

Application Note

OSRAM UART Protocol

Table of Contents

1.	Introduction	2
2.	UART Software Protocol.....	2
2.1.	Communication Protocol.....	2
2.2.	OSRAM Instruction Set.....	3
2.2.1.	Command Structure	3
2.2.1.1.	Lamp On (25h).....	3
2.2.1.2.	Lamp Off (26h).....	4
2.2.1.3.	Reset lamp driver (3Ah).....	4
2.2.1.4.	Enable Communication (3Eh).....	4
2.2.1.5.	Set Power (82h yyh xxh)	4
2.2.1.6.	Sync External On (default)	5
2.2.1.7.	Sync External Off.....	5
2.2.2.	Query Structure	5
2.2.2.1.	Read Temperature (20h)	6
2.2.2.2.	Read Status Byte (22h)	7
2.2.2.3.	Read Power (27h).....	7
2.2.2.4.	Read SW Version (28h).....	8
2.2.2.5.	Read Lamp Voltage (21h)	8
2.2.2.6.	RAM Read (60h).....	8
2.2.2.7.	EEPROM Read (84h)	8
2.2.2.8.	Get Peak + Plateau Value (2Fh).....	8
2.2.2.9.	Get Error Status (3Dh).....	9
2.2.2.10.	Read HW Version (29h)	9
2.2.2.11.	Read Kernel Version (3Bh).....	9
2.2.3.	Error Response Structure.....	9
2.2.4.	Caption	10
3.	Hardware Protocol	10
3.1.	Communication Interface	10
3.2.	UART Parameters and Timing.....	10
3.3.	Example of a single Byte command.....	12
4.	Examples	13
4.1.	Example 1: Reading out current pulse over plateau ratio.....	13
5.	Application Note Change Log.....	13

1. Introduction

The UART interface, whose hardware details are described in the application note „5-Pin Control Interface”, offers the possibility to establish a communication via two wires between projector and OSRAM lamp driver. Every meaningful communication needs a protocol to be clarified: the interface software requirements for the user are specified in this application note. The UART communication offers a lot of beneficial functions and enables a smart control of the OSRAM lamp system.

In particular, the major section of this document defines the software communication protocol, which is based on the instruction set (containing commands and queries); the communication timing is specified too, and a detailed description of the UART commands is given. The last paragraph, presenting an overview about the hardware protocol (with its communication interface, parameters and timing), acts like a bridge to the above mentioned application note „5-Pin Control Interface”.

2. UART Software Protocol

2.1. Communication Protocol

The communication takes place as the user first sends an **instruction** via the receive data signal (RxD), according to what is described in the paragraphs 2.2.1 and 2.2.2 below. The OSRAM lampdriver determines if the instruction is a **command or query**. For a command, the OSRAM lampdriver replies back to the user with an echo via the transmit data signal (TxD). For a query, the OSRAM lampdriver replies back to the user with echo and response via the transmit data signal (TxD). Figure 1 shows a flow diagram of the communication protocol.

