

Actual list of RS232-commands (Scanlinux version 5.0.8 , excluding the commands for the touch screen editor):

The laser has two serial ports installed. Comport 2 is reserved for the Handheld terminal and cannot be used by any other device. Comport 1 is used by the optional touchscreen and can be used to connect other devices instead of the touchscreen. Only the RX , TX and GND lines are connected to the accessible DB-9 connector at the rear of the laser.

The default settings for Comport 1 are 9600, Non parity, 8 bits, 1 stopbit. The baudrate is the only parameter that can be changed externally by a command (see this document), but usually should be avoided.

The protocol that understands the laser is based on a pure master-slave protocol. The external device is always the master and the laser does not send anything to the outer world without being ordered to do so. The protocol structure is as follows:

Master (external): sends a command according to the command structure

Slave (laser): the laser responds to each command with exactly one answer.

Command structure:

STX ADDR [optional CRC SUPRESSBYTE] CMD DATABYTES CRC ETX

where

STX == 0x02

ADDR==0xFE (or any other value except 0x02, 0x03, 0x1B set within the configuration)

ETX==0x03

optional CRC SUPRESSBYTE==0xAA (or any other value except 0x02, 0x03, 0x1B set within the configuration), usually this byte is NOT SEND and it is only available for modern versions (refer to command 23. in this document).

CMD==the command byte (see available commands in this document)

DATABYTES==the data bytes to be sent which depend on the commandbyte !

CRC==the checksum byte, this byte MUST BE SENT, its value is calculated according to the following procedure:

CRC = the arithmetic sum of all bytes (except the ESC sequence bytes) between STX and CRC, masked with 0xFF (low byte of the sum) , that is:

$CRC = (ADDR + CMD + SUM(DATABYTES)) \& 0x000000FF;$

**IMPORTANT NOTE:**

ADDR and CMD cannot be set to 0x02(STX), 0x03(ETX) or 0x1B(ESCAPE). If the DATABYTES or the resulting CRC contain STX, ETX or ESC, an additional ESC-sequence (0x1B) must be preceded to the databyte or the CRC. Note, that the CRC MUST be calculated without these additional Esc-sequence bytes !!!

The laser usually responds with a similar command structure explained in detail for each command. If wrong data sequences are sent to the laser (e.g. unknown commands or commands with too many, or too less databytes, wrong checksums, noise on the line, etc.....), the laser usually answers with

0x02 ADDR 0x36 0x34 0x03

**Buffer overrun:**

The laser's serial port has usually a 16 bytes wide hardware input buffer. For some specific realtime actions the laser disables interrupts for short periods. This could lead to overrun conditions of the hardware input buffer if you send frames with more than 16 bytes lenght. In case of a buffer overrun, the laser will not accept the command and sends the following frame out of the serial port:

0x02 ADDR 0x36 0x15 0x49 0x03

We recommend to split the sending of RS232 commands with frames longer than 16 bytes into small junks of 16 bytes and include a delay of about 50 ms between the sending of the junks. This avoids possible overruns.

**1. Get messages that are stored in the laser:**

STX ADDR 0x26 NUMBER CRC ETX

NUMBER: 1 byte. The names of the messages are packed into frames of 10 messages. This byte defines the framenummer you request. The first framenummer is 0x00.

Answer: a. STX ADDR 0x26 CONTROL MESS CRC ETX

CONTROL: 1byte. Framenummer that was requested. If CONTROL is 0xFF then no more data can be requested (frame is empty).

MESS: up to 80 bytes. Name of 10 messages (each message is 8 bytes long, without extension).

**1 b. Get files of certain extension that are stored in the laser (ScanLinux >=5.0.8 required):**

STX ADDR 0x29 NUMBER EXTENSION CRC ETX

NUMBER: 1 byte. The names of the files are packed into frames of 10 . This byte defines the framenummer yourequest. The first framenummer is 0x00.

EXTENSION: n bytes: the requested file extension.

Answer: a. STX ADDR 0x29 CONTROL MESS CRC ETX

CONTROL: 1byte. Framenummer that was requested. If CONTROL is 0xFF then no more data can be requested (frame is empty).

MESS: up to 120 bytes. Name of 10 files (each filename is 12 bytes long (filled up with '\0') and given without extension).

**2. Get name of actual message:**

STX ADDR 0x27 CRC ETX

Answer: a. STX ADDR 0x27 ACK NAME DATA CRC ETX

NAME: 8bytes. Name of the actual message.

DATA: 2 bytes. 0x00 0x00, with no meaning.

**3. Get active alarm of the laser:**

STX ADDR 0x2C NUMBER CRC ETX

NUMBER: 1 byte,; 0x00: only the last active alarm code is requested  
>0x00: the last 10 alarm codes are requested

Answer: a. STX ADDR 0x2C ACK DATA CRC ETX

DATA: 2 bytes. for Number== 0x00; Encodes the last active alarm.  
DATA: 10 bytes. for Number> 0x00;Encodes the last 10 alarms.

**The following codes do not define an alarm:**

0x0001: Start of writing the alarmlist

0x000E: all alarms are cleared

0x000F: Laser is ON

**The following codes define an alarm:**

0x02: Laser is OFF (interlock open)

0x03: shutter is closed

0x04: DC Power fails

0x05: Overtemp of amplifier

0x06: Q-switch overtemp

0x07: high reverse power

0x08: low forward power

0x09: YAG RS232 error

0x0A: belt stopped

0x0B: program check not ok

0x0C: wrong figure-types in file

0x0D: no memory available

0x10: file not found

0x11: waterflow fails

0x12: interlock

0x14: laserfail

0x15: invalid font (or not existing)

0x16: overtemp.

0x25: shutter closed

0x26: laser not ready

: 0x30: overspeed

0x31: harddisk full

0x32: file not allowed to print

0x33: barcode creation failed (wrong parameters)

0x34: no barcode license

0x35: no barcode library (wrong version)

**Note:**

The alarmcode depends on the lasertype fo the system. Some alarms will be cleared automatically on startup of the system.

