Programmers Guide

Programmer's Guide

BrixX®

PhoxX®

LuxX®

LuxX+

QuixX

LedMOD®

Led HUB

Omicron-Laserage Laserprodukte GmbH
Raiffeisenstrasse 5e
63110 Rodgau, Germany
Tel.: +49-61 06 82 24-0
Fax: +49-61 06 82 24-10
mail@omicron-laser.de
www.omicron-laser.de

ISO 9001:2008

Imprint

Omicron-Laserage Laserprodukte GmbH

Raiffeisenstrasse 5e

D-63110 Rodgau

Germany

Phone + 49 – (0) 6106 – 8224-0

Fax + 49 – (0) 6106 – 8224-10

Internet www.omicron-laser.de

E-mail mail@omicron-laser.de

Images inside this manual may show slight deviation from the actually delivered laser system.
This does not affect function or operation.
Responsible for the content published: Omicron-Laserage Laserprodukte GmbH

The manufacturer keeps the right to change information and technical equipment without notice.

Printed in Germany 05/2019

Safety Rules

WARNING!

Read the safety precautions in this section before installing, powering, operating or servicing this product.

Symbols used in this manual

The following symbols are used to identify important safety information

WARNING! Laser/led emission is possible. Assure laser/led safety precautions.

CAUTION: Special attention to this point is necessary to meet health regulations and to avoid damage.

An important note to ensure an unproblematic operation.

Important safety information

This manual describes the programming functions of the PhoxX®, LuxX®, LuxX+®, BrixX® and QuixX® lasers and the LedHUB and LEDMOD.v2 led.
Before operating a laser or led system, read the user manual of the laser/led hardware carefully to prevent damages to human persons or animals, integrated devices and connected devices.

Always follow the local or IEC-safety rules (IEC 60825 / IEC 62471) during the operation of the system.

WARNING: The laser/led systems emit visible or invisible continuous laser/led radiation. The systems are variably equipped with different laser/led diodes and different output power.

ATTENTION: Do not look into a laser beam or into reflections of the beam from surfaces!

Table of contents

IMPORTANT: This document describes the global handling of the serial protocol of omicron devices and includes a complete command reference for omicron off-the-shelf products.

If you are programming a customized product always refer to the product specific command list to check what commands are valid for your specific device!

1. Introduction

The Omicron xX-laser series and led products are equipped with two serial interfaces:

- 1x USB
- 1x RS232

Both interfaces are equal and may be used in parallel.

If you like to use the USB port for communication you will need a suitable driver for your operating system. A driver should be automatically installed by Windows Update after connecting a laser or led device to the PC (assumed it is connected to the internet). A pre-installation setup file is provided on the Omicron Control Center installation medium for offline installations.

Drivers are also separately available at http://www.ftdichip.com/Drivers/D2XX.htm for several operating systems like Windows, Linux, Mac OS X, Windows CE and Android.

You may use the Virtual COM Port (VCP) provided by the VCP driver.
On Windows the VCP will appear in the device manager after you connected a laser or led device to the system. It will be assigned to the next free COM port number.
The virtual COM-port will appear totally transparent to legacy RS232 software. However the USB baud rate of 500000 may not be available within some older programs were you cannot select custom baud rates. In that case you may use the legacy RS232 port with 57600 baud instead. If your PC is not equipped with a legacy RS232 port you can use any standard USB-to-Serial adapter cable to control a laser or led device via RS232.

A second way of communication is direct USB programming supported by the D2XX driver. That gives you the additional possibility to recognize a laser directly after a connection is established. (For details on direct DLL programming please refer to the FTDI's D2XX_Programmer's_Guide which you can download at www.ftdichip.com/Documents/ProgramGuides.htm)

Regardless of how you control the laser/led (legacy RS232, VCP or USB DLL) the serial protocol is always the same, except for an optionally extension on USB called ad-hoc mode.

In principle the protocol is based on a common question (command) and answer scheme (master-slave protocol).

The ad-hoc mode extension adds the possibility for the laser/led to send a message directly if any state or parameter changes (multi-master protocol). This protocol extension has some advantages:

- There is no need for polling all necessary parameters in a closed loop since the computer will be notified if anything changes.
- The software is immediately up-to-date if anything unforeseen happens or measurement values are changing.

Some legacy RS232 software and machine interfaces may have problems with unexpected communication initiated by the device. Therefore this extension can be switched off for the USB port and is not available at the RS232 port.

Although the RS232 and USB interfaces are equal and may be used in parallel a command on the USB port will be answered on USB only and a command on the RS232 port will be answered on RS232 only.
In case the ad-hoc mode is enabled for USB, a command on RS232 will result in an ad-hoc message on the USB port after parameters or operating modes are changed.

As an example of use: this behavior enables you to send commands via RS232 from a machine interface and monitor the status changes via USB on a connected PC (e.g. with our Omicron Control Center).

Even though it is possible to use the USB and legacy RS232 port of the device simultaneously it is not possible to open the virtual COM port plus a communication port through direct USB DLL programming for the same device, since this is not supported by the FTDI chip.

Important: if the Omicron Control Center software is actually executed on your computer you are not able to open the virtual COM port of a connected laser/led since our software opens the USB communication port right after it recognizes a compatible device.

2. Interface Properties

Communication interfaces are pre-defined with the following parameters:

- USB: 500000 baud (direct DLL and virtual COM port programming)
- RS232: 57600 baud
- 8 data bits
- 1 Stop Bit
- no parity
- No hardware handshake

3. Protocol Definition

The logical protocol definition is specified as follows:

- All communication strings consist of printable ASCII characters, naturally "a-z", "A-Z", "0-9", ">" (ASCII greater than), "!" (ASCII exclamation mark), "?" (ASCII question mark), "$" (ASCII dollar sign) and the "§" (section sign [extended ASCII A7hex]) , alternatively "|"(vertical bar) *(see note for details)*.
  (Some parameters may also contain a decimal point "." or other special character like "/" or "-", e.g. the model code string).

  IMPORTANT NOTE:
  If a communication string contains more than one parameter the parameters will be separated by a "§" sign. The "§" (section sign) is not a pure ASCII character (0-127). It is an iso-8859-1, also called "Latin1" encoded character (extended ASCII range 128-255). It has a corresponding hex byte value of 0xA7 (167_{dec}) and is available on the legacy Windows codepage 1252. (be aware that the "§" may be displayed as a "?" if the codepage is not loaded within the terminal)

  On a standard US-keyboard you should get the §-sign by pressing the following key sequence:

  Alt + 0 1 6 7 (typed on numeric keypad)

  For further details like keyboard layout also see
  http://en.wikipedia.org/wiki/Section_sign

  Since we noticed that the handling of the "§" sign may produce some problems on computers that are not supporting the section sign by default like German language does, we added the possibility to change the delimiter "§" to the standard ASCII character "|" (vertical bar) with newer firmware versions.
  For details please refer to chapter "4.1 Get Firmware".

- The protocol is case sensitive.
- Each communication string ends with a carriage return (CR).
- Each command (question) sent from the host computer to the device controller starts with a "?" question mark.
- Each answer to a question, sent from the device controller to the host computer, starts with a "!" exclamation mark.
- Each ad-hoc message, which is sent from the device controller to the host computer without a preceding question starts with a "$" dollar sign.
- The preceding sign is followed by a 3 characters command code plus optional parameters.
- **Multi-Channel devices only:** The command code may be followed by a sub-device index in square brackets. (*For details please refer to chapter "Controlling Light Engines").*

- The lengths of the communication strings are not unified. Some questions and answers contain parameters with a variable length. The maximum command length is 42 characters including preceding sign and carriage return.
- If a communication string contains more than one parameter they will be separated by a "§" sign (alternatively "|" – see important note above).
- A successful execution of a command that sets any parameter or operating mode is indicated by a ">" greater than sign enclosed to the answer.
- An unsuccessful execution is indicated by an "x" sign.
  This can happen, if the value you try to set is outside the valid range or if it is not possible to execute the command within the actual operating state.
- An unknown command or an incomplete command is followed by a *"!UK"* answer.
- The device will usually answer a command within 100ms. A Timeout condition of 500ms will be safe.
- There is an "in-between characters" timeout of 100ms. Ensure that commands are always sent in one block to the device.

abbreviation	character	ASCII HEX
CR	carriage return	0D
>	greater than	3E
$	dollar sign	24
?	question mark	3F
!	exclamation mark	21
§	section sign	A7
|	vertical bar	7C
[	opening square bracket	5B
]	closing square bracket	5D

Table 1: ASCII command abbreviations

If you should have difficulties with our command philosophy referring the prefixed question mark for all types of commands (questions) the following explanation may help.

There are two types of commands:

- *Those who really ask the laser about parameters or operating states.*
  They are easily called a question.
- *And those who set parameters and operating modes.*
  They are more noticed as a command that you may associate with an exclamation mark(!).

Here is a small crib to understand why this is not:

> *Accepting that we always are very polite to our lasers, a set-command should be shaped as followed:*
>
> *"Would you please be so kind and set… for me?"*
>
> *Answered by a straight:*
>
> *"SIR YES SIR !"*

So keep in mind: ***command = question***

☺☺☺

3.1 Questions (PC ➔ device)

A question (command) from the host computer to the laser controller always starts with a "?" question mark followed by a three character long command indication. Keep in mind that the protocol is case sensitive.

A question that asks for a parameter is completed by a CR carriage return directly after the command indication.

A question that asks to set something is normally followed by one or more parameters that need to be set and concluded with a CR carriage return afterwards.
Parameters are separated by a "§" sign (alternatively "|"– see chapter protocol definitions).
Depending on the command the desired values sometimes have to be given in decimal integer, floating point or hexadecimal representation.

3.2 Answers (device ➔ PC)

Every question (command) causes an answer from the laser controller. The computer must not send any more commands until it gets the answer, unless you expect that a communication error occurred because of a timeout situation.
If the laser/led device is unable to identify the question it will send back a *"!UK"* (Unknown Command).

3.3 Ad-hoc Messages (device ➔ PC)

You may enable the ad-hoc mode for USB (and virtual COM port) communication. (See "SetOperatingMode" command for details) - (This is not possible for the legacy COM port)

If ad-hoc mode is enabled you will receive self-initiated messages from the device each time a value or state changes.

