USER MANUAL

COMMAND INSTRUCTIONS FOR UC500 - XCOM

GRAVOTECH

EXPRESSION OF THINGS

Table of contents

A. Foreword

1. Appreciation

Gravotech is pleased to count you among the users of its engraving and traceability solutions.

For help, contact Gravotech.

For more information on products, visit www.gravotech.com website.

2. Information

	To ensure security and productivity, read this manual before starting-up the equipment. It provides details about the installation and use of the equipment. Keep this manual in case you need to refer to it.

	For the attention of users having an individual cardiac assist device fitted: Our equipment is designed and manufactured with the greatest care in order to guarantee their compliance with the EMC Directive currently in force. This means that the levels of electromagnetic emissions produced by this equipment when in operation are limited and do not exceed the thresholds defined by the Directive. However, multiple factors make it impossible to guarantee the total absence of risk for users having a cardiac assist device fitted. Consequently, it is recommended that standing for a prolonged period within less than 1 m (3.281 ft) of an operating machine should be avoided.

B. Legal notices

Last updated: 09/2020

The purpose of this document is to provide users (hereinafter the User(s)) with information and to ensure their safety. It has no contractual value and Gravotech group (hereinafter Gravotech) reserves the right, at any time and without notice, to make such changes or improvements as it deems fits, or to substitute any new equipment and/or material and/or part and/or image to its equipment, software and/or associated manuals or documentation (hereinafter the Product(s)).

This manual, including texts, images, photos, graphics, design, or any compilation, digital conversion or data contained in it, is subject to copyright. This manual shall not be reproduced, disseminated, transmitted, transcribed, translated or stored electronically, on any medium whatsoever regardless of its format without the express and written permission of Gravotech, to the exception of software backup copies as provided by law.

The intellectual property rights relating to the Products and to this manual, including - but not limited to - patents, trademarks, models, copyright, domain names and also the know-how, trading name or company name, are owned by Gravotech Marking S.A.S or any company of the Gravotech group. Under no circumstances does the transmission of this manual or the supply of Products or services constitute an assignment of or any express or tacit license for any intellectual property right owned by Gravotech.

To the extent permitted by law, Gravotech provides hereby no warranty (in particular no warranties of performance, non-infringement, merchantability or fitness for a particular purpose) relating to the supply of its Products, other than those conferred upon the User by Gravotech's general terms and conditions of sale or any contractual document agreed between Gravotech and the User. Nor does Gravotech guarantee the compatibility of its software with any software package not supplied by it, or any defect in assembly, adaptation, design, compatibility and operation with any or part of a combination created by the User.

Gravotech shall not be liable for any damages, that the User or its property, a third party or the Product itself may suffer, caused by the Product and arising from any inappropriate use or misuse of the Product, negligence, carelessness, inadequate supervision or maintenance, failure to observe the safety or usage instructions described herein or otherwise communicated to the User, the use of poor-quality or non-recommended lubricants, fluids and additives or where there is fault on the part of the User or a third party. As provided in this manual, the User shall furthermore (i) observe the normal conditions of use, (ii) not exceed the recommended maximum number of hours during which the equipment may be operated on and (iii) refrain from proceeding to any Product's repair or make it proceed by any unqualified third party, or without the appropriate personal protective equipment.

The Product's specifications are altered by (i) any Product's modification or alteration, (ii) any adaptation and installation of accessories that are not recommended by Gravotech, (iii) the integration of a control system and (iv) the connection to an external device. Such specifications' alterations may lead to the non-compliance of the Product with applicable rules and standards. Shall the Product be non-compliant, the person in charge of the Product's installation shall be responsible of the final workstation's compliance. In no event, Gravotech shall be liable for any damages arising from such non-recommended or unauthorized Product's alterations. It is precised that the warranty shall not apply in such case.

Under no circumstances shall Gravotech be held liable for any indirect, incidental, special, consequential punitive or other similar damages, including any economic loss, loss of profit, loss of data or opportunity, whether or not foreseeable by or communicated to Gravotech, caused by this manual or the supply of Products or services concerned by the said manual.

To the widest extent permitted by law, Gravotech shall only be held liable for direct damage arising from personal injury caused by a fault proven in its Product (including this manual).

Gravotech® and the trademarks of products marketed by Gravotech group are used, pending or registered trademarks of Gravotech Marking or one of Gravotech group subsidiaries.

The products and names of third party companies which appear in this manual are used solely for the necessary purposes of reference, and in particular for issues of compatibility. All the trademarks mentioned in this manual remain the property of their respective owners. Windows® is (are) a used, pending or registered trademark(s) of Microsoft Corporation. Postscript® is (are) a used, pending or registered trademark(s) of Adobe Systems Incorporated.

C. Introduction

A Gravotech marking machine is made up of an electro-mechanical unit, which carries out the marking and of an electronic unit, which commands the marking.

