



Eye-BERT Micro Software Programming Guide

USB Driver:

In order for Windows to recognize the Eye-BERT Micro the USB driver must first be installed, after which the Eye-BERT Micro appears as an additional COM port on the computer. Currently Windows XP, Vista, and 7 are supported.

1. Copy the file "cdc_NTXPV764.inf" from the supplied CD to the hard drive.
2. Plug the Eye-BERT into a free USB port. When the hardware installation wizard asks for the driver location, browse to the "cdc_NTXPVista.inf" file on the hard drive.
3. After the driver has been installed right click "my computer" and select "properties". In the properties window select the "hardware" tab. Click on "device manager" and expand the "Ports (COM & LPT)" item. Locate the "Spectronix, Inc." entry and note the assigned COM number, (ie "COM4"). This is the COM port that the software will use to communicate with the Eye-BERT Micro.

Note, on some operating systems such as Window 7, manual USB driver installation may be necessary. If the hardware installation wizard fails, go to "My Computer" > "Properties" > "Hardware" > "Device Manager", and find the "Spectronix" or "SERIAL DEMO" entry under "Other Devices" and select "Update Driver". At this point you will be able to browse to the location of the driver.

USB Commands:

The Eye-BERT Micro uses a combination of ASCII and binary data to communicate with a host computer; the tables below list the individual commands, parameters, and responses from the Eye-BERT Micro.

Notes:

1. All communication is initiated by the host.
2. Commands are not case sensitive.
3. A space or equal sign should be inserted between the command and any parameters.
4. All commands should be terminated with a <CR> <LF>.
5. Responses from the Eye-BERT are not terminated unless noted.
6. Text inside quotations (""") are ASCII and those not in quotations are binary.

| Get Unit Information | |
|---------------------------|--|
| <u>Command:</u> | <u>Parameters:</u> |
| "?" | (none) |
| <u>Response:</u> | <u>Parameters:</u> |
| Unit name | "Eye-BERT Micro: " |
| Transceiver Vendor | (string, up to 18 characters) |
| Transceiver Serial Number | (string, up to 18 characters) |
| String Termination | 0x00 |
| Notes: | Useful for identifying the unit type when multiple units are used with the same computer |

| Set the wavelength (V 2.0 and above) | |
|--------------------------------------|-------------------------------|
| <u>Command:</u> | <u>Parameters:</u> |
| "SetWL" | "#####.##" (Wavelength in nm) |
| <u>Response:</u> | <u>Parameters:</u> |
| (none) | |
| Notes: | Example: "setwl=1550.12" |

| Set the data rate | |
|-------------------|---|
| <u>Command:</u> | <u>Parameters:</u> |
| "SetRate" | "#####" (Bit Rate in bps) |
| <u>Response:</u> | <u>Parameters:</u> |
| (none) | |
| Notes: | Example: "setrate=155520000" for 155.52Mbps |

| Set the pattern (generator and detector) | |
|--|---|
| <u>Command:</u> | <u>Parameters:</u> |
| "SetPat" | "7" (PRBS 2^7-1) "2" (PRBS $2^{23}-1$) "3" (PRBS $2^{31}-1$) "K" (K28.5) "J" (CJTPAT) "R" (CRPAT) |

| | |
|------------------|--|
| | <p>"S" (GSPAT)</p> <p>"C#####" (custom, "#####" = 10 ASCII coded hex characters representing a 40 bit repeating pattern)</p> <p>"L" (Loop-back: data on the input is retransmitted on the output)</p> |
| <u>Response:</u> | <u>Parameters:</u> |
| (none) | |
| Notes: | Example: "setpat=8C14FAC14FA" generates the K28.5 pattern using the custom pattern option |

| Turn Transceiver Laser on / off | |
|---------------------------------|--|
| <u>Command:</u> | <u>Parameters:</u> |
| "TX" | <p>"0" (laser off)</p> <p>"1" (laser on)</p> |
| <u>Response:</u> | <u>Parameters:</u> |
| (none) | |
| Notes: | Example: "tx=1" turns the laser on |

| Reset error counters, BER, and test timers | |
|--|--------------------|
| <u>Command:</u> | <u>Parameters:</u> |
| "Reset" | (none) |
| <u>Response:</u> | <u>Parameters:</u> |
| (none) | |
| Notes: | |

| Get Transceiver Register Information and Values | |
|---|--|
| <u>Command:</u> | <u>Parameters:</u> |
| "SFP" | |
| <u>Response:</u> | <u>Parameters:</u> |
| SFP information | (ASCII text formatted information about the SFP including: Vendor, Model, Serial Number, Power Levels, and data from both the ID and Diagnostic registers) |
| Notes: | |

| Get Transceiver Register Information and run Automated Performance Testing | |
|--|--------------------|
| <u>Command:</u> | <u>Parameters:</u> |
| "Test" | |

| | |
|---|---|
| <u>Response:</u> | <u>Parameters:</u> |
| SFP information and Test Results | <i>(ASCII text formatted information about the SFP including: Vendor, Model, Serial Number, Power Levels, and data from both the ID and Diagnostic registers)</i> |
| Notes: | <i>Test results are evaluated on the information read from the SFP registers. The test takes about 17 seconds to complete.</i> |