However there are situations where a message is sent on both ports (USB and legacy RS232) regardless of the ad-hoc mode setting:

- Switching on the power supply always causes the laser controller to send an *"$RsC>" (Reset Controller Ready)* message after the boot-up sequence.
- A reset command also implicates the "Reset Controller Ready" after booting is finished (see command description for details).
- The "Calibrate Laser Diode" command is closured by a "Calibrate Ready" message (see command description for details).

4. Command Reference

This chapter describes the detailed syntax for every single command.

If not otherwise described all commands are valid for all laser and led types.

- Possible differences are marked like this in the text

For additional information about protocol extensions related to light engines like the LedHUB® please refer to chapter "Controlling Light Engines".

If you are programming a customized product always refer to the product specific command list to check what commands are valid for your specific device!

4.1 Get Firmware

The first action after opening the communication port should always be to identify the laser/led we are connected to.

This will happen with the command "Get Firmware" (?GFw):

command type:	GFw															
function:	Get Firmware															
explication:	Ask for the model code, device ID and firmware version															
byte number:	**1**	**2**	**3**	**4**	**5**	**...**	**...**	**...**	**...**	**...**	**...**	**...**	**...**	**...**	**...**	**n**
command:	?	G	F	w	cr											
answer:	!	G	F	w	*Model code*			§	*Device-ID*			§	*Firmware*			cr

The answer will contain the model code, a device-ID and the firmware version of the device. The most important information here is the device-ID and the firmware version.

- PhoxX® lasers have the controller separated from the laser head. This command queries the controller data which is the main device data. The additional Firmware version of the laser head can be optionally queried with the command "?GFH" described below.

- The device-ID is a fixed integer value that is unique for each device type and gives you the possibility to identify the device type you are talking to:
  - **PhoxX®:** device-ID 3
  - **LuxX®:** device-ID 4
  - **LuxX®.HSA:** device-ID 31
  - **LuxX+®:** device-ID 18
  - **BrixX®:** device-ID 100, 104 and 105 (depending on the build in laser diode but not relevant for communication)
  - **BrixX®.UHP:** device-ID 103
  - **QuixX®:** device-ID 101
  - **LEDMOD.v2:** device-ID 19
  - **LedHUB®:** device-ID 20 *(see chapter "Controlling Light Engines" for details)*

- The Firmware version needs to be interpreted as a floating point value indicating the laser/led (controller) firmware version. This may be important for future firmware releases and enables you to handle newly added commands and slightly variances in existing commands.

Anyway this one is the first command for any communication and will keep compatible for any future releases.

- The model code is a string. This model code is for displaying only and should not be used for distinguishing between laser/led models. (device specifications will be passed by a separate command)

As described in chapter 3, with newer firmware versions we added the possibility to change the default delimiter "§" to the standard ASCII character "|" (vertical bar) to avoid problems with the nonstandard ASCII character "§".

If you send "?GFw|" instead of "?GFw" as the first command of your communication the delimiter for commands with more than one parameter will be "|" instead of "§" (after a device reset the delimiter is reset to default. Send this command again as the first command of your communication after a device reset).

- This functionality is available for:
  - PhoxX® (Firmware >= 2.83)
  - LuxX® (Firmware >= 2.0)
  - LuxX®.HSA
  - LuxX+®
  - BrixX® (Firmware >= 1.60)
  - BrixX®.UHP
  - QuixX®
  - LEDMOD.v2
  - LedHUB®

command type:	GFw|															
function:	Get Firmware and set delimiter															
explication:	Ask for the model code, device ID and firmware version and overwrite the standard delimiter sign to "|" for all following commands.															
byte number:	**1**	**2**	**3**	**4**	**5**	**6**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**n**
command:	?	G	F	w	|	cr										
answer:	!	G	F	w	*Model code*		|	*Device-ID*			|	*Firmware*			cr

Please note that if you are using this command extension all occurrences of the "§" sign need to be replaced by a "|" in the following command reference.

4.2 Get Firmware Head *(PhoxX® lasers only)*

- This command is valid for PhoxX® lasers only.

This command may be used to get some additional information about the connected laser head.

The syntax is equal to the "?GFw" command.

command type:	GFH															
function:	Get Firmware Head															
explication:	Ask for the head's model code, device-ID and firmware version															
byte number:	**1**	**2**	**3**	**4**	**5**	**...**	**...**	**...**	**...**	**...**	**...**	**...**	**...**	**...**	**...**	**n**
command:	?	G	F	H	cr											
answer:	!	G	F	H	*Model code*			§	*Device-ID*			§	*Firmware*			cr

The heads firmware version is normally different to the controller's one. Anyhow with this information it is possible to identify if the head's firmware is on the same development level as the controller and they harmonize together. This may be interesting if you own a larger number of PhoxX lasers that are produced over a longer time range and you expect that head's may be switched over to other systems because of service reasons. If you need a detailed firmware compatibility list, please contact Omicron for details.

The Device-ID and model code string is redundant and should be the same as with the controller.

4.3 Get Serial Number

This command queries the serial number of the controller. The serial number needs to be interpreted as a string. It may contain special characters like "/" or "-".
The serial number is for informative use only.

- PhoxX® lasers have the controller separated from the laser head. This command queries the controller data which is the main device data. The laser head serial number for PhoxX® lasers can be queried with the command "?GSH" described below.

command type:	GSN															
function:	Get Serial Number															
explication:	Ask for the controller's serial number															
byte number:	**1**	**2**	**3**	**4**	**5**	**...**	**...**	**...**	**n**							
command:	?	G	S	N	cr											
answer:	!	G	S	N	*serial number*				cr							

4.4 Get Serial Number Head *(PhoxX® lasers and LEDMOD.v2 only)*

- This command is valid for PhoxX® lasers and LEDMOD.v2 modules only.

This command gives back the serial number of the separated laser/led head.

Since the LEDMOD.v2 is divided into head and controller modules within a LedHUB® this command identifies the connected LED head.

The serial number needs to be interpreted as a string. It may contain special characters like "/" or "-".

The heads serial number is for informative use only.

command type:	GSH															
function:	Get Serialnumber Head															
explication:	Ask for the head's serial number.															
byte number:	**1**	**2**	**3**	**4**	**5**	**...**	**...**	**...**	**n**							
command:	?	G	S	H	cr											
answer:	!	G	S	H	*serial number*				cr							

4.5 Get Spec Info

This command receives the specified power and the wavelength of the connected device.

These specifications are for informative use only.

With light engines like the LedHUB® this command gives back important additional information. Please refer to chapter "Controlling Light Engines" for details.

command type:	GSI															
function:	Get Spec Info															
explication:	Ask for the Specs of the Laser/led device															
byte number:	**1**	**2**	**3**	**4**	**5**	**...**	**...**	**...**	**...**	**...**	**...**	**...**	**...**	**n**		
command:	?	G	S	I	cr											
answer:	!	G	S	I	*wavelength*				§	*spec power*				cr		

- The wavelength is an integer value with the entity nm (nano meters).
  White light LED modules will give the color temperature in K (Kelvin) instead.
- The spec power is an integer value with the entity mW (milli watts).

IMPORTANT: The spec power value is for informative use only and should not be used to calculate any percentage ↔ power value conversions. This should be done with the "Maximum Power" parameter available by the "Get Maximum Power"-command.

4.6 Get Maximum Power

This command gives back the truly maximum available power of the laser/led system.

This parameter may differ from the "Spec Power" parameter. For example if a laser is equipped with a direct fiber output option you will not reach the specified power after the fiber because of damping issues. Since we are interested in the power of the whole laser system the maximum power will be slightly below the specified diode power in that case.

command type:	GMP															
function:	Get Maximum Power															
explication:	Get the maximum laser/led power value															
byte number:	**1**	**2**	**3**	**4**	**5**	**...**	**...**	**n**								
command:	?	G	M	P	cr											
answer:	!	G	M	P	*max power*			cr								

- The "maximum power" parameter may be used to calculate percentage ↔ power value conversions and a power area boundary for the Get/Set-power commands.

4.7 Get Working Hours

This command receives the working hours of the laser diode or led.

The working hours are counting if the laser/led is in LASER-ON/LED-ON state.

command type:	GWH															
function:	Get Working Hours															
explication:	Ask for the working hours of the device															
byte number:	**1**	**2**	**3**	**4**	**5**	**...**	**...**	**...**	**n**							
command:	?	G	W	H	cr											
answer:	!	G	W	H	*working hours*				cr							
ad-hoc message:	$	G	W	H	*working hours*				cr							

- The "working hours" value is an integer counted in whole hours.
- If the ad-hoc mode is enabled for USB you will receive a message each passing hour.

4.8 Measure Diode Power

This command measures the actual laser power via the internal light pickoff.

➢ For LEDMOD.v2 the command is implemented since firmware v0.565 although it is not equipped with a light pickoff. The diode power is internally calculated by the measured diode current instead.

command type:	MDP															
function:	Measure Diode Power															
explication:	Measure the emitted laser diode power															
byte number:	**1**	**2**	**3**	**4**	**5**	**...**	**...**	**n**								
command:	?	M	D	P	cr											
answer:	!	M	D	P	*diode power*			cr								
ad-hoc message:	$	M	D	P	*diode power*			cr								

- The measured diode power is given in milliwatts as a floating point value with 2 decimal places.
- If the ad-hoc mode is enabled for USB you will receive a message every time the value has changed. In that case there is no need to poll this command in a closed loop.

4.9 Measure Temperature Diode

This command measures the actual temperature of the laser diode/led.

command type:	MTD															
function:	Measure Temperature diode															
explication:	Measure the temperature of the laser diode/led															
byte number:	**1**	**2**	**3**	**4**	**5**	**6**										
command:	?	M	T	D	cr											
answer:	!	M	T	D	*temperature*						cr					
ad-hoc message:	$	M	T	D	*temperature*						cr					

- The measured laser diode/led temperature is given in degrees centigrade as a floating point value with 1 decimal place.
- If the ad-hoc mode is enabled for USB you will receive a message every time the value has changed. In that case there is no need to poll this command in a closed loop.

