TCP/IP socket connection to the lasersystem

General description:

The TCP/IP connection is a socket connection in the sense of a master/slave connection. The lasersystem is the slave, running a server application, and any client (master) can connect to the server via TCP/IP.

IP- address:
The standard factory settings for the IP-address, netmask and gateway are the following: IP: 192.168.0.180 mask: 255.255.255.0 gateway: no defined

With the GUI (graphical user interface), a handheld terminal, or a touchscreen, these settings can be viewed and changed to any valid IP-address/mask.

Port:
The standard port for the TCP/IP connection is portnumber 3490. This portnumber is fixed internally and cannot be changed.

A connection can be established in the following way:

1. Connect to laser socket server (TCP/IP, port nr. 3490) until you are accepted. The server is continously trying to accept incoming clients. Up to 12 clients can be simultanously connected to the server. The next incoming accepted client would kick out the first connected client (FIFO principle). Usually, there is only one client connected to the laser.

2. When the server has accepted the incoming client the laser sends immediately a specific BYTE sequence to the client to indicate which software version is running on the laser. 6 bytes (10 bytes from version 3.3 on) are sent to the client. The byte values depend on internal hardware, firmware and software. Note, that the exact number of bytes initially sent by the laser can change in newer version of the firmware. However, as TCP/IP is packet orientated, all of these initial bytes are usually contained in the first received packet.

byte 1:
- 0xF0 : 32-bit firmware with internal barcode library
- 0xF1 : 64-bit firmware with internal barcode library
- 0xFF : for firmware without internal barcode library (obsolete)

byte 2-5: 4 digit ASCII number indicating the version nr. of the firmware

byte 6:
- 0xFF : in case of a program shutdown (if the firmware does not run anymore due to corrupted data, programming errors, etc… the system starts a different server program to indicate that the firmware is no longer running. In this case the version number will be 0000. The system must be rebooted to restart the firmware).
- 0xXX : codifying the internal hardware.

byte 7-10: a 4-byte hexcode codifying more internal hardware settings (from version 3.3 on; with version nr. >9)

3. Once accepted by the server, the client can send commands to the laser.

A command consists of a specific byte-sequence codifying the command and necessary parameters. The laser (in case of recognizing the byte sequence as a valid command) always sends an answer back to the client.
The byte sequence, both command and answer, depends always on the specific command.

General byte sequence:

The general byte sequence for all firmware versions is as follows:

|STX|DATA-BYTECOUNT|CMD-WORD|DATA-DWORDS...|ETX

STX: 0x02 (indicates start of a command frame)

ETX: 0x03 (indicates the end of a command frame)

DATA_BYTECOUNT:
Defines the number of following bytes INCLUDING the CMD_WORD but EXCLUDING the ETX-byte.

CMD-WORD:
(2 bytes) defines the command.
NOTE: the CMD-WORD is BYTE switched (little endian), that means that the lower byte is sent before the upper byte !

DATA-DWORDS:
n * (4 bytes) : the DATA DWORDS contain the DATA to be sent, typically ordered in DWORD blocks (4 byte blocks). The interpretation of the DWORD blocks depends on the command.

Notes:

a. The DATA_BYTECOUNT needs to be set correctly according to the DATA-DWORDS for each command. According to this number the server expects the ETX as the last BYTE. If this is not the case, the server will not accept the command as a valid command and will not send any answer to the client.

b. If the frame data are incomplete the server discards the data after a minimum time of 10 seconds within the arrival of the next data.
e.g. if you send STX DAT-BYTECOUNT CMD_WORD only, the server is waiting for more data up to 10 seconds. With the arrival of the next data packet after 10 seconds, the laser will discard the command.

c. As the DATA_BYTECOUNT is just a single byte the length of a command frame is limited with this command sequence. To be able to send and receive longer frames the 'extended' command sequence has to be used.

Extended byte sequence:

The extended byte sequence is as follows:

|STX|0x04|EXTENDED CMD-WORD|DATA-BYTECOUNT-WORD|DATA-BYTES....|ETX

STX: 0x02 (indicates start of a command frame)

ETX: 0x03 (indicates the end of a command frame)

EXTENDED CMD-WORD:
(2 bytes) defines the command. In an EXTENDED CMD-WORD the upper byte is always different from 0x00.
NOTE: the EXTENDED CMD-WORD is BYTE switched (little endian), that means that the lower byte is sent before the upper byte !

DATA_BYTECOUNT-WORD:
(2 bytes) defines the number of the DATA-BYTES.
NOTE: the DATA_BYECOUNT-WORD is BYTE switched (little endian), that means that the lower byte is sent before the upper byte !

Note that the number of bytes in a complete frame is limited internally to 2048 including header and trailer bytes.

Examples:

Getting the status of the system:

CMD: 0x0070

With this command you will receive the actual status of the laser system.

sequence:

|0x02|0x02|0x70 0x00|0x03|

The laser answers with:

0x02 | 0x32 | 0x70 0x00 |STATUS DATA BYTES…| 0x03

STATUS DATA BYTES: 48 bytes containing the status. The 48 bytes come in blocks of 4 bytes that form a DWORD (execept the name-Bytes). All DWORDS are sent **in little endian format (LSByte comes first, MSByte last, also called byte switched).**

1 – 4 : d_counter (byte switched) ; a 32-bit counter indicating the number of o.k. prints since entering into the printing mode.

5 – 8 : s_counter (byte switched); a 32-bit counter indicating the number of prints since entering into the printing mode.

9 – 12: messageport (byte switched); indicates the bit combination of the external message selection (when external message mode is activated)

13 – 16: (byte-switched)

Byte0: mode byte;

0x00: default , system is in standard mode.
0x01: system is in external message selection mode
0x04: system is in a batch-job mode.

Byte1:option byte; (internal use), indicates the type of request that the system is doing.

Byte2 : request byte; (internal use)
0x00: default value.
0x01: request; the system is requesting information to the GUI (the external controlling software of the lasersystem).

Byte3 : Start byte:

Bit0: when set, system is in printing mode (waiting for photocell/PLC)

Bit1: when set, system is actually printing a message

Bit2: when set, system is waiting for a 'alarm reset' signal (DSP cards only).

Bit3: when set, system is waiting for an input signal (DSP cards only)

Bit4: when set, system is waiting for an external motorized axis to reach a new position.

Bit5: when set, system is in 'printsession' mode.

Bit6...Bit7: reserved

17 – 20: t_counter; (byte switched) ; a 32-bit counter indicating the total number of prints of the laser system. Each time when the laser is booting it reads the total number of prints from the harddisk. This value is saved each time when the system leaves the printing mode (e.g. pressing the STOP button, sending the STOP command). It is not saved automatically after each print, because this would consume an important amount of CPU time.

21 – 24: copies (byte switched); the actual number of print copies. This number is set within the Start Print command and defines the number of prints that the laser should do until leaving the printing mode. Typically, this value is set to ZERO, meaning infinitely printing.

25 – 28: (byte-switched)

WORD0: alarm; codifies the alarm status of the system.

0x0000:	no alarms
0x0C0E:	wrong messageport (no file found for the external message selection)
0x0848:	alarms active; some alarms are active (interlock, shutter,….)
0xFFFF:	initialization of the firmware has failed due to some hardware failure.

WORD1: alarm code (byte switched); codifies the last active alarm. If all alarms are cleared (no longer active), it codifies the last alarm, that was active. To check the alarm status of the system, you have to check the alarm WORD. If the alarm WORD is ZERO, the system is ready for printing. If any alarm is active (alarm WORD not ZERO), the system will not be ready for printing and you can get information of the last active alarm.
(see list of alarmcodes later in this document)

29 – 32: printtime (byte switched); gives the printing time of the last printed message in milliseconds.

33 – 40: name; the actual filename of the message (up to 8 bytes).

41 – 44: alarm-mask (byte switched). Each bit codes an alarm according to the following list:

0x00000001	Interlock
0x00000002	OEM-shutter
0x00000004	Overtemperature
0x00000008	Shutter
0x00000010	Laser not ready
0x00000020	X-scanner failure
0x00000040	Y-scanner failure
0x00000080	power failure(D5000 series and FIBER-laser (MO))

0x00000100	Z-scanner failure
0x00000200	Laser not armed
0x00000400	XY outofrange
0x00000800	Q-switch (D-5000 B-series)
0x00001000	triggersignal
0x00002000	file not allowed (wrong version)
0x00004000	overspeed
0x00008000	harddisk full
0x00010000	barcode creation failure
0x00020000	barcode licence failure
0x00040000	barcode library failure
0x00080000	invalid file
0x00100000	database failure
0x00200000	max.-distance alarm
0x00400000	min.-distance alarm
0x00800000	client-timeout (tcpip)
0x01000000	invalid font
0x02000000	belt stopped
0x04000000	empty message
0x08000000	initialization error
0x10000000	memory error
0x20000000	warmup in progress
0x40000000	OEM alarm active
0x80000000	extended alarm active (extended alarm mask of the extended status codes the extended alarms)

45 – 48: signalstate (byte switched): codes the state of the inputs of the system when it was triggered for printing (for internal purpose only)

Getting the extended status of the system:

CMD: 0x0170

With this command you will receive the actual status of the laser system.

sequence:

|STX|0x04|0x70 0x01|0x00 0x00|ETX

The laser answers with:

0x02 | 0x04 | 0x70 0x01 |DATA-BYTECOUNT-WORD|DATA-BYTES....|ETX

DATA-BYTECOUNT-WORD: (number of DATA-BYTES = n)
DATA BYTES: n bytes containing the status. The number of DATA-BYTES can vary within different firmware versions. The application should only read and interpret the received DATA-BYTES according to the received DATA-BYTECOUNT-WORD. It is a common feature that in newer firmware versions the extended status information grows with respect to older versions.

1 – 32: same content as in the Status command, except that all DWORDS according the above documentation are not byte-switched, i.e. the most significant Byte is sent first (big endian).

33 – 36: alarm-mask (MSB comes first). See Status command.
37 – 40: signalstate (MSB comes first). See Status command.
41 – 56: name; the actual filename + extension of the message (up to 16 bytes).
57 – 72: eventhandler; the actual filename + extension of the eventhandler-file (up to 16 bytes).

73 – 76: extended alarm-mask (MSB comes first). See Status command.
When the alarm-mask has the 'extended alarmmask' -bit (0x80000000) set, this DWORD codes the extended alarms.

0x00000001	Lasermeasurement failed
0x00000002	UV laser not ready
0x00000004	Pixmap out of range
0x00000008	Channelstatus error
0x00000010	PWM out of range
0x00000020	RTC alarm (real time clock)
0x00000040	CPU overtemperature
0x00000080	Board overtemperature
0x00000100	Undervoltage 5V
0x00000200	Undervoltage 3.3V
0x08000000	DSP alarmmask failure
0x10000000	Fpga-watchdog
0x20000000	Wrong lasertype selected with dipswitches
0x40000000	DSP is paused
0x80000000	Fpga failure

77 : not used
78 : not used
79-80 : Bit0-Bit9: defines the used tracking distance in dynamic applications in permille of the available scanfield.

