# COBRA MULTISPEC & HYPERSPEC

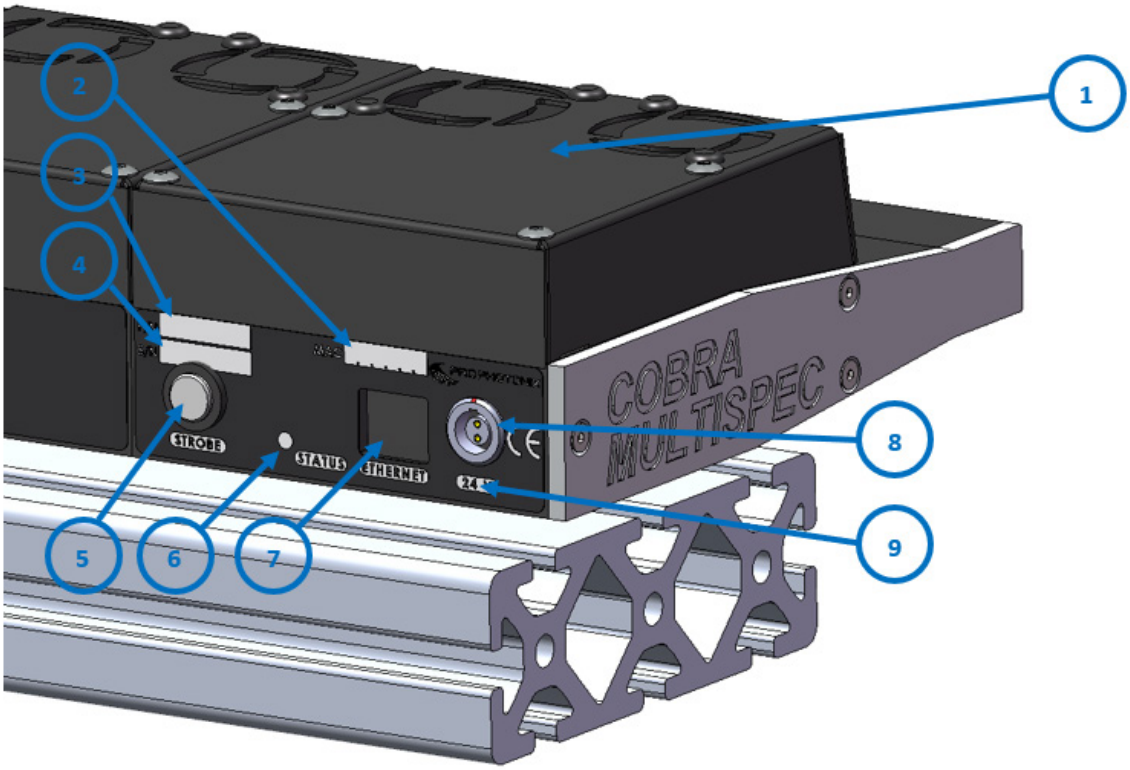
Operations Manual 1.1
16 Nov. 2022


PRO PHOTONIX
An Exaktera Company

# Table of Contents

# 1.0 Electrical Specification

## 1.1 Hardware Overview (EL and AL Versions)




| | |
|---|---|
| 1 | Fan (TEL and TAL part number only) |
| 2 | Ethernet MAC address |
| 3 | Part number |
| 4 | Serial number |
| 5 | Strobe input (9-contact circular LEMO 736-FGG0T309CLAC50Z) |
| 6 | Status Indicator |
| 7 | Ethernet Connector (standard RJ45) |
| 8 | 24V input power connector (e.g. Lemo EGG.1B.302.CLL) |
| 9 | Ethernet IP Reset Switch (behind power connector) |

## 1.2 Power Supply

The COBRA MultiSpec/HyperSpec line light is powered by a 24VDC ±10% power supply. Each 100mm module can take up to 2Amps (Part number TxL).

## 1.3 Status Indicator

A status indicator LED is fitted at the back of each 100mm module. This indicator gives information on the light condition.

| Status Indicator | Meaning |
|---|---|
|  | Solid red on → thermal error |
|  | Blinking orange → broken CAN-bus connection |
|  | Flash green → flash their 'position' according to module number |

## 1.4 Thermal

To improve LEDs stability and reduce system noise and maintenance, a pair of fans are installed (by default) on top of each 100mm module. In the COBRA MultiSpec/HyperSpec, these fans are switched on permanently.