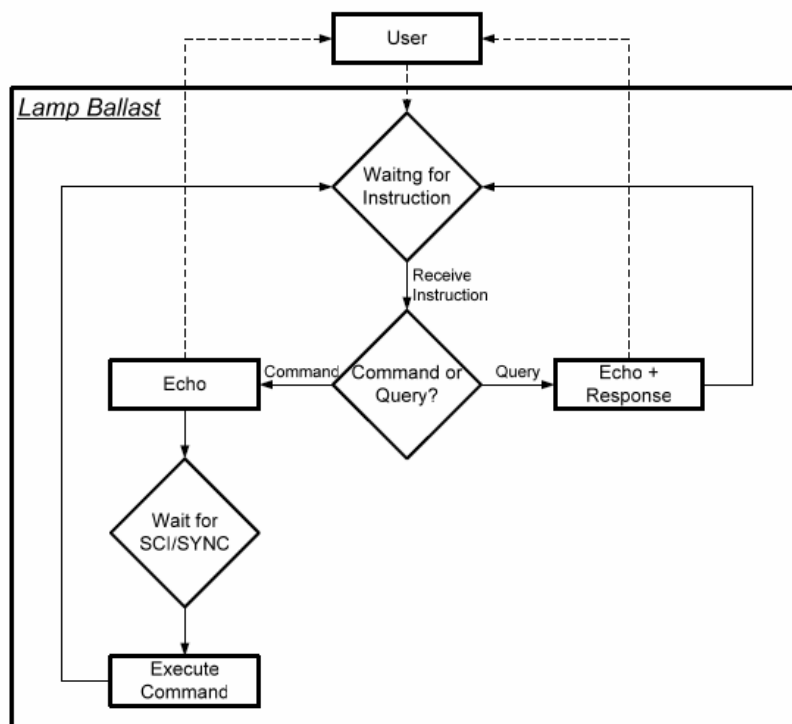


Figure 1: Communication Protocol Flow

2.2. OSRAM Instruction Set

This instruction set is valid from Control Board Gen3V0 @ kernel version DA03.

As above mentioned, there are two types of instructions: commands and queries. For a command, the OSRAM lampdriver replies with an echo. For a query, the OSRAM lampdriver replies with echo and response. Table 1 shows the instruction response structure.

Instruction	Response Structure
Command	Echo
Query	Echo + Response

Table 1: Instruction Response Structure

2.2.1. Command Structure

Each command consists of a command key and, if necessary, of additional 1 to 3 argument bytes. The lamp driver always answers with a complete echo. The echo consists at least of the command key and, depending on the command, of the additional argument bytes, too.

Command Name	Key	Argument	Execution
Lamp On	25h	none	immediately (from standby)
Lamp Off	26h	none	immediately
Reset Lamp Driver	3Ah	none	immediately
Enable Communication	3Eh	none	Immediately
Set Power	82h	Desired Wattage (2 Byte)	Immediately
Sync External On (default)	2Bh	none	Immediately
Sync External Off	2Ch	None	immediately

Table 2: Command Set. For some commands, one or more additional parameter bytes have to be added to the command byte in order to specify values, data, or addresses (see third column). More details about the exact meaning of the arguments are explained in the next paragraphs.

2.2.1.1. Lamp On (25h)

Command Byte: 25H

Lampdriver Response: 25H

As the SCI signal has a “higher turning on priority” than the UART software command, the command “Lamp On” will take effect only if the SCI signal is valid at the same time.

Furthermore, after having turned off the lamp it will be possible to turn it on again earliest after a waiting time. This minimum OFF time is always performed and corresponds to the lamp cooling down period. Hereafter, if the SCI signal is valid and no operation error occurred, the lamp driver will try to ignite the lamp again.

If the lamp has already been turned on via SCI signal, it is not necessary to send an additional "Lamp On" command.

2.2.1.2. Lamp Off (26h)

Command Byte: 26H

Lampdriver Response: 26H

With this command the lamp is switched off and the "Error Status" byte is set to zero. If an operation failure has occurred, it is necessary to send the "Lamp Off" command in order to reset the "Error Status" byte and then to be able to start the lamp again.

2.2.1.3. Reset lamp driver (3Ah)

Command Byte: 3AH

Lampdriver Response: none

The "Reset lamp driver" command has the same effect like an OFF-ON input power cycle. All parameters will be set to power up conditions.

WARNING: The use of the reset command clears the delay time before a new ignition, and the execution of successive ignition cycles without delay time can cause permanent damage to the lamp driver (Danger of overheating the resonance choke and full bridge transistors!).

2.2.1.4. Enable Communication (3Eh)

Command Byte: 3EH

Lampdriver Response: 3EH

This command **opens** up the communication between the user interface and the lampdriver. Afterwards, once the access is enabled, it remains enabled until power is removed from the ballast.

2.2.1.5. Set Power (82h yyh xxh)

The output wattage of the lampdriver can be adjusted in steps of 1 Watt. The adjustable range and the number of steps are depending on the lamp specification (e.g. lamp dimmable from 200 W to 250 W means 50 successive dim steps).