Bit10-Bit13: codes the actual printing mode

0: static
1: dynamic
2: dynamic-distance
3: static-dynamic
4: stepper mode
5: XY stepper mode

Sending a global internal counter:

A global internal counter is a counter identified by a field number. Any numerical value can be sent to the field (64 bit counter).

CMD: 0x0090
1. DATA_DWORD: field number of the counter (BYTE switched)
2. DATA-DWORD: high DWORD of the counter value (BYTE switched)
3. DATA-DWORD: low DWORD of the counter value (BYTE switched)

In this case, the DATA_BYTECOUNT will be 1 x 2 bytes + 3 x 4 bytes = 14.

Example: field = 3 , counter value = 7

|STX |DATA-BYTECOUNT|CMD-word|FIELDNUMBER-DWORD|upper DWORD of counter|lower DWORD of counter|ETX

|0x02|0x0E|0x90 0x00|0x03 0x00 0x00 0x00|0x00 0x00 0x00 0x00| 0x07 0x00 0x00 0x00|0x03|

The laser will answer with:

STX|n data-bytes|CMD-word|Fieldnumber-DWORD|ETX

|0x02|0x06|0x90 0x00|0x03 0x00 0x00 0x00| 0x03

If the sent fieldnumber is invalid (values from 0 - 15) the laser answers with a Fieldnumber-DWORD value of |0xFF 0xFF 0x00 0x00|.

Requesting a global internal counter:

CMD: 0x0092

1.DATA-DWORD: field number of the requested counter

example: field 3, (and the counter value is 7)

sequence:
|STX| DATA-BYTECOUNT|CMD-word|FIELDNUMBER-DWORD|ETX

|0x02|0x06|0x92 0x00|0x03 0x00 0x00 0x00|0x03|

The laser answers with:

|STX| DATA-BYTECOUNT|CMD-word|FIELDNUMBER-DWORD|upper DWORD of counter|lower DWORD of counter|ETX

|0x02|0x0E|0x92 0x00|0x03 0x00 0x00 0x00|0x07 0x00 0x00 0x00| 0x00 0x00 0x00 0x00| 0x03|

If the requested field number is invalid, or the counter is still not initialized (e.g. still not set with the 0x90 command or still no message loaded

with an internal counter after a reboot), the FIELDNUMBER-DWORD will be set to |0xFF 0xFF 0x00 0x00|, and the counter value will be zero.

Changing "repeats" of a global internal counter:

A global internal counter is a counter identified by a field number. The "repeats" is the number of prints of an internal counter without increasing it by the defined "steps". Usually, "repeats" and "step" is set in the message for each counter independently. With this command you can change the number of "repeats" for the internal counters of the actual loaded message. The counters, whose "repeats" are to be changed are identified by the "fieldnumber".

Not only the "repeats" are set with this command, but also the number of prints of this counter. Usually, the number of prints is zero, and the counter will perform its next increment after "repeats" prints. If the number of prints are set to a value > 0, the counter will perform its next increment after ("repeats" – "number of prints") prints and from then on it will increment after "repeats" prints.

CMD: 0x0093
1. DATA_DWORD: field number of the counter (BYTE switched)
2. DATA-DWORD: repeats of the counter (BYTE switched)
3. DATA-DWORD: number of prints of the counter value (BYTE switched)

Example: setting the "repeat" of counters of field nr 2 to 8 repeats and the actual number of prints to 3.
In this case, the counters will increment after the next (8-3)=5 prints, and then they will increment after each 8 prints.

|0x02|0x0E|0x93 0x00|0x02 0x00 0x00 0x00|0x08 0x00 0x00 0x00| 0x03 0x00 0x00 0x00|0x03|

The laser will answer with:

STX|n data-bytes|CMD-word|Fieldnumber-DWORD|ETX

|0x02|0x06|0x93 0x00|0x02 0x00 0x00 0x00| 0x03

If the sent fieldnumber is invalid (allowed values from 0 - 15) the laser answers with a Fieldnumber-DWORD value of |0xFF 0xFF 0x00 0x00|.

Start of printing:

CMD: 0x002D

This command switches the laser into the printing mode and the laser will (if no alarm occurs) stay in the printing loop until the STOP signal (see later) is sent. When the laser is in the printing mode it waits, according to the configuration settings of the system, for photocell or PLC input signal to activate printing.

sequence:

|0x02|0x16|0x2D 0x00|MODE-DWORD| COPIES-DWORD| BATCH-DWORD| 8-bytes name of file|0x03|

MODE-DWORD (byte switched):

0xFFFF: activates the external message selection (in this case name of the file must be empty or NULLs !)

0x0000-0x00FF: default mode.
If the filename is empty the internal filename is set to to <num>.msf where <num> is the numerical decimal value of the MODE-DWORD, and the external message table mode , if active, will be deactivated.
In combination with the BATCH-DWORD you can activate a specific entry in the message table in batch-mode.
If BATCH_DWORD is "0" the external messagetable and the batch job mode are disabled.
If BATCH_DWORD is non zero, the batch job mode is enabled and the n.th entry of the batch job table is activated where <n> is determined by the MODE_DWORD.

If the filename is explicitly given the actual mode is not changed. In batch job mode it is not recommended to set explicitly the filename !

0xFFFFFFFF: Only applies when the filename is empty and the batch mode is not active ! In this case, the mode is set to the default mode and the actual loaded file will not be reloaded. Thus, the system just switches into the printing mode without reloading any file.
This option is available from Version 5.6.6 on. It allows to use the start command without reloading of the actual selected file.

COPIES-DWORD (byte switched):

Sets the number of prints to be done. If set to zero, the system prints infinitely (depending on the configuration of the photocell/PLC). When set to “1” the systems performs immediately one print without waiting for the photocell/PLC signal.
To print just 1 copy but with waiting for the photocell/PLC signal you have to set this DWORD to 0xFFFFFFFF.

BATCH-DWORD (byte switched):

0x0000: default (deactivates any batch-job)
0xXXXX: any other value not ZERO activates the batch-job mode of the laser system, except in case of a 0xFFFF MODE-DWORD which in any case activates the external message selection.
The batch-job mode is a special application mode, where the system prints different messages due to previously defined table.

8.bytes filename: without extension and filled up with ZEROs.

Note: from firmware 5.0.8 on the filename can be up to 12 bytes long and the extension can be sent, too (thus up 16 bytes can be sent as the filename). This

is especially important if you want to print xml files. When no extension is given, the file is assumed to be a msf-file.
If the filename length is not a multiple of 4 you must add ALWAYS a termination NULL character !

example : filename test.msf

0x02 | 0x16 | 0x2D 0x00 |0x00 0x00 0x00 0x00| 0x00 0x00 0x00 0x00| 0x00 0x00 0x00 0x00| 0x74 0x65 0x73 0x74 0x00 0x00 0x00 0x00| 0x03

The laser answers with:

0x02 | 0x06 | 0x2D 0x00| DWORD| 0x03|, where DWORD (byte switched !) indicates the following:

0x0000FFF1: O.K. laser switched into printing mode.

0xx00000C0C: Not OK, file is not valid or not existent. The laser will not be switched into the printing mode.

0x00000848: Not OK, the file is a valid file, but there are some alarms active (e.g. shutter closed, interlock open, …).

Stop of printing:

CMD: 0x002E

This command stops the printing mode immediately and interrupts a possible marking process.

sequence:

|02|02|2E|00|03

The laser answers with:

|02|02|2E|00|03

Start/stop printsession:

CMD: 0x0080

A 'printsession' prepares the laser for the printing mode. It usually enables the laser for printing, opens an optional shutter, moves the scanner to the optional “keepwarm” position and sets the diode current for YAG systems to the minimum powerlevel. If the laser has the autopointer option activated, this call activates the red pointer.
This call should usually be used before a 'Start' command.

When a printsession is ended the laser controller will be disabled as well as the red pointer.

sequence:

|0x02|0x16|0x80 0x00|DWORD0|DWORD1|DWORD2|DWORD3|DWORD4|0x03|

DWORD0: 0x00000000 ends a printsession
0x00000001 starts a printsession
DWORD1 – DWORD4: unused and should be 0x00000000.

The laser answers with:

|02|02|80|00|03

STX ADDR 0x80 COMMANDBYTE CRC ETX

COMMANDBYTE: 0x00 end a printsession and printing mode
0x01 start a printsession

Answer: STX ADDR 0x80 ACK CRC ETX

Select a message (file):

CMD: 0x0057

sequence:

|0x02|0x0A|0x57 0x00|8-bytes name of file|0x03|

8-bytes filename: without extension and filled up with ZEROs. (see the Note)

example : filename test.msf

0x02 | 0x0A | 0x57 0x00 |0x74 0x65 0x73 0x74 0x00 0x00 0x00 0x00| 0x03

The laser answers always with:

0x02 | 0x02 |0x57 0x00| 0x03|

Note: from firmware 5.0.8 on the filename can be up to 12 bytes long and the extension can be sent, too (thus up 16 bytes can be sent as the filename). This is especially important if you want to print xml files. When no extension is given, the file is assumed to be a msf-file.
If the filename length is not a multiple of 4 you must add ALWAYS a termination NULL character !

Sending a usermessage:

The direct sending/requesting of a usermessage requires a new byte sequence scheme. This new scheme is only available from firmware version 3.3 on.

A usermessage is identified by its field number (numbers from 0 – 255). Note that fields 0 – 15 (0 – 35 from firmware 5.4.9 on) may be set as FIFO buffers (see ‘buffered usermessages’).

CMD: 0x0141

The sequence must be:

a) sending

|STX|0x04|0x41 0x01|DATA-BYTECOUNT-WORD|Option-BYTE|FIELDNUMBER-BYTE|ASCII STRING|ETX|

b) requesting

|STX|0x04|0x41 0x01|DATA-BYTECOUNT-WORD|Option-BYTE|FIELDNUMBER-BYTE|INDEX-WORD|ETX|
Note that the DATA-BYTECOUNT must be set to 4 !!!

DATA-BYTECOUNT-WORD:	2 bytes (byte switched); is the number of the following bytes of the frame excluding ETX.
Option-BYTE:	1 byte; 0x00: a usermessage will be set. 0x01: a usermessage will be requested. 0x02: an entry of a FIFO field is requested 0x03: all FIFO buffers are dumped into a file
FIELDNUMBER-BYTE:	1 byte; defines the fieldnumber of the usermessage
ASCII-STRING:	the ASCII byte string (without terminating ‘\0’) to be sent if a usermessage is sent to the laser (option-byte == 0x00). In case of a request no ASCII STRING data has to be sent !
INDEX-WORD:	2 bytes (byte switched), mandatory when option-byte=0x02 Defines the n-th last entry inside a FIFO field. The index ‘0’ gives you the last value in the FIFO.

