value.</i>
8	Calibration quota	Indicates the height at which the axis is calibrated. <i>Numeric value.</i>
9	Calibration FC search speed	Indicates the speed of the axis being calibrated. During calibration, the axis moves in search of the calibration FC (End of Travel). <i>Numerical value.</i>
10	Calibration offset	Indicates a possible offset to be applied to the motor calibration dimension. <i>Numeric value.</i>
11	Acceleration time	Indicates the time in <i>seconds</i> to be used in the axis acceleration ramps. More precisely, the parameter indicates how long it takes to reach the set maximum speed (parameter 5) starting from standstill. <i>Numeric value.</i>
12	Deceleration time	This indicates the time in <i>seconds</i> to be used in the axis deceleration ramps. More precisely, the parameter indicates how long it takes the axis to stop from the set maximum speed (parameter 5). <i>Numeric value.</i>
13	Calibration direction	Indicates the calibration direction of the axis. This value depends on the direction of movement of the axis. FORWARD is towards large dimensions while REVERSE is towards small dimensions. <i>Value of your choice (BACK - FORWARD).</i>
14	Type FC calibration	Indicates the type of management of the axis calibration micro. The calibration micro can be NO (normally open) or NC (normally closed). <i>Choice of value (NO - NC).</i>
15	Calibration FC output quota	Indicates the displacement dimension of the axis if it is already on the limit switch during calibration. <i>Numeric value.</i>
16	Direction reversal	Indicates whether to reverse the Modbus axis movement directions. <i>Choice value (NO - YES)</i>

NOTE

If a Modbus axis is set up that is already configured as an optional third axis, in particular the Z and Theta axes, the optional third axis will be taken as the reference and the conflict will be displayed when configuring the Modbus axis.

4.23 AC500 Store

There is a section in the software dedicated exclusively to the purchase of additional functionalities that are not present in the software by default. These functionalities do not have a direct impact on the use of the software, as they do not modify its performance or mark-up management, and should therefore be considered as extensions that can be applied to the software. Among these extensions we can find:

- MODBUS SLAVE
- FIELDBUS
- CUSTOMISED SHIFT CODE MANAGEMENT
- LOADING PROGRAMMES FROM THE NETWORK
- ENABLING MARKING CYCLE
- CUSTOM HOUR CODES
- ENABLING TEXT DATA CONTROL
- REPEATABILITY CHECK
- RESET EXECUTIONS (OPERATOR LEVEL)
- EXTENDED PROTOCOL / ADVANCED FIELDBUS
- MULTI-POINT MARKING MANAGEMENT
- JULIAN DAY SPECIAL MANAGEMENT (FLAG COMMAND)

Below are descriptions of the extensions that can be purchased via the Store and the procedures to be followed to enable them.

To access the AC500 Store environment:

1. From the auxiliary functionality screen, press the system setup button;



System setup button

2. A screen will open showing the possible system parameters that can be changed;



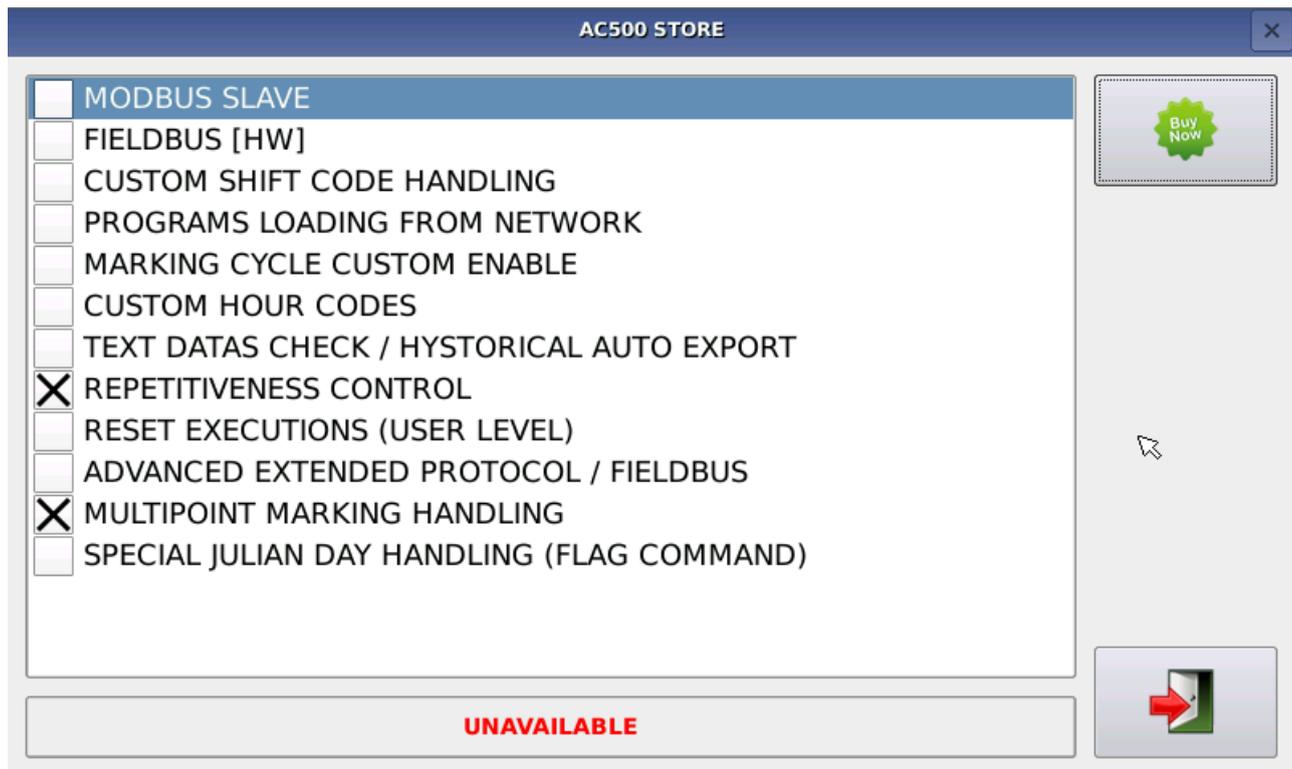
System parameter configuration screen

To access the AC500 Store environment, simply press on the environment access button.



Access button for the AC500 Store

3. The software will display a screen showing all functions/extensions that can be purchased for the software

**AC500 Store screen**

From this screen it will be possible to proceed with the purchase of extensions to be enabled on the AC500 control unit. It is important to bear in mind that the purchase of a function is only valid for the control unit on which the function is enabled. It is not possible in any way to enable a function of the store without first purchasing it on the control unit.

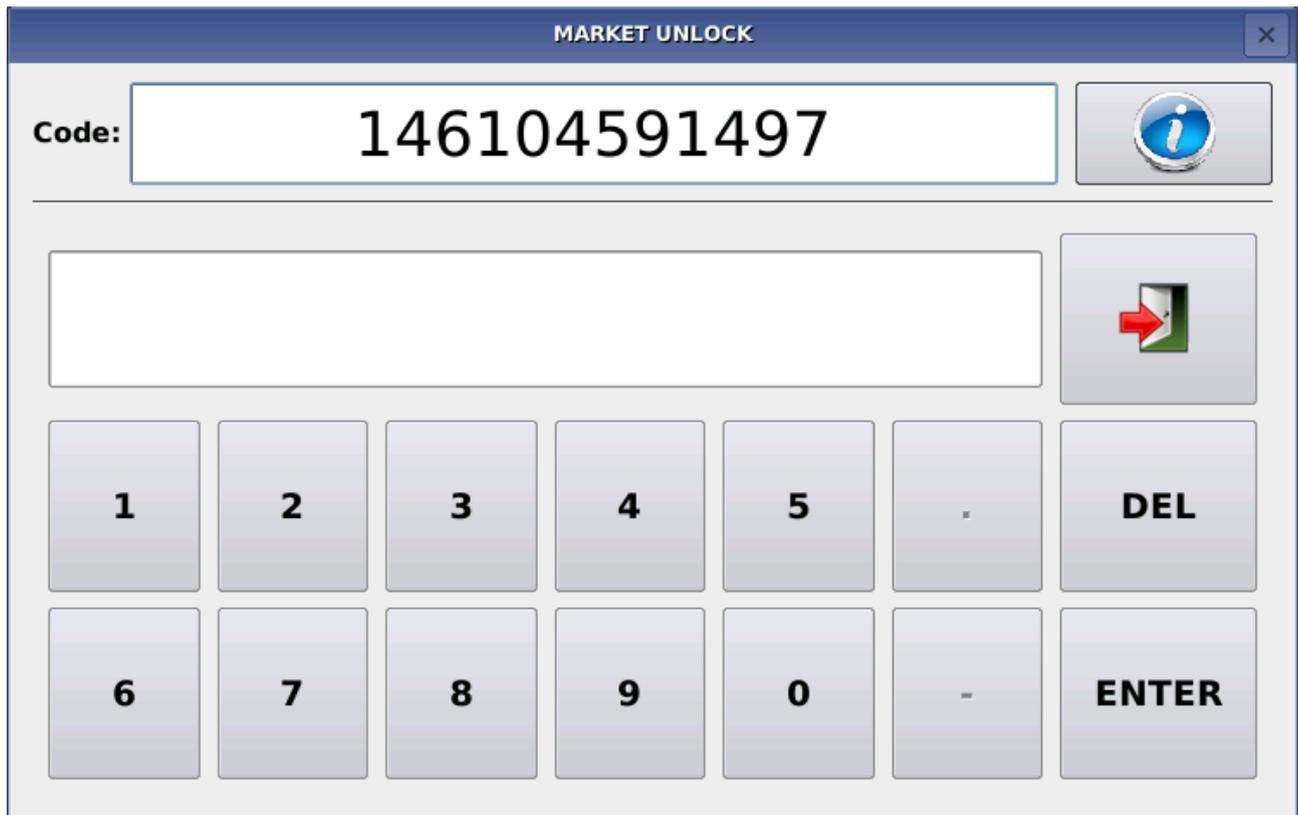
NOTE

Backing up a purchased function will not be possible.

To purchase a function from the AC500 Store environment, simply select any of the functions in the list and press the purchase button. If a function is already purchased and present on the controller, it will be displayed with a tick and the buy button will be disabled.

**Button purchase functionality**

The following screen will be displayed:



Unlocking Functionality Screen

The code associated with the required function to be purchased will be displayed at the top of this screen. In order to proceed with the purchase of this function, it is necessary to communicate the code to the control unit supplier. You will then be notified of the code to be entered to enable the purchase of the function.

If this is successful, the success of the operation will be announced on the screen and the newly purchased function will be displayed as available.

NOTICE

The code to be communicated for unlocking the functionality is valid for 7 days. This means that the unlock code must be entered within 7 days of it being communicated.

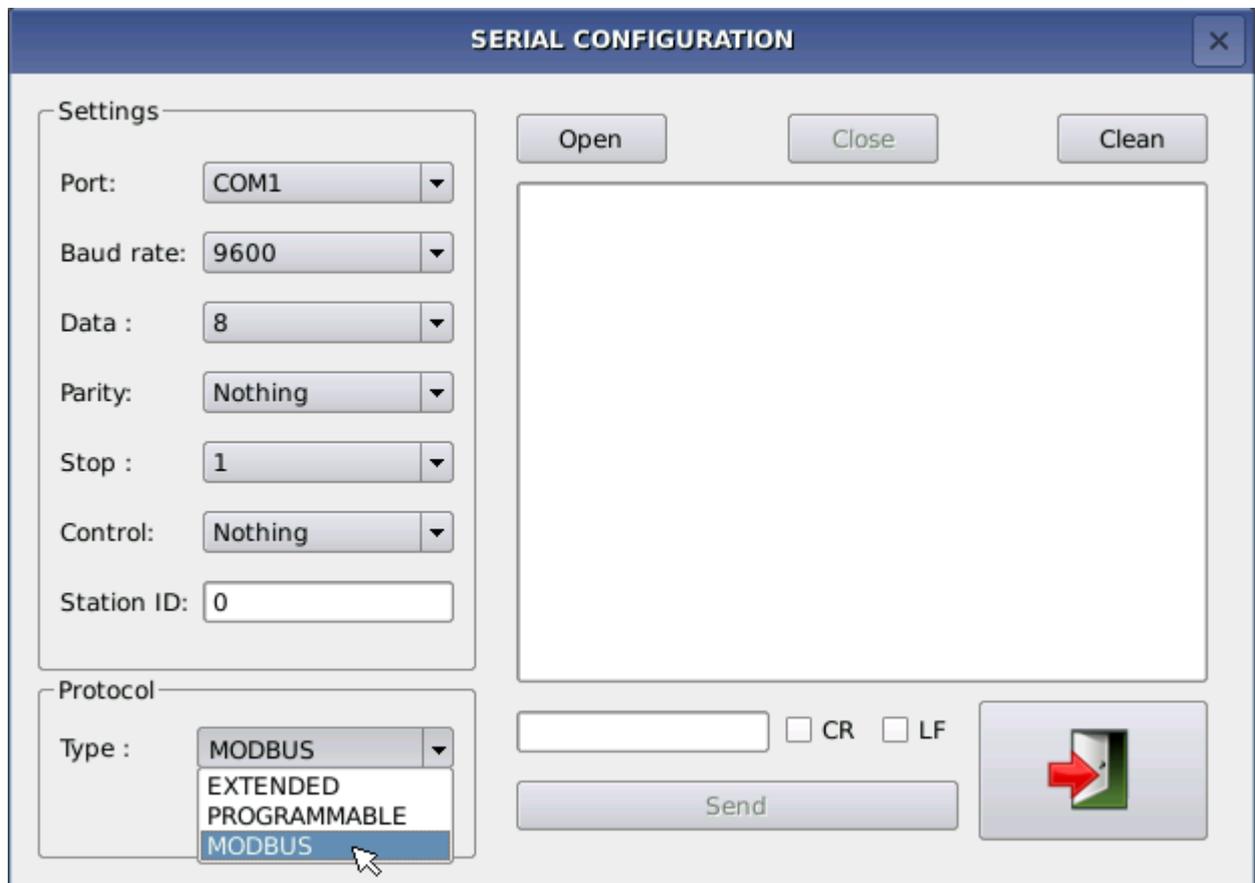
Modbus Slave

The Modbus Slave function allows a new protocol to be added for remote control of the control unit. With this management it will therefore be possible to control the AC500 control unit from any Modbus Master device. The operations that can be performed by the Modbus protocol are the same as those performed using the extended protocol (see the chapter on remote control unit management for further details), such as being able to put the control unit online, start a marking cycle, load a programme, etc.

The Modbus communication used for this particular functionality is of the RTU type and it will be possible to use it either from the RS232 channel or from the RS485 channel (see the chapter on serial wiring for possible connections).

Since the Modbus requires an identifier in order to recognise a node within a network, and since the RS485 can also guarantee the configuration of a possible network of control units connected via Modbus, it is necessary to set a unique ID for each control unit which will be used in the network (indicated as Station ID within the serial configuration page). Therefore, in order to ensure the correct functioning of the Master-Slave connection, it is absolutely necessary to specify on the Master which ID of the Modbus network it must communicate with. This is absolutely necessary for both RS232 and RS485 communication. It is important to bear in mind that the Master will also be part of the Modbus network and therefore the ID associated with it must be different from that of the various Slave nodes present.

To enable the Modbus protocol for remote control unit management, simply go to the serial port configuration environment and select Modbus as the protocol to be used.



Serial configuration screen with modbus protocol enabled

In accordance with the specifications of the Modbus industrial protocol, there is an Object Dictionary in the appendix of this manual, which is used to exchange data for remote control of the control unit.

Fieldbus

The Fieldbus Card Enable function allows you to manage the expansion cards that can be inserted on the controller for the managed Fieldbus protocols. Through this function on the controller, the remote control possibilities of the controller can be extended.

Fieldbus protocols are industrial protocols used worldwide. The AC500 allows some of these protocols to be managed precisely via these expansion cards that can be added to the controller.

Once the possibility of using Fieldbus protocols has been unlocked, it will be possible to enable the relevant parameter from the general data environment of the software in order to use this function.

An example of a managed Fieldbus protocol is EtherNet/IP. For full details on managed Fieldbus protocols, please refer to the appendices attached to the manual.

NOTICE

In order to use Fieldbus protocols, the expansion board mounted on the controller is required. Without it, it will not be possible to use the Fieldbus protocols as remote controller management.

NOTE

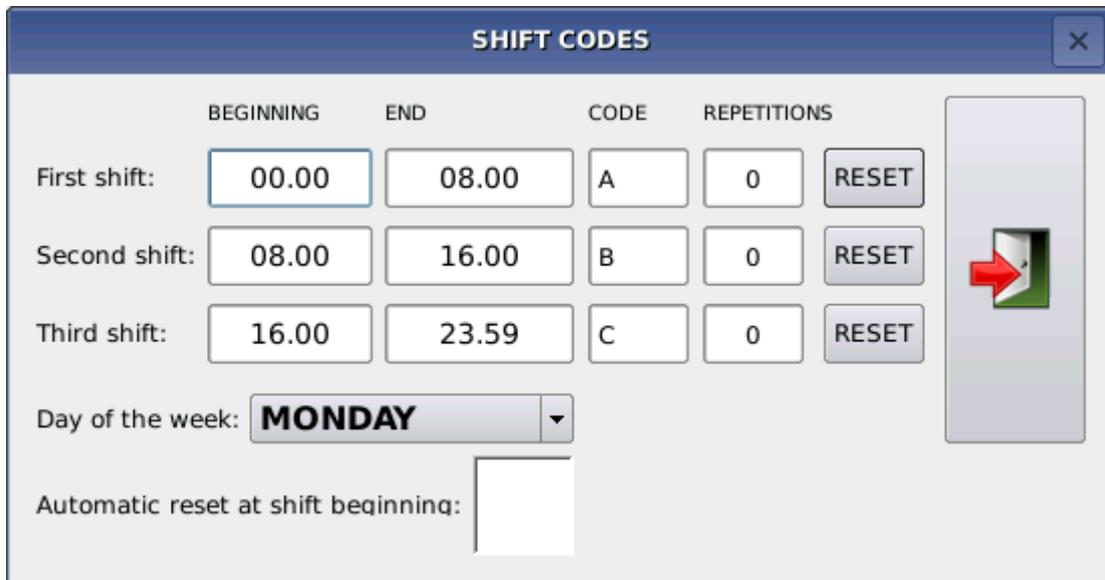
Board B can handle an automatic reset of the fieldbus board when the software is started. This operation is necessary to ensure correct operation when starting up the fieldbus module, which could cause some faults on control units with Board A present.

Customised shift code management

The customised shift code management function enables fully customised management of shift codes. Once this functionality has been purchased, it will be possible to enable the general parameter *Enabling customised management of shift codes*.

If the parameter *Enabling customised management of shift codes* is enabled, it will be possible to configure and customise advanced shift management settings. Advanced shift code configurations guarantee the entry of the three shift codes for each day of the week. This means that it will be possible to enter the

specific shift codes for each individual day of the week. When using the flag command associated with the shift code management with this management enabled, a check will be made according to the day of the week to take the correct shift code value.



Customised shift code parameter setting screen

With customised shift code management, the *Saturday* option will also be available. This particular mode can be optionally enabled by the operator and is used to manage the special case where the Saturday day must be divided into two shifts that alternate consecutively in the weeks.

To better explain this concept, let us imagine that we have two rounds, round A and round B. Shift A must work the first and third Saturday of the month, while Shift B must work the second and fourth Saturday of the month. It will be possible to manage this particular shift management mode.

In order to be able to manage this shift alternation correctly, the software requires the setting of a reference date from which to start considering the alternation of Saturdays, starting of course with the consideration of shift one. Attention: the date must be set before or at the same time as the first Saturday to be worked by shift one.

Suppose, for example, we assign the following values as values for the day Saturday:

- First round: **U**
- Second round: **V**

Depending on the configurations set for the *Saturday Option* management, the marking shifts will be entered as follows:

- First Saturday: **U**
- Second Saturday: **V**
- Third Saturday: **U**
- and so on...

	BEGINNING	END	CODE	REPETITIONS	
First shift:	00.00	08.00	P	0	RESET
Second shift:	08.00	16.00	Q	0	RESET
Third shift:	16.00	23.59	R	0	RESET

Day of the week: **SATURDAY**

Automatic reset at shift beginning: Saturday option:

Saturday Option parameter setting screen

Loading programmes from the network

The programme upload function from the network allows the AC500 controller to be configured with a network server that can be directly accessed to upload files such as work programmes, bitmap images, dxf, etc.

To use this management, you must configure the network server where you want to share files with the control unit. Of course, several power units can be connected to the same network and with the same server set up, so that multiple files can be shared between all AC500 power units.

For further details on the configuration and use of management, please refer to the appropriate section in this manual.

Enabling the marking cycle

The marking cycle enable function allows you to introduce safeties to the marking cycle start command from the touch screen. By appropriately setting the general management parameters, it will be possible to display a verification password before performing marking operations on the system.

This functionality can be very useful in control units placed on an automatic line where you want to exclude head control functions from touch. However, management leaves total control of the head from remote communication.

It is also possible to associate an expiry time to the unlocking, such that the password will be re-presented once the set time has elapsed since the previous unlocking operation.

Custom Time Codes

Enabling custom time codes will enable the configuration screen for customisable time codes. Just as is normally the case for other types of customisable codes (year codes, month codes, etc.), it will be possible to associate a customised string for each hour of the day. The flag command that can be used for management will be '%h'.

Enabling text data control

The text data control enable function will enable a new control management of the content of a text field to be marked. With this management a new menu will be proposed within the text fields within the editor environment from which it will be possible to select whether to enable the management and specify the control text to be used before a cycle start. If the text present does not comply with the format of the displayed text, marking will not be possible.

Repetitiveness control

The repeatability check function will enable the possibility of assigning a textual field whether it should be checked (and thus stored at the end of a cycle) to ensure the non-repetitiveness of the field's contents.

Reset executions (Operator level)

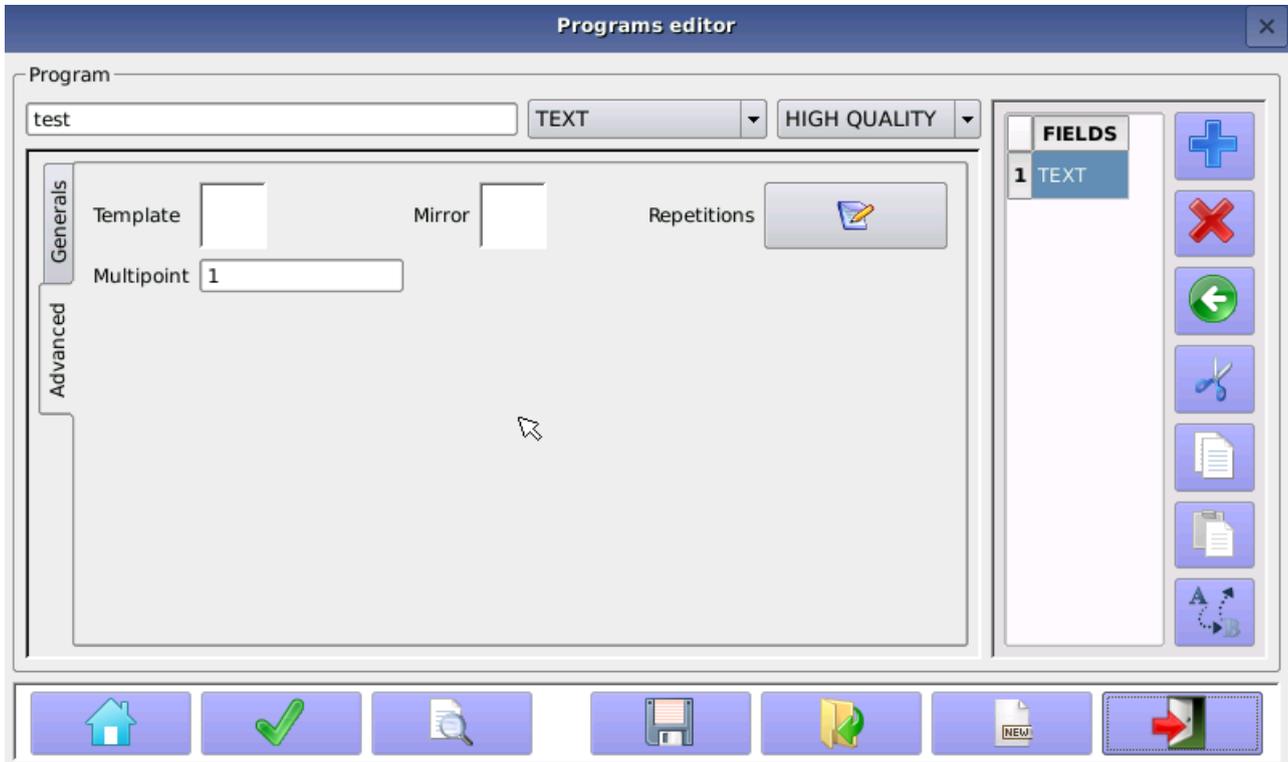
The reset executions function for the operator level will enable the possibility of resetting the counter of marking programme executions also from the operator level.

Extended Protocol / Advanced Fieldbus

The extended protocol and advanced fieldbus management function allows the use of command types designed for editing text or arc text fields.

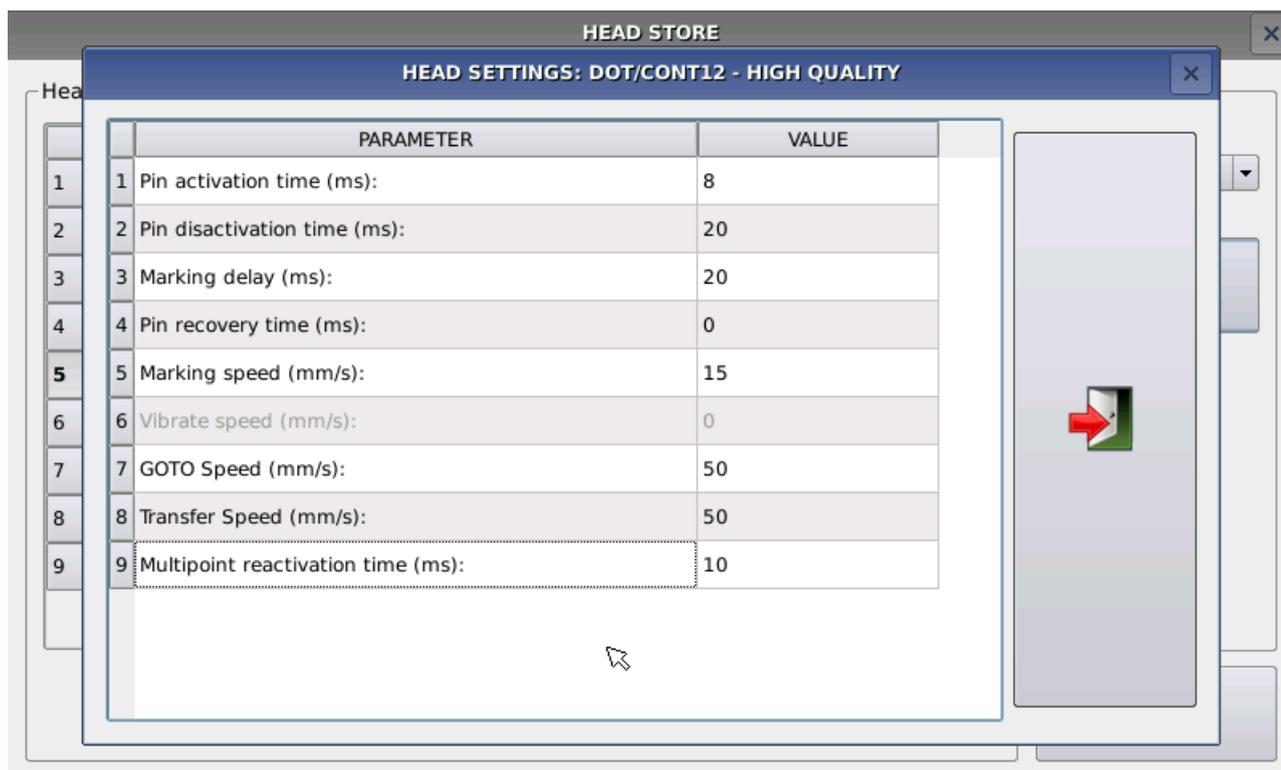
Multi-point marking management

The multi-point marking management function allows each point in the programme to be marked several times. With this management it will be possible to display and modify within the advanced data of each marking field the number of repetitions of the points filled in such a way that each point of that field will be executed n times according to the set value.



Example page of a text field with multi-point management

During execution, these repetitions will be managed in such a way as to command the descent/ascent of the tip for the number of times set in the programme field, taking into account the time required for the head to ascend before commanding the repetition of the descent. This particular time can be set within the marking head data.



Example head configuration with multipoint management

Special Julian Day Management (flag command)

The special Julian day management function will allow the `%j` flag command to be used within the programme fields. This management provides for the same operation as the `%J` flag command with the particularity that the Julian day increment will occur every n days, where n is a value which can be set within the general data of the software.

An example:

Date	Julian Day (standard)	Julian Day (special, increase to 1).
01/01/2015	001	001
02/01/2015	002	001
03/01/2015	003	003
04/01/2015	004	003
17/11/2015	321	321
18/11/2015	322	321
19/11/2015	323	323
20/11/2015	324	323

4.24 Fieldbus card configuration

Once the management of Fieldbus cards is enabled from the AC500 Store environment and from the parameter in the general data, it will be possible to access the configuration environment of the Fieldbus card connected to the controller. The software is able to

automatically recognise the type of card connected and then automatically propose the necessary configurations for the correct use of the card.

The management of Fieldbus protocols only adds further possibilities in the remote control of the control unit, as is the case with the extended protocol on the RS232 serial.

NOTICE

The Fieldbus board uses the RS232 serial port internally. This means that it will not be possible to use this serial port for other software functions.

To access the Fieldbus configuration environment:

1. From the auxiliary functionality screen, press the system setup button;



System setup button

2. A screen will open showing the possible system parameters that can be changed;



System parameter configuration screen

To access the Fieldbus configuration environment, simply press the environment access button.



Fieldbus configuration button

3. The software will display a screen where all information concerning the connected Fieldbus module will be shown:

FIELDBUS CONFIGURATION

Fieldbus data

Serial number: 1077006538

Firmware: 2.15.1

Type: EtherNet/IP

Fieldbus configuration

IP address: 0 0 0 0

Subnet mask: 0 0 0 0

Gateway: 0 0 0 0

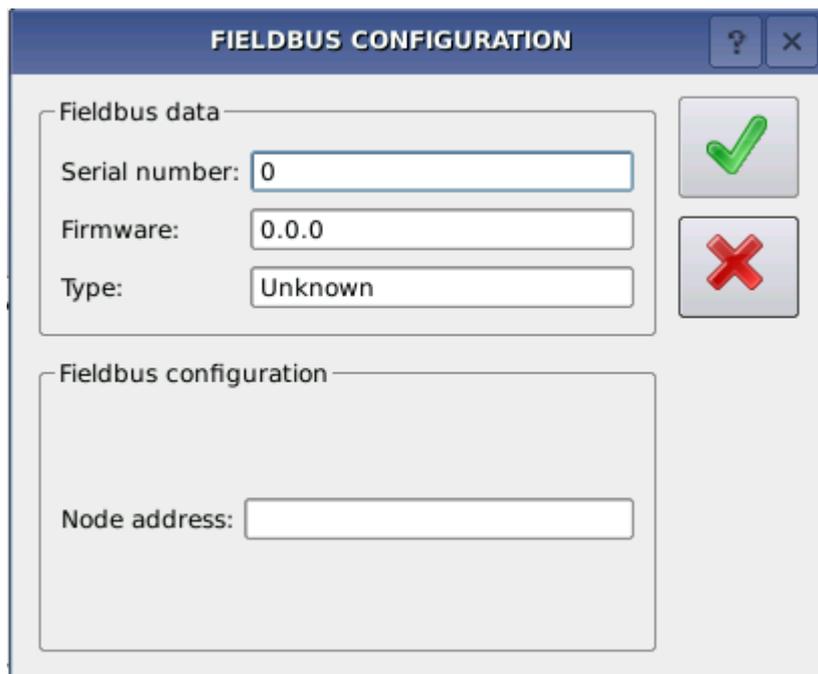
DHCP ON DHCP OFF

Example Fieldbus card configuration screen

At the top of the screen displayed will be read-only information that is the same for all Fieldbus modules:

- *Serial number*: production serial number of the Fieldbus card;
- *Firmware*: firmware version on the Fieldbus module;
- *Type*: the type of connected Fieldbus module (e.g. Etheret/IP).

This information is also useful for checking the correct functioning of the Fieldbus card connected to the AC500 series. In fact, if no module has been connected to the control unit, this data will be displayed as follows:



Example of screenshot with unrecognised module

Should the control unit fail to detect the module information, it is necessary to try restarting the control unit and check that the connections have been made correctly.

At the bottom of the screen, however, all information concerning the type of module connected will be displayed. These configurations may not be necessary in some cases.

For any information on the modules and their configurations, please refer to the dedicated appendices in this manual.

4.25 Programme loading management from the network

Once the management of program loading from the network is enabled from the AC500 Store environment and from the parameter present in the general data, it will be possible to access the network server configuration environment that contains the files to be shared with the AC500 controller. It is important to bear in mind that the server with which the AC500 controller will connect must be running a Windows operating system.

To configure network server settings:

1. From the auxiliary functionality screen, press the system setup button;



System setup button

2. A screen will open showing the possible system parameters that can be changed;



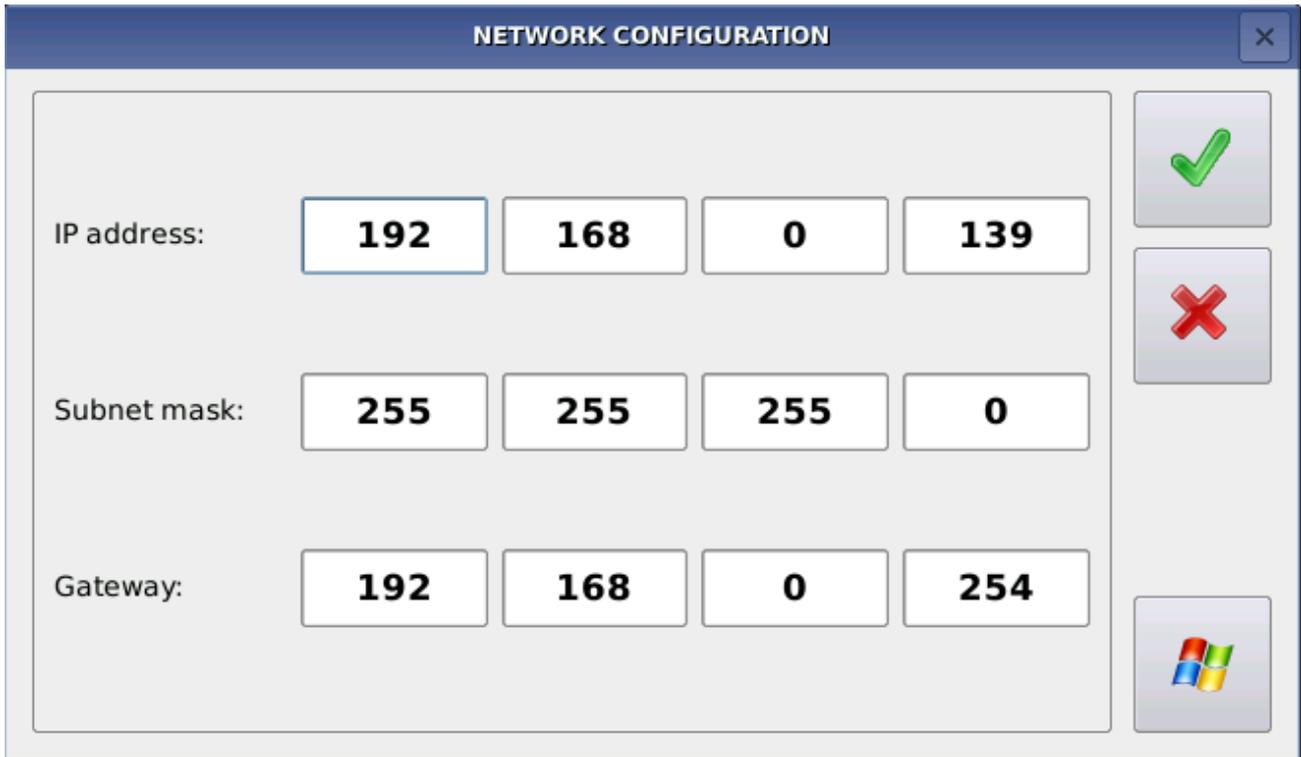
System parameter configuration screen

To change the configuration of the network server settings, press the Network Settings Configuration button.



Network settings configuration button

3. The controller's network settings screen will open



Network settings configuration screen

To change the network server settings, press **t h e** network server configuration button.



Network server settings configuration button

4. The software will display a screen on the screen where you can set the network server settings data

The screenshot shows a software window titled "NETWORK SERVER". Inside, there is a "Server configuration" section. The IP address is set to 192.168.0.69. The shared folder is "AC500". There are checkboxes for "Read only" and "Show password characters". On the right side of the configuration area, there are two buttons: a green checkmark and a red X.

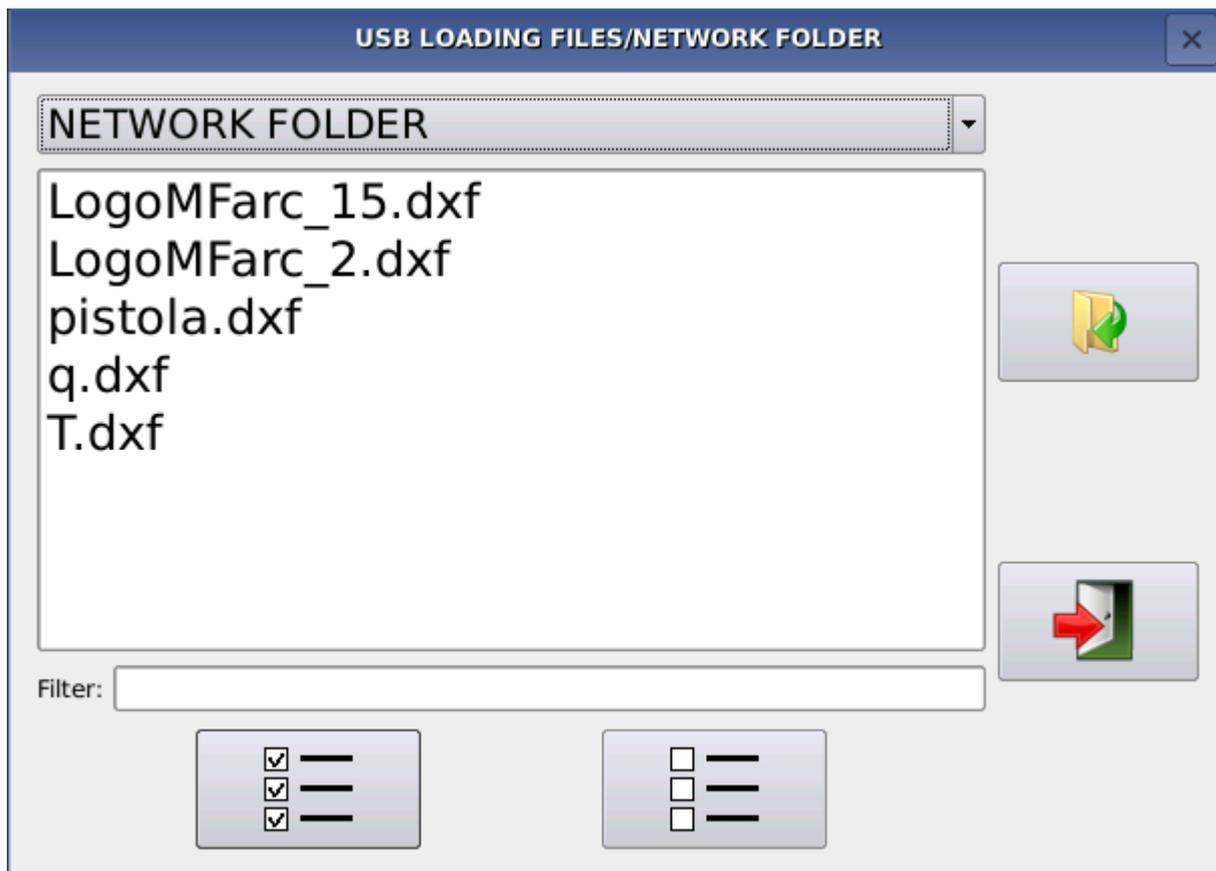
Network server configuration screen

This environment will contain all the parameters needed to configure the network server and the connection that the control unit must establish to guarantee access to the shared files.

<i>IP address</i>	The IP address of the network server.
<i>Shared folder</i>	The name of the folder on the server and set with the share property on the network.
<i>Reading only</i>	Indicates whether the shared folder on the server has read-only privileges (no write operations allowed).
<i>Username</i>	Indicates a user name, if any, to be used in the event that the server requires a login to the connection.
<i>Password</i>	Indicates the password to be used for the user set up.