The wattage is set by the command byte, followed by two bytes describing the output value in Watt with the most significant byte first.

Command Byte: 82H yyH xxH

Lampdriver Response: 82H yyH xxH

- For the answer of the Set Power Command the value of the appropriate register is sent (depending of the sent value limited to the maximum or minimum wattage). The answer does not content a measured actual power, therefore it cannot be used for determining the status of the lamp.

- Any previous power setting will be reset if the lamp driver is completely switched off or if the reset command (->[5.1.16](#)) will be executed.

- A previous power setting will remain unchanged if the lamp driver is been switched OFF and ON by a UART command or via a SCI sequence.

2.2.1.6. Sync External On (default)

Switches the external synchronization on. This is also the default value after restarting the lampdriver.

Command Byte: 2BH

Lampdriver Response: 2BH

2.2.1.7. Sync External Off

Forces the lampdriver to ignore external synchronization. Intended to use for debugging external synchronization to avoid DC content in the lamp current.

Command Byte: 2CH

Lampdriver Response: 2CH

2.2.2. Query Structure

Table 3 lists the query set. The third column (“Response”) indicates the structure of the response given from the lamp driver after having received the corresponding query, which is indicated in the column “Key”. More detailed information about every single query are given later on in this chapter.

Query Name	Key	Argument	Response	No. of Response Bytes
Read Temperature	20h	none	20h + Temperature Byte	2
Read Status Byte	22h	none	22h + Status Byte	2
Read Power	27h	none	27h + Actual Power	3
Read SW Version	28h	none	28h + SW Version (3 Byte)	4
Read Lamp Voltage	21h	none	21h + Lamp Voltage	2
RAM Read	60h	aah	60h + aah + xxh	3
EEPROM Read	84h	bbh aah	84h + bbh + aah + xxh	4
Get Peak + Plateau Value	2Fh	none	2Fh + Peak Value + Plateau Value	3
Get Error Status	3Dh	none	3Dh + Error Number	2
Read HW Version	29h	none	29h + HW Version (3 Byte)	4
Read Kernel Version	3Bh	none	Kernel Version (4 Byte)	4

Table 3: Query Set

2.2.2.1. Read Temperature (20h)

Command Byte: 20H

Lampdriver Response: 20H xxH

The value of the parameter byte represents the temperature at the NTC sensor on the main board PCB. The real temperature in °C can be calculated from the response byte (xxH) by using the proper formula in the diagram below. The curve represents the actual sensor temperature within an error of ± 1.5 °C. Because the NTC resistor has an additional tolerance of 5%, the estimated readout error is ± 3 °C.

There are two temperature sensor versions in use :

1. All In One Ballasts (which have two NTC sensors implemented) and
2. Lamp drivers (which have only one NTC sensor)

Please refer to the related lamp driver specification for the appropriate type of driver to choose the proper curve and formula in the diagram below.