The laser answers with:

a) Setting a usermessage:

|STX|04|41|01|XXXX|NUMBER-BYTE|BYTE0|BYTE1|….|ETX

XXXX : DATA-BYTECOUNT-WORD (number of following bytes excluding ETX)
NUMBER-BYTE: gives the number of usermessages that were sent with this command (default: 0x01)

BYTE0,1,2…: For each sent usermessage field within a single command an additional byte is returned indicating if the string for this field has been accepted or not. ‘1’ means that it has been accepted, ‘0’ means it has not been accepted, ‘2’ means that is has been accepted and was written at the index that is going to be printed next. This can be important when sending multiple strings within a single command in combination with an enabled FIFO-buffer. With the BYTE0,1,2,…. You can test if each single message has been placed into the FIFO.

b) Getting a usermessage:

|STX|04|41|01| XXXX | FIELDNUMBER-BYTE | ASCII-STRING| ETX

XXXX : DATA-BYTECOUNT-WORD (number of following bytes excluding ETX)

c) Getting an entry of a FIFO-field:

|STX|04|41|01|04|00|1. FIELDNUMBER-BYTE|INDEX-WORD|FIFO_ELEMENTS-WORD| ASCII-STRING|03|

FIFO_ELEMENTS-WORD: 2 bytes (byte switched), the number of actual entries in the FIFO-field

d) Dumping of the FIFO fields

|STX|04|41|01|00|00|03

The dump file "umdump.tmp" is created in the RAM-disk of the laser and can be retrieved later with the copy command. Once retrieved with the copy command it will be automatically be removed from the RAM-disk.
Note: a dump file will only be created when the system is not in printing mode or when there is an alarm active !!! In printing-mode the command is ignored and a NACK message is sent back.

In case of missing or wrong data the laser responds with a NACK message.

|STX|04|15|00|00|00|03

Note: You can set/request various usermessages at once (within one command) in the following way.

Setting multiple usermessages (using '0x00' as separation character):

|STX|04|41|01|XXXX|Option-byte==0x00|1. FIELDNUMBER-BYTE|ASCII-STRING| 0x00|2.FIELDNUMBER-BYTE|ASCII-STRING|0x00|……..|n.FIELDNUMBER-BYTE|ASCII-STRING|03| where

XXXX : DATA-BYTECOUNT-WORD (number of following bytes excluding ETX)

The laser answers with:

|STX|04|41|01|XXXX|NUMBER-BYTE|BYTE0|BYTE1|….|ETX where

XXXX : DATA-BYTECOUNT-WORD (number of following bytes excluding ETX)
NUMBER-BYTE is set to the number of usermessages that were set within the command, and BYTE0,1,2,…. indicate if the message for the field has been accepted ('1') or not ('0').

Getting multiple usermessages :

|STX|04|41|01| XXXX|Option-byte==0x01|1. FIELDNUMBER-BYTE | 2.FIELDNUMBER-BYTE|3.FIELDNUMBER-BYTE|…..| ETX| where

XXXX : DATA-BYTECOUNT-WORD (number of following bytes excluding ETX)

The laser answers with:

|STX|04|41|01|XXXX|1. FIELDNUMBER-BYTE|ASCII-STRING|0x00|2.FIELDNUMBER-BYTE|ASCII-STRING|0x00|……..|n.FIELDNUMBER-BYTE|ASCII-STRING|03| where

XXXX : DATA-BYTECOUNT-WORD (number of following bytes excluding ETX)

Note: Frames are always limited to a length of 2048 including leading and trailing bytes. Thus, be careful when you send or request multiple usermessage fields within a single command !

Example:

Sending "ABCDEFG" to field number Zero.

Send: Command: 0x0141
DATA-BYTECOUNT-WORD: 0x0009
Option-BYTE: 0x00
FIELDNUMBER-BYTE: 0x00
ASCII-STRING: 0x41|0x42|0x43|0x44|0x45|0x46|0x47

Frame (in hex): |02|04|41|01|09|00|00|00|41|42|43|44|45|46|47|03

Answer (in hex): |02|04|41|01|01|00|01|03

Sending "ABC" to field 0 and "DEF" to field 1.

Frame (in hex): |02|04|41|01|0A|00|00|00|41|42|43|00|01|44|45|46|03

Answer (in hex): |02|04|41|01|01|00|02|03

Sending a UTF8 usermessage:

The text can also be sent as an UTF8 string. The command sequence is the same as the 0x0141 command.

CMD: 0x0143

The sequence must be:

|STX|0x04|0x43 0x01|DATA-BYTECOUNT-WORD|Option-BYTE|FIELDNUMBER-BYTE|UTF8 STRING|ETX|

Note that the DATA-BYTECOUNT must be set to 4 !!!

DATA-BYTECOUNT-WORD: 2 bytes (byte switched);
is the number of the following bytes of the frame excluding ETX.

Option-BYTE: 1 byte;

0x00: a UTF8 usermessage will be set.
0x01: a UTF8 usermessage will be requested.

FIELDNUMBER-BYTE: 1 byte; defines the fieldnumber of the usermessage

UTF8-STRING: the UTF8 string (without terminating '\0') to be sent if a usermessage is sent to the laser (option-byte == 0x00). In case of a request no UTF8 STRING data has to be sent !

The laser answers with:

in case of setting a usermessage:

|STX|04|43|01|01|00|NUMBER-BYTE|ETX

NUMBER-BYTE: gives the number of usermessages that were sent with this command (default: 0x01)

in case of getting a usermessage:

|STX|04|43|01| DATA-BYTECOUNT-WORD| FIELDNUMBER-BYTE | UTF8-STRING| ETX

Enable/disable buffered usermessages:(from firmware 3.6.2 on)

You can enable buffered usermessages for field 0 – 15 (0-35 from version 5.4.9 on). For this purpose you have to enable the buffersize with this command. Allowed buffersizes range from 0 – 255 (0 – 1000 from version 5.4.9 on). A buffersize of 0 disables the buffer and the usermessages are used as usual. Typical buffersizes are 20 – 40.
Once enabled, the fields 0 – 15 (0-35) are used as buffers (FIFO). Each Usermessage command puts a value into the buffer (FIFO) until the buffer is full. If the buffer is full, the Usermessage command returns 0 as NUMBER-BYTE, so you may have to wait until sending more usermessages. Do not send multiple usermessages within one command for this purpose, if you are not sure that they fit into the buffer, because otherwise you will not know which element did not fit into the buffer.

The laser will print from its internal buffer in a FIFO order. Each entry is printed exactly one-time. The laser does not check if two entries have the same value or not! Each entry that is printed is removed after printing. If the buffer is empty and the laser is triggered to print, a "EMPTMESSAGE" alarm is raised and the laser stops printing. The buffer must have at least one element to proceed printing. If the laser was not set to autostart-mode the laser needs a StartPrint command to proceed printing, else a sending of a Usermessage will fill the empty buffer and printing proceeds.

Firmware 3.6.2 -4.2.8

CMD: 0x0063

sequence:

|02|0A|63|00|DWORD1|DWORD2|03

DWORD1: 0x00000000 (enable the buffer according to the buffersize)
0x00000001 (gets the actual reserved buffersize)

DWORD2: defines the buffersize to be reserved for the usermessages
(values : 0 (disables the buffered usermessage) to 255)

The laser answers with:

|02|06|63|00|DWORD1|03

DWORD1: the size of the reserved buffer

All DWORDs are byte-switched.

From firmware 4.2.9 on

CMD: 0x0063

sequence:

|02|XX|63|00|DWORD1|DWORD2|DWORD3|03

XX : DATA-BYTECOUNT (number of following bytes excluding ETX)

DWORD1: 0x00000000 (enable the buffer according to the buffersize)
0x00000001 (gets the actual reserved buffersize and fillstatus)
0x00000002 (resets the FIFO of a field gets the fillstatus before the reset)

DWORD2: defines the buffersize to be reserved for the usermessages (values : 0 (disables the buffered usermessage) to 255)

DWORD3: (optional)
enable (DWORD1=0): number of consecutive fields to be used for buffering. The default value of fields is 36 (field 0 – 35). If this value is set to 0, no change of the number of fields is done, if the buffer has been enabled once (i.e. it will remain at 36 fields). Changing the number of fields to be used for buffering is available from firmware 5.5.0 on.

request(DWORD1=1): field to be requested.

reset (DWORD1=2): field to be resetted.

The laser answers with:

|02|XX|63|00|DWORD1|DWORD2|DWORD3|03

DWORD1: the size of the reserved buffer
DWORD2: the requested field number (in case of request/reset) or the number of fields used for buffering (in case of enable the buffer).

DWORD3: the fillstatus of the requested field (number of elements in the field-FIFO)

All DWORDs are byte-switched.

Note: The number of fields reserved for buffered usermessages has increased during the evolution of the control-program from 4 (version 3.6.2) to 16 (5.3.4)

Sending a datastring to an OEM-bitmap field:

OEM-bitmaps are bitmaps whose pixel-data are filled with a TCP-command. Internally up to 6 datastrings are reserved to fill the pixels of OEM-bitmaps, each identified by its fieldnumber (0,1,2,3,4,5). The 6 internal datastring fields are similar to the internal usermessage fields, just that its content is an arbitrary byte-sequence that codifies the pixel-data of a bitmap.

The format of the pixel-data can be defined in the settings of the bitmap object. Two formats are allowed:

1-bit per pixel: one byte in the sequence holds 8 pixel data. A set bit means black color and an unset bit means white color.

8-bit per pixel: one byte holds the greyscale value of 1 pixel.

The pixeldata of an OEM-bitmap are filled from the bottom left of the bitmap to the upper right, scanning the bitmap from left to right.
No padding,leading or trailing bytes/bits are in the datastring sequence. If the datastring is longer than the size of the bitmap the remnant bytes/bits are ignored.
For the '1-bit per pixel' format the order of the bits should be from MSB to LSB.
If the datastring does not fill the OEM-bitmap, the OEM-bitmap will not change.

CMD: 0x0144

The sequence must be:

|STX|0x04|0x44 0x01|DATA-BYTECOUNT-WORD|SubCmd-BYTE|FIELD-BYTE|FRAME-BYTE|DATA STRING|ETX|

DATA-BYTECOUNT-WORD: 2 bytes (byte switched);
is the number of the following bytes of the frame excluding ETX.

SubCmd-BYTE: 1 byte; a bitmask

0x00: get a <frame> of <field> of a datastring.