The electro-mechanical unit exists in different presentations depending on the industrial environment.

This machine works by moving the stylus along the X and Y axes of a grid.

The electronic unit, or CCU, is equipped with internal memory containing the movement management program, character fonts as well as logos. The CCU is equipped with a 8I / 4O board.

The UC500 control unit works with the T05 firmware. The XCOM control unit works with the T08 firmware.

The two interconnected units receive marking data from any computer system capable of sending the data on a RS232 serial port or on an Ethernet port via a telnet session (TCP/IP).

1. Definition of a command instruction

A command instruction is used to carry out a remote function by the machine via the RS232 - TCP/IP connection.

Command instructions are text commands for which the name consists of 2 characters. This is not case sensitive.

Character strings must be between quotation marks (code 34). This is case sensitive.

Each command may be followed by several parameters, separated by spaces. Each type of parameter depends on the instruction.

Each command has a maximum size of 300 000 characters.

All command instructions involve an answer. Always wait for the response(s) before sending another command.

Errors are signaled via codes, more or less explicit depending on the mode configured.

Introduction

2. Protocol

This part lists and describes the different communication protocols supported by the T05 / T08 program. According to the physical support selected (RS232, Ethernet or USB), not all protocols are possible.

■ **Possible protocol(s) on RS232**

Use only recommended with earlier-generation machines.

Parameter the RS232 connection (baud rate, parity, number of data bits, etc) in the graphic interface of the T05 / T08 program.

Select the "Command instructions" mode (with or without checksum) in the T05 / T08 program (Communication menu). This mode is used to pilot the machine via an evolved RS232 peripheral (PLC or PC).

This step is critical for enabling the dialog with the connected RS232 peripheral.

■ **Operating principle**

Data sent to the machine must be organised by character strings. The machine confirms reception of all character strings. It never takes the initiative of sending, it only responds to requests.

■ **Formatting character strings**
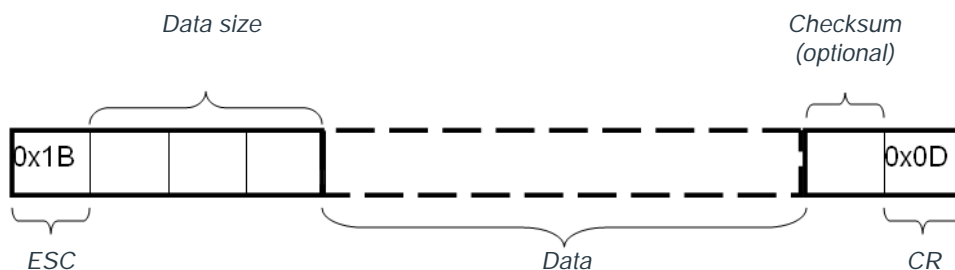
Each character string must start by the starting character <ESC> (27 in decimal = 1B in hexadecimal).

Each character string must end by the end character <CR> (13 in decimal = 0D in hexadecimal).

Introduction

- Data size

It corresponds to the size of the data sent in the character string.

Its maximum value is 300 000 - 6 = 299 994.

It is encoded over 3 bytes, with the least significant bit on the right.

- Checksum

The checksum is calculated with the operator "XOR" on the size of the data and on every data.

XOR = exclusive bit to bit "or"

Example:

	1	0	1	0
XOR	1	1	0	0
=	0	1	1	0

The checksum is not activated by default. If needed, activate it in the T05 / T08 program. Refer to the user manual for the T05 / T08 program.

- Machine's answer

A correctly formatted character string, whose size of the data and checksum are correct, is acknowledged by a <ACK> byte (6 in decimal = 6 in hexadecimal).

In the opposite case, the program responds <NAK> (21 in decimal = 15 in hexadecimal). The program is not aware of the character string, which must be sent back.

All command instructions involve an answer. It is given in the form of a character string with the same formatting containing corresponding data.

Introduction

Example: sending a LS command and reception of the answer

Character string sent (with checksum)

Data size: 2 Bytes

Characters	<ESC>				L	S		<CR>
Decimal	27	0	0	2	76	83	29	13
Hexadecimal	1B	0	0	2	4C	53	1D	0D

Acknowledgement

Characters:	<ACK>
Decimal	6
Hexadecimal	6

■ Possible protocol(s) on Ethernet

Activate the command instructions in the interface of the T05 / T08 program (Communication menu).

The initial protocol is TCP/IP. The machine must be correctly parameterized and connected to a functional network.

Telnet session

Connection to the machine is made via a telnet session, on the 55555 port (value by default, modifiable on the machine). The machine is like a telnet server. The connection is made without providing an ID or password. Only one connection at a time is possible.

Connection may be done with a standard telnet client, for a manual sending of the commands, or directly via a user program which opens a connection (socket) on the port. In this case, line breaks and the prompt, useful when displaying in a Telnet client, are no longer necessary. Deactivate sending in the "Configuration" menu. Refer to the user manual for the T05 / T08 program.