4.10 Measure Temperature Ambient

This command measures the actual temperature inside the laser/led head.

command type:	MTA															
function:	Measure Temperature ambient															
explication:	Measure the value of the ambient temperature in the laser/led head															
byte number:	**1**	**2**	**3**	**4**	**5**	**...**	**...**	**...**	**...**	**...**	**n**					
command:	?	M	T	A	cr											
answer:	!	M	T	A	*temperature*						cr					
ad-hoc message:	$	M	T	A	*temperature*						cr					

- The measured ambient temperature is given in degrees centigrade as a floating point value with 1 decimal place.
- For a successful startup this temperature needs to be between -15°C and 50°C for laser systems and between -15°C and 80°C for led systems. Otherwise the laser/led will fall into internal interlock state. In that case the laser/led needs a reset command after the temperature has achieved the valid range. Anyway if a laser is already running a lockout will occur at 65°C (led systems at 85°C) but a warning message should be given at 50°C by the control software.
  -> If the laser/led is running hot please check the cooling airflow.

- If the ad-hoc mode is enabled for USB you will receive a message every time the value has changed. In that case there is no need to poll this command in a closed loop.

4.11 Measure Temperature Board *(PhoxX® lasers only)*

This command measures the actual temperature at the PCB inside the laser controller.

command type:	MTB															
function:	Measure Temperature board															
explication:	Measure the value of the PCB temperature in the laser controller.															
byte number:	**1**	**2**	**3**	**4**	**5**	**...**	**...**	**...**	**...**	**...**	**n**					
command:	?	M	T	B	cr											
answer:	!	M	T	B	*temperature*						cr					
ad-hoc message:	$	M	T	B	*temperature*						cr					

- The measured board temperature is given in degrees centigrade as a floating point value with 1 decimal place.
- For the startup this temperature needs to be between -15°C and 65°C. Otherwise the laser will fall into internal interlock state. In that case the laser needs a reset command after the temperature has achieved the valid range. Anyway if the laser is already running a lockout will occur at 80°C but a warning message should be given at 65°C by the control software.
  -> If the laser controller is running hot please check the cooling airflow.
- If the ad-hoc mode is enabled for USB you will receive a message every time the value has changed. In that case there is no need to poll this command in a closed loop.

4.12 Get Actual Status

This command gives back the actual operating status of the laser/led.

command type:	GAS
function:	Get Actual Status
explication:	Ask for the actual laser status

byte number:	1	2	3	4	5	...	...	...	n							
command:	?	G	A	S	cr											
answer:	!	G	A	S	*actual status*				cr							
ad-hoc message:	$	G	A	S	*actual status*				cr							

- The actual status is binary coded in an integer value. The value is given in hexadecimal notation.

Bit No#	15	14	13	12	11	10	9	8
Status	Reserved	Reserved	External Sensor connected	Reserved	Reserved	Reserved	System Power	Toggle Key
Bit No#	7	6	5	4	3	2	1	0
Status	Key Switch	Laser/led enable	Reserved	Attention	Reserved	Preheating	Laser/led ON	Error State

- **Bit 13 – external sensor connected:**
  This bit is set, if an external light sensor is connected to the device. (LedHUB controller only)

- **Bit 9 – system power:**
  If the bit is set, the laser/led system is powered-up. This will happen automatically if the laser/led is in auto power-up mode (default state).

- **Bit 8 – toggle key switch:**
  This bit relies to laser CDRH operation only. If the bit is set, a key-switch toggle is needed to release laser operation.

- **Bit 7 – key-switch state:**
  This bit represents the state of the key-switch input pin at the control-port.

- **Bit 6 – laser-enable/led-enable state (electronic shutter):**
  This bit represents the state of the laser-enable/led-enable input pin at the control-port.
  Note: if the laser-enable/led-enable input is not connected it will stay active and the bit is set.
  LedHUB: the bit is set if "shutter" on the front panel is set to "open".

- **Bit 4 – attention:**
  This bit is signalized if a situation occurred that needs special attention.
  LedHUB controller: one or more channels are in interlock state. In this situation the functionality of the LedHUB is restricted, but it still may be

operated with the remaining wavelengths.
QuixX: The bit is set if the laser is in pulse mode and triggered by the external digital input, but the externally applied frequency is far from the given set point. (see QuixX manual for details)
For all other devices this bit is reserved.

- **Bit 2 – preheating:**
  This bit indicates if the device is actually preheating. This is a temporary state. If the laser/led is already switched on, the "Laser-ON"/ "Led-ON" bit will be signaled beside the "preheating" bit but the laser/led will not emit light during this situation. Immediately after the diode temperature has reached the valid range the laser/led will start into operation.

- **Bit 1 – Laser ON/Led-ON:**
  If the bit is set the laser/led is switched on and the working hours are counting. Note: there are some other dependencies that may prevent the laser/led from emitting light. Please refer to the "Best practices" chapter for details.

- **Bit 0 – error state:**
  This bit indicates whether any preceded or pending error prevents the device from starting into normal operation. Only if this bit is unset the laser/led will operate as expected. Please refer to the "error handling" chapter for details.

- **Bit 15/14/12-10/5/3 – reserved:**
  These bits are not allocated by any function and should be ignored.

- If the ad-hoc mode is enabled for USB you will receive a message every time a state has changed. In that case there is no need to poll this command in a closed loop.

4.13 Get Failure Byte

This command gives back the actual error status of the device.

command type:	GFB
function:	Get Failure Byte
explication:	Ask for the actual error status of the device

byte number:	**1**	**2**	**3**	**4**	**5**	**…**	**…**	**…**	**n**
command:	?	G	F	B	cr				
answer:	!	G	F	B	*error status*				cr
ad-hoc message:	$	G	F	B	*error status*				cr

Please refer to the "Get Latched Failure" chapter below for a detailed description of the "error status" return value.

- The "Get Failure Byte" command returns the actual pending error state. This enables you to check if any static factor actually would prevent you from successfully resetting the device. Only if all bits, except the "error state" Bit, are unset a "Reset Controller" command is promising.
- The "Get Failure Byte" command is for receiving detailed information about pending error sources only. If the "Get Actual Status" command does not signal an "error state" on bit 0, there is no need to query the "Get Failure Byte" command.
- Some reasons for a laser/led system lockout are present for a short time only (for example an over current). These errors are stored in the "Latched Failure Byte" only and will not be catchable with the "Get Failure Byte" command.
- In contrast to what the name of the command implies, the error status is not a single byte but a 16bit integer value given in hexadecimal notation.
- If the ad-hoc mode is enabled for USB you will receive a message every time the error state has changed.

4.14 Get Latched Failure

This command gives back the error that caused an actual lockout.

command type:	GLF															
function:	Get Latched Failure															
explication:	Ask for the latched error of the laser/led that caused a lockout															
byte number:	**1**	**2**	**3**	**4**	**5**	**...**	**...**	**...**	**n**							
command:	?	G	L	F	cr											
answer:	!	G	L	F	*error status*				cr							

- The "Get Latched Failure" command is suitable for error logging purposes since these error bits are cleared after a system reset only.
- If the "Get Actual Status" command does not signal an "error state" on bit 0 there is no need to query the "Get Latched Failure" command since all bits will be zero.
- The error status is binary coded in an integer value. The value is given in hexadecimal notation.

Bit No#	15	14	13	12	11	10	9	8
Status	Diode Power	Internal Error	Test error	Diode Temp.	Ambient Temp.	Diode Current	External Interlock	Under/over Voltage
Bit No#	7	6	5	4	3	2	1	0
Status	High Power	K1 Relay Error	Internal Com. Error	CDRH Error	Reserved	Reserved	Reserved	Error State

- **Bit 15 – diode power:**
  The diode power exceeded the maximum allowed value.

- **Bit 14 – internal error:**
  An internal error occurred. This indicates serious electronic problems.

- **Bit 13 – test error:**
  The test error was triggered.
  This test error can be triggered by sending "?TIS" (test interlock state).

- **Bit 12 – diode temperature:**
  The diode temperature exceeded the valid temperature range.
  (still exceeds if bit is set in "Get Failure Byte" command)

- **Bit 11 – ambient temperature:**
  The ambient temperature in the laser head exceeded the valid temperature range.
  (still exceeds if bit is set in "Get Failure Byte" command)

- **Bit 10 – diode current:**
  The diode current exceeded the maximum allowed value.

- **Bit 9 – external interlock:**
  The external interlock loop was open.
  (it is still open if this bit is set in "Get Failure Byte" command)
  Note: if the "Auto Reset" function is active this will also be signalized in the "Latched Failure" as long the interlock loop is still open, since the device will automatically reset itself after the interlock is closed again.

- **Bit 8 – under/over voltage:**
  an under voltage or overvoltage occurred.
  (is still pending if bit is set in "Get Failure Byte" command)

- **Bit 7 – High Power Controller needed:** (Possible with PhoxX lasers only)
  some diodes of PhoxX lasers need a specially signed "high power controller". If you own high and low power PhoxX lasers and do mix up the controllers this bit will indicate that the actual connected low power controller is not suitable to drive the connected high power laser head.

- **Bit 6 – K1 relay error:** (Possible with PhoxX lasers only)
  An internal error occurred (the K1 relay did not operate). This indicates serious electronic problems.

- **Bit 5 – internal communication error:**
  A controller<->head communication error occurred.
  With PhoxX lasers this mostly indicates that the laser head is not connected correctly to the controller or the cable is defect.
  Otherwise this indicates serious electronic problems.

- **Bit 4 – CDRH error:**
  This error bit may be signaled in two situations:
  - a laser is configured as CDHR compliant but no CDRH-kit is connected to the laser
  - a laser is not configured as CDHR compliant (OEM) but a CDRH-kit is connected
- **Bit 0 – error state:**
  This bit indicates that the laser system is in internal error state (safety lockout). This bit is signalized in the "Get Actual Status" value (bit 0), too.

Note: the bits 6, 10, 13 and 15 can show up in the "Latched Failure" only, since these errors will never stay after the lockout occurred (see previous command).