0x01(bit0): empty and set a <field> of a datastring.
This clears a datastring field and fills it with new data.

0x02(bit1): add and terminate a <field> of a datastring.
This adds data to a datastring field and terminates the datastring. You need to set this bit when all data for a field has been sent.

0x04(bit2): add data to a <field>
This just adds more data to a datstring field. This bit is only used when you need to send more than 2000 bytes.

Thus, if you need to send less than 2000 data you should set the SubCmd to 0x03.

If you need to send more than 2000 data, you would have to pack them into frames. Within the first frame you set the subcmd to 0x01. The next frames would have a subcmd of 0x04 and the last frame would have a subcmd of 0x02.

Note: when the FIFO buffer is enabled, then a new entry will only be written when bit1 is set !!!

FIELD-BYTE: 1 byte; defines the fieldnumber of the datastring.

FRAME-BYTE: 1 byte; defines the frame number in case of getting a datastring. In case of setting a datastring this value will be ignored and should be set to 0.

DATA-STRING: The byte-sequence to be sent if a datastring is sent to the laser. In case of a getting a datastring no DATA-STRING has to be sent ! The maximum size of the DATA-STRING sequence is 2000. If more data has to be sent/get the data must be split into several commands.

The laser answers with:

in case of setting a DATASTRING:

|STX|04|44|01|DATA-BYTECOUNT-WORD|FIELD-BYTE|INFO-BYTE|ETX

FIELD-BYTE: gives the fieldnumber of the DATASTRING that was sent with this command.
INFO-BYTE: 1 byte; a bitmask

0x01: data received

In case of enabled buffered DataStrings the following two more bits may be set in the answer.

0x02: data were set at the index that is the next index to be printed.

0x04: data were not added because the FIFO buffer is full.

in case of getting a DATASTRING:

|STX|04|44|01| DATA-BYTECOUNT-WORD| FRAME-BYTE | FIELD-BYTE | DATA-STRING| ETX

FIELD-BYTE: the fieldnumber of the DATASTRING that was sent with this command.

FRAME-BYTE: the next framenumber of the DATASTRING in case that more than 2000 data are get. If the framenumber is equal to the requested framenumber then no more data are within the datastring.

Enable/disable buffered datastrings (OEM bitmaps):

CMD: 0x0064

sequence:

|02|XX|64|00|DWORD1|DWORD2|DWORD3|03

DWORD1: 0x00000000 (enable the buffer according to the buffersize)
0x00000001 (gets the actual reserved buffersize and fillstatus)
0x00000002 (resets the FIFO of a field gets the fillstatus before

the reset)

DWORD2: defines the buffersize to be reserved for the usermessages
(values : 0 (disables the buffered datastrings) to 255)

DWORD3: (optional)
defines the field (0,1,2,3,4,5) in case of requesting
(DWORD1=0x00000001) or in case of resetting the FIFO of a field
(DWORD1=0x00000002)

The laser answers with:

|02|XX|63|00|DWORD1|DWORD2|DWORD3|03

DWORD1: the size of the reserved buffer
DWORD2: the requested field number (in case of request/reset)
DWORD3: the fillstatus of the requested field (number of elements in the field-FIFO)

All DWORDs are byte-switched.

Getting the internal system time:(from firmware 4.0.2 on)

CMD: 0x0038

sequence:

|02|02|38|00|03

The laser answers with:

|02|1E|38|00|DWORD1|DWORD2|DWORD3|DWORD4|DWORD5|DWORD6|DWORD7|03

All DWORDs are byte-switched.

DWORD1: day of the month (1-31)
DWORD2: month (1-12)
DWORD3: year (**number of years since 1980** !!!)
DWORD4: day of the week (ignored)
DWORD5: hour (0-23)
DWORD6: minute (0-59)
DWORD7: seconds (0-59)

Setting the internal system time:

CMD: 0x0039

sequence:

|02|1E|39|00|DWORD1|DWORD2|DWORD3|DWORD4|DWORD5|DWORD6|DWORD7|03

The laser answers with:

|02|06|39|00|ERROR-DWORD|03

All DWORDs are byte-switched.

DWORD1: day of the month (1-31)
DWORD2: month (1-12)
DWORD3: year (1980-2038)
DWORD4: day of the week (ignored)
DWORD5: hour (0-23)
DWORD6: minute (0-59)
DWORD7: seconds (0-59)

ERROR-DWORD: 0x00000000 -> time settings saved.
0xFFFFFFFF or other value->time setting failed

NOTE: the interpretation of the year is different when setting or getting the signal.

Requesting messages inside the laser: (from firmware 4.0.4 on)

CMD: 0x0126

The sequence must be:

|STX|0x04|0x26 0x01| Data-bytecount-WORD |Filetype-BYTE|Frame-BYTE|EXTENSION| ETX|

DATA-BYTECOUNT-WORD: (byte switched) counts the following bytes except ETX.

Filetype-BYTE: 1 byte; defines the filetypes to be retrieved and is a bit combination with the following meaning.

0x01: *.msf, *.db, *.xml and *.evh files
(message files, database files, xml-message files, eventhandler files)

0x02: *.mfs and *.unx (font files)

0x04: *.mft (TrueType font files)

0x00: *.EXTENSION files, where EXTENSION specifies the file-extension to be requested (from firmware 5.1.6 on)

0x10: when set, the number of filenames per frame is set to 10, else to 25.

Frame-BYTE: 1 byte; defines the frame to be requested; data are sent in frames of 10/25 filenames per command. The first frame must have the frame byte 0x00, the next 0x01 etc….

Extension-bytes: optional file extension for filetype-byte = 0

The laser answers with:

|STX|04|26|01|DATA-BYTECOUNT-WORD|DATA|ETX

DATA-BYTECOUNT-WORD: (byte switched) counts the following bytes except ETX.

DATA: the ASCII names of the requested filetypes and frame separated by 0x0A.

Request sequential numbers/user messages of the actual message: (from firmware 4.0.4 on)

CMD: 0x019D

The sequence must be:

|STX|0x04|0x9D 0x01|0x02 0x00|Objecttype-BYTE|ID-BYTE|ETX|

Objecttype-BYTE: 1 byte: 0x00 : request sequential number of actual message
0x01: request internal global number (must NOT necessarily be in actual message)
0x02: request text of usermessage (must NOT necessarily be in actual message
0x04: request n-th (defined by ID-byte) internal global number in actual message

0x08: request n-th (defined by ID-byte) usermessage in actual message

ID-BYTE:

1 byte: number or field to be requested.

If we request a sequential number (Objecttype-byte=0x00), then this byte defines the index of the sequential number to be requested.

Example: (Objecttype byte = 0x00)

Second byte:

0x00 : we request the first sequential number of the actual message.

0x01: we request the second sequential number of the actual message.

n: we request the (n + 1)-th sequential number of the actual message.

The index of the sequential numbers in a message are due to the creation of the message. That means, that the first sequential number that we have created in the message will have the index 0, the second one will have the index 1 and so on....

If we request an internal global number or a usermessage, this byte defines the fieldnumber of the requested internal global number/usermessage.

The laser answers with:

|STX|04|9D|01|DATA-BYTECOUNT-WORD|ID-WORD|DATA|ETX

DATA-BYTECOUNT-WORD: (byte switched) counts the following bytes except ETX.

ID-WORD (byte switched):
2 bytes, representing the ID of the requested object, and has the same value as the ID-BYTE (but as a WORD !). If the ID-WORD is 0xFFFF the requested data are not available (i.e. the requested object does not exist inside the actual message)

DATA: the ASCII string, usually without termination '\0', of the requested object.

Set sequential numbers of the actual message (no global internal counter !) (from firmware 4.0.4 on):

CMD: 0x019E

The sequence must be:

|STX|0x04|0x9E 0x01|Data-bytecount-WORD|ID-WORD|DATA|ETX|

DATA-BYTECOUNT-WORD: (byte switched) counts the following bytes except ETX.

ID-WORD (byte switched):

2 bytes, representing the ID of the sequential number.

e.g. ID-WORD= 0x00 0x00 refers to the first sequential number inside the actual message, 0x01 0x00 refers to the next sequential number.

DATA: the ASCII string of the value to be set (can include a termination '\0'.

The laser answers with:

|STX|04|9E|01|01 00|ACK|ETX in case that the sequential number exists.

|STX|04|9E|01|01 00|NACK|ETX in case that the sequential number does not exist.

Note: when the sequential number object is of "local" type, each time the message is loaded for printing (e.g. with the startprint command 0x2D) the value of the counter is reset to the initial value.

Software Triggersignal to print one sample:

CMD: 0x0056

The sequence must be:

|STX|0x02|0x56 0x00|ETX|

Answer:

|STX|0x02|0x56 0x00|ETX| (in case of no alarms and printing mode)

|STX|06|56|00|15|00|00|00|ETX| (in any other case)

Note: this command simulates a hardware trigger of the PLC/Photocell input. You can use it as a software trigger instead of the PLC/Photocell input. The laser must be set previously into the print modus with the PrintStart command (0x2D, typically with 0 copies).

Closing the connection:

It is recommended that you send a special command to the laser to indicate when you want to close the connection (e.g. when you close your controlling software). The "Knockout" command (0x00F0) has the following sequence:

|STX|n data-bytes|CMD-word|ETX

|0x02|0x02|0xF0 0x00|0x03|

the laser answers with

|STX|n data-bytes|CMD-word|ETX

|0x02|0x02|0xF0 0x00|0x03|

After sending the answer to the client, the laser closes the socket connection to the client.

Alarmcodes (byte 27-28 of the Status data):

For a complete list of the alarm codes see 'alarmcodes.pdf'.

Copy a file to the laser:

To upload a message (msf or xml file) to the laser you should know that only files with extensions should be sent to the laser. Files without extension are identified as executables and internally treated different. To avoid overwriting of important system files, do no send any files without extensions to the laser, as the integrity of the laser's filesystem may not be guaranteed.

A file is copied to the laser in frames and as a sequence of consecutive TCPIP protocol commands. As soon as the laser starts a file copy, the laser is exclusively switched to the copy process and does not mark or respond to anything else as long as the file copy has finished. This is due to internal security to maintain the consistency of the filesystem.

A copy command starts with the following sequence:

CMD: 0x0061

|STX| XX|0x61 0x00|FILE_SIZE_DWORD|OPTIONCOMMANDBYTE OPTIONBYTE 0x00 0x00 | FILENAME……|ETX|

XX : DATA-BYTECOUNT (number of following bytes excluding ETX)

FILE_SIZE_DWORD: size of file to be copied to the laser (byte switched)

OPTIONCOMMANDBYTE: optional TCPIP command executed directly after having copied the file to the laser. Allowed commands are:

- 0x00: no additional command
- 0x58 (TCP_RELOAD); reloads the actual message.
- 0x50 (TCP_ASCIICONFIG): the sent file is the ASCII configuration file and should be loaded into memory.
- 0x49 (TCP_PARTIALASCIICONFIG): the sent file contains some configuration parameters that should be loaded into memory.
- 0x55 (TCP_CONFIG): the sent file is the binary configuration file and should be loaded into memory.

