Document Title:

UG Luxbeam rapid system – LRS WQ

Rev:	Description:	Written: Date:	Written: Sign:	Checked: Date:	Checked: Sign:	Approved: Date:	Approved: Sign:
PA1	Initial Document	06/06/15	BV				
A.1	Added Minimum exposure time						
A.2	Added initled command to i2c_cmd						
A.3	Added minimum exposure / bit-depth. Added led-power and sequence i2c packet	15/09/15	BV				
A.4	Added ledtemplimit and boardtemplimit in SW. Also added some led-driver init function to tweak overshoot/undershoot, OCP and OPP parametres. Added display port support in SW						

Table of Contents

1 Scope

This document defines how to use the LRS WQ interface

1.1 Abbreviations, acronyms and definitions

DLP Registered trademark of Texas Instruments Inc, acronym for Digital Light Processing
DMD Digital Micromirror Device
FW Firmware
HTTP HyperText Transfer Protocol
HW Hardware
I2C and I^2C Inter–Integrated Circuit
I/O Input/Output
IS Interface Specification
LED Light Emitting Diode
Light Engine A self-contained DLP sub-module, comprising DLP electronics, optics, light source, power supply and mechanical enclosure
LRS LuxBeam Rapid System
LUXBEAM Registered trademark of VISITECH AS, used for VISITECH designed DLP products
N/A Not Applicable
PSU Power Supply Unit
RLE Run-Length Encoding
SW Software
TBD To Be Determined
TI Texas Instruments Inc.
USB Universal Serial Bus
UV Ultra Violet

2 Reference documents

No.	Description
1	
2	
3	
4	

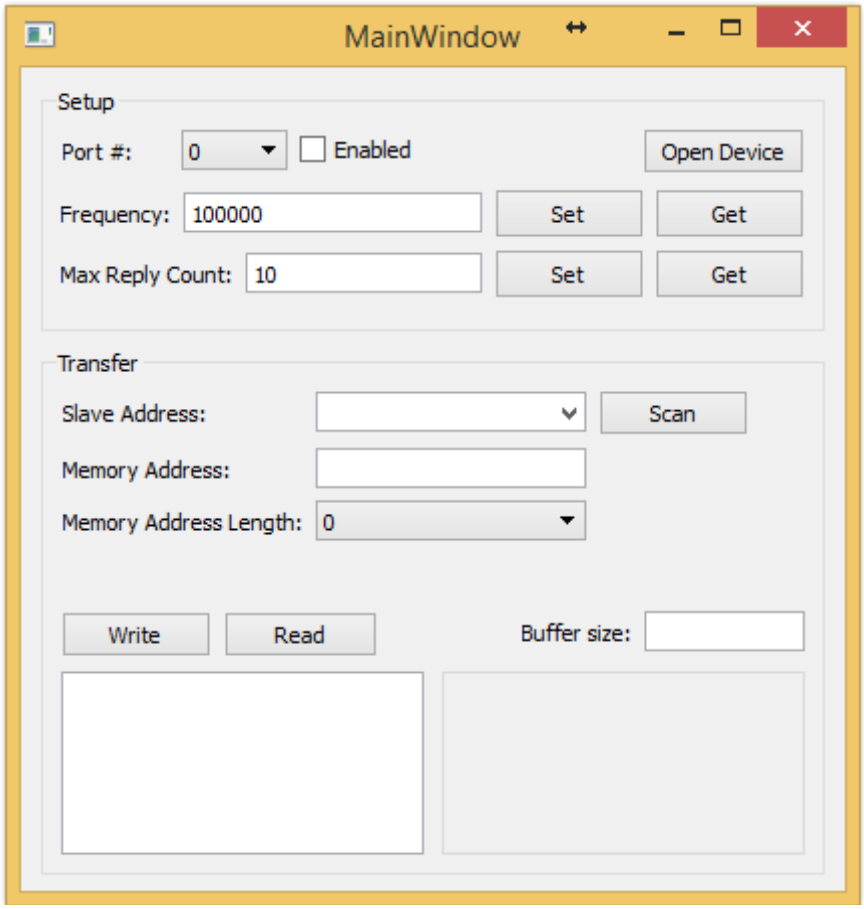
3 Software set up

Recommended set up: Host computer running windows 7/8/8.1
LRS WQ uses i2c for communication. To make this more user-friendly a usb-i2c adpter by Diolan is included inside the box.