Once the necessary data has been entered, it is sufficient to close the environment with consent. Of course, upon exiting the environment, the software will attempt to establish a connection with the set server and if problems are presented, a message window with information on the errors presented will appear on the screen.

Once the server has been properly configured, it will be possible to access the shared folder from any file selection screen.



Example of file display from network folder

Furthermore, once the server has been configured to load programmes from the network, a new button will be displayed on the main screen of the software.



Main software screen with programme loading from network enabled



From this button  it will be possible to force a reconnection operation to the server by the user. This operation may come in handy in the event that there are anomalies in the connection to the server, which can possibly be reset or verified with a reconnection attempt.

4.26 Alarm history display

An alarm history is managed in the software to display all alarms and emergencies presented in the software. The alarm history is updated each time an alarm is detected by the software. If the system is in an alarm state, it will not be possible to carry out axis movements and marking cycles until the reported alarm has been reset.

Below is a list of the alarms managed and displayed by the software:

- **User stop** alarm: alarm recorded when the user presses the movement stop or cycle stop;
- **Thermal** alarm: alarm recorded when a thermal alarm occurs in the controller;
- **Power On** alarm: alarm recorded when the user presses the Power Off button on the controller;
- **User abort** alarm: alarm recorded when a user abort occurs via an external button connected to the appropriate input;
- **Abort print** alarm: alarm recorded when a user stop occurs via a logic signal command;
- **Laser system ok alarm** (*only for laser machines*): alarm recorded when the digital input of laser system ok is not high. The laser is not connected or powered;
- **Laser armed alarm** (*only for laser machines*): Alarm recorded when the laser armed input is not high. The laser has not armed after the laser enable output has been raised.
- **Axes out of check area** alarm: alarm recorded when stopping the execution of a programme if one of the axes has gone outside the boundaries of the programme check area.
- **Solenoid valve short circuit alarm**: alarm recorded when the firmware detects a short circuit on the solenoid valve.

- **Motor short circuit alarm:** alarm recorded when the firmware detects a short circuit on a motor.
- **Modbus motor alarm:** alarm recorded when an alarm occurs on a modbus motor.
- **Modbus motor:** alarm recorded when any fault occurs on a modbus motor.

It is possible to view the alarm history from the auxiliary functions environment. The history can be viewed by all users and at any operational level.

To view alarm history:

1. From the auxiliary functions screen, press on the alarm history button;



Alarm history button

2. A screen will open where a table with all the alarms presented by the software will be displayed;

FAULT HISTORY				
	ALARM	DATE	MARKING HEAD	
1	MOTORS ALARM (11)	Sat Mar 26 16:09:50 2011	DOTS	M
2	POWER OFF ALARM (3)	Sat Mar 26 16:09:37 2011	DOTS	M
3	LASER SYSTEM OK ALARM (6)	Sat Mar 26 16:11:53 2011	VIBRATE	M
4	STOP BY USER (1)	Sat Mar 26 14:14:20 2011	DOTS	U
5	MOTORS ALARM (11)	Sat Mar 26 09:54:41 2011	DOTS	U
6	MOTORS ALARM (11)	Sat Mar 26 05:05:30 2011	DOTS	U
7	STOP BY USER (1)	Sat Mar 26 04:04:52 2011	DOTS	M
8	STOP BY USER (1)	Sat Mar 26 03:55:02 2011	DOTS	M
9	MOTORS ALARM (11)	Fri Mar 25 19:34:37 2011	DOTS	U
10	MOTORS ALARM (11)	Fri Mar 25 19:34:37 2011	DOTS	U
11	STOP BY USER (1)	Fri Mar 25 22:32:21 2011	DOTS	U
12	MOTORS ALARM (12)	Fri Mar 25 22:00:21 2011	DOTS	U
13	MOTORS ALARM (12)	Fri Mar 25 20:55:43 2011	DOTS	U

Alarm history display screen

For each alarm, a description of the alarm will be displayed with its unique identifier, the date and time it was recorded, the head type selected when the alarm was recorded, the user level selected and the programme that was loaded when the alarm was recorded.

4.27 Displaying system statistics

The software stores a series of statistics to display a range of useful data for the management and maintenance of the software. The stored statistics are related to software execution times, processing times, number of marking cycle executions, etc.

It is possible to view system statistics from the auxiliary functions environment. System statistics are visible to all users and with any operating level. Statistics resetting operations are only accessible by a user with operating level *manufacturer* (resetting of all statistics) or *distributor* (resetting of partial statistics).

To view system statistics:

1. From the auxiliary functions screen, press on the system statistics button;



System statistics button

2. A screen will open where system statistics concerning execution times and programmes in the controller's memory will be displayed;

STATISTICS X

Execution time

TOTAL

	TIME	PERCENTAGE
STANDBY	13 days, 14:19:33	106.0%
MARKING	0 days, 05:39:25	-6.0%
TOTAL	12 days, 19:58:57	100.0%

PARTIAL

	TIME	PERCENTAGE
STANDBY	13 days, 14:19:33	106.0%
MARKING	0 days, 05:39:25	-6.0%
TOTAL	12 days, 19:58:57	100.0%

Programs

	NAME	EXECUTIONS	PERCENTAGE
1	DEMO DOT.prg	4	1.9%
2	DEMO VIBRA...	0	0.0%
3	PGM DOT.prg	0	0.0%
4	PGM VIB.prg	0	0.0%
5	TEST 5090.prg	92	43.4%
6	test.prg	8	3.8%
7	test2.prg	108	50.9%

Executed programs total:

Executed programs partial:

System Statistics Display Screen

System statistics are divided into: system execution times, system maintenance dates and programme execution.

Execution time

TOTAL

	TIME	PERCENTAGE
STANDBY	13 days, 14:19:33	106.0%
MARKING	0 days, 05:39:25	-6.0%
TOTAL	12 days, 19:58:57	100.0%

PARTIAL

	TIME	PERCENTAGE
STANDBY	13 days, 14:19:33	106.0%
MARKING	0 days, 05:39:25	-6.0%
TOTAL	12 days, 19:58:57	100.0%

Runtime display page

Execution times are divided into total times and partial times. Total times relate to the total execution of the system, i.e. the total actual time in which the software was executed. Partial time is the current system execution time, i.e. the time from which the system was started. The total time takes into account all partial execution times.

Partial and total execution times are divided into stanby time, marking time and total time. The marking time indicates the actual time in which the

software is in the marking phase; the stanby time indicates the time when the software is running but is not marking, and the total time indicates the sum of the marking and stanby times.

Total execution times are not resettable from any operational level while partial execution time can be reset by the *distributor* or *manufacturer* user.

Execution time

TOTAL

	TIME	PERCENTAGE
STANDBY	13 days, 14:19:33	106.0%
MARKING	0 days, 05:39:25	-6.0%
TOTAL	12 days, 19:58:57	100.0%

RESET

PARTIAL

	TIME	PERCENTAGE
STANDBY	13 days, 14:19:33	106.0%
MARKING	0 days, 05:39:25	-6.0%
TOTAL	12 days, 19:58:57	100.0%

RESET

Runtime display page for distributor/manufacturer user

To reset the partial run time, simply press the *reset* button next to the table and confirm the cancellation request.

Programs

	NAME	EXECUTIONS	PERCENTAGE
1	DEMO DOT.prg	4	1.9%
2	DEMO VIBRA...	0	0.0%
3	PGM DOT.prg	0	0.0%
4	PGM VIB.prg	0	0.0%
5	TEST 5090.prg	92	43.4%
6	test.prg	8	3.8%
7	test2.prg	108	50.9%

Executed programs total:

Executed programs partial:

**Programme execution display page**

Programme execution statistics are used to display how many times individual programmes in the controller's memory have been executed. In the programme statistics table, you can see all the programmes in the controller's memory with their execution statistics. The statistics of a programme concern the number of executions and the percentage of executions of that programme in relation to the total number of executions.

The user with a *manufacturer* operating level can change the number of partial programme executions.

Programs			
	NAME	EXECUTIONS	PERCENTAGE
1	DEMO DOT.prg	4	1.9%
2	DEMO VIBRA...	0	0.0%
3	PGM DOT.prg	0	0.0%
4	PGM VIB.prg	0	0.0%
5	TEST 5090.prg	92	43.4%
6	test.prg	8	3.8%
7	test2.prg	108	50.9%

Executed programs total:

Executed programs partial:



Display page of programme executions per manufacturer user

To *reset* the number of partial programme executions, press the *reset* button next to the text box where the number of executed programmes is displayed and confirm the request to delete the data.

4.28 Backing up system data

The software allows backup files to be created from which the status of the system data can be restored at any time. Backup files can also be saved on a USB stick and taken out of the controller. It is also possible to load a backup from the USB key to restore the system data to the previously saved settings.

System data can also be deleted from the backup environment. The data that can be deleted are exclusively those saved on the controller memory.

In order to make system backups or restore software settings, the user's operating level must be set to *distributor*, *manufacturer* or *supervisor*.

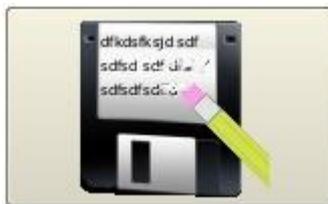
The data that can be stored in backups are as follows:

- **General software parameters;**
- **Axis parameters;**

- **Warehouse heads;**
- **General software configurations** (Omni serial number, shift codes, year codes, etc.).
- **Statistics;**
- **Programmes;**
- **Bitmap images;**
- **DXF drawings;**
- **TTF font;**
- **Machine maintenance parameters;**
- **Languages;**
- **Custom fonts;**
- **Scoring history;**
- **CSV;**
- **Machine database;**
- **Virtual keyboards;**
- **Printer;**
- **Marking control.**

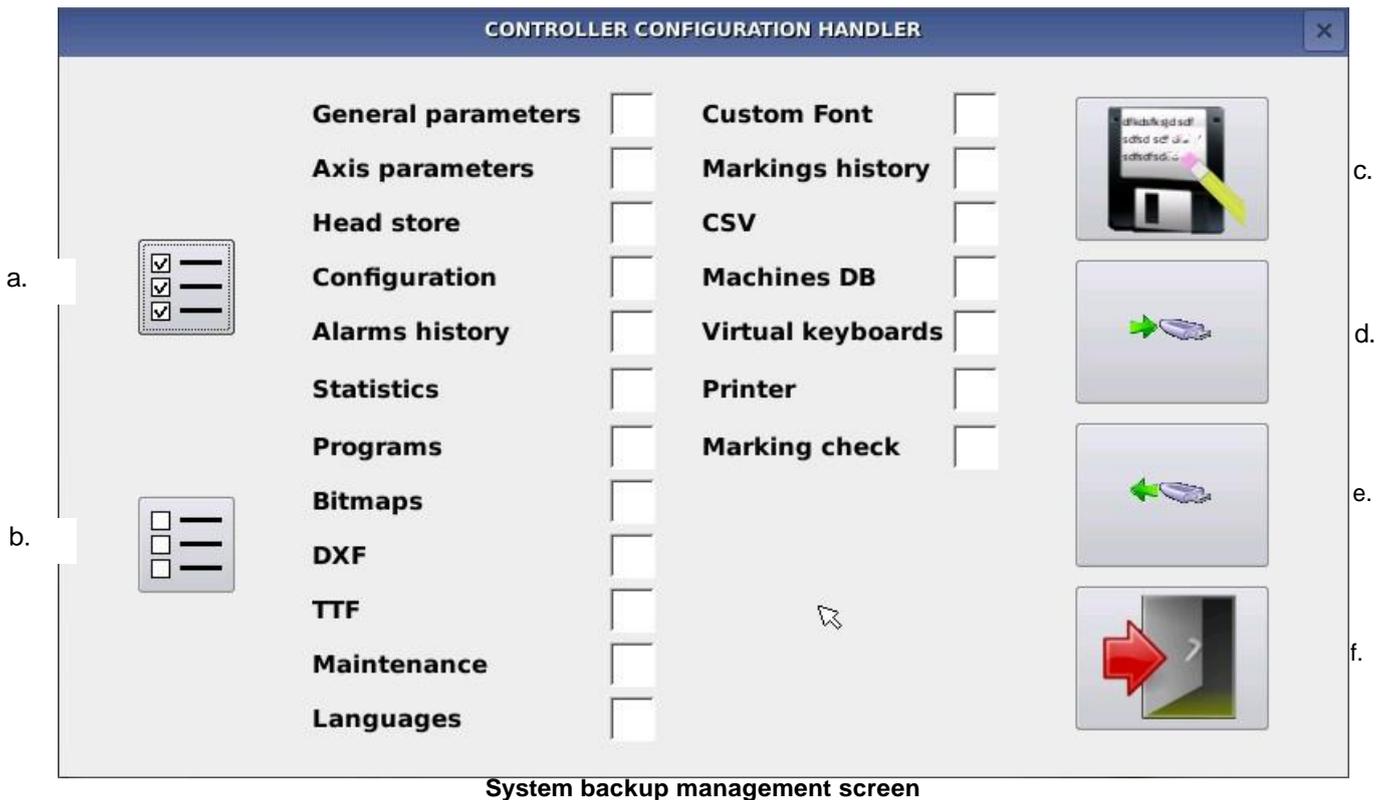
To back up system data:

1. From the auxiliary functionalities screen, press the backup button;



Backup button

2. A screen will open where the possible backups to be generated or restored will be displayed;



- a. Button for selecting all backup parameters;
- b. Button to deselect all backup parameters;
- c. Button to delete selected backup parameters;
- d. Backup creation button for selected parameters;
- e. Backup reset button for selected parameters;
- f. Exit button for the backup environment.

From this environment, it is possible to select the backup parameters that can be managed for backup, delete or restore operations.

3. Select the parameters with which you want to perform a backup operation by pressing on the parameter or by selecting all parameters from the appropriate button.
4. Once the parameters have been set, press the button for the operation you want to perform:
 - Deletion of **selected** parameters: Deleting selected parameters will reset all parameter settings.

NOTICE

By deleting selected parameters, all data of those parameters in the controller will be lost. Make sure you have a backup of those parameters before proceeding with the deletion.

NOTE

It is not possible to delete all TTFs from the backup environment. To delete TTFs, it is necessary to go to the TTF management environment from the file management environment.

- **Backup of selected parameters:** the selected parameters will be saved on the USB stick inserted in the controller.

When backing up the selected data, a prompt will appear to enter the name under which the backup is to be saved. In this way it will be possible to save several backups of the software configuration and each backup can be named as desired. Backups will be named with the header "*mcs_<name>*" where <name> will be the data entered by the user.



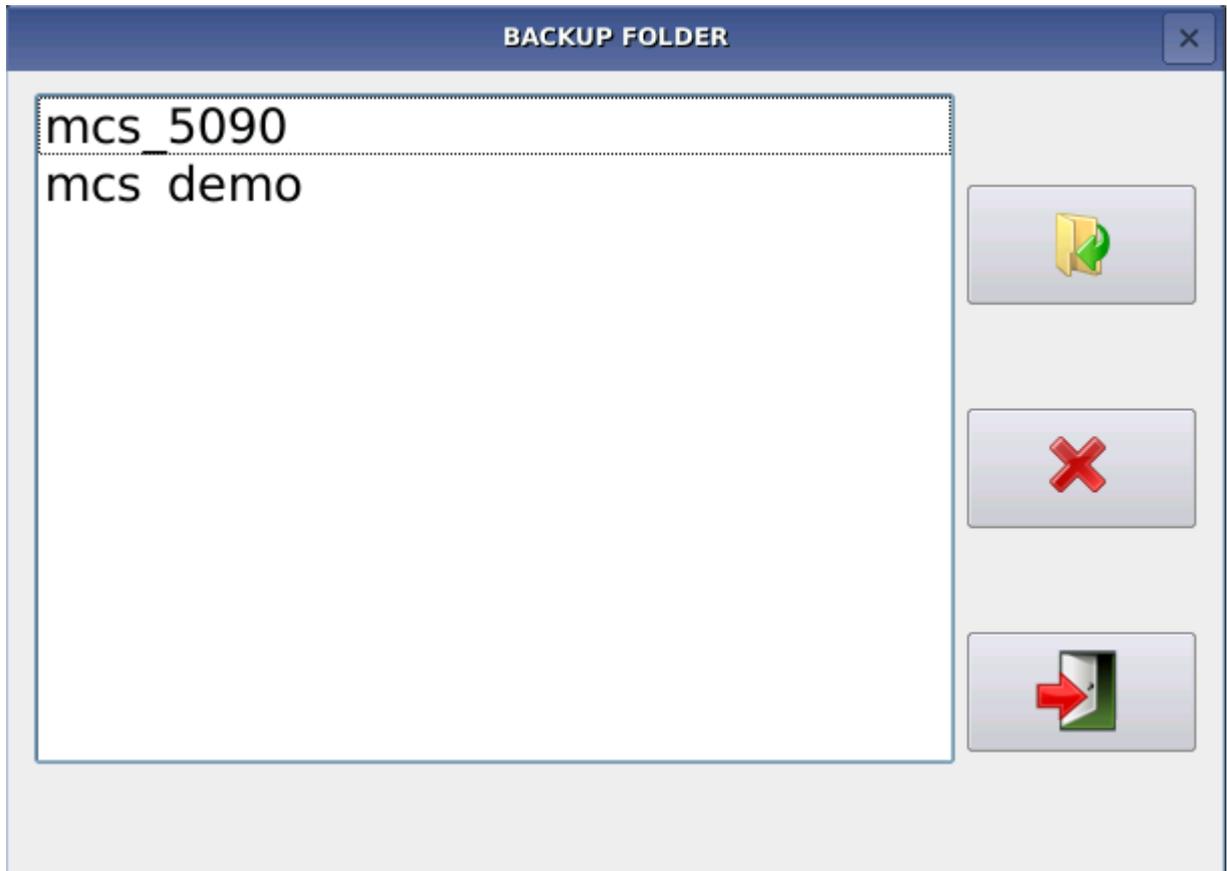
Backup name prompt window

NOTE

If a backup with the same name already exists on the stick, a request to replace the file will be displayed. To proceed with saving, confirm the request otherwise the backup will be cancelled.

- **Resetting of selected parameters:** The selected parameters will be loaded from the USB stick inserted in the controller and saved in the controller's memory.

When restoring a backup, a screen will appear where you can select the backups on the USB stick inserted in the controller.

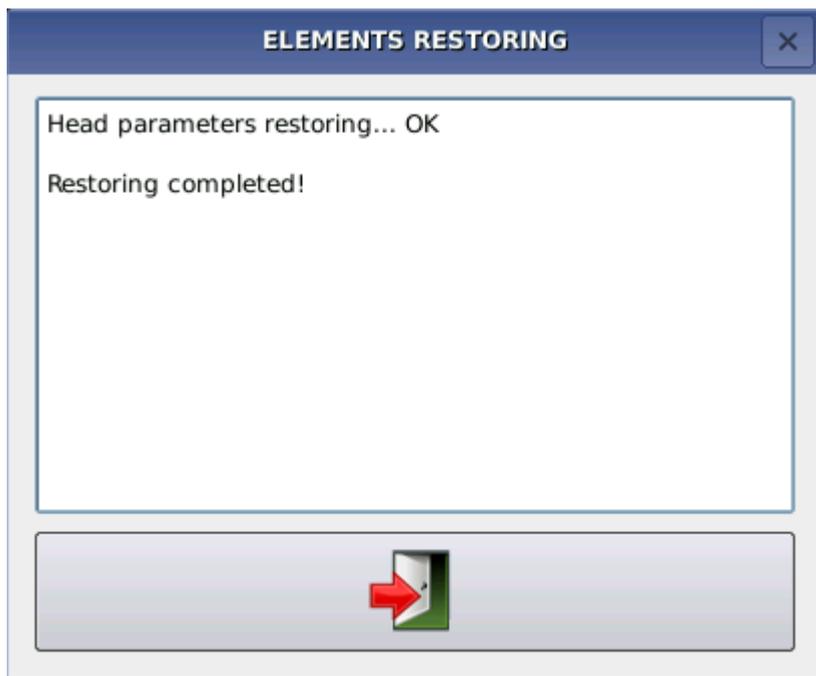


Backup selection screen

NOTICE

Restoring selected parameters will lose all data of those parameters currently present on the controller. Make sure you have a backup of those parameters before proceeding with the restore.

5. Once the operation to be performed has been selected, the software will display a report screen of the required operations with the relative success or failure of the operation indicated.



Example of backup report screen

The backup environment is also very useful for loading 'standard' data into the controller. Simply create a set of parameters, create the backup and load this backup onto the controller you want. In this way, the controller will already be set up with a working software configuration.

4.29 System Update

A very important operation in controller management is updating the software. Software, firmware and kernel updates may be required for various reasons (e.g. to add a new feature to the software) and it is necessary to know how to complete the update procedure correctly.

The system update is based on the loading of update files from the USB stick. The update files are as follows:

- **mcs** - file containing the software update;
- **ulimage** - file containing the kernel update;
- **FEAU100.dat** - file containing the firmware update;
- **escript** - folder containing scripts for updating the operating system.

To carry out the software update, it is necessary to have these files on the USB key that you will insert into the control.

In order to be able to update the system, you must be a *distributor* or *manufacturer* user, as the update operations that you will be performing are extremely delicate and require special care when updating.

To carry out the software update:

1. From the auxiliary functions screen, press the about button;



About button

2. The software version information screen will open;



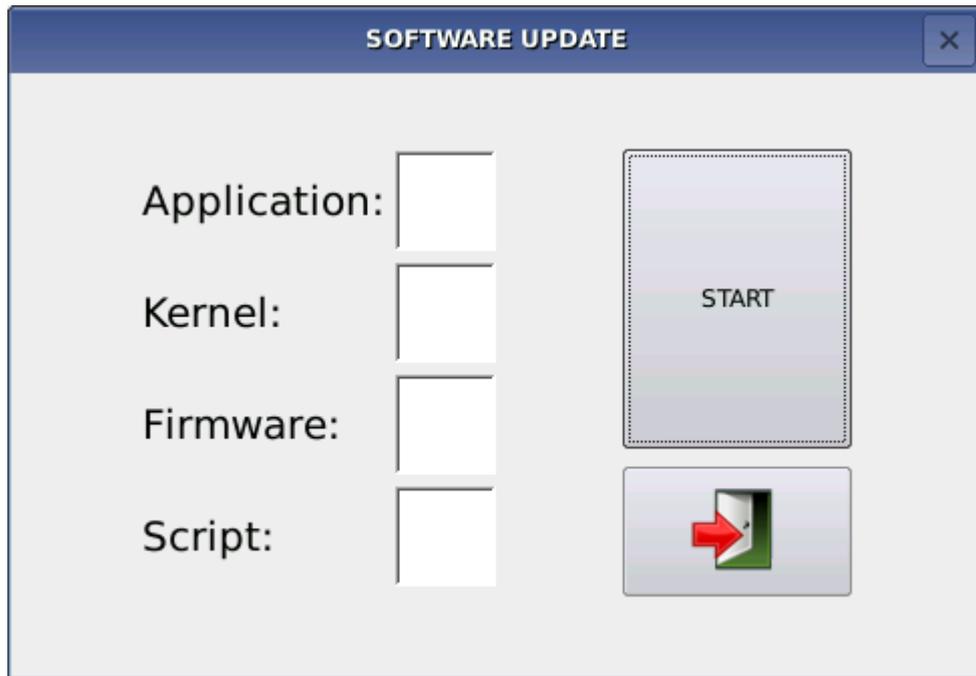
System Information Display Screen

Press the software update button to access the update screen.

NOTE

The update screen is protected by a password provided by the manufacturer. To update the software, enter the password indicated.

3. In the update screen, select the one you wish to update.



Update screen

Once the files to be updated have been selected, press the *start* button.

NOTICE

If you update the kernel, firmware or run an update script, you will be prompted to restart the system after the update is complete. In this case, the controller must be switched off and on again.

4.30 Display of free memory space

From the system information environment, it is also possible to view the space occupied in memory by programmes, DXF, TTF and settings in the controller. The display of free disk space is allowed at all user levels.

To carry out the software update:

1. From the auxiliary functions screen, press the about button;



About button

2. The software version information screen will open;



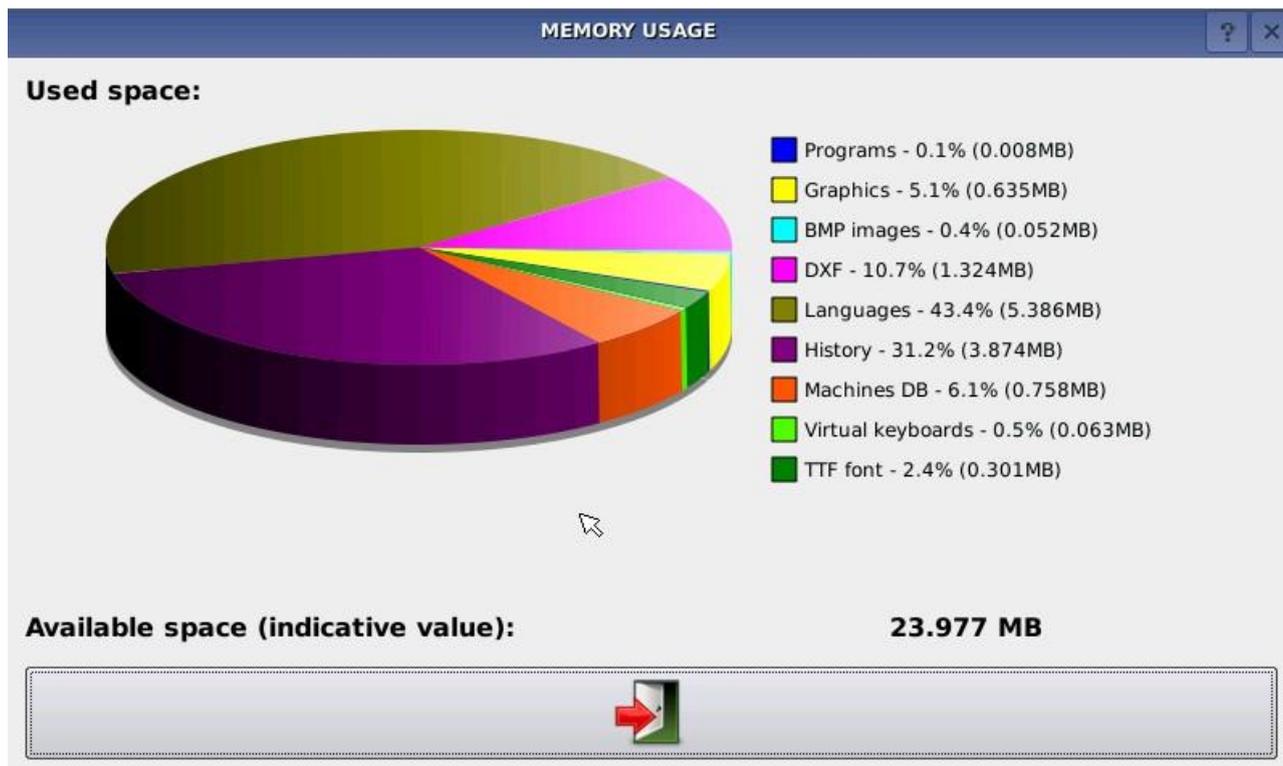
System Information Display Screen

3. Press on the memory management button to display information on the space occupied in the controller memory;



Memory occupancy button

4. A new window with the graph and memory occupancy percentages will appear on the screen;



Display screen showing occupied memory space

This screen will display all memory occupancy information by type (TTF fonts, custom fonts, programmes, images, history, etc.) and it will be possible to see which of these types is the most memory-intensive.

4.31 Management of external mouse and keyboard via USB

The software allows the management of USB mice and keyboards to be plugged into the appropriate USB port on the controller. The system also supports the presentation of a USB HUB so that a mouse and keyboard can be used simultaneously. To use the USB mouse and keyboard, simply connect them to the controller.

While the mouse does not require any configuration by the user, the keyboard needs to be configured with the appropriate mapping. To set up a mapping, it is necessary to go to the general software parameters and change the *USB Keyboard Mapping* parameter appropriately. See the general parameters section in this manual for further details.

4.32 Viewing Marking History

The software keeps a series of data in memory to display the markings. This history makes it possible to trace the markings made at a certain time.

It is possible to view the marking history from the auxiliary functions environment. The marking history is visible to all users at any operating level. Operations to reset the history are only accessible by a user with *manufacturer*, *distributor* or *supervisor* operating level.

In addition, it is possible to enable advanced history management, which allows the information of marked fields to be stored and subsequently displayed for each programme. Naturally, enabling this management involves a greater amount of data and therefore the use of the control unit's resources will increase.

To view the marking history:

1. From the auxiliary functions screen, press on the marking history button;



Marking history button

2. A screen will open where the history of markings made on the control unit will be displayed;

MARKINGS HISTORY			
Record visualized:			1 - 250
	DATE	HOUR	PROGRAM NAME
1	25/03/2011	10:16:09	TEST 5090
2	25/03/2011	10:11:15	TEST 5090
3	25/03/2011	10:19:13	test
4	25/03/2011	11:25:06	test
5	25/03/2011	11:25:38	test
6	25/03/2011	11:25:44	test
7	25/03/2011	11:26:06	test
8	25/03/2011	11:26:18	test
9	25/03/2011	11:26:24	test
10	25/03/2011	11:26:49	test
11	25/03/2011	11:26:56	test
12	25/03/2011	11:27:00	test

Marking history display screen

The marking history is subdivided into pages to facilitate reading of the data present. Information on the various records displayed can be found in the drop-down menu at the top right of the history table. To change the page of the displayed records, simply select a different page in the same drop-down menu.

To reset the marking history:

- press the  button;
- confirm the cancellation request.

In addition, setting the parameter *Maximum number of historical files* (see description of general system parameters) will enable the management of saving images of marked programmes. With this management enabled, it will be possible to save an image file (PNG) containing the complete preview of the programme just marked. In this way, it will be possible to trace the marked fields for each programme.

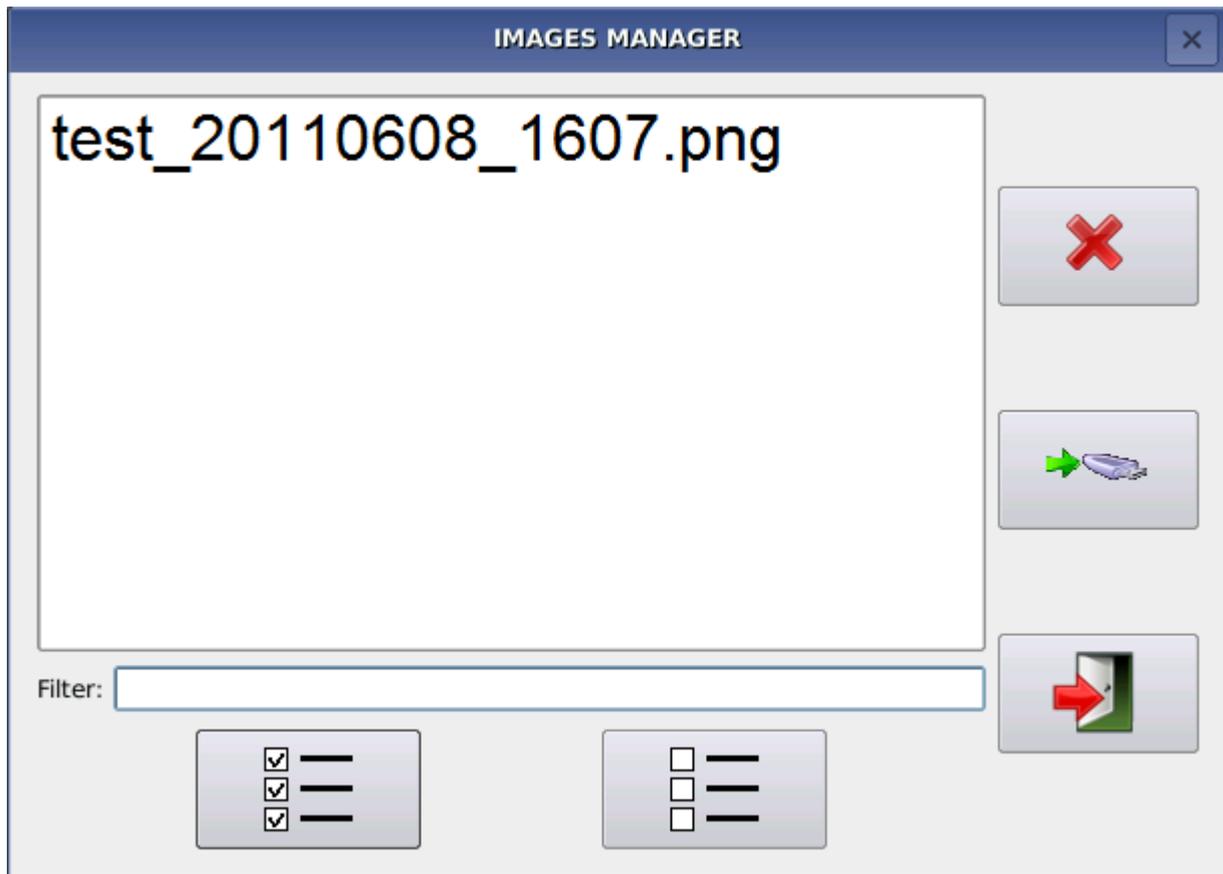
Marking history files are saved in the controller memory and can be copied to a USB stick from the history image management environment.

To go to the image management environment of the historian, simply press the button:



History image management button

In the image management environment, all images in the controller's memory will be reported.



History image management screen

Possible operations in this environment are deleting an image file or saving files to a USB stick.

The image files are saved in a way that identifies the date and time the programme was run:

<programme name>_<date>_<time>.png

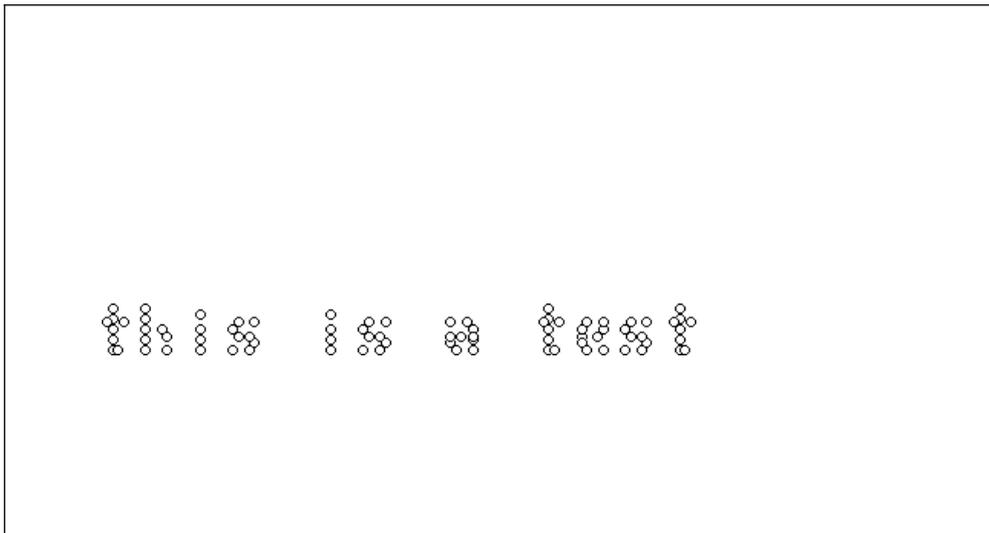
NOTICE

If the number of files in the history reaches the value of the *maximum historical file number* parameter will be displayed a message warning message will be displayed at the start of a new marking cycle. The message will be displayed u n t i l the history is cleared.

NOTICE

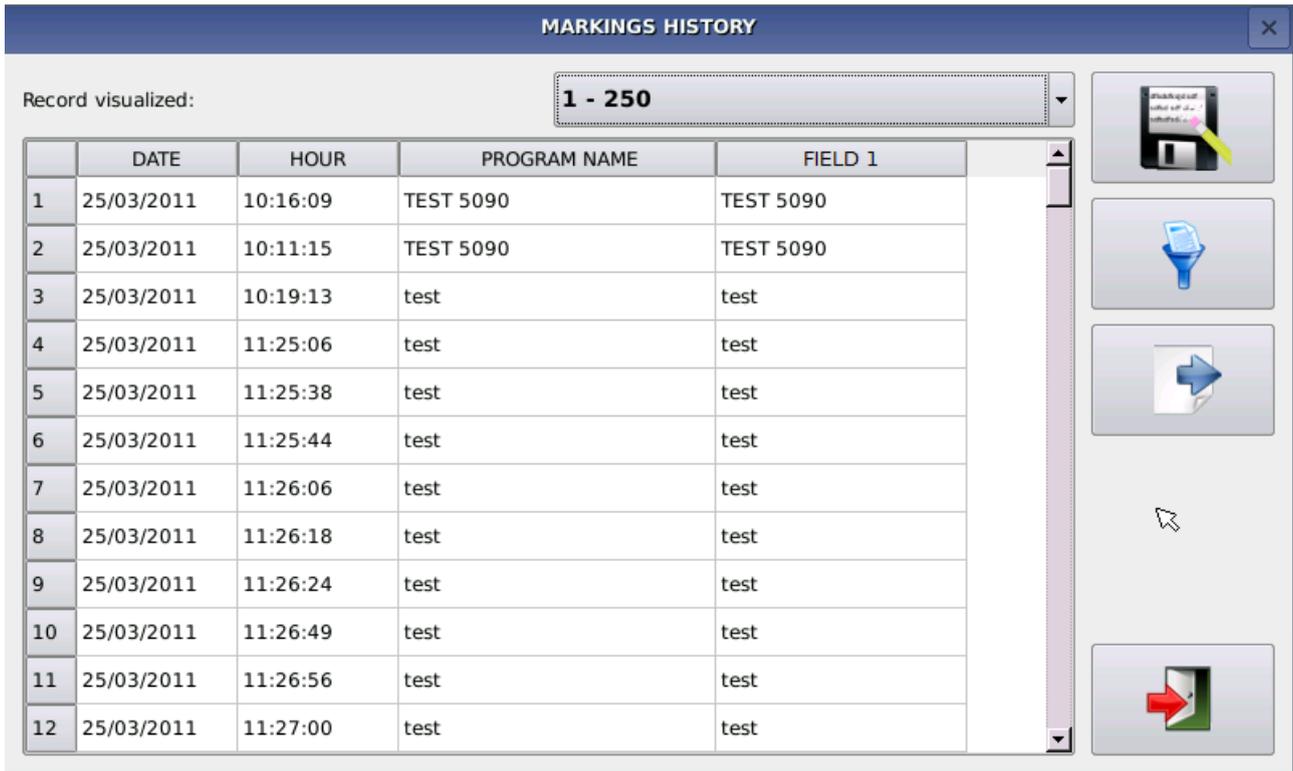
The *history file save type* parameter affects whether or not history files are saved. This is because if it is set to SINGLE FILE and the programme is run a second time after a history file has been saved, it will not be saved. See the general parameters section for more information.

Below is an example of an image saved for the marking history:



Example of an image file of the marking history

In the case of advanced marking history enabling, it will be possible to activate/deactivate and customise filters on the history data. It will also be possible to export the current history table in CSV format in order to have a report on the data stored in the control unit.



The screenshot shows a window titled "MARKINGS HISTORY" with a close button in the top right. Below the title bar, it says "Record visualized: 1 - 250". The main area contains a table with the following data:

	DATE	HOUR	PROGRAM NAME	FIELD 1
1	25/03/2011	10:16:09	TEST 5090	TEST 5090
2	25/03/2011	10:11:15	TEST 5090	TEST 5090
3	25/03/2011	10:19:13	test	test
4	25/03/2011	11:25:06	test	test
5	25/03/2011	11:25:38	test	test
6	25/03/2011	11:25:44	test	test
7	25/03/2011	11:26:06	test	test
8	25/03/2011	11:26:18	test	test
9	25/03/2011	11:26:24	test	test
10	25/03/2011	11:26:49	test	test
11	25/03/2011	11:26:56	test	test
12	25/03/2011	11:27:00	test	test

On the right side of the window, there is a vertical sidebar with several icons: a floppy disk, a funnel, a document with an arrow, a mouse cursor, and a red arrow pointing right.

Advanced marking history screen

To apply a filter to the history table, simply click on the filter configuration button:



Filter configuration button

An environment will be displayed on the screen allowing you to configure the filter to be applied to the table as you wish.



The screenshot shows a dialog box titled "FILTER" with a close button in the top right. It contains the following elements:

- A "Filter:" label followed by an empty text input field and a red eraser icon.
- A "Field:" label followed by a dropdown menu currently showing "[ALL FIELDS]".
- Two large buttons at the bottom: one with a green checkmark and one with a red 'X'.

Filter configuration screen

To configure the filter, you can enter a text that will be searched on the fields selected in the drop-down menu. It is possible to specify to apply the filter to all fields or to one in particular. The text that has been specified will be searched on the selected fields and if it is present, even if only in part, then it will be considered to be displayed.

For example: set to search for the text 'test' in all programme fields. This will include all programmes that contain the text 'test' in the programme name ('test marking', 'prog_test', etc.) or as a marked field.

It is also possible to search and filter all programmes that contain a particular compo flag (e.g. '%s').