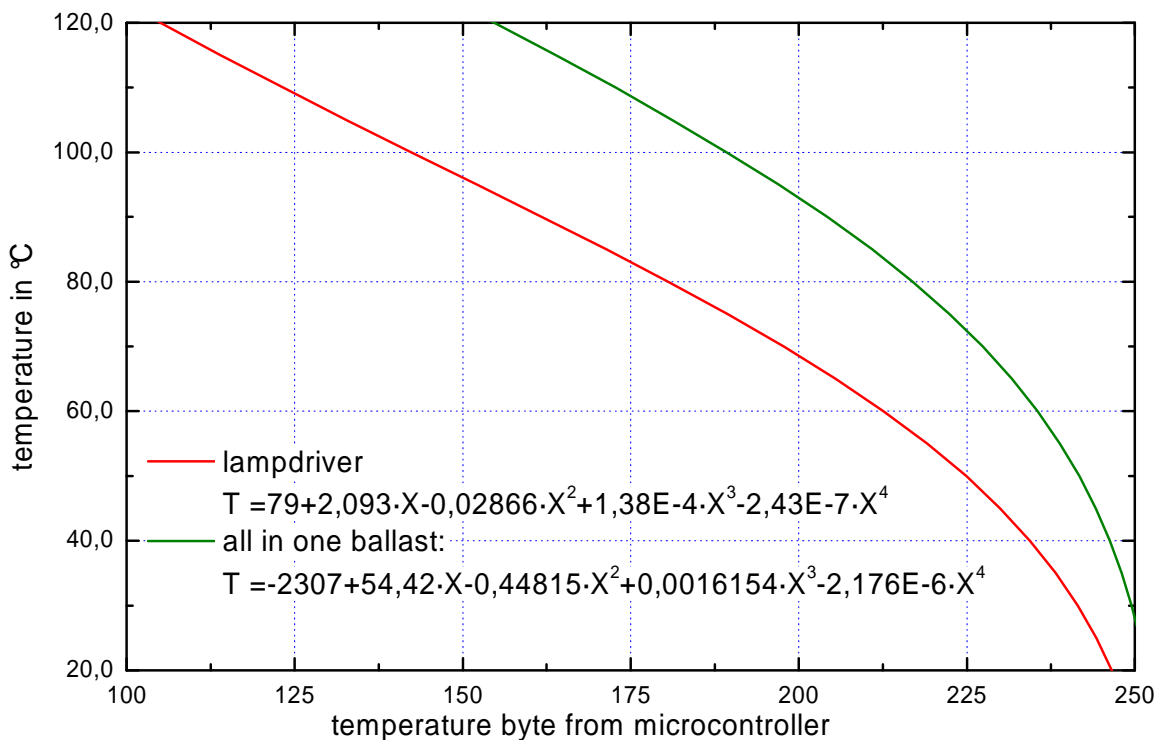


Figure 2: Temperature calculation

To calculate the actual temperature in °C you have to insert the response byte (xxH) in the wild card character X.

2.2.2.2. Read Status Byte (22h)

Command Byte: 22H

Lampdriver Response: 22H xxH

The value of the Parameter byte describes the actual status of the lampdriver. The meaning of the bits is described in the table:

Bit	Explanation
0	Step down converter has been stopped
1	Operation error has occurred (e.g. too high temperature)
2	Vcc (control board supply voltage) too low
3	Shutdown temperature reached
4	External synchronization detected
5	Run up cycle of lamp (during flag is set, internal frequency is used)
6	Maximum time for ignition (4 s) has exceeded
7	Lamp burning detected (lamp lit)

Table 4: Status Byte

Examples for lampdriver status bytes :

Lampdriver condition	Binary value	Note
Lamp is burning	1100 0000	without ext. sync
Lamp is switched off by command	0000 0001	without ext. sync
Lamp has ignited (max. ignition time has not exceeded)	1010 0000	without ext. sync
Lamp has ignited (max. ignition time has exceeded)	1111 0000	with ext. sync
Lamp has not ignited (max. ignition time has not exceeded)	0010 0000	without ext. sync
Lamp has not ignited (max. ignition time has exceeded)	0111 0011	with ext. sync
Lamp voltage too high	0101 0011	with ext. sync
External synchronization detected, lamp lit	1101 0000	with ext. sync
Shutdown temperature exceeded	0101 1011	with ext. sync

The 5th bit is set for 10 sec after switching on the lampdriver (by UART or with SCI), afterwards it is set to 0. The above mentioned examples with bit 5 is set to 1 are status byte queries within the first 10 sec after lamp start.

Table 5: Status Byte Examples

2.2.2.3. Read Power (27h)

The actual output wattage (in Watt), which has been set with a foregoing Set Power Command or the default start value by powering up the lampdriver.

Command Byte: 27H

Lampdriver Response: 27H yyH xxH

- The Read Power does not measure the actual power, it reflects the content of the desired wattage register in the controller. When the lamp is switched off, the answer for the command will be still the value which has been set. Therefore this command cannot be used to determine the actual status of the lamp.

2.2.2.4. Read SW Version (28h)

The version of the actual software release is responded with three bytes (MSB first).