***4. Start printing***

STX ADDR 0x2D NAME NUMBER CRC ETX

NAME: 8 bytes. Messagename to be printed. If name is 0x00 0x00 ... then the last message will be printed.

Note: from 5.0.8 on the NAME can be up 16 bytes long (filename 12 bytes + '.' + 3 bytes extension) and can include the extension. If no extension is given the standard extension "msf" is assumed.

Version < 5.0.8: NAME must be exactly 8 bytes (filled up with '\0') and may not include the extension

Version >= 5.0.8: NAME can be up to 16 bytes (including extension) and must not be filled up with '\0' bytes.

NUMBER: 2 bytes.

Number of prints to be done. If 0x00 0x00 then the laser is printing infinitely.

If Number == 0x00 0x01, the system prints immediately without waiting for a trigger signal (testprint)

Answer: a. STX ADDR 0x2D ACK CRC ETX. Printing will be started.

b. STX ADDR 0x2D NACK DATA CRC ETX. Printing will not be started.

DATA: 2 bytes. Codifies the reason why printing is not started.

0x0848 means there are some alarms active.

0x0C0C means that the message does not exist.

### **5. Stop printing**

STX ADDR 0x2E CRC ETX

Answer: a. STX ADDR 0x2E ACK CRC ETX. Printing stopped.

### **6. Activate/deactivate external messageselection/batchjob**

STX ADDR 0x3A CHECKUMT CRC ETX

CHECKUMT: 1 byte.

0x00: deactivate (set to default mode)

0x01: activate external selection

0x02: request actual mode

0x04: activate batchjob execution

Answer: a. STX ADDR 0x3A ACK DATA CRC ETX.

DATA: 1 byte. Returns the actual mode as defined above.

### **7. Send usermessage**

STX ADDR 0x41 MNUMBER LENGTH MESSAGE LAST CRC ETX

MNUMBER: 1 byte. Fieldnumber of the usermessage

LENGTH: 1 byte. Number n of chars in the message

MESSAGE: n bytes. ASCII-code of the message (maximum 127 chars)

LAST: 1 byte, meaningless

Answer: a. STX ADDR 0x41 ACK CRC ETX, data received correctly

b. STX ADDR 0x41 NACK DATA CRC ETX, data not correctly received.

DATA: 2 bytes. (0x00 0x00) no meaning.

### **8. Set global internal counter (sets the value of a sequential number, defined as global internal):**

STX ADDR 0x90 CNUMBER LENGTH COUNTER CRC ETX

CNUMBER: 1 byte. Fieldnumber of the external global counter

LENGTH: 1 byte. Number n of digits in the counter

COUNTER: n bytes. ASCII-code of the counter (e.g. for 123 -> 0x31 0x32 0x33).

Answer: a. STX ADDR 0x90 ACK CRC ETX, data received correctly

b. STX ADDR 0x90 NACK DATA CRC ETX, data not correctly read.

DATA: 2 bytes. no meaning.

### **9. Get internal time:**

STX ADDR 0x38 CRC ETX

Answer: a. STX ADDR 0x38 ACK DATE TIME CRC ETX

DATE: 5 bytes.

struct \_dosdate\_t{

```

        unsigned char day;//day of month:1-31
        unsigned char month;//month of year: 1-12
        WORD year;//Year: 0-119, relative to 1980
        unsigned char dayofweek;//day of week: 0-6
    };
    TIME: 4 bytes.
    struct _dos_time_t{
        unsigned char hour;//hour: 0-23
        unsigned char minute;//minute: 0-59
        unsigned char second;//second: 0-59
        unsigned char hsecond;//1/100 second: 0-99
    };

```

### 10. Set internal time:

STX ADDR 0x39 DATE TIME CRC ETX

DATE: 5 bytes (see above)

TIME: 4 bytes (see above)

Answer: a. STX ADDR 0x39 ACK DATA CRC ETX , time changed

DATA: 2 bytes, meaningless

b. STX ADDR 0x39 NACK DATA CRC ETX , error in time format

DATA: 2 bytes 0x1F0F : date not valid

0x0F0F : time not valid

### 11. Get Status:

STX ADDR 0x70 CRC ETX

Answer: a. STX ADDR 0x70 ACK DATA CRC ETX

DATA: 40 bytes (status information)

44 bytes (from ScanLinux 4.2.9 on)

status information:

4 bytes: d\_counter (number of o.k. prints since start signal)

4 bytes: s\_counter (number of total prints since start signal)

4 bytes: message\_port (actual number of bit-selection, ext. message table)

1 byte: Start (0x00: laser not in printing mode, 0x01: laser in printing mode)

1 byte: Request (0x00: laser not in request mode, 0x01: laser in request mode)

1 byte: option (internal option)

1 byte: mode (actual mode, 0x00: default, 0x01: external message table, 0x04:

batch job table

4 bytes: t\_counter (number of total prints)

4 bytes: copies (number of prints to be printed)

4 bytes: alarm : upper WORD: Last active alarm code, lower WORD: 0x0848

(alarm active)

0x0000 (no alarms) 0xFFFF (initialization failed)

4 bytes: time (time of last print in milliseconds)

8 bytes: name of actual printing file

4 bytes: (from Scanlinux 4.2.9 on) Alarm bitmask, each bit indicates an actual active alarms of the system according 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	water-temperature (obsolete)
0x00000200	water-level (obsolete)
0x00000400	overpressure (obsolete)
0x00001000	triggersignal (obsolete)
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	unknown figure

**11 b. Get extended Status (from ScanLinux 5.0.8 on):**

STX ADDR 0x70 0x00 CRC ETX

Answer: a. STX ADDR 0x70 ACK DATA CRC ETX

DATA: 72 bytes (status information)

status information:

4 bytes: d\_counter (number of o.k. prints since start signal)

4 bytes: s\_counter (number of total prints since start signal)

4 bytes: message\_port (actual number of bit-selection, ext. message table)

1 byte: Start (0x00: laser not in printing mode, 0x01: laser in printing mode)

1 byte: Request (0x00: laser not in request mode, 0x01: laser in request mode)

1 byte: option (internal option)

1 byte: mode (actual mode, 0x00: default, 0x01: external message table, 0x04:

batch job table

4 bytes: t\_counter (number of total prints)

4 bytes: copies (number of prints to be printed)

4 bytes: alarm : upper WORD: Last active alarm code, lower WORD: 0x0848

(alarm active)

0x0000 (no alarms) 0xFFFF (initialization failed)

4 bytes: time (time of last print in milliseconds)

4 bytes: Alarm bitmask

4 bytes: reserved

16 bytes: actual message name (with extension)

16 bytes: actual eventhandler filename (with extension)

### **12. Get fontnames:**

STX ADDR 0x23 NUMBER CRC ETX (gets the fontnames of the simple fonts (ASCII fonts))

NUMBER: 1 byte. The names of the fonts are packed into frames of 10 fontnames. This byte defines the framenummer you request. The first framenummer is 0x00.

Answer: a. STX ADDR 0x23 CONTROL MESS CRC ETX

CONTROL: 1byte. Framenummer that was requested. If CONTROL is 0xFF then no more data can be requested (frame is empty).

MESS: up to 120 bytes. Name of 10 fonts (each fontname is 12 bytes long and does not contain the extension).

STX ADDR 0x25 NUMBER CRC ETX (gets the fontnames of the extended fonts (UNICODE fonts), since Scanlinux 4.3.3)

NUMBER: 1 byte. The names of the fonts are packed into frames of 10 fontnames. This byte defines the framenummer you request. The first framenummer is 0x00.

Answer: a. STX ADDR 0x24 CONTROL MESS CRC ETX

CONTROL: 1byte. Framenummer that was requested. If CONTROL is 0xFF then no more data can be requested (frame is empty).

MESS: up to 120 bytes. Name of 10 fonts (each fontname is 12 bytes long and does not contain the extension).

### **13. Change actual message configuration:**

STX ADDR 0x34 DATA CRC ETX

DATA: 10 bytes.

2 bytes: resolution (1-40)

2 bytes: printing speed in mm/sec (1-40 000)

2 bytes: laser power in 1/1000 (1-1000)

2 bytes: laser frequency in 1/10 kHz (10 - 1000)

2 bytes: print delay in 1/10 mm (dynamic printing, values: 0 - 65535) or in 1/10 milliseconds

(static printing, values 0-65535 or 0 sec - 6.5 sec),

Answer: a. STX ADDR 0x34 ACK CRC ETX , data accepted and written to file

b. STX ADDR 0x34 NACK DATA CRC ETX , data not accepted (file is protected against writing)

DATA: 2 bytes, no meaning

### **14. Get actual message configuration:**

STX ADDR 0x91 CRC ETX

Answer: a. STX ADDR 0x91 ACK DATA CRC ETX

DATA: 10 bytes, see above

### **15. Set selected file as default file (will change the actual printed message)**

STX ADDR 0x57 NAME CRC ETX

NAME: 8 bytes.

Messagename to be set as default (without extension for versions < 5.0.8).

From ScanLinux 5.0.8 on the Messagename can be 16 bytes long including the extension of the file. If no extension is given the default "msf" extension is assumed.

Version < 5.0.8: NAME must be exactly 8 bytes (filled up with '\0') and may not include the extension

Version >= 5.0.8: NAME can be up to 16 bytes (including extension) and must not be filled up with '\0' bytes.

Answer: a. STX ADDR 0x57 ACK CRC ETX. Actual message changed to selected file.

b. STX ADDR 0x57 NACK DATA CRC ETX. Selected file does not exist !  
DATA : 2 bytes , meaningless.

#### **16. Reset counter (internal d\_counter,s\_counter,t\_counter)**

STX ADDR 0x9C CRC ETX

Answer: a. STX ADDR 0x9C ACK CRC ETX. Internal counter resetted.

#### **17. Panic command (when sent during startup of the program before starting of the main execution code)**

STX ADDR 0x75 CRC ETX

Answer: a. STX ADDR 0x75 ACK CRC ETX. Panic command accepted, system will not load any file on startup.

b. STX ADDR 0x75 NACK DATA CRC ETX. Panic command not accepted, because system has already started.

DATA: 2 bytes, meaningless

**Note:** This command is a "backdoor" to prevent the system of loading any files on startup. This command should be used only

when the system hangs up during startup (probably because of some corrupted files). You should try to send this command every second during startup, until you receive an answer. If the answer contains ACK, the system will not have loaded any of its configuration files. If the answer is NACK, the system will have already started up and the command will not have any effect.

#### **18. Request sequential numbers/user messages of the actual message:**

STX ADDR 0x9D DATA CRC ETX

DATA: 2 bytes.

(1.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 2.byte) internal global number in actual message

0x08: request n-th (defined by 2.byte) usermessage in actual message

0x10: request the comment of the n-th (defined by 2.byte) usermessage in the actual message. (ScanLinux 5.1.5 or higher required). The coding of the answer will then be in UTF-8.

(2.Byte) 1 byte: number or field to be requested.

If we request a sequential number (1.byte=0x00), then this byte defines the index of the sequential number to be requested.

Example: (first 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.

NOTE: when DATA contains the byte values 0x02, 0x03 or 0x1B you have to precede these bytes with an additional 0x1B (ESC-sequence) !!!!

Answer: a. STX ADDR 0x9D ACK ID COUNT [CODING COUNTS] DATA CRC ETX

ID 2 bytes: ID of the requested sequential number, or field number of the internal global counter/usermessage.

