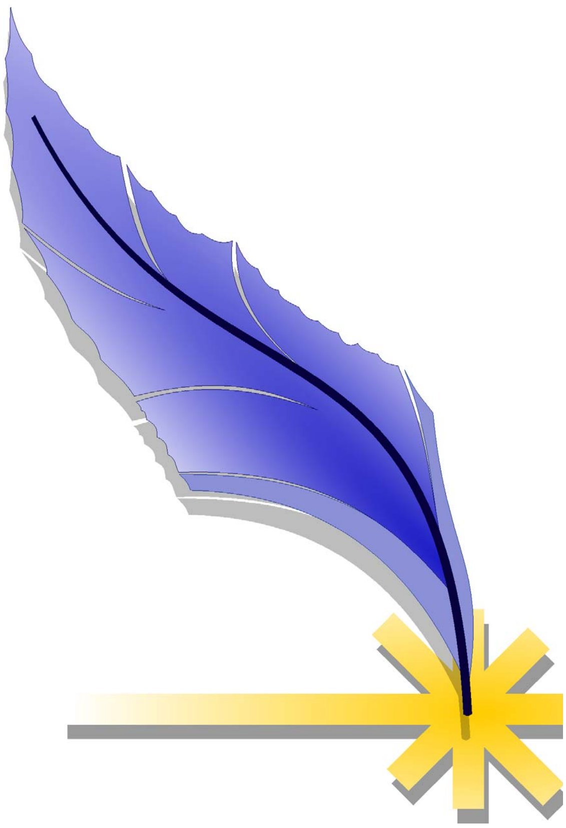
TCP Server Protocol

USER MANUAL



Communication Protocol 4.0

DATALOGIC

Datalogic S.r.l.
Via S. Vitalino, 13
40012 Calderara di Reno (BO)
Italy
Tel. +39 051 3147011
Fax +39 051 3147205

Electronic versions of this document may be downloaded from the Datalogic website (www.datalogic.com). If you visit our website and would like to make comments or suggestions about this or other Datalogic publications, please let us know via the "Contact" page.

Disclaimer

Datalogic has taken reasonable measures to provide information in this manual that is complete and accurate, however, Datalogic shall not be liable for technical or editorial errors or omissions contained herein, nor for incidental or consequential damages resulting from the use of this material. Datalogic reserves the right to change any specification at any time without prior notice.

Trademarks

Datalogic and the Datalogic logo are registered trademarks of Datalogic S.p.A. in many countries, including the U.S.A. and the E.U.

Arex™, Ulyxe™, Eox™, Vlase™, Matrix™, Marvis™ and Lighter™ Suite are trademarks of Datalogic S.p.A. and/or its affiliates. All other trademarks and brands are property of their respective owners.

Ed: 03/2020
This manual refers to Lighter™ Suite software version 7.3 and later

TABLE OF CONTENTS

PREFACE

ABOUT THIS MANUAL

This User Manual (UM) provides users with information on how to use the communication protocol. Other publications associated with this product can be downloaded free of charge from the website listed on the back cover of this manual.

Manual Conventions

The symbols listed below are used in this manual to notify the reader of key issues or procedures that must be observed when using the laser marker:

NOTE

Notes contain information necessary for properly diagnosing, repairing and operating the laser marker.

CAUTION

The CAUTION symbol advises you of actions that could damage equipment or property.

WARNING

The WARNING symbol advises you of actions that could result in harm or injury to the person performing the task.

TECHNICAL SUPPORT

Support Through the Website

Datalogic provides several services as well as technical support through its website. Log on to (www.datalogic.com).

For quick access, from the home page click on the search icon, and type in the name of the product you're looking for. This allows you access to download Data Sheets, Manuals, Software & Utilities, and Drawings.

Hover over the Support & Service menu for access to Services and Technical Support.

CHAPTER 1
TCP SERVER PROTOCOL

FOREWORD

Lighter™ Suite integrates a TCP/IP protocol called TCPServer able to fully control the laser marker.

The TCPServer protocol implements a full duplex communication between a master and a slave.

Laser Engine represents the slave, interpreting and executing the commands provided by a third-parts service that represent the master.

Depending on Lighter™ Suite version the TCPServer protocol is implemented in two ways:

- starting from Lighter™ Suite 6.3.0 the TCPServer protocol has been embedded in the source code of Lighter™ Suite. Due to this, it is no longer necessary to use the TcpServer script.
- for previous Lighter™ Suite versions the TCPServer protocol was implemented only by a LaserEngine script ("`TcpServer.xqs`").

This documentation refers both to TCPServer protocol embedded in Lighter™ Suite and previous TCPServer script.

DISABLING THE WINSAT TASK

WinSAT is a Windows Scheduled Task, which is executed weekly every Sunday at 1 AM (by default, on our Embedded PCs): more information about this task can be found at the following link https://en.wikipedia.org/wiki/Windows_System_Assessment_Tool.

The execution of this Task can create issues to the Laser Marker during communication with PLC, as it interferes with the communication stacks which are included inside Lighter: because of this, if the Laser Marker is going to be connected to the PLC via Prof-iNet IO/Ethernet IP/ TcpServer, the user must manually disable this scheduled task, so to be sure that the PC on board the Laser Marker doesn't execute such task.

Check that the Laser Marker has the WinSAT task disabled. If not, follow these steps:

- System running Windows 7:

  - Skip to step 1

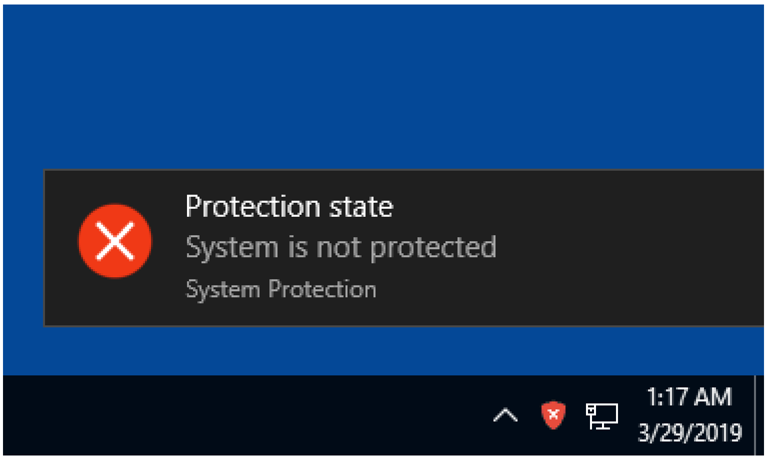
- System running Windows 10:

  - **Disable system protection** (see laser marker user's manual -> Disable the system protection...)

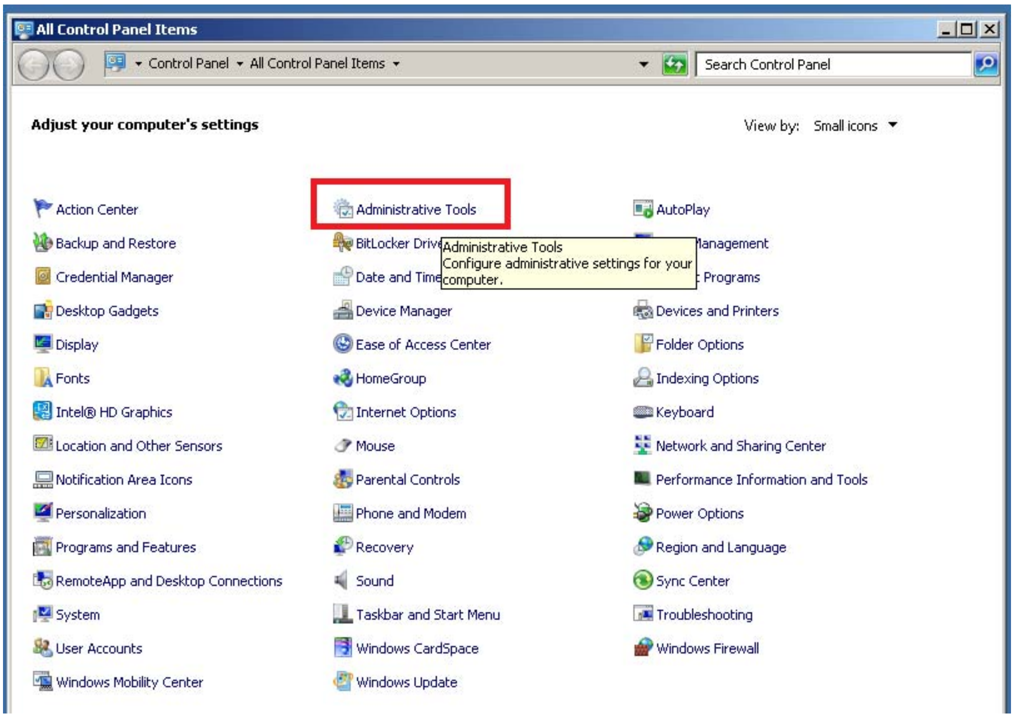
NOTE

When the System Protection is disabled the system is not protected against disk corruption or malware attacks. Disable the protection only for the time necessary to make disk changes.

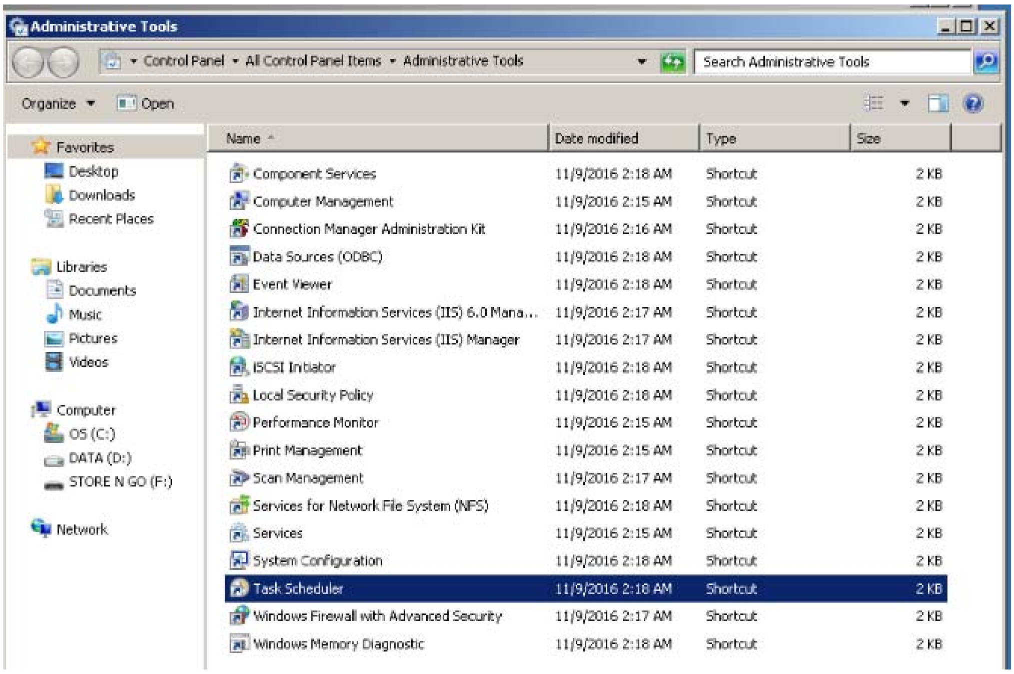
- Wait for the operating system to restart
- Check that the System protection is disabled (red icon):



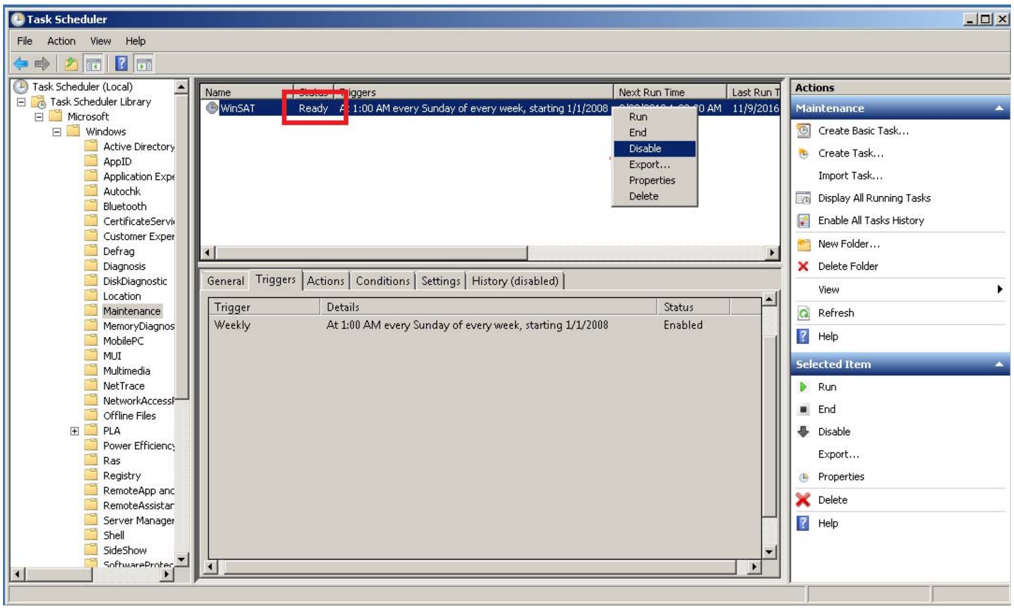
1. Open the "**Administrative Tools**" from the Control Panel:



2. Select the "**Task Scheduler**":



3. Follow the filepath Task Scheduler *(Local)\Task Scheduler Library\Microsoft\Windows\Maintenance* and check WinSAT. If the Status is '**Ready**', then right-click on the WinSAT task and select '**Disable**'.



To make these settings permanent on the Laser Marker, please follow this procedure:

- System running Windows 7:

  - Close all the open windows

  - Double click on the '`Save-Data.bat`' icon, present in the Desktop of the Laser Marker

  - Restart Windows.

- System running Windows 10:

  - Close all the open windows

  - **Enable system protection** (see laser marker User's Manual -> Enable the system protection...)

NOTE

When the System Protection is disabled the system is not protected against disk corruption or malware attacks.