4.15 Get/Set Level Power

This command determines the actual stored set point value for the optical laser/led power:

command type:	GLP															
function:	Get Level Power value															
explication:	Get the value set with SLP or SPP command															
byte number:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**								
command:	?	G	L	P	cr											
answer:	!	G	L	P	*power value*			cr								
ad-hoc message:	$	G	L	P	*power value*			cr								

This command sets the laser/led power value:

command type:	SLP														
function:	Set Level Power														
explication:	Set the laser power														
byte number:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**							
command:	?	S	L	P	*power value*			cr							
answer success:	!	S	L	P	>	cr									
answer failed:	!	S	L	P	x	cr									

- The laser/led power value is an integer in hexadecimal notation from 000 to $FFF_{hex,}$ representing a percental value from 0…100% in 4096 steps. The resulting optical power depends on the diode and may be calculated with use of the "Maximum Power" parameter:

*Actual laser/led power set point = GMP * GLP / 4095*

Note: for QuixX lasers this calculation is not valid in pulse mode.

- The adjusted optical power is emitted if the laser/led is in "on"-state and CW-mode is activated or digital modulation mode is activated and the external input is high.
  If analog modulation mode is enabled the power level indicates the maximum level that is reachable with 100% analog voltage input. That means the analog input is scaled down by this parameter if it's not FFF_{hex}.
- The Set-command will fail if the parameter is not a valid hexadecimal value or the value is not in the given range.
- If supported by your device you may use the "Get/Set Power as Percentage" commands alternatively.
- The value is stored in a remanent memory so it will be valid after a reset or after toggling the mains.
  NOTE: applications that do need to set new power values continuously (cycles more than 10.000 times per year) should use the "Get/Set Temporary Power" command to save memory lifetime.
- **IMPORTANT:** if the "Temporary Power" command is supported by your device and you did set a different temporary value, "GLP" reads the non-volatile stored value only. In this case "GLP" would not give the actual power level. "SLP" will reset the temporary value to be equal (See chapter "Get/Set Temporary Power" for details).
- If the ad-hoc mode is enabled for USB you will receive a message from the GLP command if you are using the RS232 port in parallel and setting a new value via RS232.

4.16 Get/Set Power as Percentage

- This commands are available for:
  - PhoxX® (Firmware >= 2.83)
  - LuxX® (Firmware >= 2.0)
  - LuxX®.HSA
  - LuxX+®
  - BrixX® (Firmware >= 1.60)
  - BrixX®.UHP
  - QuixX®
  - LEDMOD.v2

This command determines the actual stored set point value for the optical laser/led power:

command type:	GPP
function:	Get Power setpoint as percentage
explication:	Get the value set with SPP or SLP command

byte number:	**1**	**2**	**3**	**4**	**5**	**...**	**...**	**n**
command:	?	G	P	P	cr			
answer:	!	G	P	P	*power value*			cr
ad-hoc message:	$	G	P	P	*power value*			cr

This command sets the laser/led power value:

command type:	SPP
function:	Set Power as Percentage
explication:	Set the laser/led power

byte number:	**1**	**2**	**3**	**4**	**5**	**6**	**...**	**n**
command:	?	S	P	P	power value			cr
answer success:	!	S	P	P	>	cr		
answer failed:	!	S	P	P	x	cr		

- The laser/led power value is a floating point number representing the percentage value from 0…100%.
  The resulting optical power depends on the diode and may be calculated with use of the "Maximum Power" parameter:

*Actual laser/led power set point = GMP * GPP / 100*

Note: for QuixX this calculation is not valid in pulse mode.

- The adjusted optical power is emitted if the laser/led is in "on"-state and CW-mode is activated or digital modulation mode is activated and the external input is high.
  If analog modulation mode is enabled the power level indicates the maximum level that is reachable with 100% analog voltage input. That means the analog input is scaled down by this parameter if it's not 100.
- The Set-command will fail if the parameter is not a floating point value between 0.0 and 100.0.
- These commands may be used alternatively to the "Get/Set Level Power" commands.
- The value is stored in a remanent memory so it will be valid after a reset or after toggling the mains.
  NOTE: applications that do need to set new power values continuously (cycles more than 10.000 times per year) should use the "Get/Set Temporary Power" command to save memory lifetime.
- **IMPORTANT:** if the "Temporary Power" command is supported by your device and you did set a different temporary value, "GPP" reads the non-volatile stored value only. In this case "GPP" would not give the actual power level. "SPP" will reset the temporary value to be equal (See chapter "Get/Set Temporary Power" for details).
- If the ad-hoc mode is enabled for USB you will receive a message from the GPP-command in the case you are using the RS232 port in parallel and set a new power value via RS232.

4.17 Get/Set Temporary Power

➢ For availability of this command see previous chapter.

The command gets/sets the laser/led power value.

The value is **NOT** stored in a remanent memory so it is lost after a reset or after toggling the mains. The device will always boot-up with the last value that was set by a "Set Level Power" or "Set Power as Percentage" command.

This command is intended for use in applications that do need to set new power values continuously (cycles more than 10.000 times per year) to save memory lifetime.

command type:	TPP															
function:	Temporary Power as Percentage															
explication:	Set the laser/led power tempoarily															
byte number:	**1**	**2**	**3**	**4**	**5**	**6**	**...**	**n**								
set command:	?	T	P	P	power value			cr								
answer success:	!	T	P	P	>	cr										
answer failed:	!	T	P	P	x	cr										
query command:	?	T	P	P	cr											
query answer:	!	T	P	P	power value			cr								
ad-hoc message:	$	T	P	P	power value			cr								

- The laser/led power value needs to be given as a floating point number from 0.0…100.0.
  The resulting optical power depends on the diode and may be calculated with use of the "Maximum Power" parameter.
- The optical power is emitted if the laser/led is in "on"-state and a CW-only mode is activated or digital modulation mode is activated and the input level is high.
  If analog modulation mode is enabled the power level indicates the maximum level that is reachable with 100% analog voltage input. That means the analog input is scaled down by this parameter if it's not 100.
- The command will fail if the parameter is not a floating point value between 0.0 and 100.0.
- Use the command without a parameter ("?TPP*cr*") to query the actual value.
- If the ad-hoc mode is enabled for USB you will receive a message in the case you are using the RS232 port in parallel and set a new temporary power value via RS232. You will also receive an ad-hoc message if you send a "Set Level Power" or "Set Power as Percentage" command, since this will reset the temporarily value to be equal.

4.18 Get/Set Operating Mode

This command gives back the stored operating mode settings:

QuixX lasers have special handling for changing the operating mode. Please refer to chapter "Change Operating Mode" for details.

command type:	GOM															
function:	Get operating mode															
explication:	Ask for the bit combination showing the actual operating mode settings.															
byte number:	**1**	**2**	**3**	**4**	**5**	**...**	**...**	**...**	**n**							
command:	?	G	O	M	cr											
answer:	!	G	O	M	*operating mode*				cr							
ad-hoc message:	$	G	O	M	*operating mode*				cr							

This command sets a new operating mode:

command type:	SOM															
function:	Set operating mode															
explication:	Set operating modes in one go															
byte number:	**1**	**2**	**3**	**4**	**5**	**6**	**...**	**...**	**n**							
command:	?	S	O	M	*operating mode*				cr							
answer success:	!	S	O	M	>	cr										
answer failed:	!	S	O	M	*x*	cr										

- The operating mode is binary coded in an integer value. The value is given in hexadecimal notation.

Bit No#	15	14	13	12	11	10	9	8
Mode	Auto Power-up	Auto Startup	USB Ad-hoc Mode	Analog Input impedance	Digital Input impedance	Reserved	Reserved	APC mode
Bit No#	7	6	5	4	3	2	1	0
Mode	Analog input release	Reserved	Digital input release	operating level release	bias level release	Internal clock generator (ICG)	Reserved	Reserved

- **Bit 15 – auto power-up:**
  If the bit is set, the laser/led system will power up all electronic circuits directly after it is connected to the mains or after a reset. (default state)
  If the bit is not set the laser/led system will boot into energy save mode where serial communication is possible only (e.g. no temperature control is enabled). In that case the system will power-up after you sent the "Power On"-command. This functionality is for handling a larger amount of

lasers/leds. To avoid power peaks on the supply line the lasers/leds may be powered on program-controlled in sequence.

- **Bit 14 – auto startup:**
  If the bit is set, the laser/led system will automatically switch to "On"-state after the controller is connected to the mains or after a reset. If auto power-up is not set beside auto startup a "Power On"-command triggers the auto startup.
  Note: Auto startup will not work, if a laser system is in CDRH mode since in this case a manual key toggle is necessary. The laser will then start emitting after the key toggle is executed.

- **Bit 13 – USB ad-hoc mode:**
  If the bit is set, you will receive "Ad-hoc messages" on the USB port any time a parameter or state changes without being requested by the PC (default state). This enables you to stay up to date without polling a large amount of "Get-commands" in a closed loop. However if your program logic does not work together with this philosophy you may switch off this behavior.

Be aware that using the OCC together with any device through its USB port will enable the ad-hoc mode of each device. This configuration will stay until you disable it again by your software.