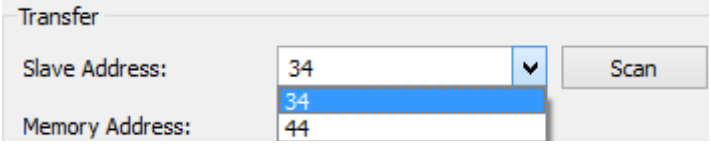
3.1 Diolan sw driver set up and test

Run the included dln.3.0.2.exe and install with default settings.

Power up LRS WQ and connect your computer to the usb-interface of LRS WQ. Wait for the drivers to auto-install. When the drivers are successfully installed, test that you have connection with the diolan usb-server by running i2c_master_gui.exe. A window like this should appear:



Press enabled and scan. In the box to the right slave address 34 and 44 should appear like this:

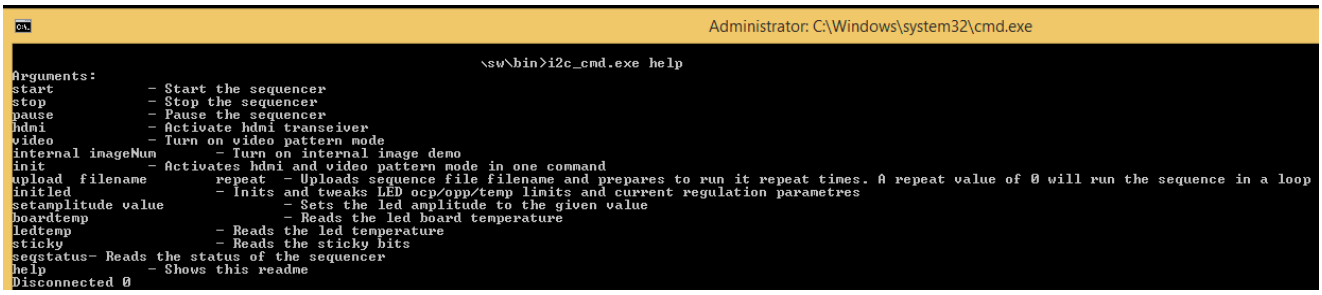


This means you have connection to all LRS devices. You may now close i2c_master_gui.exe

4 i2c_cmd

For simple communication with the LRS a small program called i2c_cmd.exe is provided. The sources are also available. i2c_cmd.exe is run with different arguments to send instructions to the LRS.

Try to run 'i2c_cmd.exe help' to see all available arguments.



4.1 Return values

The i2c_cmd application returns 0 if success and another value for error as described:

```
#define ERROR_OK 0
#define ERROR_DLN_ADAPTER_OPEN 1
#define ERROR_GET_I2C_PORT_COUNT 2
#define ERROR_NO_I2C_PORTS 3
#define ERROR_DLN_SERVER_CONNECT_FAILED 4
#define ERROR_MASTER_SET_FREQUENCY_FAILED 5
#define WARNING_FREQUENCY_ROUNDED 6
#define ERROR_GET_FREQUENCY_FAILED 7
#define ERROR_MASTER_ENABLE_FAILED 8
#define ERROR_MASTER_DISABLE_FAILED 9
#define ERROR_MASTER_SCAN_FAILED 10
#define ERROR_MASTER_IS_ENABLED_FAILED 11
#define ERROR_SET_REPLY_COUNT_FAILED 12
#define ERROR_GET_REPLY_COUNT 13
#define ERROR_UNVALID_INPUT 14
#define ERROR_READ_FAILED 15
#define ERROR_WRITE_FAILED 16
#define ERROR_INVALID_WRITE_DATA 17
#define ERROR_SEQUENCE_TO_MANY_PATTERNS 18
#define ERROR_COULD_NOT_FIND_SEQUENCE_FILE 19
#define ERROR_COULD_NOT_FIND_ANY_LINES_IN_SEQUENCE_FILE 20
#define ERROR_MISSING_ARGUMENTS 21
#define ERROR_COMMAND_ARGUMENT_MISMATCH 22
#define ERROR_TO_HIGH_IMAGE_NUM 23
#define ERROR_TO_HIGH_LED_AMPLITUDE 24

#define ERROR_SEQUENCE_NUM_ARGS 1000
#define ERROR_SEQUENCE_VALUES 0x10000
//LAST HEX DIGIT IS ARGUMENT NUM, hex digit 3,2,1 is line num

#define LED_TEMP_RETURN_BASE 2000
#define LED_BOARD_TEMP_RETURN_BASE 3000
#define LED_STICKY_RETURN_BASE 4000
#define TI_SEQUENCE_RETURN_BASE 5000
//Sticky bits: (4):OCP (3): Door_open (2): Fan stopped (1): Board_overtemp, (0): LED_overtemp
```

