Protocol for communication with KL light sources

KL 2500 LED

Port Settings

The light source is connected via the USB port, but treated as a virtual serial connection. The communication can take place the same way as if the RS232 serial port was there in hardware. The settings for the port are:

- 9600 baud
- 8 data bits
- 1 stop bit
- no parity

Basic Format

This is a description of the communication protocol version 2.0.

Commands (which are sent to the light source) and answers (which are received from the light source) are with a few exceptions 8 byte long ASCII strings with the following structure:

Bytes / Pattern	
A * * * * * * * [0-9A-F]	Address byte to select a channel of the light source(s). Currently only light sources with one channel exist, where this byte is always ASCII ‘0’. Values from ASCII ‘0’ through ASCII ‘F’ are allowed by the protocol (version 1.0).
*MN***** [A-Z] [A-Z]	A two character mnemonic of the command. Only the commands listed below are allowed. All other combinations are reserved for future use.
***HVAL; [0-9A-F] {0,4} ;	If there is no question mark, the command is a “set” command with the value following directly on the mnemonic. The value is a hexadecimal encoded 16 bit number finally followed by the “end-of-command” sign ';'.
***?;... \?;	A question mark '?' introduces a “get”-command. Currently (protocol version 1.0) it is always directly followed by a “end-of-command” sign ';'.
***RETN; [0-9A-F] {4} ;	Each command is acknowledged by an answer string. The address and the command mnemonic are identical to the ones sent with the command string. For “get” commands the fetched value follows directly on the mnemonic and is terminated with the “end-of-command” sign ';'. The answer to “set” commands is the set value as the return value for checking.
***!ERR; ! [0-9A-F] {3} ;	In case the command failed, an exclamation mark '!' is returned after the mnemonic, followed by an hexadecimal error number and the “end-of-command” sign ';'.

'*' denotes any character which is not important for the described command part. The given patterns are Posix ERE's and also match the described part of a command.

SCHOTT
glass made of ideas

Protocol KL 2500 LED

Command Mnemonics

These mnemonics are defined for protocol version 2.0. The characters are all upper case and should be used this way. The reaction to lower case characters is undefined. The light source might or might not react to the command.

Mneumonic	set	get	Description
BR	yes	yes	Set or get the brightness of the light source. The brightness is more or less proportional to the value, but this is not guaranteed. Ambient temperature, aging, binning of the LED's etc. will also have an influence. At th time of writing the allowed range is 0 (off) to 3E8 (1000). '0' will always be off, the maximum may change. To set the maximum brightness use the special value 'FFFF'. Example: 0BR0200; → Sets brightness to 51,2%
ID	no	yes	Identifies the light source. The returned string might be longer than 8 bytes. Allocate a buffer of 256 bytes to make sure that the return string will always fit into the buffer. Example: 0ID?; → KL 2500 LED V2.0
LK	yes	yes	Lock the front panel controls at the light source to allow only remote commands. 0 will unlock, 1 will lock the light source. Example: 0LK0001; → locks the front panel
PR	yes	no	Recall a preset. The value of the “set” command is the index into the five preset banks and may be between 1 and 5. Example: 0PR0001; → load the brightness setting of preset 1
PS	yes	no	Store actual settings as preset in the light source. The given value is the index into the five preset banks and may be between 1 and 5. Example: 0PS0005; → save brightness setting to preset 5 The first preset is also used for the front panel control. If it is pushed the current brightness setting will be saved to preset 1. It can be overwritten through the USB control.
PV	no	yes	Get the protocol version. The first byte represents the version, the second byte the revision. If your program doesn't know the protocol with the identical version number, it shouldn't operate the light source any further and display an information to the user. It should usually be safe to operate a light source with the same version number and a higher revision number than the protocol version used at the development of the program. Currently the protocol version is 2.0 which correspondents to the return value 0200 (hexadecimal) or 512 (decimal).
SF	yes	yes	A footswitch or push button can be connected to the KL 2500. If a switch is used, a ‘1’ has to be sent; if a push button is used, a ‘0’ has to be sent. This setting is saved in the flash of the light source. Example: 0SF0001; → Switch Example: 0SF0000; → Push Button
SH	yes	yes	The shutter command emulates shutters as they are known from halogen light sources. ‘1’ has the meaning of “close shutter” (light is off); ‘0’ has the meaning of “open shutter” (light is on). Example: 0SH0001; → Shutter is activated
TX	no	yes	Returns the temperature of the LED PCB as an integer in steps of 0.0625 Kelvin.