- **Bit 12 – analog input impedance:**
  This bit selects the input impedance and voltage range for the analog modulation input.
  - Bit set: input voltage range 0..5V, impedance 1.2kΩ
  - Bit unset: input voltage range 0..1V, impedance 50Ω
    - ➢ This bit is unused in LuxX, LuxX.HSA lasers and LEDMOD.v2

- **Bit 11 – digital input impedance:**
  This bit selects the input impedance and voltage range for the digital modulation input.
  - Bit set: input voltage range 0..5V/TTL, impedance 200Ω
  - Bit unset: input voltage range 0..1V, impedance 50Ω
    - ➢ This bit is unused in LuxX, BrixX lasers and LEDMOD.v2

- **Bit 8 – APC / ACC mode selector:**
  - Bit set: APC mode active (auto power control)
  - Bit unset: ACC mode active (auto current control)

- Since PhoxX lasers and led devices are not equipped with an APC mode feature this bit stays 0 for PhoxX lasers and led devices.
- Since QuixX lasers have a separate command for changing the operating mode this bit is unused for QuixX lasers. (see chapter "Change Operating Mode" for details)

- **Bit 7 – analog input release:**
  If the bit is set, the analog input is enabled. The laser/led will not emit light until an analog input signal is applied.
- **Bit 5 – digital input release:**
  If the bit is set, the digital input is enabled. The laser/led will not emit light until a digital-high input signal is detected.
  - This bit is unused in LuxX lasers and stays 0
- **Bit 3 and 4 – current sources release:**
  This bits signalize whether the current sources for the laser diode/led are released or not. These two bits always have an equal state: if only one bit is set by the "?SOM"-command, the other will be set automatically. Light might be emitted only if these bits are set.
  The unset state is called "emission standby-mode".
- **Bit 2 – Internal Clock Generator (ICG):**
  If this bit is set the ICG is enabled. If so, the optical output is modulated or pulses are triggered by an internally generated digital signal. The frequency of the ICG is controlled by the "Get/Set ICG Frequency"-command.
  - This bit is available for QuixX and LEDMOD.v2 only.
- **Bit 10/9/6/3/1/0 – reserved:**
  These bits may have varying states and should not be changed by the software to avoid unexpected behavior.

- **NOTE**: a QuixX laser will emit laser pulses in pulse mode only, if at least one of the following terms is given:
  - The internal clock generator is enabled (bit 2)
  - The digital input is released (bit 5) and an external modulation signal is applied
- The Set-command will fail if the "operating mode" parameter is not a valid hexadecimal value.
- If the ad-hoc mode is enabled for USB you will receive a message if you are using the RS232 port in parallel and setting a new value via RS232.

Always readout the actual operating mode parameter with "?GOM" first. Manipulate the desired bit in memory and write back the new value with "?SOM". Be aware of using the "?SOM"- command with predefined static hex values (without asking for the actual value by "?GOM" first), since this may force some reserved bits to toggle and may result in unexpected behavior.

"Get/Set Operating Mode" are nutshell commands.
If you need to deal with single states only or for easy manual handling by a terminal program it may be an easier way to use the following wrapper commands witch will readout or manipulate the desired bits only:

(If you are using "Get/Set Operating Mode" the following commands are redundant)

4.18.1 Set Auto Power-up

This command sets the auto power-up behavior.

command type:	SAP
function:	Set auto power-up
explication:	Set the auto power-up bit

byte number:	**1**	**2**	**3**	**4**	**5**	**6**
command:	?	S	A	P	p1	cr
answer success:	!	S	A	P	>	cr
answer failed:	!	S	A	P	*x*	cr

- If this operating mode is activated the system will power up all electronic circuits directly after the device is connected to the mains or after a reset command (default state) .
  If deactivated the system will boot into energy save mode where serial communication is possible only (e.g. no temperature control is enabled). In that case the system will power-up after you sent the "Power On"-command. This functionality is for handling a larger amount of lasers/leds. To avoid power peaks the devices may be powered on program-controlled in sequence.
- Valid values for the parameter "p1" are "0" (off) and "1" (on). All other values will be refused.
- You may query the actual state of this mode by sending the command without a parameter ("?SAP*cr*").

- If the ad-hoc mode is enabled for USB you will receive a message from the "Get operating mode"-command since this will change the "auto power" bit for the operating mode.

4.18.2 Set Auto Startup

This command sets the auto startup behavior.

command type:	SAS															
function:	Set Auto Start															
explication:	Set the Auto Start Function ON or OFF															
byte number:	**1**	**2**	**3**	**4**	**5**	**6**										
command:	?	S	A	S	p1	cr										
answer success:	!	S	A	S	>	cr										
answer failed:	!	S	A	S	*x*	cr										

- If this operating mode is activated, the laser/led will be switched on directly after the device is connected to the mains or after a reset command. (a slight "preheating" delay may occur)
  If deactivated, you need to send a "Laser On"/"Led ON" command to switch on the laser/led.
  If auto power-up is not set beside auto startup a "Power On"-command triggers the auto startup.
  Note: Auto startup will not work, if a laser system is in CDRH mode since in this case a manual key toggle is necessary. The laser will then start emitting after the key toggle is executed.

- Valid values for the parameter "p1" are "0" (off) and "1" (on). All other values will be refused.

- You may query the actual state of this mode by sending the command without a parameter ("?SAS*cr*").

- If the ad-hoc mode is enabled for USB you will receive a message from the "Get operating mode"-command since this will change the "auto startup" bit for the operating mode.

4.18.3 Set Impedance Digital *(not valid for LuxX®, BrixX® lasers and LEDMOD.v2)*

This command selects the input impedance and voltage range for the digital modulation input.

command type:	SID															
function:	Set Impedance Digital															
explication:	Set the impedance and voltage range of the digital input															
byte number:	**1**	**2**	**3**	**4**	**5**	**6**										
command:	?	S	I	D	p1	cr										
answer success:	!	S	I	D	>	cr										
answer failed:	!	S	I	D	*x*	cr										

- Valid values for the parameter "p1" are:
  - "1": input voltage range 0..5V/TTL, impedance 200Ω
  - "0": input voltage range 0..1V, impedance 50Ω

  All other values will be refused.
- You may query the actual state of this mode by sending the command without a parameter ("?SID*cr*").
- If the ad-hoc mode is enabled for USB you will receive a message from the "Get operating mode"-command since this will change the "digital impedance" bit for the operating mode.

4.18.4 Set Impedance Analog *(not valid for LuxX®, LuxX.HSA lasers and LEDMOD.v2)*

This command selects the input impedance and voltage range for the analog modulation input.

command type:	SIA															
function:	Set Impedance Analog															
explication:	Set the impedance and voltage range of the analog input															
byte number:	**1**	**2**	**3**	**4**	**5**	**6**										
command:	?	S	I	A	p1	cr										
answer success:	!	S	I	A	>	cr										
answer failed:	!	S	I	A	*x*	cr										

- Valid values for the parameter "p1" are:
  - "1": input voltage range 0..5V, impedance 1.2kΩ
  - "0": input voltage range 0..1V, impedance 50Ω

  All other values will be refused.

- You may query the actual state of this mode by sending the command without a parameter ("?SIA*cr*").

- If the ad-hoc mode is enabled for USB you will receive a message from the "Get operating mode"-command since this will change the "analog impedance" bit for the operating mode.

4.18.5 Recall Operating Mode

This command sets some bits of the operating mode through indexed presets.
(Operating and bias level release, ACC/APC, analog/digital input releases and ICG activation)

Note: the index numbers of this command are not unified for all lasers and led types!
If you plan to operate several device types by a single software it is better to use the "Set Operating Mode" command since the bit-positions are fixed.

command type:	ROM
function:	Recall Operating Mode
explication:	Recall an Operating Mode

byte number:	**1**	**2**	**3**	**4**	**5**	**6**										
command:	?	R	O	M	p1	cr										
answer:	!	R	O	M	>	cr										
answer failed:	!	R	O	M	*x*	cr										

- Valid values for the parameter "p1" with *PhoxX® lasers* are:
  - "0": emission standby (the diode current source is set to 0)
  - "1": cw only – no modulation active
  - "2": digital modulated
  - "3": analog modulated
  - "4": digital and analog modulated

  All other values will be refused.

- Valid values for the parameter "p1" with *LuxX® lasers* are:
    - "0": emission standby (the diode current source is set to 0)
    - "1": ACC mode – no modulation active
    - "2": APC mode – no modulation active
    - "3": ACC mode – analog modulated

  All other values will be refused.

- Valid values for the parameter "p1" with *LuxX+®, LuxX®.HSA, BrixX® and BrixX®.UHP lasers* are:
    - "0": emission standby (the diode current source is set to 0)
    - "1": ACC mode – no modulation active
    - "2": APC mode – no modulation active
    - "3": ACC mode – digital modulated
    - "4": ACC mode – analog modulated
    - "5": ACC mode – digital and analog modulated

  All other values will be refused.

- Valid values for the parameter "p1" with *LEDMOD.v2* are:
    - "0": emission standby (the diode current source is set to 0)
    - "1": cw only – no modulation active
    - "2": digital modulated
    - "3": analog modulated
    - "4": digital and analog modulated
    - "5": Internally PWM modulated - internal clock generator active (ICG)
    - "6": Internally PWM modulated - gated by external digital input
    - "7": Internally PWM modulated - externally modulated by analog input
    - "8": Internally PWM modulated - gated by digital input + externally modulated by analog input

  All other values will be refused.

- Valid values for the parameter "p1" with *QuixX®* lasers (Firmware >= 3.14) are:
    - "0": emission standby (the diode current source is set to 0)
    - "1": CW ACC mode – no modulation active
    - "2": APC mode – no modulation active
    - "3": CW ACC mode – digital modulated
    - "4": CW ACC mode – analog modulated
    - "5": CW ACC mode – digital and analog modulated
    - "6": Internally PWM modulated CW ACC mode – internal clock generator active (ICG)
    - "7": Internally PWM modulated CW ACC mode - gated by external digital input
    - "8": Internally PWM modulated CW ACC mode - externally modulated by analog input
    - "9": Internally PWM modulated CW ACC mode - gated by digital input + externally modulated by analog input
    - "10": Single shot low power pulse mode - triggered by digital input (sync in)
    - "11" Single shot low power pulse mode - triggered by digital input (sync in) + external analog modulation of the pulse height
    - "12" Continuous low power pulse mode
    - "13" Continuous low power pulse mode - gated by digital input
    - "14" Continuous low power pulse mode - externally modulated by analog input
    - "15" Continuous low power pulse mode - gated by digital input + externally modulated by analog input
    - "16": Single shot high power pulse mode - triggered by digital input (sync in)
    - "17" Single shot high power pulse mode - triggered by digital input (sync in) + external analog modulation of the pulse height
    - "18" Continuous high power pulse mode
    - "19" Continuous high power pulse mode - gated by digital input
    - "20" Continuous high power pulse mode - externally modulated by analog input

- "21" Continuous high power pulse mode - gated by digital input + externally modulated by analog input
- "22": Single shot pulse mode with adjustable pulse shape - triggered by digital input (sync in)
- "23" Single shot pulse mode with adjustable pulse shape - triggered by digital input (sync in) + external analog modulation of the pulse height
- "24" Continuous pulse mode with adjustable pulse shape
- "25" Continuous pulse mode with adjustable pulse shape - gated by digital input
- "26" Continuous pulse mode with adjustable pulse shape - externally modulated by analog input
- "27" Continuous pulse mode with adjustable pulse shape - gated by digital input + externally modulated by analog input

All other values will be refused.