Finally, it is possible to export the marking history in CSV format. The export will work on the current configuration of the history, with filters or without, and the result will be saved on USB with the name:

export_<data>_<time>.csv

To export a marking history, simply press the export button:



Export button

When the button is pressed, you will be asked whether to proceed with the operation or not. If you wish to continue you will need to confirm this request. For very large histories as stored data it may take some time to complete the operation. Once the export is complete, a message confirming that it has been saved to USB will be displayed.

NOTICE

Export of the history file will only be possible on a USB key. It will therefore be necessary to insert a USB key into the controller to carry out the export operation.

NOTE

The size of the data stored in the marking history affects the display and processing performance of the history. Filters also add extra weight to the display and processing performance.

If the functionality of loading programmes from a shared network folder is enabled, it will be possible to export the history directly to the shared folder.

Furthermore, if the automatic history export parameter is enabled, a file will automatically be generated within the shared folder containing the markings contained within the history.

NOTE

The automatic export file is automatically updated every 5 minutes.

4.33 Virtual keyboard management

The software also allows the management of customised virtual keyboards with dynamic and user-created mappings. The mapping of a virtual keyboard is based on a standard XML file from which you can add, modify or delete any character you wish to use on the AC500 control unit.

Of course, virtual keyboards can be used to edit text fields in marking programmes.

NOTE

Virtual keyboards use UTF-8 encoding and therefore the use of non-ASCII characters is permitted. However, if you enter text to be marked with non-ASCII characters, you will need to use either a TTF font or a custom font, as the proprietary fonts within the software only handle ASCII characters.

Virtual keyboards can be entered from the file management environment and can be set from the general data environment. Obviously, setting the general data parameter will change the display of the keyboards presented in the software (please note that some keyboards do not use custom mappings).



Example of virtual keyboard with Cyrillic mapping

4.34 Board versions

The AC500 controller has several hardware versions called Board A, Board B and Board C. The software automatically recognises which board is on the controller.

Board B

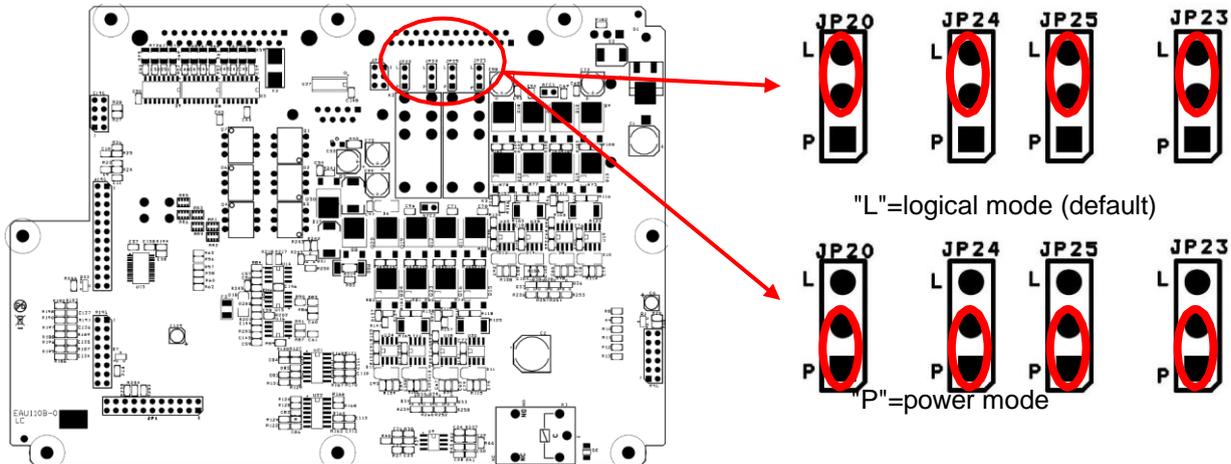
Board B is the board of the previous hardware revision in which numerous improvements for handling the axis control of the marking head and other important improvements for the use of the marking system (and its contour functions) have been reported.

The most interesting detail concerning this board is the control of the third axis, which can be done in logic (command to external drive) or in power (directly to the motor). While Board A can only take advantage of the control in COMMUTE of the third axis, Board B is configurable (both HW and SW) and to do this you must follow the steps below.

Configuration is via four jumpers located on the power board (EAU110B-0), near the head connector (plus a sw parameter).

JUMPER MODIFICATION PROCEDURE (BOARD B)

- disconnect the power supply
- remove the back cover and screws, and separate the two boards¹
- set the four jumpers to either logic ('L') or power ('P') mode, as in the figure



- reassemble the boards, screws and cover
- set the general 'Third Axis Type' parameter in the sw to 'STEP/DIR' (logic mode) or 'SWITCHED' (power mode)

Board B, as already mentioned, allows for greater functionality than Board A, and these are all listed in the manual (those specific to Board B are explicitly stated).

Board C

Board C is the latest revision board that incorporates further improvements and optimisations as well as the ability to control XY axes in logic mode. This particular configuration is necessary if you want to use a control unit to control a machine with a sliding head. As for the management of the third axis in logic already implemented in the previous Board B, special HW/SW configurations are required to use the XY axes in logic. These configurations are described in detail below.

JUMPER MODIFICATION PROCEDURE (BOARD C)

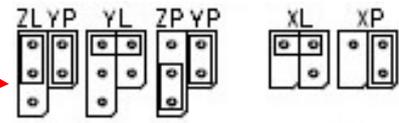
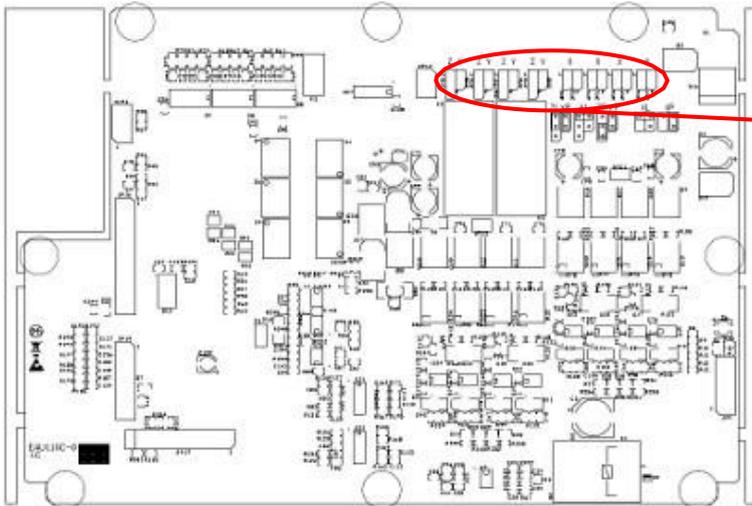
- disconnect the power supply
- remove the rear cover and screws, and separate the two boards² in the case of the first version of the board (EAU110C-0) which has internal jumpers
- set the jumpers according to the instructions on the board

¹ Unthread and re-thread the boards with attention to the connectors and handle the boards from the edges

² Unthread and re-thread the boards with attention to the connectors and handle the boards from the edges

INSTRUCTIONS FOR USE OF THE SOFTWARE

AUTOMATOR MARKING SYSTEMS SRL - Via Meucci n.8-20094 Corsico (MI) Italy
Tel 02 48601445 Fax 02 48601503 e-mail info@automator.com www.automator.com



As shown by the silkscreen print on the board, these are the possible jumper settings for the three axes.

In the case taken as an example, in order to configure the management of the smear head, it will be necessary to set the

NOTE: For the EAU110-1 board, it has been made possible to set the jumpers externally without having to dismantle the two controller boards. In this case, the jumpers should be inserted in the same way (i.e. inserting the jumpers in a mirror-image fashion from the rear of the board).

- reassemble the boards, screws and cover
- set the smear head configuration from the head magazine in the sw

Finally, it is possible to consult the system information environment to find out which Board is present on the control unit in use.



System Information Display Screen

NOTE

It is necessary to pay attention to the board mounted on an AC500 series control unit because different board versions need different firmware versions. Installing the wrong firmware on a board could cause the system to fail.

ATTENTION

Do not mount boards on motherboards with different revisions. A motherboard of revision A is not compatible with a B/C board, etc.

4.35 Printer settings configuration

Once printer management is enabled by parameter in the general data, it will be possible to access the configuration environment of the printer connected to the control unit via serial port.

NOTICE

If one of the fieldbus or modbus functionality is used, it will be impossible to use the printer. In this case it will be necessary to disable these functionalities to have a COM available

To access the printer configuration environment:

1. From the auxiliary functionality screen, press the system setup button;



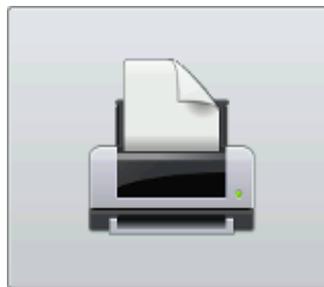
System setup button

2. A screen will open showing the possible system parameters that can be changed;



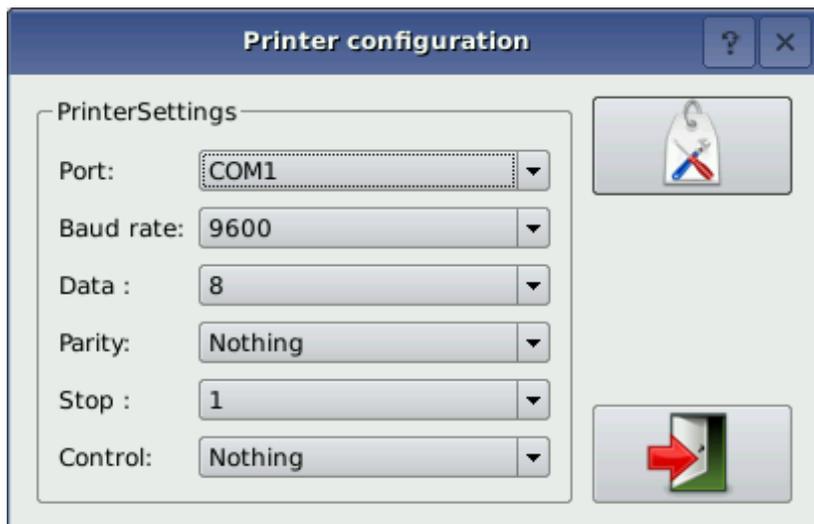
System parameter configuration screen

To access the printer configuration environment, simply press the environment access button.



Printer configuration button

3. The software will display an on-screen screen where all printer-related settings will be shown and the label configurator can be accessed:



Example of printer configuration screen

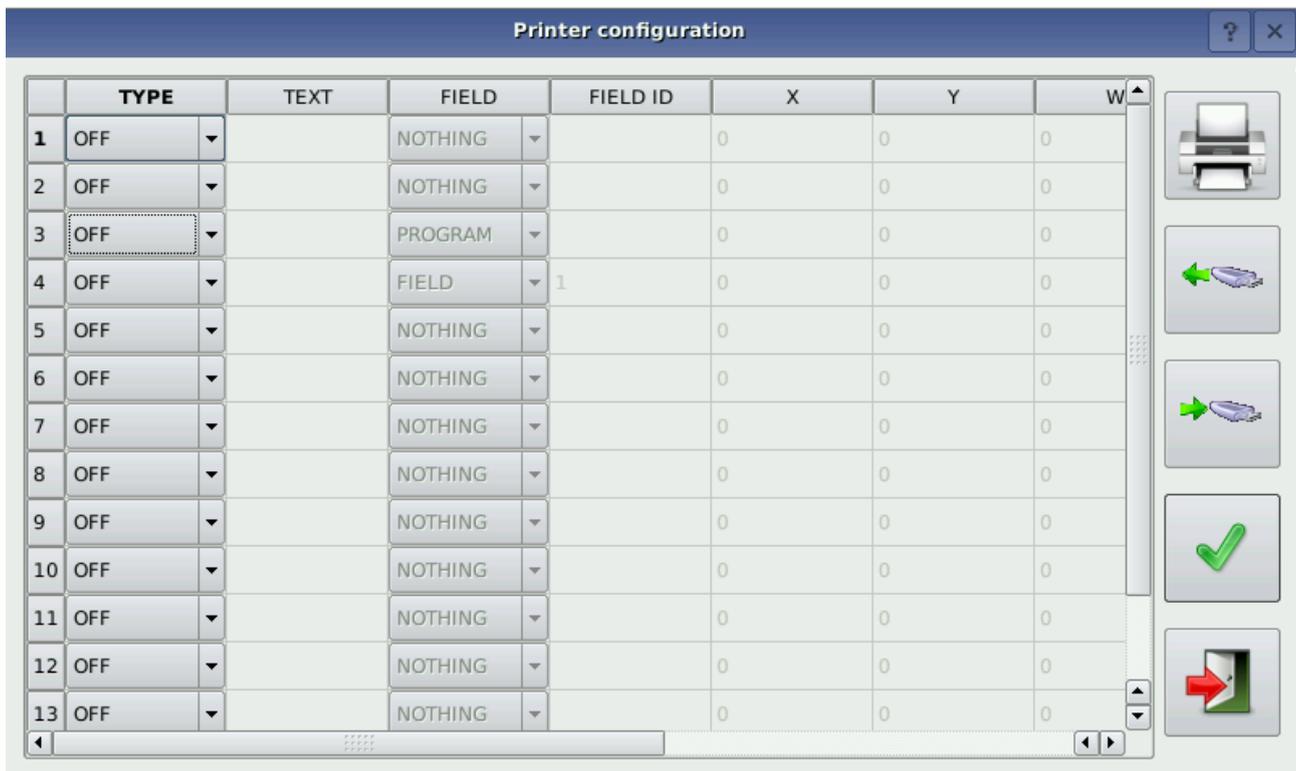
The left-hand side of the screen will display the settings for the connected serial port for the printer being used.

To access the label configurator, simply press the access button:



Configuration button for labels

After pressing the button, a screen for configuring label printing will be displayed:



Example of label configuration screen

The columns of the configurator table contain the parameters that must be set to manage the printing of the label.

- The first column indicates the *TYPE* of print to be made. You can choose between **TEXT**, **BARCODE UPCA**, **BARCODE AN13**, **BARCODE 128** and **IMAGE**.
- The second column contains the *TEXT* to be printed. If *TYPE* is set to *IMAGE* in this column, you will need to specify the name of the image in the printer's memory that you wish to load.
- The third column contains the *FIELD* to be printed. You can choose **PROGRAMME** to print the name of the marking programme or **IDCAMPO** by indicating in the next column the number of the field to be printed.
- The fourth column contains the number of the field to be printed and will only be enabled in the column after the **IDCAMPO** option has been selected.
- The fifth column contains the co-ordinate of the **X-axis**.
- The sixth column contains the co-ordinate of the **Y-axis**.
- The seventh column contains the value for the width of the text identified with **W**.
- The eighth column contains the value for the height of the text identified with **H**.
- The ninth column indicates the *ROTATION* of the text or barcode to be printed. You can choose between **0°**, **90°**, **180°** and **270°**.
- The tenth and final column indicates the type of *FONT* to be used for the text. You can choose between **STANDARD**, **REDACTED** and **BOLD**.

On the same screen it will be possible:

- Make a test print via the button
- Upload a configuration file via the
- Save a configuration file via the



CHAPTER 5

System Testing and Calibration

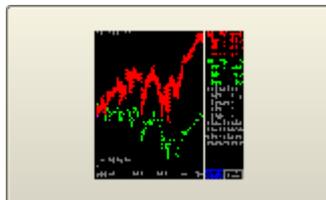
The software makes it possible to launch an axis calibration and test the system. Axis calibration is not an operation that must necessarily be called up by the operator, as calibration is performed by the software whenever the system is set in the online state. The system test, on the other hand, is a fundamental operation for fine-tuning the system, because in this environment it is possible to verify the operation of digital inputs and outputs, axis frequency outputs and machine parameters and connections. Therefore, the system test is always the first operation to be performed when mounting the controller on the machine.

5.1 System Testing

The test environment is used to analyse and check the basic operation of the controller connected to the machine. If the test does not work, it means that there are problems in the machine's connections and therefore it is necessary to check and fix possible problems in order to mark them successfully. The test can be used to test the digital inputs and outputs, the axis frequency outputs and the working area of the machine.

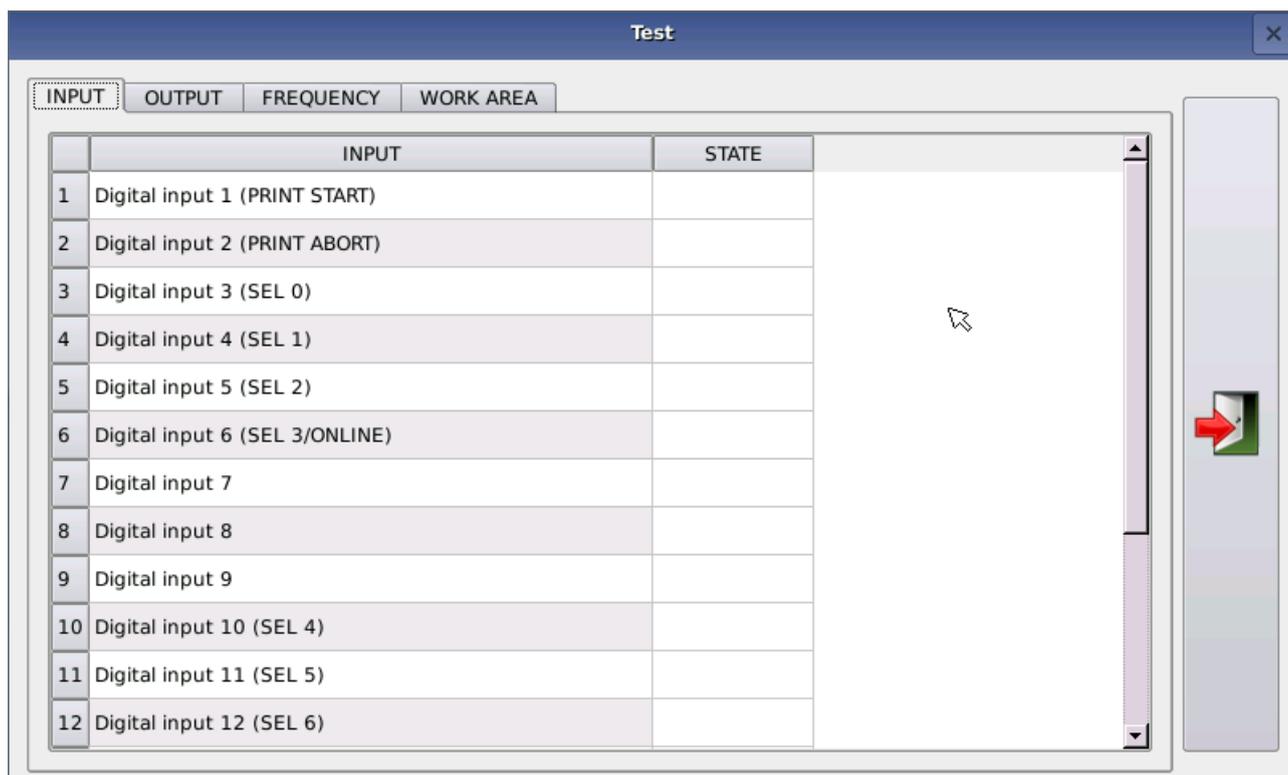
To test the system, the operator level must be set to *distributor* or *manufacturer*.

To enter the system test, simply press the *test* button on the auxiliary functionality screen.



Test button

The test screen appears as follows:



System test screen

The screen consists of a series of tabs, in which all elements required to test the machine are entered. The tabs in the test environment are:

- **Inputs**
From this board it is possible to test the operation of the machine's digital inputs.
- **Outputs**
From this board it is possible to test the operation of the machine's digital outputs.
- **Frequency**
From this tab you can set a frequency and check whether the selected axis is moving correctly.
- **Working area**
From this tab, it is possible to test the work area according to the set work area parameters.

5.1.1 Testing of digital inputs

In order to test the digital inputs, the digital inputs tab in the system test environment must be selected.

On the digital input test tab there is a table showing all the inputs managed by the software. Each row of the table corresponds to an input connected to the controller, and the rows consist of two columns, one for the description and one to display the current status of the digital input.

	INPUT	STATE
1	Digital input 1 (PRINT START)	
2	Digital input 2 (PRINT ABORT)	
3	Digital input 3 (SEL 0)	
4	Digital input 4 (SEL 1)	
5	Digital input 5 (SEL 2)	
6	Digital input 6 (SEL 3/ONLINE)	
7	Digital input 7	
8	Digital input 8	
9	Digital input 9	
10	Digital input 10 (SEL 4)	
11	Digital input 11 (SEL 5)	
12	Digital input 12 (SEL 6)	

Digital input display table

If the status cell is empty, it means that the current state of the input is closed, whereas if the status cell is coloured, it means that the current state of the input is open.



Current status of the input closed.



Current state of the open input.

5.1.2 Testing of digital outputs

In order to be able to test the digital outputs, it is necessary to select the digital outputs tab in the system test environment.

In the digital output test tab there is a table showing all outputs managed by the software. Each row of the table corresponds to an output connected to the controller, and the rows consist of two columns, one for the description and one to display the current status of the digital output.

	OUTPUT	STATE
1	Digital output 1 (DONE)	
2	Digital output 2 (READY)	
3	Digital output 3 (PAUSE)	
4	Digital output 4	
5	Digital output 5	
6	Digital output 6	
7	Electrovalve	
8	Green LED	
9	Red LED	
10	Fan	
11	Commutation relay	

Digital output display table

If the status cell is empty, it means that the current state of the output is closed, whereas if the status cell is coloured, it means that the current state of the output is open.



Current output status closed.



Current output status open.

5.1.3 Frequency output testing

In order to test the frequency outputs, the frequency output tab in the system test environment must be selected.

The frequency output test tab contains a number of parameters which, when set appropriately, allow testing whether the machine axes are moving correctly. It is possible to set a frequency value and the direction in which to move the selected axis. Axis movement depends on the parameters set in the *axis parameters*.

The image shows a software interface for testing frequency output. It consists of two identical sections, one for the X-axis and one for the Y-axis. Each section has a 'BACK' button (highlighted in green), a 'STOP' button, a minus sign button, a text input field containing the number '0', a plus sign button, and a 'FORWARD' button. The X-axis section is labeled 'X Axis' and the Y-axis section is labeled 'Y Axis'.

Frequency output test page

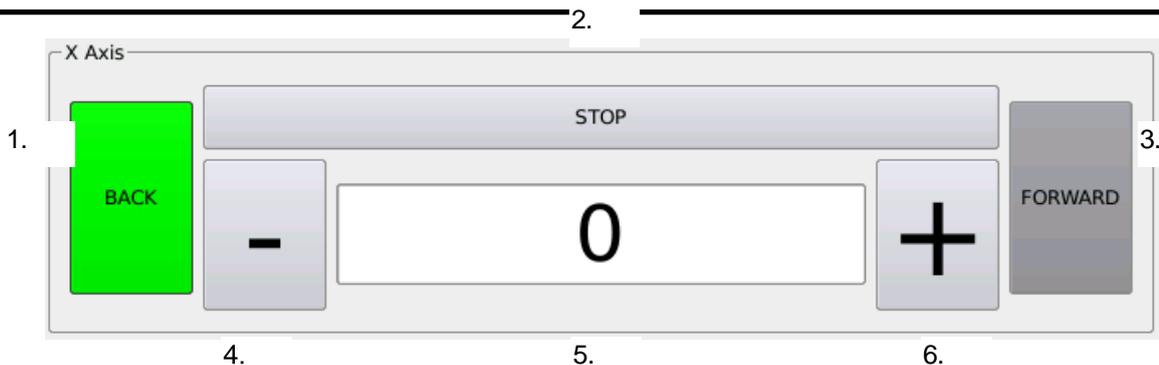
The forward and backward buttons can be used to select the axis movement direction and set the frequency with which the axis will move. Depending on the axis frequency set, it will be possible to move the X-axis, if a frequency is set in the X-axis text box, or the Y-axis, if a frequency is set in the Y-axis text box. To set a frequency, simply press the + and - buttons to increase or decrease the current frequency by one 100 Hz step. You can also set the frequency by pressing on the text box of the axis you wish to move. The axis will move in the selected direction.

NOTICE

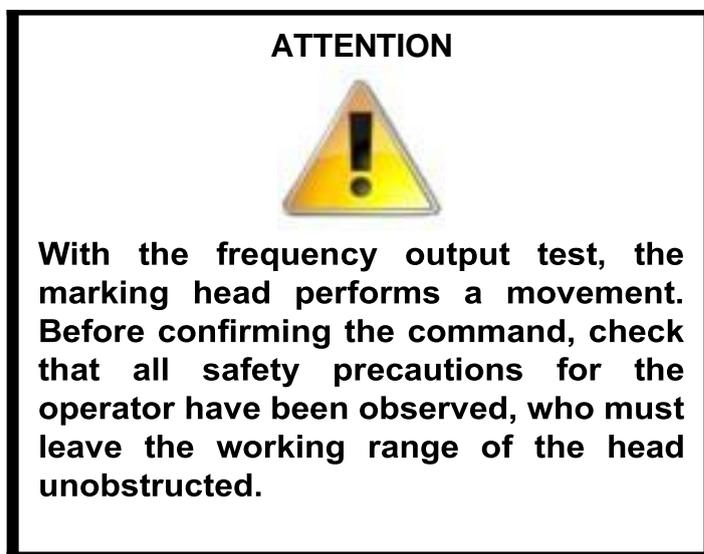
Before moving the axis, make sure that you have set the axis parameters correctly. The result of incorrect movement may be that the wrong data has been set.

NOTICE

In the test phase, the axes are not controlled by the software. Therefore, no checks are made on the possible positioning at maximum and minimum dimensions. Particular care is therefore required on the part of the user carrying out this operation, and it is recommended to stop movement before these dimensions are reached.

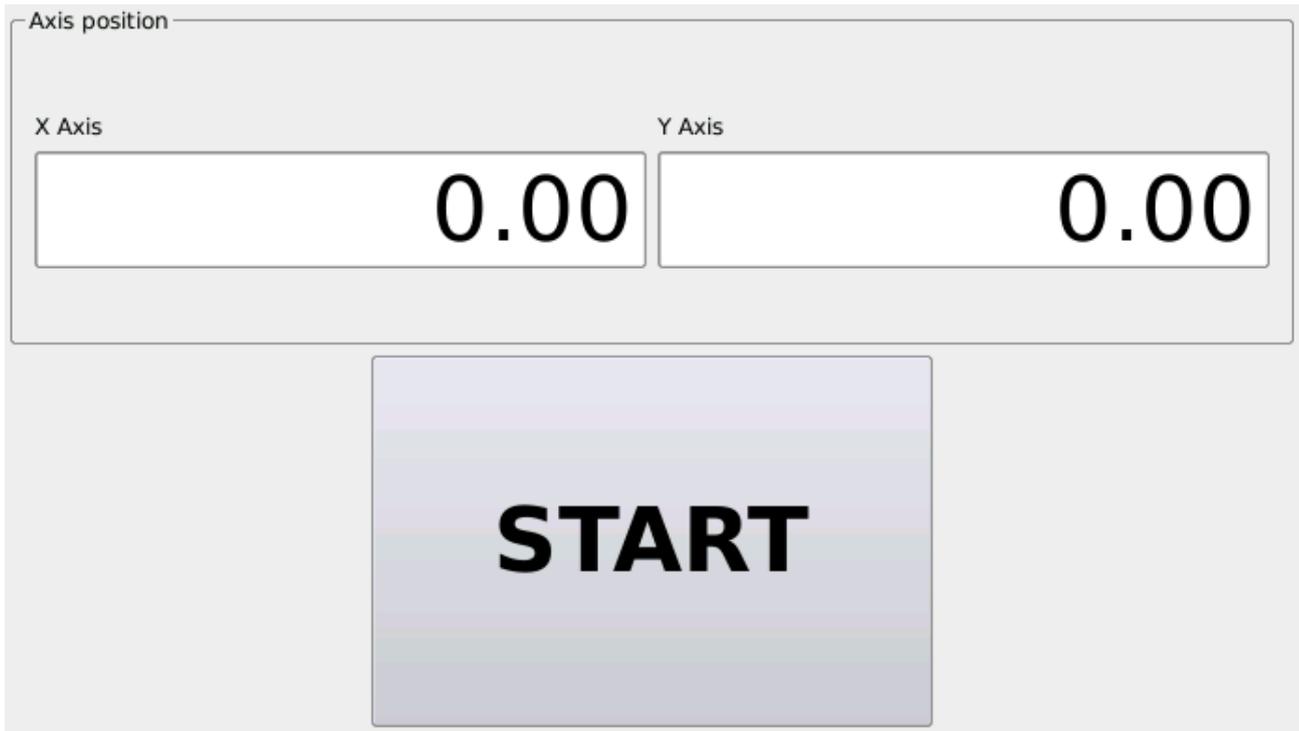


1. Backward movement selection button;
2. Movement stop button (resetting the commanded frequency value to zero);
3. Forward movement selection button;
4. Set frequency value decrease button;
5. Text box for displaying and setting the frequency to be controlled;
6. Set frequency value increment button.



5.1.4 Testing the work area

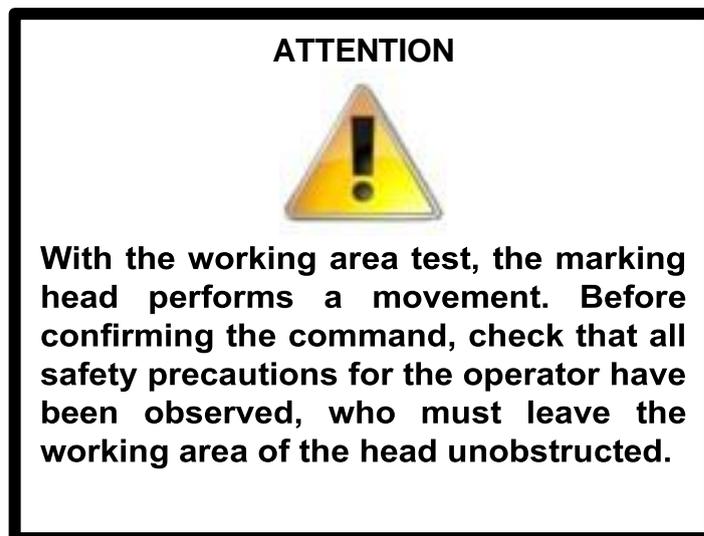
The work area test allows the limits of the machine's working area to be checked. The limits of the working area are defined according to the parameters set in the *general parameters*. In this test it will be possible to move the axes automatically to the 4 limits of the working area. The software will display the current machine dimensions.



Workspace test screen

To start the work area test, press the *start* button.

In this test, the axes will move at the speed set in the general parameter *work area test speed*. Make sure the parameter is set correctly before proceeding with the test.



5.2 Calibration of axes

Calibrating the axes is an operation that is carried out every time the online status is set. Therefore, as previously mentioned, it is an optional operation on the part of the operator. Axis calibration is required to calibrate the axes. Calibration is performed according to the set axis parameters.

During calibration, the axis will be moved according to the calibration direction set when searching for the limit switch. Once the end stop has been reached, the calibration altitude set in the parameters will be set as the axis' current altitude. If an axis is already on its limit switch at the time of calibration, the axis will be moved according to the calibration exit dimension parameter and then calibrated again.

To test the system, the operator level must be set to *distributor or manufacturer*.

To carry out axis calibration:

1. From the auxiliary functionality screen, press the axis calibration button;

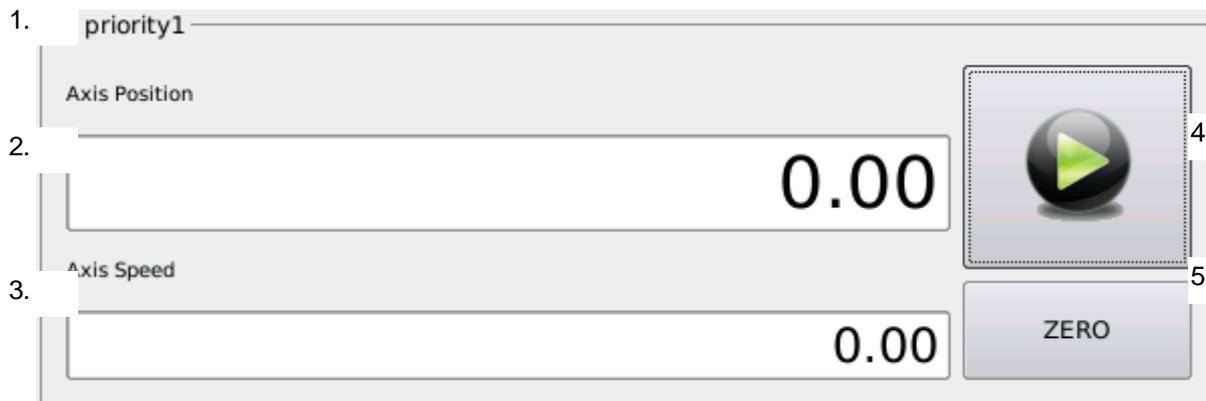


Axis calibration button

2. A screen will open where you can call up the axis calibration operation;

**Axis calibration screen**

From the calibration screen, it is possible to launch the calibration of a single axis or the complete machine calibration. The calibration environment displays the current dimension of each axis and the speed at which the axis is moving. It is also possible to interrupt calibration at any time.



- 1) Label displaying the reference axis and associated calibration priority;
- 2) Box displaying the current axis dimension;
- 3) Box displaying the current axis speed;
- 4) Button to launch axis calibration;
- 5) Push button to reset the current quota.



1.

1. Button to launch full machine calibration;
2. Button to interrupt a calibration, either axis or full calibration.



2.



ATTENTION



With axis calibration, the marking head performs a movement. Before confirming the command, check that all safety precautions for the operator have been observed, who must leave the working range of the head unobstructed.

CHAPTER 6

Connections and communication with the controller

This chapter will detail the various connection and communication systems with the AC500 controller. These systems allow the controller to be connected to an external terminal and the machine to be controlled directly from the terminal. The communication systems are:

- Logical signals
- Serial Communications

6.1 Remote Communications

The input/output control signals and communication hosts of the marking system allow the marking head and all marking operations to be controlled remotely.

The I/O port allows I/O signals to be transmitted between the controller and the remote I/O device. The I/O device can remotely select and programs to load and start or stop the marking cycle. Output signals from the marking system can be transmitted to the I/O device to give a status report. Please refer to the chapter on *logic signals* for further details.

The TTL I/O port allows you to start a marking operation using a Programmable *Logic Controller* (PLC) or by connecting a simple device to control the *start of marking*. Please refer to the *TTL signals* chapter for further details.

The serial port allows a host to be connected to the controller. Via RS-232 serial communication, it is possible to transmit data, select programmes to be marked and control the operations of the marking head. Host communication can also be RS-485 to control a network of several marking heads connected to the same host. Please refer to the chapter on *host communications* for further details.

6.2 I/O Signals

6.2a AC500+ I/O Signals

The AC500+ controller is designed to operate with 12 to 24 VDC PNP I/O signals. The I/O port on the back of the controller interface panel provides access to the marking cycle control circuitry. If you decide to use the I/O port, make sure:

- I/O signals are within limits;
- Use shielded cables;
- The protective shield of the cable is connected (either on the controller side, on the DSUB25 connector, or on the PLC side).

INPUT	LIMIT
Voltage	PNP: 12...24 Vdc
Current	5mA

Input signals

OUTPUT	LIMIT
Voltage	PNP 12...24 Vdc
Current	0.5A

Output signals

The I/O signals and their pin connections are described in the following table:

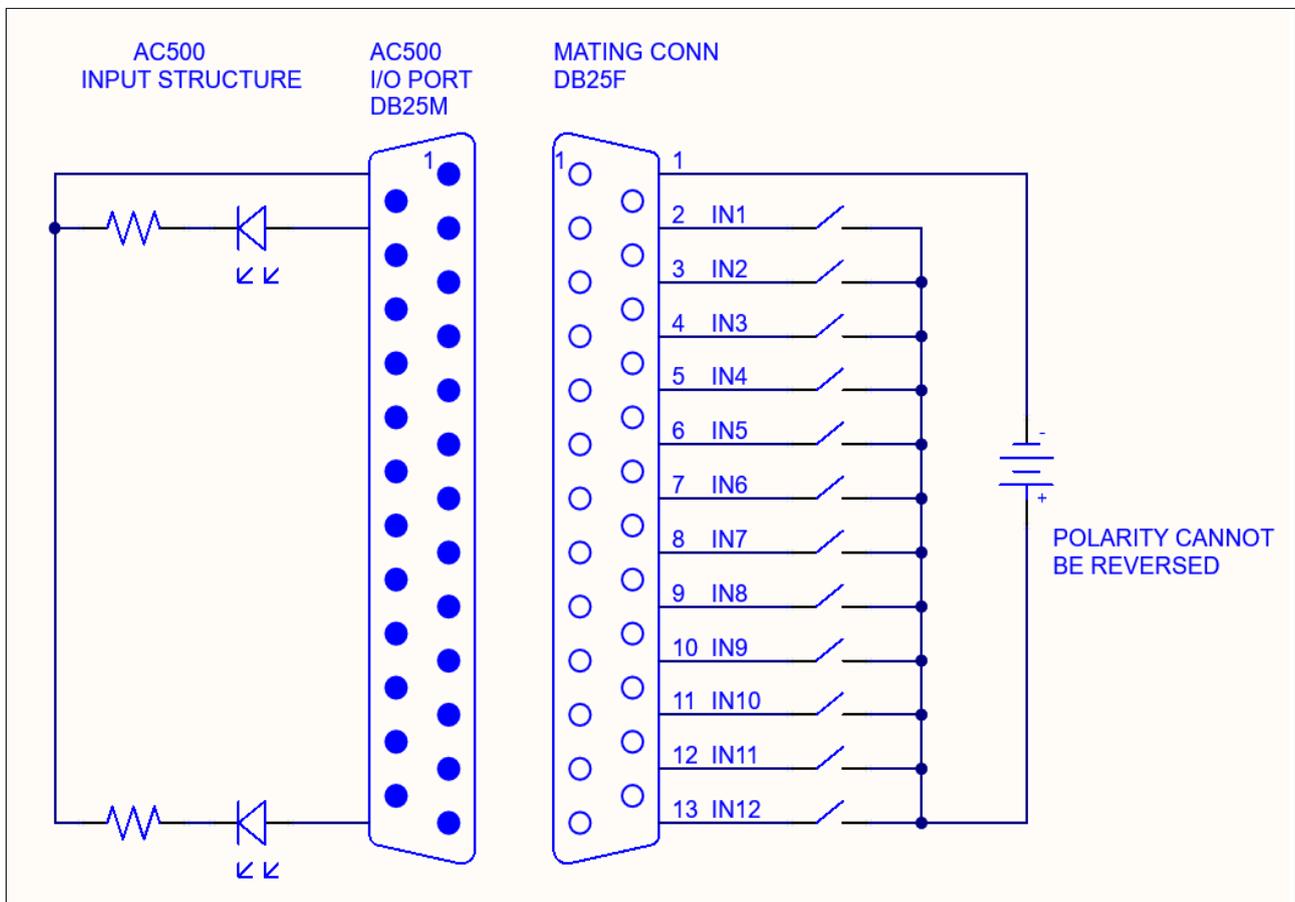
PIN	SIGNAL	DESCRIPTION
1	INPUT COMMON	Input common (positive for NPN input or negative for PNP input)
2	START PRINT	Input 1: Starts a marking cycle
3	ABORT	Input 2: stop a marking cycle
4	SEL_0	Input 3: see <i>remote programme selection</i> for details
5	SEL_1	Input 4: see <i>remote programme selection</i> for details
6	SEL_2	Input 5: see <i>remote programme selection</i> for details
7	SEL_3 / ONLINE ⁽¹⁾	Input 6: see <i>remote programme selection</i> for details
8	PAUSE	Input 7: pauses a marking cycle
9	Available	Input 8: available for future development
10	Available	Input 9: available for future development
11	SEL_4	Input 10: see <i>remote programme selection</i> for details
12	SEL_5	Input 11: see <i>remote programme selection</i> for details
13	SEL_6	Input 12: see <i>remote programme selection</i> for details
14	POS OUTPUT SUPPLY ⁽²⁾	Positive power input output
15	DONE	Output 1: marking cycle completed
16	READY	Output 2: ready to select a programme or start marking
17	PAUSE	Output 3: paused marking cycle
18	AUTOMATION 1 / ONLINE ⁽³⁾	Output 4: usable for automation
19	AUTOMATION 2 / ERROR ⁽⁴⁾	Output 5: usable for automation
20	AUTOMATION 3	Output 6: usable for automation
21	AUTOMATION 4 ⁽⁵⁾	Output 7: usable for automation
22	AUTOMATION 5	Output 8: usable for automation
23	AUTOMATION 6	Output 9: usable for automation
24	AUTOMATION 7	Output 10: usable for automation
25	NEG OUTPUT SUPPLY ⁽²⁾	Negative power output

I/O port connector pin table

Notes

1. The signal can be configured for message selection or for placing the marking head online. Please refer to the chapter on *general parameters* for details.
2. Unlike the first AC500 version, the AC500+ control unit requires positive and negative output power (PNP) on pins 14 and 25 respectively. This connection is also backward compatible with the first AC500 version with PNP-type digital outputs.
3. The signal can be configured as an automation output or to indicate the status of the ONLINE controller. Please refer to the chapter on *general parameters* for details.
4. The signal can be configured as an automation output or to indicate the status of an error controller. Please refer to the chapter on *general parameters* for details.
5. The new AC500+ controller version has ten digital outputs (unlike the first AC500 version, which had only six)

The following figure illustrates a simple closing contact for input signals using a customer-supplied DC power supply.