  - Wait for the operating system to restart

  - Check that the System protection is enabled (green icon):

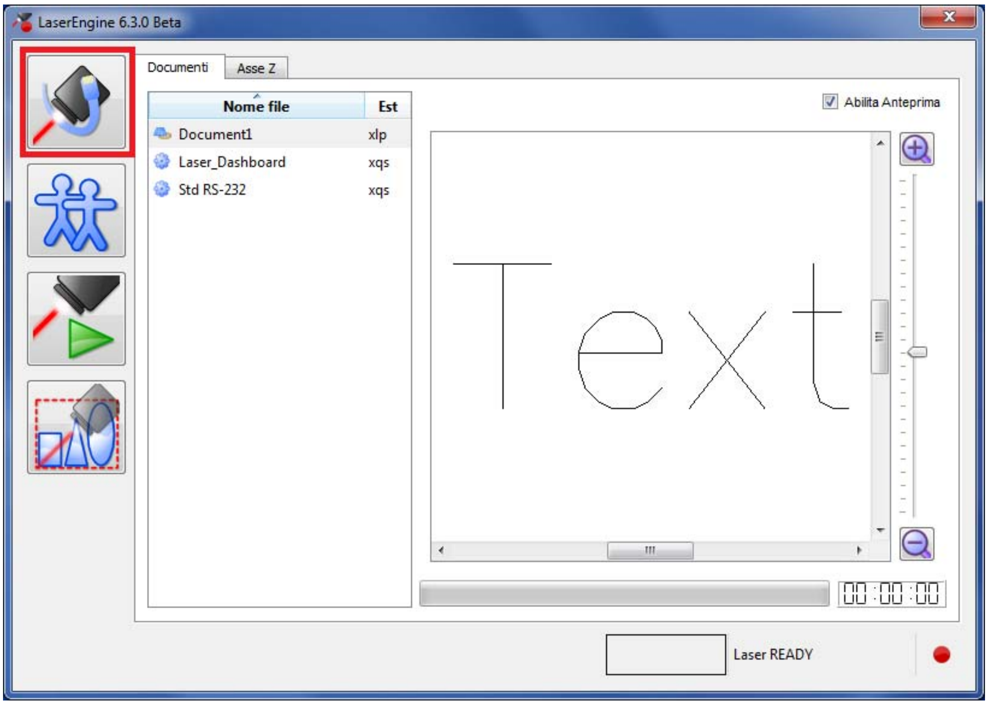
EMBEDDED TCP SERVER

Starting from Lighter™ Suite 6.3.0 the TCP Server protocol has been integrated in the source code of Lighter™ Suite. Due to this, it is no longer necessary to use the TcpServer script. These paragraphs refer only to TCP Server protocol embedded in Lighter™ Suite 6.3.0 or higher.

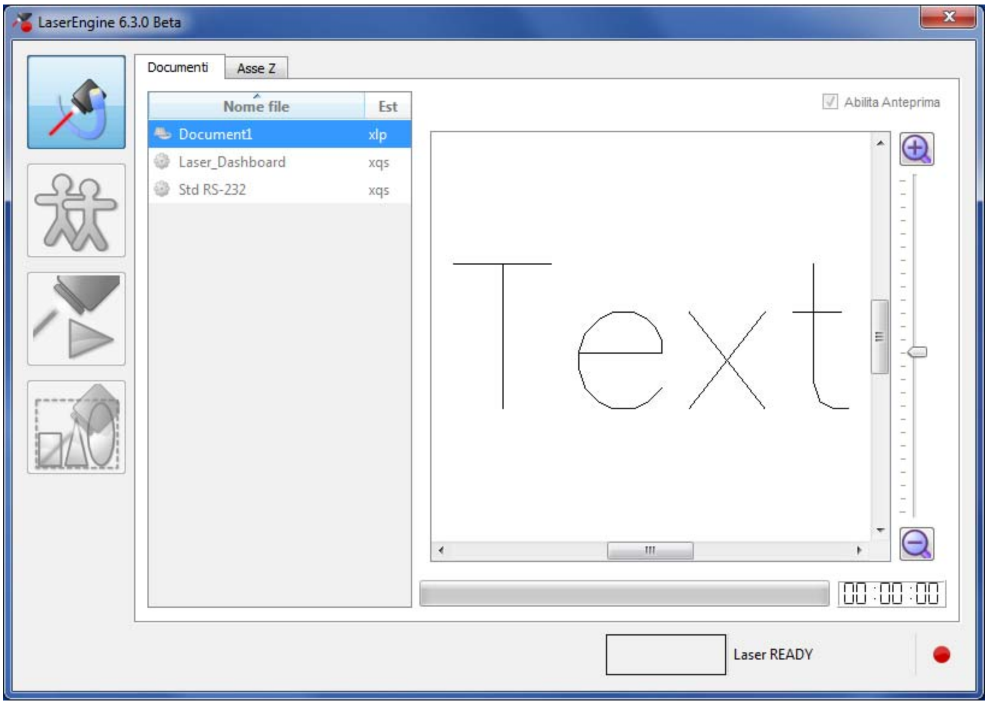
Enable TCP Server

In order to work using TCPServer protocol, Laser Engine must be set in remote mode.

1. Click "**To Remote"** button in Laser Engine GUI:

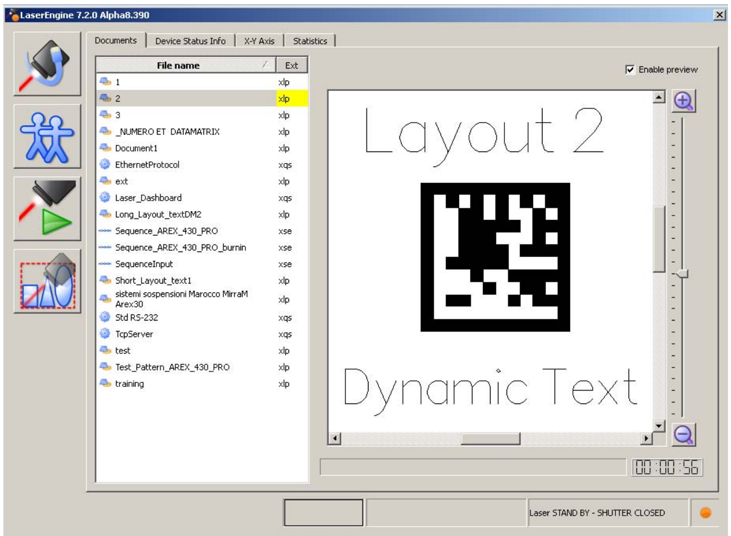


2. Laser Engine is now set in listening mode, waiting for a connection and therefore for a valid command sequence:

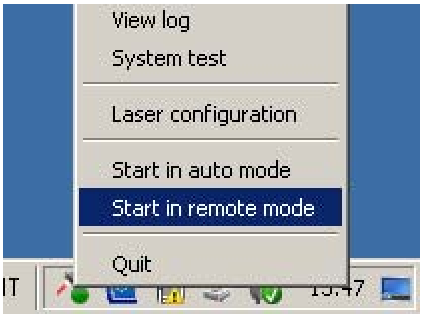


Set Laser Engine in remote mode

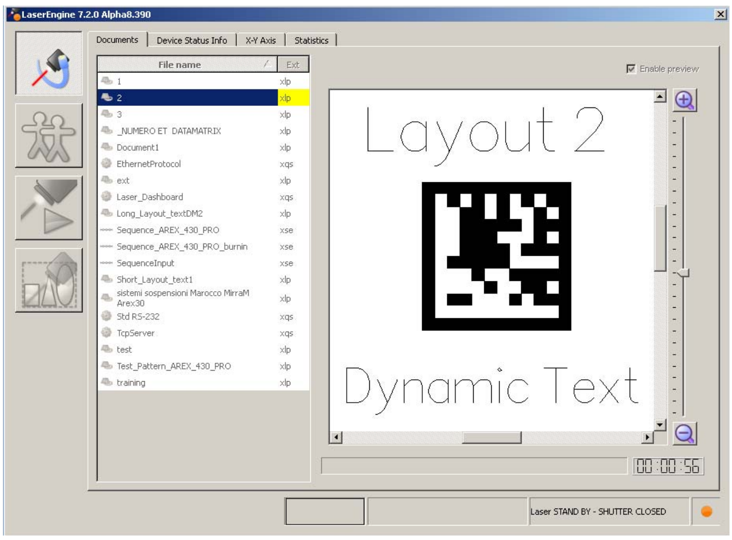
To activate ProfiNet I/O communication on the Datalogic Laser Marker side, it is sufficient to set Laser Engine in Remote Mode, by clicking on the indicated Push Button from the Laser Engine GUI:



In order to have the Laser Engine starting in Remote Mode also after the Laser Marker has rebooted, enable 'Start in remote mode' from the Laser Engine icon in the Windows tray bar.



When in Remote Mode, the Laser Engine GUI will look like this (having all the Push Buttons disabled except for the one which brings Laser Engine back to Local Mode).

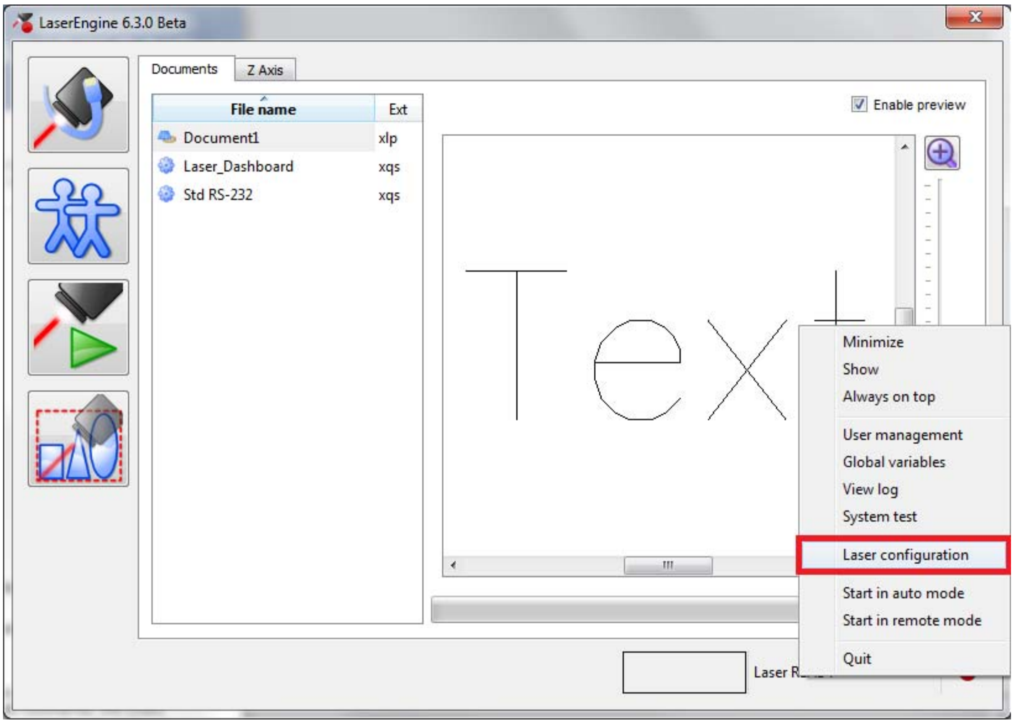


Configure TCP Server Port

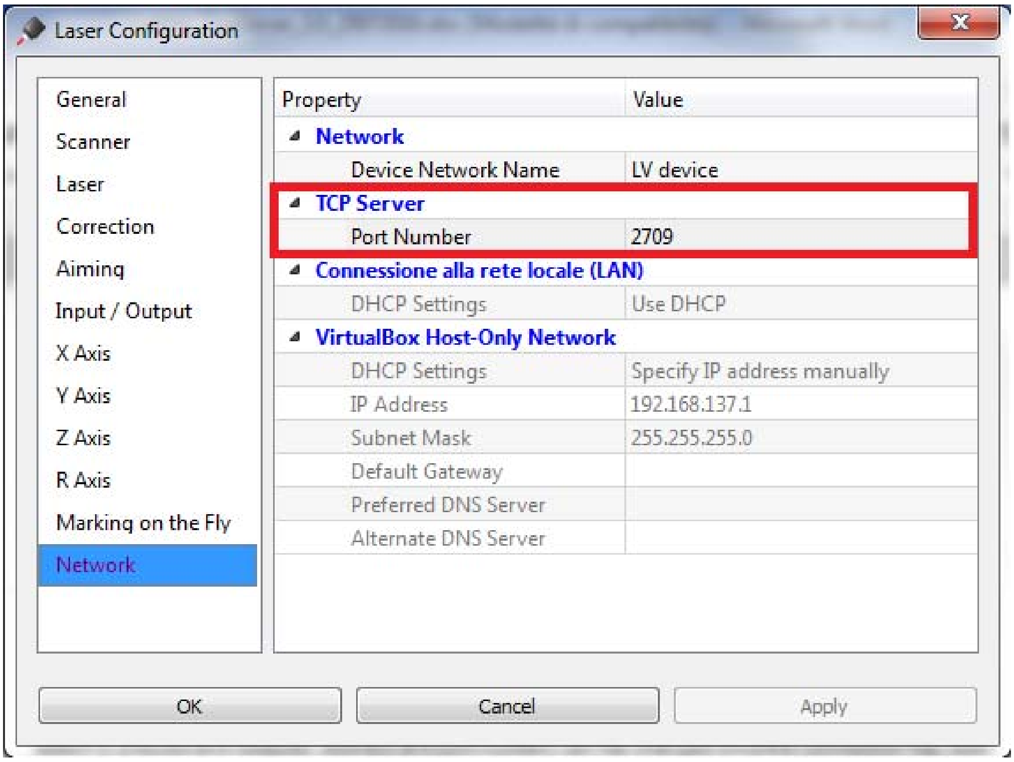
Laser Engine provides a configuration interface able to set the port used by the TCP/IP protocol.

As default the TCP/IP protocol port is set to **2709**.

To change the TCP/IP protocol port value open Laser Configuration:



Select Network section and then in TCP Server set the Port Number to the desired value:



Press “**Apply**” and “**OK**”

TCP SERVER SCRIPT

For Lighter™ Suite versions up to 6.2.3 the TCPServer protocol was implemented by a LaserEngine script only.

The script "`TcpServer.xqs`" can be used also in later Lighter™ Suite versions.

Pulse Profile management for MOPA laser source is not available for TCPServer protocol implemented by script.

NOTE

TCP Server script supports the MARVIS™ commands starting from Lighter™ Suite 7.1.1.

Distance Sensor and Green Spot commands are NOT supported in the script.

Customization

This project is designed to provide a common application that can be used for testing laser application capabilities and as a project for easy customization where third party applications need to control the laser system. The user needing this project works for its purposes should check and set just the TCP port that the script is supposed to listen.

Project Description

The project consists of 6 source files:

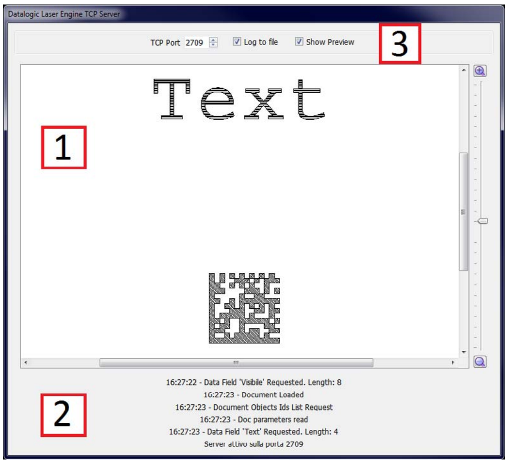
- **main.qs** - hosts the entry point function "main", the objects allocation and the main interaction, the layouts table
- **GUI.qs** - defines the graphical user interface through the classMainWindow class
- **Utils.qs** - contains some functions used in many places of the code
- **Protocol.qs** - defines the protocol management and parsing methods
- **Constants.qs** - contains some constant strings and values, used to report the status of the system
- **Crc.qs** - contains functions used to calculate final part of the message

Project's GUI Description

The TCPServer script running on the Laser is provided with a GUI interface:

1. Document preview - a graphical area where the current document is shown as changes occurs (according to "Show Preview" option").
2. Received commands - a list of last received commands.