(If 0xFFFF, then the request failed (no more usermessages/global numbers in the actual message))

COUNT 1 byte: number n of following bytes ; if COUNT = 0xFF, the two following bytes CODING and COUNTS are sent accordingly, else CODING and COUNTS are not sent !

CODING 1 byte: defines the coding of the following bytes (is only sent when COUNT=0xFF)

0x00: coding is in ASCII

0x01: coding is in UTF8

COUNTS 1 byte: defines the number n of the data bytes incase that COUNT=0xFF, else COUNTS was not sent in the answer.

DATA n bytes: contains the ASCII-text or UTF-8--sequence of the requested object according the CODING byte. If the CODING byte was not sent, ASCII is used.

Example1: (Request first usermessage inside the actual message)

STX ADDR 0x9D 0x08 0x00 CRC ETX (1. byte = 0x08, 2. byte = 0x00)

Answer:

STX ADDR 0x9D ACK FIELDNUMBER BYTENUMBER TEXT..... CRC ETX

FIELDNUMBER: 2 bytes, indicating the field of the usermessage or 0xFFFF if there is no first usermessage inside the actual message.

BYTENUMBER: 1 byte, indicating the length of the following text (the contents of the usermessage)

TEXT... the contents of the usermessage.

Example2: (Request usermessage of field 2)

STX ADDR 0x9D 0x1B 0x02 0x1B 0x02 CRC ETX (1. byte = 0x02, 2. byte = 0x02, not that both are preceded by 0x1B, the ESC-character !)

Answer:

STX ADDR 0x9D ACK FIELDNUMBER BYTENUMBER TEXT..... CRC ETX

FIELDNUMBER: 2 bytes, indicating the requested field of the usermessage  
BYTENUMBER: 1 byte, indicating the length of the following text (the contents of the usermessage)

TEXT...the contents of the usermessage.

### **19. Set sequential numbers of the actual message:**

STX ADDR 0x9E DATA CRC ETX

DATA: i bytes.

2 bytes: ID of the sequential number to be set.

1 byte: number n of following bytes

n bytes: contains the ASCII-text of the number to be set

If the number n is zero, the sequential number will be reset to its start value.

Answer: a. STX ADDR 0x9D ACK CRC ETX If command was executed correctly.

b. STX ADDR 0x9D NACK DATA CRC ETX The ID does not correspond to a sequential number.

DATA: 2 bytes, meaningless

Note: a. If you do not know the ID of the sequential number, you like to set, you have to request first the ID's of the first, second, ...sequential number in your message with the 0x2D command (see above). If you have created the message with Marca, the ID of the sequential number is shown in the "general properties" when you select the object. If you have created the message with the editor, the order you created the objects defines the ID of the object (e.g. the first object will have ID=0, the second ID=1, ...).

b. If the message is created with Marca, the sequential number cannot be saved in the file with this command. Therefore, this command must be sent AFTER the "start print" command. If it is sent BEFORE the "start print" command, the "start print" command will reset the value to the original start value of the sequential number.

c. If the message is created with the handheld terminal (or the touch screen), AND you select the sequential number with the option "Maintain Counter", the actual value of the sequential number will be saved in the file each time you send this command or when you change the actual message. With a sudden loss of power the actual value will be lost in any case !

d. If you wish to control the value of the sequential number by RS-232 it is recommended to use always an internal global counter. You can define up to 16 different internal global counters (identified by their field number 0-15). The value of an internal global counter will be maintained when changing the message and can be set with the command 0x90 and requested with the command 0x9D. On startup of the laser you should ALWAYS set the internal global counter to an initial value.

### **20. Select the printing mode (static/dynamic)**

STX ADDR 0x21 MODE CRC ETX

MODE: 1 byte.

0x00: set mode to static

0x01: set mode to dynamic-standard

0x02: set mode to dynamic-distance

0x03: set mode to dynamic-static

0x08: request the actual mode

Answer: a. STX ADDR 0x21 ACK DATA CRC ETX. Command o.k.

DATA: 2 bytes. Returns the actual mode as defined above as a WORD

in the order |upper byte|lower byte|.

b. STX ADDR 0x21 NACK DATA CRC ETX. selected mode is not valid.

DATA: 2 bytes , no meaning.

ScanLinux version 4.2.7:

STX ADDR 0x21 MODE DATA CRC ETX

MODE: 1 byte.

0x00: set mode to static  
0x01: set mode to dynamic-standard  
0x02: set mode to dynamic-distance  
0x03: set mode to dynamic-static  
0x08: request the actual mode

0x10: set dynamic printdistance or dynamic velocity (5 DATA bytes)  
0x20: get dynamic printdistance or dynamic velocity (1 DATA byte)

DATA: optional 1 or 5 bytes: |Byte1|Byte2|Byte3| Byte4| Byte5|

Byte1: 0x00: set or get dynamic distance (in micrometers)  
0x01: set or get dynamic velocity (in mm/min)

Byte2....Byte5: dynamic distance or dynamic velocity as a DWORD in the order  
|upper byte|....|lower byte| (data obligatory when MODE = 0x10).

Answer: a. STX ADDR 0x21 ACK DATA CRC ETX. Command o.k.

DATA: 2 bytes for MODE 0x00, 0x01, 0x02, 0x03,0x08 (returns the actual mode as a  
WORD in the order |upper byte|lower byte|.

4 bytes for MODE 0x20 returning the dynamic printdistance (0x10) or dynamic velocity  
(0x20) as a DWORD in the order |upper byte|....|lower byte|

b. STX ADDR 0x21 NACK DATA CRC ETX. selected mode is not valid or data not valid.

DATA: 2 bytes , no meaning.