Command Byte: 28H
Lampdriver Response: 28H zzH yyH xxH

2.2.2.5. Read Lamp Voltage (21h)

The value of the Parameter byte describes the actual lamp voltage in a hexadecimal byte.

Command Byte: 21H
Lampdriver Response: 21H xxH

According to the byte value you can calculate the voltage in Volts with the following formula:

$$U_{\text{lamp}}(\text{V}) = 0.703 * U_{\text{lamp}}(\text{Byte value})$$

- *The Read Lamp Command should be only used in conjunction with the Status Byte or Error Status Command to determine the status of the lamp. The open circuit voltage measured by the micro controller with an extinguished lamp might be in the range of a lit lamp.*

2.2.2.6. RAM Read (60h)

Returns the value of the specified memory byte.

Command Byte: 60H aaH
Lampdriver Response: 60H aaH xxH

The value of xxH is the content of the addressed (aaH) RAM Byte. The address range has to be between 80H and FFH.

2.2.2.7. EEPROM Read (84h)

Returns the value of the addressed EEPROM cell.

Command Byte: 84H bbH aaH
Lampdriver Response: 84H bbH aaH xxH

The value of xxH is the content of the addressed (bbH aaH) EEPROM Byte. The address range has to be between 0000H and 0200H (Parameter EEPROM Area).

2.2.2.8. Get Peak + Plateau Value (2Fh)

Returns the command echo and two 8-bit values representing the pulse current (first byte) and the plateau current (second byte). The ballast's answer on the pulse/plateau-command is good for calculating the quotient, wherewith you can calculate the pulse height relative to the plateau current. Due to the fact that the 8-bit values represent currents they are linear to the current in Ampere, but the factor is a variable parameter of the lamp driver.

Command Byte: 2FH
Lampdriver Response: 2FH xxH xxH

2.2.2.9. Get Error Status (3Dh)

Returns the command echo and one byte (error number) representing the status of the lampdriver.

Command Byte: 3DH

Lampdriver Response: 3DH xxH

Error Number xxH	Status of Lampdriver
00H	No error
01H	Temperature too high
02H	Lamp voltage too low
03H	Lamp voltage too high (end of life)
04H	Lamp does not ignite
05H	Lamp extinguished
06H	Lamp extinguished after ignition
07H	EEPROM write error
08H	EEPROM buffer overflow
09H	UART buffer overflow
0AH	Current value zero
0BH	Configuration error
0CH	Lamp voltage too low
0DH	Wrong kernel version

Table 6: Error numbers

2.2.2.10. Read HW Version (29h)

Returns the command echo and three 8 bit values representing the hardware version of the control board (e.g. 33H 56H 30H for actual control board).

Command Byte: 29H

Lampdriver Response: 29H zzH yyH xxH

2.2.2.11. Read Kernel Version (3Bh)

This query returns the maximum allowed gain value for the lamp. (gain calculation formula see Returns four 8 bit values to identify the kernel version of the software (e.g. 44H 41H 30H 33H).

Command Byte: 3BH

Lampdriver Response: zzH yyH xxH wwH

2.2.3. Error Response Structure

In case of a corrupted command transmission or unknown (undefined) commands the lampdriver responds the error byte AAH. If the complete command has not been received in less than 50 ms, the complete receive queue is cleared and the byte ABH (timeout) is sent. Table 7 shows the error response structure.

Parameter	Response Structure
Unknown command	AAh
Timeout during transmission	ABh

Table 7: Error response structure

2.2.4. Caption

- xx ... Low Byte of Parameter Byte
- yy ... High Byte of Parameter Byte
- zz yy xx ... Three Byte Response with Most Significant Byte First
- zz yy xx ww ... Four Byte Response with Most Significant Byte First
- aa ... Address Low
- bb ... Address High

3. Hardware Protocol

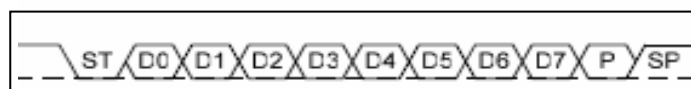
3.1. Communication Interface

The implemented UART interface is a serial two wire opto isolated interface. It consists of a transmit data line (TxD) and a receive data line (RxD). RxD and TxD lines are used for two-way (full-duplex) communications. No special software protocol nor any hardware handshake is used.