3. Settings - Options to specify the listening port, if a file trace is active, if the preview control should show every update.



LASER REMOTE DASHBOARD

The LaserRemoteDashboard is an application distributed with source code, that can be used by developers as an example to build their own application able to control the laser marker using the TCP/IP protocol.

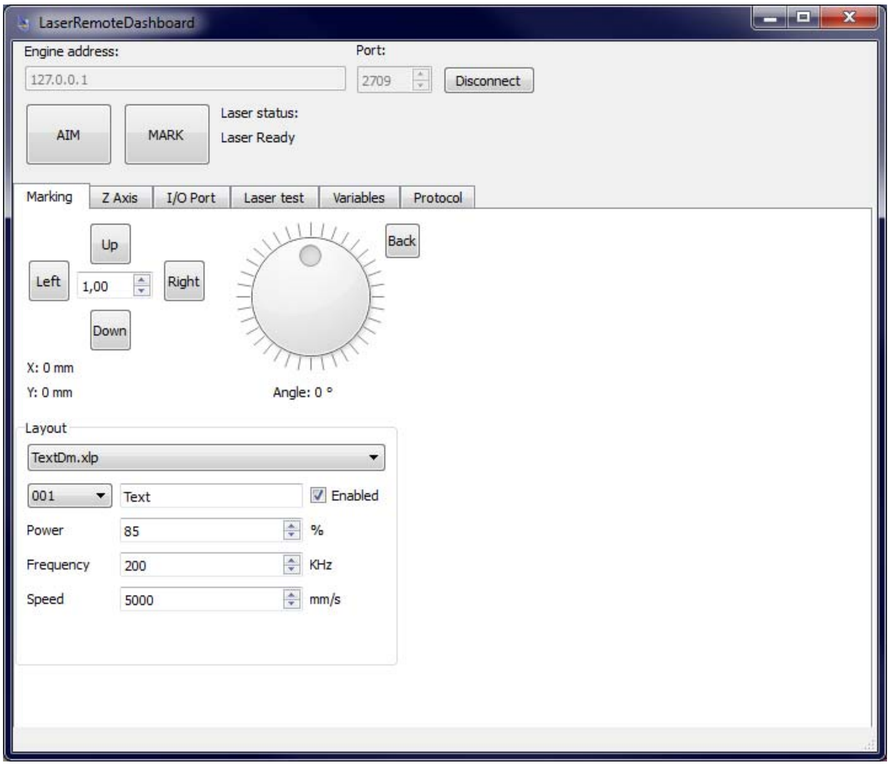
Contact DATALOGIC "Technical Support" on page 2 to get the LaserRemoteDashboard installer for Windows.

This application sends TCP/IP commands acting as a Master while Laser Engine is set in remote mode acting as a Slave.

LaserRemoteDashboard can run locally or remotely, according to network capabilities. The application interface has an editable control to let the user specify the IP address of the laser marker where Laser Engine (slave) is running, and another editable control to let the user specify the port where the Laser Engine (slave) is listening to the commands. The connection is activated using the Connect/Disconnect button. IP address and port numbers can't be changed once the connection has been established until a disconnection occurs.

Two buttons allow the user to start the projection of the Aiming beam or to mark the current layout. Here nearby the laser device status is reported.

Below these controls always visible, there are several tabs that can be shown alternatively: Marking, Z Axis, I/O port, Laser test, Variables, Protocol.



Marking Panel

This panel allows to control layout placement and orientation through an offset that can be applied in the four directions, and a rotation. The working layout can be chosen between those stored on device side and provided to the operator through a dropdown control.

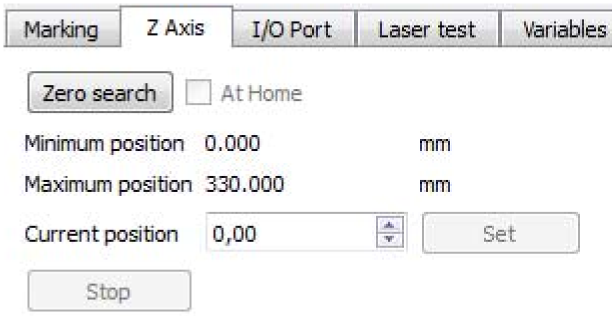
If the document has programmable content (objects with assigned ID) they can be accessed through a another dedicated dropdown control. These objects can be modified for textual content and enabled or disabled.

Main laser document parameters as power, frequency and speed can be changed through dedicated controls.

Z Axis Panel

This panel allows the user to simply operate with the most used axis. Here the operator can start a search for the home sensor, verify minimum and maximum settings, set a new position for the axis and control when the movements complete.

This panel can be disabled if the Z Axis is disabled in the laser device configuration.



I/O Port Panel

This panel is intended to demonstrate how to interact with I/O integrated in the control board.

NOTE

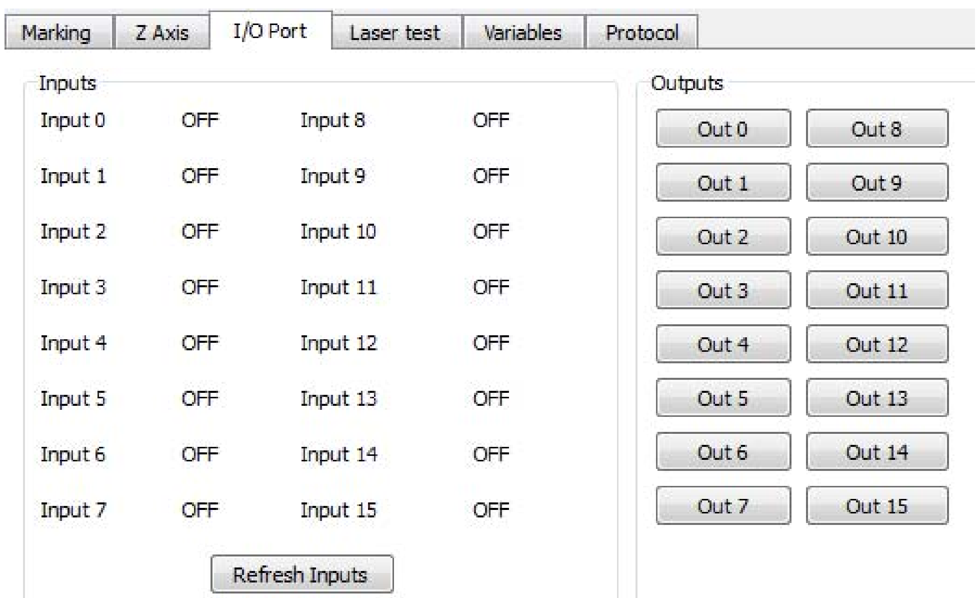
Unlike the Lighter Engine System Test, this panel is not aligned to device and I/O configuration. So this panel is fully enabled even with USB Ulyxe device which has no I/O. Axis settings which allocate some inputs and outputs on most laser devices, do not affect the controls. By the way setting an output on a device where not allowed results in no effect.

In order to avoid continuous polling of the inputs values, refresh is executed just on user interaction with the "Refresh Inputs" button.

Clicking an output button triggers an immediate output change request. These button are two states fashion, so a pressed button denotes a stimulated output while a released one is for unsolicited output.

NOTE

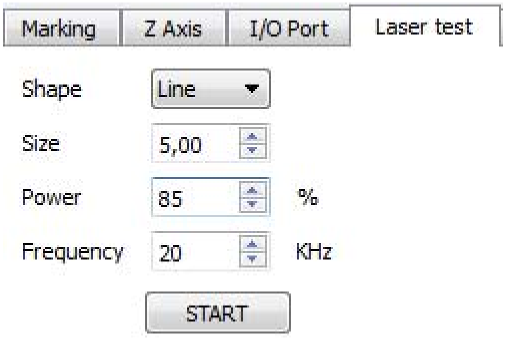
Please verify the available output according your device model and settings.



Laser Test Panel

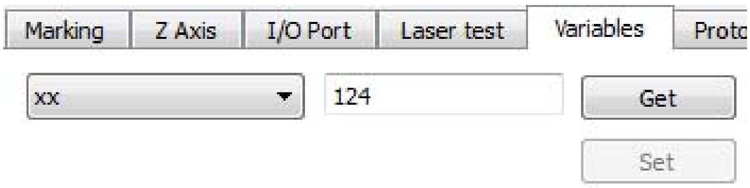
This panel allows to control the laser in order to project or mark (according the laser device state) some simple figures in continuous way. This functionality helps operator to test laser correct behaviour and focus distance.

This panel replicates the "System test" in Lighter Engine providing same shapes: line, square, circle, dot.



Variables Panel

This panel provides read/write capabilities for global variables stored on laser device side. All the variables, both numeric and literals are provided together in a dropdown list. Current value is shown on name side and a "Get" button allows to force an update request. Once the value is changed, the "Set" button allows the variable value over-write.



Protocol Panel

This panel allows to look at the commands exchange. Here the Outgoing area shows the outgoing bytes in hexadecimal format on the left and in ASCII on the right.

The Incoming area shows incoming data in hexadecimal format on the left, ASCII on the right.

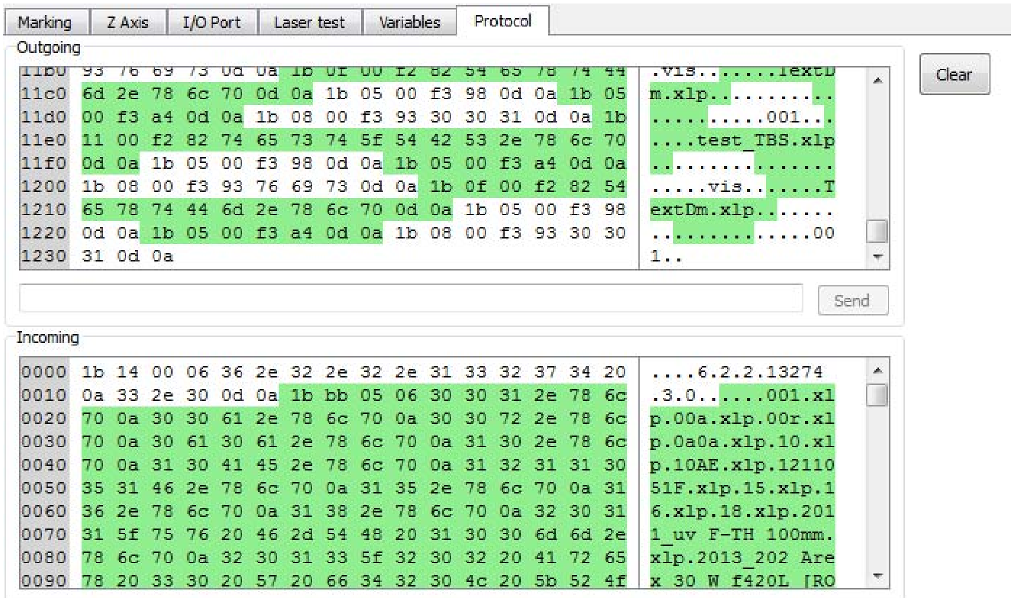
NOTE

Please take care that even if the right panel shows data in ASCII format, messages are in Unicode UTF-8.

The user can also compile its own message and test it. Once compiled the dedicated control, the user can click the Send button to see the data outgoing and verify the answer.

Both outgoing and incoming data are shown alternating white and green colour background.

In order to see a specific session data exchange, clicking the "Clear" button, both outgoing and incoming data can be cleared.



CHAPTER 2
PROTOCOL

COMMANDS DESCRIPTION

Each message is a sequence of bytes starting with the ESCAPE and terminating with CR and LF bytes.

BYTE	DEC	HEX	BIN
ESCAPE	27	0x1B	00011011
LF	10	0x0A	00001010
CR	13	0x0D	00001101
ACK	6	0x06	00000110
NACK (non ACK)	21	0x15	00010101

Commands and answers are built according to the following scheme:

- Commands

0x1B	<msg length>		<command class> 1 byte	<command> 1 byte	<parameters> 0...Nbyte	<msg terminator>	
	Low Byte	High Byte				0x0D	0x0A

- Answers

0x1B	<msg length>		<ACK/NACK> 1 byte	<answer detail> 0...Nbyte	<msg terminator>	
	Low Byte	High Byte			0x0D	0x0A

<msg length> : represents the number of bytes in the message[1]

<msg terminator> : CR and LF bytes

Example:

Answer (Hex):	1B	08 00	15	30 30 30 31	0D 0A
Answer (UTF-8):				0001	

NOTE

The command message MUST have the correct <msg length>.
If the command is shorter than the <msg length> the server will NOT answer.
If the command is longer than the <msg length> the server will answer with an error code.

1. <msg length> : msg length <Low Byte> and <High byte> (2 Bytes) are included in the byte count
<msg length> : <msg terminator> bytes (2 Bytes) are not included in the byte count

GENERAL COMMANDS	
0 x F1 81	Get Version
0 x F1 82	Get Version (Verbose)
0 x F1 91	Get Laser Status
0 x F1 92	Get Laser Status (Verbose)
0 x F1 93	Get Command Error
0 x F1 94	Get Command Error (Verbose)
0 x F1 A1	Get System Date & Time
0 x F1 A2	Set System Date & Time

FILES AND I/O	
0 x F2 81	Get Documents List
0 x F2 82	Open Document From Device
0 x F2 83	Open Document From File System
0 x F2 84	Save Document
0 x F2 91	Set I/O Port
0 x F2 92	Get I/O Port
0 x F2 93	Get Green Spot Type
0 x F2 94	Set Green Spot Type
0 x F2 95	Get Green Spot Indicator Time
0 x F2 96	Set Green Spot Indicator Time

DATA HANDLING	
0 x F3 81	Get Global Counter List
0 x F3 82	Get Global String List
0 x F3 83	Get Global Counter Value
0 x F3 84	Set Global Counter Value
0 x F3 85	Get Global String Value
0 x F3 86	Set Global String Value
0 x F3 91	Enable/Disable Data Field
0 x F3 92	Set Data Field Value
0 x F3 93	Get Data Field Value
0 x F3 96	Set Imported Field Value
0 x F3 98	Get Objects ID’s
0 x F3 A1	Move Data Field
0 x F3 A2	Move And Rotate Document
0 x F3 A4	Get Document Parameters
0 x F3 A5	Set Document Parameters
0 x F3 A6	Move and Rotate Data Field

AXIS HANDLING	
0 x F5 81	Move Axis
0 x F5 82	Reset Axis
0 x F5 83	Is Axis In Home Position
0 x F5 84	Get Axis Range
0 x F5 85	Get Axis Position
0 x F5 86	Is Axis Enabled
0 x F5 87	Stop Axis
0 x F5 88	Check Axis Movement
0 x F5 89	Get Distance Sensor Status
0 x F5 90	Autofocus
0 x F5 91	Set Distance Sensor Reference
0 x F5 92	Is On Focus

LASER HANDLING	
0 x F5 E1	Start Laser Test
0 x F5 E2	Stop Laser Test
0 x F5 F1	Start Aiming
0 x F5 F2	Start Marking
0 x F5 FF	Stop System

MARVIS™ HANDLING	
0 x F6 80	Get Reader Result
0 x F6 81	Get Match Result
0 x F6 82	Get Overall Grade Result
0 x F6 83	Get Metric Grade Result
0 x F6 84	Set Verification
0 x F6 85	Get Verification
0 x F6 86	Set Grade Value
0 x F6 87	Set Metric Grade Value
0 x F6 88	Get Grade Value
0 x F6 89	Get Metric Grade Value
0 x F6 90	Enable MARVIS™
0 x F6 91	Get MARVIS™ Status

GENERAL COMMANDS

Get Version

Command: **0xF1 0x81**

This command returns Laser Engine and TCP Server Protocol version.