5 First sequence test

Now we want to test an HDMI streamed sequence.

After boot, the LRS WQ will display a demo sequence. Run 'i2c_cmd.exe stop' to stop it.
Put your hdmi or display port cable into the host computer and into the LRS WQ port (depending on if you have an HDMI or display port activated light engine) . The markings of the Displayport and HDMI is swiched on the light engines on first hw revision).

We now want to open a connection over the HDMI or DP interface. Verify on the host computers screen resolution that you are running the LRS WQ HDMI port with a WQXGA resolution (2560x1600).

Now do 'i2c_cmd.exe init hdmi' if you use HDMI or 'i2c_cmd.exe init dp' if you use displayport. This will open the HDMI port and activate the streaming hdmi pattern interface.

Try to upload the demo sequence called demo.txt by running 'i2c_cmd.exe upload demo.txt 0' the 0 is indicating that you want the sequence to run indefinitely.

Run 'i2c_cmd.exe start' to start the sequencer. You should see some flickering light showing the HDMI display.

6 Sequence language

This chapter will walk through the different sequence parametres

6.1 i2c_cmd sequence file parametres

Bit depth - 0 = 1 bit, 1 = 2 bit.... 7 = 8 bit

Color – Does not matter for this application – set to 0.

Wait for trigger – 1 = Wait for VSYNC before displaying the pattern, 0 = Continue running after previous pattern

Dark display – Valid range 2^24. Dark display time following exposure
Bit position in the image pattern – See explanation under the image below.

Bit position in the image pattern – Valid range 0-23

Clear the pattern after exposure – Usually automaticly handled unless running a 1 bit pattern.

Each line in the sequence file is one pattern, each argument is seperated by ,

6.2 i2c upload sequence file packet (for making your own software)

BYTE	BITS	DESCRIPTION	RESET	TYPE
1:0	15:0	Pattern Index (range 0 through 511)	d0	w
4:2	23:0	Pattern exposure in micro seconds		
5	0	Clear the pattern after exposure. This is only applicable for 1 bit pattern with external trigger. For other patterns, the clear is automatically handled.		w
	3:1	Select desired bit-depth b000 = 1 bit b001 = 2 bit b010 = 3 bit ... b111 = 8 bit		w
	6:4	b000 = All LEDs disabled b001 = Red b010 = Green b011 = Yellow (Green + Red) b100 = Blue b101 = Magenta (Blue + Red) b110 = Cyan (Blue + Green) b111 = White (Blue + Green + Red)		w
	7	1 = Wait for trigger before displaying the pattern 0 = Continue running after previous pattern		
8:6	23:0	Dark display time following the exposure (in micro seconds)		w
9	0	1 = Disable trigger 2 output for this pattern 0 = Enable trigger 2 output for this pattern		w
	7:1	Reserved		w
11:10	10:0	Image pattern index (Not applicable in video pattern mode) Valid Range 0-255		w
	15:11	Bit position in the image pattern (Frame in video pattern mode) Valid range 0-23		w