## 21. Send an msf-file to the laser

**Comment:** sending a file to the laser is done using a sequence of commands, because usually the data cannot be sent within one single command. The sequence is like the following:

1. send the filename and the number of frames you will have to sent to the laser. The number of frames to be sent to the laser is calculated by  $n = \text{filesize} / 128$ .
2. after having received an ACK from the laser, the frames (the framenummer + the data) must be sent one by one. Each time you send a frame, you have to check the answer of the laser for an ACK.
3. having sent all frames to the laser, you have to sent the "last command" to the laser, indicating the laser that the filesending has terminated. If all data were received correctly and the laser can create the file on its harddisk, the laser will respond you with an ACK, otherwise you will receive a NACK.

**Send the filename and framenummer:**

STX ADDR 0x24 NAME FRAMES CRC ETX

NAME: 8 bytes. Name of the msf file to be sent (without extension)

FRAMES: 2 bytes. Number of frames that will be sent (one frame contains 128 bytes)  
example: your datafile is 260 bytes, so you will have to sent 3 frames  
(1. frame 128bytes, 2.frame 128 bytes, 3.frame 4 bytes; 128 + 128 + 4  
= 260 bytes)

Answer: a. STX ADDR 0x24 ACK DATA CRC ETX. Command o.k., laser prepared to receive the frame data.

DATA: 2 bytes. no meaning

b. STX ADDR 0x24 NACK DATA CRC ETX. laser not prepared to receive data.

DATA: 2 bytes , no meaning.

### **Sending the frames (in case of receiveing ACK from the laser):**

Send frame: STX ADDR 0x24 FRAMENUMBER DATA CRC ETX

FRAMENUMBER: 2 bytes (0 - 0xFFF0), number of received frame.

DATA: up to 128 bytes (ESC characters NOT included)

Answer: a. STX ADDR 0x24 ACK FRAMENUMBER CRC ETX, frame received correctly.

FRAMENUMBER: 2 bytes number of received frame.

b. STX ADDR 0x24 NACK FRAMENUMBER CRC ETX, frame not received correctly

FRAMENUMBER: 2 bytes number of received frame.

### **Sending last command:**

STX ADDR 0x24 FRAMENUMBER DATA CRC ETX, all frames sent.

FRAMENUMBER: 2 bytes = 0xFFFF

Answer: a. STX ADDR 0x24 ACK DATA CRC ETX, file correctly received.

DATA: 2 bytes, no meaning

b. STX ADDR 0x24 NACK DATA CRC ETX, file not correctly received because of some previous error, or

because of no more diskpace available.

DATA: 2 bytes. no meaning

## **22. Get simple Status of the laser**

STX ADDR 0x40 CRC ETX

Answer: a. STX ADDR 0x40 ACK CRC ETX. No alarms active.

b. STX ADDR 0x40 NACK CRC ETX. Some alarms are active.

c. STX ADDR 0x40 0x0C CRC ETX. No alarms active and laser is in printmodus.

d. STX ADDR 0x40 0x0D CRC ETX. Some alarms active and laser is in printmodus.

## **Additional commands from version 2.6b on**

### **23. Change Baudrate of RS-232 communication**

STX ADDR 0xBB NUMBER CRC ETX

Number: 2 bytes: codifies the new baudrate

(in the order HIGH BYTE | LOWBYTE)

0: 9600  
1: 19200  
2: 38400  
3: 57600  
4: 115200

Answer: a. STX ADDR 0xBB ACK CRC ETX.

Note: When you switch the baudrate you have to take care doing it always synchronously with the laser's actual baudrate. After having set the new baudrate, you have to switch also your device to the new baudrate before restarting communication with the laser.

#### **24. Add message to message table:**

STX ADDR 0x9A DATA CRC ETX

ATA: 1 byte : index, where to be added  
8 bytes: name of message to be added

Answer: a. STX ADDR 0x9A ACK CRC ETX if correct.  
b. STX ADDR 0x9A NACK CRC ETX if error.

#### **25. Delete message table**

STX ADDR 0x9B CRC ETX

Answer: a. STX ADDR 0x9B ACK CRC ETX

## **Additional commands from version 2.7 on**

#### **23. Prevent checksum check**

STX ADDR 0xAA CMD ..... CRC(arbitrary) ETX

If 0xAA precedes any arbitrary RS-232 command, the laser will be prevented from doing a checksum analysis. This is useful if you, or your hardware, cannot calculate the checksum of each command to be sent.

An example is a barcode reader that actualizes automatically an usermessage field via the RS-232 port. Typically, you can only program some fixed start/end-header bytes to the barcode read by a scanner. As the checksum would depend on the data that are read by the scanner, each time the barcode reader would have to calculate a new checksum. If you use "0xAA" preceding the command, you can set a fixed checksum for all of your RS-232 commands (e.g. set the fixed checksum to 0x00).

Note: Using this option does simple mean that the value of the sent checksum doesn't mind. However, you are not allowed to set this value to any of the control bytes like STX(0x2), ETX(0x3) or ESC (0x1B) !!!

#### **24. Select printing file using a predefined message table (key <---> message)**

STX ADDR 0x22 KEY CRC ETX

KEY: n bytes. ASCII code of the key (up to 32 BYTES)

Answer: a. STX ADDR 0x57 ACK CRC ETX. Actual message changed to selected key.  
b. STX ADDR 0x57 NACK DATA CRC ETX. Selected file of the key does not exist !  
DATA : 2 bytes , meaningless.

Note: when the laser receives this command it looks at the "msftable.db" file and scans the "msftable.db" for the key. It then tries to open the corresponding message. Note, that the answer CMD is not 0x22 but 0x57 !!!!

The "msftable.db" can be created by any texteditor (e.g. notepad) and must be saved as a txt-file

("msftabe.txt"). With Marca (with in the control dialog) this file can be sent as a database file (extension=".db") to the laser.

The structure of the "msftabe.db" file is as follows:

\*\*\*Example for database file\*\*\*

Format:

```
<Key><TAB><Filename>
A045461ertEW      2dmatrix
k0235361FRG      newfile
B037657678hhr    quad
.....
```

## 25. Enable/disable buffered usermessages (from ScanLinux 3.6.2 on)

a)

STX ADDR 0x63 SIZE CRC ETX

SIZE: 1 byte. sets the size of the buffer (typically < 40); if size==0x00 the buffered mode will be disabled (default mode).