SCHOTT
glass made of ideas

Protocol KL 2500 LED

Error Codes

These are the error codes as they are defined for protocol version 1.0 (initial version).

Error Code	Description
0	OK. No error.
1	Unspecified error.
2	Syntax error.
3	Unknown command.
4	Non set-able command. (missing '?')
5	Non get-able command ('?' not allowed)
6	Value out of range.
7	Value too low.
8	Value too high.
9	Value not a number.
A	Unfinished previous command.
B	Command not supported.
F	Illegal preset index number.
10	
11	
12	
13	
14	

KL Control Tool

The KL Control tool is a little Java program, which can be used to get acquainted with a KL light source under computer control. The KL Control tool should work under Windows, Mac Os, Linux and other operating systems. However no guarantees of any kind are given for proper function. The tool was only tested with Windows as operating system and with Sun's JRE 1.5.

Requirements

Except for Java itself all required software is included in the installation package. If you encounter problems when you try to use the KL Control tool, try to update to the latest versions of the following software.

- FTDI virtual COM port driver
  The light source is connected to the computer via the USB port. The KL Control tool accesses the light source by a virtual serial port. The latest drivers are available at http://www.ftdichip.com/FTDrivers.htm.

- JRE (Java Runtime Environment) 1.5 or higher
  A Java program requires a runtime engine to run on. The JRE is already installed on most systems. If it is not installed or the version is too old, the latest version is available at http://java.sun.com/javase/downloads/index.jsp

- RXTX native communication library for Java
  Used to communicate with the virtual COM port within the Java program.
  The latest version is available at http://www.rxtx.org

- SWT (Standard Widget Toolkit)
  Used to display the windows and control elements of the KL Control tool.
  The latest version is available at http://www.eclipse.org/swt/

Installation

For the installation of the required software please refer to the instructions of the software providers. The KL Control tool itself requires no special installation. Just make sure that the required software can be found in the search path and start the “KLControl.jar” file with the javaw command. The way to start a Java program depends on the underlying operating system. On standard Windows installations the “.jar” extension is registered with the JRE and can simply be started with a double click on the Jar-file.

Usage

The KL Control tool is very simple to use and pretty much self explanatory. Every control has a tool tip which should become visible when you hover with your mouse over the control. Experts / developer can start the tool with additional command line options, which enable some extra features (precede the option with a dash '-' or a slash '/'):

- develop
  Produces a slightly different configuration of the program window, which is better suited for development. It allows to issue single commands whereas the standard configuration combines commands for a better user experience.

- con
  Enables a menu item to open a console window. With the console window you can monitor the communication of the program with the light source on the ASCII protocol level.

- simulate
  Adds a light source simulation to the connection port menu which an be used without an actual hardware light source. The simulation can be directly invoked by the “port” option with the identifier “simulation”. Although similar, the simulated light source might behave differently than a hardware light source. It is meant to simulate just the protocol part of a real light source.

- port <*port identifier*>
  Select a communication port to be connected with from the start. The port identifier is the name you also see in the communication port menu. (Or “simulation” for the simulated light source.)

- hidePorts
  This option hides the menu to select the communication port. This option is to be used together with the “port” option. It is meant as an option for show cases, where the user should be prevented from “messing up” things.

- lang <*language specification*>
  Select an other language than the default to be used in the user interface. The language specification is the standard two letter ISO code (e. g. “en”, “en_US” etc.). The fall back language is English if the selected language doesn't exist.

All specifications are subject to change without prior notice. This datasheet or any extracts thereof may only be used in other publications with express permission of SCHOTT. © SCHOTT AG

Lighting and Imaging
SCHOTT AG
Hattenbergstrasse 10
55122 Mainz
Germany
Phone: +49 (0) 6131/66-7796
Fax: +49 (0) 6131/66-7850
lightingimaging@schott.com
www.schott.com/lightingimaging

SCHOTT
glass made of ideas