OPTIONBYTE: a byte that defines where to copy the file to.

0x0F: file is copied to the RAMdisk and to the harddisk
0x00: file is copied to the RAMdisk only

FILENAME…: up to 40 bytes specifying the filename under which the data should be saved inside the laser. The filename should be composed of ASCII characters and must include an extension ! If the filename length is not a multiple of 4 you must add a termination NULL character (the byte '\0') !

Answer:

If the laser has received the command it will respond with

|STX|0x02|0x61 0x00|ETX|

Then you have to send the content of file in frames 2048 databytes to the laser. No STX and no ETX must be packed into the data. The data frame just contains the sequential binary data of the file.
Each time the laser has received 2048 bytes it will answer with a Framecommand:

|STX|0x06|0x81 0x00|FRAME-DWORD|ETX

Where FRAME-DWORD is the byte switched received block number. The first frame received has the block number "1", the second frame the block number "2" ,etc..

If an error occures the laser will not respond and you should stop sending frames and send the final command to obtain the error code.

The last frame is the only frame that may have less than 2048 bytes.

After having sent all frames to the laser and after having received the frame command for each sent frame from the laser you have to send a final command to the laser to obtain the error code of the copy command:

Command:
|STX|0x06|0x81 0x00| 0x00 0x00 0x00 0x00|ETX

The laser then answers with

Answer:
|STX|0x06|0x81 0x00| ERROR_CODE_DWORD|ETX

where the ERROR_CODE_DWORD is a byte switched error code.

Error code meaning:

0: no error
8: file could not be opened for writing into the RAMdisk
16: not temporal file could be created on the harddisk
32: temporal harddisk file could not be renamed to the destination filename
1: no memory available to accept frames,no memory available in RAMdisk or an internal socket error has occurred during reception of a frame.
2: timeout during reception of a frame (waiting for more than 4 seconds)

Copy a file from the laser:

CMD: 0x0061

|STX|XX|0x61 0x00|0x00 0x00 0x00 0x00|0x00 OPTIONBYTE 0x00 0x00 |FILENAME……|ETX|

XX : DATA-BYTECOUNT (number of following bytes excluding ETX)

OPTIONBYTE: a byte that defines from where to copy the file.

0xF0: file is copied from the harddisk
0xFF: file is copied from the RAMdisk

FILENAME…: up to 40 bytes specifying the filename.

If the filename length is not a multiple of 4 you must add a termination NULL character (the byte '\0') !

Answer:

If the laser has received the command it will respond with

|STX|0x06|0x61 0x00|FILE_SIZE_DWORD|ETX|

FILE_SIZE_DWORD: size of file to be copied from the laser (byte switched)

From now on, you have to request frame by frame of maximum 2048 bytes from the laser. The first frame has the FRAME-DWORD "0", the next "1", etc…

Framecommand:

|STX|0x06|0x81 0x00|FRAME-DWORD|ETX

Answer:

The laser sends frames of 2048 with the sequential data bytes of the file.
Only the last frame may have less then 2048 bytes.

After having received all <FILE_SIZE_DWORDS> bytes you have to send a final framecommand to end the command sequence.
The final framecommand:

|STX|0x06|0x81 0x00|FRAME-DWORD|ETX where FRAME-DWORD is the number of the received frames (FRAME-DWORD increases by '1' each time a frame has been received, even for the last received frame).

Delete a file:

Files are deleted always in the RAMdisk as well as in the harddisk. There is no specific command to delete selectively files only in RAM or harddisk.

CMD: 0x0037

The sequence must be:

|STX|XX|0x37 0x00|FILENAME-bytes….|ETX|

XX : DATA-BYTECOUNT (number of following bytes excluding ETX)
FILENAME-bytes: the filename of the file to be deleted (up to 48 bytes)

If the filename length is not a multiple of 4 you must add a termination NULL character (the byte '\0') !

Answer:

|STX|0x06|0x37 0x00|error-DWORD|ETX|
Error-DWORD: 0x00000000 file was deleted
              0x00000001 file not found

Reading some digital Input/encoder signals:

Using this command blocks the laser for 100 milliseconds, thus, this command is just for test purposes and should not be used during marking.

CMD: 0x0072

The sequence must be:

|STX|0x06|0x72 0x00|0x00 0x00 0x00 0x00|ETX|

Answer:

|STX|XX|0x72 0x00|DWORD0| DWORD1| DWORD2| DWORD3| DWORD4| DWORD5| DWORD6| DWORD7|DWORD8|DWORD9|DWORD10|DWORD11|ETX|

All DWORDs are byte switched.

XX : DATA-BYTECOUNT (number of following bytes excluding ETX)

DWORD8 -DWORD11 are sent only for (SM170/270 DSPCARD).

DWORD0: internal use

DWORD1: internal use

DWORD2: Digital Input

(SM121)
Bit0: photocell
Bit4: interlock
Bit5: PLC start mark
Bit23 – Bit31: external bits at customer connector

(SM120)
Bit8: photocell
Bit13: PLC start mark
Bit14: interlock
Bit16 – Bit21: external bits at customer connector

(SM140)
Bit0: photocell
Bit1: interlock

(SM108)
Bit0: PLC start mark
Bit1: interlock
Bit15: photocell

(SM170/270 DSPCARD)

Bit0: state of INP6
Bit1: state of INP7
Bit2: state of INP8
Bit3: state of INP9
Bit4: state of INP10
Bit5: state of INP11
Bit6: state of INP12
Bit7: state of INP13
Bit8: state of INP14
Bit9: state of INP15
Bit10: state of INP13

```
Bit11: state of INP28 (bit0 customer)
Bit12: state of INP29 (bit1 customer)
Bit13: state of INP30 (bit2 customer)
Bit14: state of INP31 (bit3 customer)
Bit15: state of INP32 (bit4 customer)
Bit16: state of INP33 (bit5 customer)
Bit17: state of INP34 (bit6 customer)
Bit18: state of INP35 (bit7 customer)
Bit19: state of INP36 (bit8 customer)
Bit20: state of INP37 (bit9 customer)
Bit21: state of INP38
Bit22: state of INP39 (bit11 customer)
Bit23: state of INP1 (photocell)
Bit24: state of INP2 (PLC start mark)
Bit25: state of INP3 (Interlock input)
Bit26: state of INP4 (Shutter input)
```

```
DWORD3: internal use

DWORD4: encoder pulses; usually a value between 0 and 65535, counting always in
           positive direction

            (SM170/270 DSPCARD)

        encoder pulses as signed 32-bit integer, counts in both directions.

DWORD5: encoder direction; values 0x00000000 or 0x00000001

      (SM170/270 DSPCARD)
       internal use


DWORD6: Photocell state ;

      Values:
      0x000000000 or 0x00000001 (SM121, SM140)
      0x000000000 or 0x00000080 (SM120)
      0x000000000 or 0x00008000 (SM108)
      0x000000000 or 0x00000001 (SM121)
      0x000000000 or 0x00000001 (SM117)

      (SM170/270 DSPCARD)
       internal use



DWORD7: encoder velocity in mm/sec

DWORD8-DWORD11: (SM170/270 DSPCARD)
       internal use
```

Power/speed scaling or bitmap pixeltime scaling (from firmware 4.1.0 on)

With this command you can apply a scaling-factor to the power or speed properties of the layers, or to the bitmap properties “pixeltime” and “pixelpower”. The scaling factors are valid for all messages until a reboot of the machine or until a new scaling factor is sent to the laser. The scaling factors are applied always to the original values of the messages !

CMD: 0x0076

The sequence must be:

|STX|0x0E|0x76 0x00|SET-DWORD | MEMBER-DWORD| VALUE-DWORD |ETX|

All DWORDs are byte-switched.

Answer:

|STX|0x0E|0x76 0x00|Error-DWORD| MEMBER-DWORD | DATA-DWORD |ETX|

All DWORDs are byte-switched.

SET-DWORD:	0x00000000: sets a new scaling factor 0x00000001: gets am actual scaling factor
MEMBER-DWORD:	0x00000000: set or get the pixel time scaling (for bitmaps or 2D codes) 0x00000001: set or get the pixel power scaling (for bitmaps or 2D codes and only for YAG and Fiber laser) 0x00000002: set or get the layer's marking speed 0x00000003: set or get the layer's marking power
VALUE-DWORD:	the scaling factor to be set. The scaling factor is given in [permille] . A numerical value of 1000 means a scaling factor of 1.000, a value of 500 means a scaling factor of 0.500. Note that the scaling factor applies to the specified member and the marking result depends on the member that you change. E.g. a scaling value of 500 applied to the pixel time decreases the time per pixel of a bitmap (decreases the time of marking), while, applied to the layer speed, it decreases marking speed (thus *increasing* the marking time !). If the command is a "get" command (SET-DWORD==0x00000001), the value of VALUE-DWORD is ignored.
DATA-DWORD:	the requested scaling factor.
ERROR-DWORD:	0x00000000 no error 0x00000001 invalid VALUE-DWORD (< 0) 0x00000002 invalid MEMBER-DWORD

Example: Decrease the bitmap pixeltime to 80 percent of the original value.

SETORGET = 0x00

MEMBER = 0x00

VALUE (800) in HEX = 0x0320

VALUE(upper byte) = 0x03 (don't forget ESC sequence !)

VALUE(lower byte)=0x20

Thus the command is

STX ADDR 0x76 0x00 0x00 0x1B 0x03 0x20 CRC 0x03

Set an X/Y-offset for the message to be printed:

With this command you can shift the complete message in x and y in the scanfield. You can specifiy the units that you wish to use and if the offset should be absolute or relative to a previous offset. You can also decide if the offset should be automatically resetted after the next print, or if it should remain. The Offset is always applied to the next print and if you send this command during an actual print, it will be applied in the next print.

CMD: 0x0059

The sequence must be:

|STX|0x16|0x59 0x00| DWORD0| DWORD1| DWORD2| DWORD3| DWORD4| ETX|

All DWORDs are byte switched.

DWORD0: absolute (value = 0) or relative offset (value= 1)

DWORD1: X offset value

DWORD2: Y offset value

DWORD3: defines the units for the XY offset:

0X00000000:
Ideal coordinates; the scanfield has coordinates ranging from [0 – 100 000]. For a 100mm x 100mm lens , ideal coordinates map to 1 micron.

0X00000001:
Coordinates are given in microns.

0x00000002:
Coordinates are given in 1/10 mm.