- You may query the actual state of this mode by sending the command without a parameter ("?ROM*cr*").
- If the ad-hoc mode is enabled for USB you will receive a message from the "Get operating mode"-command since this will change the modulation bits for the operating mode.

4.19 Change Operating Mode *(QuixX® only)*

➢ This command is available for QuixX® lasers only.

Since the QuixX is equipped with 5 separate operating modes (excluding the combination of modulation possibilities) this modes can be selected by the Change Operating Mode command.

command type:	COM															
function:	Change Operating Mode															
explication:	Selects an operating mode															
byte number:	**1**	**2**	**3**	**4**	**5**	**6**	**...**	**n**								
set command:	?	C	O	M	p1	cr										
answer success:	!	C	O	M	>	cr										
answer failed:	!	C	O	M	x	cr										
query command:	?	C	O	M	cr											
query answer:	!	C	O	M	p1	cr										
ad-hoc message:	$	C	O	M	p1	cr										

- The parameter p1 is the index of the operating mode.

  The 5 operating modes are:

  1) cw – ACC mode
  2) cw – APC mode
  3) pulse mode - low power
  4) pulse mode - high power
  5) pulse mode - expert (pulse mode with selectable pulse shape)

for QuixX lasers the behavior of the commands "Get/Set Operating Mode" differ as follows:

1. Bit 8 (APC mode) is not used (reserved), since the APC mode is selected within the Change Operating Mode command.
2. Bits 2, 5 and 7 of GOM/SOM are stored for each operating mode selected with COM separately. This means that selecting a combination of modulations inputs (Analog, Digital and ICG) will be stored for the actual operating mode only. The last state will be recalled within the COM command. This behavior is also for the power level (SPP/SLP) and the ICG Frequency.

- The command will fail if the parameter is not a value between 1 and 5.
- Use the command without a parameter ("?COM*cr*") to query the actual value.
- If the ad-hoc mode is enabled for USB you will receive a message in the case you are using the RS232 port in parallel and set a new operating mode via RS232.
  You will receive ad-hoc messages from other commands that are stored per operating mode.

4.20 Get/Set ICG Frequency *(QuixX® and LEDMOD.v2 only)*

➢ These commands are available for LEDMOD.v2 and QuixX® lasers only.

This command determines the actual stored frequency for the internal clock generator (ICG):

command type:	GPF															
function:	Get PWM/Pulse frequency															
explication:	Get the value set with SPF command															
byte number:	**1**	**2**	**3**	**4**	**5**	**...**	**...**	**n**								
command:	?	G	P	F	cr											
answer:	!	G	P	F	*frequency*			cr								
ad-hoc message:	$	G	P	F	*frequency*			cr								

This command sets the frequency for the internal clock generator (ICG):

command type:	SPF															
function:	Set PWM/pulse Frequency															
explication:	Set the frequency of the ICG															
byte number:	**1**	**2**	**3**	**4**	**5**	**6**	**...**	**n**								
command:	?	S	P	F	frequency			cr								
answer success:	!	S	P	F	>	cr										
answer failed:	!	S	P	F	x	cr										

- The frequency value is a floating point number representing the frequency in Hz. This value may be given in scientific notation.
- The Set-command will fail if the parameter is not a floating point value in between the valid range.
  For LEDMOD.v2 the range is valid from 0.1 to 200000.0.

For QuixX lasers the range needs to be queried by the "Get Frequency Limits" command.

- If the ad-hoc mode is enabled for USB you will receive a message from the GPF-command in the case you are using the RS232 port in parallel and set a new value via RS232.

4.21 Get Frequency Limits *(QuixX® only)*

➢ This command is available for QuixX® lasers only.

This command queries the frequency limits for the internal clock generator (ICG) at the actual operating mode:

command type:	LPF															
function:	Limit PWM/Pulse frequency															
explication:	Get the Limits for PWM/pulse frequency at actual operating mode															
byte number:	**1**	**2**	**3**	**4**	**5**	**…**	**…**	**…**	**…**	**…**	**…**	**n**				
command:	?	L	P	F	cr											
answer:	!	L	P	F	*lowest freq.*			§	*highest freq.*			cr				
ad-hoc message:	$	L	P	F	*lowest freq.*			§	*highest freq.*			cr				

Since QuixX® lasers have different frequency ranges for ACC PWM mode and the pulse modes you need to query the valid range for the actual operating mode with this command.

- The frequency limit values are floating point numbers representing the frequency in Hz.
  These values may be given in scientific notation.
- If the ad-hoc mode is enabled for USB you will receive a message from the LPF-command if you change the operating mode.

4.22 Get/Set ICG Duty Cycle *(LEDMOD.v2 only)*

➢ These commands are available for LEDMOD.v2 only.

This command determines the actual stored duty cycle for the internal clock generator (ICG):

command type:	GDC															
function:	Get duty cycle															
explication:	Get the value set with SDC command															
byte number:	**1**	**2**	**3**	**4**	**5**	**...**	**...**	**n**								
command:	?	G	D	C	cr											
answer:	!	G	D	C	*duty cycle*			cr								
ad-hoc message:	$	G	D	C	*duty cycle*			cr								

This command sets the duty cycle of the internal clock generator (ICG):

command type:	SPF															
function:	Set duty cycle															
explication:	Set the duty cycle of the ICG															
byte number:	**1**	**2**	**3**	**4**	**5**	**6**	**...**	**n**								
command:	?	S	D	C	duty cycle			cr								
answer success:	!	S	D	C	>	cr										
answer failed:	!	S	D	C	x	cr										

- The duty cycle value is a floating point number representing the duty cycle in %.
- The Set-command will fail if the parameter is not a floating point value between 0.0 and 100.0.
- If the ad-hoc mode is enabled for USB you will receive a message from the GDC-command in the case you are using the RS232 port in parallel and set a new value via RS232.

4.23 Set Power On

The command powers on the main electronic circuits.

command type:	POn															
function:	Power On															
explication:	Set system power on															
byte number:	**1**	**2**	**3**	**4**	**5**	**6**										
command:	?	P	O	n	cr											
answer:	!	P	O	n	>	cr										
answer failed:	!	P	O	n	x	cr										

- This command is necessary only if auto power-up mode is disabled (this is not by default) or the system power was switched off manually.

- If the command was successful the system power will be switched on and the diode temperature control starts into operation. This is a precondition for switching on the laser/led.
- The command will fail if the error state is active. Please refer to the "error handling" chapter for details.
- This command will not send ad-hoc messages.
  If the ad-hoc mode is enabled for USB you will receive a message from the "GetActualState"-command, since the power state is indicated by this command.

4.24 Set Power Off

The command powers down the main electronic circuits.

command type:	POf															
function:	Power Off															
explication:	Set system power off															
byte number:	**1**	**2**	**3**	**4**	**5**	**6**										
command:	?	P	O	f	cr											
answer:	!	P	O	f	>	cr										

- Switching off system power will result in switching off the laser/led, if it was switched on before. The system will remain in energy save mode where serial communication is possible only.
- This command will not fail.
- This command will not send ad-hoc messages.
  If the ad-hoc mode is enabled for USB you will receive a message from the "GetActualState"-command, since the power state is indicated by this command.

4.25 Set Laser On / Set LED On

This command sets the laser/led into "on"-state.

command type:	LOn
function:	Laser On / LED On
explication:	Set the Laser/led to Laser ON/Led ON Status

byte number:	**1**	**2**	**3**	**4**	**5**	**6**
command:	?	L	O	n	cr	
answer success:	!	L	O	n	>	cr
answer failed:	!	L	O	n	x	cr

- If the command was successful, the laser/led will be switched "on" and the working hours are counting.
  Note: there are some other dependencies that may prevent the laser/led from emitting light. Please refer to the "Best practices" chapter for details.
- The command will fail if the system power is not enabled. The command will fail also if the error state is active. Please refer to the "error handling" chapter for details.
- This command will not send ad-hoc messages.
  If the ad-hoc mode is enabled for USB you will receive a message from the "GetActualState"-command, since the laser ON state is indicated by this command.

4.26 Set Laser Off / Set LED Off

This command sets the laser/led into "off"-state.

command type:	LOf
function:	Laser Off / LED Off
explication:	Set the Laser/led to Laser OFF/LED OFF Status

byte number:	**1**	**2**	**3**	**4**	**5**	**6**
command:	?	L	O	f	cr	
answer:	!	L	O	f	>	cr

- After the laser/led is switched off the working hours will stop counting.
- This command will not fail.
- This command will not send ad-hoc messages.
  If the ad-hoc mode is enabled for USB you will receive a message from the "GetActualState"-command, since the laser ON state is indicated by this command.

4.27 Reset Controller

This command resets the device.

command type:	RsC															
function:	Reset Controller															
explication:	The command resets the laser/led controller															
byte number:	**1**	**2**	**3**	**4**	**5**	**6**										
command:	?	R	s	C	cr											
answer:	!	R	s	C	cr											
Reset ready msg:	$	R	s	C	>	cr										
ad-hoc message:	$	R	s	C	*r1*	cr										

- Since a reset is a process that takes some time to finish, this command is divided into two parts:
  - After sending the "?RsC" command you will promptly receive the answer "!RsC", meaning that the command was accepted and the device will reset now.
  - If the reset process is finished you will receive a "reset ready message" ("$RsC>") to signalize that the device is online again, no matter if the ad-hoc mode is enabled or not (the reset ready message is send also on RS232) **Important:** during reset, between "!RsC" and "$RsC>", you may receive some undefined characters produced by glitches of the device internal UART chip. Please assure to ignore all characters before the "$RsC>" message.
- During the reset process no communication is possible. Commands are not handled and may result in an "!UK" answer after the reset is finished.
- The "reset ready message" is send on USB and RS232 simultaneously. This will happen also after connecting the system to the mains (Initial reset).
- The "reset ready message" does not indicate that the prior reset was error free. Please query the "Get Actual Status" command to determine if any error occurred.
- If the ad-hoc mode is enabled for USB you will receive a message if you toggle the key-switch or you are using the RS232 port in parallel and initiate a reset via RS232 (do not confuse with the "reset ready message").
  The message is sent with a parameter "*r1*" identifying the initiating source:
  - "0": the reset was initiated by the key-switch
  - "3": the reset was initiated by RS232
  - "4": the reset was triggered by auto reset
- The "reset ready message" will automatically be followed by a "Get Actual Status" message if ad-hoc mode is enabled.