Every 100mm module in the unit has its own over temperature protection. In the event of fan failure or if the ambient temperature gets too high, an automatic thermal shut down is initiated in the affected module in order to protect the LEDs. The LEDs are forced off if the substrate temperature reaches 60°C and stays off until the substrate cools down to 45°C. The same applies for low temperatures: the light will not start for sub zero temperatures.

## 1.5 Error Detection

The COBRA MultiSpec/HyperSpec continuously checks for a 24VDC input power and current limit. Its self-diagnostics will detect and react to any of the 3 failures listed in the table below.

| Error Name | What Triggered the Error | COBRA Action |
|---|---|---|
| Power Limit | Power is exceeded | Soft Error. Reset by turning off light and back on again. |
| Current Trip | Hardware current trip | Hard Error. Special command: "TRST=1" needs to be sent to the unit |
| Over/Under Temperature | The substrate temperature has reached 60°C or is below 0°C. | Flag the error and force the LEDs off until the temperature reaches between 0°C and 45°C. |

The errors are flagged via the "Status Indicator" and the "Operating Status Register" (see 2.4.5 Operating Status.

The "Status Indicator" will indicate solid red as long as the error is detected and for 5 more seconds after the error has been rectified.

## 1.6 Strobe Input (AL Version ONLY)

The "Strobe Input" for the COBRA MultiSpec and HyperSpec product is an 8-core (4 twisted pairs) cable allowing for individual strobe control of up to 4 strobe groups.

| Colour | Designation |
|---|---|
| Brown/White | Strobe 1 |
| Black/White | Strobe 2 |
| Orange/White | Strobe 3 |
| Red/White | Strobe 4 |

All strobe signals can be kept electrically separate, or commoned by joining respective Returns (White leads).

The "Strobe Input" can be directly controlled by a 0-5 Vpk-pk (typical), 0-3 Vpk-pk (minimum), 0-24 Vpk-pk (maximum) signal with input current draw at 1.8-2.4mA (typical), 4.6mA (maximum). Reverse input signal voltage must not exceed 5V.

# 2.0 Ethernet Control

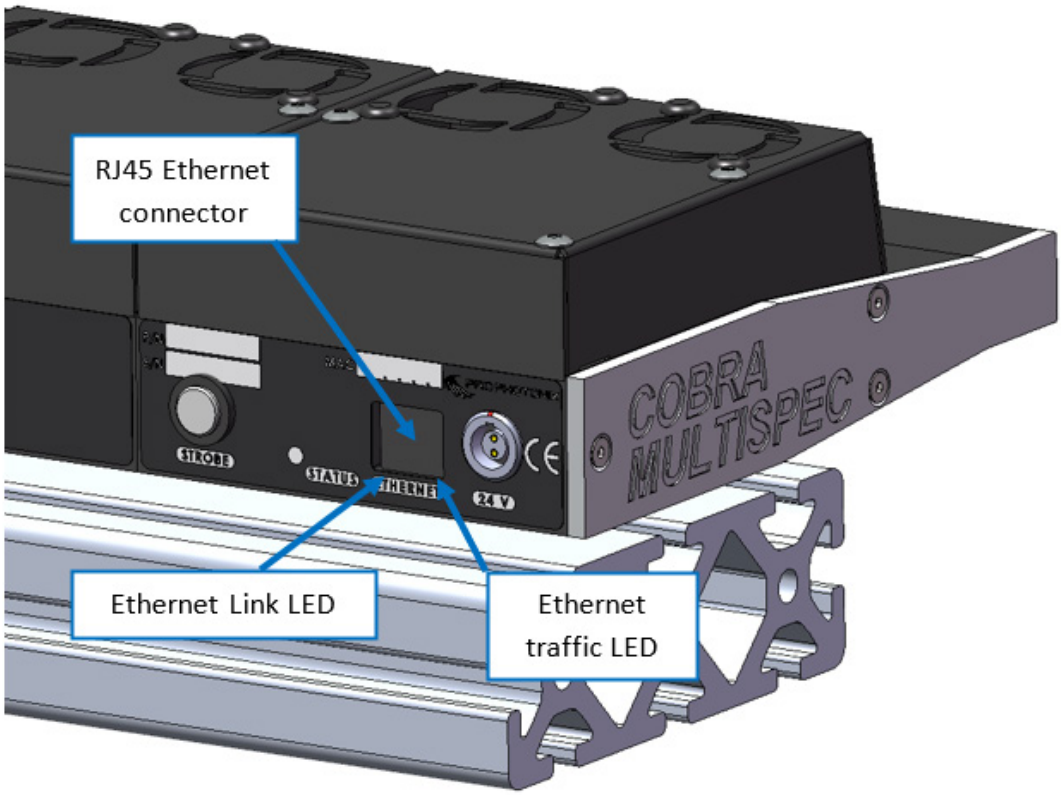
## 2.1 Overview

The Ethernet control option allows full control of the COBRA MultiSpec/HyperSpec Line Light. It also allows the user to reconfigure the unit, monitor performances and upgrade firmware.