DWORD4: defines a reset of the XY-offset after its use.

0x00000000:	XY-offset is remained until a new command is send or the configuration of the laser is manually changed.
0x00000001:	XY-offset is resetted to ZERO after the next print (so this offset will only be valid for the next print)

Answer:

|STX|0x0a|0x59 0x00|DWORD0| DWORD1| ETX|

DWORD0: new absolute X offset value (using the same units as set by the command)

DWORD1: new absolute Y offset value (using the same units as set by the command)

Set a Z-defocus or Z-position value:

With this command you can apply a z-defocus shift to the laser beam, if the system is equipped with a dynamic z-axis.

The sent value can be considered as an absolute value or a relative shift to any previous sent value. To get the actual absolute z-defocus value, you can simply set a relative shift of 0.

For firmware versions > 5.6.4 (version number > 88) you can also set/get a global Z-position.

CMD: 0x0066

The sequence must be:

|STX|0x0e |0x66 0x00| DWORD0| DWORD1| DWORD2| ETX|

All DWORDs are byte switched.

DWORD0:
absolute (value = 0) or relative offset (value= 1) of the z-defocus
absolute (value = 2) or relative offset (value= 3) of the z-position

DWORD1: Z defocus of Z position value

DWORD2: defines the units for the z defocus/position value:

0X00000000:
Ideal coordinates; the scanfield has coordinates ranging from [0 – 100 000]. For a 100mm x 100mm lens , ideal coordinates map to 1 micron.

0X00000001:
Coordinates are given in microns.

0x00000002:
Coordinates are given in 1/10 mm.

Answer:

|STX|0x06|0x66 0x00|DWORD0| ETX|

DWORD0: new absolute z defocus/position value (using the same units as set by the command)

Shift and rotate a loaded message:

With this command you can shift the actual loaded message in x and y and rotate it.

CMD: 0x0165

The sequence must be:

|STX|0x04|0x65 0x01| Data-bytecount-WORD | TRANSFORMATION-STRING| ETX|

TRANSFORMATION-STRING: n bytes (Data-bytecount-WORD) with the following format:

x=<value>y=<value>p=<value>x0=<value>y0=<value> [optionally l=<layer-name> or o=<object-name>]

where <value> stands for a numerical value.

x: an offset in x-direction in units of [mm] (default 0)

y: an offset in y-direction in units of [mm] (default 0)

p: a rotation angle in degrees (default 0)

x0: the x-coordinate of the rotation center in units of [mm] (default : center of the scanfield)

y0: the y-coordinate of the rotation center in units of [mm] (default : center of the scanfield)

The order of occurence of each token does not mind. If a token is missed its default value is assumed.

l: defines a layer's name. Only objects of the layer with this name are transformed.

o: defines an object's name. Only the object with this name is transformed.

The 'l' and 'o' options are valid only for firmware versions >= 5.3.2 from 15/02/2014 on. If 'l' or 'o' options are omitted the whole document is shifted/rotated.

The optional <layer name> or <object name> must be utf8-coded strings. If you use only ASCII charactes for object-names or layer-names, the utf8-coding is the same as ASCII coding. Only one of the options l=<layer-name> or o=<object-name> is allowed ! If the l=<layer-name> or o=>object-name> option is not the last token in the TRANSFORMATION-STRING, then it must be separated from the following token by a SPACE character. Layernames or objectnames must not contain SPACE characters.

Examples:

shifting in x by +5.5 mm and in y by -3mm would result in:

x=5.5y=-3

shifting in x by +3.8mm and in y by -5.2mm and rotation of 45 degrees around (-70 mm, -50 mm):

x=3.8y=-5.2p=45x0=-70y0=50

Answer:

|STX|0x04|0x65 0x01|01 00|ACK|ETX| command correctly accepted.

Or

|STX|0x02|0x65 0x01|01 00|NACK|ETX| command accepted but transformation cannot be applied due to a pending reload of the message.

Activate/deactivate the internal eventhandler: (firmware >= 5.1.0)

CMD: 0x0030

The sequence must be:

|STX|XX|0x30 0x00|SUBCOMMAND-DWORD|FILENAME-bytes….|ETX|

XX : DATA-BYTECOUNT (number of following bytes excluding ETX)
SUBCOMMAND-DWORD: 0x00000000 (must have this value, other values are currently not supported)

From firmware 5.1.3 Version 05/03/2011 on the SUBCOMMAND can be:

0x00000000	(activate handler and add appropriate parameter to the commandline of the firmware in the harddisk)
0x00000001	(activate handler but do not touch the commandline)
0x00000002	(deactivate handler and add appropriate parameter to the commandline of the firmware in the harddisk)
0x00000003	(deactivate handler but do not touch the commandline)

FILENAME-bytes: the filename of the eventhandler file to be loaded (up to 48 bytes with extension)

If the filename length is not a multiple of 4 you must add a termination NULL character (the byte '\0') !

Note: SUBCOMMAND_DWORDS must be byte switched for sending !

Answer:

|STX|0x02|0x30 0x00|ETX| command correctly accepted.

Or

|STX|0x02|0x15 0x00|ETX| SUBCOMMAND-DWORD not supported.

Note: If the filename is empty or an invalid filename (not existing file), the eventhandler will be deactivated.

The command itself does not give any feedback if the handler was loaded correctly.

The eventhandler file must reside inside the RAM- or harddisk of the laser. The laser first tries to load the file from the RAMdisk and in case that this was not successful it tries to load the file from the harddisk.

Set/get dynamic velocity/distance and/or change the dynamic mode

CMD: 0x0021

The sequence must be:

|STX|XX|0x21 0x00|SUBCOMMAND-DWORD|optional VARIABLE-DWORD|optional VALUE-DWORD|
ETX|

XX : DATA-BYTECOUNT (number of following bytes excluding ETX)

SUBCOMMAND-DWORD:

0x00000010 : sets a value to a dynamic variable

0x00000020 : gets a value of a dynamic variable

0x00000008 : request actual dynamic mode

0x00000000 : set mode to static mode

0x00000001 : set mode to dynamic mode

0x00000002 : set mode to dynamic-distance mode

0x00000003 : set mode to dynamic-static mode

VARIABLE-DWORD:

Optional parameter to be set when SUBCOMMAND is 0x8,0x10 or 0x20.

0 : printdistance will be set/get

1: internal encoder velocity will be set/get

VALUE-DWORD:

Optional parameter when SUBCOMMAND is 0x10.

The value of the variable when it is set (subcommand = 0x10).

The value must be given in microns.

Answer:

|STX|XX|0x21 0x00| optional VALUE-DWORD|ETX| command correctly accepted.

XX : DATA-BYTECOUNT (number of following bytes excluding ETX)

VALUE-DWORD:

The value of a requested variable or the value of the actual mode when it is requested or set.

|STX| 0x02|0x15 0x00|ETX| some error in sent command (wrong variable or subcommand)

Turn on/off the red diode pointer (firmware > 5.0.7 required)

CMD: 0x0060

The sequence must be:

|STX|XX|0x60 0x00|SUBCOMMAND-DWORD| DWORD2|DWORD3|DWORD4|DWORD5| ETX|

XX : DATA-BYTECOUNT (number of following bytes excluding ETX)

SUBCOMMAND-DWORD:

0x00000016 :

Turns on the red pointer with actual enclosing rectangle of the actual loaded message. The optional DWORD2 – DWORD5 should not be sent.

0x00000020 :

Turns on the red pointer with the enclosing rectangle coordinates sent with the command in DWORD2 – DWORD5. The optional DWORD2 – DWORD5 should be sent with the command and define the rectangle that the pointer should mark.

0x00000030:

Turns on the red pointer making a cross at the position defined by the configuration (focal pointer position).

0x00000200 : turns off the red pointer

Note: SUBCOMMAND_DWORD must be byte switched for sending !

DWORD2 – DWORD5: optional

DWORD2: left position of rectangle (ideal coordinates [-50000,50000])

DWORD3: top position of rectangle (ideal coordinates [-50000,50000])

DWORD4: right position of rectangle (ideal coordinates [-50000,50000])

DWORD5: bottom position of rectangle (ideal coordinates [-50000,50000])

Answer:

|STX|0x02|0x60 0x00| ETX| command correctly accepted.

|STX|0x06|0x60 0x00|0x15 0x00 0x00 0x00| ETX| command not accepted (busy)

Turn on/off the laser

CMD: 0x0060

The sequence must be:

|STX|XX|0x60 0x00|SUBCOMMAND-DWORD| DWORD2|DWORD3|DWORD4|DWORD5| ETX|

XX : DATA-BYTECOUNT (number of following bytes excluding ETX)

SUBCOMMAND-DWORD:

0x00000001 :

Turns the laser on.

0x00000000 :

Turns the laser off.

Note: SUBCOMMAND_DWORD must be byte switched for sending !
This command tries to open the shutter and activates the laser directly. When used to turn off you should check the answer in any case (if the system was busy the alser will not turn off). As alternative you could use the 'PrintStop' command 0x2e which turns the laser off via interrupt.

DWORD2 – DWORD5: optional.

DWORD2: power of the laser in percent [0 - 100]
DWORD3: frequency of the laser in kHz [4 - 1000]
DWORD4: x coordinate of the scanner (ideal coordinates [-50000,50000])
DWORD5: y coordinate of the scanner (ideal coordinates [-50000,50000])

Answer:

|STX|0x02|0x60 0x00| ETX| command correctly accepted.

|STX|0x06|0x60 0x00|0x15 0x00 0x00 0x00| ETX| command not accepted (busy)

Switch mode between default/message table/batchjob

CMD: 0x003A

The sequence must be:

|STX|0x06|0x3A 0x00|SUBCOMMAND-DWORD| ETX|

SUBCOMMAND-DWORD:

0x00000000 : enable standard mode (disabled message table and batchjob)
0x00000001 : enable external messagetable
0x00000004 : enable batch job mode
0x00000002 : request actual mode

Note: SUBCOMMAND_DWORD must be byte switched for sending !

Answer:

|STX|0x06|0x3A 0x00| SUBCOMMAND_DWORD | ETX| command correctly accepted.

SUBCOMMAND-DWORD:

0x00000000 : standard mode enabled

0x00000001 : external message table enabled

0x00000004 : batch job mode enabled

Actualize the message table(batch job and/or external message table, version >=5.2.2 required)

Command used to force the laser to read the actual messagetable file ("table.cnf" in laser's harddisk). The messagetable file is an ASCII file with the entries in the following format:

<pos>: <repeat prints> <name>
..
..
..

where <pos> stands for the position (0 - 255) , <repeat prints> stands for the number of prints of the file in case of the batch job mode, and <name> stands for the file to be printed (filename with extension (msf or xml)).
The fields must be separated by a single space !