Message (Hex):	1B	<msg length> 05 00	F1	81	<msg terminator> 0D 0A		
Answer (Hex):	1B	<msg length> <LB> <HB>	06	<Laser Engine Version>	0A	<Protocol Version>	<msg terminator> 0D 0A

<msg length> : 2 Bytes (see Commands description)

<msg terminator> : 2 Bytes (see Commands description)

<Laser Engine Version> : Variable Length (UTF-8)

<Protocol Version> : Variable Length (UTF-8)

Example: Laser Engine version "6.2.2.13274" and TCP Protocol version "3.0"

Index:	1	2 3	4	5	6 7
Command (Hex):	1B	05 00	F1	81	0D 0A

Index:	1	2 3	4	5...16	17	18 19 20	21 22
Answer (Hex):	1B	14 00	06	36 2E 32 2E 32 2E 31 33 32 37 34 20	0A	33 2E 30	0D 0A
Answer (UTF-8):				6.2.2.13274		3.0	

Get Version (Verbose)

Command: **0xF1 0x82**

This command returns Laser Engine and TCP Server Protocol version written in a user-readable way.

Message (Hex):	1B	<msg length> 05 00	F1	82	<msg terminator> 0D 0A		
Answer (Hex):	1B	<msg length> <LB> <HB>	06	<Laser Engine Version>	0A	<Protocol Version>	<msg terminator> 0D 0A

<msg length> : 2 Bytes (see Commands description)

<msg terminator> : 2 Bytes (see Commands description)

<Laser Engine Version> : Variable Length (UTF-8)

<Protocol Version> : Variable Length (UTF-8)

Example: "Laser Engine version 6.2.2.13274 "; "TCP Protocol version 3.0"

Index:	1	2 3	4	5	6 7
Command (Hex):	1B	05 00	F1	82	0D 0A

Index:	1	2 3	4	5...37	38	39...62	63 64
Answer (Hex):	1B	3E 00	06	4C 61 73 65 72 20 45 6E 67 69 6E 65 20 76 65 72 73 69 6F 6E 20 36 2E 32 2E 32 2E 31 33 32 37 34 20	0A	54 43 50 20 50 72 6F 74 6F 63 6F 6C 20 76 65 72 73 69 6F 6E 20 33 2E 30	0D 0A
Answer (UTF-8):				Laser Engine version 6.2.2.13274		TCP Protocol version 3.0	

Get Laser Status

Command: **0xF1 0x91**

This command returns the current laser status.

Message (Hex):	1B	<msg length> 05 00	F1	91	<msg terminator> 0D 0A
Answer (Hex):	1B	<msg length> 05 00	06	<Laser Status>	<msg terminator> 0D 0A

<msg length> : 2 Bytes (see Commands description)

<msg terminator> : 2 Bytes (see Commands description)

<Laser Status> : 1 Byte (UTF-8)

<Laser Status>

HEX	UTF-8	DESCRIPTION
30	0	LASER OFF
31	1	LASER WARM UP
32	2	LASER WAIT FOR START
33	3	LASER STANDBY
34	4	LASER STANDBY SHUTTER CLOSED
35	5	LASER READY
36	6	LASER READY SHUTTER CLOSED
37	7	LASER EMISSION
38	8	LASER BUSY SHUTTER CLOSED
39	9	LASER WARNING
3A	:	LASER ERROR

Example: current laser status is LASER_READY

Index:	1	2 3	4	5	6 7
Command (Hex):	1B	05 00	F1	91	0D 0A

Index:	1	2 3	4	5	6 7
Answer (Hex):	1B	05 00	06	35	0D 0A
Answer (UTF-8):				5	

Get Laser Status (Verbose)

Command: **0xF1 0x92**

This command returns the current laser status written in a user-readable way.

Message (Hex):	1B	<msg length> 05 00	F1	92	<msg terminator> 0D 0A
Answer (Hex):	1B	<msg length> <LB> <HB>	06	<Laser Status>	<msg terminator> 0D 0A

<msg length> : 2 Bytes (see Commands description)

<msg terminator> : 2 Bytes (see Commands description)

<Laser Status> : Variable Length (UTF-8)

Example: current laser status is "Laser Ready"

Index:	1	2 3	4	5	6 7
Command (Hex):	1B	05 00	F1	92	0D 0A

Index:	1	2 3	4	5...15	16 17
Answer (Hex):	1B	0F 00	06	4C 61 73 65 72 20 52 65 61 64 79	0D 0A
Answer (UTF-8):				Laser Ready	

Get Command Error

Command: **0xF1 0x93**

This command returns the Error Code corresponding to the last command error returned by the TCP server.

Message (Hex):	1B	<msg length> 05 00	F1	93	<msg terminator> 0D 0A
Answer (Hex):	1B	<msg length> 08 00	06	<Command Error Code>	<msg terminator> 0D 0A

<msg length> : 2 Bytes (see "Commands description" on page 11)

<msg terminator> : 2 Bytes (see "Commands description" on page 11)

<Command Error Code> : 4 Byte (UTF-8)

<Command Error Code>[2]

VALUE (UTF-8)	DESCRIPTION
0001	Command not recognized
0002	Invalid date value
0003	File does not exist
0004	File opening error
0005	Invalid I/O port
0006	Global variable does not exist
0007	Global variable is not a counter
0008	Global variable is not a string
0009	Bad command
0010	Invalid field
0011	No document loaded
0012	No document saved
0013	Laser already stopped
0014	Command not allowed by device status
0015	Invalid Field Symbol Object ID
0016	Invalid Reader result
0017	Result not found
0018	Symbol not found
0019	Bad Grade Required Validation
0020	MARVIS is not enabled
0021	MARVIS License is not enabled
0022	Focal Distance Sensor Unavailable
0023	Green Spot Type cannot be set
0024	Focal Distance Sensor Focus Error
0025	Focal Distance Sensor Reference Invalid
0026	Focal Distance Sensor Out Of Range
0027	Focal Distance Sensor Connection Error
0028	Focal Distance Sensor Comunication Error
0029	Focal Distance Sensor Invalid Focus Search

Example: no command error occurred

Index:	1	2 3	4	5	6 7
Command (Hex):	1B	05 00	F1	93	0D 0A

2. If no error occurred <Command Error Code> is not sent

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

Example: last command error is COMMAND NOT RECOGNIZED

Index:	1	2 3	4	5	6 7
Command (Hex):	1B	05 00	F1	93	0D 0A

Index:	1	2 3	4	5...8	9 10
Answer (Hex):	1B	08 00	06	30 30 30 31	0D 0A
Answer (UTF-8):				0001	

Get Command Error (Verbose)

Command: **0xF1 0x94**

This command returns the error message corresponding to the last command error returned by the TCP server, written in an user-readable way. If no error occurred <Command Error Code> is not sent.

Message (Hex):	1B	<msg length> 05 00	F1	94	<msg terminator> 0D 0A
Answer (Hex):	1B	<msg length> <LB> <HB>	06	<Command Error Code>	<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 11)

<msg terminator> : 2 Bytes (see “Commands description” on page 11)

<Command Error Code> : Variable length (UTF-8)

Example: last command error is "Command not recognized"

Index:	1	2 3	4	5	6 7
Command (Hex):	1B	05 00	F1	94	0D 0A

Index:	1	2 3	4	5...27	28 29
Answer (Hex):	1B	1B 00	06	43 6f 6d 6d 61 6e 64 20 6e 6f 74 20 72 65 63 6f 67 6e 69 7a 65 64 2e	0D 0A
Answer (UTF-8):				Command not recognized	

Get System Date & Time

Command: **0xF1 0xA1**

This command returns system date and time.

Message (Hex):	1B	<msg length> 05 00	F1	A1	<msg terminator> 0D 0A
Answer (Hex):	1B	<msg length> 15 00	06	<System Time>	<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 11)

<msg terminator> : 2 Bytes (see “Commands description” on page 11)

<System Time> : 17 Bytes (UTF-8)

<System Time> string formatted as YYYYMMDDHHMMSSmmm

YYYY	4 byte	4-digit year
MM	2 byte	2-digit month
DD	2 byte	2-digit day of the month
HH	2 byte	2-digit hour of the day in 24 hours format
MM	2 byte	2-digit minutes
SS	2 byte	2-digit seconds
mmm	3 byte	3-digit millisecond

Example: system date is 2015/02/24, system time is 11h47m32s953ms

Index:	1	2 3	4	5	6 7
Command (Hex):	1B	05 00	F1	A1	0D 0A

Index:	1	2 3	4	5...21	22 23
Answer (Hex):	1B	15 00	06	32 30 31 35 30 32 32 34 31 31 34 37 33 32 39 35 33	0D 0A
Answer (UTF-8):				20150224114732953	

Set System Date & Time

Command: **0xF1 0xA2**

This command sets system date and time.

Message (Hex):	1B	<msg length> 13 00	F1	A2	<System Time>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A		
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 11)

<msg terminator> : 2 Bytes (see “Commands description” on page 11)

<System Time> : 14 Bytes (UTF-8)

<Error Code> : 4 Bytes (see Get Command Error)

<System Time> string formatted as YYYYMMDDHHMMSS

YYYY	4 byte	4-digit year
MM	2 byte	2-digit month
DD	2 byte	2-digit day of the month
HH	2 byte	2-digit hour of the day in 24 hours format
MM	2 byte	2-digit minutes
SS	2 byte	2-digit seconds

Example: set system date to 2015/02/24, set system time to 11h47m32s

Index:	1	2 3	4	5	6...19	19 20
Command (Hex):	1B	13 00	F1	A2	32 30 31 35 30 32 32 34 31 31 34 37 33 32	0D 0A
Command (UTF-8):					20150224114732	

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

FILES AND I/O

Get Documents List

Command: **0xF2 0x81**

This command returns a list of `.xlp` files stored in Lighter™ *Data\Docs\Layouts*.

Message (Hex):	1B	<msg length> 05 00	F2	81	<msg terminator> 0D 0A				
Answer (Hex):	1B	<msg length> <LB> <HB>	06	<File name>	0A	<File name>	0A	<...>	<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<File Name> : Variable Length (UTF-8)

<File Name> file name and extension

Example: "`001.xlp`" and "`00a.xlp`" are stored into Lighter™ *Data\Docs\Layouts*.

Index:	1	2 3	4	5	6 7
Command (Hex):	1B	05 00	F2	81	0D 0A

Index:	1	2 3	4	5...11	12	13...19	20 21
Answer (Hex):	1B	13 00	06	30 30 31 2E 78 6C 70	0A	30 30 61 2E 78 6C 70	0D 0A
Answer (UTF-8):				001.xlp		00a.xlp	

DATALOGIC

Open Document From Device

Command: **0xF2 0x82**

This command opens a `.xlp` file stored in Lighter™ *Data\Docs\Layouts*.

Message (Hex):	1B	<msg length> <LB> <HB>	F2	82	<File Name>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A		
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)
<msg terminator> : 2 Bytes (see “Commands description” on page 16)
<File Name> : Variable Length (UTF-8)
<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<File Name> name and extension of the `.xlp` file to be opened
Example: open "`CC.xlp`" file stored in Lighter™ *Data\Docs\Layouts*.

Index:	1	2 3	4	5	6...11	12 13
Command (Hex):	1B	0B 00	F2	82	43 43 2E 78 6C 70	0D 0A
Command (UTF-8):					CC.xlp	

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

Open Document From File System

Command: **0xF2 0x83**

This command opens a `.xlp` file stored in the specified path.

Message (Hex):	1B	<msg length> <LB> <HB>	F2	83	<File Name>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A		
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)
<msg terminator> : 2 Bytes (see “Commands description” on page 16)
<File Name> : Variable Length (UTF-8)
<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<File Name>: file path, name and extension of the `.xlp` file to be opened
Example: open "`CC.xlp`" file stored in *C:*

Index:	1	2 3	4	5	6...14	15 16
Command (Hex):	1B	0E 00	F2	83	43 3A 5C 43 43 2E 78 6C 70	0D 0A
Command (UTF-8):					C:\CC.xlp	

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

Save Document

Command: **0xF2 0x84**

This command saves the currently opened document.

Message (Hex):	1B	<msg length> 05 00	F2	84	<msg terminator> 0D 0A>	
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A		
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A>

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

Example: save the currently opened document

Index:	1	2 3	4	5	6 7
Command (Hex):	1B	05 00	F2	84	0D 0A

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

Set I/O Port

Command: **0xF2 0x91**

This command sets/resets the selected OUTPUT status.

Message (Hex):	1B	<msg length> 09 00	F2	91	<Port>	<LB>	<HB>	<Value>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A					
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>	<msg terminator> 0D 0A				

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Port> : 1 Byte

<LB> : 1 Byte

<HB> : 1 Byte

<Value> : 1 Byte (UTF-8)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<Port> default value is 0x00

<LB> bit mask representing the OUTPUT from 0 to 7

BIT0	BIT1	BIT2	BIT3	BIT4	BIT5	BIT6	BIT7
OUTPUT0	OUTPUT1	OUTPUT2	OUTPUT3	OUTPUT4	OUTPUT5	OUTPUT6	OUTPUT7

<HB> bit mask representing the OUTPUT from 8 to 15

BIT0	BIT1	BIT2	BIT3	BIT4	BIT5	BIT6	BIT7
OUTPUT8	OUTPUT9	OUTPUT10	OUTPUT11	OUTPUT12	OUTPUT13	OUTPUT14	OUTPUT15

<Value>

HEX	UTF-8	DESCRIPTION
30	0	Turn OFF the outputs specified in <LB>,<HB>
31	1	Turn ON the outputs specified in <LB>,<HB>

Example: Turn ON OUTPUT0, OUTPUT1 and OUTPUT8 and leave the other unchanged

Index:	1	2 3	4	5	6	7 8	9	15 16
Command (Hex):	1B	09 00	F2	91	00	03 01	31	0D 0A
Command (UTF-8):							1	

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

Example: Turn OFF OUTPUT9 and leave the other unchanged

Index:	1	2 3	4	5	6	7 8	9	10 11
Command (Hex):	1B	09 00	F2	91	00	00 02	30	0D 0A
Command (UTF-8):							0	

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

Get I/O Port

Command: **0xF2 0x92**

This command reads the INPUT status.