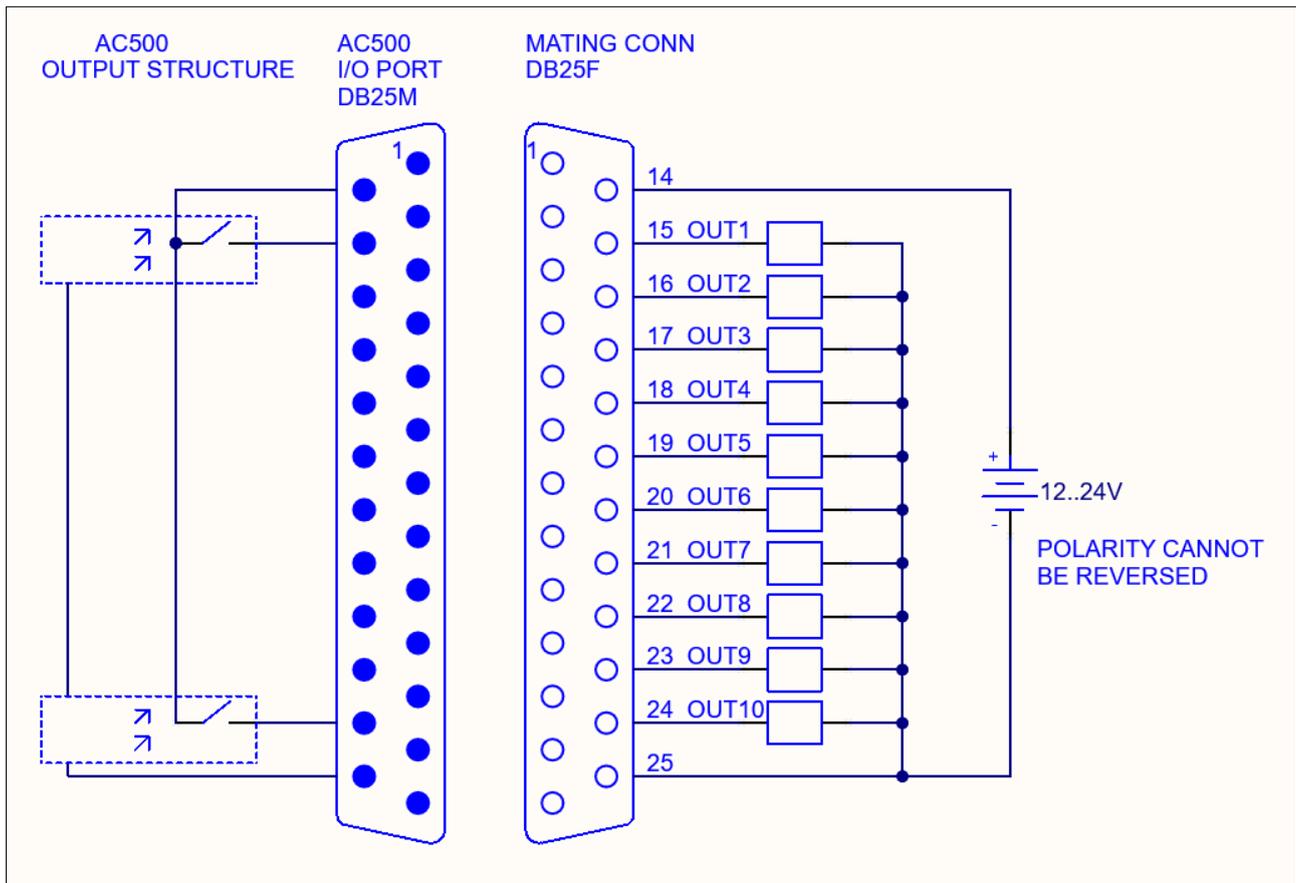


VDC input connections

The following figure shows the connections for output signals using a customer-supplied DC power supply.

In contrast to the previous version, the new AC500+ controller only provides PNP outputs, and requires that both the positive (on pin 1) and negative (on pin 25) of the output bank be brought to the connector.

(The addition of the negative output power pole on pin 25 compared to the previous model AC500 is however backward compatible with the previous model).



Note: In the case of replacing an old AC500 control unit with a new AC500+, in a system with also I/O, please ensure that

- the outputs are PNP type (or modify and adapt them)
- add the negative supply voltage connection of the outputs
 - pin 25 (this modification is backward compatible with older AC500 control units)

6.2b AC500 I/O Signals

The AC500 controller (first version) is originally configured to operate with 12 to 24 VDC I/O signals. The I/O port on the back of the controller interface panel provides access to the marking cycle control circuitry. If you decide to use the I/O port, make sure:

- I/O signals are within limits;
- Use shielded cables;
- The protective shield of the cable is limited (either to the DB15S connector of the controller or to the source, not both).

INPUT	LIMIT
Voltage	PNP/NPN: 12...24 Vdc
Current	5mA

Input signals

OUTPUT	LIMIT
Current	0.25A
Voltage	12...24 Vac/dac

Output signals

The I/O signals and their pin connections are described in the following table:

PIN	SIGNAL	DESCRIPTION
1	INPUT COMM	For all inputs (+ or -)
2	START PRINT	Input: starts a marking cycle
3	ABORT	Input: stop a marking cycle
4	SEL_0	Input: see <i>remote programme selection</i> for details
5	SEL_1	Input: see <i>remote programme selection</i> for details
6	SEL_2	Input: see <i>remote programme selection</i> for details
7	SEL_3 ⁽¹⁾	Input: see <i>remote programme selection</i> for details
8	PAUSE	Input: pauses a marking cycle
9	LASER SYSTEM OK	Input: laser input OK (laser machine only)
10	LASER ARMED	Input: armed laser input (laser machine only)
11	SEL_4	Input: see <i>remote programme selection</i> for details
12	SEL_5	Input: see <i>remote programme selection</i> for details
13	SEL_6	Input: see <i>remote programme selection</i> for details
14	OUTPUT COMM	For all outputs (+ or -)
15	DONE	Output: marking cycle completed
16	READY	Output: ready to select a programme or start marking
17	PAUSE	Output: paused marking cycle
18	AUTOMATION 1 / ONLINE ⁽²⁾	Output: usable for automation
19	LASER ENABLE / AUTOMATION 2 / ERROR ⁽³⁾	Output: laser enabling (laser machine only)
20	LASER POINTER / AUTOMATION 3 ⁽⁴⁾	Output: enable laser pointer (laser machine only)
21	do not connect	
22	do not connect	
23	do not connect	
24	do not connect	
25	do not connect	

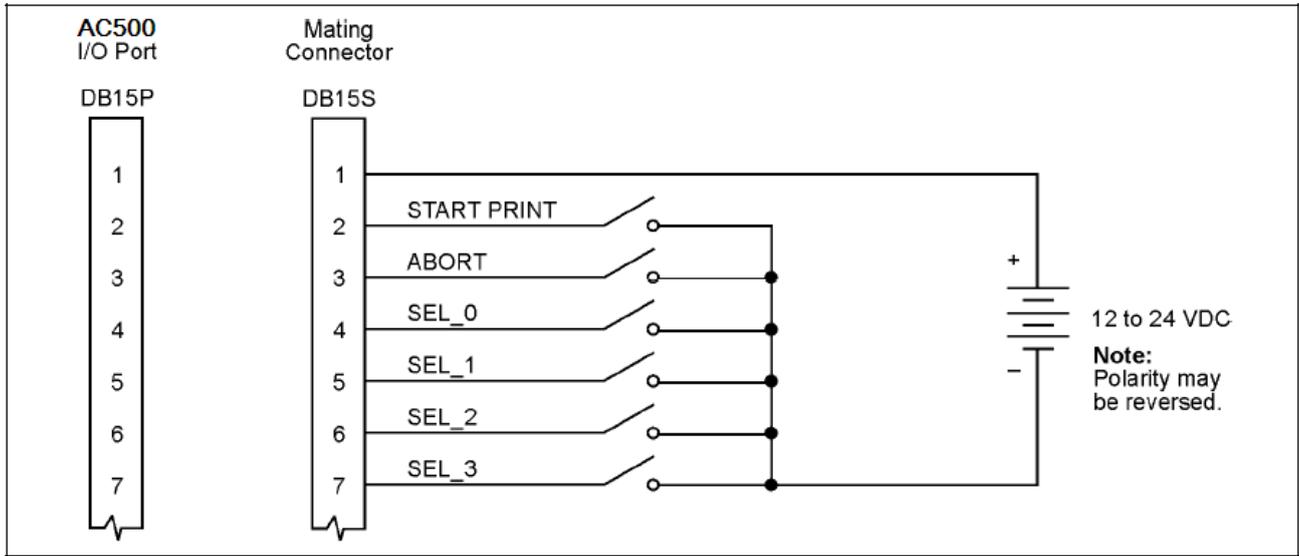
I/O port connector pin table

Notes

1. The SEL_3 signal can be configured for message selection or for placing the marking head online. Please refer to the chapter on *general parameters* for details.
2. The signal can be configured as an automation output or to indicate the status of the ONLINE controller. Please refer to the chapter on *general parameters* for details.
3. In the case of a NON-laser machine, the signal can be configured as an automation output or to indicate the status of an error controller. Please refer to the chapter on *general parameters* for details.
4. In the case of a NON-laser machine, the signal can be used as an automation output.

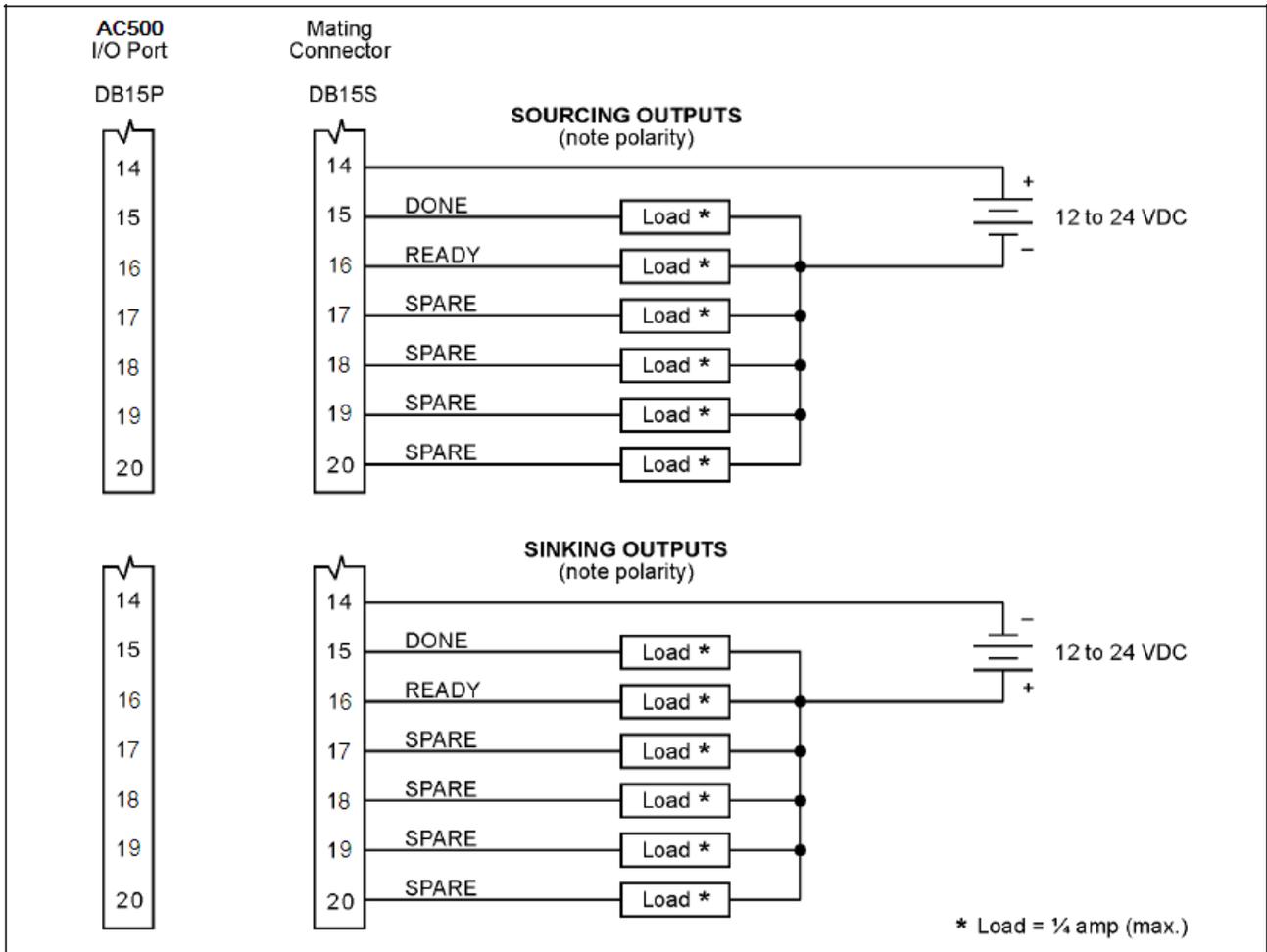
INSTRUCTIONS FOR USE OF THE SOFTWARE

The following figure illustrates a simple closing contact for input signals using a customer-supplied DC power supply.



VDC input connections

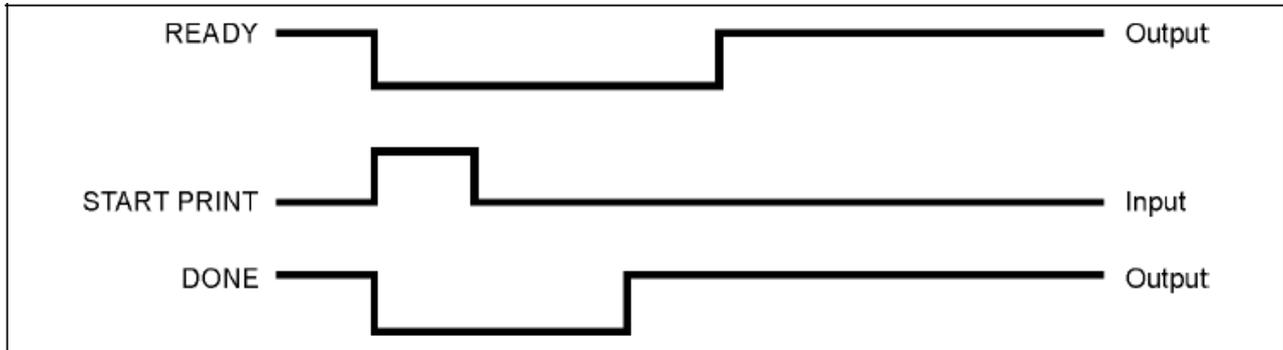
The following figure shows the connections for the output signals using a customer-supplied DC power supply. To demonstrate the difference in polarity, the drawing shows both the *sourcing* of signals and the output *sinking*.



VDC output connections

6.2 c Timing

The timing of the three head control signals (START PRINT, READY and DONE) are illustrated in the following figure:



Input/Output signal timing diagram

READY and *DONE* are used to determine when the head is ready for the next command. The head will only recognise the *START PRINT* command when *READY* and *DONE* are both on.

In case of simulation marking ENABLED, the *READY* signal will be switched off.

When the head receives a *START PRINT* command, both *READY* and *DONE* are switched off (off).

DONE will become active again after the marking has been completed.

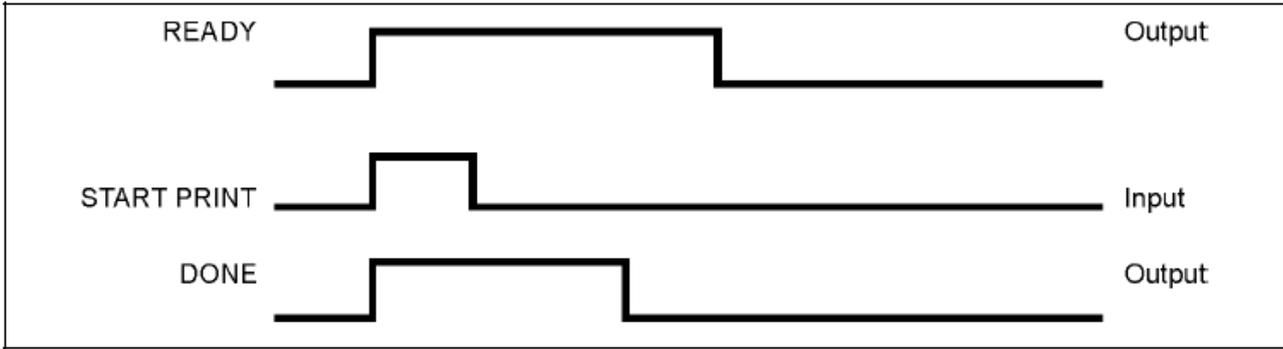
READY will become active again after the head has returned to the park position, provided that any parameters, e.g. serial numbering, are within the set limits.

If the marking cycle is aborted (ABORT), *DONE* will be active (on), but *READY* will be inactive (off).

The *PAUSE* signal is used to indicate whether marking is in a pause state. Normally the signal is deactivated (off) and is activated (on) whenever the marking cycle enters a pause state. The marking pause state can be due to a user-requested pause or the presence of a pause field within the programme.

If the management of the *DONE* and *READY* signals is turned around, see description of general software parameters, the operating logic will be completely inverted, i.e. when a signal is turned off (off) it will instead be on (on) and vice versa.

INSTRUCTIONS FOR USE OF THE SOFTWARE



Timing diagram of Input/Output signals (READY and DONE with inverted logic)

VDC output connections**6.3 Remote programme selection**

Remote programme selection allows the system to monitor decoded input signals transmitted by a customer-supplied I/O device. These signals (SEL_0 to SEL_6) are received by the I/O controller connector. The ON/OFF combination of the seven input signals generate binary decimal codes (BCD) in a range from 0000000 to 1111111. Each BCD value corresponds to a specific programme name stored on the controller. When remote input signals are received, the system interprets the BCD value of the seven input signals and loads the corresponding programme.

The SEL_3 signal can be configured for remote programme selection or to place the marking head online. If configured to place the head online, its ON/OFF status will be ignored during programme selection. If the marking head is already in the online state but is not in the marking cycle and the signal ON state is read, the marking head will restart the homing procedure. Please refer to the *general parameters* chapter for further information.

If the SEL_3 is set as online for the marking head, it will only be possible to select 63 programmes with remote selection. Obviously in this case, SEL_4 will behave like SEL_3, SEL_5 like SEL_4 and so on.

Reserved programme names correspond to 127 (or 63 in the case of SEL_3 configured as online) possible BCD values. **To ensure correct programme selection, programmes stored in the controller and available for remote selection must be identified with specific names.**

NOTE

If all seven input signals are OFF, the resulting BCD value is 0000. This effectively disables remote programme selection so that programmes can only be loaded by the operator from the controller.

For full details on possible SEL combinations, please refer to the table in Appendix C of this manual.

6.4 Control signal TTL Input

Input signals for TTL I/O control are handled in the motor tray connector on the controller's rear panel. The signals are prepared for use with optional Automator devices, such as the remote Start/Stop button.

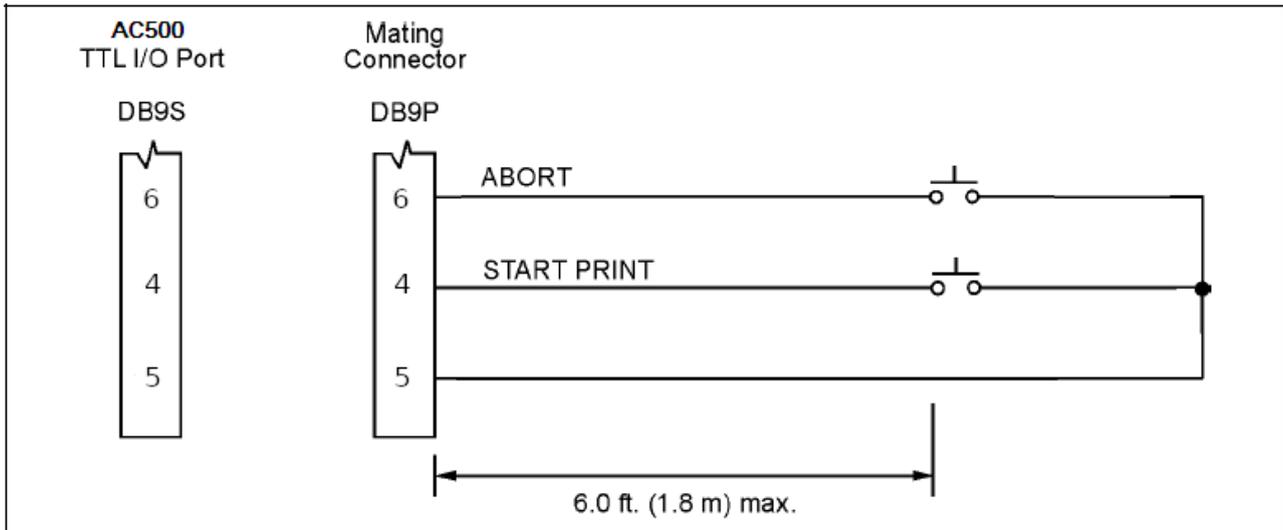
INPUT	LIMIT
Voltage	5 V
Current	5mA

The TTL input signals and their pin connections are described in the following table:

PIN	SIGNAL	DESCRIPTION
1	MOTOX A1	<i>Private</i>
2	MOTOX A2	<i>Private</i>
3	MOTOX B1	<i>Private</i>
4	MOTOX B2	<i>Private</i>
5	MOTOY A1	<i>Private</i>
6	MOTOY A2	<i>Private</i>
7	MOTOY B1	<i>Private</i>
8	MOTOY B2	<i>Private</i>
9	MOTOZ A1	<i>Private</i>
10	MOTOZ A2	<i>Private</i>
11	MOTOZ B1	<i>Private</i>
12	MOTOZ B2	<i>Private</i>
13	48Vout	<i>Private</i>
14	EV	<i>Private</i>
15	PWM	<i>Private</i>
16	AO	<i>Private</i>
17	START	Input 1: Start marking cycle
18	ABORT	Input 2: Abort/Interrupt marking cycle
19	0Vdc	<i>Private</i>
20	5Vout	<i>Private</i>
21	FC X	<i>Private</i>
22	FC Y	<i>Private</i>
23	FC Z	<i>Private</i>
24	Power supply output 15Vdc/100mA on AC500-LS version	<i>Private</i>
25	GND	<i>Private</i>

Motor port connector with TTL inputs

The next figure shows a simple contact closure for START PRINT and ABORT using TTL I/O signals.

**TTL Input Signal Connections**

6.5 Host Communication

The serial port on the interface panel behind the controller is capable of operating as either RS-232 or RS-485. The type of hardware (232 or 485) and software protocol of the marking system must match the serial device with which it is communicating.

The system software allows the parameters of the test to be set so that it communicates with the connected host. For maximum flexibility, the software works with a communication protocol called *extended protocol*. The protocol is described below.

The RS-232 interface is often used with remote devices, such as host computers, terminals or barcode scanners. The RS-232 interface can be used with the *extended protocol*.

The RS-485 interface is normally used when communication covers a long distance or when the marking head is part of a network. The RS-485 interface is required if several marking heads are to be controlled from one host. When using the RS-485 communication interface, use of the *extended protocol* is required.

The host port connections are described in the following table.

On AC500 control units with Board B or higher revision, there will be two additional TTL signals (like those on the motor port) which will allow you to command a cycle START or an ABORT of a cycle in progress. Obviously on a Board A these signals are not handled and will be considered as n.c. in the pinout.

PIN	SIGNAL	DESCRIPTION
1	n.c.	
2	RS-232 - RX	RS-232 data to controller
3	RS-232 - TX	RS-232 given by the controller
4	START TTL	Input 1: Start marking cycle (Board B or higher only)
5	0Vdc	
6	ABORT TTL	Input 2: Abort/Interrupt marking cycle (only Board B or higher)
7	RS-485 - A	
8	RS-485 - B	
9	5Vout	
Carcass	GND	Earth

Serial port connector table

6.6 RS-232 interface

The RS-232 interface is often used with remote devices, such as host computers, terminals or barcode scanners. The RS-232 interface can use the *extended protocol*.

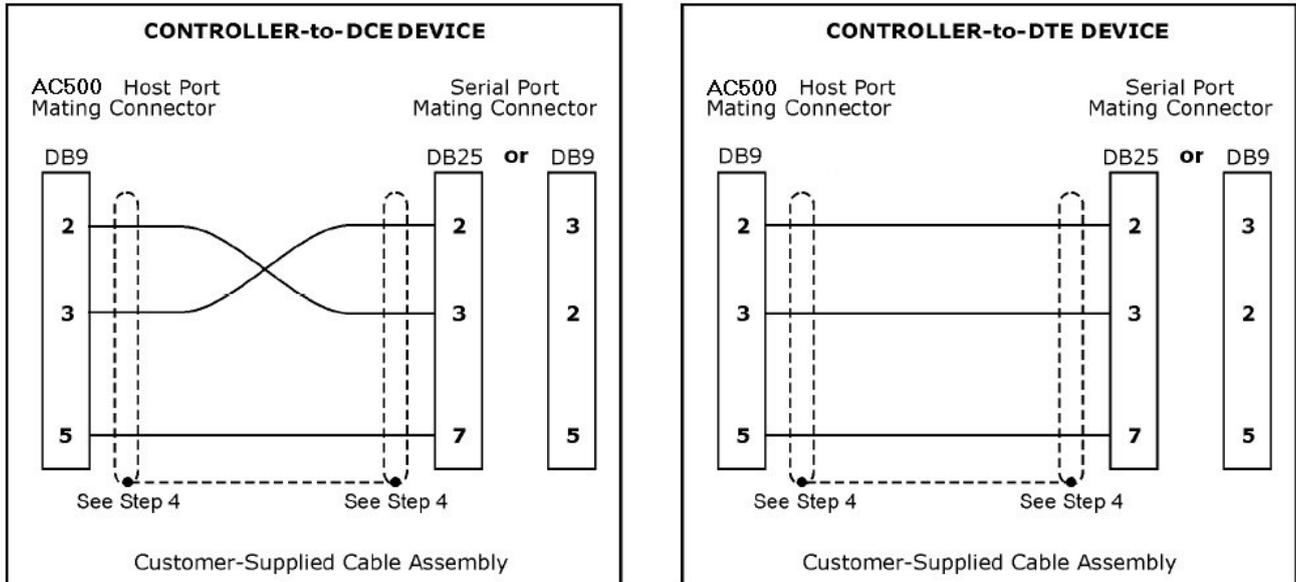
For connections to DTE devices (see figure below, left), you get a standard RS-232 PC-to-PC cable with a DB25S connector and a connector suitable for mating with the PC's serial port.

If you want to create an interface cable, make sure that the cable is constructed according to the following criteria:

- Use a shielded cable.
- Use connectors with black metal shells or black soldered shells.
- Solder the cable as follows.

The system supports XON/XOFF handshaking. When the system is unable to accept serial transmissions, the character XOFF is transmitted. When the serial port is available, the character XON is transmitted.

Similarly, if the serial port receives an XOFF signal during transmission, transmission is suspended until the XON character is received.



RS-232 cable diagram

NOTICE

It is necessary not to send data on the controller's RS-232 during the first 3 seconds after switching on the device. It is generally recommended to start the data exchange when the application is launched.

6.7 RS-485 interface

The RS-485 interface is normally used when there is long-range communication or when several marking heads are connected to the same host in a network. The RS-485 interface is required in applications with marking heads in a network. When using the RS-485 interface, use of the *extended protocol* for communication is required.

If the RS-485 interface is to be used, it is necessary to ensure that:

- Shielded cables are used.
- The cable shielding is terminated (either at the DB9 connector at the controller or at the host, but not both).

When heads are used in a network, each must be assigned a unique station ID number. The station ID number is used by the communication protocol to address messages to a specific head.

6.8 Extended Protocol

The extended protocol offers two-way communication with error checking. It is designed to guarantee security in the transmission of communication with an *intelligent* host device, using predefined message formats and response formats for typical messages. It also provides handshaking to reveal errors in transmitted messages, effectively verifying that the data was received correctly.

The extended protocol offers the possibility of network communications so that several heads can be connected together to a single host. All communications are conducted in a master/slave relationship in which only the host has the ability to initiate a communication. The head only communicates via a reply.

The following paragraphs explain the formats of messages and replies, model messages and Block Check Code (BCC) calculations.

Message format

Messages from the host computer to the marking head are transmitted in the following format:

SOH TYPE [##] STX [DATA_TEXT] ETX [BCC] CR

Where:

SOH ASCII Start/Start character (hexadecimal value 001H). The system ignores all characters received before the SOH character.

Single markable ASCII character defining the type of the current message.

TYPE

[##] Two optional ASCII decimal digits specifying the station ID number of the marking head. If only one head is used, the ID field can be eliminated and the system will set the value as '00'. For networked applications, i.e. using more than one marking head, the station ID value serves to identify them individually. In marking head networks, the station ID value of the marking head must be implemented. The station ID value must be between 01 and 31.

ASCII character for the start of the text (hexadecimal value 002H).

STX

[TEXT_DATA] An optional field which may be required for some message types. This field contains the current data of the message.

ETX ASCII character for end of text (hexadecimal value 003H).

[BCC] Block Check Code (BCC) generated and sent to improve reliability in error detection.

ASCII carriage return character (hexadecimal value 00DH).

CR

Response format

The head can respond to the host computer in one of two ways:

- If the host transmission is error-free, the marking head responds with an acknowledgement message (ACK) in the following format:

SOH TYPE [##] ACK STX [DATA_TEXT] ETX [BCC] CR

- If the host transmission has errors, the marking head responds with a non-acknowledgement message (NAK) in the following format.

SOH TYPE [##] NAK STX [DATA_TEXT] ETX [BCC] CR

Where:

SOH ASCII Start/Start character (hexadecimal value 001H). The system ignores all characters received before the SOH character.

Single markable ASCII character defining the type of the current message.

TYPE

[##]

Two optional ASCII decimal digits specifying the station ID number of the marking head. If only one head is used, the ID field can be eliminated and the system will set the value as '00'. For networked applications, i.e. using more than one marking head, the station ID value serves to identify them individually. In marking head networks, the station ID value of the marking head must be implemented. The station ID value must be between 01 and 31.

ACK ASCII recognition character (hexadecimal value 006H). Sent if the message was received in the correct format and without errors. Note: This only recognises that the transmission of the communication was successful. It therefore does not identify that the requested action was successfully completed.

ASCII character not recognised (hexadecimal value 015H). Sent if the message was received with an error.

NAK

ASCII character for the start of the text (hexadecimal value 002H).

STX

[TEXT_DATA] An optional field which may be required for some message types. This field contains the current data of the message.

ETX

[BCC]

ASCII character
for end of text
(hexadecimal
value 003H).

INSTRUCTIONS FOR USE OF THE SOFTWARE

AUTOMATOR MARKING SYSTEMS SRL - Via Meucci n.8-20094 Corsico (MI) Italy
Tel 02 48601445 Fax 02 48601503 e-mail info@automator.com www.automator.com

Block
Check
Code
(BCC)
gener
ated
and
sent to
improv
e
reliabili
ty in
error
detecti
on.

CR ASCII carriage return character (hexadecimal value 00DH).

If the host computer does not receive a response from the marking system within three seconds, the host computer transmits the original message again. If no

response after three attempts, the host interrupts the sequence and declares the communication aborted.

Message Types

The following message types are recognised by the marking system when using the extended communication protocol:

- Type 1** Rewrites the contents of the specific field with the data received from the host message. It should be noted that if the field contains flags, these will be replaced and not updated. This type of message uses the format: 1nn<string> where 'nn' represents the program field number and <string> represents the data to be entered into the field. The field number must consist of two digits, so a leading 0 is required for fields 01 to 09.
- Type V** Updates the text variable in the specific field with the data received from the host message. This type of message uses the format: Vnn<string> where 'nn' represents the field number in the programme and <string> the data to be entered in the field. The specific field must contain a text variable flag command (%#V) (see the dedicated chapter in the manual). The field number must consist of two digits, whereby a leading 0 is required for fields 01 to 09.
- Type Q** Updates the text in the specific query table with the data received from the host message. This type of message uses the format: Qnn<string> where 'nn' does not represent the field number of the programme, but represents the number of the query in the specific query table and <string> represents the data to be entered in the query. The field number must consist of two digits, so a leading 0 is required for fields 01 to 09.
- Type P** Indicates that the data received from the host message constitutes the name of the message to be loaded. If the message contains no data, the controller will return the name of the programme currently loaded on the controller.
- Type O** Puts the head online.
- Type G** Indicates a marking cycle.

Type I

This type of message requires the status of the head's output. The head will return a single hexadecimal value to report the status of the PAUSE, READY, DONE, ONLINE and ERROR signals. The system will return a value in relation to the following signal states:

Return value	ERROR	ONLINE	PAUSE	DONE	READY
0	Off	Off	Off	Off	Off
1	Off	Off	Off	Off	ON
2	Off	Off	Off	ON	Off
3	Off	Off	Off	ON	ON
4	Off	Off	ON	Off	Off
5	Off	Off	ON	Off	ON
6	Off	Off	ON	ON	Off
7	Off	Off	ON	ON	ON
8	Off	ON	Off	Off	Off
9	Off	ON	Off	Off	ON
10	Off	ON	Off	ON	Off
11	Off	ON	Off	ON	ON
12	Off	ON	ON	Off	Off
13	Off	ON	ON	Off	ON
14	Off	ON	ON	ON	Off
15	Off	ON	ON	ON	ON
16	ON	Off	Off	Off	Off
17	ON	Off	Off	Off	ON
18	ON	Off	Off	ON	Off
19	ON	Off	Off	ON	ON
20	ON	Off	ON	Off	Off
21	ON	Off	ON	Off	ON
22	ON	Off	ON	ON	Off
23	ON	Off	ON	ON	ON
24	ON	ON	Off	Off	Off
25	ON	ON	Off	Off	ON
26	ON	ON	Off	ON	Off
27	ON	ON	Off	ON	ON
28	ON	ON	ON	Off	Off
29	ON	ON	ON	Off	ON
30	ON	ON	ON	ON	Off
31	ON	ON	ON	ON	ON

Type S

This type of message requests the error status of the head. The head will return a value representing a particular type of error. Each error type has one or more messages that can be displayed on the controller when an error occurs. See the table below for further information:

Response value	Type of error Error message displayed
0x0000	(no error)

0x0001	ONLINE_ERROR Interval ... Home position cannot be found!
0x0002	PATTERN_LOAD_ERROR Error loading a message (Host)! Message not found (Host): <message name>.
0x0004	DISALLOWED_NO_PATTERN No message loaded! No message loaded for marking! No message loaded. Preposition which one?
0x0008	DISALLOWED_OFFLINE Not allowed! The head is offline! Denied! The head is offline! No parking or loading! The head is offline!
0x0010	PATTERN_FIELD_ERROR Text variable not found! Text/Text on arc/Matrix/not found!
0x0020	MARKER_ABORTED_ERROR The head was interrupted!
0x0040	(not used)
0x0080	PIX_OUT_OF_RANGE_ERROR Message Points - Limits Error! Field <number> - Boundary error! Note: The bit is also activated in the case of a non-executable field with character index mode.
0x0200	SN_RANGE_ERROR Error! Beware of serial number limits!
0x0400	(not used)

Type H

Allows you to set the height of a field of type proprietary text or TTF, or bitmap or DXF images. For proprietary texts, the set value will refer to mm and will maintain the proportions of width and height if necessary; for TTF texts, on the other hand, the set value will be used to set the points of the current font; for bitmap images, the height of the image will be changed without maintaining the proportions; for DXFs, the scaling factor will be changed.

This type of message uses the format: Hnn<string> where 'nn' represents the programme field number and <string> represents the value in string format to be used. The field number must consist of two digits, so a leading 0 is required for fields 01 to 09.

Finally, this type of message will only be usable once the

Type C

Allows you to set the position of a text or arc text type field. Text type fields will consider this value to change their justification point position based on what is passed, while for arc text type fields the value will be used to change the position of the centre of the arc. To use this command on an arc text, it must have been constructed with an arc in degree mode.

This type of message uses the format: Cnn<string> where 'nn' represents the field number of the programme and <string> represents the value in string format that is to be used to extract the XY point information to be used to position the text. The field number must consist of two digits, so a leading 0 is required for fields 01 to 09. The string, on the other hand, must necessarily separate the XY value with a ';' character (e.g. 10;10).

Finally, this type of message will only be usable once the relevant functionality has been unlocked from the AC500 Store.

Type D

Allows the diameter of an arc text type field to be set. To use this command on an arc text, it must have been constructed with an arc in degree mode.

This type of message uses the format: Dnn<string> where 'nn' represents the programme field number and <string> represents the value in string format to be used to set the diameter of the arc. The field number must consist of two digits, so a leading 0 is required for fields 01 to 09.

Finally, this type of message will only be usable once the relevant functionality has been unlocked from the AC500 Store.

Type B

Lets you set the starting angle of an arc text type field. To use this command on an arc text, it must have been constructed with an arc in degree mode.

This type of message uses the format: Bnn<string> where 'nn' represents the programme field number and <string> represents the value in string format to be used to set the angle of the arc. The field number must consist of two digits, so a leading 0 is required for fields 01 to 09.

Finally, this type of message will only be usable once the relevant functionality has been unlocked from the AC500 Store.

Type T

Allows you to set the cylinder diameter in marking when using the Theta axis.

This type of message uses the format: T<string> where <string> represents the value in string format to be used to set the diameter.

Block check code

A Block Check Code (BCC) is calculated by performing an addition between the eight type and character bits of the text data. The calculated value is transmitted as a three-digit ASCII decimal number (000 to 255). If the sum is greater than 255, the most significant bit is redundant and is discarded. If the host system is unable to generate a BCC code, this can be omitted with the risk of unidentified errors in data transmission.

The following example is typical of a transmission that includes the calculation of the BCC. To download the character string 'ABC123' to be marked in the first field of the following programme, first wait until the marking cycle is running.

The host sends the following message. (All data are in hexadecimal format).

SOH 1 STX 01ABC123 ETX 238 CR

Where:

Type of message.

1

- 01** Field number. Text
- ABC123** string.
- 238** Block Check Code (BCC).

The BCC code is calculated as follows:

1. Add the character of the message type + the characters of the text (1+01+ABC123):

Character	Hexadecimal value	Description
1	031H	Message type
0	030H	First digit of the field number
1	031H	Second digit of the field number
A	041H	First character of the text
B	042H	Second character of the text
C	043H	Third character of the text
1	031H	Fourth character of the text
2	032H	Fifth character of the text
3	033H	Sixth character of the text
	1EEH	BCC sum

2. The system is only interested in the lower eight bits, so it discards the fifth digit of the sum and keeps the other two lower digits, resulting in the BCC code of EE (hexadecimal). So if you convert the hexadecimal value to decimal:

EE hexadecimal = 238 decimal

3. The decimal value is then converted to its equivalent ASCII representation (where '2'=032H, '3'=033H and '8'=038H). This is the transmitted BCC code. The data transmitted by the host for the entire message are (in hexadecimal):

001H 031H 002H 030H 031H 041H 042H 043H 031H 032H 033H 003H 032H 033H 038H 00DH

4. The head will respond with the transmission of:

001H 031H 006H 002H 003H 030H 034H 039H 00DH

which is equivalent to the following ASCII message:

SOH 1 ACK STX ETX 049 CR

Note that if the system encountered an error when receiving the message from the host, it would respond with a NAK character (015H) instead of the ACK character. The value of BCC would not be changed since ACK/NAK characters are not included in the calculation of BCC.

6.9 Programmable protocol

The programmable protocol is used to communicate with devices where one-way communication (no response from the controller) is required, such as barcode readers. No error checking or data recognition is given. Programmable protocol parameters define which characters to analyse from the received data and how to use them during the marking operation. These parameters are described below.

NOTE

The message transmitted by the host may contain ASCII characters that define the Start/Start, Terminator and Ignore characters within the transmitted data. To identify these ASCII characters, it is necessary to specify their equivalent decimal value when setting these parameters. Please refer to the ASCII code table in Appendix A of the manual for information on decimal or hexadecimal ASCII character codes.

PROTOCOL CONFIGURATION

Programmable protocol

Start char: Terminator:

Position: Length:

Ignore: Msg type:

Timeout (s):

Echo mode

String:

Programmable protocol configuration screen

Start Character. (*Optional*) Identifies the ASCII character in the host message from which the marking system will start counting character positions.

Character Terminator. Identifies the ASCII character in the host message that represents the end of the transmitted string. The Terminator character is generally the ASCII *carriage return* character (CR, 13 decimal).

Position Parameter. Identifies the character position in the received data string from where the system is to begin extracting the data transmitted in the host message.

Length parameter. (*Optional if the Terminator character is used*) Identifies how many characters are to be extracted from the host message. The Position and Length parameters work together to extract data from the host message.