Example with 3 positions:

0: 1 moert.xml
5: 1 test.msf
7: 1 testfile.xml

CMD: 0x0079

The sequence must be:

|STX|0x02|0x79 0x00| ETX|

Answer:

|STX|0x02|0x79 0x00| ETX| command correctly accepted.

Note: this command does not change the mode (standard, external messagetable or batchjob !)

Store the sent usermessages/global counters in the harddisk (version >=5.2.5 required)

Command used to force the laser to save all actual usermessage fields or global counters in the harddisk.

The sequence must be:

|STX|0x06|0x78 0x00| Subcommand-DWORD | ETX|

Answer:

|STX|0x02|0x78 0x00| Subcommand-DWORD | ETX| command correctly accepted.

Subcommand-DWORD: defines what to be saved with a bitwise logic.

Bit0 set: usermessages are stored

Bit1 set: global counters are stored

Bit2 set: last filename is stored

Get a frame of filenames with the supplied extension

Command used to get frames with the filenames of file in the RAMdisk with the supplied extension.

The sequence must be:

|STX|0x04|0x26 0x01|Data-bytecount-WORD | TYPE-BYTE | FRAME-BYTE | EXTENSION-STRING| ETX|

TYPE-BYTE: must be 0x00

FRAME-BYTE: 0x00 – 0xFF , defines the frame to be requested. Filenames are sent in frames of up to 25 filenames, each one separated by a LF (0x0a).

EXTENSION-STRING: the extension of the files whose name are requested (do not include the '.')

Answer:

|STX|0x04|0x26 0x01| Data-bytecount-WORD | FILENAMES| ETX|

FILENAMES: up to 25 filenames with the requested extension separated by a LF character.

Note: to get the first 25 filenames set the frame byte to 0x00, then increment the frame-byte by one and get the next 25 filenames until less than 25 filenames are received within the FILENAMES.

Get a preview file in svg-format

Command used to create a svg-file in the laser's RAMDISK with the actual message data drawn. (firmware >= 5.2.7 07/02/2013)

|STX|XX|0x85 0x00| Byte0 - Byte15|DWORD5|Byte20-Byte48|ETX|

XX : DATA-BYTECOUNT (number of following bytes excluding ETX)

Byte0 - Byte15: optional, up to 16 bytes for the filename to be created. If the filename is less than 16 bytes a terminating NULL-Byte must be added.

If the filename is omitted or NULL the created file in RAMDISK will be named "dump.svg".

DWORD5: optional for version > 5.6.3; you can filter the svg output with this parameter

0: no filter is applied (same as when FILTER is omitted)

1: the svg output is filtered by the layername defined by DWORD6-DWORD12

2: the svg output is filtered according the layer's signalmasks

and the actual signalstate of the laser. Only objcts of layers whose signalmask settings are such that the layer will be printed are output into the svg file.

3: the svg output is filtered according the last latched signalstate. This signalstate is latched at each start of a print.

Byte20-Byte48: optional for version > 5.7.3; may define the layername when DWORD5 is set to 1.

If less than 28 bytes are used then you have to add a termination character 0x00.

Only objects of the layer defined by the layername will be output into the svg file.

Answer:

|STX|0x02|0x85 0x00| ETX| if no error has occurred.

|STX|0x02|0x15 0x00| ETX| the svg could not be created.

Set the internal status counters (d_counter,s_counter, t_counter)

CMD: 0x0042

The sequence must be:

|STX|0x0a|0x42 0x00|DWORD1| ETX|

DWORD1:

0x7FFFFFFF : this will reset the d_counter, s_counter, t_counter to zero

In all other cases the DWORD1 will determine the new value of the d_counter.

Answer:

|STX|0x0a|0x42 0x00| DWORD1 | DWORD2 | ETX| in case that the sent

DWORD1 was not 0x7FFFFFFF.
The DWORD1 in the answer is the d_counter before it was changed with this command. The DWORD2 in the answer is the s_counter before it was changed with this command.

|STX|0x0E|0x42 0x00| DWORD1 | DWORD2 | DWORD3 |ETX| in case that the sent

DWORD1 was 0x7FFFFFFF. The DWORD1 in the answer is the d_counter before it was changed with this command. The DWORD2 in the answer is the s_counter before it was changed with this command. The DWORD3 in the answer is the t_counter before it was changed with this command.

Get/set the internal barcode license

CMD: 0x0073

The sequence must be:

|STX|XX|0x73 0x00|DWORD1|DWORD2|.....|DWORD9| ETX|

XX : DATA-BYTECOUNT (number of following bytes excluding ETX)

DWORD1: 0x00000000 : get the barcode license status

0x00000001 : send the barcode license key

The lower byte is sent first (byte-switched).

DWORD2 – DWORD 9:

the 8-bytes of the supplied license key. Each byte is coded in one DWORD. Thus, each DWORD has its upper 3-bytes set to 0x00 and lowest byte is set to the corresponding byte of the license key.

The lower byte is sent first (byte-switched).

Example: supplied license key = “FA01320000FF78ED”

DWORD2 = FA DWORD 3 = 01 …..DWORD9 = ED

Answer:

|STX|XX|0x73 0x00| DWORD1 | DWORD2 |....DWORD8 ETX|

XX : DATA-BYTECOUNT (number of following bytes excluding ETX)

DWORD1 : 0x00000000 if no barcode license is present/valid
0x00000001   if the barcode license is present/valid

DWORD2...DWORD7:
The 6 bytes of the MAC-address of the laser, each byte coded within a single DWORD (byte-switched).
DWORD 8: should be ignored

Note: the 6-byte MAC address is the code that you have to submit to obtain a barcode license code.

USB-drive command

```
CMD: 0x0187

The sequence must be:

|STX|0x04|0x87 0x01| Data-bytecount-WORD | SUBCMD-BYTE| DRIVE-BYTE | ETX|
```

SUBCMD_BYTE:

0x00 : get USB drive status;

DRIVE-BYTE is ignored and no action on the USB-drive is performed.

0x01: store configuration

The sysvars.cnf and the oemvars.cnf are stored in the USB-drive according the DRIVE-BYTE. The files are store in the ./config directory of the USB-drive. If the directory does not exist it will be created.

0x02: store messages

All 'msf', 'xml' and 'db' files are stored in the USB-drive according the DRIVE-BYTE. The files are store in the ./messages directory of the USB-drive. If the directory does not exist it will be created.

0x03: store fonts

All 'mfs', 'mft', 'unx' and 'unf' files are stored in the USB-drive according the DRIVE-BYTE. The files are store in the ./fonts directory of the USB-drive. If the directory does not exist it will be created.

0x04: store truetype fonts

All 'ttf', 'otf', and 'ttc' files are stored in the USB-drive according the DRIVE-BYTE. The files are store in the ./fonts directory of the USB-drive. If the directory does not exist it will be created.

0x05: store backup

All files of the laser's harddisk are stored in the USB-drive according the DRIVE-BYTE. The files are store in the ./backup directory of the USB-drive. If the directory does not exist it will be created.

0x06: load configuration

The files sysvars.cnf and the oemvars.cnf present in the ./config directory of the USB-drive according the DRIVE-BYTE are copied to the laser's harddisk.

0x07: load messages

All 'msf', 'xml' and 'db files present in the ./messages directory of the USB-drive according the DRIVE-BYTE are copied to the laser's harddisk.

0x08: load fonts

All 'mfs', 'mft', 'unx' and 'unf' files present in the ./messages directory of the USB-drive according the DRIVE-BYTE are copied to the laser's harddisk.

0x09: load truetype fonts

All 'ttf', 'otf', and 'ttc' files present in the ./fonts directory of the USB-drive according the DRIVE-BYTE are copied to the laser's harddisk.

0x0a: load backup

All files present in the ./backup directory of the USB-drive according the DRIVE-BYTE are copied to the laser's harddisk.

DRIVE_BYTE:

0xXX: Indicates the USB-drive to read/write (0,1,2). Usually the value is 0x00 as long as only one USB-drive is connected to the laser (there is only one external USB-connector available, but internally two more USB-pens could be connected)

Answer:

|STX|0x04|0x87 0x01|Data-bytecount-WORD|BYTE1|BYTE2|BYTE3|BYTE4|BYTE5....|ETX|

```
      BYTE1 : total number of actually mounted USB-drives (0,1,2,3)
      BYTE2 : number of USB-drive to read/write (0,1,2)
      BYTE3 : busy indicator,
                                    0x00  USB not busy
                                    0x01  USB is busy

      BYTE4: last USB error         0x00  no error
                                    0xff  USB read/write error

      BYTE5: state of USB-drive 0
                                    0x00  not mounted
                                    0x01  mounted

      BYTE5: state of USB-drive 1
                                    0x00  not mounted
                                    0x01  mounted

      BYTE5: state of USB-drive 2
                                    0x00  not mounted
                                    0x01  mounted
```

All USB-drive commands are queued inside the laser. So you have to send another USB-drive command with the SUBCMD = 0x00 (status) to see if the USB-process is still busy or not. Otherwise, turning off the system or removing the USB-pen may result in a corrupted harddrive or USB-pen.

Get core parameters

Command used to get some core parameters of the scanning board, like the CPU coretemperature, board temperature, humidity, etc...(parameters depend on the used scanning board).

CMD: 0x009B

The sequence must be:

|STX|0x02|0x9b 0x00|ETX|

Answer:

|STX|XX|0x9b 0x00| DWORD1 | DWORD2 |.... ETX|

XX : DATA-BYTECOUNT (number of following bytes excluding ETX)

The number of received DWORDS may increment in future versions. Actually 5 DWORD are received. All DWORDS are byte-switched.

DWORD1 : CPU core temperature in units of 1/1000 degrees Celsius.
(INT_MIN if not available)
DWORD2 : Scanning board temperature in units of 1/1000 degrees Celsius.
(INT_MIN if not available)
DWORD3 : Humidity in units of 1/1000 %.
(INT_MIN or -1 if not available)
DWORD4 : 5V DC-voltage at the scanning board in units of .1/1000 Volts.
(INT_MIN or -1 if not available)
DWORD5 : 3.3V DC-voltage at the scanning board in units of .1/1000 Volts.
(INT_MIN or -1 if not available)
DWORD6 : fancontroller: local temperature sensor in Celsius
(INT_MIN if not available)
DWORD7 : fancontroller: current PWM of the fan controller in percent
(INT_MIN if not available)
DWORD8 : fancontroller: fan tacho in cps
(INT_MIN if not available)
DWORD9 : fancontroller: remote temperature sensor in Celsius
(INT_MIN if not available)

Get system information

Command used to get some system information like available harddisk space.