Message (Hex):	1B	<msg length> 06 00	F2	92	<Port>	<msg terminator> 0D 0A
Answer (Hex):	1B	<msg length> 06 00	06	<LB	<HB>	<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)
<msg terminator> : 2 Bytes (see “Commands description” on page 16)
<Port> : 1 Byte
<LB> : 1 Byte Bit mask representing the input from 0 to 7
<HB> : 1 Byte Bit mask representing the input from 8 to 15

<Port> default value is 0x00

<LB> bit mask representing the INPUT from 0 to 7[3]

BIT0	BIT1	BIT2	BIT3	BIT4	BIT5	BIT6	BIT7
INPUT0	INPUT1	INPUT2	INPUT3	INPUT4	INPUT5	INPUT6	INPUT7

<HB> bit mask representing the INPUT from 8 to 15

BIT0	BIT1	BIT2	BIT3	BIT4	BIT5	BIT6	BIT7
INPUT8	INPUT9	INPUT10	INPUT11	INPUT12	INPUT13	INPUT14	INPUT15

Example: INPUT8 and INPUT9 are in HIGH status while all the other INPUT are in LOW status

Index:	1	2 3	4	5	6	7 8
Command (Hex):	1B	06 00	F2	92	00	0D 0A

Index:	1	2 3	4	5	6	7 8
Command (Hex):	1B	06 00	06	00	03	0D 0A

Example: INPUT0 is in HIGH status while all the other INPUT are in LOW status

Index:	1	2 3	4	5	6	7 8
Command (Hex):	1B	06 00	F2	92	00	0D 0A

Index:	1	2 3	4	5	6	7 8
Command (Hex):	1B	06 00	06	01	00	0D 0A

3. HIGH value of the bit means that the correspondent INPUT status is HIGH; LOW value of the bit means that the correspondent INPUT status is LOW

Get Green Spot Type[4]

Command: **0xF2 0x93**

This command returns the Green Spot Type

Message (Hex):	1B	<msg length> 05 00	F2	93	<msg terminator> 0D 0A
Answer (Hex):	1B	<msg length> 05 00	06	<Green Spot Type>	<msg terminator> 0D 0A

<msg length> : 2 Bytes (see "Commands description" on page 16)

<msg terminator> : 2 Bytes (see "Commands description" on page 16)

<Green Spot Type> : 1 Byte representing the Green Spot Type

<Green Spot Type>

HEX	UTF-8	DESCRIPTION
30	0	OFF
31	1	SYSTEM READY TO MARK
32	2	MARKING CONFIRMATION
33	3	MARVIS VERIFICATION

Example: Read Green Spot Type when it is set to MARVIS™ Verification

Index:	1	2 3	4	5	6 7
Command (Hex):	1B	05 00	F2	93	0D 0A

Index:	1	2 3	4	5	6 7
Answer (Hex):	1B	05 00	06	33	0D 0A

4. Green Spot functionality is available only on Arex™ 400 family markers

Set Green Spot Type[5]

Command: **0xF2 0x94**

This command sets the Green Spot Type.

Message (Hex):	1B	<msg length> 06 00	F2	94	<Green Spot Type>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A		
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>	<msg terminator> 0D 0A	

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Green Spot Type> : 1 Byte

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<Green Spot Type>

HEX	UTF-8	DESCRIPTION
30	0	OFF
31	1	SYSTEM READY TO MARK
32	2	MARKING CONFIRMATION
33	3	MARVIS VERIFICATION

Example: Set Green Spot Type to 2

Index:	1	2 3	4	5	6	7 8
Command (Hex):	1B	06 00	F2	94	32	0D 0A

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

5. Green Spot functionality is available only on Arex™ 400 family markers

Get Green Spot Indicator Time[6]

Command: **0xF2 0x95**

This command returns the Green Spot Indicator Time.

Message (Hex):	1B	<msg length> 05 00	F2	95	<msg terminator> 0D 0A
Answer (Hex):	1B	<msg length> <LB> <HB>	06	<Green Spot Indicator Time>	<msg terminator> 0D 0A

<msg length> : 2 Bytes (see "Commands description" on page 16)

<msg terminator> : 2 Bytes (see "Commands description" on page 16)

<Green Spot Type> : up to 4 Bytes representing the Green Spot Indicator Time

<Green Spot Indicator Time>

3 or 4 bytes	A number between 100 and 5000, specified with each character coded in UTF 8

Example: Read Green Spot Indicator Time when it is 4500 ms

Index:	1	2 3	4	5	6 7
Command (Hex):	1B	05 00	F2	95	0D 0A

1	2 3	4	5...8	9 10
1B	08 00	06	34 35 30 30	0D 0A

6. Green Spot functionality is available only on Arex™ 400 family markers

Set Green Spot Indicator Time[7]

Command: **0xF2 0x96**

This command sets the Green Spot Indicator Time.

Message (Hex):	1B	<msg length> <LB> <HB>	F2	96	<Green Spot Indicator Time>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A		
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>	<msg terminator> 0D 0A	

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Green Spot Indicator Time> : up to 4 Bytes

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<Green Spot Indicator Time>

3 or 4 bytes	A number between 100 and 5000, specified with each character coded in UTF 8

Example: Set Green Spot Indicator Time to 2400 ms

Index:	1	2 3	4	5	6...9	10 11
Command (Hex):	1B	09 00	F2	96	32 34 30 30	0D 0A

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

7. Green Spot functionality is available only on Arex™ 400 family markers

DATA HANDLING

Get Global Counter List

Command: **0xF3 0x81**

This command returns a list of global counters stored in the system.

Message (Hex):	1B	<msg length> 05 00	F3	81	<msg terminator> 0D 0A				
Answer (Hex):	1B	<msg length> <LB> <HB>	06	<Field Name>	0A	<...>	0A	<...>	<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Field Name> : Variable Length (UTF-8)

<Field Name> global counter name with indications of type and base (in brackets)

Example: there are two global counters, "xx" and "counter1", stored in the system

Index:	1	2 3	4	5	6 7
Command (Hex):	1B	05 00	F3	81	0D 0A

Index:	1	2 3	4	5...11	12	13...25	26 27
Answer (Hex):	1B	19 00	06	78 78 28 62 31 30 29	0A	63 6f 75 6e 74 65 72 31 28 62 31 30 29	0D 0A
Answer (UTF-8):				xx(b10)		counter1(b10)	

Get Global String List

Command: **0xF3 0x82**

This command returns a list of global string stored in the system.

Message (Hex):	1B	<msg length> 05 00	F3	82	<msg terminator> 0D 0A				
Answer (Hex):	1B	<msg length> <LB> <HB>	06	<Field Name>	0A	<...>	0A	<...>	<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Field Name> : Variable Length (UTF-8)

<Field Name> global string name

Example: there are two global string, "AFTER_ENABLE" and "STEP", stored in the system

Index:	1	2 3	4	5	6 7
Command (Hex):	1B	05 00	F3	82	0D 0A

Index:	1	2 3	4	5...19	20	21...24	26 27
Answer (Hex):	1B	18 00	06	41 46 54 45 52 5f 45 4e 41 42 4c 45 5f 4d 53	0A	53 54 45 50	0D 0A
Answer (UTF-8):				AFTER_ENABLE		STEP	

Get Global Counter Value

Command: **0xF3 0x83**

This command returns the value of the specified global counter.

Message (Hex):	1B	<msg length> <LB> <HB>	F3	83	<Field Name>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> <LB> <HB>	06	<Value>		<msg terminator> 0D 0A
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Field Name> : Variable Length (UTF-8)

<Value> : Variable Length (UTF-8)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<Field Name> global counter name

<Value> global counter value (expressed in decimal base)

Example: the value of global counter "counter1" is 0

Index:	1	2 3	4	5	6...13	14 15
Command (Hex):	1B	0D 00	F3	83	63 6f 75 6e 74 65 72 31	0D 0A
Command (UTF-8):					counter1	

Index:	1	2 3	4	5	6 7
Answer (Hex):	1B	05 00	06	30	0D 0A
Answer (UTF-8):				0	

Set Global Counter Value

Command: **0xF3 0x84**

This command sets the value of the specified global counter.

Message (Hex):	1B	<msg length> <LB> <HB>	F3	84	<Field Name>	0A	<Value>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A				
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>	<msg terminator> 0D 0A			

<msg length> : 2 Bytes (see "Commands description" on page 16)

<msg terminator> : 2 Bytes (see "Commands description" on page 16)

<Field Name> : Variable Length (UTF-8)

<Value> : Variable Length (UTF-8)

<Error Code> : 4 Bytes (see "Get Command Error" on page 23)

<Field Name> global counter name

<Value> global counter value (expressed in decimal base)

Example: set the value of global counter "xx" to 125

Index:	1	2 3	4	5	6 7	8	9...11	12 13
Command (Hex):	1B	0B 00	F3	84	78 78	0A	31 32 35	0D 0A
Command (UTF-8):					xx		125	

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

Get Global String Value

Command: **0xF3 0x85**

This command returns the value of the specified global string.

Message (Hex):	1B	<msg length> <LB> <HB>	F3	85	<Field Name>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> <LB> <HB>	06	<Value>		<msg terminator> 0D 0A
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Field Name> : Variable Length (UTF-8)

<Value> : Variable Length (UTF-8)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<Field Name> global string name

<Value> global string value (not sent if global string is void)

Example: the value of global string "xx" is "ψÈ"

Index:	1	2 3	4	5	6 7	8 9
Command (Hex):	1B	07 00	F3	83	78 78	0D 0A
Command (UTF-8):					xx	

Index:	1	2 3	4	5 6	7 8
Answer (Hex):	1B	06 00	06	CF 88	0D 0A
Answer (UTF-8):				ψÈ	

Set Global String Value

Command: **0xF3 0x86**

This command sets the value of the specified global string present in the laser system.

Message (Hex):	1B	<msg length> <LB> <HB>	F3	86	<Field Name>	0A	<Value>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A				
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A		

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Field Name> : Variable Length (UTF-8)

<Value> : Variable Length (UTF-8)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<Field Name> global string name

<Value> global string value

Example: set the value of the global string "xx" to "ψ”

Index:	1	2 3	4	5	6 7	8	9...12	13 14
Command (Hex):	1B	0C 00	F3	86	78 78	0A	CF 88 C3 88	0D 0A
Command (UTF-8):					xx		ψ	

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

Enable/Disable Data Field

Command: **0xF3 0x91**

This command enables/disables the object with the specified object ID present in the currently open document.

Message (Hex):	1B	<msg length> <LB> <HB>	F3	91	<Field Name>	0A	<Status>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0x0D 0x0A				
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>	<msg terminator> 0D 0A			

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Field Name> : Variable Length (UTF-8)

<Status> : 1 Byte (UTF-8)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<Field Name> object's ID

<Status> enable/disable the object

HEX	UTF-8	DESCRIPTION
30	0	OBJECT DISABLED
31	1	OBJECT ENABLED

Example: disable the object with object's ID "1"

Index:	1	2 3	4	5	6	7	8	9 10
Command (Hex):	1B	08 00	F3	91	31	0A	30	0D 0A
Command (UTF-8):					1		0	

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

Set Data Field Value

Command: **0xF3 0x92**

This command sets the value of a TEXT or CODE type object with the specified object ID present in the currently open document.

Message (Hex):	1B	<msg length> <LB> <HB>	F3	92	<Field Name>	0A	<Value>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A				
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A		

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Field Name> : Variable Length (UTF-8)

<Value> : Variable Length (UTF-8)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<Field Name> object's ID

<Value> value to be assigned to the object

Example: sets the value of the TEXT object with object's ID "xx" to "ψæ"

Index:	1	2 3	4	5	6 7	8	9...12	13 14
Command (Hex):	1B	0C 00	F3	92	78 78	0A	CF 88 C3 A6	0D 0A
Command (UTF-8):					xx		ψæ	

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

Get Data Field Value

Command: **0xF3 0x93**

This command returns the value of a TEXT or CODE type object with the specified object ID present in the currently open document.

Message (Hex):	1B	<msg length> <LB> <HB>	F3	93	<Field Name>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> <LB> <HB>	06	<Value>		<msg terminator> 0D 0A
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Field Name> : Variable Length (UTF-8)

<Value> : Variable Length (UTF-8)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<Field Name> object's ID

<Value> value of the object

Example: the value of the TEXT object with object's ID "xx" is "ψ"

Index:	1	2 3	4	5	6 7	8 9
Command (Hex):	1B	07 00	F3	93	78 78	0D 0A
Command (UTF-8):					xx	

Index:	1	2 3	4	5 6	7 8
Answer (Hex):	1B	06 00	06	CF 88	0D 0A
Answer (UTF-8):				ψ	

Set Imported Field Value

Command: **0xF3 0x96**

This command sets file path and name of an IMPORTED object which has the specified object ID in the currently open document.

Message (Hex):	1B	<msg length> <LB> <HB>	F3	96	<Field Name>	0A	<Value>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A				
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A		

<msg length> : 2 Bytes (see “Commands description” on page 11)

<msg terminator> : 2 Bytes (see “Commands description” on page 11)

<Field Name> : Variable Length (UTF-8)

<Value> : Variable Length (UTF-8)

<Error Code> : 4 Bytes (see Get Command Error)

<Field Name> object's ID

<Value> file path and name of the imported object

Example: sets the file path and name of the IMPORTED object with object's ID "aa" to "C:\CC.xlp"

Index:	1	2 3	4	5	6 7	8	9...17	18 19
Command (Hex):	1B	11 00	F3	96	61 61	0A	43 3A 5C 43 43 2E 78 6C 70	0D 0A
Command (UTF-8):					aa		C:\CC.xlp	

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

Get Objects ID’s

Command: **0xF3 0x98**

This command returns the list of object's ID present in the currently open document.