Ignore Character. Identifies the character in the host message that the system should ignore. The Ignore character is generally the ASCII *line feed* character (LF, 10 decimal).

Message Type. Allows the message types to be recognised for the programmable protocol that defines how the system will use the data received from the host. The possible types are: P, V, 1, Q and 0 (zero).

P Loads a specific programme, identified by the data extracted from the host message.

V Updates the first variable text field of the loaded programme with data extracted from the host message.

1 Overwrites the contents of the first text field of the loaded programme with data extracted from the host message. Please note that if the field contains flag commands, these will be overwritten and not updated.

Q Updates the text of the first query set in the table with data extracted from the message from the host.

0 Indicates that the host message will provide information on how to handle the data just received. This option allows greater flexibility, leaving the possibility of selecting a programme or modifying its data, directly from the host.

The host message must use the following format: **Tnn<string>**.

where:

T = P, V, 1 or Q. They indicate the type of message.

nn = two-digit number indicating the number of the field or query table where the data is to be entered. In the case of message type P (selection of a programme) this number is not used.

<string> = programme name (message type P) or field data (message type V, 1 or Q)

Timeout. This identifies a time in seconds within which the receive buffer will be completely cleaned. Thus, if an incomplete packet is received, it will not be retained in the receive buffer, but will be automatically cleared to ensure consistency between data in subsequent transmissions.

Echo Mode

Echo mode makes it possible to send a string from the controller to the host whenever a marking has been finished. This mode uses special flags to extract data from the programme or to output ASCII characters.

To use Echo mode, you must have set the programmable protocol as the serial protocol and enabled Echo mode.

In the Echo mode parameter string, the data that will be sent to the host must be specified.

- Use the **%##Q** flag to indicate a specific field in the programme, where **##** represents the field number from 01 to 99 (see example below).
- Use the **%###C** flag to indicate a single ASCII character, where **###** represents the decimal value of the ASCII character. Please refer to Appendix A for ASCII characters and their corresponding decimal values.
- Use the command **\n** or **\N** to indicate a carriage return ASCII character and a line feed ASCII character.
- Use the command **\r** or **\R** to indicate a carriage return ASCII character only.

Examples:

Suppose you mark a programme with three fields:

field 01= **SN:2004**
field 02= **247-14:06**
field 03= **AUTOMATOR**

- If the Echo mode parameter string is set as:

DATE=%1Q;%2Q%13C%10C

The data transmitted by the controller will be:

DATA=SN:2004;247-14:06<CR><LF>

- If the Echo mode parameter string is set as:

DATA=%2Q;%3Q\R

The data transmitted by the controller will be:

DATA=247-14:06;AUTOMATOR<CR>

6.10 Programmable USB barcode protocol

The USB programmable barcode protocol is used to communicate with USB barcode reader devices. As with the serial programmable protocol, there is no error checking or data recognition. The parameters of the programmable barcode protocol define which characters to parse from the received data and how to use them during the marking operation. These parameters are described below.

NOTE

The message transmitted by the barcode may contain ASCII characters that define the Start/Start, Terminator and Ignore characters within the transmitted data. To identify these ASCII characters, it is necessary to specify their equivalent decimal value when setting these parameters. Please refer to the ASCII code table in Appendix A of the manual for information on decimal or hexadecimal ASCII character codes.

NOTICE

The USB barcode must be set up as a keyboard simulation in order to function properly on the system.

Programmable protocol configuration screen for barcode

Start Character. (*Optional*) Identifies the ASCII character in the barcode message from which the marking system will start counting character positions.

Character Terminator. Identifies the ASCII character in the barcode message that represents the end of the transmitted string. The terminator character is generally the ASCII *carriage return* character (CR, 13 decimal). **It is also possible to specify an alternative terminator.**

Position Parameter. Identifies the character position in the received data string from where the system should start extracting the data transmitted in the barcode message.

Length parameter. (*Optional if the Terminator character is used*) Identifies how many characters are to be extracted from the barcode message. The Position and Length parameters work together to extract data from the barcode message.

Ignore Character. Identifies the character in the barcode message that the system should ignore, The ignore character is generally the ASCII *line feed* character (LF, 10 decimal).

Message Type. Allows the message types to be recognised for the programmable protocol that defines how the system will use the data received from the barcode. The possible types are: P, V, 1, Q, A, N, C and 0 (zero).

P Loads a specific programme, identified by the data extracted from the barcode message.

V Updates the first variable text field of the loaded programme with the data extracted from the barcode message.

- 1** Overwrites the contents of the first text field of the loaded programme with the data extracted from the barcode message. Please note that if the field contains flag commands, these will be overwritten and not updated.
- Q** Updates the text of the first query set in the table with the data extracted from the barcode message.
- A** Adds the data extracted from the barcode message to the content of the first text field of the loaded programme.
- N** Selects the next field of the loaded programme.
- C** Launches the programme character checking procedure. The message received will contain the theoretical total number of characters in the programme, including the characters of the programme name, and these will be compared with the actual number of characters in the programme. If the check fails, the programme will be reloaded.
- S** Launches a printout. Only if the labeller management feature is enabled and set to BARCODE.
- O** Allows you to set the operator code in case the login management operator is enabled by machine parameter (see chapter 2.20)
- 0** Indicates that the barcode message will provide information on how to handle the data just received. This option allows greater flexibility, allowing the possibility of selecting a programme or modifying its data, directly from the barcode.
The barcode message must use the following format: **Tnn<string>**.

where:

T = Character indicating the type of message.

nn (optional)= two-digit number indicating the number of the field or query table where the data is to be entered. In the case of the Pattern message type (selection of a programme) this number is not used. If this value is to be used, the Field ID Enable parameter must be set.

<string> = programme name (message type *Pattern*) or field data. Not required in the case of message type *Next*.

In the case of message type 0 (zero), a configuration screen of the message types managed by the barcode will be made possible. This allows an identification character to be associated for each message type. When analysing the packet, the system will check the message type character and if a match is found with the message types, the system will use the appropriate handling of the newly received data.

In order to configure message types, press the button



NOTE

The message type configuration button is only enabled if message type 0 (zero) is selected.

MESSAGE CONFIGURATION					
	TYPE	MSG	ENABLE	SPECIAL	
1	Pattern	P	YES	▼	NO
2	Update	V	YES	▼	NO
3	Overwrite	1	YES	▼	NO
4	Query	Q	YES	▼	NO
5	Append	A	YES	▼	NO
6	Next	N	YES	▼	NO
7	Check	C	YES	▼	NO
8	All (0)	0	YES	▼	NO





Message type configuration screen

This screen shows all the message types that the system can manage. For each message type, it is possible to set the character contained in the barcode message that identifies the type of message, whether the management of each message type is enabled and whether the specified character is to be treated as a special character. A special character is defined as a character that can be repeated several times within the barcode message and that is not to be considered as actual message data.

- associated with the function of overwriting the contents of a text field.
- **Query:** type of message associated with the function of overwriting the contents of a Query field.

TYPES OF MESSAGES

- **Pattern:** type of message associated with the loading function of a programme.
- **Update:** type of message associated with the update function of a variable text field.
- **Overwrite:** type of message associated with the function of overwriting the contents of a text field.
- **Query:** type of message associated with the function of overwriting the contents of a Query field.

- **Append:** type of message associated with the function of inserting the message read from the barcode at the end of the selected text field. This message type can be left blank ("") so that all barcodes that do not fall into any of these message types can be handled as append message types.
- **Check:** type of message associated with the character check function of the programme. Checking the number of characters also includes the programme name.
- **Print:** type of message associated with the print function of a label.

Below are some examples of how the handling of message types configured for the barcode protocol works.

MESSAGE CONFIGURATION					
	TYPE	MSG	ENABLE	SPECIAL	
1	Pattern	@	YES	YES	
2	Update	V	NO	NO	
3	Overwrite	1	NO	NO	
4	Query	Q	NO	NO	
5	Append		YES	NO	
6	Next	#	YES	NO	
7	Check	C	YES	NO	
8	All (0)	0	YES	NO	

Example of message type configuration

Examples (with reference to the above configuration)

- One is supposed to receive such a message:

@TEST

the system will handle the message as a *Pattern* ('@') and will load the programme called TEST. Since the message was declared as *special* if a message like:

@TEST@

the system will analyse all characters defined for *the Pattern* type and they will be removed, so the programme that will be loaded will always be called TEST.

- One is supposed to receive such a message:

#

the system will handle the message as Next ('@') and will select the next field in the programme.

- One is supposed to receive such a message:

€131

the system will handle the message as a Check ('€') and will launch the character number check function of the currently loaded programme.

- Any other fields will be handled as *Append* as messages of type *Update*, *Overwrite* and *Query* have been disabled.

Example of barcode usage procedure

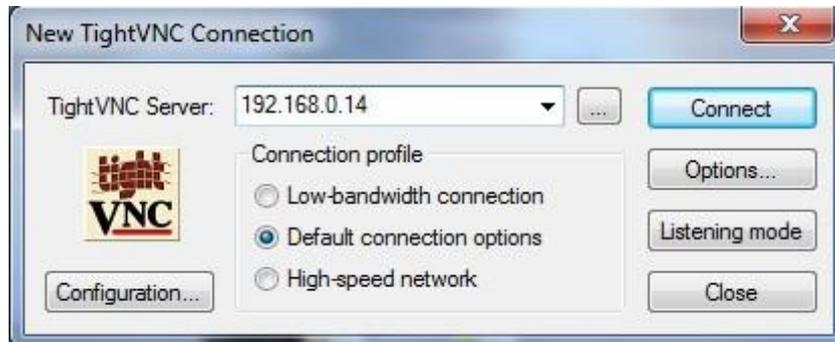
A possible procedure for using barcodes could be as follows:

- Load the programme from the barcode;
- Set the data in the fields with the appropriate barcode; change, if necessary, the selected field to be changed;
- Check the programme from the barcode.

6.11 Remote controller control

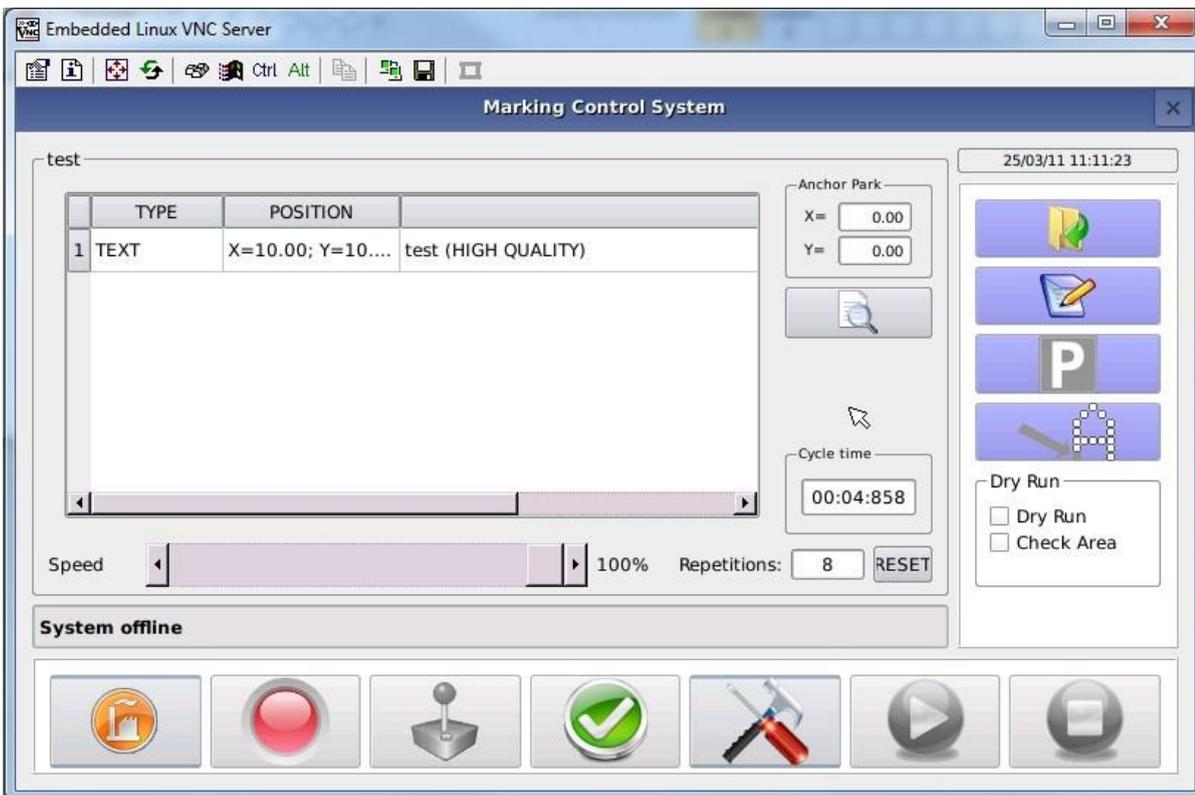
The system allows remote software to be used to control the control unit interface. The software to be used is the VNC Viewer. The remote control of the controller works via an Ethernet connection between the controller and any computer. The computer only needs to be configured to use the VNC Viewer.

To use remote control of the control unit, simply connect the control unit to a PC via Ethernet cable, open the VNC software and enter the IP address of the control unit. The default IP address is 192.168.0.14.



VNC Viewer software screen

Once the VNC Viewer software has been configured, simply press the *Connect* button to establish a connection with the controller.



VNC Viewer software screen
with remote control of the control unit
ATTENTION



With remote control of the control unit, you will control all functions of the machine. Make sure you have taken all necessary precautions before starting a remote control session of the machine.

Document:

INSTRUCTIONS FOR USE OF THE SOFTWARE

AUTOMATOR MARKING SYSTEMS SRL - Via Meucci n.8-20094 Corsico (MI) Italy
Tel 02 48601445 Fax 02 48601503 e-mail info@automator.com www.automator.com

Appendix A

ASCII - Decimal - Hexadecimal cross-reference

The table below presents the standard ASCII characters with their decimal and hexadecimal equivalents. This table can be useful when communications between the controller and the host computer occur.

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

Appendix B

Configuration and use of Theta axis (rotational)

This chapter will detail the configuration parameters and use of the optional Theta axis managed by the AC500 controller. The optional Theta axis is used to add further functionality to the marking system offered by the AC500, particularly for marking cylindrical or multi-face parts.

The Theta-axis is an optional axis that is switched with the standard Y-axis via a special relay on the TTL I/O signal connector. This means that the Theta-axis can only be controlled if the machine's Y-axis is not currently being controlled. However, the two axes can coexist but cannot be commanded at the same time. In the machine logic, these axes will be handled sequentially and optimised.

As mentioned above, the Theta axis allows parts with cylindrical surfaces and parts with faces to be marked. Two different machining modes have been implemented to handle the differences between the two modes: *perpendicular* mode (for cylindrical parts), *index* mode (for parts with faces) and *character index* mode (for cylindrical parts where it is not possible to move the theta axis interpolated with X or Y).

B.1 Enabling and configuring the Theta axis

The optional Theta axis works in conjunction with the *rotational support* to mark cylindrical parts or multiple faces of square or hexagonal parts. The rotational support is an electromechanical spindle that holds and rotates the part to be marked. It is also possible to adjust the spindle attachment to the size of the part to be marked. In order to use the Theta axis, **it is absolutely necessary to ensure that the axis is enabled and configured before using it.**

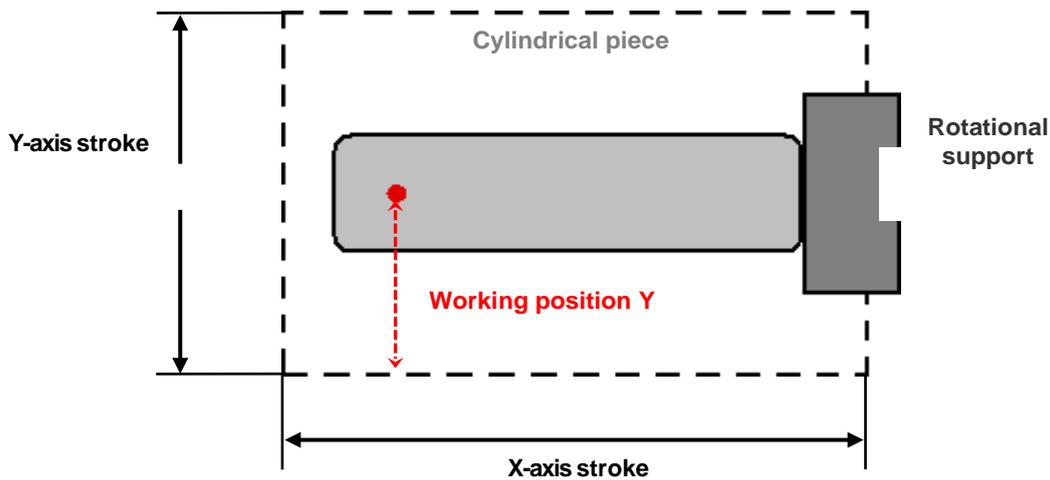
To enable the Theta axis, the general setting ("*Third Axis Management*") must be set to THETA AXIS. For further details, please refer to the section "*Setting General Parameters*" in the manual.

Setting the value for third-axis management in this way will enable Theta-axis management on the AC500 control unit.

In the general parameter environment, certain parameters exclusive to the Theta axis can be set:

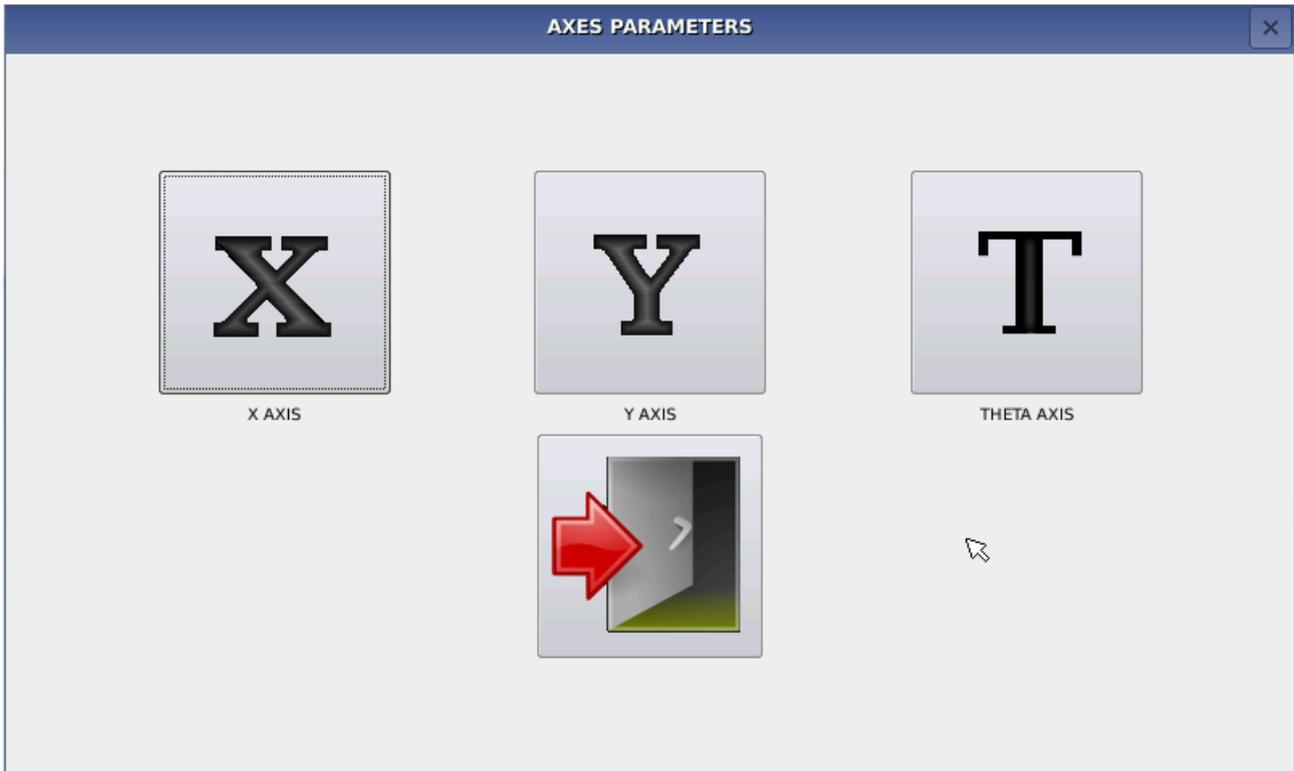
- *Standard workpiece diameter theta axis*: parameter indicating the default workpiece diameter to be used when creating the programme. This parameter is used when marking cylindrical workpieces (*perpendicular mode* or *character index*).

- *Y (X) position for theta-axis machining*: Parameter indicating the position to which the Y (or X) axis is to be brought during the theta-axis marking phase. This parameter is used when marking cylindrical workpieces (*perpendicular mode* or *character index*). The axis involved is Y if the parameter ("*Axis aligned with theta-axis*") is set to X-axis, and X in the opposite case.



- *Theta axis drive diameter*: parameter indicating the diameter of any roller used in the indirect rotation of the cylinder to be marked. If the parameter is set to zero, the cylinder to be marked is considered to be directly connected to the rotational support.
- *Calibration theta-axis*: Parameter that allows the theta-axis to be excluded from the search sequence for calibration positions when the control unit comes online. If the parameter is set to NO, no movement is commanded to the theta-axis and the current position is set to the one indicated in the parameter ("*Calibration dimension*") of the theta-axis itself.
- *Axis aligned with theta rotation axis*: Parameter to define which of the Cartesian axes X and Y is the one aligned with theta rotation axis. For example in the figure above, the Cartesian axis aligned with theta's rotation axis is X.
- *Flat surface tolerance theta axis*: default value set at the time of creation in the equivalent parameter in the work programme.

In addition, the axis parameter environment will display the environment for setting the parameters for the optional axis. For more information on axis parameters, see the section "*Setting Axis Parameters*" in the manual.



Axis parameter screen with Theta enabled

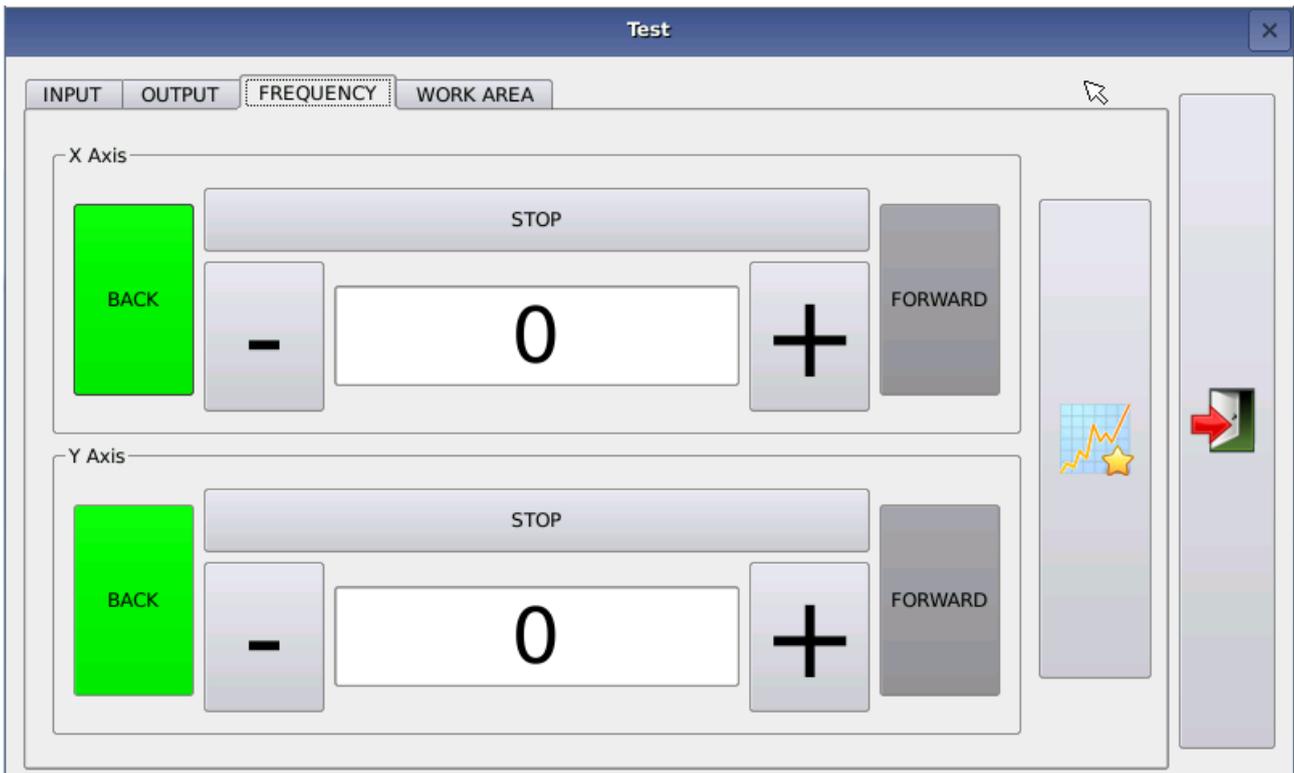
	PARAMETER	VALUE
1	Encoder pulses:	10500
2	Quote / round encoder (degrees):	360
3	Maximum quote (degrees):	360
4	Minimum quote (degrees):	0
5	Maximum speed (degrees/s):	200
6	Microswitch speed (degrees/s):	50
7	Acceleration time (s):	0.2
8	Deceleration time (s):	0.2
9	Mx. Frequency (HZ):	5833
10	Setting quote (degrees):	0
11	Setting direction:	MINUS
12	Setting microswitch channel:	19

Theta axis parameter screen

B.2 Theta axis test

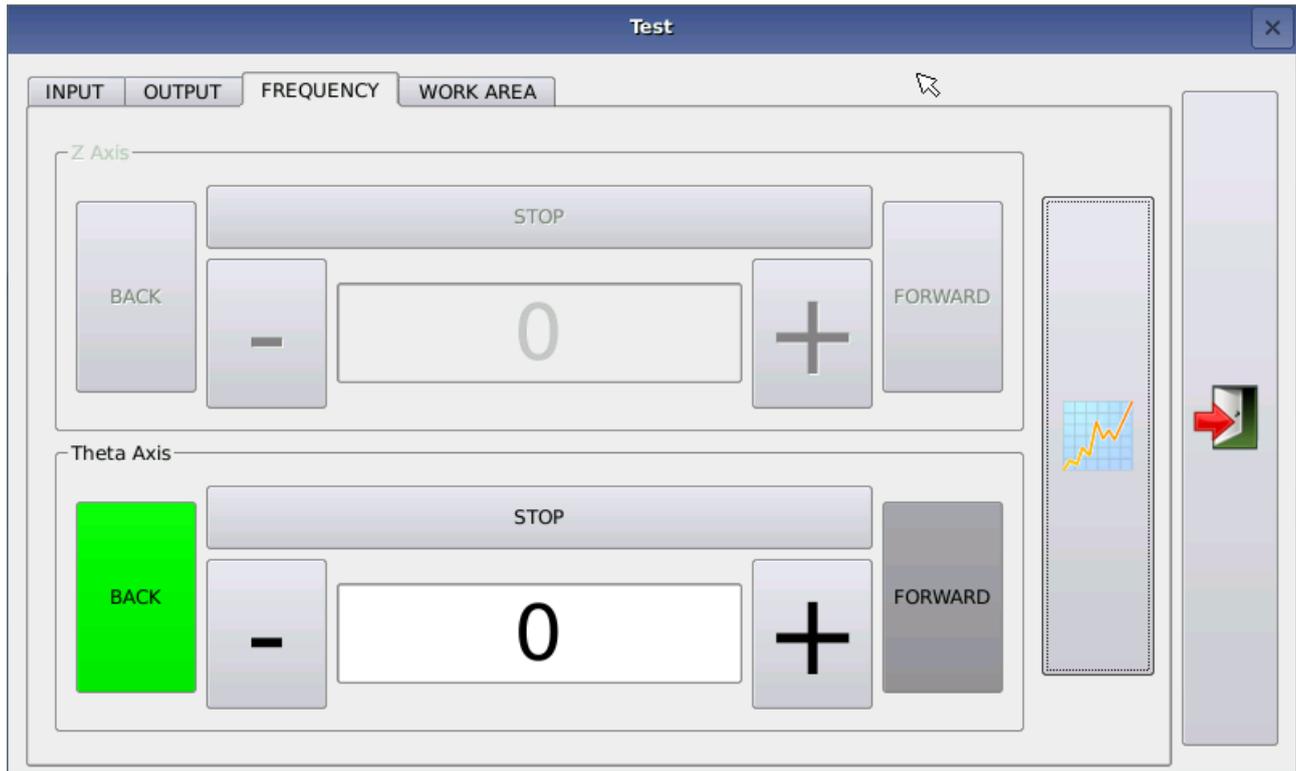
Once the management of the Theta axis is enabled, it is possible to proceed with the normal axis test procedure from the Test environment (see section 'System Test' in the manual).

To test the Theta axis, simply go to the frequency output test window:



Frequency output test window

And press the  button to display information about the frequency of the Theta axis:



Theta axis frequency output test window

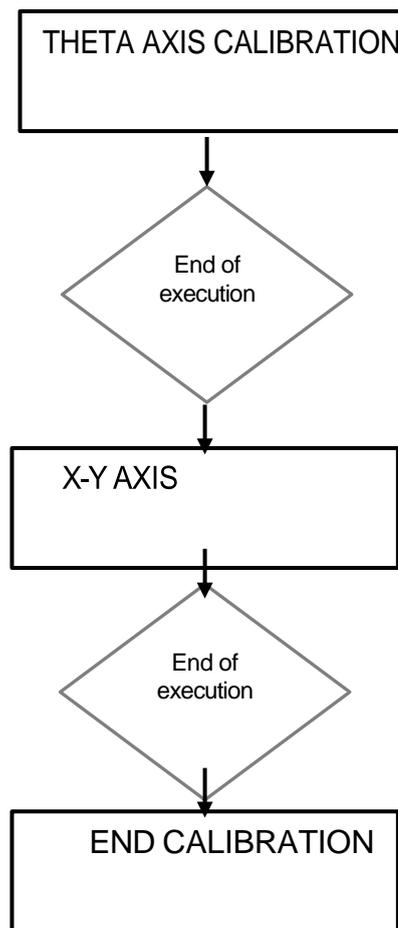
From this window, you can proceed with all the instructions described in the manual (see section '*System Test*').

B.3 Calibration of the Theta axis

Calibrating the Theta axis is an operation that is performed every time the machine's online status is set. Therefore, as with the standard X and Y axes, it is an optional operation on the part of the operator.

Calibration is a necessary operation to calibrate the Theta axis to the set parameters. The calibration procedure is the same as for standard axes. The axis will be moved according to the set calibration direction in search of the limit switch. Once the limit switch has been reached, the calibration altitude set in the parameters will be set as the current axis altitude. If an axis is already on its limit switch at the time of calibration, the axis will be moved according to the calibration exit dimension parameter and then calibrated again.

As previously mentioned, the Theta axis is switched with the standard Y-axis and therefore cannot be calibrated together with the Y-axis. If the Theta axis is enabled, the complete calibration procedure for the machine (in the online state) will be as follows:

**NOTE**

If the general parameter *Axis at rest torque* is set to *Theta*, the calibration order will be reversed. In this case, the calibration of the X- and Y-axes will be commanded first, and only after this will the calibration of the Theta-axis be commanded.

For further information see chapter "*Setting General Data*".

To proceed with manual calibration of the Theta axis, you must enter the axis calibration environment (see *the 'Axis Calibration'* section in the manual). The axis calibration environment will look like this:



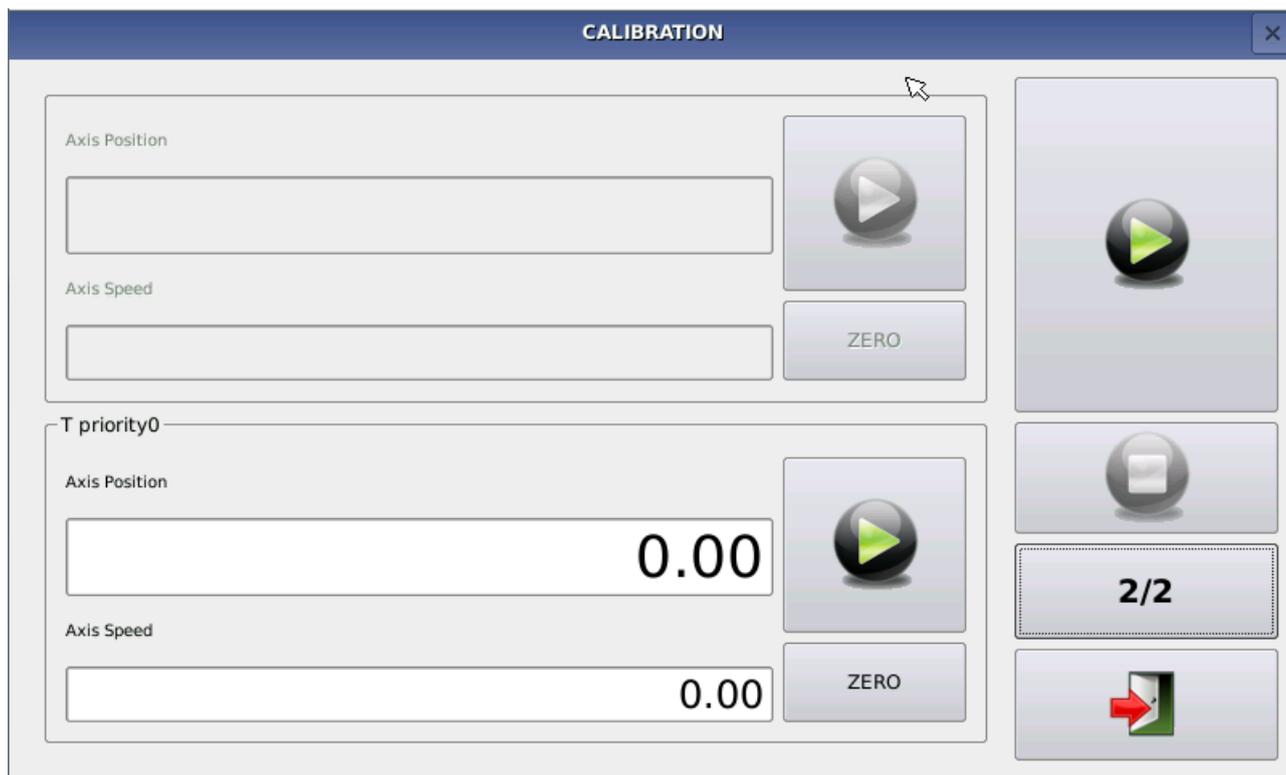
Axis calibration screen

To proceed with the calibration of the individual Theta axis, press the button:



Button for optional axis calibration

In this way, the calibration screen will display the calibration information of the Theta axis:



Optional axis calibration screen

From this screen, you can proceed with the normal instructions already mentioned in the section "*Calibration of axes*" in the manual.

NOTE

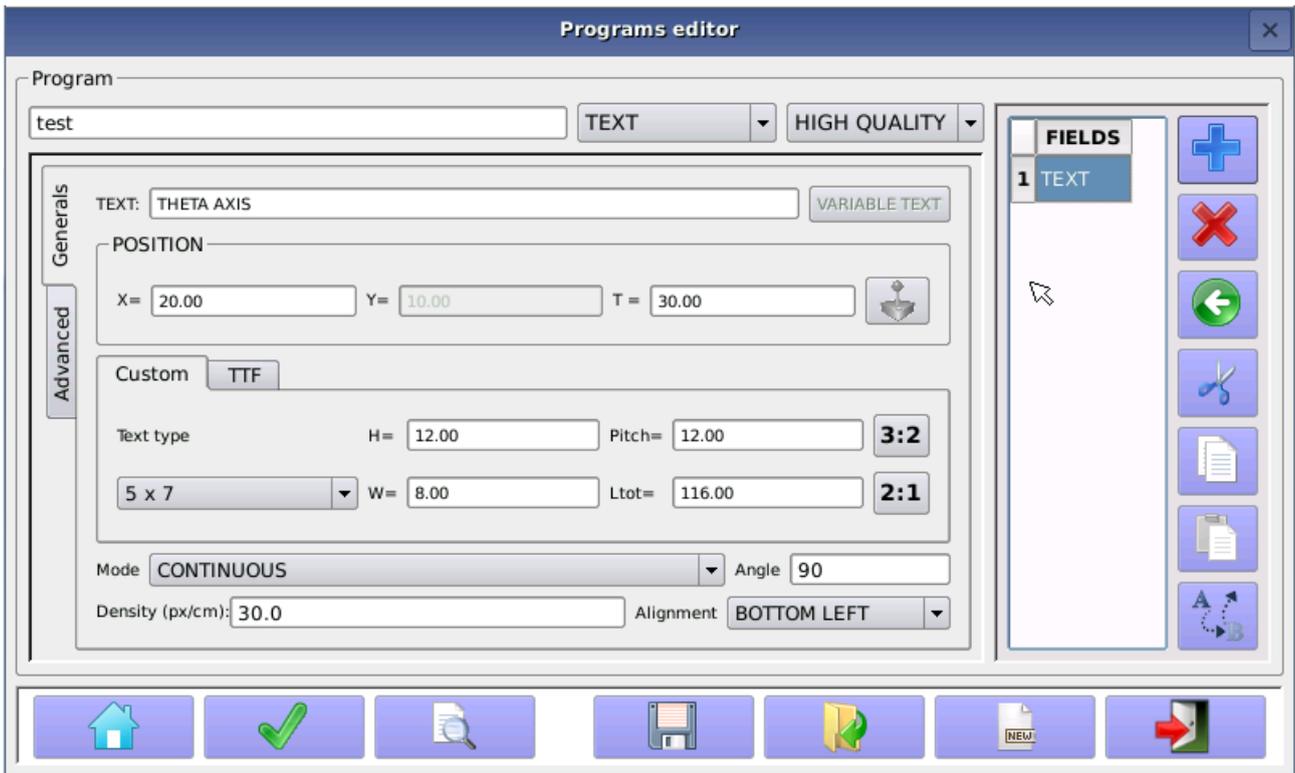
It is also possible to calibrate the Theta axis from the complete machine calibration procedure. In this case, the order in which the axes are calibrated will be the one used for bringing the machine online.

If *the* general datum ("*Calibration axis theta*") is set to NO, the axis will not perform any movement and will load the position set in the parameter ("*Calibration altitude*") of the axis as the current position.

B.4 Modifying programmes with Theta axis

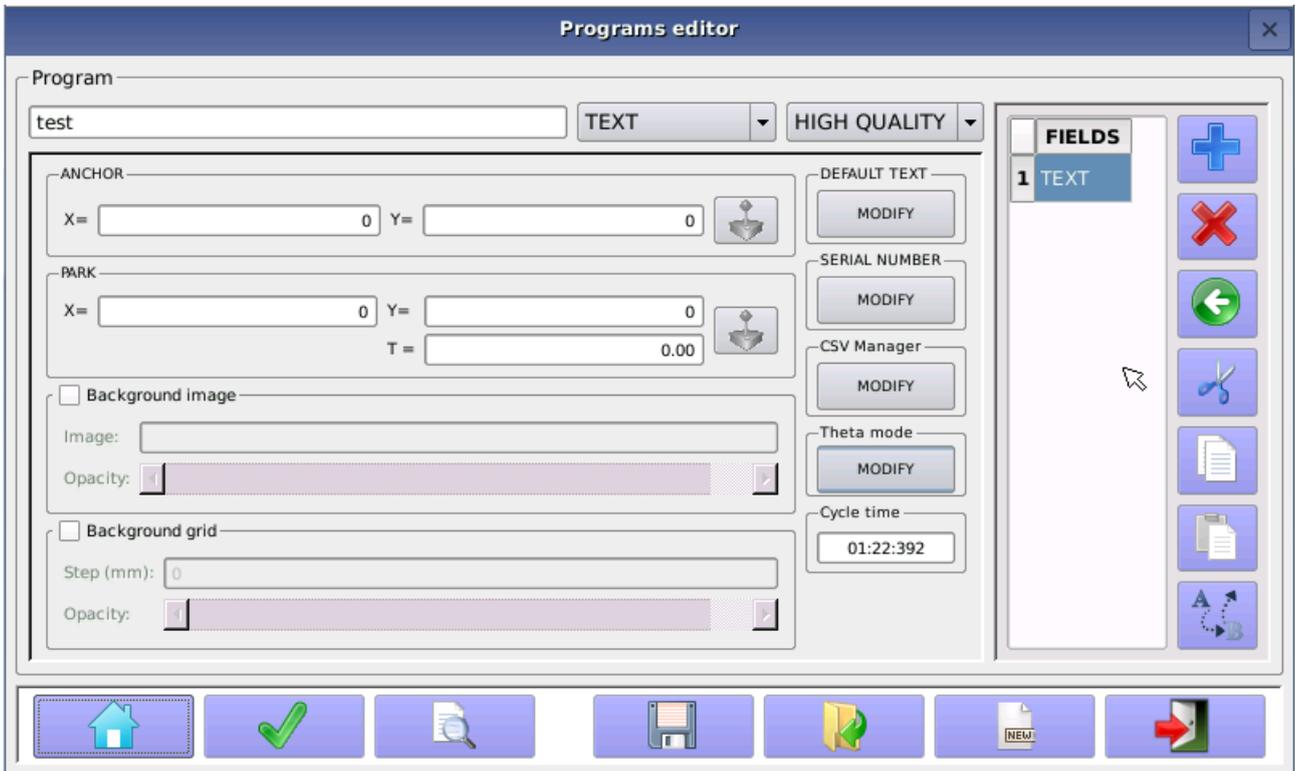
If the Theta axis is enabled, the programme creation and editing environment will adapt to display useful and necessary information for correctly programming programme data with Theta axis management. All normal functions of the programme editing environment are also available when the Theta axis is enabled. To see all information about the programme editing environment, please refer to the chapter "*Creating a programme*" in the manual.