For the logic level definition of the transmit and receive line as well as for the electrical interface specification refer to the application note "5-Pin Interface".

3.2. UART Parameters and Timing

The UART serial communication parameters comply with EIA-RS-232-C Interface Standard. The frame properties, are specified in table 9. Every frame starts with the **start bit** (which is always **0**), followed by the least significant data bit (indicated with D0). Then the next data bits are succeeding, ending with the most significant bit (indicated with D7).



Frame Structure (ST=Start Bit, Dn=Data Bits, P=Parity Bit, SP=Stop Bit)

The parity bit is inserted after the data bits, before the **stop bit** (which is always **1**). When a complete frame is transmitted, it can be directly followed by a new frame, or the communication line can be set to an idle (high) state. Each **frame** corresponds to a **single byte information**.

Each command or query can consist of one or several successive **RxD frames**. The lamp driver will answer accordingly with one or several successive **TxD frames**. Table 15 show the communication flow and define the timing limit for receiving and transmitting $T_{MAX\ Command}$ and $T_{MAX\ Response}$.

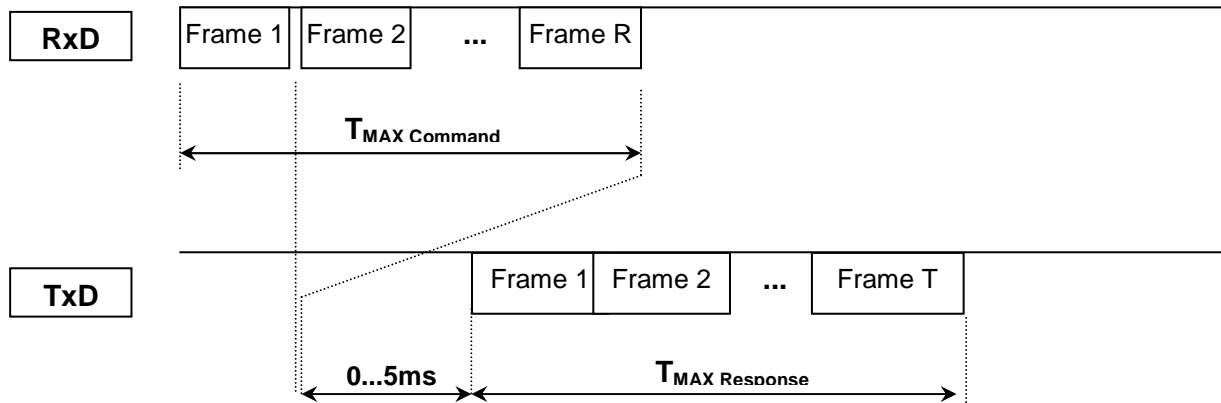


Table 8 Communication flow chart

Following timing limits in the communication flow have to be kept:

- if after the time $T_{MAX\ Command} = 50\ ms$ a whole command is not transmitted completely to the lamp driver, the command queue is cleared (this is necessary to synchronize the command decoding algorithm with the projector electronics) and the error byte ABh is returned;
- the response will start earliest after having received the first frame of a command and latest after having received the last frame of a command plus a processing time, which can vary between zero and 5 ms. The whole response time is guaranteed to be finished after $T_{MAX\ Response} = 10\ ms$.

NOTE: Because no special protocol and hardware handshake is used, we highly recommend to send a command only after having received the answer of the previous command. This is to avoid an overrun of the receiving buffer and to ensure the proper execution of the commands!