TCP/IP ensures reliable data transfer. Commands may therefore be entered directly, without protocol overlayering in order to validate their integrity or to acknowledge their reception.

The telnet session and the command instructions are in text mode.

Introduction

Example

Connection via a telnet client on order line:

- IP address: 192.168.0.211

- port: 55555

The first line is manually typed (telnet command).
The following lines are information emitted by the telnet client.
The last line is sent by the machine (prompt).

```
$ telnet 192.168.0.211 55555
Trying 192.168.0.211...
Connected to 192.168.0.211 (192.168.0.211).
Escape character is '^]'.

 >
```

It is thus possible to enter commands, sent after pressing the Enter key.

```
$ telnet 192.168.0.211 55555
Trying 192.168.0.211...
Connected to 192.168.0.211 (192.168.0.211).
Escape character is '^]'.

 >LS
13
0.lo3
1.lo3
2.lo3
3.po3
0.tml
00.tml
001.tml
f3.tml
test.tml
 >
```

The LS command is manually entered and sent to the machine via the Enter key.

The following lines are emitted by the machine and received by the telnet client.

Introduction

3. Alphabetical list of commands

		Description	See page(s)
A	**AD**	Fault acknowledgement	14
	AM	Stop marking	15
G	**GF**	File reception	16
	GI	Information request	17
	GO	Start marking	18
H	**HP**	On-line help	19
L	**LD**	Loading the marking files	20
	LS	List of files stored on the machine	21
M	**MV**	Stylus movement	22
O	**OG**	Request of return to origin	23
P	**PF**	Sending the file (file compatible with the machine)	24
Q	**QS**	End of a telnet session - without exiting controlled mode	26
	QT	End of a telnet session - exit controlled mode	27
R	**RM**	Deleting files	28
S	**SI**	Definition of information	29
	ST	State of machine operation	30
V	**VG**	Interrogation of a variable's value	31
	VM	Configuration of the response mode (human/machine)	32
	VS	Assigning a variable	33

Introduction

4. Possibilities for using commands

The table shows the commands which can be used according to the status of the machine.

	ALIVE	READY	MARKING	FAULT	READY/ PAUSE
AD				X	
AM		X	X		X
GF	X	X		X	X
GI	X	X	X	X	X
GO		X			X
HP	X	X		X	X
LD	X				
LS	X			X	
MV	X				
OG	X	X			
PF	X			X	
QS	X	X	X	X	X
QT	X				
RM	X			X	
SI	X			X	
ST	X	X	X	X	X
VG	X	X		X	X
VM	X	X		X	X
VS	X	X		X	X

Note:

X	Useable command: XCOM + UC500
X	Useable command: only for XCOM machines
X	Useable command: only for UC500 machines
	Non-useable command