4.28 Auto Reset

The default behavior of all devices is that if a device is in interlock state because of an open external interlock input it needs to get reset manually by the "Reset Controller" command after the interlock loop is closed again.

If auto reset is enabled the device resets automatically after the external interlock input is closed again.

Enabling auto reset is possible with lasers up to class 3B only.

command type:	Ars															
function:	Auto Reset															
explication:	Set the Auto Reset behavior ON or OFF															
byte number:	**1**	**2**	**3**	**4**	**5**	**6**										
command:	?	A	R	s	p1	cr										
answer success:	!	A	R	s	>	cr										
answer failed:	!	A	R	s	*x*	cr										

- Auto reset will not work for class 4 lasers or if a laser system is in CDRH mode.
- Valid values for the parameter "p1" are "0" (off) and "1" (on). All other values will be refused.
- You may query the actual state of this mode by sending the command without a parameter ("?ARs").

4.29 Calibrate Laser Diode *(not valid for LEDMOD.v2)*

This command recalibrates the laser diode because of aging issues.

Please start calibration only if the laser shows a significant power drop.

If the laser does not achieve its specified laser power because of aging issues recalibrating will compensate the power drop by recalculating the maximum diode current and bias offset.

CAUTION: during the calibration progress the laser will emit laser light.

command type:	CLD															
function:	Calibrate Laser Diode															
explication:	Calibrates the Laser Diode maximum power and bias offset															
byte number:	**1**	**2**	**3**	**4**	**5**	**...**	**n**									
command:	?	C	L	D	cr											
answer:	!	C	L	D	>	cr										
Cal. Ready msg:	$	C	L	D	*c1*		cr									
ad-hoc message:	$	C	L	D	>	cr										

- Since the calibration process takes some time to finish this command is divided into two parts:
  - After sending the "?CLD" command you will promptly receive the answer "!CLD", meaning that the command was accepted and the controller will start calibrating now.
  - If the calibration process is finished you will receive a "calibrate ready message" ("$CLD*c1*") to signalize that the laser is online again, no matter if the ad-hoc mode is enabled or not (the calibrate ready message is send on RS232 also)
- The "calibrate ready message" provides the "*c1*" parameter, which indicates the calibration result:
  - "0": the calibration process finished successfully.
  - "1": the calibration process finished but the maximum possible diode current is reached. The laser could not be set to its specified power.
  - "2": the key-switch is off.
  - "3": the Laser-enable input is low.
  - "4": an interlock occurred during calibration.

- "5": the diode temperature is not in the required range.
- "6": a controller<->head communication error occurred.
- "7": fatal error: the bias is out of range.
- "8": fatal error: no bias point was found.
- "9": the new calibration result is less than 95% of the previous. This may happen in case of a back reflection into the laser head.
- "10": the laser is switched off.
- "11": no calibration sensor was found.
- "12": no light was detected on sensor.
- "13": an over-power condition occurred.

- During the calibration process no communication is possible. Commands are not handled and may result in an "!UK" answer after calibration is finished.
- The calibration process may take about 2 minutes.
- If the ad-hoc mode is enabled for USB you will receive a message if you are using the RS232 port in parallel and initiate a calibration via RS232 (do not confuse with the "calibrate ready message").
- Although sending any commands during the calibration process is not possible you will receive ad-hoc messages from the "measure diode power" command to monitor the progress, assuming that the ad-hoc mode is enabled for USB.

5. Best Practices

The previous list of commands is suitable to program full featured interface software.

Anyway, you may program a laser or led communication software that gets along with just a part of the given commands but will successfully control the device within your needs.

In the following list the commands are classified according to importance:

The first contact to a laser/led system always should start with a **GetFirmware** command followed by some more or less important queries:

GetFirmware *("?GFw")*	It's a good programming style to start with the identification of the device and its version.
GetFirmwareHead *("?GFH")* *(PhoxX® only)*	Additional identification command. Not essential for communication.
GetSerialNumber *("?GSN")*	Additional identification command. Not essential for communication.
GetSerialNumberHead *("?GSH")* *(PhoxX® only)*	Additional identification command. Not essential for communication.
GetSpecInfo *("?GSI")*	This command is important for „Multi-Channel – Single-Port" devices. Please refer to chapter "Controlling Light Engines" for details. For all other devices it is an additional identification command only. Not essential for controlling.
GetMaximumPower *("?GMP")*	Should be queried if the software needs to calculate the output power set-point in mW.
GetWorkingHours *("?GWH")*	Expired lifetime query command. Not essential for controlling.
GetLevelPower *("?GLP")* or GetPowerAsPercentage *("?GPP")* *or* TemporaryPower *("?TPP")*	Valuable for getting information about the actually stored power level.
GetOperatingMode *("?GOM")*	Should be queried for getting information about the actual activated operating modes. *(The information received by this command may also be queried separately by its wrapper-commands)*
ChangeOperatingMode *("?COM")*	Valid for QuixX lasers only: This command complements the SOM command.

This initial query sequence should be followed by a closed loop of polling the **GetActualStatus** command to be up-to-date with the "error state" and other circumstances (e.g. in a one second pattern).
A more recommend approach is to query the command once with the ad-hoc mode enabled. (see the "SetOperatingMode" command description for details)

GetActualStatus *("?GAS")*	Important to query. The return value is necessary to know about the error state and other circumstances.

If the "error state" is signaled by the "GetActualStatus" command query the following failure commands:

GetFailureByte *("?GFB")*	Query in a closed loop if "error state" is signaled by "GetActualStatus" to receive information about actual pending error sources. *(alternatively use ad-hoc mode)*
GetLatchedFailure *("?GLF")*	Query if "error state" is signaled by "GetActualStatus" to receive reliable information about the lockout reason if needed.
ResetController *("?RsC")*	Call to reset the "error state" after the error sources are cleared.

To monitor some potential error sources you may query the following commands:

MeasureTemperatureDiode *("?MTD")*	Additional command to monitor the laser diode temperature.
MeasureTemperatureAmbient *("?MTA")*	Additional command to monitor the ambient temperature inside the laser head.
MeasureTemperatureBoard *("?MTB")* *(PhoxX® only)*	Additional command to monitor the ambient temperature inside the laser controller.

Use the following commands to control the laser/led:

LaserOn / Led On *("?LOn")*		Send this command to switch on the laser/led.
LaserOff /Led Off *("?LOf")*		Send this command to switch off the laser/led.
PowerOn *("?POn")*		Additional command to power on the main electronics separately if "Auto power-up" is deactivated. This is an advanced controlling sequence and not necessary by default.
PowerOff *("?POf")*		Additional command to power off the main electronics.
SetOperatingMode *("?SOM")*		Send this command to enable different operating modes or use the following wrapper commands:
	SetAutoPowerup *("?SAP")*	Advanced operating mode if disabled. (enabled by default)
	SetAutoStartup *("?SAS")*	Auto-startup behavior. (enabled by default)
	SetImpedanceDigital *("?SID")* *(PhoxX® only)*	Digital input impedance. (necessary only if digital modulation is used)
	SetImpedanceAnalog *("?SIA")* *(not for LuxX®)*	Analog input impedance. (necessary only if analog modulation is used)
	RecallOperatingMode *("?ROM")*	Choose between: standby, cw-only (ACC/APC), digital modulated and/or analog modulated…
ChangeOperatingMode *("?COM")*		Valid for QuixX lasers only: This command complements the SOM command.

SetLevelPower *("?SLP")* *or* **SetPowerAsPercentage** *("?SPP")* *or* **TemporaryPower** *("?TPP")*	Set the laser/led output power.

MeasureDiodePower *("?MDP")*	Additional command to measure the actual emitted light power.

Additional operating mode command:

Auto Reset *("?ARs")*	Needed for special setups only.

Special maintenance command:

CalibrateLaserDiode *("?CLD")*	This command is for maintenance use only.

5.1 Trouble Shooting

If you successfully established a communication with your device you may use the flowcharts in the next two subchapters as a template to program rock-solid control software for Omicron devices.

If the communication is not working please check the following points:

6) Is the laser/led connected to the mains through a power supply that meets the specifications?

7) Did you choose the correct COM-Port?

8) If using USB: Is the device correctly recognized by Windows?
(no yellow question mark in device manager – if so please refer to the "Drivers\Driver Cleanup Utility\Cleanup ReadMe.txt" file on the OCC installation medium)

9) Are your port-settings correct (baud rate, stop bit, parity…)?