Acting as a TCP/IP server with a fixed IP address, the COBRA MultiSpec/HyperSpec uses a simple text based protocol to communicate with a computer directly or over a local area network (LAN).

## 2.2 Hardware interface

### 2.2.1 MultiSpec/HyperSpec (EL and AL versions)




### 2.2.2 Connection

RJ45 terminated Ethernet cable is used to connect the COBRA Line light to a computer, network or a hub.
NOTE: Some customers' older computers may require an Ethernet cross-over cable

### 2.2.3 Ethernet Settings Reset

The COBRA MultiSpec/HyperSpec IP address and subnet mask can be changed to suit your network configuration (see 2.5 Ethernet Settings).

The default Ethernet settings are:

- IP Address: 10.0.0.10
- Subnet mask: 255.255.255.0

To restore the COBRA MultiSpec/HyperSpec to the default Ethernet settings the user needs to press the IP reset switch while the COBRA MultiSpec/HyperSpec is powered (see 1.1 Hardware Overview).

For the COBRA MultiSpec/HyperSpec, this is hidden under the power connector, behind a tiny hole. A thin object (preferably non-metal) should be used.

## 2.3 Communication Protocol

The communication is based on a simple TCP/IP text protocol with the COBRA acting as a server.
The protocol uses both query ("?") and control ("=") commands which can be followed or not by one or several arguments separated by a dot (".") and ended by a carriage return character ("CR" 13 dec, 0D hex).

Each command received is acknowledged by the COBRA with one or several values separated by a dot (“.”). Like the commands, acknowledgements are ended by a carriage return character.

## 2.3.1 Command Syntax:

Each command received is acknowledged by the COBRA with one or several values separated by a dot (“.”). Like the commands, acknowledgements are ended by a carriage return character.

### 2.3.1 Command syntax:

| | | |
|---|---|---|
| Command without argument: | <COMMAND><CR> | e.g.: “GOS?< CR >” |
| Command using 1 argument: | <COMMAND><ARG1>< CR > | e.g.: “MLI?5<CR>” |
| Command using 2 arguments: | <COMMAND><ARG1><.><ARG2>< CR > | e.g.: “MLI=5.200<CR>” |
| Command using multiple arguments: | | e.g.: “IPA=10.0.0.10<CR>” |

MLI=12.128<CR>

- Command end: Carriage Return character.
- Argument #2: ASCII characters representing a decimal or hexadecimal number, e.g.: “12”, “1F”
- Delimiter: Dot ASCII character: “.”
- Argument #1: ASCII characters representing a decimal or hexadecimal number, e.g.: “12”, “1F”
- Command: ASCII characters terminated by “?” for a query or “=” for a control, e.g.: “MLI=”, “MLI?”

## 2.3.2 Response Syntax:

| | | |
|---|---|---|
| One value Response: | <VAL1><CR> | e.g.: “1<CR>” |
| Two values Response: | <VAL1><.><VAL2><CR> | e.g.: “12.28<CR>” |
| Multiple values Response: | | e.g.: “10.0.0.10<CR>” |

| Command Type | Response | Meaning |
|---|---|---|
| Control | 1 i.e.: “1<CR>” | Command succeed |
| Control | -1 i.e.: “-1<CR>” | Command not recognized or value out of range. |
| Control | -2 i.e.: “-2<CR>” | At least one module did not answer (e.g. module not powered) or where command arguments are not correct |
| Query | Positive decimal or hexadecimal value(s). i.e.: “10.0.0.10<CR>” | Answer to the query. |
| Query | -1 i.e.: “-1<CR>” | Command not recognized. |
| Query | -2 i.e.: “-2<CR>” | At least one module did not answer (e.g. module not powered). Not all modules respond with the same value, for example, GLI? when modules are set to different intensity levels |

## 2.4 Commands

The specific COBRA MultiSpec/HyperSpec command-set is listed below. Rev1 do not appear in this document; however they can still be used.

### 2.4.1 System Status

- Global Input Current query:

Query the highest input current for a single module across the entire system (this includes the fans).