6.3 Exposure time per pattern

Exposure time is given in micro seconds. Valid range: 2^24. Minimum exposure per pattern is related to the bith depth of the pattern:

Bit depth	Minimum exposure in us
8	>4046
7	>1998
6	>1487
5	>1215
4	>733
3	>380
2	>304
1	>105

6.4 Bit patterns and bit-weight

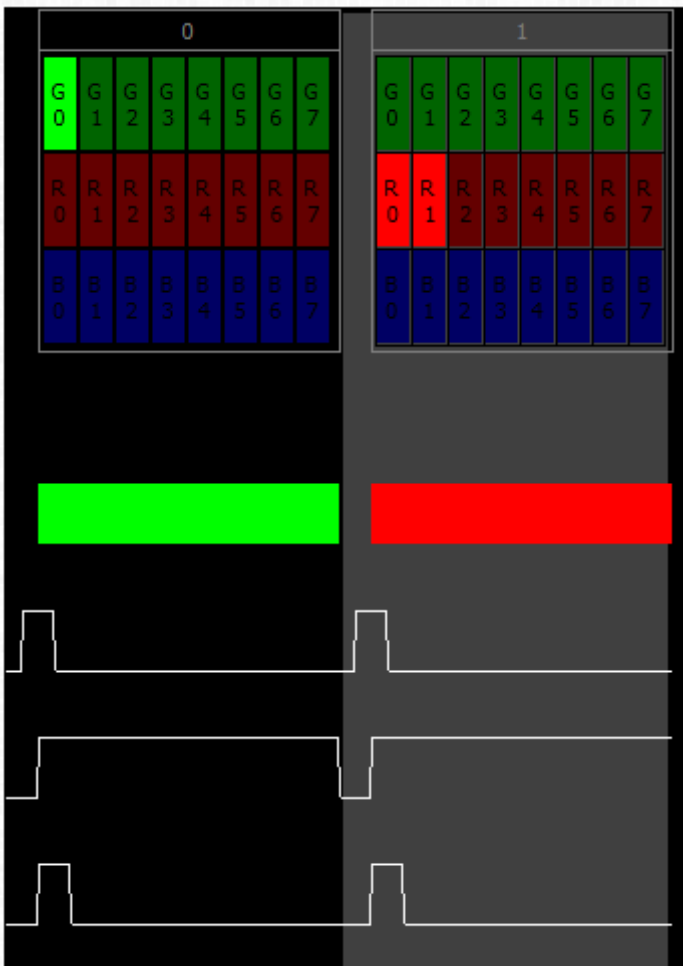
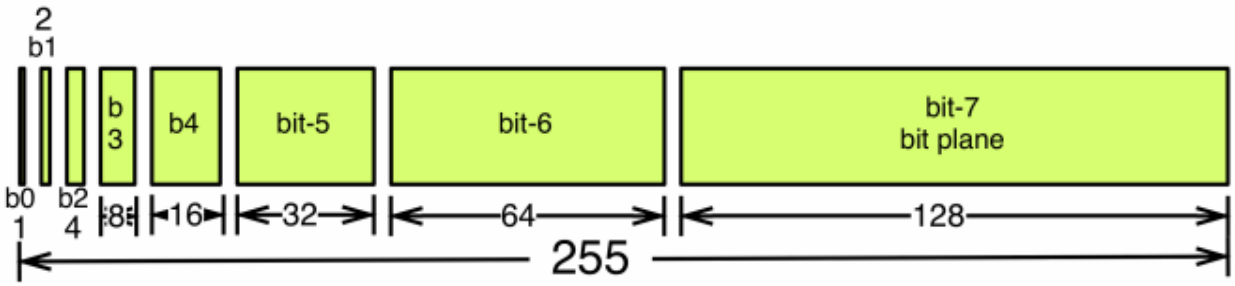


Image showing two patterns: First(green) with 1 bit depth and bit position 0, second (red) with 2 bit depth and bit position 8.



Bit partition in frame for an 8-bit monochrome image pattern.

7 Sources

Sources are located in \c_cpp\examples\qt

8 Led driver amplitude and power