Answer: a. STX ADDR 0x63 ACK SIZE CRC ETX.

SIZE : 1 byte , the actual size of the buffer

b) from ScanLinux 5.0.8 on

STX ADDR 0x63 CONTROLBYTE SIZE CRC ETX

CONTROLBYTE: 1 byte. 0x00: set the buffersize ; 0x01: get the actual buffersize

SIZE: 1 byte. sets the size of the buffer (typically < 40); if size==0x00 the buffered mode will be disabled (default mode). If the CONTROLBYTE == 0x01, the value of the SIZE byte is meaningless.

Answer: a. STX ADDR 0x63 ACK SIZE CRC ETX.

SIZE : 1 byte , the actual size of the buffer

This command can be used to switch the laser into a buffered usermessage mode. Only fields 0,1,2 and 3 are affected of this command and are the only fields that can be used for a buffered usermessage. The buffer works like a FIFO , the laser prints one by one each entry of the buffer. When you set the buffersize the FIFO is internally resetted. From then on you can fill the FIFO with the standard usermessage command (0x41). When the FIFO is full, the 0x41 command will respond with a NACK. With each print, exactly one element of the FIFO is removed.

The maximum FIFO size is 255 entries, but typically you should use 10 – 20. The advantage of the buffered usermessage is that you can send continuously your data in advance to the laser until you get a NACK. Then you just have to wait and try again until you get an ACK. There is no need to synchronize exactly the sending with the printing.

If FIFO is empty (all entries are already printed) and the laser is activated for printing it will launch the EMPTYMESSAGE error, whether this alarm is configured or not, and will pull the COMPUTERREADY signal down (in autostart mode or -x commandline mode) until a new usermessage is sent.

## 26. X/Y-Offset of complete message (from ScanLinux 4.0.1 on)

With this command you can shift the complete message in x and y in the scanfield. You can specify 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.

STX ADDR 0x59 ABSOLUTE UNITS RESET X\_OFFSET Y\_OFFSET CRC ETX

ABSOLUTE: 1 byte.                      0x00: absolute offset ;    0x01: relative offset (depends on previous offsets)

UNITS: 1 byte    0x00: ideal coordinates ( 0 – 100 000 corresponding the complete scanfield)  
          0x01: units in microns  
          0x02: units in 1/10 mm

RESET: 1 byte    0x00: XY-offset is remained until a new command is send or the configuration of the laser is manually changed.

          0x01: XY-offset is resetted to ZERO after the next print (so this offset will only be valid for the next print)

X\_OFFSET: 4 bytes            the X offset as a long Integer (4 bytes) in High Byte to Low Byte order.

          Y\_OFFSET: 4 bytes                      the Y offset as a long Integer (4 bytes) in High Byte to Low Byte order.

- Answer:
- a.        STX ADDR 0x59 NACK CRC ETX, in case that too less data have been sent.
  - b.        STX ADDR 0x59 ACK OFFSET\_X OFFSET\_Y CRC ETX, in case of a correct command.

          OFFSET\_X: 4 bytes            the total absolute x offset that will be applied for the next print in the requested units.

OFFSET\_Y: 4 bytes            the total absolute y offset that will be applied for the next print in the requested units.

**27. power/speed scaling or bitmap pixeltime scaling (from ScanLinux 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 !

STX ADDR 0x76 SETORGET MEMBER VALUE(upper byte) VALUE(lower byte) CRC ETX

SETORGET: 1 byte.                      0x00: sets a new scaling factor

          0x01: gets an actual scaling factor

MEMBER: 1 byte                      0x00: set or get the pixel time scaling (for bitmaps or 2D codes)

          0x01: set or get the pixel power scaling (for bitmaps or 2D codes and only for YAG and Fiber laser)

          0x02: set or get the layer’s marking speed

          0x03: set or get the layer’s marking power

          VALUE(upper byte): 1 byte                      the upper byte of the scaling factor. 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 (SETORGET==0x01), the value of VALUE(upper byte) and VALUE(lower byte) are irrelevant.

VALUE(lower byte): 1 byte          the lower byte of the scaling factor.

- Answer:
- a.        STX ADDR 0x76 NACK CRC ETX, in case that too less data have been sent.
  - b.        STX ADDR 0x76 ACK SETORGET MEMBER VALUE(upper byte)  
            VALUE(lower byte) CRC ETX, in case of a correct command.

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

### **28. Shift object in x and y (from ScanLinux 4.2.8 on)**

With this command you can shift individual objects in its (x,y)-position. The shift can be absolute or relative to the actual position of the object.

STX ADDR 0x77 SETORGET UNITS REFERENCEPOINT OBJECTID ABSOLUTE X-VALUE Y-VALUE CRC ETX

SETORGET: 1 byte.                          Defines, if the position should be set or just requested.

0x00: sets a new (x,y) position

0x01: gets the actual (x,y) position

UNITS: 1 byte    Defines the units to be used for the x,y-values.

0x00: ideal coordinates ( 0 – 100 000 corresponding the complete scanfield)

0x01: units in microns

0x02: units in 1/10 mm

REFERENCEPOINT: 1byte          defines the reference point for an absolute shift or when the actual (x,y) position is requestet (SETORGET=0x01)

0x00: bottom-left of the enclosing rectangle

0x01: top-left of the enclosing rectangle

0x02: bottom-right of the enclosing rectangle

0x03: top-right of the enclosing rectangle



0x04: center-point of the enclosing rectangle

OBJECTID: 2 bytes: defines the object-ID of the object that should be shifted. Each object in a message has its own ID. With Marca you can visualize this ID. When you create a message with the HH-terminal or the touchscreen the ID is given by the order you created the objects. When you group the lines together, each one as a grouped text object, the first line (grouped text) will have the ID=0, the second line ID=1, the third line ID=2 and the 4<sup>th</sup> line ID=3.