Frame Parameter	Value	Tolerance
Baud Rate	9600	±1%
Number of Start Bits	1	n/a
Number of Data Bits	8	n/a
Number of Stop Bits	1	n/a
Parity Bit	Even	n/a

Table 9: UART Parameters

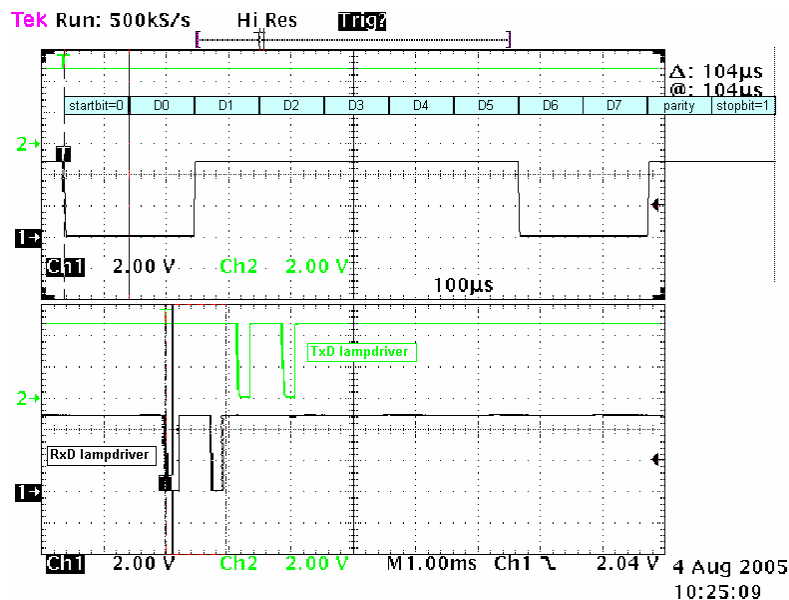
The **parity bit** is calculated by doing an **exclusive-or** of all the data bits. As the defined parity is “even”, the relation between the parity bit and the data bits ($D_0 \dots D_7$) is as follows:

$$Parity = D_0 \oplus D_1 \oplus D_2 \oplus \dots \oplus D_7$$

In other words, the parity bit is 0 if the number of “1” bits is even.

3.3. Example of a single Byte command

Following oscilloscope image is an example of transmit and receive signal for a one byte transmission:



The command transmitted from the projector is an “Enable Communication” command, in this example by means of the command byte “3Eh”. The binary conversion of “3Eh” is 00111110, represented by the last signal in the figure above and with an enlarged scale in the first position: please note that the first bit, D_0 , is the least significant one.

The green signal in the middle is the lamp driver’s answer (command echo). The measurement was done between signal pins and projector GND, using 5V supply voltage.

4. Examples

4.1. Example 1: Reading out current pulse over plateau ratio

In this example we want to read out the actual pulse over plateau ratio which depends on the current lamp voltage. First of all we need to know where the values for pulse and plateau current are stored. This depends on the kernel and might change from version to version! For our example let us assume using the DA03 kernel. For this kernel the pulse current is stored in the RAM registers **0164h** (low byte) and **0165h** (high byte). The plateau current is stored in RAM registers **0166h** and **0167h**. Both are 16 bit values stored in order low byte first, then high byte.

	Send to lamp driver	Receive
Enable Communication	3Eh	3Eh
Read pulse current low byte from RAM	88h 01h 64h	88h 01h 64h E8h
Read pulse current high byte from RAM	88h 01h 65h	88h 01h 65h 03h
Read plateau current low byte from RAM	88h 01h 66h	88h 01h 66h BCh
Read plateau current high byte from RAM	88h 01h 67h	88h 01h 67h 02h

In this example we read out the values **03E8h** (decimal 1000) for the pulse current and **02BCh** (decimal 700) for the plateau current. The pulse over plateau ratio is:

$$\frac{\text{pulse}}{\text{plateau}} = \frac{1000}{700} = 1.43$$

Therefore the pulse current is 43% over the plateau current.

5. Application Note Change Log

2005-11-01 Initial version 1.0

- Transferring content from "Application Note - Use of 5pin UART-Interface"

2006-01-01 Changes from version 1.0 to 1.1:

- Corrections at Table 5, Bit0

2006-08-21 Changes from version 1.1 to 1.2:

- Added chapter "Examples"

Authors: B.Reiter / A.Huber / C.Breuer