CMD: 0x009A

The sequence must be:

|STX|0x02|0x9a 0x00|ETX|

Answer:

|STX|XX|0x9a 0x00| DWORD1 | DWORD2 |.... ETX|

XX : DATA-BYTECOUNT (number of following bytes excluding ETX)

The number of received DWORDS may increment in future versions. Actually 5 DWORD are received. All DWORDS are byte-switched.

DWORD1 : CPU core temperature in units of 1/1000 degrees Celsius.
(INT_MIN if not available)
DWORD2 : total size of message space in the harddisk in Bytes.

DWORD3 : free size of message space in the harddisk in Bytes.

DWORD4 : total size of message space in the ramdisk in Bytes.

DWORD5 : free size of message space in the ramdisk in Bytes.

DWORD6 : total size of font space in the ramdisk in Bytes.

DWORD7 : free size of font space in the ramdisk in Bytes.

DWORD8 : total size of log space in the harddisk in Bytes.

DWORD9 : (working hours) stored as a (float) variable in 1/10 hours.

DWORD10 : upper DWORD of the total 64-bit print counter.

DWORD11 : lower DWORD of the total 64-bit print counter.

Set software signalstate

Command used to set the signalstate via software instead of using the hardware IO signalstate. Each time the laser starts to print the signalstate is latched and can be used to select specific layers to be printable or not printable. The layers has to be configured correspondingly for this purpose (see GUI help). With this command you can force the laser to use the sent 'signalstate' as its internal signalstate to determine which layers are printable/not printable. The hardware signalstate is a 32 bit register with the following meaning:

bit31,bit30,.....photocell2(bit18), photocell(bit17), PLC input(bit16),bit15, bit,14,.....bit9,bit8,external selection7,..external selection0.

Thus, with some external IO signals one can control which layer is to be printed, when the layer's settings have been adjusted accordingly.
When you send a 'software-signalstate' the 32-bit value that is sent will act as the new signalstate and the hardware signalstate will be disabled.

Available from version 5.6.4 on.

CMD: 0x009C

The sequence must be:

|STX|0x0a|0x9c 0x00|DWORD1|DWORD2|ETX|

DWORD1: 0 for getting the actual software signalstate
1 for setting the actual software signalstate
2 for disabling the software signalstate

DWORD2: the 32-bit signalstate to be set. In case of disabling or getting the actual signalstate this value should be set to 0.

Answer:

|STX|0x0a|0x9c 0x00|DWORD1|DWORD2|ETX|

DWORD1: 0 if software-signalstate is enabled
1 if software-signalstate is disabled

DWORD2: the actual software-signalstate

UV optowave laser command

Command used to get some information from the Optowave UV laser.

CMD: 0x0189

The sequence must be:

|STX|0x04|0x89 0x01|Data-bytecount-WORD| CMD BYTE| SEND BYTE| Datastring bytes | ETX|

Answer:

|STX|0x04|0x89 0x01| Data-bytecount-WORD|Datastring bytes... ETX|

The Datastring bytes form an ASCII-string with tokens separated by a ‘,’. the meaning of each token depends on the CMD BYTE.

SEND BYTE: When set to 1 then the Datastring bytes are sent out via the RS232 communication port to the internal laser controller. The internal controller may respond later and update some internal variables that may be requested with subsequent calls of this command with the CMD byte set to a value that is not ‘’0x00’ and not ‘0x81’ and the SEND byte set to ‘0’ or to ‘1’ if you want to do another request.

Datastring: Defines an ASCII string that is sent out via RS232. This string is only used with ‘set’ commands. The ‘request’ commands described in this documentation do not require and datastring to be sent.

CMD BYTE:

0x00: requests the used internal serial port number that communicates with the laser controller. The SEND byte should be 0. Sent datastring should be empty.

Received datastring bytes:

1. token: number of the internal port number
2. token: number of received port frames.

For all other CMD bytes the received datastring bytes are as follows:

1. token: diode current in 1/10 of Amperes.
2. token: the set current in 1/10 of Amperes.
3. token: the max. current in 1/10 of Amperes.
4. token: power level in 1/10 of %.
5. token: the set power level in 1/10 of %.
6. token: the internal control state.
7. token: the status of the controller.
8. token: the actual running phase.
9. token: the actual mode.
10. token: the set mode.
11. token: the PTR mode.
12. token: the set PTR mode.
13. token: the last environment temperature in 1/100 degrees.
14. token: the last head temperature in 1/100 degrees.
15. token: the last diode temperature in 1/10000 degrees.

0x81: resets the internal communication with the laser controller. SEND byte should be ‘0’.

0x16: request an actualization of the environment temperature. Set the SEND byte to ‘1’.

0x17: request an actualization of the head temperature. Set the SEND byte to ‘1’.

0x18: request an actualization of the diode temperature. Set the SEND byte to ‘1’.

Notes: Some parameters are not explained in detail here as their usage is out the scope of this document. The tokens 13,14,15 represent the last values that have been seen by the controller board. These values are not updated automatically. If you want to have an actualized value you need to set the CMD byte to the appropiate value and set the SEND byte to ‘1’, such that the main contoller requests the parameter internally from the laser controller. You should then wait at least for about 200 ms until you issue the next command with a SEND byte set to ‘0’ for reading back the actualized values.
If the SEND byte is set to ‘0’ (no request for updating some variable), then you can use any command byte except the ‘0x00’ and ‘0x81’ to retrieve the datastring tokens as decribed above.

MOPA fiber laser command

Command used to get some information from the MOPA fiber laser.

CMD: 0x0188

The sequence must be:

|STX|0x04|0x88 0x01|Data-bytecount-WORD| CMD BYTE| SEND BYTE| Datastring bytes | ETX|

Answer:

|STX|0x04|0x88 0x01| Data-bytecount-WORD|Datastring bytes... ETX|

The Datastring bytes form an ASCII-string with tokens separated by a ‘,’. the meaning of each token depends on the CMD BYTE.

SEND BYTE: When set to 1 then the Datastring bytes are sent out via the RS232 communication port to the internal laser controller. The internal controller may respond later and update some internal variables that may be requested with subsequent calls and the SEND byte set to ‘0’ or to ‘1’ if you want to do another request.

CMD BYTE:

0x00: requests the used internal serial port number that communicates with the laser controller. The SEND byte should be 0.

Received datastring bytes:

1. token: number of the internal port number
2. token: number of received port frames.

0x05: requests the temperature of the diode controller.

Received datastring bytes:

1. token: temperature in degrees Celsius

0x12: requests the pulse repetition rate in kHz.

Received datastring bytes:

1. token: Pulse repetition rate in kHz.

0x30: requests the pulse width in ns.

Received datastring bytes:

1. token: Pulse width in ns.

Maintenance command

Command used to get/set maintenance or efficiency information of the laser.

Efficiency data: there are 5 values regarding the efficiency that can be requested. All data are given in seconds.
1. The time interval within which the other values should be sampled.
2. The working time within this interval (actual consumed time).
3. The time that the system was in printing mode within the interval.
4. The time that the system was in alarm mode within the interval.
5. the time that the system was printing within this interval.

When the working time reaches the interval all entries, except the interval, will be reset to zero and a new cycle starts automatically.

The working time can be reset to zero with this command.
The interval can be set with this command.

Maintenance data: there are 6 values regarding the maintenance that can be requested. All data are given in seconds.

1. The time interval for lens cleaning.
2. The consumed time within this interval.
3. the time interval for filter changing.
4. The consumed time for this interval.
5. The time interval for general maintenance.
6. The consumed time for this interval.

All intervals can be set with this command and all consumed times can be reset.

Maintenance data are sampled and stored only when the MAINTENANCE parameter of the systemvariables is enabled.
With this command you will be informed if maintenance is enabled or not, but you cannot enable/disable it with this command.

CMD: 0x0089

The sequence must be:

|STX|XX|0x89 0x09|DWORD0|DWORD1|DWORD2|DWORD3|ETX|

DWORD0: 0 : set/get the maintenance data

1 : set/get the efficiency data

DWORD1: 0 : gets the data (no more DWORDs needs to be sent)

Bit0 set: resets a time interval.

Bit1 set: sets a new time interval

DWORD2: (optional)
Defines which time interval should be set or reset. In case of efficiency data this parameter will be ignored as there is all efficiency data share the same interval.

In case of maintenance data this parameter can have a value from 0 to 2.

0: lens cleaning interval
1: filter changing interval
2: general maintenance interval

DWORD3: (optional)
defines the new time interval [in seconds] in case that it will be set.

Answer:

|STX|XX|0x89 0x00| DWORD0 – DWORD6 |ETX|

The number of received DWORDs depends on the command.

When too less DWORDs have been sent (depends on the command), the system answers with returning just one DWORD with a value of 0x00000015 (NACK).
When the received DWORD0 was not set to 0 or 1 or the laser does not support the request, the laser answers with sending just

|STX|02|0x89 0x00||ETX|

In any other case the system responds with:

set/get/reset efficiency data:

|STX|XX|0x89 0x00| DWORD0 – DWORD4 |ETX|

DWORD0: the interval in seconds
DWORD1: the consumed time in seconds
DWORD2: the time the system was in printing mode
DWORD3: the time the system was in alarm mode
DWORD4: the time the system was printing.

set/get/reset maintenance data:

|STX|XX|0x89 0x00| DWORD0 – DWORD6 |ETX|

DWORD0: 0 if maintenance is disabled, not 0 if enabled.
DWORD1: the time interval of the lens cleaning.
DWORD2: the consumed time of the previous interval.
DWORD3: the time interval of the filter change.
DWORD4: the consumed time of the previous interval.
DWORD5: the time interval of the general maintenance.
DWORD6: the consumed time of the previous interval.

Scanning for lasersystems in your local network

If you do not know the IP address of the lasersystem you can scan your network with the UDP protocol.

Scanning for systems:

-use UDP protocol
-use the broadcast address 255.255.255.255 as the destination address
-use the destination port 7776
-send the characters “whoareyou?” to the destination port and destination address

The lasersystem , if present in your network, will respond to the sender with the following data:

“xxx.xxx.xxx.xxx-mmm.mmm.mmm.mmm-ggg.ggg.ggg.ggg-XXXXXX”

xxx.xxx.xxx.xxx	= actual IP-address
mmm.mmm.mmm.mmm	= actual netmask
ggg.ggg.ggg.ggg	= actual gateway
XXXXXX	= MAC address in hexadecimal

Changing the IP-address with the UDP protocol:

-use UDP protocol
-use the broadcast address 255.255.255.255 as the destination address
-use the destination port 7776
-send the characters “youarehow” followed by “xxx.xxx.xxx.xxx-mmm.mmm.mmm.mmm-ggg.ggg.ggg.ggg-XXXXXX” (the new settings) to the destination port and destination address. The lasersystem whose MAC address matches the XXXXXX of the sent data will change its IP settings to the sent data.
If no gateway address is used set the gateway address to 0.0.0.0.