GRAVOTECH

Introduction

■ Command instructions: impact on the machine status
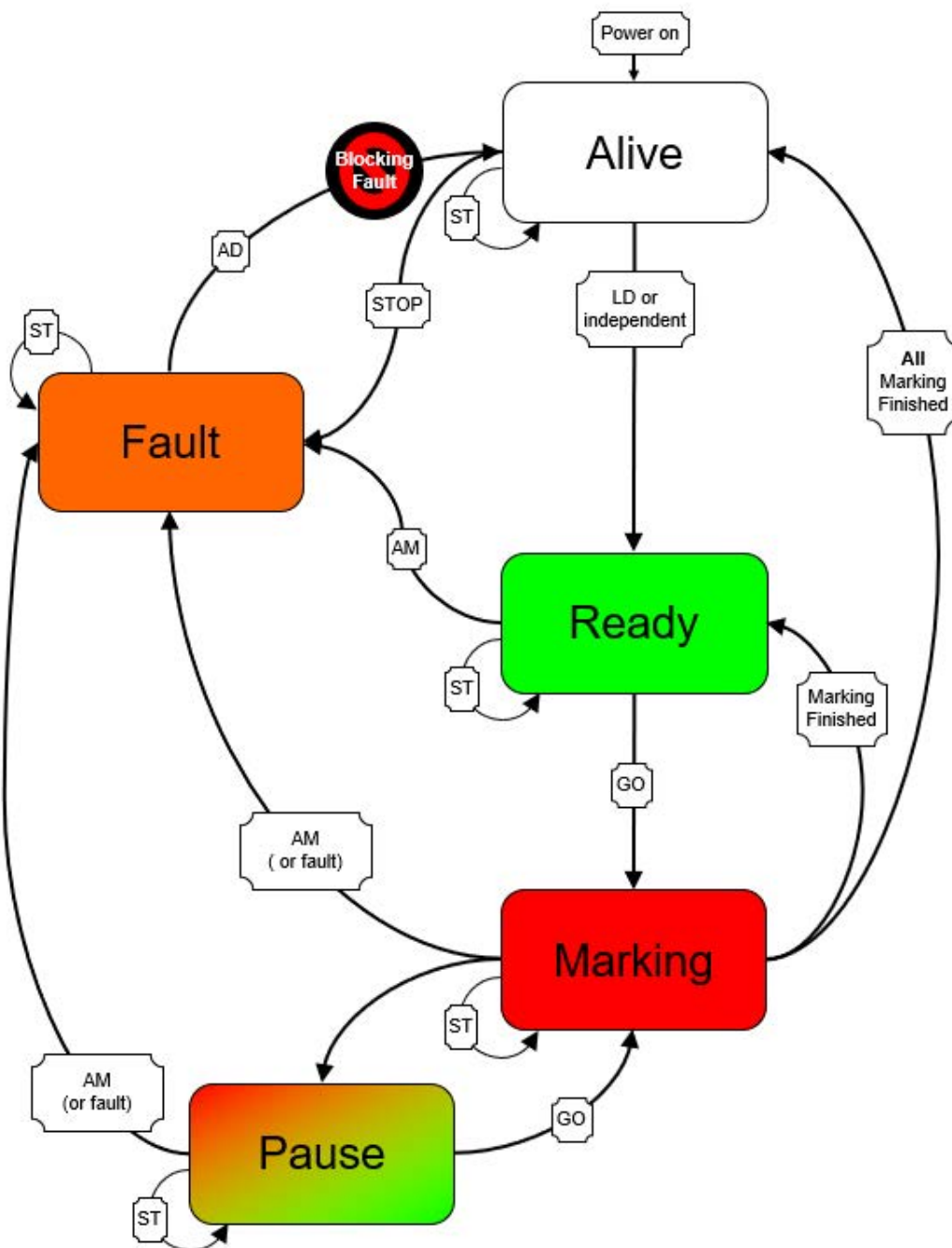
Introduction

5. Example: marking cycle

The "test.tml" file is stored in the machine. It marks a text to be updated by the production line PLC at each marking cycle (saving text in a variable - in this example: V0).

Procedure:

1. **Update the text to be marked (text to place in the variable: V0).**

2. **Load the marking file.**

3. **Launch the marking cycle.**

Example:

Command	Description	Machine's answer (machine mode)
`VS 0 "1234"`	Update: V0 = "1234"	`VS 1`
`LD "test.tml" 1 N`	Loading the marking files: "test.tml" Marking "One time"	`LD 1`

`GO`	Start marking	**`GO 1`**
		`GO M (Start marking)`
		`GO F (End of marking cycle)`

D. Command description

⚠️	**In this document:**
	• **text - red: UC500 machines only (or XCOM: UC500 compatibility)**
	• **text - green: XCOM machines only**

	AD
Description	Fault acknowledgement
Syntax used	`AD`
Parameter	None
Machine's answer (if execution without error)	`AD 1`
Possible error codes	ER 1 3 ER 2 1 ER 2 2 ER 2 3 ER 2 4 ER 2 14 **ER 3 1** **ER 4 1**
Example	**Command:** `AD` **Machine's answer:** `AD 1`

Command description

	AM
Description	Stop marking
Note	Interrupting a marking operation puts the machine into "Fault" mode. Fault acknowledgement: See: AD
Syntax used	AM
Parameter	None
Machine's answer (if execution without error)	AM 1
Possible error codes	ER 1 3 ER 2 2 ER 2 4 **ER 3 1** **ER 4 1**
Example	**Command:** AM **Machine's answer:** AM 1

Command description

<table>
<tr><td></td><td colspan="5">GF</td></tr>
<tr><td>Description</td><td colspan="5">File reception</td></tr>
<tr><td>Syntax used</td><td colspan="5"><code>GF "filename" option</code>
Returns the file whose name is specified between quotation marks.
The.tml extention is not mandatory for files.</td></tr>
<tr><td rowspan="4">Parameter</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Parameter</td><td>Type</td><td>Description</td><td colspan="2">Possible values</td></tr>
<tr><td><code>filename</code></td><td>UTF-8 text between " "</td><td>File name (Add "log/" - "csv/" to place the file in the log files or with the.csv files.)</td><td colspan="2"></td></tr>
<tr><td><code>option</code></td><td>character</td><td>Sending mode</td><td colspan="2">R, H</td></tr>
<tr><td></td><td colspan="5">R = Raw: sending bytes as they are in the file (only compatible with RS232)
H = Hexadecimal: sending hexadecimal codes of the bytes</td></tr>
<tr><td>Machine's answer (if execution without error)</td><td colspan="5">If: option = R: list of bytes the file is made of
If: option = H: list of the byte values which the file consists in, in their hexadecimal representation, without spaces</td></tr>
<tr><td>Possible error codes</td><td colspan="5">ER 1 2 ER 1 3 ER 1 9 ER 1 10 ER 1 11 ER 1 12
ER 2 1 ER 2 2 ER 2 3 ER 2 14
<strong style="color:green">ER 3 1
<strong style="color:green">ER 4 1</td></tr>
<tr><td>Example</td><td colspan="5">Command: <code>GF "XCOM.tml" H</code>
Machine's answer: EFBBBF544D4C28290D0A424828290D0A2053502835302C35302
C313030290D0A204F4828412C592C322C31290D0A454828290D0A424D28290D0A2
04242284F4E2C4C2C22222C4C3252290D0A20204D562831302E30302C31302E303
0290D0A2020464F28302C322E352C3130302C3130302C4C2C4E2C302C4E290D0A2
02053502835302C35302C313030290D0A20204D5028302E3030290D0A2020514C2
835302C33290D0A20204D4B28223132333422290D0A20454228432C31290D0A454
D28290D0A424628290D0A204F4828412C592C302C290D0A45462829