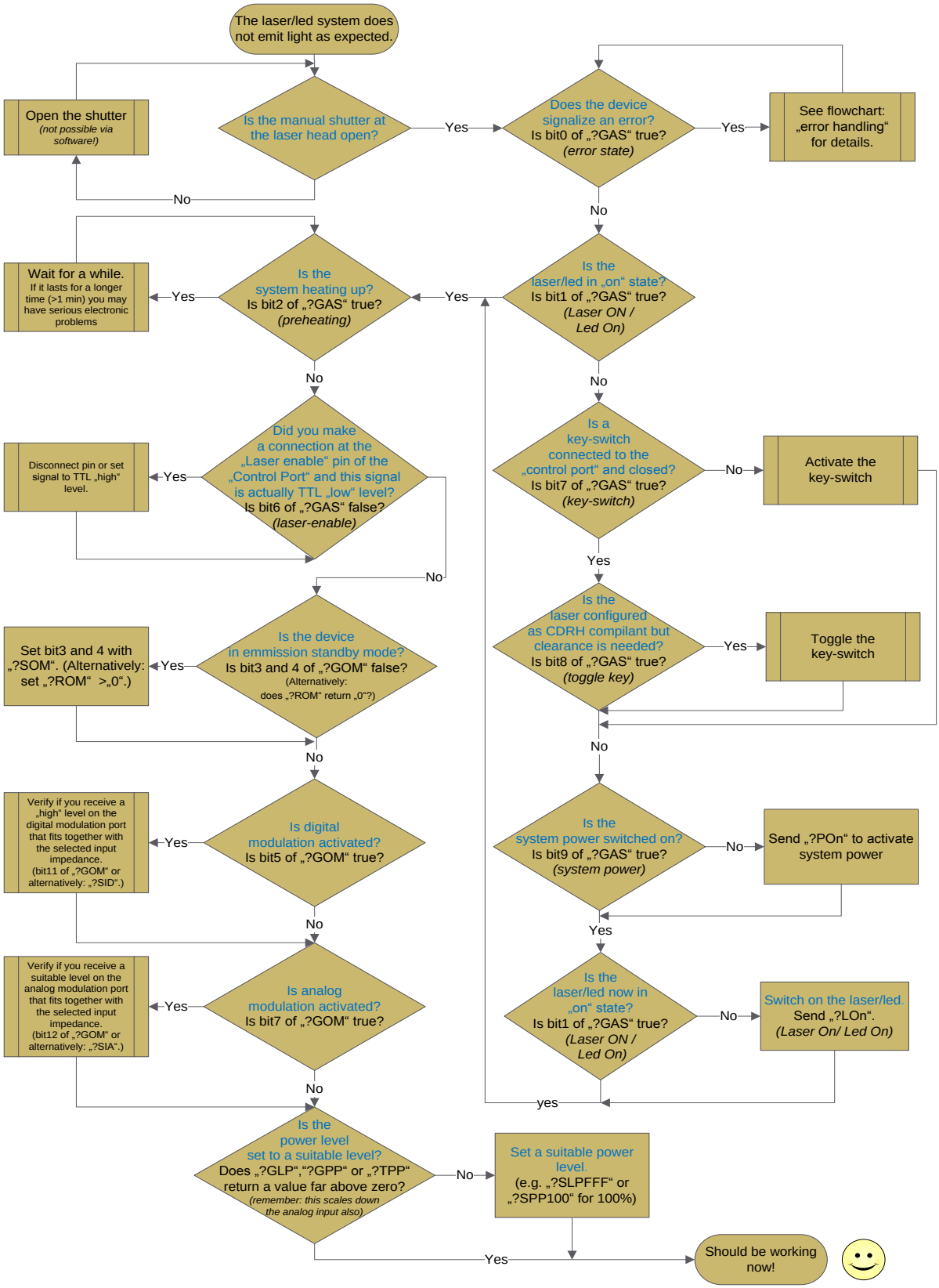
Important:

a. the baud rate for legacy RS232 communication is 57600 baud

b. the baud rate for the virtual COM-port provided by the USB chip is 500000 baud

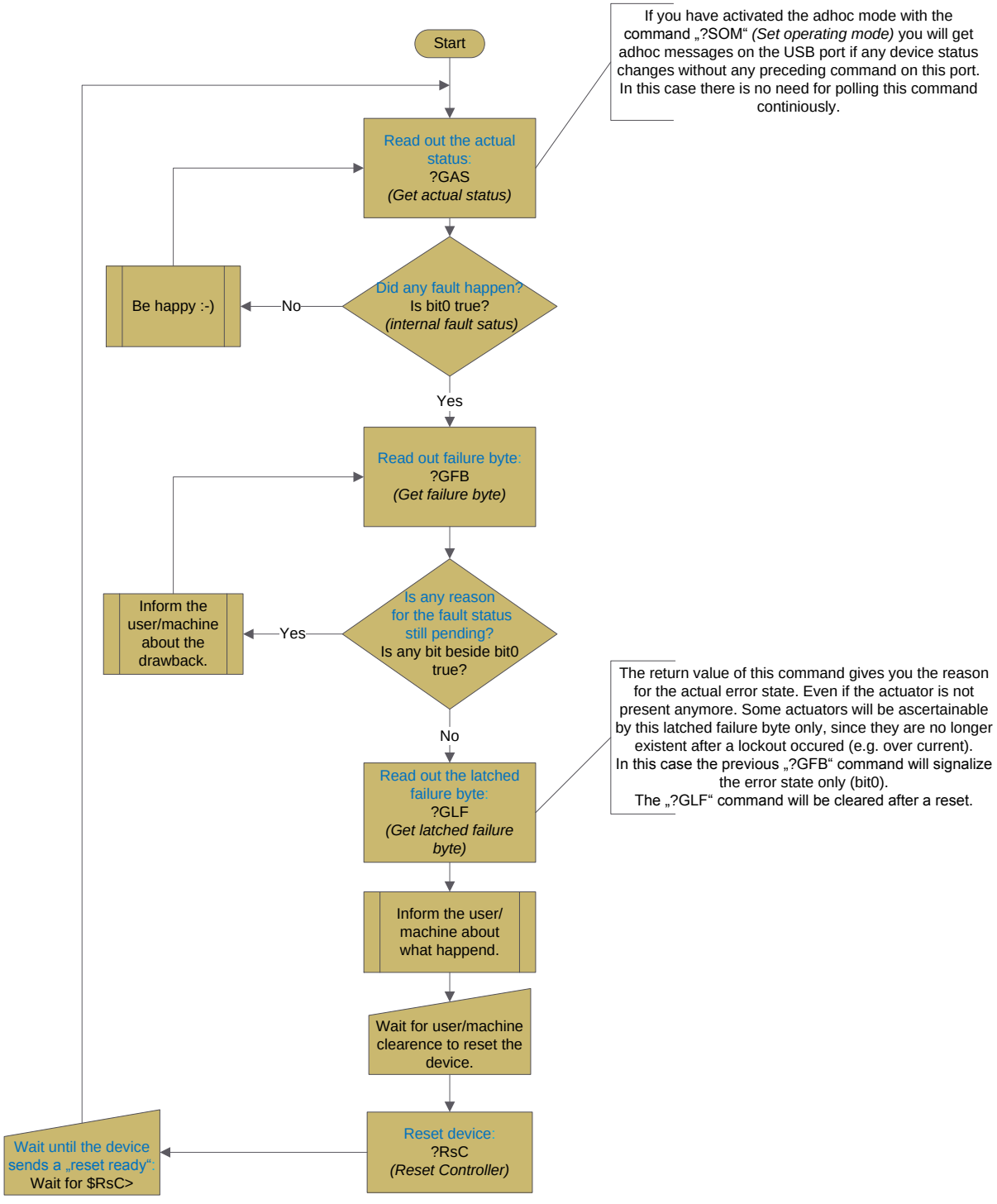
10) Did you correctly implement the protocol definitions?

5.1.1 Precautions to Enable the Laser/Led



Flowchart 1: Omicron device trouble shooting

5.1.2 Error Handling



Flowchart 2: error handling

6. Controlling Light Engines

Light engines need to be differentiated between „multi-channel – multi-port“ and „multi-channel – single-port” devices.

- Omicron laser light engines like the LightHUB® and SOLE® are „multi-channel – multi-port“ devices. This means every single laser in the light engine has its own communication port. Although the SOLE® is equipped with a single USB-connector the lasers are internally linked together by an USB-Hub. Therefore the PC needs to open one communication port for each laser. The advantage is that communication does not differ in communication with single lasers. So with a LightHUB® or SOLE® you can safely ignore this chapter.
- In contrast the Omicron LedHUB® is a „multi-channel – single-port” device. This means there is only one communication port to be opened. The communication with the single channels (LEDMOD.V2 modules as sub-devices) is tunneled through the main controller.

6.1 Controlling „Multi-Channel – Single-Port“ devices

Single port devices are equipped with a main controller acting as a trunk for the whole device. To initiate a communication with the device the main controller should be addressed first.

Valid commands for the LedHUB® main controller are:

12) Get Firmware (?GFw)

13) Get Serial Number (?GSN)

14) Get Spec Info (?GSI) - see text below

15) Get Working Hours (?GWH)

16) Measure Temperature Ambient (?MTA) - Cooling Airflow

17) Get/Set Operating Mode (?GOM/?SOM) - bits 13-15 only

18) Get Actual Status (?GAS)

19) Get Failure Byte, Get Latched Failure (?GFB, ?GLF) - bits 14,11,9,8,5 and 0 only

20) Set Power On / Set Power Off (?POn/?POf)

21) Set LED On / Set LED Off (?LOn/?LOf)

22) Set Auto Power-up, Set Auto Startup (?SAP, ?SAS)

23) Reset Controller (?RsC)

The most important command after "Get Firmware" is "Get Spec Info". The answer on this command may look like:

!GSI[m63]0§5000

The 0 and 5000 are the parameters as described in chapter "command reference". (wavelength is not applicable so it stays 0 and should be ignored. The spec power (here 5000) is the specified power for the complete system)

New is [m63] in square brackets after the command code.
This gives you the number of channels (sub-devices) installed in your light engine.

The preceding "m" indicates that the following integer number needs to be interpreted as a bit mask, showing the channels that are equipped with a sub-device.
63 in binary notation is 111111, indicating 6 active channels. (m59 in contrast would indicate that channel 3 is missing (binary 111011)).

Channel 6 (Lambda 6)	Channel 5 (Lambda 5)	Channel 4 (Lambda 4)	Channel 3 (Lambda 3)	Channel 2 (Lambda 2)	Channel 1 (Lambda 1)
Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

With this information you are able to address each of the sub-devices by adding an index for every command (counting from 1), e.g.:

"?GFw[1]" - Ask the first channel (sub-device) for device ID and Firmware version.

"?SPP[2]50.0" - Set 50 Percent Power Level for channel 2.

etc.

Since the LedHUB® is equipped with LEDMOD.v2 modules for each channel all related commands from chapter "Command Reference" are valid for the indexed commands of the LedHUB® also.