Message (Hex):	1B	<msg length> 05 00	F3	98	<msg terminator> 0D 0A		
Answer OK (Hex):	1B	<msg length> <LB> <HB>	06	<Value>	0A	<Value>	<msg terminator> 0D 0A
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>			<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Field Name> : Variable Length (UTF-8)

<Value> : Variable Length (UTF-8)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<Value> object ID's

Example: the document contains objects with object's ID "1" and "2"

Index:	1	2 3	4	5	6 7
Command (Hex):	1B	05 00	F3	98	0D 0A

Index:	1	2 3	4	5	6	7	8 9
Answer (Hex):	1B	07 00	06	31	0A	32	0D 0A
Answer (UTF-8):				1		2	

Move Data Field

Command: **0xF3 0xA1**

This command moves the object with the specified object ID present in the currently open document, to the given position. The X,Y coordinates refer to the centre of the marking area. The object's positioning is done considering the object's origin.

Message (Hex):	1B	<msg length> <LB> <HB>	F3	A1	<Field Name>	0A	<X>, <Y>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A				
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>	<msg terminator> 0D 0A			

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Field Name> : Variable Length (UTF-8)

<X> : Variable Length (UTF-8)

<Y> : Variable Length (UTF-8)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<Field Name> : object's ID

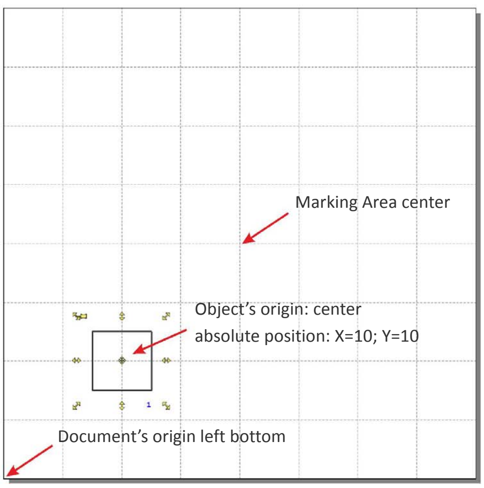
<X> : X position (mm or inch), decimal point "." for fractional part, positive/negative value

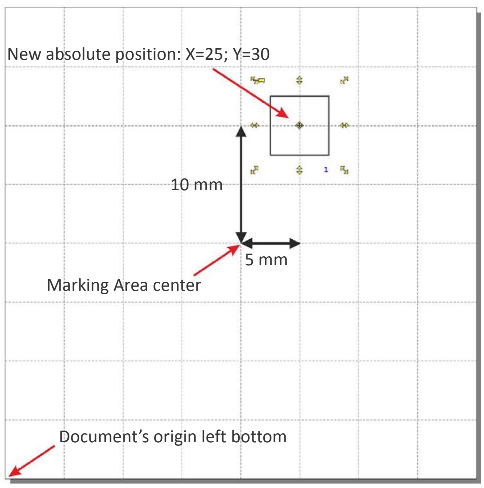
<Y> : Y position (mm or inch), decimal point "." for fractional part, positive negative value

Example: this command moves the object with object's ID "1" in position X=5, Y=10 referred to the centre of the marking area

Index:	1	2 3	4	5	6	7	8...19	20 21
Command (Hex):	1B	13 00	F3	A1	31	0A	35 2E 30 30 30 2C 31 30 2E 30 30 30	0D 0A
Command (UTF-8):					1		5.000,10.000	

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A





Move And Rotate Document

Command: **0xF3 0xA2**

This command moves the document's origin and its content of a given offset and then rotates it of the given angle. The rotation is done considering the new document's origin.

Message (Hex):	1B	<msg length> <LB> <HB>	F3	A2	<X>, <Y>, <Angle>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A		
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>	<msg terminator> 0D 0A	

<msg length> : 2 Bytes (see ‘Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<X> : Variable Length (UTF-8)

<Y> : Variable Length (UTF-8)

<Angle> : Variable Length (UTF-8)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<X>: X offset (mm or inch), decimal point "." for fractional part, positive/negative value

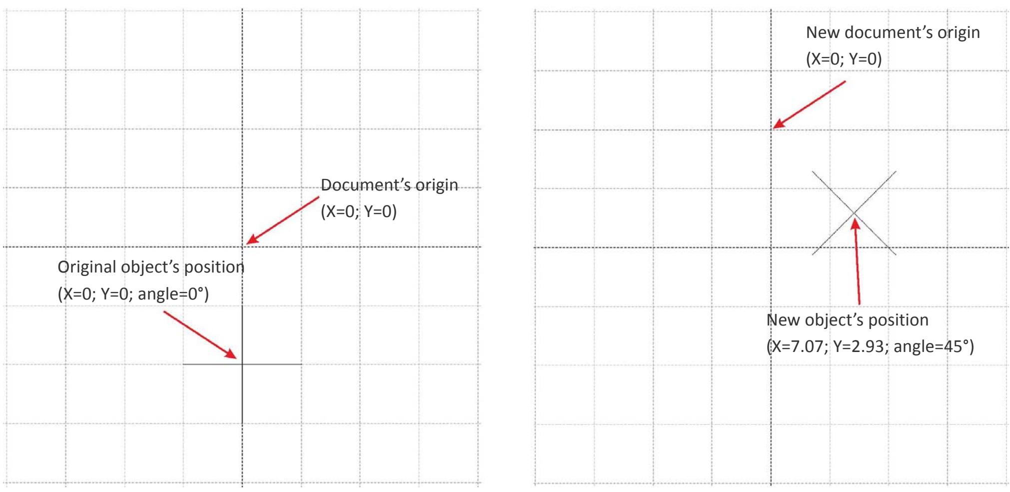
<Y> Y offset (mm or inch), decimal point "." for fractional part, positive negative value

<Angle> : angle (degree), decimal point "." for fractional part, positive/negative value

Example: this command moves the entire document and its content of 0mm in X direction and 10mm in Y direction and then rotate it of +45°. The rotation is done considering the new document's origin (X=0; Y=10)

Index:	1	2 3	4	5	6...12	13 14
Command (Hex):	1B	0C 00	F3	A2	30 2c 31 30 2c 34 35	0D 0A
Command (UTF-8):					0,10,45	

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A



DATALOGIC

Get Document Parameters

Command: **0xF3 0xA4**

This command returns power level, currently open frequency, scan speed, pulse profile (*) of the currently open document.

Message (Hex):	1B	<msg length> 05 00	F3	A4	<msg terminator> 0D 0A						
Answer OK (Hex):	1B	<msg length> <LB> <HB>	06	<Power>	0A	<Frequency>	0A	<Speed>	0A	<Pulse Profile>	<msg terminator> 0D 0A
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>	<msg terminator> 0D 0A						

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Power> : Variable Length (UTF-8)

<Frequency> : Variable Length (UTF-8)

<Speed> : Variable Length (UTF-8)

<Pulse Profile> : 1 Byte (UTF-8)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<Power> power level (%)

<Frequency> frequency (Hz)

<Speed> scan speed (mm/s or inch/s)

<Pulse Profile> pulse profile index[8]

HEX	UTF-8	DESCRIPTION
30	0	4 ns
31	1	8 ns
32	2	12 ns
33	3	30 ns
34	4	50 ns
35	5	100 ns
36	6	200 ns
37	7	250 ns

Example : laser parameters of the opened document are:

- 85% power level; 200000Hz frequency; 1000mm/s scan speed; 100ns pulse profile

Index:	1	2 3	4	5	6 7
Command (Hex):	1B	05 00	F3	A4	0D 0A

Index:	1	2 3	4	5 6	7	8...13	14	15...18	19	20	21 22
Answer (Hex):	1B	14 00	06	38 35	0A	32 30 30 30 30 30	0A	31 30 30 30	0A	35	0D 0A
Answer (UTF-8):				85		200000		1000		5	

8. meaningful for MOPA laser source only

Set Document Parameters

Command: **0xF3 0xA5**

This command sets the power level, frequency, scan speed, pulse profile (*) of the currently open document.

Message (Hex):	1B	<msg length> <LB> <HB>	F3	A5	<Power>	0A	<Frequency>	0A	<Speed>	0A	<Pulse Profile>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A								
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>	<msg terminator> 0D 0A							

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Power> : Variable Length (UTF-8)

<Frequency> : Variable Length (UTF-8)

<Speed> : Variable Length (UTF-8)

<Pulse Profile> : 1 Byte (UTF-8)

<Value> : Variable Length (UTF-8)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<Power> set the power level of the document (%)

<Frequency> set the frequency of the document (Hz)

<Speed> set the scan speed of the document (mm/s or inch/s)

<Pulse Profile> pulse profile index[9]

HEX	UTF-8	DESCRIPTION
30	0	4 ns
31	1	8 ns
32	2	12 ns
33	3	30 ns
34	4	50 ns
35	5	100 ns
36	6	200 ns
37	7	250 ns

Example using standard laser source: set the laser parameters of the opened document to:

- 86% power level; 20000Hz frequency; 5000mm/s scan speed

Index:	1	2 3	4	5	6 7	8	9...13	14	15...18	19 20
Command (Hex):	1B	12 00	F3	A5	38 36	0A	32 30 30 30 30	0A	35 30 30 30	0D 0A
Command (UTF-8):					86		20000		5000	

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

9. mandatory for MOPA laser source; not necessary for other laser sources

Example using for MOPA laser source: set the laser parameters of the opened document to:

- 86% power level; 20000Hz frequency; 5000mm/s scan speed; 100ns pulse profile

Index:	1	2 3	4	5	6 7	8	9...13	14	15...18	19	20	21 22
Command (Hex):	1B	14 00	F3	A5	38 36	0A	32 30 30 30 30	0A	35 30 30 30	0A	35	0D 0A
Command (UTF-8):					86		20000		5		5	

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

Move and Rotate Data Field

Command: **0xF3 0xA6**

This command moves and rotates the object with the specified object ID present in the currently open document, to the given position and rotates it of the given angle. The X,Y coordinates refer to the centre of the marking area. The object's positioning and rotation is done considering the object's origin.

Message (Hex):	1B	<msg length> <LB> <HB>	F3	A6	<Field Name>	0A	<X>	0A	<Y>	0A	<Angle>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A								
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>	<msg terminator> 0D 0A							

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Field Name> : Variable Length (UTF-8)

<X> : Variable Length (UTF-8)

<Y> : Variable Length (UTF-8)

<Angle> : Variable Length (UTF-8)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<Field Name> : object's ID

<X> : X position (mm or inch), decimal point "." for fractional part, positive/negative value

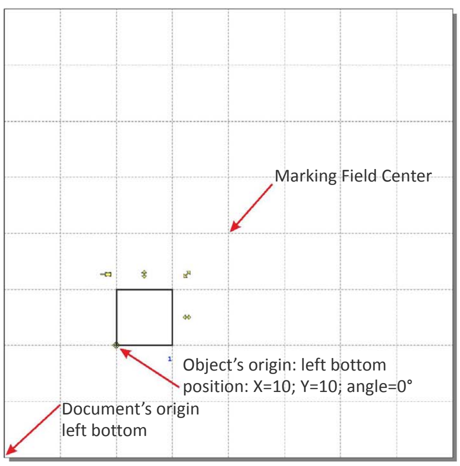
<Y> : Y position (mm or inch), decimal point "." for fractional part, positive negative value

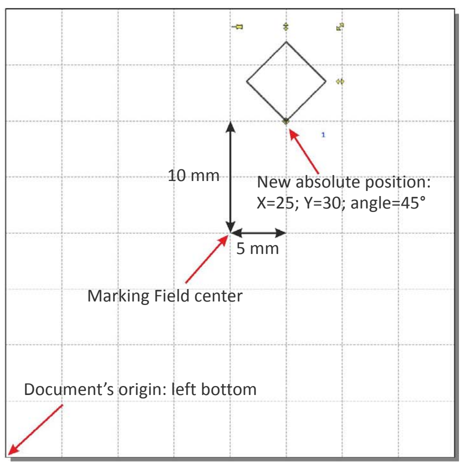
<Angle> : angle (degree), decimal point "." for fractional part, positive/negative value

Example: this command moves the object with object's ID "1", in position X=5, Y=10 referred to the centre of the marking area, and rotate it of 45° referred to the object's origin (left bottom)

Index:	1	2 3	4	5	6	7	8	9	10 11	12	13 14	15 16
Command (Hex):	1B	0E 00	F3	A6	31	0A	35	0A	31 30	0A	34 35	0D 0A
Command (UTF-8):					1		5		10		45	

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A





DATALOGIC

AXIS HANDLING

Move Axis

Command: **0xF5 0x81**

This command moves the specified axis in the required position.

Message (Hex):	1B	<msg length> <LB> <HB>	F5	81	<Axis>	0A	<Position>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A				
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A		

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Axis> : 1 Byte (UTF-8)

<Position> : Variable Length (UTF-8)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<Axis> axis to move

HEX	UTF-8	DESCRIPTION
30	0	X axis
31	1	Y axis
32	2	Z axis
33	3	R axis

<Position> : position to move the axis to, positive/negative, decimal point "." for fractional part.

NOTE

In order to check the end of the execution of the “Move Axis” command, use “Check Axis Movement” or “Get axis position” commands.

Example: move Z Axis in position 30

Index:	1	2 3	4	5	6	7	8...13	14 15
Command (Hex):	1B	0D 00	F5	81	32	0A	33 30 2e 30 30 30	0D 0A
Command (UTF-8):					2		30.000	

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

Reset Axis

Command: **0xF5 0x82**

This command moves the specified axis in home position (home sensor).

Message (Hex):	1B	<msg length> 06 00	F5	82	<Axis>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A		
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Axis> : 1 Byte (UTF-8)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<Axis> axis to reset

HEX	UTF-8	DESCRIPTION
30	0	X axis
31	1	Y axis
32	2	Z axis
33	3	R axis

NOTE

In order to check the end of the execution of the “Reset Axis” command, use “Check Axis Movement” or “Is Axis in Home Position” commands.

Example: move the Z axis in home position

Index:	1	2 3	4	5	6	7 8
Command (Hex):	1B	06 00	F5	82	32	0D 0A
Command (UTF-8):					2	

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

Is Axis In Home Position

Command: **0xF5 0x83**

This command checks if the specified axis is in home position (home sensor).

Message (Hex):	1B	<msg length> 06 00	F5	83	<Axis>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 05 00	06	<Value>		<msg terminator> 0D 0A
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Axis> : 1 Byte (UTF-8)

<Value> : 1 Byte (UTF-8)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<Axis>

HEX	UTF-8	DESCRIPTION
30	0	X axis
31	1	Y axis
32	2	Z axis
33	3	R axis

<Value>

HEX	UTF-8	DESCRIPTION
30	0	Axis not in home position
31	1	Axis in home position

Example: the Z axis is not in home position

Index:	1	2 3	4	5	6	7 8
Command (Hex):	1B	06 00	F5	83	32	0D 0A
Command (UTF-8):					2	

Index:	1	2 3	4	5	6 7
Answer (Hex):	1B	05 00	06	30	0D 0A
Answer (UTF-8):				0	

Example: the Z axis is in home position

Index:	1	2 3	4	5	6	7 8
Command (Hex):	1B	06 00	F5	83	32	0D 0A
Command (UTF-8):					2	

Index:	1	2 3	4	5	6 7
Answer (Hex):	1B	05 00	06	31	0D 0A
Answer (UTF-8):				1	

Get Axis Range

Command: **0xF5 0x84**

This command returns the minimum and maximum limits for the specified axis.