Corresponding marking file:
TML()
BH()
SP(50,50,100)
OH(A,Y,2,1)
EH()
BM()
BB(ON,L,"",L2R)
 MV(10.00,10.00)
 FO(0,2.5,100,100,L,N,0,N)
 SP(50,50,100)
 MP(0.00)
 QL(50,3)
 MK("1234")
EB(C,1)
EM()
BF()
OH(A,Y,0,)
EF()</td></tr>
</table>

Command description

	GI
Description	Information request
Syntax used	`GI param`

Parameter				
	Parameter	**Type**	**Description**	**Possible values**
	`param`	UTF-8 text	Name of the parameter requested	*, Head, Field, Sn, vSoftware, Date, Hour, Mac, Ip, SRAM, I, O

Machine's answer (if execution without error)	Value of the parameter requested
Possible error codes	ER 1 2 ER 1 3 ER 1 9 ER 2 3 **ER 2 1 ER 2 2 ER 2 14** **ER 3 1** **ER 4 1**

Example	**Command:** `GI *`
	 Machine's answer (XCOM):
	 `Head: XF510Sp`
	 `Field: 100*80`
	 `Sn: 2018-05-123456-01`
	 `vSoftware: v2.60`
	 `Date: Fri 04/09/2020`
	 `Hour: 11:16:00`
	 `MAC: E8:E0:8F:00:4A:A8`
	 `IP: 192.168.0.25`
	 `SRAM: 93%`
	 `I: 0.1, 1.1, 2.1, 3.1, 4.1, 5.1, 6.1, 7.1`
	 `O: 0.0, 1.0, 2.0, 3.0`
	 `GEN_UC: 8`

Command description

	GO	
Description	Launching the marking of a pre-loaded file with LD	
Note	While marking, only AM - QS - ST - GO commands (in the event of a break) are accepted. All other commands are refused until the end of the marking.	
Syntax used	GO	
Parameter	None	
Machine's answer (if execution without error)	GO 1	
	Marking in progress	GO M
	Marking suspended	GO P
	Fault: Marking stopped	GO S
	Marking done	GO F
	When there is a fault, the AD command is used to acknowledge it.	
Possible error codes	ER 1 3 ER 2 2 ER 2 3 ER 2 4 ER 2 15 **ER 3 1** **ER 4 1**	
Example	**Command:** GO **Machine's answer:** GO 1 GO M GO F	

Command description

	HP			
Description	On-line help (human mode)			

Syntax used	`HP`
	Help of all commands
	`HP command`
	Help for the specified command			

Parameter				

Parameter	Type	Description	Possible values
`command`	Command	Name of the command on which help must be brought	2 characters or nothing

Machine's answer (if execution without error)	Human mode: help message on the specified command or list of commands separated by spaces - See: VM Machine mode: `HP 1`
Possible error codes	ER 1 3 ER 1 7 ER 2 3

Example	Configuration of the response mode (human/machine): Human mode
	
	Command: `HP`
	Machine's answer:
	`HP is the online help command.`
	`Try: HP command`
	`Available commands are: AD,AM,GF,GO,GI,HP,LD,LS,MV,OG,PF,QS,QT,RM,`
	`SI,ST,VG,VM,VS`
	`--End of help--`
	
	Command: `HP GF`
	Machine's answer:
	`Online help for the GF (Get File) command.`
	`Syntax : GF filename mode`
	`Effect  : Downloads a file from the machine`
	`Parameters:`
	`        filename      : name of the file, between quotes`
	`        mode          : raw (R) or hexa (H)`
	`--End of help--`

Command description

	LD
Description	Loading of a TML file with the number of markings requested and their mode
Syntax used	`LD "filename.tml" nbmarking mode`
Note	The .tml extention is not mandatory for files. The name of the file must take into account caps.
Parameter	(see table below)