| Read the measurement results | |
|---|---|
| <u>Command:</u> | <u>Parameters:</u> |
| "R" | (none) |
| <u>Response:</u> | <u>Parameters:</u> |
| Bit Rate (four bytes) | $Rate (bps) = \text{byte1} * 2^{24} + \text{byte2} * 2^{16} + \text{byte3} * 2^8 + \text{byte4}$ A Bit Rate value of 0 indicates that there is a frequency error most likely caused by an out of range value. |
| PRBS (one byte) | (per "SetPat" command above) |
| Receive Optical power in dBm (two bytes) | $Power (dBm) = (32768 - (\text{byte1} * 256 + \text{byte2})) / 100$ |
| Transmit Optical power in dBm (two bytes) | $Power (dBm) = (32768 - (\text{byte1} * 256 + \text{byte2})) / 100$ |
| Transmit Wavelength (three bytes) | $Wavelength (nm) = (\text{byte1} * 2^{16} + \text{byte2} * 2^8 + \text{byte3}) / 100$ |
| SFP Temperature in °C (two bytes) | $Temp (°C) = (32768 - (\text{byte1} * 256 + \text{byte2})) / 100$ |
| Optical Receiver status (one byte) | <ul style="list-style-type: none"> 0 (not used) 1 (no signal) 2 (signal and sync) 3 (signal but no lock) |
| Total bit count (four bytes) | $Count = (\text{byte1} * 2^{16} + \text{byte2} * 2^8 + \text{byte3}) * 2^{(\text{byte4} - 24)}$ |
| Error count (four bytes) | $Errors = (\text{byte1} * 2^{16} + \text{byte2} * 2^8 + \text{byte3}) * 2^{(\text{byte4} - 24)}$ |
| Termination (one byte) | 0x00 (termination character) |
| Notes: | BER = Errors / Count |

| Read SFP Register (V 1.3 and above) | |
|--|--|
| <u>Command:</u> | <u>Parameters:</u> |
| "RdSFP" | <p>"t" "#" "l" : register type – either "I" for information or "D" for diagnostic, "#": register number in hex</p> <p><i>Example: "RdSFP I 0x44"</i></p> <p><i>Reads the first byte of the serial number from the information register at address 0x44</i></p> |
| <u>Response:</u> | <u>Parameters:</u> |
| Register type, register number, value | <p><i>Example: "a0:44 = 35"</i></p> <p><i>(information register (0xA0), register number (0x44), value (5 ASCII))</i></p> |
| <u>Notes:</u> | <p><i>The physical address of the information register is 0xA0 and the physical address of the diagnostic register is 0xA2. All values passed in and returned are in hex, preceding "0x" is optional. Input parameters should be separated by a space. Note, not all SFP vendors support reading and writing all locations. See SFF-8472 for more information.</i></p> |

| Write SFP Register, then respond with read back value (V 1.3 and above) | |
|---|---|
| <u>Command:</u> | <u>Parameters:</u> |
| "WrSFP" | <p>"t" "#" "v" "l" : register type – either "I" for information "D" for diagnostic, or "T" for tuning, "#": register number in hex, "v": value to be written in hex.</p> <p><i>Example: "WrSFP D 0x80 0x55"</i></p> <p><i>Writes 0x55 to the first byte of the user writable EEPROM area at register at address 0x80.</i></p> |
| <u>Response:</u> | <u>Parameters:</u> |
| Register type, register number, value | <p><i>Example: "a2:80 = 55"</i></p> <p><i>(diagnostic register (0xA2), register number (0x80), value read back (0x55))</i></p> |
| <u>Notes:</u> | <p><i>The physical address of the information register is 0xA0 and the physical address of the diagnostic and tuning register is 0xA2. All values passed in and returned are in hex, preceding "0x" is optional. Input parameters should be separated by a space. Note, not all SFP vendors support reading and writing all locations. See SFF-8472 for more information.</i></p> |

Factory / Test Commands:

| Print Terminal Formatted Menu / Status | |
|--|--------------------|
| <u>Command:</u> | <u>Parameters:</u> |
| "!" | (none) |
| <u>Response:</u> | <u>Parameters:</u> |
| Notes: | |

| Toggles Command Echo | |
|----------------------|--------------------|
| <u>Command:</u> | <u>Parameters:</u> |
| "echo" | (none) |
| <u>Response:</u> | <u>Parameters:</u> |
| Notes: | |

| Clears the Terminal Screen | |
|----------------------------|--------------------|
| <u>Command:</u> | <u>Parameters:</u> |
| "Clear" | (none) |
| <u>Response:</u> | <u>Parameters:</u> |
| Notes: | |

| Displays the Si5338 clock generator Registers | |
|---|--------------------|
| <u>Command:</u> | <u>Parameters:</u> |
| "RdCG" | (none) |
| <u>Response:</u> | <u>Parameters:</u> |
| Notes: | |

| Displays the CDR Registers | |
|----------------------------|--------------------|
| <u>Command:</u> | <u>Parameters:</u> |
| "RdCDR" | (none) |
| <u>Response:</u> | <u>Parameters:</u> |
| Notes: | |

| Performs Self Test using a loopback cable | |
|---|--------------------|
| <u>Command:</u> | <u>Parameters:</u> |
| "Bist" | (none) |

| | |
|---------------------------|-----------------------------|
| Response: | Parameters: |
| Notes: | |

| Map the bit rates where the CDR can lock | |
|--|---|
| Command: | Parameters: |
| "Map" | "#" (<i>bit rate resolution in %</i>) |
| Response: | Parameters: |
| (none) | |
| Notes: | <i>Example: "map=10" tests the CDR locking from 100Mbps to 4.29Gb increasing each point by 10%.</i> |