Message (Hex):	1B	<msg length> 06 00	F5	84	<Axis>		<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> <LB> <HB>	06	<Min>	0A	<Max>	<msg terminator> 0D 0A
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>			<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Axis> : 1 Byte (UTF-8)

<Min> : Variable Length (UTF-8)

<Max> : Variable Length (UTF-8)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<Axis>

HEX	UTF-8	DESCRIPTION
30	0	X axis
31	1	Y axis
32	2	Z axis
33	3	R axis

<Min> : the minimum allowed axis limit, positive/negative value, decimal point "." for fractional part

<Max> : the maximum allowed axis limit, positive/negative value, decimal point "." for fractional part

Example: the Z axis limits are between 0 and 330

Index:	1	2 3	4	5	6	7 8
Command (Hex):	1B	06 00	F5	84	32	0D 0A
Command (UTF-8):					2	

Index:	1	2 3	4	5	6	7...9	10 11
Answer (Hex):	1B	09 00	06	30	0A	33 33 30	0D 0A
Answer (UTF-8):				0		330	

Example: the Z axis limits are between -100 and 0

Index:	1	2 3	4	5	6	7 8
Command (Hex):	1B	06 00	F5	84	32	0D 0A
Command (UTF-8):					2	

Index:	1	2 3	4	5...8	9	10	11 12
Answer (Hex):	1B	0A 00	06	2d 31 30 30	0A	30	0D 0A
Answer (UTF-8):				-100		0	

DATALOGIC

Get Axis Position

Command: **0xF5 0x85**

This command returns the current position of the specified axis.

Message (Hex):	1B	<msg length> 06 00	F5	85	<Axis>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> <LB> <HB>	06	<Position>		<msg terminator> 0D 0A
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Axis> : 1 Byte (UTF-8)

<Position> : Variable Length (UTF-8)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<Axis>

HEX	UTF-8	DESCRIPTION
30	0	X axis
31	1	Y axis
32	2	Z axis
33	3	R axis

<Position> : current axis position, positive/negative, decimal point "." for fractional part

Example: the Z axis is currently in position 300

Index:	1	2 3	4	5	6	7 8
Command (Hex):	1B	06 00	F5	85	32	0D 0A
Command (UTF-8):					2	

Index:	1	2 3	4	5...7	8 9
Answer (Hex):	1B	07 00	06	33 30 30	0D 0A
Answer (UTF-8):				300	

Is Axis Enabled

Command: **0xF5 0x86**

This command checks if the specified axis is enabled.

Message (Hex):	1B	<msg length> 06 00	F5	86	<Axis>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 05 00	06	<Value>		<msg terminator> 0D 0A
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A

<msg length> : 2 Bytes (see "Commands description" on page 16)

<msg terminator> : 2 Bytes (see "Commands description" on page 16)

<Axis> : 1 Byte (UTF-8)

<Value> : 1 Byte (UTF-8)

<Error Code> : 4 Bytes (see "Get Command Error" on page 23)

<Axis>

HEX	UTF-8	DESCRIPTION
30	0	X axis
31	1	Y axis
32	2	Z axis
33	3	R axis

<Value>

HEX	UTF-8	DESCRIPTION
30	0	Axis not enabled
31	1	Axis enabled

Example: the Z axis is enabled

Index:	1	2 3	4	5	6	7 8
Command (Hex):	1B	06 00	F5	86	32	0D 0A
Command (UTF-8):					2	

Index:	1	2 3	4	5	6 7
Answer (Hex):	1B	05 00	06	31	0D 0A
Answer (UTF-8):				1	

Example: the Z axis is disabled

Index:	1	2 3	4	5	6	7 8
Command (Hex):	1B	06 00	F5	86	32	0D 0A
Command (UTF-8):					2	

Index:	1	2 3	4	5	6 7
Answer (Hex):	1B	05 00	06	30	0D 0A
Answer (UTF-8):				0	

Stop Axis

Command: **0xF5 0x87**

This command stops the specified axis's movement.

Message (Hex):	1B	<msg length> 06 00	F5	87	<Axis>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A		
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Axis> : 1 Byte (UTF-8)

<Axis>

HEX	UTF-8	DESCRIPTION
30	0	X axis
31	1	Y axis
32	2	Z axis
33	3	R axis

NOTE

In order to check the end of the execution of the “Stop Axis” command, use “Check Axis Movement” command.

It is recommended to perform a “Reset Axis” command after the “Stop Axis” command.

Example: stop Z axis

Index:	1	2 3	4	5	6	7 8
Command (Hex):	1B	06 00	F5	87	32	0D 0A
Command (UTF-8):					2	

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A
Answer (UTF-8):				

Example: stop X axis

Index:	1	2 3	4	5	6	7 8
Command (Hex):	1B	06 00	F5	87	30	0D 0A
Command (UTF-8):					0	

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A
Answer (UTF-8):				

Check Axis Movement

Command: **0xF5 0x88**

This command checks if the specified axis is still moving.

Message (Hex):	1B	<msg length> 06 00	F5	88	<Axis>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 05 00	06	<Value>		<msg terminator> 0D 0A
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A

<msg length> : 2 Bytes (see "Commands description" on page 16)

<msg terminator> : 2 Bytes (see "Commands description" on page 16)

<Axis> : 1 Byte (UTF-8)

<Value> : 1 Byte (UTF-8)

<Error Code> : 4 Bytes (see "Get Command Error" on page 23)

<Axis>

HEX	UTF-8	DESCRIPTION
30	0	X axis
31	1	Y axis
32	2	Z axis
33	3	R axis

<Value>

HEX	UTF-8	DESCRIPTION
30	0	Axis is not moving
31	1	Axis is moving

Example: the Z axis is moving

Index:	1	2 3	4	5	6	7 8
Command (Hex):	1B	06 00	F5	88	32	0D 0A
Command (UTF-8):					2	

Index:	1	2 3	4	5	6 7
Answer (Hex):	1B	05 00	06	31	0D 0A
Answer (UTF-8):				1	

Example: the Z axis is not moving

Index:	1	2 3	4	5	6	7 8
Command (Hex):	1B	06 00	F5	88	32	0D 0A
Command (UTF-8):					2	

Index:	1	2 3	4	5	6 7
Answer (Hex):	1B	05 00	06	30	0D 0A
Answer (UTF-8):				0	

Get Distance Sensor Status[10]

Command: **0xF5 0x89**

This command gets the sensor status.

<table>
<tr><td>Message (Hex):</td><td>1B</td><td><msg length>
05 00</td><td>F5</td><td>89</td><td><msg terminator>
0D 0A</td><td></td></tr>
<tr><td>Answer OK (Hex):</td><td>1B</td><td><msg length>
04 00</td><td>06</td><td colspan="2"><msg terminator>
0D 0A</td><td></td></tr>
<tr><td>Answer KO (Hex):</td><td>1B</td><td><msg length>
08 00</td><td>15</td><td colspan="2"><Error Code></td><td><msg terminator>
0D 0A</td></tr>
</table>

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

10. Distance Sensor functionality is available only on Arex™ 400 family markers

Autofocus[11]

Command: **0xF5 0x90**

This command starts/stops the autofocus process. The autofocus is an asynchronous command and the result is returned at the end of process.

Message (Hex):	1B	<msg length> 06 00	F5	90	<Value>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A		
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A

<msg length> : 2 Bytes (see "Commands description" on page 16)

<msg terminator> : 2 Bytes (see "Commands description" on page 16)

<Value> : 1 Byte (UTF-8)

<Error Code> : 4 Bytes (see "Get Command Error" on page 23)

<Value>

HEX	UTF-8	DESCRIPTION
30	0	Stop autofocus
31	1	Start autofocus

NOTE

In order to check the end of the execution of the "Autofocus" command, use "Check Axis Movement" or "Is On focus" commands.

11. Distance Sensor functionality is available only on Arex™ 400 family markers

Set Distance Sensor Reference[12]

Command: **0xF5 0x91**

This command sets the axis current position as distance reference value.

Message (Hex):	1B	<msg length> 05 00	F5	91	<msg terminator> 0D 0A	
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A		
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

12. Distance Sensor functionality is available only on Arex™ 400 family markers

Is On Focus[13]

Command: **0xF5 0x92**

This command check if the axis is on focus position.

Message (Hex):	1B	<msg length> 05 00	F5	92	<msg terminator> 0D 0A
Answer (Hex):	1B	<msg length> 05 00	06	<Value>	<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Value> : 1 Byte

<Value>

HEX	UTF-8	DESCRIPTION
30	0	Axis not in focus position
31	1	Axis in focus position

13. Distance Sensor functionality is available only on Arex™ 400 family markers

LASER HANDLING

Start Laser Test

Command: **0xF5 0xE1**

This command starts the laser test using the specified parameters.

<table>
<tr><td>Message (Hex):</td><td>1B</td><td><msg length>
<LB> <HB></td><td>F5</td><td>E1</td><td><Tipe>,<Size>,<Power>,<Frequency>,<Pulse Profile></td><td><msg terminator>
0D 0A</td></tr>
<tr><td>Answer OK (Hex):</td><td>1B</td><td><msg length>
04 00</td><td>06</td><td colspan="2"><msg terminator>
0D 0A</td><td></td></tr>
<tr><td>Answer KO (Hex):</td><td>1B</td><td><msg length>
08 00</td><td>15</td><td colspan="2"><Error Code></td><td><msg terminator>
0D 0A</td></tr>
</table>

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Type> : 1 Byte (UTF-8)

<Size> : Variable Length (UTF-8)

<Power> : Variable Length (UTF-8)

<Frequency> : Variable Length (UTF-8)

<Pulse Profile> : 1 Byte (UTF-8)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

<Type>

HEX	UTF-8	DESCRIPTION
30	0	Line
31	1	Square
32	2	Circle
33	3	Dot

<Size> : size expressed in mm or inch, decimal point "." for fractional part

<Power> : power level (%)

<Frequency> : frequency (Hz)

<Pulse Profile> pulse profile index[14]

HEX	UTF-8	DESCRIPTION
30	0	4 ns
31	1	8 ns
32	2	12 ns
33	3	30 ns
34	4	50 ns
35	5	100 ns
36	6	200 ns
37	7	250 ns

14. mandatory for MOPA laser source; not necessary for other laser sources

Example: start laser test using the following parameters:

Line; 5mm size; 85% power level; 20000Hz frequency; 100ns Pulse Profile[15]

Index:	1	2 3	4	5	6...19	20 21
Command (Hex):	1B	13 00	F5	E1	30 2c 35 2c 38 35 2c 32 30 30 30 30 2c 35	0D 0A
Command (UTF-8):					0,5,85,20000,5	

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

15. mandatory for MOPA laser source; not necessary for other laser sources

Stop Laser Test

Command: **0xF5 0xE2**

This command stops the laser test.

Message (Hex):	1B	<msg length> 05 00	F5	E2	<msg terminator> 0D 0A	
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A		
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

Example:

Index:	1	2 3	4	5	6 7
Command (Hex):	1B	05 00	F5	E2	0D 0A

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

Start Aiming

Command: **0xF5 0xF1**

This command displays the limits of the currently open document.

Message (Hex):	1B	<msg length> 05 00	F5	F1	<msg terminator> 0D 0A	
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A		
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

Example:

Index:	1	2 3	4	5	6 7
Command (Hex):	1B	05 00	F5	F1	0D 0A

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

Start Marking

Command: **0xF5 0xF2**

This command starts to mark the currently open document.

Message (Hex):	1B	<msg length> 05 00	F5	F2	<msg terminator> 0D 0A	
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A		
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

Example:

Index:	1	2 3	4	5	6 7
Command (Hex):	1B	05 00	F5	F2	0D 0A

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

Stop System

Command: **0xF5 0xFF**

This command stops the marking process and/or the axis movement.

Message (Hex):	1B	<msg length> 05 00	F5	FF	<msg terminator> 0D 0A	
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A		
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

Example:

Index:	1	2 3	4	5	6 7
Command (Hex):	1B	05 00	F5	FF	0D 0A

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

MARVIS™ HANDLING

Get Reader Result

Command: **0xF6 0x80**

This command returns the symbol, match and grade results of the last MARVIS™ acquisition.

Message (Hex):	1B	<msg length> 05 00	F6	80	<msg terminator> 0D 0A					
Answer OK (Hex):	1B	<msg length> 09 00	06	<Symbol Read>	0A	<Match Results>	0A	<Grade Results>	<msg terminator> 0D 0A
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>	<msg terminator> 0D 0A				

<msg length> : 2 Bytes (see "Commands description" on page 16)

<msg terminator> : 2 Bytes (see "Commands description" on page 16)

<Symbol Read> : 1 Byte (UTF-8)

HEX	UTF-8	DESCRIPTION
30	0	Negative result
31	1	Positive result

<Match Results> : 1 Byte (UTF-8)

HEX	UTF-8	DESCRIPTION
30	0	Negative result
31	1	Positive result

<Grade Result> : 1 Byte (UTF-8)

HEX	UTF-8	DESCRIPTION
30	0	Negative result
31	1	Positive result

<Error Code> : 4 Bytes (see "Get Command Error" on page 23)

Example: MARVIS™ verification which succeeds in term of symbols read and Match result but returned grade does not satisfy the overall threshold.

Index:	1	2 3	4	5	6 7
Command (Hex):	1B	05 00	F6	80	0D 0A

Index:	1	2 3	4	5	6	7	8	9	10 11
Answer (Hex):	1B	09 00	06	31	0A	31	0A	30	0D 0A
Answer (UTF-8):				1		1		0	

Get Match Result

Command: **0xF6 0x81**

This command gets the last match result and the content for the specified object.

Message (Hex):	1B	<msg length>	F6	81	<Field Name>		<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length>	06	<Match Results>	0A	<Received Text>	<msg terminator> 0D 0A
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>			<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Field Name> : Object ID - Variable Length (UTF-8)

<Match Results> : 1 Byte (UTF-8)

HEX	UTF-8	DESCRIPTION
30	0	Negative result
31	1	Positive result

<Received Text> : Variable Length (UTF-8)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

Example: positive match result for marked object ID = "xx" and the MARVIS™ process has returned text "CC".

Index:	1	2 3	4	5	6 7	8 9
Command (Hex):	1B	07 00	F6	81	78 78	0D 0A

Index:	1	2 3	4	5	6	7 8	10 11
Answer (Hex):	1B	08 00	06	31	0A	43 43	0D 0A
Answer (UTF-8):				1		CC	

Get Overall Grade Result

Command: **0xF6 0x82**

This command gets the grade result and value of the last MARVIS™ acquisition for the specified object.