Parameter	Type	Description	Possible values
`filename.tml`	UTF-8 text between " "	Name of the file to load for marking	
`nbmarking`	whole	Number of markings to execute 0 to activate the infinite mode Independent marking: nbmarking = 0	included between 0 and 9999
`mode`	character	Marking mode	A = Independent N = Normal S = Simulation SP = Simulation + Pause SS = Simplified simulation

Machine's answer (if execution without error)	`LD 1`
Possible error codes	ER 1 2 ER 1 3 ER 1 4 ER 1 5 ER 1 9 ER 1 11 ER 2 1 ER 2 2 ER 2 3 ER 2 14 ER 2 15 **ER 3 1** **ER 4 1**
Example	**Command:** `LD "test.tml" 0 A` => loading the marking files: test.tml "Infinite" independent marking **Machine's answer:** `LD 1`

Command description

	LS
Description	List of files stored on the machine according to the specified mask

Syntax used	`LS`
	List of files stored on the machine
	`LS mask`
	List of the files corresponding to the specified mask

Parameter	

Parameter	Type	Description	Possible values
`mask`	UTF-8 text	Rule of the files to display	* *.lo3 *.po3 *.tml /csv/*.csv /log/*

Note	The mask includes 2 parts: the name and the file extension. The star * is used to specify any name.

Machine's answer (if execution without error)	`NbFilesFound`
	`File1.ext`
	`File2.ext`
	`...`
	`FileN.ext`
	
	`NbFilesFound`: number of the files corresponding to the mask

Possible error codes	ER 1 3 ER 1 4 ER 1 5 ER 2 1 ER 2 3 ER 2 5 ER 2 14 **ER 2 2**

GRAVOTECH

Command description

<table>
<tr><td></td><td colspan="4">MV</td></tr>
<tr><td>Description</td><td colspan="4">Moves the stylus according to specified parameters.</td></tr>
<tr><td>Syntax used</td><td colspan="4"><code>MV relative X Y pen speed quality stroke dotnb</code></td></tr>
<tr><td rowspan="2">Parameter</td><td colspan="4"></td></tr>
<tr><td colspan="4">

Parameter	Type	Description	Possible values
`relative`	character	Absolute distance / relative distance	A, R
`X`	real	Movement: X (mm)	included between -Xmax and Xmax
`Y`	real	Movement: Y (mm)	included between -Ymax and Ymax
`pen`	character	Type of movement: Up / Down	U, D
`speed`	whole	Movement speed	included between 0 and 100
`quality`	whole	Marking quality (if: `pen` = D)	included between 0 and 100
`stroke`	whole	Marking force (if: `pen` = D)	included between 0 and 100
`dotnb`	whole	Number of impacts - Electromagnetic machines	included between 0 and 65535

</td></tr>
<tr><td>Note</td><td colspan="4">The first 3 parameters are required.
Optional parameters: `pen` - `speed`
By default: `pen` = U
By default: `speed` = Speed of the machine
If: `pen` = D => The last 3 parameters are taken into account.</td></tr>
<tr><td>Machine's answer (if execution without error)</td><td colspan="4"><code>MV 1</code></td></tr>
<tr><td>Possible error codes</td><td colspan="4">ER 1 4 ER 1 6 ER 1 8
ER 2 1 ER 2 2 ER 2 3 ER 2 14
ER 3 1
ER 4 1</td></tr>
<tr><td>Example</td><td colspan="4">Command: <code>MV A 32.50 20</code>
=> Movement without marking: X = 32.50 mm - Y = 20 mm coordinates

Machine's answer: <code>MV 1</code></td></tr>
</table>

Command description

	OG			
Description	Request of return to origin			
Syntax used	`OG param`			
Parameter				
	Parameter	**Type**	**Description**	**Possible values**
	`param`	character	Peripheral(s) If: no load => X	D, Z, X, No load
Machine's answer (if execution without error)	`OG 1`			
Possible error codes	ER 1 3 ER 1 4 ER 1 15 ER 1 16 ER 1 17 ER 1 18 ER 1 19 ER 2 1 ER 2 2 ER 2 3 **ER 1 22 ER 1 23** **ER 3 1** **ER 4 1**			
Example	**Command:** `OG` **Machine's answer:** `OG 1`			

Command description

	PF			
Description	Sending a file to the machine memory			
Syntax used	`PF "filename" option bytelist`			
Parameter				

Parameter	Type	Description	Possible values
`filename`	UTF-8 text between " "	File name	
`option`	character	Sending mode	R, H
`bytelist`	byte series	List of the byte values which the file consists in, in their hexadecimal representation, without spaces	

R = Raw: sending bytes as they are in the file (only compatible with RS232)
H = Hexadecimal: sending hexadecimal codes of the bytes

Note	The.tml extention is not mandatory for files (add "/log/" - "/csv/" to place the file in the log files or with the.csv files).
Machine's answer (if execution without error)	`PF 1`
Possible error codes	ER 1 2 ER 1 3 ER 1 5 ER 1 11 ER 1 12 ER 1 13 ER 1 14 ER 1 20 ER 2 1 ER 2 2 ER 2 3 ER 2 14 **ER 3 1** **ER 4 1**

.../...