Byte-order is: |UPPER byte | LOWER byte|

ABSOLUTE: 1 byte. Defines if the shift should be done absolute or relative to the actual position of the object

0x00: absolute offset ; the object will be placed so that the selected reference point matches exactly the desired (x,y) position.

0x01: relative offset (depends on previous offsets and is independent of the reference point)

X\_VALUE: 4 bytes: the x position of the shift in units defined by the UNITS-byte.

Byte-order is: |UPPER byte |...|...|LOWER byte|

Y\_VALUE: 4 bytes: the y position of the shift in units defined by the UNITS-byte.

Byte-order is: |UPPER byte |...|...|LOWER byte|

Ideal coordinates range from [0 – 100000], real coordinates depend on the adjusted scanfield-size (e.g. [0 – 100 mm] for 100 x 100 mm lens , [0 – 60mm] for 60 x 60 mm lens).

Answer:

- STX ADDR 0x77 NACK CRC ETX, in case that too less than 5 bytes has been sent (minimum data bytes for a GET command)
- STX ADDR 0x77 NACK 0x1F 0x0F CRC ETX, in case that too less data has been sent for a SET command or the object with the given ID does not exist.
- STX ADDR 0x77 ACK CRC ETX, in case of a correct SET command.

d. STX ADDR 0x77 ACK X\_VALUE Y\_VALUE CRC ETX, in case of a correct GET command.

NOTE: the shift in X and Y is permanent only for the actual loaded message as long as the messages is not reloaded again. The shift will in neither case be saved in the original message file. It is a shift the resides only in memory. A 0x2D command (StartPrint) will force always the message to be reloaded from harddisk into memory, thus any previous shiftxy command will not be applied.

The sequence for printing instructions must be:

- select the message to be printed with the 0x2D (StartPrint) command.
- apply the SHIFTTY command.
- each time a new 0x2D (StartPrint) or 0x57(set default message) command is sent , the shiftxy is reset to zero.

**29. Send UTF8-usermessage (sends a usermessage as a UTF8-sequence (Unicode textobjects))**

STX ADDR 0x43 MNUMBER 0xFF 0x01 LENGTH TEXT LAST CRC ETX

MNUMBER: 1 byte. Fieldnumber of the usermessage  
LENGTH: 1 byte. Number n of chars in the message  
TEXT: n bytes. UTF8 bytesequence of the text (maximum 248 bytes)  
LAST: 1 byte, meaningless

Answer: a. STX ADDR 0x43 ACK CRC ETX , data received correctly  
b. STX ADDR 0x43 NACK DATA CRC ETX, data not correctly received.  
DATA: 2 bytes. (0x00 0x00) no meaning.

**30. Activate test diode (deactivation by stopsignal):**

STX ADDR 0x60 0x16 CRC ETX

From scanlinux 5.1.3 on optionally with more parameters

STX ADDR 0x60 MODEBYTE XPOS YPOS CRC ETX  
MODEBYTE: 0x16 (activates real pointer if set in oemvars)  
0x20 (single point mode with XPOS and YPOS)  
0x30 (single point mode with internal configuration values)  
XPOS: 2 bytes  
The x-position in 1/10 units of ideal coordinates [-5000,+5000].  
YPOS: 2 bytes  
The y-position in 1/10 units of ideal coordinates [-5000,+5000].  
Byte-order is: |UPPER byte | LOWER byte|

Answer: a. STX ADDR 0x60 ACK CRC ETX

**31. Send an arbitrary file to the laser**

**Comment: sending a file to the laser is done using a sequence of commands, because usually the data cannot be sent within one single command. The sequence is like the following:**

- 1. send the filename and the number of frames you will have to sent to the laser. The number of frames to be sent to the laser is calculated by  $n = \text{filesize} / 128$ .**
- 2. after having received an ACK from the laser, the frames (the framenummer + the data) must be sent one by one. Each time you send a frame, you have to check the answer of the laser for an ACK.**
- 3. having sent all frames to the laser, you have to sent the "last command" to the laser, indicating the laser that the filesending has terminated. If all data were received correctly and the laser can create the file on its harddisk, the laser will respond you with an ACK, otherwise you will receive a NACK.**

**Send the filename and framenummer:**

STX ADDR 0xB0 NAME FRAMES CRC ETX

NAME: 16 bytes. Name of the file to be sent (with extension)  
FRAMES: 2 bytes. Number of frames that will be sent (one frame contains 128 bytes)  
example: your datafile is 260 bytes, so you will have to sent 3 frames  
(1. frame 128bytes, 2.frame 128 bytes, 3.frame 4 bytes;  $128 + 128 + 4 = 260$  bytes)

Answer: a. STX ADDR 0xB0 ACK DATA CRC ETX.  
command o.k., laser prepared to receive the frame data.  
DATA: 2 bytes. no meaning  
b. STX ADDR 0xB0 NACK DATA CRC ETX.  
laser not prepared to receive data.

DATA: 2 bytes , no meaning.

**Sending the frames (in case of receiveing ACK from the laser):**

Send frame: STX ADDR 0xB0 FRAMENUMBER DATA CRC ETX  
FRAMENUMBER: 2 bytes (0 - 0xFFF0)  
number of received rame.  
DATA: up to 128 bytes (ESC characters NOT included)

Answer: a. STX ADDR 0xB0 ACK FRAMENUMBER CRC ETX  
frame received correctly.  
FRAMENUMBER: 2 bytes  
number of received frame.  
b. STX ADDR 0xB0 NACK FRAMENUMBER CRC ETX  
frame not received correctly  
FRAMENUMBER: 2 bytes  
number of received frame.