Message (Hex):	1B	<msg length>	F6	82	<Field Name>		<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length>	06	<Grade Results>	0A	<Grade Value>	<msg terminator> 0D 0A
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>			<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Field Name> : Object ID - Variable Length (UTF-8)

<Grade Results> : 1 Byte (UTF-8)

HEX	UTF-8	DESCRIPTION
30	0	Negative result
31	1	Positive result

<Grade Value> : 1 Byte (UTF-8)

HEX	UTF-8	DESCRIPTION
30	0	Grade A
31	1	Grade B
32	2	Grade C
33	3	Grade D
34	4	Grade F
35	5	NA (Not Available)

Example : Example: negative overall grade result for marked object ID = "xx" and the MARVIS™ process has returned grade value "F".

Index:	1	2 3	4	5	6 7	8 9
Command (Hex):	1B	07 00	F6	82	78 78	0D 0A

Index:	1	2 3	4	5	6	7	8 9
Answer (Hex):	1B	07 00	06	30	0A	34	0D 0A
Answer (UTF-8):				0		4	

Get Metric Grade Result

Command: **0xF6 0x83**

This command gets the grade result and value of the last MARVIS™ acquisition for the specified Metric of the object.

<table>
<tr><td>Message (Hex):</td><td>1B</td><td><msg length></td><td>F6</td><td>83</td><td colspan="3"><Field Name></td><td>0A</td><td><Metric ID></td><td><msg terminator>
0D 0A</td></tr>
<tr><td>Answer OK (Hex):</td><td>1B</td><td><msg length></td><td>06</td><td colspan="2"><Grade Results></td><td>0A</td><td><Metric Result></td><td>0A</td><td><Grade Value></td><td><msg terminator>
0D 0A</td></tr>
<tr><td>Answer KO (Hex):</td><td>1B</td><td><msg length>
08 00</td><td>15</td><td colspan="6"><Error Code></td><td><msg terminator>
0D 0A</td></tr>
</table>

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Field Name> : Object ID - Variable Length (UTF-8)

<Metric ID> : 1 or 2 Bytes (HEX)

HEX	UTF-8	DESCRIPTION
30	0	OVERALL
31	1	DECODE
32	2	CONTRAST
33	3	MODULATION
34	4	DECODABILITY
35	5	MINIMUMEDGECONTRAST
36	6	AXIALNONUNIFORMITY
37	7	UNUSEDERRORCORRECTION
38	8	PRINTGROWTH
39	9	MINIMUMREFLECTANCE
31 30	10	DEFECTS
31 31	11	FIXEDPATTERNDAMAGE
31 32	12	GRIDNONUNIFORMITY

<Grade Results> : 1 Byte (UTF-8)

HEX	UTF-8	DESCRIPTION
30	0	Negative result
31	1	Positive result

<Metric Results> : 1 Byte (UTF-8)

HEX	UTF-8	DESCRIPTION
30	0	Negative result
31	1	Positive result

<Grade Value> : 1 Byte (HEX)

HEX	UTF-8	DESCRIPTION
30	0	Grade A
31	1	Grade B
32	2	Grade C
33	3	Grade D
34	4	Grade F
35	5	NA (Not Available)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

Set Verification

Command: **0xF6 0x84**

This command enables/disables MARVIS™ verification for the specified object, present in the currently open document.

Message (Hex):	1B	<msg length>	F6	84	<Field Name>	0A	<Enable/Disable>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A				
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>	<msg terminator> 0D 0A			

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Field Name> : Object ID - Variable Length (UTF-8)

<Enable/Disable> : 1 Byte (UTF-8)

HEX	UTF-8	DESCRIPTION
30	0	Disable
31	1	Enable

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

Get Verification

Command: **0xF6 0x85**

This command gets the MARVIS™ verification status for the specified object.

Message (Hex):	1B	<msg length>	F6	85	<Field Name>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length>	06	<Enabled/Disabled>		<msg terminator> 0D 0A
Answer KO (Hex):	1B	<msg length>	15	<Error Code>		<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Field Name> : Object ID - Variable Length (UTF-8)

<Enable/Disable> : 1 Byte (UTF-8)

HEX	UTF-8	DESCRIPTION
30	0	Disabled
31	1	Enabled

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

Set Grade Value

Command: **0xF6 0x86**

This command sets the grade threshold value for the specified object, present in the currently open document.

Message (Hex):	1B	<msg length>	F6	86	<Field Name>	0A	<Grade Value>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A				
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A		

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Field Name> : Object ID - Variable Length (UTF-8)

<Grade Value> : 1 Byte (UTF-8)

HEX	UTF-8	DESCRIPTION
30	0	Overall Grade A
31	1	Overall Grade B
32	2	Overall Grade C
33	3	Overall Grade D
34	4	Overall Grade F
35	5	CUSTOM (Loads the last metric values which have been saved)

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

Set Metric Grade Value

Command: **0xF6 0x87**

This command sets the grade threshold value for the specified Metric of the object.

Message (Hex):	1B	<msg length>	F6	87	<Field Name>	0A	<Metric ID>	0A	<Grade Value>	<msg terminator> 0D 0A

Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A

Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>	<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Field Name> : Object ID - Variable Length (UTF-8)

<Metric ID> : 1 or 2 Bytes (HEX)

HEX	UTF-8	DESCRIPTION
30	0	OVERALL
31	1	DECODE
32	2	CONTRAST
33	3	MODULATION
34	4	DECODABILITY
35	5	MINIMUMEDGECONTRAST
36	6	AXIALNONUNIFORMITY
37	7	UNUSEDERRORCORRECTION
38	8	PRINTGROWTH
39	9	MINIMUMREFLECTANCE
31 30	10	DEFECTS
31 31	11	FIXEDPATTERNDAMAGE
31 32	12	GRIDNONUNIFORMITY

<Grade Value> : 1 Byte (HEX)

HEX	UTF-8	DESCRIPTION
30	0	Grade A
31	1	Grade B
32	2	Grade C
33	3	Grade D
34	4	Grade F

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

Example : setting to "A" the MODULATION metric threshold of object ID "xx"

Index:	1	2 3	4	5	6 7	8	9	10	11	12 13
Command (Hex):	1B	0B 00	F6	87	78 78	0A	33	0A	30	0D 0A

Index:	1	2 3	4	5 6
Answer (Hex):	1B	04 00	06	0D 0A

DATALOGIC

Get Grade Value

Command: **0xF6 0x88**

This command gets the grade threshold value of the specified object.

Message (Hex):	1B	<msg length>	F6	88	<Field Name>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 05 00	06	<Grade Value>		<msg terminator> 0D 0A
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Field Name> : Object ID - Variable Length (UTF-8)

<Grade Value> : 1 Byte (HEX)

HEX	UTF-8	DESCRIPTION
30	0	Overall Grade A
31	1	Overall Grade B
32	2	Overall Grade C
33	3	Overall Grade D
34	4	Overall Grade F
35	5	CUSTOM

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

Get Metric Grade Value

Command: **0xF6 0x89**

This command gets the grade threshold value for the specified Metric of the object.

Message (Hex):	1B	<msg length>	F6	89	<Field Name>	0A	<Metric ID>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length>	06	<Metric ID>	0A	<Grade Value>		
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>	<msg terminator> 0D 0A			

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Field Name> : Object ID - Variable Length (UTF-8)

<Metric ID> : 1 or 2 Bytes (HEX)

HEX	UTF-8	DESCRIPTION
30	0	OVERALL
31	1	DECODE
32	2	CONTRAST
33	3	MODULATION
34	4	DECODABILITY
35	5	MINIMUMEDGECONTRAST
36	6	AXIALNONUNIFORMITY
37	7	UNUSEDERRORCORRECTION
38	8	PRINTGROWTH
39	9	MINIMUMREFLECTANCE
31 30	10	DEFECTS
31 31	11	FIXEDPATTERNDAMAGE
31 32	12	GRIDNONUNIFORMITY

<Grade Value> : 1 Byte (HEX)

HEX	UTF-8	DESCRIPTION
30	0	Grade A
31	1	Grade B
32	2	Grade C
33	3	Grade D
34	4	Grade F

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

Example: returns the FIXEDPATTERNDAMAGE metric threshold "C" of object ID "xx"

Index:	1	2 3	4	5	6 7	8	9 10	11 12
Command (Hex):	1B	0A 00	F6	89	78 78	0A	31 31	0D 0A

Index:	1	2 3	4	5 6	7	8	9 10
Command (Hex):	1B	08 00	06	31 31	0A	32	0D 0A

Enable MARVIS™

Command: **0xF6 0x90**

This command enables/disables the MARVIS™ functionality.

Message (Hex):	1B	<msg length> 06 00	F6	90	<Enabled/Disabled>	<msg terminator> 0D 0A
Answer OK (Hex):	1B	<msg length> 04 00	06	<msg terminator> 0D 0A		
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>		<msg terminator> 0D 0A

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Field Name> : Object ID - Variable Length (UTF-8)

<Enable/Disable> : 1 Byte (UTF-8)

HEX	UTF-8	DESCRIPTION
30	0	Disable
31	1	Enable

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

Get MARVIS™ Status

Command: 0xF6 0x91

This command gets the MARVIS™ enabled/disabled status and checks the MARVIS™ license status.

Message (Hex):	1B	<msg length> 05 00	F6	91	<msg terminator> 0D 0A		
Answer OK (Hex):	1B	<msg length> 07 00	06	<Marvis enabled/ disabled>	0A	<License enabled/ disabled>	<msg terminator> 0D 0A
Answer KO (Hex):	1B	<msg length> 08 00	15	<Error Code>	<msg terminator> 0D 0A		

<msg length> : 2 Bytes (see “Commands description” on page 16)

<msg terminator> : 2 Bytes (see “Commands description” on page 16)

<Marvis Enabled/Disabled> : 1 Byte (UTF-8)

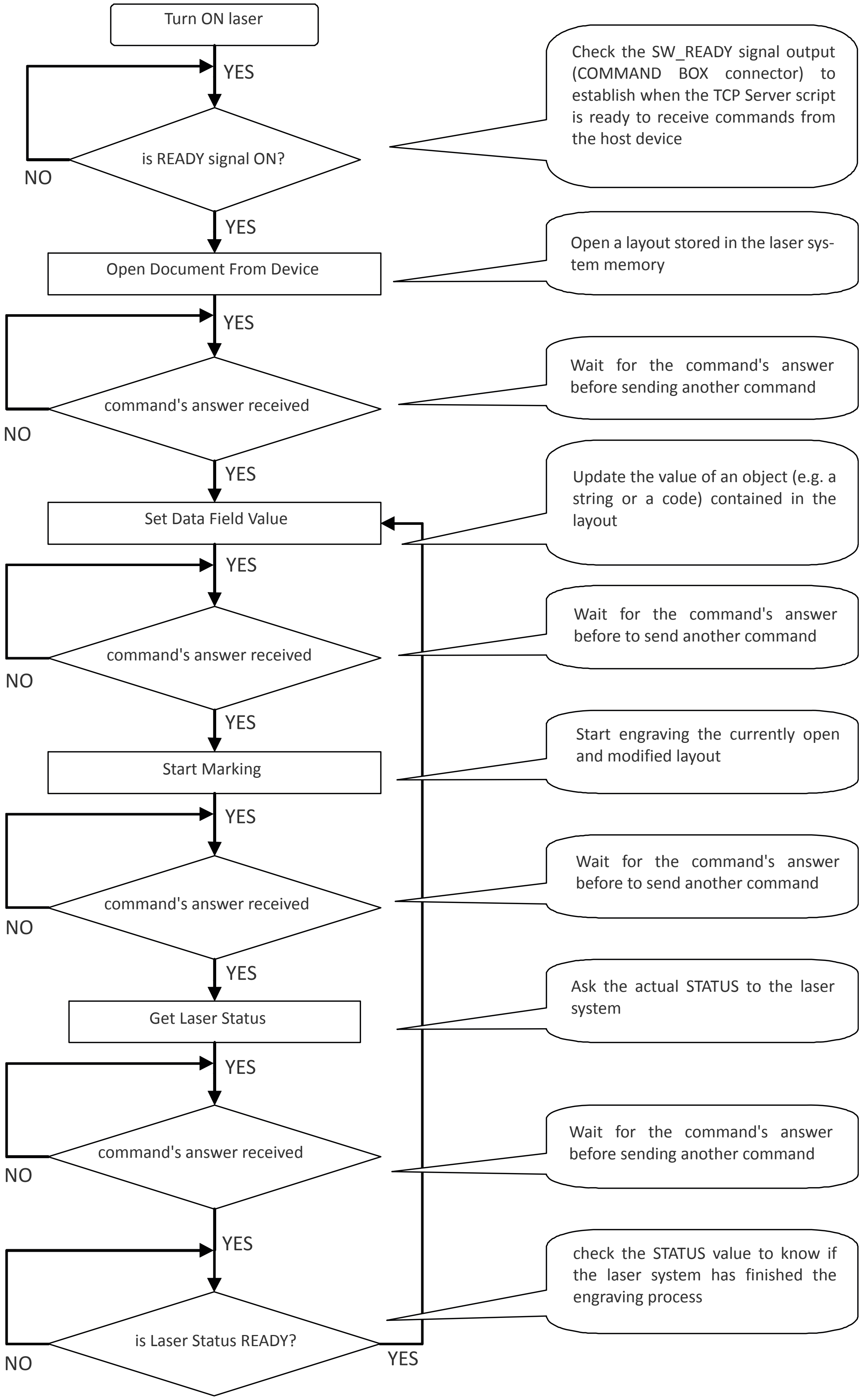
HEX	UTF-8	DESCRIPTION
30	0	Disabled
31	1	Enabled

<License Enabled/Disabled> : 1 Byte (UTF-8)

HEX	UTF-8	DESCRIPTION
30	0	Disabled
31	1	Enabled

<Error Code> : 4 Bytes (see “Get Command Error” on page 23)

EXAMPLE


Turn ON laser
YES
is READY signal ON?
NO
YES
Open Document From Device
YES
command's answer received
NO
YES
Set Data Field Value
YES
command's answer received
NO
YES
Start Marking
YES
command's answer received
NO
YES
Get Laser Status
YES
command's answer received
NO
YES
is Laser Status READY?
NO
YES
Check the SW_READY signal output (COMMAND BOX connector) to establish when the TCP Server script is ready to receive commands from the host device
Open a layout stored in the laser system memory
Wait for the command's answer before sending another command
Update the value of an object (e.g. a string or a code) contained in the layout
Wait for the command's answer before to send another command
Start engraving the currently open and modified layout
Wait for the command's answer before to send another command
Ask the actual STATUS to the laser system
Wait for the command's answer before sending another command
check the STATUS value to know if the laser system has finished the engraving process

www.datalogic.com

Datalogic S.r.l.
Via S. Vitalino, 13 | 40012 Calderara di Reno | Bologna - Italy
Tel. +39 051 3147011 | Fax +39 051 3147205

Ed. 03/2020