Command description

	PF

Example	**Marking file to send:**
	TML()
	BH()
	SP(50,50,100)
	OH(A,Y,2,1)
	EH()
	BM()
	BB(ON,L,"",L2R)
	 MV(10.00,10.00)
	 FO(0,2.5,100,100,L,N,0,N)
	 SP(50,50,100)
	 MP(0.00)
	 QL(50,3)
	 MK("1234")
	EB(C,1)
	EM()
	BF()
	OH(A,Y,0,)
	EF()
	
	Command: `PF "XCOM.tml" H`
	`EFBBBF544D4C28290D0A424828290D0A2053502835302C35302C313030290D0A20`
	`4F4828412C592C322C31290D0A454828290D0A424D28290D0A204242284F4E2C4C`
	`2C22222C4C43252290D0A20204D562831302E30302C31302E3030290D0A2020464F`
	`28302C322E352C3130302C3130302C4C2C4E2C302C4E290D0A202053502835302C`
	`35302C313030290D0A20204D5028302E3030290D0A2020514C2835302C33290D0A`
	`20204D4B28223132333422290D0A20454228432C31290D0A454D28290D0A424628`
	`290D0A204F4828412C592C302C290D0A45462829`
	
	Machine's answer: `PF 1`

Command description

	QS
Description	End of a telnet session Without exiting controlled mode
Syntax used	QS
Parameter	None
Remarks	No use for RS232 communication
Machine's answer (if execution without error)	CCU does not respond
Possible error codes	ER 1 3 ER 2 5 **ER 2 4**
Example	**Command:** QS **Machine's answer:** CCU does not respond

Command description

	QT
Description	End of a telnet session **Exit controlled mode** **Function identical to QS**
Syntax used	QT
Parameter	None
Remarks	No use for RS232 communication
Machine's answer (if execution without error)	CCU does not respond
Possible error codes	ER 1 3 ER 2 1 ER 2 2 ER 2 3 ER 2 4 ER 2 14
Example	**Command:** QT **Machine's answer:** CCU does not respond

GRAVOTECH

Command description

	RM
Description	Deleting files
Syntax used	`RM mask`
Parameter	

Parameter	Type	Description	Possible values
`mask`	UTF-8 text	Name or file mask	

Note	The mask includes 2 parts: the name and the file extension. The star * is used to specify any name (add "/log/" - "/csv/" to place the file in the log files or with the.csv files).
Machine's answer (if execution without error)	`RM 1`: Deleting files `RM 0`: File(s) not deleted

Possible error codes	`ER 1 2` `ER 1 3` `ER 1 5` `ER 1 9`
	`ER 2 1` `ER 2 3` `ER 2 14`
	ER 2 2
	ER 3 1
	ER 4 1

Example	`RM "exemple.tml"`
	
	`RM "*.tml"`
	
	`RM "152.lo3 "`
	
	Command: `RM example.tml`
	Machine's answer: `RM 1`
	
	Command: `RM *.tml`
	Machine's answer: `RM 1`

Command description

	SI
Description	Definition of information
Syntax used	`SI param:value`

Parameter	

Parameter	Type	Description	Possible values
`param`	UTF-8 text	Value of the parameter	date time
`value`	DD/MM/YYYY	Definition of the date	DD = day MM = month YYYY = year
	hh:mm:ss	Definition of the time	hh = hour mm = minute ss = second

Machine's answer (if execution without error)	`SI 1`

Possible error codes	`ER 1 2   ER 1 3   ER 1 4   ER 1 9`
	`ER 2 1   ER 2 3   ER 2 14`
	ER 2 2
	ER 3 1
	ER 4 1

Example	**Command:** `SI date:01/12/2018` **Machine's answer:** `SI 1` **Command:** `SI time:12:00:00` **Machine's answer:** `SI 1`

Command description

ST	
Description	State of machine operation
Syntax used	`ST`
Note	**Do not use this command to continuously monitor the machine state. Always wait for the response(s) before sending another command.** **The behavior of the machine is indeterminate if these rules are not respected.**
Machine's answer (if execution without error)	`ST state IOs` State: whole - 0 to 32