The editor screen will be as follows:



Programme editing screen

The general programme data screen will display the machining parameters of the Theta axis.



General programme data screen

To display the data of the Theta axis, press the button below the '*Theta mode*' label:



Theta mode button

The screen displayed will contain information on the working mode of the Theta axis and the diameter of the workpiece to be marked.



Theta mode configuration screen

Mode: Defines the working mode of the Theta axis for the current programme. Settable values are: *perpendicular*, *index*, *off* and *character index*.

In *perpendicular* mode (marking cylindrical workpieces), the Theta axis will be used to rotate the workpiece mounted on the support while the X axis will be positioned for marking points at the calculated coordinates of the various fields. In this mode, the Y-axis will not be used and will be positioned, at the start of the cycle, at the position set in the general parameter "*Y-axis machining position Theta*". If the parameter ("*Axis aligned with theta rotation axis*") is set to *Y-axis*, the roles of the X and Y axes will be reversed.

In *index* mode (marking workpieces with faces) the Theta axis will be used to rotate the workpiece mounted on the support, then the X and Y axes will be positioned for marking points at the calculated coordinates of the various fields. In this mode, the Theta axis will be used when changing the theta positioning value between fields. Once Theta is positioned, the X and Y axes will work in standard mode.

In *off* mode, the Theta axis will be disabled for the current programme and the machine will work in standard mode.

In *character index* mode, the configuration window display will be as follows:

The image shows a software dialog box titled "THETA CONFIGURATION". It contains several input fields and buttons. At the top right is a close button (X). The "Mode" dropdown is set to "CHARACTER INDEX". Below it, the "Part diameter" text box contains "200.0". A section titled "CHARACTER INDEX settings" contains a "Type" dropdown set to "CHARACTER", a "Plane surface tolerance" text box with "1.00" and a unit indicator "I", and a "Marking area" text box with "28.2". At the bottom, there are two buttons: a green checkmark button on the left and a red X button on the right.

Theta mode configuration screen character index

In this mode, the system will perform a movement of the Theta axis before the execution of each character of a text, and then marking will be performed by moving the X and Y axes in the traditional way. Read the description of the configuration parameters for this mode for further details.

Workpiece diameter

Defines the size of the workpiece to be marked. Based on this value, the system will calculate the rotational displacements required to mark the programme on the circular surface of the workpiece.

The workpiece diameter value is only used for the working modes *perpendicular and index character* and not for *index* mode.

Type

Defines how the movement of the *Theta* axis is calculated in the *character index* mode. If the "*CHARACTER*" option is selected, the system will perform a movement of the Theta axis before the execution of each character of a *TEXT* or *TEXT ON ARC* field, or before the execution of any of the other field types. If the "*OPTIMISED*" option is selected, the system will attempt to minimise the number of Theta axis movements by marking as many characters as possible in the current useful area.

Flat surface tolerance

Defines the tolerance used (along with the cylinder diameter) in the calculation of the useful area, with the geometric meaning shown in the figure:



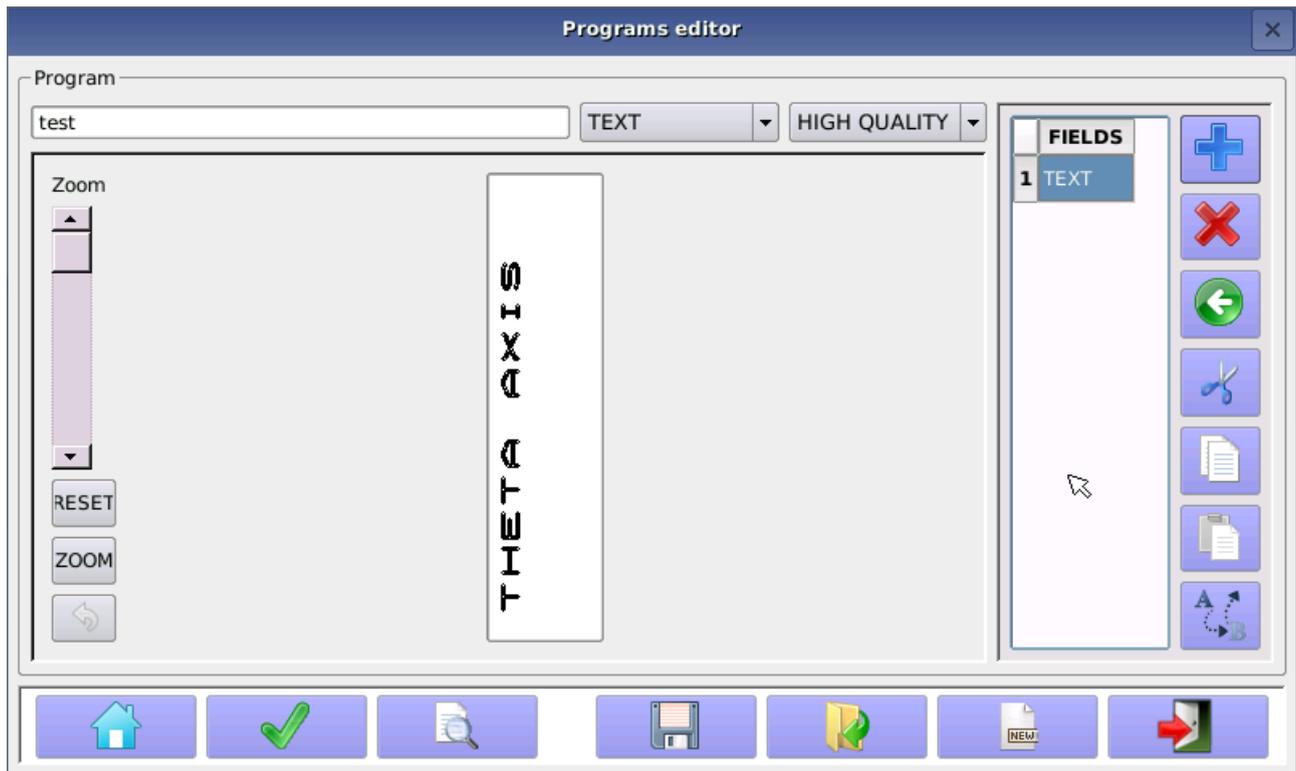
Character index mode configuration parameters The useful area is used in two distinct ways:

1. in the '*OPTIMIZED*' mode for evaluating the number of characters that can be marked without moving the Theta axis
2. in both modes to assess the feasibility of marking individual characters and non-text fields. In the event that a character in a text or a non-text field is larger than the current usable area, the field will be marked as incorrect and the message '*FIELD NOT ENABLE IN CHARACTER INDEX MODE*' will be displayed on the screen.

In the window below the tolerance setting, you can read off the size of the useful area according to the current settings (field cannot be changed).

In the programme's general data, it is also possible to enter the parking height to which the Theta axis must be brought at the end of a cycle or at the command of a parking operation.

For each program field, the height at which the Theta-axis is to be positioned when marking for machining the field can be set. In the case of *perpendicular* working mode for the Theta axis, the Y-axis position fields will be disabled because they are not used (in fact, the axis will be positioned, at the start of the cycle, in the position set in the general parameter "*Y position machining Theta axis*"). If the parameter ("*Axis aligned with theta-axis*") is set to Y-axis, the X-axis position fields will be disabled.



Program editing screen with preview

B.5 Manual environment with Theta axis

From the jog environment, it is possible to move the Theta axis within its maximum limits in the same way as the X and Y axes. More information on jog movements can be found in the '*Moving the marking head*' section of the manual.

With the Theta axis enabled, the following jog screen will be displayed:



Axis jog screen

To move the Theta axis from the jog screen, it is necessary to press on the optional jog display button:



Optional jog display button

The screen that will be opened will be as follows:



Theta jog axis screen

We can now carry out the same instructions as in the section *"Moving the marking head"* to move the Theta axis.

B.6 Working environment with Theta axis

By enabling the Theta axis, the main screen of the software will be changed to display the Theta axis information for the currently loaded programme.



Main screen with Theta axis enabled

All functions in the main environment will also be available if the Theta axis is enabled. For further information on managing the main environment, please refer to the chapter "The Operating System" in the manual.



Main environment screen with Theta axis enabled

In the case of *Parking* and *Prepositioning* movements, the logic of the Theta axis will depend on the Theta mode set in the currently loaded programme:

- *Perpendicular mode:*

In the case of this mode, pre-positioning will bring the Y-axis to the working position (general datum '*Y-axis machining position Theta*') and subsequently bring X and Theta to the position dimension of the first field to be marked.

Parking, on the other hand, will first place X and Theta at their respective parking quotas and then take Y to its own parking quota.

- *Index mode:*

In the case of this mode, pre-positioning will bring the Theta axis to the position specified in the first field to be marked and subsequently bring X and Y with normal pre-positioning operation to the field.

Parking, on the other hand, will first place Theta at its parking quota and then bring X and Y with the normal parking operation of a programme.

- *Character index mode:*

In the case of this mode, pre-positioning will bring the Theta axis to the position corresponding to the marking position of the first field to be marked and subsequently bring X and Y with normal pre-positioning operation to the field.

Parking, on the other hand, will first place Theta at its parking quota and then bring X and Y with the normal parking operation of a programme.

Appendix C

Remote programme selection table

SEL_6 Signal status	SEL_5 Signal status	SEL_4 Signal status	SEL_3 Signal status	SEL_2 Signal status	SEL_1 Signal status	SEL_0 Signal status	Resulting BCD value	Name of the reserved programme
OFF	0000000	(none - function disabled)						
OFF	OFF	OFF	OFF	OFF	OFF	ON	0000001	PAT01
OFF	OFF	OFF	OFF	OFF	ON	OFF	0000010	PAT02
OFF	OFF	OFF	OFF	OFF	ON	ON	0000011	PAT03
OFF	OFF	OFF	OFF	ON	OFF	OFF	0000100	PAT04
OFF	OFF	OFF	OFF	ON	OFF	ON	0000101	PAT05
OFF	OFF	OFF	OFF	ON	ON	OFF	0000110	PAT06
OFF	OFF	OFF	OFF	ON	ON	ON	0000111	PAT07
OFF	OFF	OFF	ON	OFF	OFF	OFF	0001000	PAT08
OFF	OFF	OFF	ON	OFF	OFF	ON	0001001	PAT09
OFF	OFF	OFF	ON	OFF	ON	OFF	0001010	PAT10
OFF	OFF	OFF	ON	OFF	ON	ON	0001011	PAT11
OFF	OFF	OFF	ON	ON	OFF	OFF	0001100	PAT12
OFF	OFF	OFF	ON	ON	OFF	ON	0001101	PAT13
OFF	OFF	OFF	ON	ON	ON	OFF	0001110	PAT14
OFF	OFF	OFF	ON	ON	ON	ON	0001111	PAT15
OFF	OFF	ON	OFF	OFF	OFF	OFF	0010000	PAT16
OFF	OFF	ON	OFF	OFF	OFF	ON	0010001	PAT17
OFF	OFF	ON	OFF	OFF	ON	OFF	0010010	PAT18
OFF	OFF	ON	OFF	OFF	ON	ON	0010011	PAT19
OFF	OFF	ON	OFF	ON	OFF	OFF	0010100	PAT20
OFF	OFF	ON	OFF	ON	OFF	ON	0010101	PAT21
OFF	OFF	ON	OFF	ON	ON	OFF	0010110	PAT22
OFF	OFF	ON	OFF	ON	ON	ON	0010111	PAT23
OFF	OFF	ON	ON	OFF	OFF	OFF	0011000	PAT24
OFF	OFF	ON	ON	OFF	OFF	ON	0011001	PAT25
OFF	OFF	ON	ON	OFF	ON	OFF	0011010	PAT26
OFF	OFF	ON	ON	OFF	ON	ON	0011011	PAT27
OFF	OFF	ON	ON	ON	OFF	OFF	0011100	PAT28
OFF	OFF	ON	ON	ON	OFF	ON	0011101	PAT29
OFF	OFF	ON	ON	ON	ON	OFF	0011110	PAT30
OFF	OFF	ON	ON	ON	ON	ON	0011011	PAT31
OFF	ON	OFF	OFF	OFF	OFF	OFF	0100000	PAT32
OFF	ON	OFF	OFF	OFF	OFF	ON	0100001	PAT33
OFF	ON	OFF	OFF	OFF	ON	OFF	0100010	PAT34
OFF	ON	OFF	OFF	OFF	ON	ON	0100011	PAT35
OFF	ON	OFF	OFF	ON	OFF	OFF	0100100	PAT36
OFF	ON	OFF	OFF	ON	OFF	ON	0100101	PAT37
OFF	ON	OFF	OFF	ON	ON	OFF	0100110	PAT38
OFF	ON	OFF	OFF	ON	ON	ON	0100111	PAT39
OFF	ON	OFF	ON	OFF	OFF	OFF	0101000	PAT40
OFF	ON	OFF	ON	OFF	OFF	ON	0101001	PAT41
OFF	ON	OFF	ON	OFF	ON	OFF	0101010	PAT42
OFF	ON	OFF	ON	OFF	ON	ON	0101011	PAT43

INSTRUCTIONS FOR USE OF THE SOFTWARE

AUTOMATOR MARKING SYSTEMS SRL - Via Meucci n.8-20094 Corsico (MI) Italy
 Tel 02 48601445 Fax 02 48601503 e-mail info@automator.com www.automator.com

OFF	ON	OFF	ON	ON	OFF	OFF	0101100	PAT44
OFF	ON	OFF	ON	ON	OFF	ON	0101101	PAT45
OFF	ON	OFF	ON	ON	ON	OFF	0101110	PAT46
OFF	ON	OFF	ON	ON	ON	ON	0101111	PAT47
OFF	ON	ON	OFF	OFF	OFF	OFF	0110000	PAT48
OFF	ON	ON	OFF	OFF	OFF	ON	0110001	PAT49
OFF	ON	ON	OFF	OFF	ON	OFF	0110010	PAT50
OFF	ON	ON	OFF	OFF	ON	ON	0110011	PAT51
OFF	ON	ON	OFF	ON	OFF	OFF	0110100	PAT52
OFF	ON	ON	OFF	ON	OFF	ON	0110101	PAT53
OFF	ON	ON	OFF	ON	ON	OFF	0110110	PAT54
OFF	ON	ON	OFF	ON	ON	ON	0110111	PAT55
OFF	ON	ON	ON	OFF	OFF	OFF	0111000	PAT56
OFF	ON	ON	ON	OFF	OFF	ON	0111001	PAT57
OFF	ON	ON	ON	OFF	ON	OFF	0111010	PAT58
OFF	ON	ON	ON	OFF	ON	ON	0111011	PAT59
OFF	ON	ON	ON	ON	OFF	OFF	0111100	PAT60
OFF	ON	ON	ON	ON	OFF	ON	0111101	PAT61
OFF	ON	ON	ON	ON	ON	OFF	0111110	PAT62
OFF	ON	ON	ON	ON	ON	ON	0111111	PAT63
ON	OFF	OFF	OFF	OFF	OFF	OFF	1000000	PAT64*
ON	OFF	OFF	OFF	OFF	OFF	ON	1000001	PAT65*
ON	OFF	OFF	OFF	OFF	ON	OFF	1000010	PAT66*
ON	OFF	OFF	OFF	OFF	ON	ON	1000011	PAT67*
ON	OFF	OFF	OFF	ON	OFF	OFF	1000100	PAT68*
ON	OFF	OFF	OFF	ON	OFF	ON	1000101	PAT69*
ON	OFF	OFF	OFF	ON	ON	OFF	1000110	PAT70*
ON	OFF	OFF	OFF	ON	ON	ON	1000111	PAT71*
ON	OFF	OFF	ON	OFF	OFF	OFF	1001000	PAT72*
ON	OFF	OFF	ON	OFF	OFF	ON	1001001	PAT73*
ON	OFF	OFF	ON	OFF	ON	OFF	1001010	PAT74*
ON	OFF	OFF	ON	OFF	ON	ON	1001011	PAT75*
ON	OFF	OFF	ON	ON	OFF	OFF	1001100	PAT76*
ON	OFF	OFF	ON	ON	OFF	ON	1001101	PAT77*
ON	OFF	OFF	ON	ON	ON	OFF	1001110	PAT78*
ON	OFF	OFF	ON	ON	ON	ON	1001111	PAT79*
ON	OFF	ON	OFF	OFF	OFF	OFF	1010000	PAT80*
ON	OFF	ON	OFF	OFF	OFF	ON	1010001	PAT81*
ON	OFF	ON	OFF	OFF	ON	OFF	1010010	PAT82*
ON	OFF	ON	OFF	OFF	ON	ON	1010011	PAT83*
ON	OFF	ON	OFF	ON	OFF	OFF	1010100	PAT84*
ON	OFF	ON	OFF	ON	OFF	ON	1010101	PAT85*
ON	OFF	ON	OFF	ON	ON	OFF	1010110	PAT86*
ON	OFF	ON	OFF	ON	ON	ON	1010111	PAT87*
ON	OFF	ON	ON	OFF	OFF	OFF	1011000	PAT88*
ON	OFF	ON	ON	OFF	OFF	ON	1011001	PAT89*
ON	OFF	ON	ON	OFF	ON	OFF	1011010	PAT90*
ON	OFF	ON	ON	OFF	ON	ON	1011011	PAT91*
ON	OFF	ON	ON	ON	OFF	OFF	1011100	PAT92*
ON	OFF	ON	ON	ON	OFF	ON	1011101	PAT93*
ON	OFF	ON	ON	ON	ON	OFF	1011110	PAT94*
ON	OFF	ON	ON	ON	ON	ON	1011111	PAT95*
ON	ON	OFF	OFF	OFF	OFF	OFF	1100000	PAT96*
ON	ON	OFF	OFF	OFF	OFF	ON	1100001	PAT97*
ON	ON	OFF	OFF	OFF	ON	OFF	1100010	PAT98*

ON	ON	OFF	OFF	OFF	ON	ON	1100011	PAT99*
ON	ON	OFF	OFF	ON	OFF	OFF	1100100	PAT100*
ON	ON	OFF	OFF	ON	OFF	ON	1100101	PAT101*
ON	ON	OFF	OFF	ON	ON	OFF	1100110	PAT102*
ON	ON	OFF	OFF	ON	ON	ON	1100111	PAT103*
ON	ON	OFF	ON	OFF	OFF	OFF	1101000	PAT104*
ON	ON	OFF	ON	OFF	OFF	ON	1101001	PAT105*
ON	ON	OFF	ON	OFF	ON	OFF	1101010	PAT106*
ON	ON	OFF	ON	OFF	ON	ON	1101011	PAT107*
ON	ON	OFF	ON	ON	OFF	OFF	1101100	PAT108*
ON	ON	OFF	ON	ON	OFF	ON	1101101	PAT109*
ON	ON	OFF	ON	ON	ON	OFF	1101110	PAT110*
ON	ON	OFF	ON	ON	ON	ON	1101111	PAT111*
ON	ON	ON	OFF	OFF	OFF	OFF	1110000	PAT112*
ON	ON	ON	OFF	OFF	OFF	ON	1110001	PAT113*
ON	ON	ON	OFF	OFF	ON	OFF	1110010	PAT114*
ON	ON	ON	OFF	OFF	ON	ON	1110011	PAT115*
ON	ON	ON	OFF	ON	OFF	OFF	1110100	PAT116*
ON	ON	ON	OFF	ON	OFF	ON	1110101	PAT117*
ON	ON	ON	OFF	ON	ON	OFF	1110110	PAT118*
ON	ON	ON	OFF	ON	ON	ON	1110111	PAT119*
ON	ON	ON	ON	OFF	OFF	OFF	1111000	PAT120*
ON	ON	ON	ON	OFF	OFF	ON	1111001	PAT121*
ON	ON	ON	ON	OFF	ON	OFF	1111010	PAT122*
ON	ON	ON	ON	OFF	ON	ON	1111011	PAT123*
ON	ON	ON	ON	ON	OFF	OFF	1111100	PAT124*
ON	ON	ON	ON	ON	OFF	ON	1111101	PAT125*
ON	ON	ON	ON	ON	ON	OFF	1111110	PAT126*
ON	ON	ON	ON	ON	ON	ON	1111111	PAT127*

Remote Message Selection

* Messages named PAT64 to PAT127 can be selected remotely if the input signal SEL_3 is configured for message selection. Please refer to the *general parameters* chapter for further information. If the SEL_3 is set as online for the marking head, only 63 programmes can be selected remotely. Obviously in this case, SEL_4 will behave like SEL_3, SEL_5 like SEL_4 and so on.

Appendix D

Alarm description table

ID	Alarm	Description	Action/Resolution
1	<i>Stop as user</i>	Alarm recorded when the user commands an abort of a marking cycle.	-
2	<i>Thermal</i>	Alarm recorded when a thermal alarm occurs in the controller.	<ul style="list-style-type: none"> • Ensure that there is no excessive motor absorption; • Check that the fan is turning and is not obstructed by external objects. • Check that there is sufficient air circulation to ensure cooling of the control unit; • Switch off the control unit and wait a few minute before switching back on.
3	<i>Power On</i>	Alarm recorded when the user presses the Power button on the controller.	-
4	<i>User abort</i>	Alarm recorded when a user stop occurs via an external button connected to the appropriate input.	<ul style="list-style-type: none"> • Check that there is no uncontrolled 24V signal on pin number 3 (IN_ABORT) of the connector I/O port.
5	<i>Abort print</i>	Alarm recorded when a user stop occurs via a logic signal command.	<ul style="list-style-type: none"> • Check that a 5V signal does not arrive in an uncontrolled manner on pin number 18 (ABORT) of the motor connector; • Check that it does not arrive in a non check a 5V signal on pin number 6 (ABORT) of the serial port connector;
6	<i>Laser system OK</i>	Alarm recorded on ALEA system when it is verified that the digital input of laser system ok is not high.	<ul style="list-style-type: none"> • Check that the connection cable to the laser head is correctly connected; • Check that the laser source is properly fed.
7	<i>Laser armed</i>	Alarm recorded when it is verified that the armed laser input is not after the laser enable has been commanded.	<ul style="list-style-type: none"> • Check that the connection cable to the laser head is correctly connected.
8	<i>Axes outside the check area</i>	Alarm recorded when stopping the execution of a programme if one of the axes went outside the boundaries of the programme's check area.	-
9	<i>Reserved</i>	-	-
10	<i>Reserved</i>	-	-
11	<i>Engine short circuit X</i>	Alarm recorded when the firmware detects a short circuit on motor X. It is necessary to ensure that the motors are connected and that there are no problems in the wiring.	<ul style="list-style-type: none"> • Check that the marking head cable is correctly connected to the motor port on the control unit; • Check the integrity of the motor cable inside the marking head; • Check that the motor is not damaged (short circuit or circuit

			<p>open);</p> <ul style="list-style-type: none"> • Check that the power/logic output management jumpers are correctly configured for the axis that must be commanded.
12	<i>Motor short circuit Y</i>	Alarm recorded when the firmware detects a short circuit on the Y motor. It is necessary to ensure that the motors are connected and that there are no problems in the wiring.	<ul style="list-style-type: none"> • Check that the marking head cable is correctly connected to the motor port on the control unit; • Check the integrity of the motor cable inside the marking head; • Check that the motor is not damaged (short circuit or open circuit); • Check that the power/logic output management jumpers are correctly configured for the axis to be controlled; • Check that the switching relay of the Y/Z channel is not blocked on the Z-axis control. It is possible from the test environment to check, if a Z-axis is available, if it moves or if changing the digital switching output produces a noise mechanics coming from the control unit.
13	<i>Modbus motor alarm</i>	Alarm recorded when an alarm occurs on a modbus motor.	<ul style="list-style-type: none"> • The alarm on the motor may be due to a loss of steps. Check that the axis is properly configured for the load to be moved and act on the speeds e ramps acceleration/deceleration ramps to avoid too abrupt movements and to ensure the necessary torque for the motor to perform the movements; • The motor alarm may be due to overheating of the drive. This particular alarm is automatically triggered by the drive when you exceed temperatures of over 90°C.
14	<i>Modbus motor</i>	Alarm recorded when any communication fault occurs on a modbus motor. When this emergency is displayed, the type of error encountered and the id of the motor concerned will be reported.	<ul style="list-style-type: none"> • Check that the motor is correctly connected to the RS485 of the control unit; • Check that the motor's network ID is correct and that it is the same as the one programmed on the drive; • Check that the communication baud rate set on the control unit for the RS485 channel (COM2) is the same as that of that programme on the drive.

Known hardware faults:

- BLUE LED OFF
 - check power supply fuse;
 - check led lighting on power supply.

Document:

INSTRUCTIONS FOR USE OF THE SOFTWARE

AUTOMATOR MARKING SYSTEMS SRL - Via Meucci n.8-20094 Corsico (MI) Italy
Tel 02 48601445 Fax 02 48601503 e-mail info@automator.com www.automator.com

In the event that none of these points related to known hardware faults resolve the highlighted problem, please contact the supplier for assistance.

Appendix E

Touch Screen Calibration Procedure

To calibrate the touch screen, an *empty* text file (without any content inside) called **calibrate** must be saved on a USB stick. It is important that the file does not contain any extensions in the name.

Once the file has been placed on the USB stick, it must be inserted on the controller and restarted with the stick still inserted. After booting the controller, the touch screen calibration procedure will be started. When the calibration is complete, the application will be started.

Appendix F

Object Dictionary Modbus Slave function

For any reference on command types (Type O, Type G, etc.) please refer to the section on host communication via serial in the extended protocol section.

Address	Object	Type	Description	Rights
0x5000	Type O	BYTE	Setting the system's online status. Entering the register with the value 1 will start the system's online procedure.	W
0x5001	Type G	BYTE	Marking cycle start setting. Writing the register with the value 1 will start the marking cycle start procedure.	W
0x5002	Type I	BYTE	Request information on the status of exits.	R
0x5003	Type S	WORD	Requesting information on error status.	R
0x5100	Type 1 Write Word	BYTE	Enabling write command Type 1	W
0x5101	Type 1 Write Field number	WORD	Number of the field to be modified.	W
0x5102-0x51FF	Type 1 Write String	WORD	Value of the field to be modified. Each register will contain two ASCII characters. If character 0 is found, the analysis of the value will be terminated.	W
0x5201	Type 1 Read Field number	WORD	Number of the field to be read.	W
0x5202-0x52FF	Type 1 Read String	WORD	Value of the field to be read. Each register will contain two ASCII characters. A 0 character will be inserted at the end of the string.	R
0x5300	Type V Write Word	BYTE	Type V command write enable	W
0x5301	Type V Write Field number	WORD	Number of the field to be modified.	W
0x5302-0x53FF	Type V Write String	WORD	Value of the field to be modified. Each register will contain two ASCII characters. If the character 0 is found, the value analysis will be terminated.	W
0x5401	Type V Read Field number	WORD	Number of the field to be read.	W
0x5402-0x54FF	Type V Read String	WORD	Value of the field to be read. Each register will contain two ASCII characters. A 0 character will be inserted at the end of the string.	R
0x5500	Type Q Write Word	BYTE	Type Q command write enable	W

0x5501	Type Q Write Field Number	WORD	Number of the query to be modified.	W
0x5502-0x55FF	Type Q Write String	WORD	Value of the query to be modified. Each register will contain two ASCII characters. If character 0 is found, the value analysis will be terminated.	W
0x5600	Type P Write Word	BYTE	Type P command write enable	W
0x5601-0x56FF	Type P Write String	WORD	Value of the programme to be loaded. Each register will contain two ASCII characters. If the character 0 is found, the value analysis will be terminated.	W
0x5701-0x57FF	Type P Read String	WORD	Value of the programme currently loaded on the controller. Each register will contain two ASCII characters. At the end character 0 will be inserted in the string.	R

Example Online Head Command

1. Writing the value **1** of in register **0x5000**.

Example Start command marking current programme

1. Writing the value **1** of in register **0x5001**.

Example Reading Field

1. Writing the number of the field to be read into register **0x5201**.
2. Register reading **0x5202-0x52FF**.

Register:	0x5202	0x5203	0x5204
ASCII value (Hex):	54 45	53 54	00 00
String:	T E	S T	/0 /0

Example WRITING Field

2. Write field number to register **0x5101**.
3. Register write **0x5102-0x51FF**.

Register:	0x5102	0x5103	0x5104
ASCII value (Hex):	54 45	53 54	31 00
String:	T E	S T	1 /0

4. Write value **1** into register **0x5100**.

Appendix G

EtherNet/IP Fieldbus Module

EtherNet/IP

EtherNet/IP (Ethernet Industrial Protocol) is a communications protocol developed by [Rockwell Automation](#), managed by the [Open DeviceNet Vendors Association](#) (ODVA) and designed for use in process control and other industrial [automation](#) applications.

quotation from en.wikipedia.org

Description

EtherNet/IP is an application layer protocol treating devices on the network as a series of "objects". EtherNet/IP is built on the [Common Industrial Protocol](#) (CIP), for access to objects from ControlNet and DeviceNet networks.

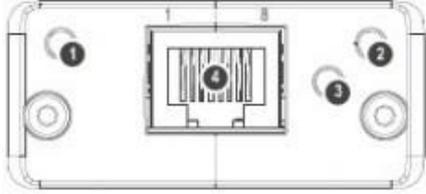
EtherNet/IP can be used in automation networks which can tolerate some amount of non-determinism. This is because Ethernet physical media might not have deterministic delays.

EtherNet/IP can be easily confused as a simple combination of Ethernet and the [Internet Protocol](#). Instead, it is an industrial application layer protocol used for communication between industrial control systems and their components, such as a [programmable automation controller](#), [programmable logic controller](#) or an [I/O](#) system. The "IP" in EtherNet/IP, is not an abbreviation for "Internet Protocol" but instead stands for "Industrial Protocol".

quotation from en.wikipedia.org

The EtherNet/IP port on AC500

The EtherNet/IP port on an AC500 unit.

	Item	
1	Network Status LED	
2	Module Status LED	
3	Links/Activity	
4	Ethernet Interface	

Network status LED.

Note: An LED lighting sequence is done when the module is switched on.

LED State	Description
Off	No power or no IP address
Green	On-line, one or more connections established (CIP Class 1 or 3)
Green, flashing	On-line, no connections established
Red	Duplicate IP address, FATAL error
Red, flashing	One or more connections timed out (CIP Class 1 or 3)

Module status LED.

Note: An LED lighting sequence is done when the module is switched on.

LED State	Description
Off	No power
Green	Controlled by a Scanner in Run state
Green, flashing	No configured or Scanner in Idle state
Red	Major fault (EXCEPTION -state, FATAL error etc...)
Red, flashing	Recoverable fault(s)

Link/activity LEDs

LED State	Description
Off	Link, no activity
Green	Established links
Green, flickering	Activity

Configuring EtherNet/IP

Before starting to use the EtherNet/IP communication module on the control unit, you must ensure that you have performed all the necessary steps to configure it correctly.

First, it will be necessary to access the auxiliary functionality environment:



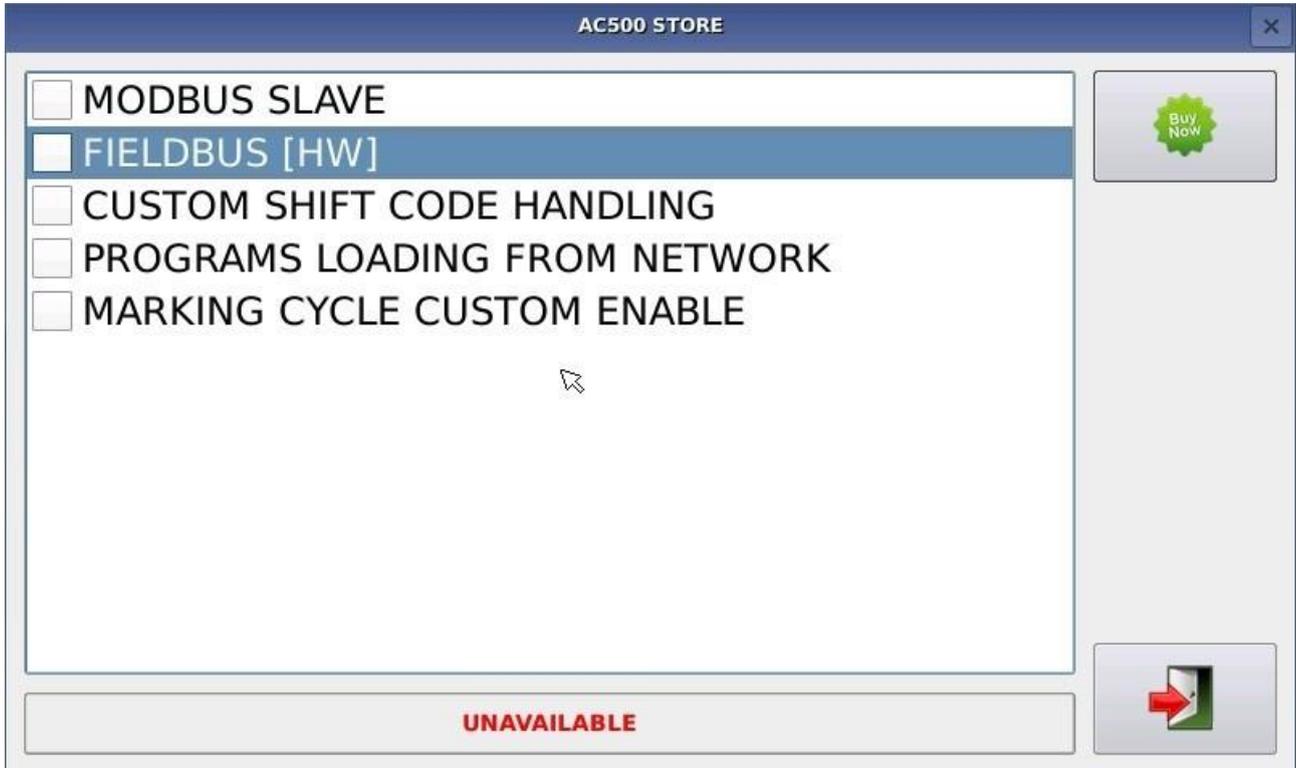
Access the system configuration environment from the auxiliary functionality menu.



Once you have entered the system configuration environment, press the **AC500 STORE** button to access the environment that will allow you to purchase the Fieldbus protocol management functionality:



Select the Filedbus entry to unlock the management of Fieldbus protocols, and press the button on the right:



On the screen that will be displayed, you will be asked to send the given code to the controller supplier. Once this is done, you will be notified of the code to be entered to proceed with the purchase of the function.

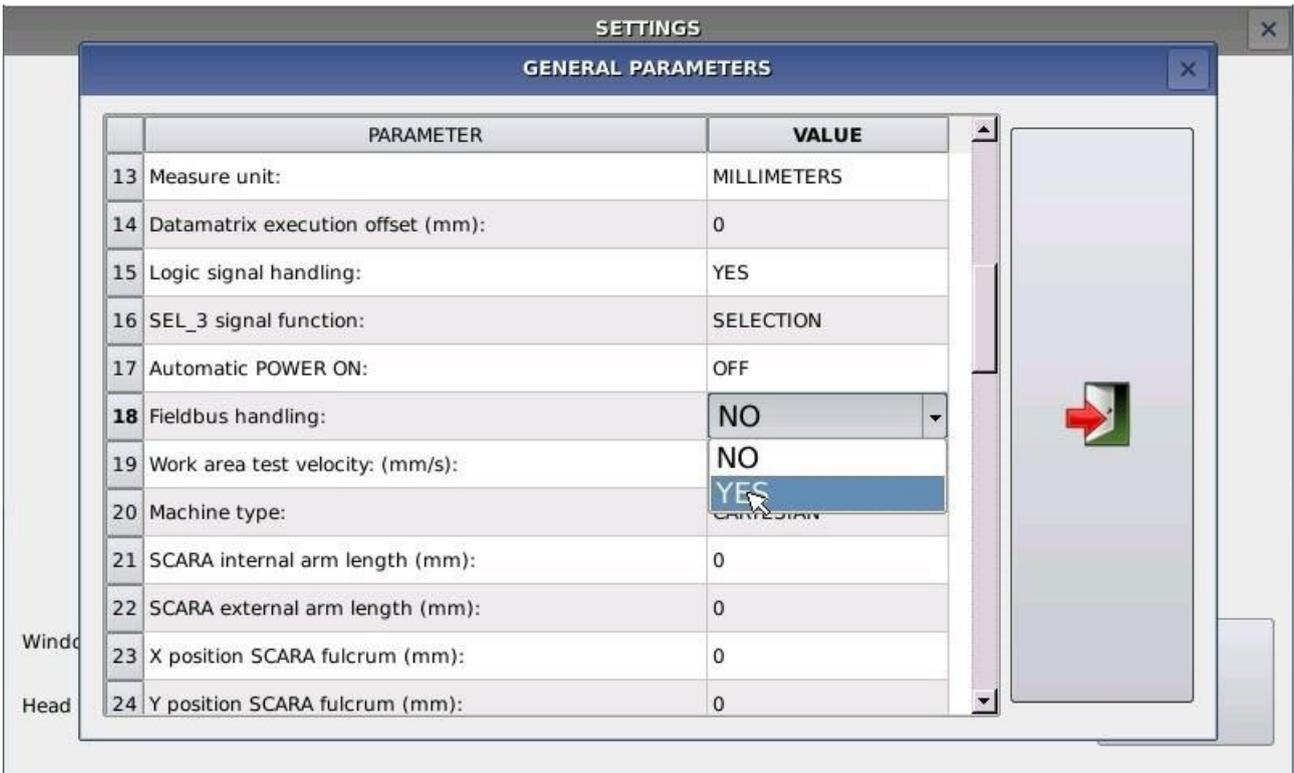


Once the unlock code has been entered, the **ENTER** button must be pressed. If the code is correct, confirmation that the operation was successful will be displayed.



Now it will be possible to go to the general data configuration environment, contained in the system settings environment, and enable the **Fieldbus Management** parameter.

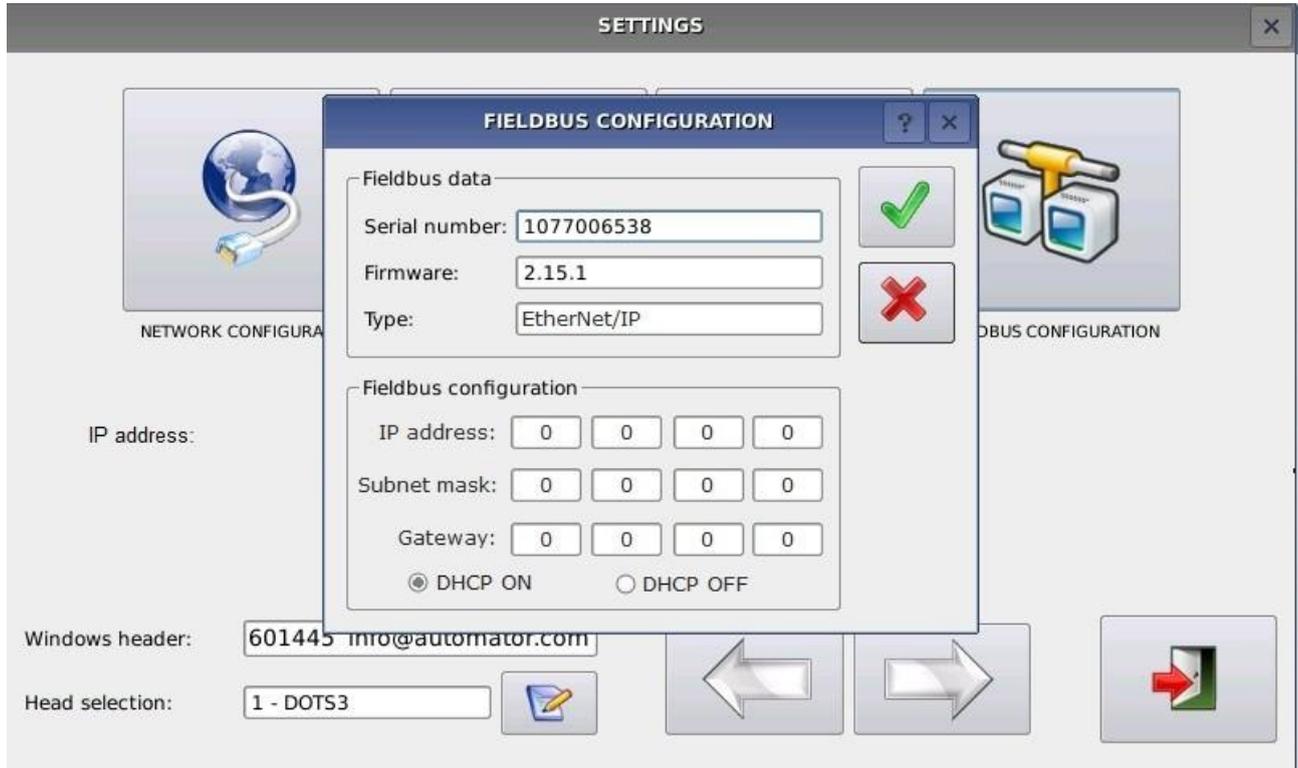




Once Fieldbus protocol management is enabled, it will be possible to access the configuration environment:



In this environment, it will be possible to check whether the module has been correctly connected and enter the necessary data to configure it correctly:



While proceeding with the input of the module configuration data, it is necessary to keep the following things in mind:

- If the *serial number* and *firmware* version are set to 0 (zero) and the *module is unknown*, this means that the AC500 Series driver has failed to recognise the fieldbus module; in this case, the control unit must be restarted and the module connections must be checked to ensure that they have been made correctly.
- An IP address already set on the Ethernet port of the control unit must not be set.