**Sending last command:**

STX ADDR 0xB0 FRAMENUMBER DATA CRC ETX  
all frames sent.  
FRAMENUMBER: 2 bytes = 0xFFFF

Answer: a. STX ADDR 0xB0 ACK DATA CRC ETX  
file correctly received.  
DATA: 2 bytes, no meaning  
b. STX ADDR 0xB0 NACK DATA CRC ETX  
file not correctly received because of some previous error, or  
because of no more diskspace available.  
DATA: 2 bytes. no meaning

***Note: using command 0xB0 copies the sent file always into RAMdisk and harddisk.  
From scanlinux 5.1.3 on and using commandbyte 0xB3 instead of 0xB0 copies the file  
only to the RAMdisk.***

***32. Delete an arbitrary file inside the laser***

STX ADDR 0xB1 NAME CRC ETX  
NAME: 16 bytes. Name of the file to be deleted (with extension)

Answer: a. STX ADDR 0xB1 ACK DATA CRC ETX.  
command o.k., file deleted.  
DATA: 2 bytes. no meaning  
b. STX ADDR 0xB1 NACK DATA CRC ETX.  
file protected.  
DATA: 2 bytes , no meaning.

***33. Copy an arbitrary file from the laser***

***Comment: copying a file from the laser is done using a sequence of commands, because usually the  
data cannot be sent within one single command. The sequence is like the following:***

1. *send the filename to the laser. The laser calculates the number of frames  
The number of frames to be sent by the laser is calculated by  $n = \text{filesize} / 128$ .*
2. *after having received an ACK and the number of frames from the laser, you can request  
frame by frame from the laser*
3. *having received all frames from the laser the command sequence has terminated.*

*Send the filename :*

STX ADDR 0xB2 NAME CRC ETX

NAME: 16 bytes. Name of the file to be sent (with extension)

Answer: a. STX ADDR 0xB2 ACK DATA CRC ETX.

command o.k., laser prepared to receive the frame data.

DATA: 2 bytes. The number of frames of the file.

b. STX ADDR 0xB2 NACK DATA CRC ETX.

File is not available in the laser.

DATA: 2 bytes , no meaning.

**Requesting the frames (in case of receiveing ACK from the laser):**

Request frame: STX ADDR 0xB2 FRAMENUMBER CRC ETX

FRAMENUMBER: 2 bytes (0 - 0xFFF0)

number of requested frame.

Answer: a. STX ADDR 0xB2 ACK FRAMENUMBER DATA CRC ETX

DATA: up to 128 bytes (ESC characters NOT included)

FRAMENUMBER: 2 bytes

number of requested frame.

b. STX ADDR 0x36 0x34 ETX

requested frame number not correct.

**Sending last command to finish file sending:**

STX ADDR 0xB2 0xFF 0xFF CRC ETX

all frames received.

Answer: a. STX ADDR 0xB2 ACK DATA CRC ETX

file sending finished.

DATA: 2 bytes, no meaning

b. STX ADDR 0x36 0x34 ETX

file sending aborted.

**Note: using command 0xB2 searches the file to be sent always in the harddisk.**

**From scanlinux 5.1.3 on and using commandbyte 0xB4 instead of 0xB2 searches the file to be sent only in the RAMdisk.**

**34. Set the laser's configuration file (from ScanLinux 5.0.8 on)**

STX ADDR 0x50 NAME CRC ETX

NAME: up to 32 bytes. Name of the file to be loaded as the configuration file.

Answer: a. STX ADDR 0x50 ACK DATA CRC ETX.

configuration file found and loaded.

DATA: 2 bytes. no meaning

b. STX ADDR 0x50 NACK DATA CRC ETX.

configuration file not found and not loaded.

DATA: 2 bytes , no meaning.

If NAME is left empty the last internal configuration is tried to be loaded.

**35. Change parameters of the laser's configuration file (from ScanLinux 5.1.3 on)**

STX ADDR 0x49 NAME CRC ETX

NAME: up to 32 bytes. Name of the file to be loaded as the configuration file.

Answer: a. STX ADDR 0x49 ACK DATA CRC ETX.

configuration file found and loaded.

DATA: 2 bytes. no meaning

b. STX ADDR 0x49 NACK DATA CRC ETX.

configuration file not found and not loaded.

DATA: 2 bytes , no meaning.

The sent file does not need to contain a complete set of the configuration's parameters. Only the parameters that are inside the file will be changed.

**36. Transformation of actual message (from ScanLinux 5.2.1 on)**

With this command you can shift and rotate the complete message. The shift and rotation is valid until the message is reloaded. Note, that in dynamic printing a shift in x-direction with this command has only a real effect if the "autoshift" parameter is not enabled !

STX ADDR 0x65 TRANSFORMATION-BYTES CRC ETX

TRANSFORMATION\_BYTES: n bytes

The n bytes are forming a character string. The string has the following format:

x=<value>y=<value>p=<value>x0=<value>y0=<value>, 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 (default : center of the scanfield)

y0: the y-coordinate of the rotation center (default : center of the scanfield)

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

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: a. STX ADDR 0x65 NACK CRC ETX, in case that a reload of the message is pending.

b. STX ADDR 0x65 ACK CRC ETX, in case of a correct command.

**37. 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.  
STX ADDR 0x78 Subcommand-BYTE CRC ETX

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

Bit0 set: usermessages are stored

Bit1 set: global counters are stored.

Answer: STX ADDR 0x78 ACK Subcommand-Byte CRC ETX

**38 Get a preview file in svg-format**

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

STX ADDR 0x85 NAME-BYTES CRC ETX

NAME-Bytes:

16 Bytes defining the name of the svg file. If the name has less than characters the NAME-Bytes must be filled up with NULL bytes (0x00). If no name is defined (all NULL-Bytes) the created svg-file will be named "dump.svg".

Answer: STX ADDR 0x85 ACK CRC ETX Command o.k. and SVG file created.

STX ADDR 0x85 NACK CRC ETC Command not o.k.