0: Alive	17: Row content in CSV file is not valid
1: Ready to mark	18: Separator error in CSV file
2: Marking in progress	19: No file found
3: Marking paused	20: No file counter found
4: Origin fault	21: Not a numeric value
5: Stop mark activated	22: Fault called
6: Datamatrix error	23: Incorrect batch in CSV file
7: Marking is off-limits	24: Fault detected
8: Critical temperature	25: SV510 variator fault
9: X origin error	26: File already open
10: Y origin error	27: Out of range VARTML parameter
11: Origin return error	28: Out of memory
12: COM error	29: FTP connection failed
13: XY origin error	30: FTP not activated
14: Waiting AD command	31: Result overflow
15: CSV file error	32: Add: not a numeric value
16: Row in CSV file is not valid	

IOs: whole (8 bits) - gives the state of the signals (2 Inputs - 3 Outputs) of the DB9 connector according to encoding:

Bit #	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Function	Mode: independent	RFU*	**RFU* Controlled mode**	(O) Marking	(O) Fault	(O) Ready	(I) Stop	(I) Start

O = Output, I = Input
* Reserved for Future Use (reserved for future use)

Possible error codes	**ER 1 3** **ER 2 5** **ER 3 1**
Example	**Command:** `ST` **Machine's answer:** `ST 0 2` (= state of machine operation: alive)

Command description

	VG
Description	Interrogation of a variable's value
Syntax used	`VG var#`

Parameter				
	Parameter	**Type**	**Description**	**Possible values**
	`var#`	Text	Number of the variable	0 to 9 *

Machine's answer (if execution without error)	**Variable contents** **Variable requested followed by = and its value** If: `var#` = * => all variables followed by = and their value
Possible error codes	ER 1 2 ER 1 3 ER 1 4 ER 1 9 ER 2 3 **ER 2 2** **ER 3 1** **ER 4 1**

Example	If:
	V0=example0
	V1=example1
	
	Command: `VG 0`
	Machine's answer: `example0`
	                    `V0=example0`
	
	Command: `VG 1`
	Machine's answer: `example1`
	                    `V1=example1`
	
	Command: `VG *`
	Machine's answer: `V0=example0`
	                   `V1=example1`
	                   `...`

Command description

VM	
Description	Configuration of the response mode (human/machine)
Note	By default, the machine is configured in machine response mode. The human mode is useful in TCP/IP connection with a telnet client. It provides answers in English and access to help (see: HP). Its use is recommended for demonstrations and tests in the development phase. In all other cases, the machine mode should be preferred because its responses are shorter and do not change or change very little from one firmware version to another.
Syntax used	`VM mode`

Parameter				
	Parameter	**Type**	**Description**	**Possible values**
	`mode`	character	Configuration of the response mode (human/ machine)	H, M

Machine's answer (if execution without error)	Human mode: `Human answer mode` Machine mode: `VM 1`
Possible error codes	ER 1 2 ER 1 3 ER 1 9 ER 2 3 **ER 4 1**
Example	**Command:** `VM H` **Machine's answer:** `Human answer mode` **Command:** `VM M` **Machine's answer:** `VM 1`

Command description

	VS			
Description	Assigning a variable			
Syntax used	`VS var# text`			
Parameter	**Parameter**	**Type**	**Description**	**Possible values**
	`var#`	Text	Number of the variable	0 to 9
	`text`	UTF-8 text between " "	Text to place in the variable	
Machine's answer (if execution without error)	`VS 1`			
Possible error codes	ER 1 2 ER 1 3 ER 1 4 ER 1 9 ER 1 11 ER 2 3 **ER 2 2** **ER 3 1** **ER 4 1**			
Example	**Command:** `VS 0 "ABCD"` **Machine's answer:** `VS 1`			

E. Error messages

Errors are signaled via codes, more or less explicit depending on the mode configured. The error codes always start with a ER prefix and a space. They are composed of 2 elements separated by a space:

- an error-type code
- a code specifying the error

Summary table of the error codes

| Error | Type | | Details | |
Code	Code	Description	Code	Description
ER	0	Reserved	0	Reserved
	1	Syntax error	0	Reserved
			1	Unknown command
			2	Not enough parameters
			3	Too many parameters
			4	Wrong parameter
			5	Cannot open file
			6	Origin return required
			7	Unknown parameter
			8	Out of range
			9	Wrong parameter value
			10	File size limit exceeded
			11	Parameter is not a string
			12	Invalid file extension
			13	Memory full
			14	No UTF-8 format
			15	Origin return error
			16	Origin fault
			17	XY origin error
			18	X origin error

Error messages

Error	Type		Details	
Code	**Code**	**Description**	**Code**	**Description**
ER	1	Syntax error	19	Y origin error
			20	Unknown format file
			21	Device not detected
			22	**Z origin error**
			23	**R origin error**
	2	Context error	0	Reserved
			1	Marking paused
			2	Fault detected
			3	Marking is already in progress
			4	No marking loaded
			5	Only in ethernet mode
			14	Marking is ready
			15	Reset activated
	3	Processing error	0	Reserved
			1	System error
	4	**Authorization error**	0	**Reserved**
			1	**Command reserved to the master**