After completing the module data configurations, you will be asked to restart the control unit to complete the initialisation procedure.

EXTENDED PROTOCOL - DESCRIPTION OF COMMANDS

Command	Description
Type 1	Command to overwrite the contents of an individual field with the data contained in the transmitted packet. The command contains the number of the field to be changed in the format nn (01,02,...,99). In read mode, the command allows the contents of a specific field of the currently loaded programme to be received from the controller.
Type G	Command to tell the control unit to start a

	cycle.
Type I	Command to receive system output states (DONE, READY, PAUSE, ONLINE and ERROR outputs).
Type O	Command to tell the control unit to go online.
Type P	<p>Command to communicate the loading of a programme with the same name as that contained in the package.</p> <p>In read mode, the command allows you to receive the name of the programme currently loaded on the control unit.</p>
Type Q	Command allowing the text of a query on the control unit to be changed. The command contains the number of the query to be edited in the format nn (01,02,...,99).
Type S	Command to receive the error status of the control unit. There are a number of codes indicating the type of error that has occurred.
Type V	<p>Command enabling the contents of a variable field to be modified. The command contains the number of the field to be modified in the format nn (01,02,...,99).</p> <p>In read mode, the command allows the contents of a variable field to be received.</p>
Type H	<p>Command enabling the height of a text field or an arc text field to be changed. The command contains the number of the field to be changed in the format nn (01,02,...,99).</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>
Type C	<p>Command to change the position of a text field, arc text field and DXF. The command contains the number of the field to be changed in the format nn (01,02,...,99).</p> <p>The two different values to be set for the X and Y coordinates must be separated in this string by the character ';'. ;</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>
Type D	<p>Command enabling the diameter of an arc text field to be changed. The command contains the number of the field to be modified in the format nn (01,02,...,99).</p> <p>This type of command is only available if it is</p>

	advanced management of fieldbus protocols from the AC500 Store was unlocked.
Type B	Command allowing the start angle of an arc text field to be changed. The command contains the number of the field to be changed in the format nn (01,02,...,99). The value is represented in degrees. This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.
Type M	Command to read, for a specific marking field, the text that will be marked, decoding any flags in it. (e.g. %A → 'Mon'). This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.

ETHERNET/IP ADIs Object (A2h)

This object is used to manage the data to be passed to the AC500 application. The CIP Object ADIs (Application Data Instances, mapped with address 0xA2) is used as a medium for exchanging data with the AC500 application. Each instance corresponds to a different extended protocol command. Each instance consists of several properties, called *attributes*.

#	Name	Access	Type	Description
1	Name	Get	Array of CHAR	Parameter name
2	Data type	Get	UINT8	Data type of instance value
3	No. of elements	Get	UINT8	Number of elements of the specified data type
4	Description	Get	UINT8	Bit field describing the access rights for this instance Bit:Meaning: 0 Set = Get Access 1 Set = Set Access
5	Value	Get/Set	Determined by attribute #2	Instance value.
6	Max value	Get		The maximum permitted parameter value (NOT USED)
7	Min value	Get		The minimum permitted parameter value. (NOT USED)
8	Default value	Get		The default parameter value. (NOT USED)

To set or read each command value associated with each instance (see object dictionary table), reference is made to the *value* attribute, i.e. attribute number 5.

ETHERNET/IP Get/Set Attribute Single

There are two data access commands, both read and write, used with the EtherNet/IP model of the AC500:

- **Get Attribute Single (0x0E):** command for reading an instance attribute;
- **Set Attribute Single (0x10):** command for writing an instance attribute.

The EtherNet/IP model set up on the AC500 only handles these two commands for resource access operations, which means that all other commands are currently not handled and are not recommended.

OBJECT DICTIONARY - ETHERNET/IP

CIP Object	Instance	Attribute	Object	Type	Description	Rights
0xA2	1	5	Type I	UINT16	Requesting information on output status.	R
0xA2	2	5	Type S	UINT16	Requesting information on error status. The upper part of the WORD will be placed in the first register, and the lower part in the second.	R
0xA2	3	5	Type O	UINT8	Setting the system's online status. Setting the least significant bit to 1 will send the system online. ALL other values passed will be ignored. <i>Example: 01h (online system)</i>	W
0xA2	4	5	Type G	UINT8	Marking cycle start setting. Setting the least significant bit to 1 will launch the start of the cycle. ALL other values passed will be ignored. <i>Example: 01h (online system)</i>	W
0xA2	5	5	Type 1	Array of CHAR	Instance dedicated to the Type 1 command. Through this instance, it will be possible to perform read and write operations on the programme fields. The maximum size of the string passed is 255 characters.	R/W

0xA2	6	5	Type V	Array of CHAR	<p>Instance dedicated to the Type V command.</p> <p>Through this instance, it will be possible to perform read and write operations on the variable fields of the programme.</p> <p>The maximum size of the string passed is 255 characters.</p>	R/W
0xA2	7	5	Type P	Array of CHAR	<p>Instance dedicated to the Type P command.</p> <p>Through this instance it will be possible to perform read and load programme operations. By reading it will be possible to request the name of the programme loaded on the control unit, and by writing a value it will be possible to load that programme.</p> <p>The maximum size of the string passed is 255 characters.</p>	R/W
0xA2	8	5	Type Q	Array of CHAR	<p>Instance dedicated to the Type Q command.</p> <p>Through this instance, it will be possible to perform read and write operations on a query of the control unit.</p> <p>The maximum size of the string passed is 255 characters.</p>	R/W
0xA2	9	5	Type 1 FIELD	UINT8	<p>Instance dedicated to the field number to be taken into account when performing read/write operations on instance 5.</p> <p>The value set here will be used to specify the number of the field that will be read/modified with instance 5.</p>	W
0xA2	10	5	Type V FIELD	UINT8	<p>Instance dedicated to the field number to be taken into account when performing read/write operations on</p>	W

Document:

INSTRUCTIONS FOR USE OF THE SOFTWARE

AUTOMATOR MARKING SYSTEMS SRL - Via Meucci n.8-20094 Corsico (MI) Italy
Tel 02 48601445 Fax 02 48601503 e-mail info@automator.com www.automator.com

					instance 6.	
--	--	--	--	--	-------------	--

					The value set here will be used to specify the number of the field that will be read/modified with instance 6.	
0xA2	11	5	Type Q FIELD	UINT8	Instance dedicated to the number of the field to be taken into account when performing write operations on instance 8. The value set here will be used to specify the number of the field which will be modified by instance 8.	W
0xA2	12	5	Type H	Array of CHAR	Instance dedicated to the Type H command type. This instance allows read and write operations to be performed on the height value of a field of type text or arc text. This datum will contain the height value in text string format. This means that if the current height value is 4, the value of this datum will be the string '4'. The maximum size of the string passed is 255 characters. This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.	R/W
0xA2	13	5	Type H FIELD	UINT8	Instance dedicated to the field number to be taken into account when performing read/write operations on instance 12. The value set here will be used to specify the number of the field to be read/modified with instance 12. This type of command is only available if it has been	W

					unlocked the advanced management of fieldbus protocols from the AC500 Store.	
0xA2	14	5	Type C	Array of CHAR	<p>Instance dedicated to the Type C command type.</p> <p>This instance allows read and write operations to be performed on the position of a text, arc text and DXF type field.</p> <p>This data will contain the position values in text string format. The position values in XY contained in the string will be separated by the character ';'. This means that if the current position value is x = 10 and y = 20, the value of this datum will be the string '10;20'.</p> <p>The maximum size of the string passed is 255 characters.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>	R/W
0xA2	15	5	Type C FIELD	UINT8	<p>Instance dedicated to the field number to be taken into account when performing read/write operations on instance 14.</p> <p>The value set here will be used to specify the number of the field to be read/modified with instance 14.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>	W
0xA2	16	5	Type D	Array of CHAR	<p>Instance dedicated to the Type D command type.</p> <p>This instance allows</p>	R/W

					<p>perform read and write operations on the arc diameter of an arc text type field. The arc of the field must necessarily have been created by degree mode.</p> <p>This datum will contain the diameter value in text string format. This means that if the current diameter value is 30, the value of this datum will be the string '30'.</p> <p>The maximum size of the string passed is 255 characters.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>	
0xA2	17	5	Type D FIELD	UINT8	<p>Instance dedicated to the field number to be taken into account when performing read/write operations on instance 16.</p> <p>The value set here will be used to specify the number of the field to be read/modified with instance 16.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>	W
0xA2	18	5	Type B	Array of CHAR	<p>Instance dedicated to the Type B command type.</p> <p>This instance allows read and write operations to be performed on the starting angle of the arc of an arc text type field. The arc of the field must necessarily have been created by degree mode.</p> <p>This data will contain the value</p>	R/W

					<p>of the arc's starting angle in text string format. This means that if the value of the current starting angle is 90, the value of this data will be the string '90'. The value is represented in degrees.</p> <p>The maximum size of the string passed is 255 characters.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>	
0xA2	19	5	Type B FIELD	UINT8	<p>Instance dedicated to the field number to be taken into account when performing read/write operations on instance 18.</p> <p>The value set here will be used to specify the number of the field that will be read/modified with instance 18.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>	W
0xA2	20	5	Type M	Array of CHAR	<p>Instance dedicated to Type M command type.</p> <p>This instance reads the string that will be marked during the marking cycle.</p> <p>The maximum size of the string is 255 characters.</p>	R
0xA2	21	5	Type M FIELD	UINT8	<p>Instance dedicated to the Type M FIELD command type.</p> <p>This instance allows the field number associated with instance 20 to be specified.</p>	W

EtherNet/IP connected to AC500

Once the control unit is correctly configured, you can use the object dictionary above to perform all queries that the AC500 software proposes. It is possible to connect to the EtherNet/IP module located on the AC500 series control unit via a standard Ethernet cable.

Examples of questionsREADING example

1. Writing the number of the field to be read into the **Type 1 FIELD** instance.
2. Reading of **Type 1** instance.

Array Index:	0	1	2	3
ASCII value (Hex):	54	45	53	54
String:	T	E	S	T

Example WRITING Field

1. Writing the number of the field to be written in the **Type 1 FIELD** instance.
2. Writing of **Type 1** instance with the following values.

Array Index:	0	1	2	3	4
ASCII value (Hex):	54	45	53	54	31
String:	T	E	S	T	1

EtherNet/IP implicit model

Reading area

Instance	Object	Type	Description
1	Type I	UINT16	Information on the status of outputs.
2	Type S	UINT16	Information on error status.
3	Read Cmd Type	UINT16	Returns the current status of the command type being read.
4	Read Id	UINT16	Returns the index of the field or query being read. Used only for FIELD, VARIABLE and QUERY type commands.
5	Read Flags	UINT16	Returns the status information of the string being read. The first byte reports the status (0 = not ready, 1 = ready) while the second byte reports the length of the string being read.
6	<i>Unused</i>	Arrays of UINT8 [6].	6 bytes free for future use.
7	Read String	Array of UINT8 [64]	Returns the contents of the string being read. Combined with flags it is possible to read the number of bytes corresponding to the string being read.

Representation:

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32

33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48

49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64

65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80

01-02: Report output status (Type I)
 03-04: Reports information status (Type S) 05-
 06: READ-CMD Current Type
 07-08: READ Current ID
 09-10: READ-String Flags.
 11-16: Unitised
 17-80: Read String data

Writing area

Instance	Object	Type	Description
1	Read Cmd Type	UINT16	Sets the type of command to be used for read operations.
2	Read Id	UINT16	Sets the index of the field or query to be used for read operations. Used only for FIELD, VARIABLE and QUERY type commands.
3	Type O	UINT8	Setting the system online. To execute the command, this BYTE must be set to a value of 1.
4	Type G	UINT8	Setting the start marking. To execute the command is It is necessary to set this BYTE to a value of 1.
5	Write Cmd Type	UINT16	Sets the type of command to be used for write operations.
6	Write Id	UINT16	Sets the index of the field or query to be used for write operations. Used only for FIELD, VARIABLE and QUERY type commands.
7	Write Flags	UINT16	Inposes the status information of the string being written. The first byte reports the status (0 = not ready, 1 = ready) while the second byte reports the length of the string being written.
8	<i>Unused</i>	Arrays of UINT8 [4].	4 bytes free for future use.
9	Write String	Array of UINT8 [64]	Sets the content of the string being written. Combined with flags it is possible to write the number of bytes corresponding to the string to be written.

Representation:

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80

01-02: Set READ-CMD Type 03-
04: Set READ ID
05: Online setting
06: Set start of a cycle
07-08: Set WRITE-CMD Type 09-
10: Set WRITE ID
11-12: Set WRITE-String Flags. 13-
16: Unused
17-80: Write String data

STRING MANAGEMENT:

The operation of the string management logic via the I/O data of the Fieldbus protocols is described below. Using a few simple support bytes, it is possible to dynamically configure read and write operations on string data types. The bytes implemented for string management make it possible to define the type of string on which to operate (program name, program field, query, etc.), the id of the field that can be requested, and flags to indicate the status of the operation and the length of the string set.

- COMMAND Type

This data is used to identify which type of string data is to be used in the read/write operation.

- 0) NULL = No string
- 1) PATTERN = String referring to programmes
- 2) TEXT = String referring to the fields of a programme
- 3) VARIABLE = String referring to the variable fields of a programme
- 4) QUERY = String referring to queries
- 5) TEXT HEIGHT = String referring to the numeric value of the height of a text field (*)
- 6) TEXT POSITION = String referring to the numerical values of the XY positions of a text field (the two values must be separated by the character ';') (*)
- 7) ARC TEXT DIAMETER = String referring to the numerical value of the diameter of an arc in the arc text field (*)
- 8) ARC TEXT START ANGLE = String referring to the numerical value of the start angle (in degrees) of an arc in the arc text field (*)
- 9) MARKED TEXT = String referring to the text to be marked in the work programme (*)

(*) = These commands are only available after unlocking the Advanced Fieldbus Protocol Management functionality from the AC500 Store.

- ID

Optional data element, to be used only in the TEXT, VARIABLE and QUERY command types and is used to identify the number of the field on which the read/write operation is to be performed.

- String Flags

This data is used to identify information about the read/write string. The first byte is used to inform whether the string is ready to be read/written or not, while the second byte is used to define the length of the string in the data area.

Example:

A number of bytes have been dedicated in the write area to define what the PLC wants to read from the AC500. By setting this data, the bytes in the read area will then be modified based on the configuration set in the write area.

For example, suppose you want to read the programme currently loaded on the AC500, you will need to write the bytes for the read string configuration in the following way:

WRITE AREA

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
00	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00

The AC500 will then modify the bytes in the reading area in this way:

READ AREA

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
00	00	00	00	00	01	00	00	00	04						

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
84	69	83	84												
(T)	(E)	(S)	(T)												

Suppose, for example, that you want to read the third field of a programme:

WRITE AREA

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
00	02	00	03	00	00	00	00	00	00	00	00	00	00	00	00

The AC500 will then modify the bytes in the reading area in this way:

READ AREA

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
00	00	00	00	00	02	00	03	00	09						

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
84	69	83	84	70	73	69	76	68							
(T)	(E)	(X)	(T)	(F)	(I)	(E)	(L)	(D)							

Conversely, if you want to go and load the **AC500** programme, you will have to configure the writing area in this way:

WRITE AREA

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
00	00	00	00	00	00	00	01	00	00	01	05	00	00	00	00

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
65 (A)	67 (C)	53 (5)	48 (0)	48 (0)	00	00	00	00	00	00	00	00	00	00	00

Attention: the AC500 will only perform the operation on the read string when the Write Flags instance status flag (eleventh byte of the representation) is set to **1**.

MANAGEMENT EDITING PROGRAMME FIELDS

The protocol managed by the AC500 can be used to change the format of fields within a marking programme. This can be done using the TYPE H, TYPE C, TYPE D and TYPE B commands.

All these commands work by string-formatting numeric values that must be used for each type of command associated with an editing functionality. The following are some simple examples that will give a better understanding of how these particular commands work.

Attention: commands for editing programme fields can only be used after unlocking the 'EXTENDED PROTOCOL / ADVANCED FIELDBUS' functionality.

Reading the height of a text field (TEXT HEIGHT)

- set the number of the field you wish to modify (refer to the object dictionary table above);
- read the data referring to the TYPE H command type;
- this data is a character string containing within it the numeric value of the height of the required text field, such as:
"4" means that the text height is 4mm (proprietary text) or 4 points (TTF text).

Writing the height of a text field (TEXT HEIGHT)

- set the number of the field you wish to modify (refer to the object dictionary table above);
- write the data referring to the TYPE H command type;
- this data is a character string containing within it the numeric value of the height of the required text field, such as:
"6" means that the text height is 6mm (proprietary text) or 6 points (TTF text).

Reading the position of a text field (TEXT POSITION)

- set the number of the field you wish to modify (refer to the object dictionary table above);
- read the data referring to the TYPE C command type;
- this data is a character string containing within it the numeric values of the position in XY separated by a ';' character, such as:
"20;15.5" means that the position is X=20mm and Y=15.5mm.

Writing the position of a text field (TEXT POSITION)

- set the number of the field you wish to modify (refer to the object dictionary table above);
- write the data referring to the TYPE C command type;
- this data is a character string containing within it the numeric values of the position in XY separated by a ';' character, such as:
"40;10" means that the position is X=40mm and Y=10mm.

Reading the Diameter of an Arc Text Field (ARC TEXT DIAMETER)

- set the number of the field you wish to modify (refer to the object dictionary table above);
- read the data referring to the TYPE D command type;

- this data is a character string containing within it the numerical value of the diameter of the required arc text field, e.g:
"30.6" means that the diameter of the arch text is 30.6mm.

Writing the diameter of an arc text field (ARC TEXT DIAMETER)

- set the number of the field you wish to modify (refer to the object dictionary table above);
- write the data referring to the TYPE D command type;
- this data is a character string containing within it the numerical value of the diameter of the required arc text field, e.g:
"40" means that the diameter of the arch text is 40mm.

Reading the starting angle of an arc text field (ARC TEXT START ANGLE)

- set the number of the field you wish to modify (refer to the object dictionary table above);
- read the data referring to the TYPE B command type;
- this data is a character string containing within it the numerical value of the starting angle of the required arc text field, e.g:
"180.7" means that the angle of the arc text is 180.7°.

Writing the starting angle of an arc text field (ARC TEXT START ANGLE)

- set the number of the field you wish to modify (refer to the object dictionary table above);
- write the data referring to the TYPE B command type;
- this data is a character string containing within it the numerical value of the starting angle of the required arc text field, e.g:
"90" means that the angle of the arch text is 90°.

Appendix H Profibus Fieldbus Module

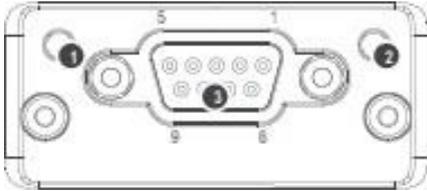
Profibus

PROFIBUS (Process Field Bus) is a standard for [field bus](#) communication in [automation technology](#) and was first promoted in 1989 by [BMBF](#) (German department of education and research) and then used by [Siemens](#). It should not be confused with the [PROFINET standard](#) for [Industrial Ethernet](#). PROFIBUS is not an openly published and royalty-free protocol as older ones like [MODBUS](#).

quotation from en.wikipedia.org

The Profibus port on AC500

The Profibus port on an AC500 unit.

	Item	
1	Operation Mode	
2	Status	
3	Profibus Connector	

Operation Mode.

State	Indication	Comments
Off	Not online / No power	-
Green	Data exchange	-
Flashing Green	Clear	-
Flashing Red (1 flash)	Parametrization error	Error while configuring the device.
Flashing Red (2 flashes)	PROFIBUS Configuration error	Configuration error between MASTER/SLAVE Profibus devices.

Status.

State	Indication	Comments
Off	Not initialised	Anybus state = 'SETUP' or 'NW_INIT'.
Green	Initialised	Anybus module has left the 'NW_INIT' state
Flashing Green	Initialised, diagnostic event(s) present	Extended diagnostic bit is set
Red	Exception error	Anybus state = 'EXCEPTION'

Profibus Connector (DB9F)

Pin	Signal	Description
1	-	-
2	-	-
3	B Line	Positive RxD/TxD, RS485 level
4	RTS	Request to send
5	GND Bus	ground (isolated)
6	+5 V Bus Output	+5 V termination power (isolated, short-circuit protected)
7	-	-
8	A Line	Negative RxD/TxD, RS485 level
9	-	-
Housing	Cable Shield	Internally connected to the Anybus protective earth via cable shield filters according to the PROFIBUS standard.

Profibus configuration

Before you start using the Profibus communication module on the control unit, you must ensure that you have performed all the necessary steps to configure it correctly.

First, it will be necessary to access the auxiliary functionality environment:



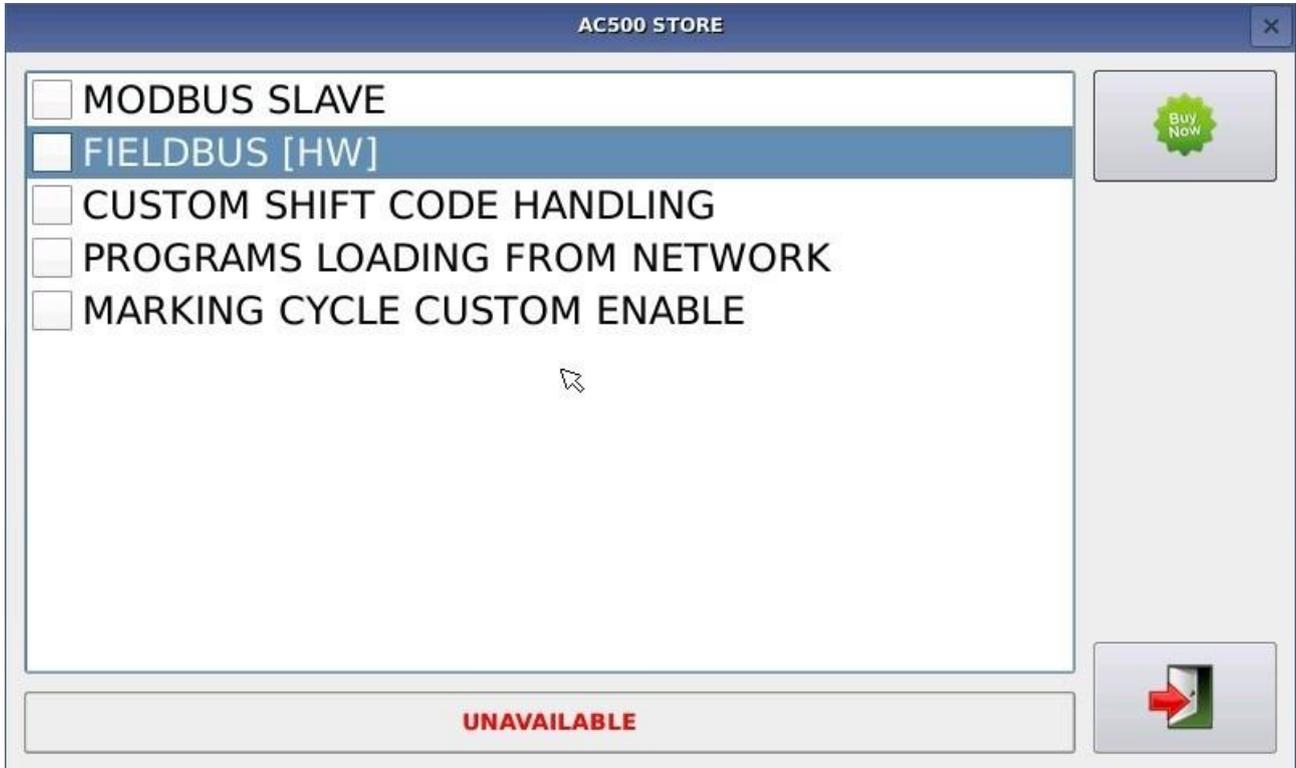
Access the system configuration environment from the auxiliary functionality menu.



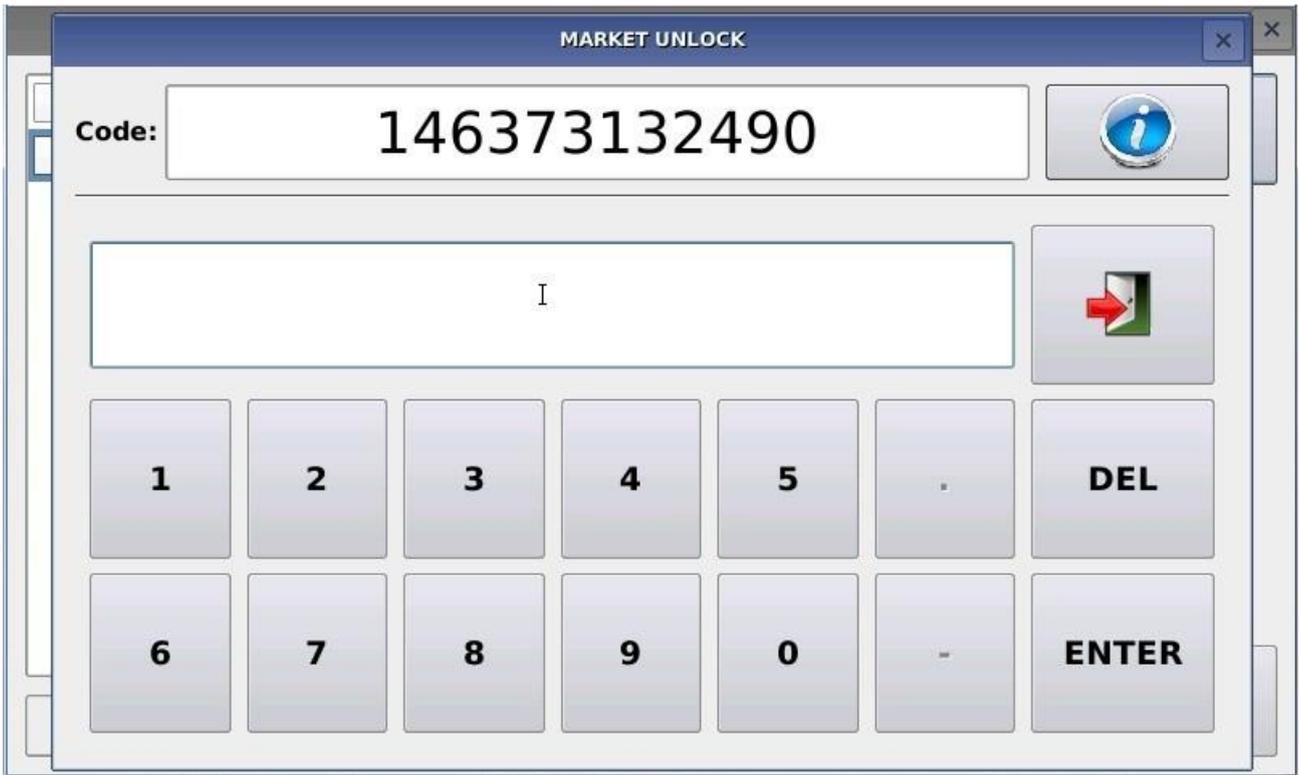
Once you have entered the system configuration environment, press the **AC500 STORE** button to access the environment that will allow you to purchase the Fieldbus protocol management functionality:



Select the Filedbus entry to unlock the management of Fieldbus protocols, and press the button on the right:



On the screen that will be displayed, you will be asked to send the given code to the controller supplier. Once this is done, you will be notified of the code to be entered to proceed with the purchase of the function.

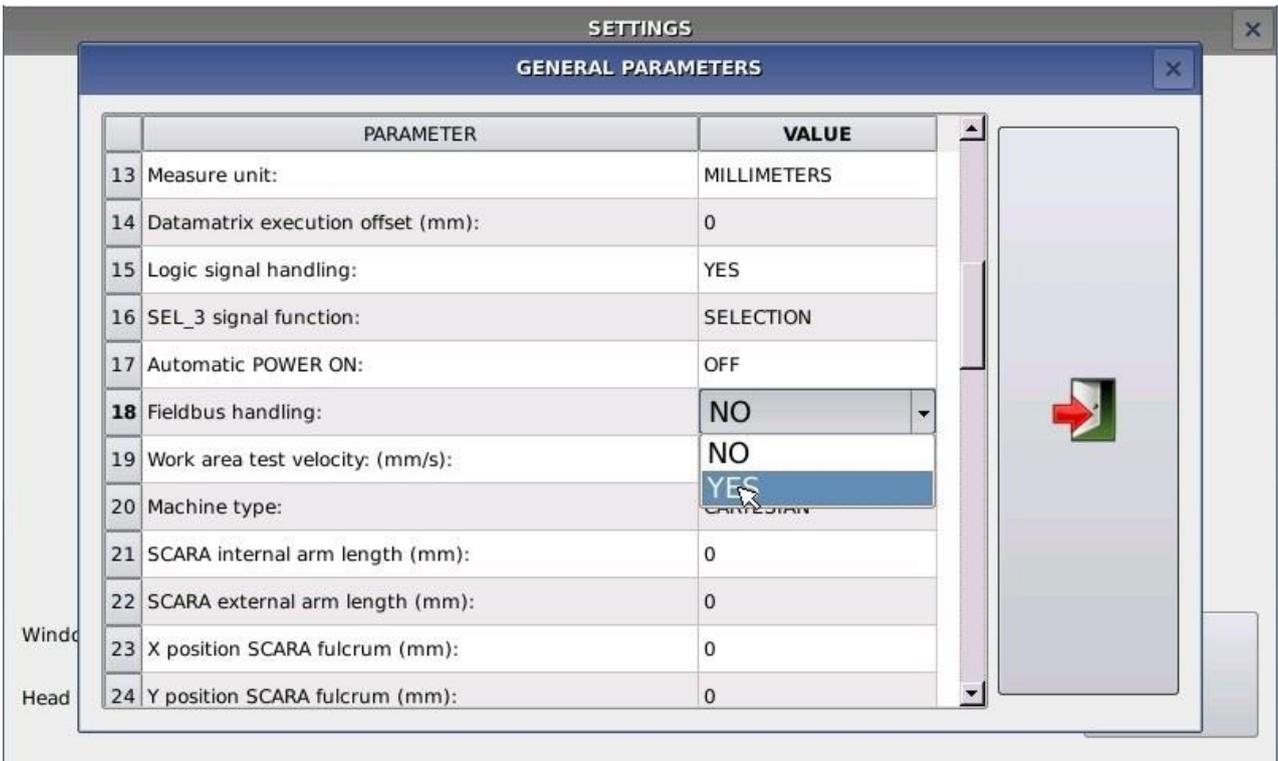


Once the unlock code has been entered, the **ENTER** button must be pressed. If the code is correct, confirmation that the operation was successful will be displayed.



INSTRUCTIONS FOR USE OF THE SOFTWARE

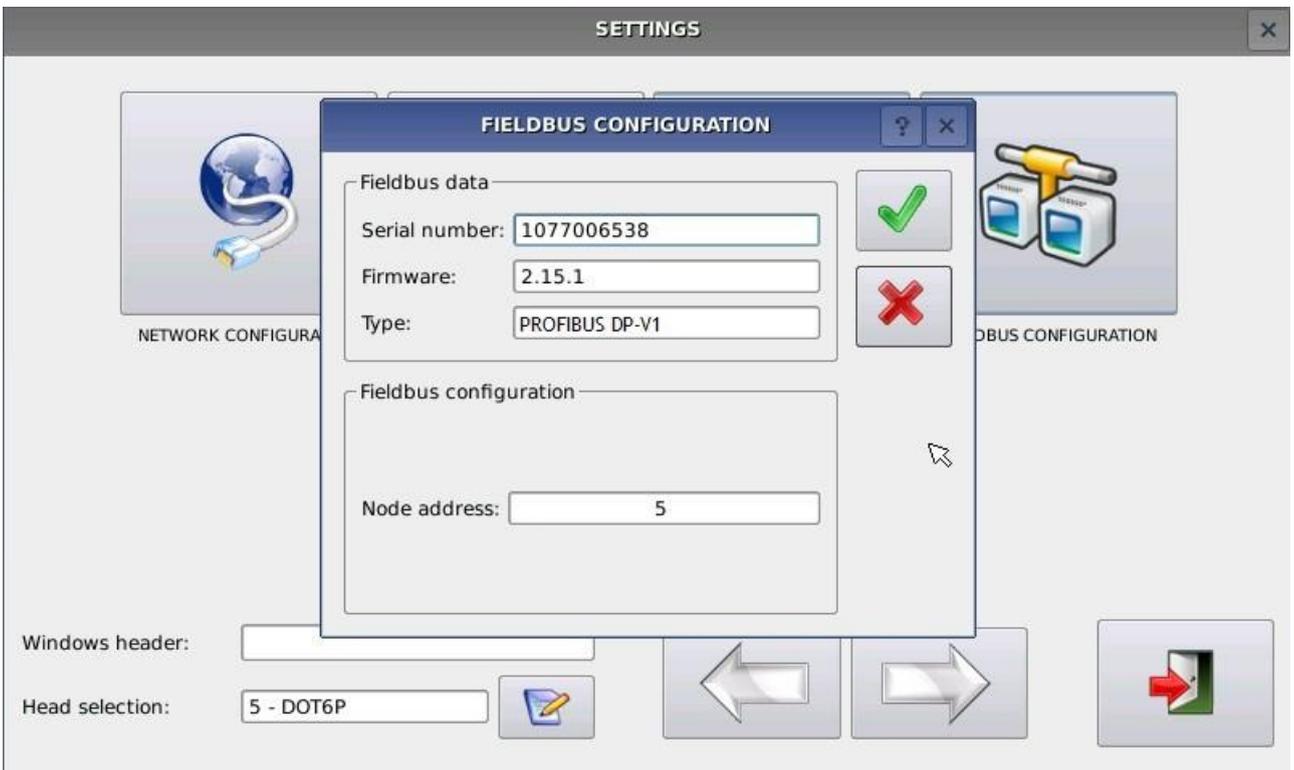
Now it will be possible to go to the general data configuration environment, contained in the system settings environment, and enable the **Fieldbus Management** parameter.



Once Fieldbus protocol management is enabled, it will be possible to access the configuration environment:



In this environment, it will be possible to check whether the module has been correctly connected and enter the necessary data to configure it correctly:



While proceeding with the input of the module configuration data, it is necessary to keep the following things in mind:

- If the *serial number* and *firmware* version are set to 0 (zero) and the *module is unknown*, this means that the AC500 Series driver has failed to recognise the fieldbus module; in this case, the control unit must be restarted and the module connections must be checked to ensure that they have been made correctly.

After completing the module data configurations, you will be asked to restart the control unit to complete the initialisation procedure.

EXTENDED PROTOCOL - DESCRIPTION OF COMMANDS

Command	Description
Type 1	Command to overwrite the contents of an individual field with the data contained in the transmitted packet. The command contains the number of the field to be changed in the format nn (01,02,...,99). In read mode, the command allows the content of a specific field of the currently loaded programme to be received from the controller.
Type G	Command to tell the control unit to start a cycle.
Type I	Command to receive system output states (DONE, READY, PAUSE, ONLINE and ERROR outputs).
Type O	Command to tell the control unit to go into online status.
Type P	Command to communicate the loading of a programme with the same name as that contained in the package. In read mode, the command allows you to receive the name of the programme currently loaded on the control unit.
Type Q	Command allowing the text of a query on the control unit to be changed. The command contains the number of the query to be edited in the format nn (01,02,...,99).
Type S	Command to receive the error status of the control unit. There are a series of codes indicating the type of error that has occurred.
Type V	Command enabling the contents of a variable field to be modified. The command contains the number of the field to be modified in the format nn (01,02,...,99). In read mode, the command allows the contents of a variable field to be received.
Type H	Command to change the height of a text field or an arc text field. The command contains the number of the field to

	<p>change to format nn (01,02,...,99).</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>
Type C	<p>Command to change the position of a text field, arc text field and DXF. The command contains the number of the field to be changed in the format nn (01,02,...,99).</p> <p>The two different values to be set for the X and Y coordinates must be separated in this string by the character ';'. ;</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>
Type D	<p>Command enabling the diameter of an arc text field to be changed. The command contains the number of the field to be modified in the format nn (01,02,...,99).</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>
Type B	<p>Command enabling the start angle of an arc text field to be changed. The command contains the number of the field to be changed in the format nn (01,02,...,99). The value is represented in degrees.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>
Type M	<p>Command to read, for a specific marking field, the text that will be marked.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>

GSD file needed: HMSB1811.gsdCyclical data

Cyclic data is required in Profibus communication and therefore a register of a BYTE in Read on the driver side has been implemented. This means that to configure the system, the master must have configured an OUTPUT module of a BYTE. The consistency of the Slave-Master side configurations is crucial and it is strictly necessary that they correspond as described above.

In reality, the AC500 software bases all its commands on acyclic data exchange, and in fact the register established for cyclic data is not managed at the actual operating level.

Acyclic data

Below is the calculation required to map the ADIs (Application Data Instances, used to execute the remote control functions of the control unit) with the acyclic requests to be forwarded by the Profibus master:

$$\text{ADI} = \text{slot} - 255 + \text{index} + 1$$

$$\text{slot} = (\text{ADI} - 1) / 255$$

$$\text{index} = (\text{ADI} - 1) \text{ MOD } 255$$

Examples:

ADI	Slot	Index
1	0	0
255	0	254
65025	254	254

ADI	Object	Type	Description	Rights
0x01	Type I	BYTE	Requesting information on output status.	R
0x02	Type S	WORD	Requesting information on error status.	R
0x03	Type O	BYTE	Setting the system's online status. Entering the register with the value 1 will start the system's online procedure.	W
0x04	Type G	BYTE	Setting start marking cycle. Writing the register with the value 1 will start the marking cycle start procedure.	W
0x05	Type 1 Field number	WORD	Number of the field to be edited or read.	W
0x06	Type V Field number	WORD	Number of the field to be edited or read.	W
0x07	Type Q Field number	WORD	Number of the query to be modified.	W
0x10-0x42	Type 1 String	WORD	Value of the field to be modified or read. Each register will contain two ASCII characters. In writing, if the character 0 is found, the analysis of the value will be terminated. In reading instead a 0 character will be placed at the end of the string.	R/W
0x43	Type 1 Page	WORD	Defaulting to 0, this parameter serves	W

			to 'expand' the number of characters in the string of the previous command.	
0x45-0x77	Type V String	WORD	Value of the field to be modified. Each register will contain two ASCII characters. In writing, if the character 0 is found, the value analysis will be terminated. When reading, on the other hand, a character 0 will be placed at the end of the string.	R/W
0x78	Type V Pag	WORD	Defaulting to 0, this parameter serves to 'expand' the number of characters of the previous command string.	W
0x7A-0xAC	Type Q String	WORD	Query value to be modified. Each register will contain two ASCII characters. In writing, if character 0 is found, the value analysis will be terminated.	W
0xAD	Type Q Pag	WORD	Defaulting to 0, this parameter serves to 'expand' the number of characters in the string of the previous command.	W
0xAF-0xE1	Type P String	WORD	Value of the programme to be loaded. Each register will contain two ASCII characters. In writing, if the character 0 is found, the value analysis will be terminated. In reading instead a 0 character will be placed at the end of the string.	R/W
0xE2	Type P Page	WORD	Defaulting to 0, this parameter serves to 'expand' the number of characters of the previous command string.	W
0xE4	Type H Field number	WORD	Number of the field to be edited or read. This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.	W

INSTRUCTIONS FOR USE OF THE SOFTWARE

0xE5-0xEF	Type H String	WORD	<p>Value of the height of a field to be edited or read. This datum can only be used on a field of type text or arc text. Each register will contain two ASCII characters. In writing, if the character 0 is found, the value analysis will be terminated. In reading, on the other hand, a character 0 will be placed at the end of the string.</p> <p>This datum will contain the height value in text string format. This means that if the current height value is 4, the value of this datum will be the string '4'.</p> <p>This type of command is available</p>	R/W
------------------	---------------	------	---	-----

			only if advanced management of Fieldbus protocols has been unlocked from the AC500 Store.	
0xF1	Type C Field number	WORD	Number of the field to be edited or read. This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.	W
0xF2-0xFC	Type C String	WORD	Value of the position of a field to be edited or read. This data can only be used on a text field, arc text field and DXF. Each register will contain two ASCII characters. In writing, if character 0 is found, the value analysis will be terminated. In reading, on the other hand, a character 0 will be placed at the end of the string. This data will contain the position values in text string format. The position values in XY contained in the string will be separated by the character ';'. This means that if the current position value is x = 10 and y = 20, the value of this datum will be the string '10;20'. This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.	R/W
0xFE	Type D Field number	WORD	Number of the field to be edited or read. This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.	W
0xFF-0x109	Type D String	WORD	Value of the arc diameter of a field to be edited or read. This data can only be used on an arc text type field, with arc configured in degree mode. Each register will contain two ASCII characters. When writing, if a character 0 is found the value analysis will be terminated. In reading, on the other hand, a character 0 will be placed at the end of the string.	R/W

			<p>This datum will contain the diameter value in text string format. This means that if the current diameter value is 30, the value of this datum will be the string '30'.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>	
0x10B	Type B Field number	WORD	<p>Number of the field to be edited or read.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>	W
0x10C-0x116	Type B String	WORD	<p>Value of the starting angle of the arc of a field to be edited or read.</p> <p>This data can only be used on an arc text type field, with arc configured in degree mode. Each register will contain two ASCII characters. In writing, if the character 0 is found, the value analysis will be terminated. In reading, on the other hand, a character 0 will be placed at the end of the string.</p> <p>This datum will contain the value of the starting angle of the arc in text string format. This means that if the value of the current starting angle is 90, the value of this datum will be the string '90'. The value is represented in degrees.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>	R/W
0x118	Type M Field number	WORD	<p>Number of the field to be read.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>	W
0x119-0x14B	Type M String	WORD	<p>Value of the text field to be marked.</p> <p>Each register will contain two ASCII characters. A 0 character will be placed at the end of the string.</p>	R
0x14C	Type M Pag	WORD	<p>This parameter, set to 0 of default, allows you to extend the number</p>	W

			of the characters of the string read with the previous command.	
--	--	--	---	--

Example Reading Field

1. Writing the number of the field to be read into register **0x05**.
2. Reading registers **0x10-0x42**.

Register:	0x10	0x11	0x12
ASCII value (Hex):	54 45	53 54	00 00
String:	T E	S T	/0 /0

Example WRITING Field

1. Write field number to register **0x05**.
2. Writing registers **0x10-0x42**.

Register:	0x10	0x11	0x12
ASCII value (Hex):	54 45	53 54	31 00
String:	T E	S T	1 /0

String Management

For strings there are 50 registers per command, each of WORD type. Since a WORD contains two characters and there are 50 registers available for strings, there is a total of 100 characters when writing/reading a string. In order to be able to extend the number of characters in a string, a sort of page management was introduced, where the first 100 characters correspond to the first page, the next 100 to the second page, and so on.

On the practical side it will be:

- pag register set to 0 (default) = the 50 registers operate on the first 100 characters of the string;
- pag register set to 1 = the 50 registers operate from 100 to 200 characters of the string.

Profibus implicit model

Reading area

Instance	Object	Type	Description
1	Type I	UINT16	Information on the status of outputs.
2	Type S	UINT16	Information on error status.
3	Read Cmd Type	UINT16	Returns the current status of the command type being read.
4	Read Id	UINT16	Returns the index of the field or query being read. Used only for FIELD, VARIABLE and QUERY type commands.
5	Read Flags	UINT16	Returns the status information of the string being read. The first byte reports the status (0 = not ready, 1 = ready) while the second byte reports the length of the string being read.
6	<i>Unused</i>	Arrays of UINT8 [6].	6 bytes free for future use.
7	Read String	Array of UINT8 [64]	Returns the contents of the string being read. Combined with flags it is possible to read the number of bytes corresponding to the string being read.

Representation:

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

01-02: Report output status (Type I)
 03-04: Reports information status (Type S) 05-
 06: READ-CMD Current Type
 07-08: READ Current ID
 09-10: READ-String Flags.
 11-16: Unitised
 17-80: Read String data

Writing area

Instance	Object	Type	Description
1	Read Cmd Type	UINT16	Sets the type of command to be used for read operations.
2	Read Id	UINT16	Sets the index of the field or query to be used for read operations. Used only for FIELD, VARIABLE and QUERY type commands.
3	Type O	UINT8	Setting the system online. To execute the command, this BYTE must be set to a value of 1.
4	Type G	UINT8	Setting the start marking. To execute the command is it is necessary to set this BYTE to a value of 1.
5	Write Cmd Type	UINT16	Sets the type of command to be used for write operations.
6	Write Id	UINT16	Sets the index of the field or query to be used for write operations. Used only for FIELD, VARIABLE and QUERY type commands.
7	Write Flags	UINT16	Imposes the status information of the string being written. The first byte reports the status (0 = not ready, 1 = ready) while the second byte reports the length of the string being written.
8	<i>Unused</i>	Arrays of UINT8 [4].	4 bytes free for future use.
9	Write String	Array of UINT8 [64]	Sets the content of the string being written. Combined with flags it is possible to write the number of bytes corresponding to the string to be written.

Representation:

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

01-02: Set READ-CMD Type 03-
04: Set READ ID
05: Online setting
06: Set start of a cycle
07-08: Set WRITE-CMD Type 09-
10: Set WRITE ID
11-12: Set WRITE-String Flags. 13-
16: Unused
17-80: Write String data

STRING MANAGEMENT:

The operation of the string management logic via the I/O data of the Fieldbus protocols is described below. Using a few simple support bytes, it is possible to dynamically configure read and write operations on string data types. The bytes implemented for string management make it possible to define the type of string on which to operate (program name, program field, query, etc.), the id of the field that can be requested, and flags to indicate the status of the operation and the length of the string set.

- COMMAND Type

This data is used to identify which type of string data is to be used in the read/write operation.

- 0) NULL = No string
- 1) PATTERN = String referring to programmes
- 2) TEXT = String referring to the fields of a programme
- 3) VARIABLE = String referring to the variable fields of a programme
- 4) QUERY = String referring to queries
- 5) TEXT HEIGHT = String referring to the numeric value of the height of a text field (*)
- 6) TEXT POSITION = String referring to the numerical values of the XY positions of a text field (the two values must be separated by the character ';') (*)
- 7) ARC TEXT DIAMETER = String referring to the numerical value of the diameter of an arc in the arc text field (*)
- 8) ARC TEXT START ANGLE = String referring to the numerical value of the start angle (in degrees) of an arc in the arc text field (*)

(*) = These commands are only available after unlocking the Advanced Fieldbus Protocol Management functionality from the AC500 Store.

- ID

Optional data element, to be used only in the TEXT, VARIABLE and QUERY command types and is used to identify the number of the field on which the read/write operation is to be performed.

- String Flags

This data is used to identify the information relating to the read/write string. The first byte is used to inform whether the string is ready to be read/written or not, while the second byte is used to define the length of the string in the data area.

Example:

A number of bytes have been dedicated in the write area to define what the PLC wants to read from the AC500. By setting this data, the bytes in the read area will then be modified based on the configuration set in the write area.

For example, suppose you want to read the programme currently loaded on the AC500, you will need to write the bytes for the read string configuration in the following way:

WRITE AREA

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
00	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00

The AC500 will then modify the bytes in the reading area in this way:

READ AREA

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
00	00	00	00	00	01	00	00	00	04						

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
84 (T)	69 (E)	83 (S)	84 (T)												

Suppose, for example, that you want to read the third field of a programme:

WRITE AREA

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
00	02	00	03	00	00	00	00	00	00	00	00	00	00	00	00

The AC500 will then modify the bytes in the reading area in this way:

READ AREA

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
00	00	00	00	00	02	00	03	00	09						

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
84 (T)	69 (E)	83 (X)	84 (T)	70 (F)	73 (I)	69 (E)	76 (L)	68 (D)							

Conversely, if you want to go and load the **AC500** programme, you will have to configure the writing area in this way:

WRITE AREA

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
00	00	00	00	00	00	00	01	00	00	01	05	00	00	00	00

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
65 (A)	67 (C)	53 (5)	48 (0)	48 (0)	00	00	00	00	00	00	00	00	00	00	00

Attention: the AC500 will only perform the operation on the read string when the Write Flags instance status flag (eleventh byte of the representation) is set to **1**.

MANAGEMENT EDITING PROGRAMME FIELDS

The protocol managed by the AC500 can be used to change the format of fields within a marking programme. This can be done using the TYPE H, TYPE C, TYPE D and TYPE B commands.

All these commands work by string-formatting numeric values that must be used for each type of command associated with an editing functionality. The following are some simple examples that will give a better understanding of how these particular commands work.

Attention: commands for editing programme fields can only be used after unlocking the 'EXTENDED PROTOCOL / ADVANCED FIELDBUS' functionality.

Reading the height of a text field (TEXT HEIGHT)

- set the number of the field you wish to modify (refer to the object dictionary table above);
- read the data referring to the TYPE H command type;
- this data is a character string containing within it the numeric value of the height of the required text field, such as:
"4" means that the text height is 4mm (proprietary text) or 4 points (TTF text).

Writing the height of a text field (TEXT HEIGHT)

- set the number of the field you wish to modify (refer to the object dictionary table above);
- write the data referring to the TYPE H command type;
- this data is a character string containing within it the numeric value of the height of the required text field, such as:
"6" means that the text height is 6mm (proprietary text) or 6 points (TTF text).

Reading the position of a text field (TEXT POSITION)

- set the number of the field you wish to modify (refer to the object dictionary table above);
- read the data referring to the TYPE C command type;

- this data is a character string containing within it the numeric values of the position in XY separated by a ';' character, such as:
"20;15.5" means that the position is X=20mm and Y=15.5mm.

Writing the position of a text field (TEXT POSITION)

- set the number of the field you wish to modify (refer to the object dictionary table above);
- write the data referring to the TYPE C command type;
- this data is a character string containing within it the numeric values of the position in XY separated by a ';' character, such as:
"40;10" means that the position is X=40mm and Y=10mm.

Reading the Diameter of an Arc Text Field (ARC TEXT DIAMETER)

- set the number of the field you wish to modify (refer to the object dictionary table above);
- read the data referring to the TYPE D command type;
- this data is a character string containing within it the numerical value of the diameter of the required arc text field, e.g:
"30.6" means that the diameter of the arch text is 30.6mm.

Writing the diameter of an arc text field (ARC TEXT DIAMETER)

- set the number of the field you wish to modify (refer to the object dictionary table above);
- write the data referring to the TYPE D command type;
- this data is a character string containing within it the numerical value of the diameter of the required arc text field, e.g:
"40" means that the diameter of the arch text is 40mm.

Reading the starting angle of an arc text field (ARC TEXT START ANGLE)

- set the number of the field you wish to modify (refer to the object dictionary table above);
- read the data referring to the TYPE B command type;
- this data is a character string containing within it the numerical value of the starting angle of the required arc text field, e.g:
"180.7" means that the angle of the arc text is 180.7°.

Writing the starting angle of an arc text field (ARC TEXT START ANGLE)

- set the number of the field you wish to modify (refer to the object dictionary table above);
- write the data referring to the TYPE B command type;
- this data is a character string containing within it the numerical value of the starting angle of the required arc text field, e.g:
"90" means that the angle of the arch text is 90°.

Appendix I

Profinet IO Fieldbus Module

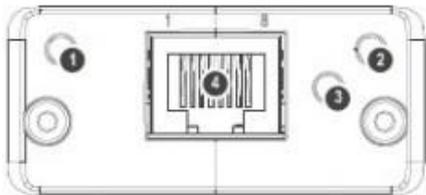
PROFINET IO

PROFINET is a standard for industrial automation using a [computer network](#). PROFINET uses standards such as [TCP/IP](#) and [Ethernet](#). PROFINET's modular structure allows users to select only needed functions for different requirements....

quotation from en.wikipedia.org

The Profinet IO port on AC500

The Profinet IO port on an AC500 unit.

	Item	
1	Network Status LED	
2	Module Status LED	
3	Links/Activity	
4	Ethernet Interface	

Network status LED.

Note: An LED lighting sequence is done when the module is switched on.

LED State	Description
Off	No power or no connection
Green	On-line, IO controller in RUN state
Green, flashing	On-line, IO controller in STOP state

Module status LED.

Note: An LED lighting sequence is done when the module is switched on.

LED State	Description
Off	Not initialised
Green	Normal Operation
Green, 1 flash	Diagnostic Event
Green, 2 flashes	Blink
Red	Exception Error
Red, 1 flash	Configuration Error
Red, 2 flashes	IP Address Error

Red, 3 flashes	Station Name Error
Red, 4 flashes	Internal Error

Link/activity LEDs

LED State	Description
Off	Link, no activity
Green	Established links
Green, flickering	Activity

Profinet IO configuration

Before you start using the Profinet IO communication module on the control unit, you must ensure that you have performed all the necessary steps to configure it correctly.

First, it will be necessary to access the auxiliary functionality environment:



Access the system configuration environment from the auxiliary functionality menu.

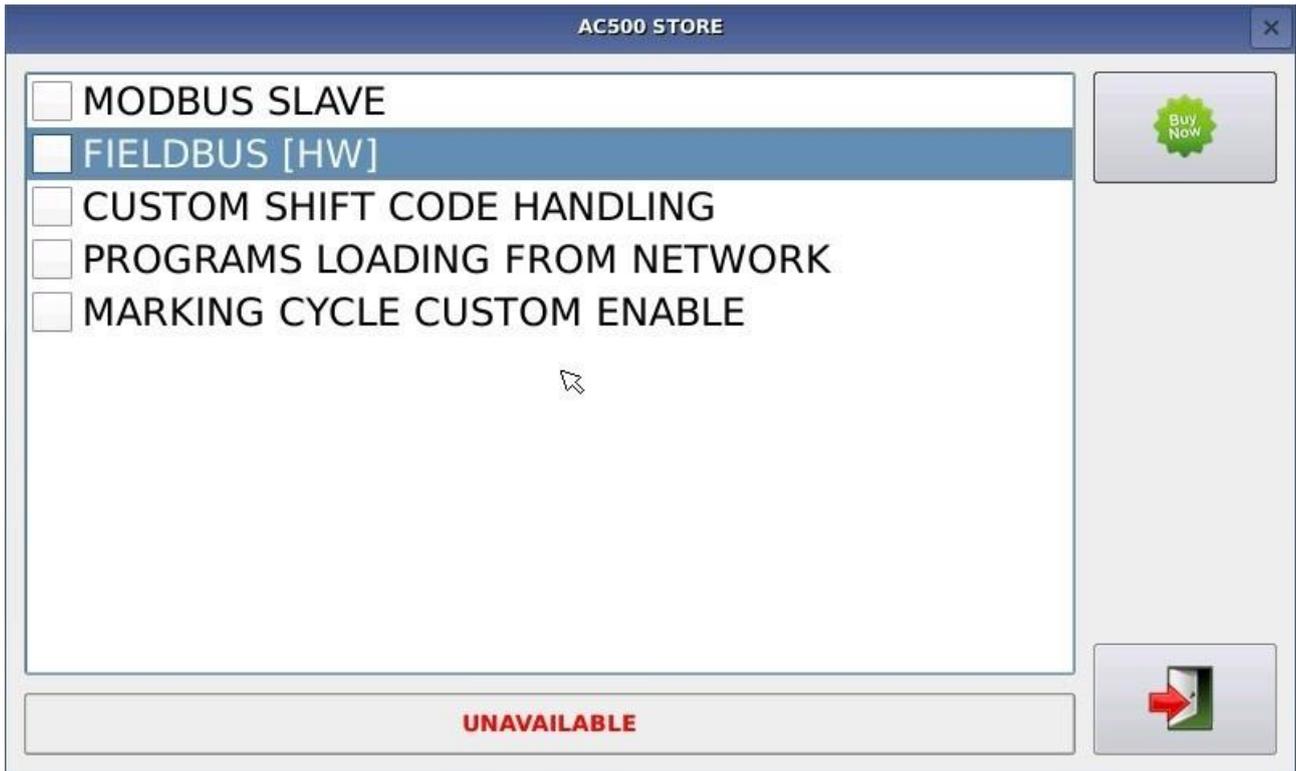
INSTRUCTIONS FOR USE OF THE SOFTWARE



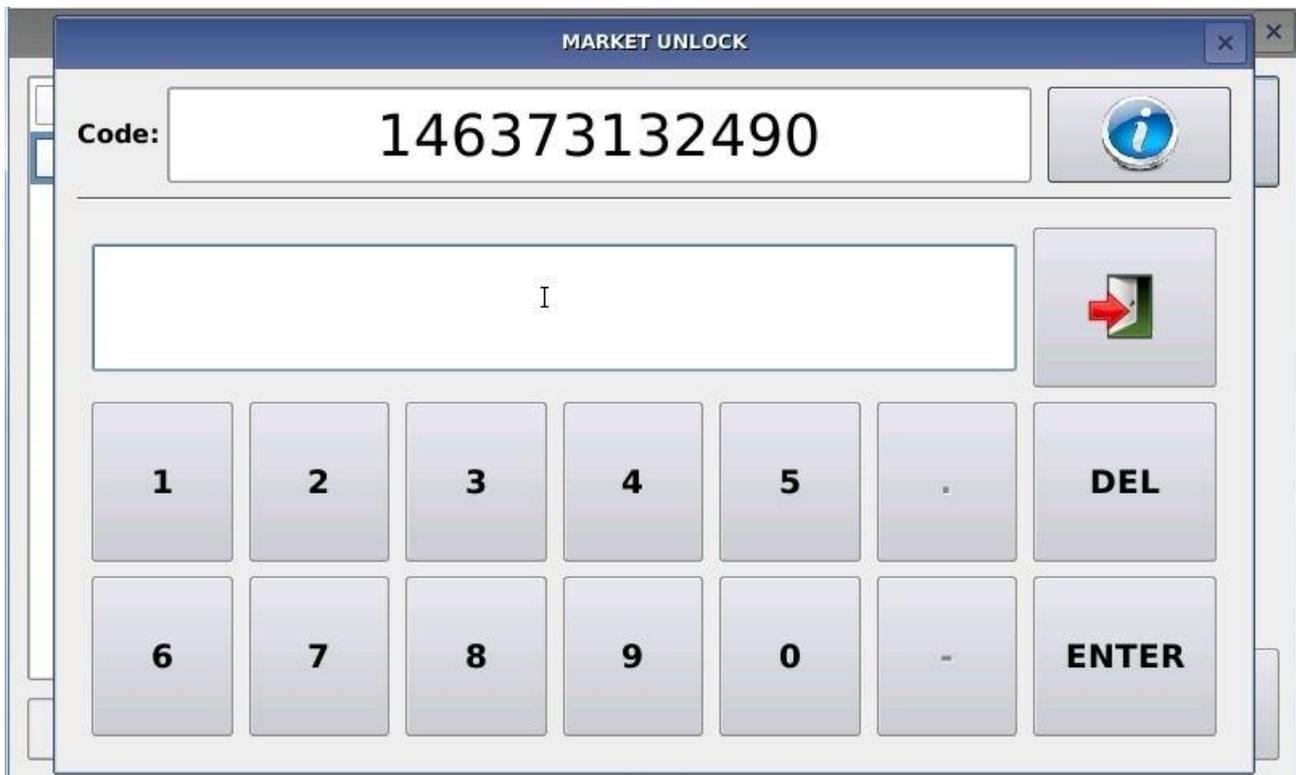
Once you have entered the system configuration environment, press the **AC500 STORE** button to access the environment that will allow you to purchase the Fieldbus protocol management functionality:



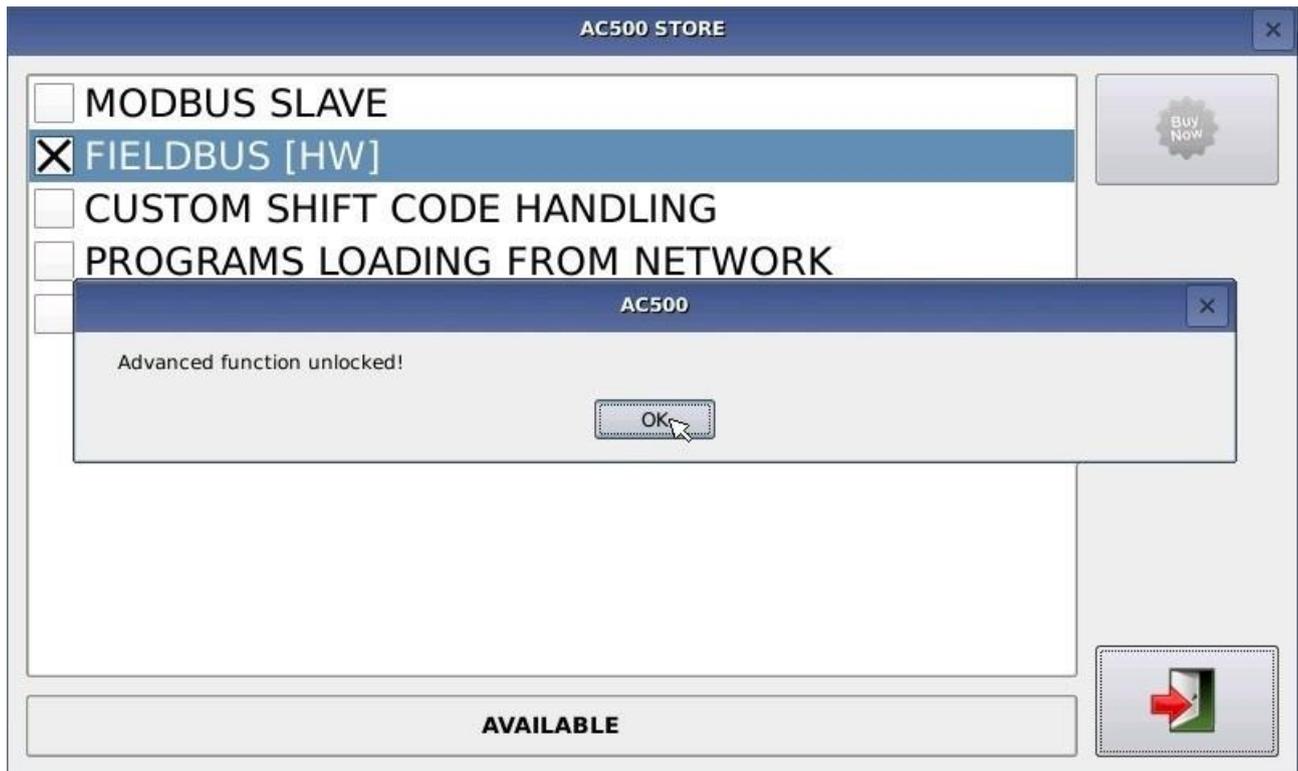
Select the Filedbus entry to unlock the management of Fieldbus protocols, and press the button on the right:



On the screen that will be displayed, you will be asked to send the given code to the controller supplier. Once this is done, you will be notified of the code to be entered to proceed with the purchase of the function.

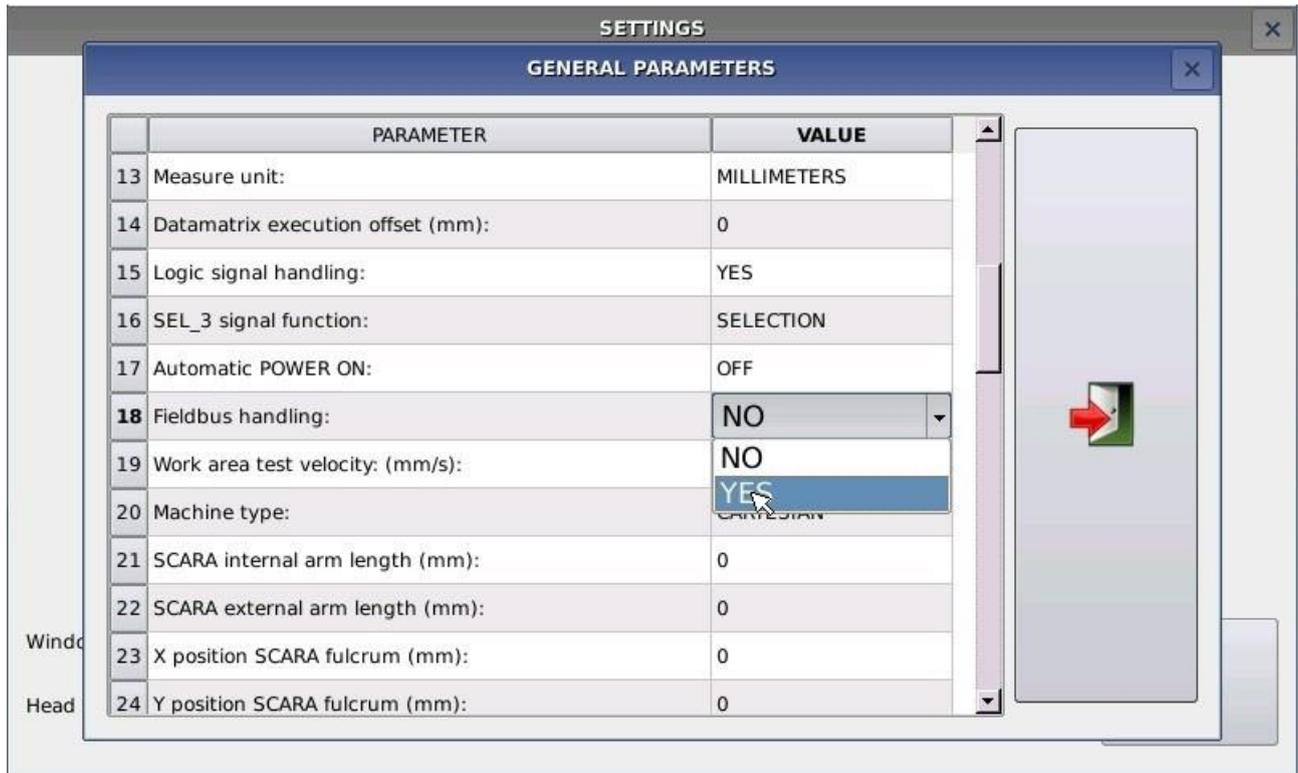


Once the unlock code has been entered, the **ENTER** button must be pressed. If the code is correct, confirmation that the operation was successful will be displayed.



Now it will be possible to go to the general data configuration environment, contained in the system settings environment, and enable the **Fieldbus Management** parameter.

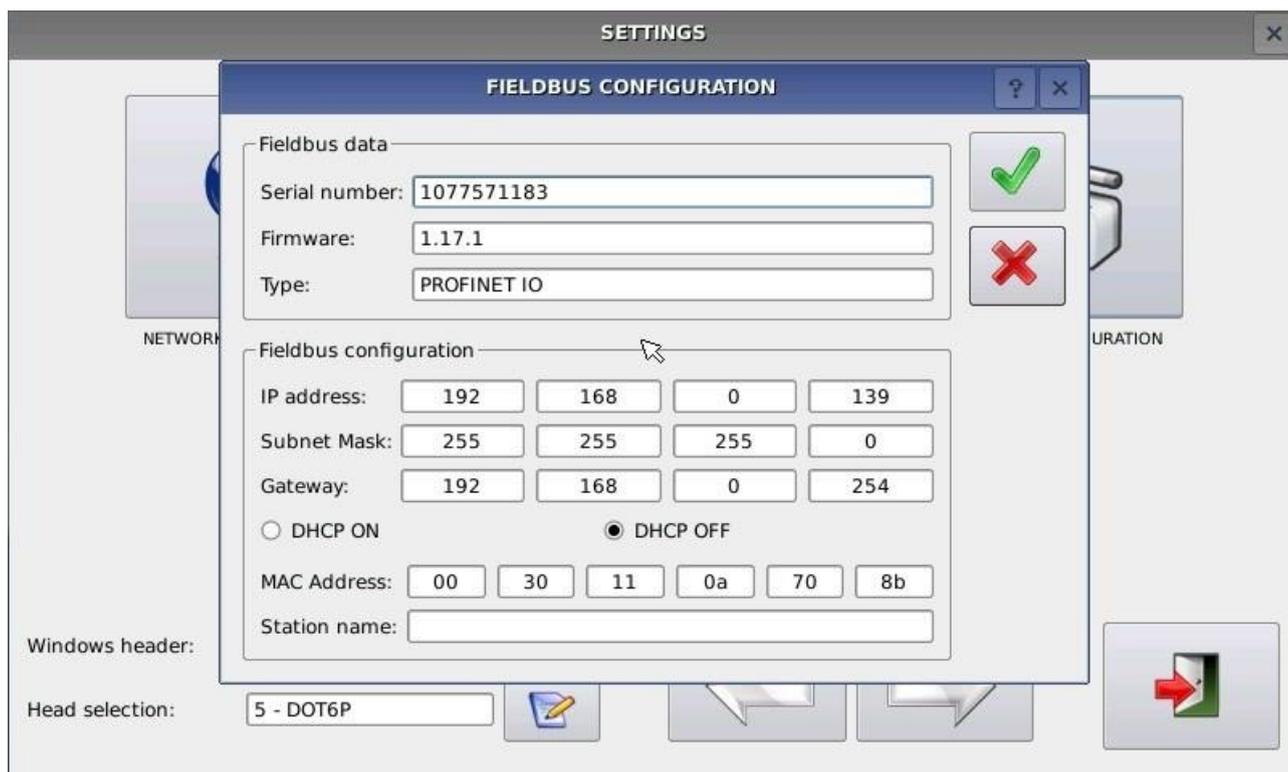




Once Fieldbus protocol management is enabled, it will be possible to access the configuration environment:



In this environment, it will be possible to check whether the module has been correctly connected and enter the necessary data to configure it correctly:



While proceeding with the input of the module configuration data, it is necessary to keep the following things in mind:

- If the *serial number* and *firmware* version are set to 0 (zero) and the *module is unknown*, this means that the AC500 Series driver has failed to recognise the fieldbus module; in this case, the control unit must be restarted and the module connections must be checked to ensure that they have been made correctly.
- An IP address already set on the Ethernet port of the control unit must not be set.

After completing the module data configurations, you will be asked to restart the control unit to complete the initialisation procedure.

EXTENDED PROTOCOL - DESCRIPTION OF COMMANDS

Command	Description
Type 1	<p>Command to overwrite the contents of an individual field with the data contained in the transmitted packet. The command contains the number of the field to be changed in the format nn (01,02,...,99).</p> <p>In read mode, the command allows the contents of a specific field of the currently loaded programme to be received from the controller.</p>
Type G	Command to tell the control unit to start a cycle.
Type I	Command to receive system output states (DONE, READY, PAUSE, ONLINE and ERROR outputs).
Type O	Command to tell the control unit to go into online status.
Type P	<p>Command to communicate the loading of a programme with the same name as that contained in the package.</p> <p>In read mode, the command allows you to receive the name of the programme currently loaded on the control unit.</p>
Type Q	Command allowing the text of a query on the control unit to be changed. The command contains the number of the query to be edited in the format nn (01,02,...,99).
Type S	Command to receive the error status of the control unit. There are a series of codes indicating the type of error that has occurred.
Type V	<p>Command enabling the contents of a variable field to be modified. The command contains the number of the field to be modified in the format nn (01,02,...,99).</p> <p>In read mode, the command allows the contents of a variable field to be received.</p>
Type H	<p>Command enabling the height of a text field or an arc text field to be changed. The command contains the number of the field to be changed in the format nn (01,02,...,99).</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>
Type C	Command to change the position of a text field, arc text field and DXF. The command contains the number of the field to be changed in the format nn (01,02,...,99).

	<p>The two different values to be set for the X and Y coordinates must be separated in this string by the character ','.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>
Type D	<p>Command enabling the diameter of an arc text field to be changed. The command contains the number of the field to be modified in the format nn (01,02,...,99).</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>
Type B	<p>Command allowing the start angle of an arc text field to be changed. The command contains the number of the field to be changed in the format nn (01,02,...,99). The value is represented in degrees.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>
Type M	<p>Command to read, for a specific marking field, the text that will be marked.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>

PROFINET IO ADIs Object (A2h)

This object is used to manage the data to be passed to the AC500 application. The CIP Object ADIs (Application Data Instances, mapped with address 0xA2) is used as a medium for exchanging data with the AC500 application. Each instance corresponds to a different extended protocol command. Each instance consists of several properties, called *attributes*.

#	Name	Access	Type	Description
1	Name	Get	Array of CHAR	Parameter name
2	Data type	Get	UINT8	Data type of instance value
3	No. of elements	Get	UINT8	Number of elements of the specified data type
4	Description	Get	UINT8	Bit field describing the access rights for this instance <u>Bit:Meaning:</u> 0 Set = Get Access 1 Set = Set Access
5	Value	Get/Set	Determined by attribute #2	Instance value.
6	Max value	Get		The maximum permitted parameter value (NOT USED)
7	Min value	Get		The minimum permitted parameter value. (NOT USED)
8	Default value	Get		The default parameter value. (NOT USED)

To set or read each command value associated with each instance (see object dictionary table), reference is made to the *value* attribute, i.e. attribute number 5.

PROFINET IO Get/Set Attribute Single

There are two data access commands, both read and write, used with the Profinet model of the AC500:

- **Get Attribute Single (0x0E):** command for reading an instance attribute;
- **Set Attribute Single (0x10):** command for writing an instance attribute.

The EtherNet/IP model set up on the AC500 only handles these two commands for resource access operations, which means that all other commands are currently not handled and are not recommended.

OBJECT DICTIONARY - PROFINET IO

CIP Object	Instance	Attribute	Object	Type	Description	Rights
0xA2	1	5	Type I	UINT16	Requesting information on output status.	R
0xA2	2	5	Type S	UINT16	Requesting information on error status. The upper part of the WORD will be placed in the first register, and the lower part in the second.	R
0xA2	3	5	Type O	UINT8	Setting the system's online status. Setting the least significant bit to 1 will send the system online. ALL other values passed will be ignored. <i>Example: 01h (online system)</i>	W
0xA2	4	5	Type G	UINT8	Marking cycle start setting. Setting the least significant bit to 1 will launch the start of the cycle. ALL other values passed will be ignored. <i>Example: 01h (online system)</i>	W
0xA2	5	5	Type 1	Array of CHAR	Instance dedicated to the Type 1 command. Through this instance, it will be possible to perform read and write operations on the programme fields. The maximum size of the string passed is 255 characters.	R/W
0xA2	6	5	Type V	Array of CHAR	Instance dedicated to the Type V command. Through this instance, it will be possible to perform read and write operations on the variable fields of the programme.	R/W

					The maximum size of the string passed is 255 characters.	
0xA2	7	5	Type P	Array of CHAR	<p>Instance dedicated to the Type P command.</p> <p>Through this instance it will be possible to perform read and load programme operations. By reading it will be possible to request the name of the programme loaded on the control unit, and by writing a value it will be possible to load that programme.</p> <p>The maximum size of the string passed is 255 characters.</p>	R/W
0xA2	8	5	Type Q	Array of CHAR	<p>Instance dedicated to the Type Q command.</p> <p>Through this instance, it will be possible to perform read and write operations on a control unit query.</p> <p>The maximum size of the string passed is 255 characters.</p>	R/W
0xA2	9	5	Type 1 FIELD	UINT8	<p>Instance dedicated field number to be taken into account almost read/write operations on instance 5.</p> <p>The value set here will be used to specify the number of the field that will be read/modified with instance 5.</p>	W
0xA2	10	5	Type V FIELD	UINT8	<p>Instance dedicated field number to be taken into account almost read/write operations on instance 6.</p> <p>The value set here will be used to specify the number of the field that will be read/modified with instance 6.</p>	W
0xA2	11	5	Type Q FIELD	UINT8	Instance dedicated to the field number to be taken into account almost	W

					<p>writing on instance 8.</p> <p>The value set here will be used to specify the number of the field which will be modified by instance 8.</p>	
0xA2	12	5	Type H	Array of CHAR	<p>Instance dedicated to the Type H command type.</p> <p>This instance allows read and write operations to be performed on the height value of a field of type text or arc text.</p> <p>This datum will contain the height value in text string format. This means that if the current height value is 4, the value of this datum will be the string '4'.</p> <p>The maximum size of the string passed is 255 characters.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>	R/W
0xA2	13	5	Type H FIELD	UINT8	<p>Instance dedicated to the field number to be taken into account when performing read/write operations on instance 12.</p> <p>The value set here will be used to specify the number of the field to be read/modified with instance 12.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>	W
0xA2	14	5	Type C	Array of CHAR	<p>Instance dedicated to the Type C command type.</p> <p>This instance allows read and</p>	R/W

					<p>writing on the position of a text, arc text and DXF type field.</p> <p>This data will contain the position values in text string format. The position values in XY contained in the string will be separated by the character ';'. This means that if the current position value is x = 10 and y = 20, the value of this datum will be the string '10;20'.</p> <p>The maximum size of the string passed is 255 characters.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>	
0xA2	15	5	Type C FIELD	UINT8	<p>Instance dedicated to the field number to be taken into account when performing read/write operations on instance 14.</p> <p>The value set here will be used to specify the number of the field to be read/modified with instance 14.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>	W
0xA2	16	5	Type D	Array of CHAR	<p>Instance dedicated to the Type D command type.</p> <p>This instance allows read and write operations to be performed on the arc diameter of an arc text type field. The arc of the field must necessarily have been created by degree mode.</p> <p>This data will contain the value</p>	R/W

					<p>of the diameter in text string format. This means that if the current diameter value is 30, the value of this data will be the string '30'.</p> <p>The maximum size of the string passed is 255 characters.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>	
0xA2	17	5	Type D FIELD	UINT8	<p>Instance dedicated to the field number to be taken into account when performing read/write operations on instance 16.</p> <p>The value set here will be used to specify the number of the field to be read/modified with instance 16.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>	W
0xA2	18	5	Type B	Array of CHAR	<p>Instance dedicated to the Type B command type.</p> <p>This instance allows read and write operations to be performed on the starting angle of the arc of an arc text type field. The arc of the field must necessarily have been created by degree mode.</p> <p>This datum will contain the value of the starting angle of the arc in text string format. This means that if the value of the current starting angle is 90, the value of this datum will be the string '90'. The value is represented in degrees.</p>	R/W

					<p>The maximum size of the string passed is 255 characters.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>	
0xA2	19	5	Type B FIELD	UINT8	<p>Instance dedicated to the field number to be taken into account when performing read/write operations on instance 18.</p> <p>The value set here will be used to specify the number of the field that will be read/modified with instance 18.</p> <p>This type of command is only available if the advanced management of Fieldbus protocols has been unlocked from the AC500 Store.</p>	W
0xA2	20	5	Type M	Array of CHAR	<p>Instance dedicated to Type M command type.</p> <p>This instance reads the string that will be marked during the marking cycle.</p> <p>The maximum size of the string is 255 characters.</p>	R
0xA2	21	5	Type M FIELD	UINT8	<p>Instance dedicated to the Type M FIELD command type.</p> <p>This instance allows the field number associated with instance 20 to be specified.</p>	W

PROFINET IO connected to AC500

Once the control unit is correctly configured, the object dictionary above can be used to perform all queries that the AC500 software proposes. It is possible to connect to the Profinet module on the AC500 control unit via a standard Ethernet cable.

Examples of questionsREADING example

3. Writing the number of the field to be read into the **Type 1 FIELD** instance.
4. Reading of **Type 1** instance.

Array Index:	0	1	2	3
ASCII value (Hex):	54	45	53	54
String:	T	E	S	T

Example WRITING Field

3. Writing the number of the field to be written in the **Type 1 FIELD** instance.
4. Writing of **Type 1** instance with the following values.

Array Index:	0	1	2	3	4
ASCII value (Hex):	54	45	53	54	31
String:	T	E	S	T	1

Profinet implicit model

The Profinet protocol has a number of important constraints that must absolutely be respected in comparison to other Profibus and EtherNet/IP modules. These constraints have led to the need to minimise the data exchanged in implicit mode

Reading area

Instance	Object	Type	Description
1	Type I	UINT16	Output status information
2	Type S	UINT16	Error status information. This log contains system errors up to PIX_OUT_OF_RANGE_ERROR, see the AC500 manual for more details.
3	Read Cmd Type	UINT16	Returns the current status of the command type being read.
4	Read Id	UINT16	Returns the index of the field or query being read. Used only for FIELD, VARIABLE and QUERY type commands.
5	Read Flag	UINT16	Returns the status information of the string being read. The first byte reports the status (0 = not ready, 1 = ready) while the second byte reports the length of the string being read.
6	Read Length	UINT16	It indicates the length of the string in reading.
7	<i>Unused</i>	Array of UINT16 [5]	5 free words for future use.
8	Read String	Array of UINT16 [20]	Returns the contents of the read string. Combined with the flags the number of bytes corresponding to the string to be read.

Representation:

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

INSTRUCTIONS FOR USE OF THE SOFTWARE

AUTOMATOR MARKING SYSTEMS SRL - Via Meucci n.8-20094 Corsico (MI) Italy
 Tel 02 48601445 Fax 02 48601503 e-mail info@automator.com www.automator.com

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L

- 01: Set READ-CMD Type
- 02: Set READ ID
- 03: Set online
- 04: Set start of a cycle
- 05: Set WRITE-CMD Type
- 06: Set WRITE ID
- 07: Set WRITE-String Flag
- 08: Set WRITE-String
- Length 09-11: Unused
- 12-31: Write String data

STRING MANAGEMENT:

Strings must be entered taking into account the top-bottom order of the word, one character per byte.

Example:

12		13	
84	69	83	84
(T)	(E)	(S)	(T)