Syntax: GIIN?<CR>

Example: "GIIN?<CR>" → Query the input current for all the modules together.

Response: "1373<CR>" → 1373mA input current is the highest current being drawn by a single module across the entire system.

- Module Input Current query:

Query the input current for a particular module (this includes the fans).

Syntax: MIIN?<module#><CR>

Example: "MIIN?1<CR>" → Query the input current for module #1 in the system.

Response: "1373<CR>" → 1373mA input current is being drawn through module #1

Example: "MIIN?2<CR>" → Query the input current for module #2 in the system.

Response: "-2<CR>" → Module #2 doesn't exist on the system or cannot be queried

- Global Input Power query:

Query the highest power consumption for a single module across the entire system (this includes the fans).

Syntax: GPIN?<CR>

Example: "GPIN?<CR>" → Query the highest power consumption for a single module across the entire system.

Response: "32869<CR>" → 32869mW (32.869W) is the highest power consumption for a single module across the entire system.

Note: it does not designate which module this is. Individual Module Input Power queries or the Input Power Array query are required for this (see below).

- Input Power Array query:

Query the power consumption for all modules across the entire system (this includes fans).

Syntax: ARR?<CR>

Example: "ARR?<CR>" → Query the power consumption as an array for all modules across the entire system

Response: "2812,2049,2100,2213<CR>" → Power consumption for module #1 through to module #4 is 2812mW, 2049mW, 2100mW, and 2213mW respectively. For a 4-module system.

- Module Input Power query:

Query the power consumption for a particular module (this includes the fans).

Syntax: MPIN?<module#><CR>

Example: "MPIN?1<CR>" → Query the power consumption for module #1 in the system.

Response: "32869<CR>" → 32869mW (32.869W) power consumption for module #1

Example: "MPIN?2<CR>" → Query the power consumption for module #2 in the system.

Response: "-2<CR>" → Module #2 doesn't exist on the system or cannot be queried

- ## Global Version Number query:

Query the software version number of all the light modules. This assumes the same software version is installed on each module. "-2" is returned if at least one module has a different software version or if a module did not answer (see 2.4.8 Answer array).

Syntax: GVN?<CR>
Example: "GVN?<CR>" → Query the software version of all the modules.
Response: "1.7<CR>" → Every module uses the software version 1.7.
Response: "-2<CR>" → At least one module has a different software version.

- ## Module Version Number query:

Query the software version number of a single module.

Syntax: MVN?<module#><CR>
Example: "MVN?12<CR>" → Query the software version of the module 12.
Response: "1.7<CR>" → The module 12 uses the software version 1.7.

- ## Global Number of Modules query:

Query the number of modules connected in this light.

Syntax: GNM?<CR>
Example: "GNM?<CR>" → Query the number of module connected.
Response: "22<CR>" → There are 22 modules connected (2.2 meter light).

- ## Global System Status query:

Query the light status of all the modules – light ON or OFF. "1" is returned if all the lights are on, 0 if all the lights are off. "-2" is returned if at least one module has a different status or if a module did not answer (see 2.4.8 Answer array).

Syntax: GSS?<CR>
Example: "GSS?<CR>" → Query the light status of all the modules.
Response: "1<CR>" → All the lights are on.
"0<CR>" → All the lights are off.
"-2<CR>" → At least one module has a different status.

- ## Module System Status query:

Query the light status of a single module – light ON or OFF. "1" is returned if the module light is on, 0 if the module light is off.

Syntax: MSS?<module#><CR>
Example: "MSS?10<CR>" → Query the light status of the module 10.
Response: "1<CR>" → The light of the module #10 is on.
"0<CR>" → The light of the module #10 is off.
"-2<CR>" → The module #10 did not answer.

- ### Global System Status control:

Set the light status of all the modules – light ON or OFF. "1" is returned if the command has succeeded for every module. "-2" is returned if the command did not succeed for at least one module (see 2.4.8 Answer array).

Syntax: GSS=<value><CR>

Example: "GSS=0<CR>" → Turn off the light of all the modules.

"GSS=1<CR>" → Turn on the light of all the modules.

Response: "1<CR>" → All the lights have been set according to the command.

"-2<CR>" → At least one module did not answer.

- ### Module System Status control:

Set the light status of a single module – light ON or OFF. "1" is returned if the command has succeeded.

Syntax: MSS=<module#>.<value><CR>

Example: "MSS=5.0<CR>" → Turn off the light of the module 5.

"MSS=5.1<CR>" → Turn on the light of the module 5.

Response: "1<CR>" → The light has been set according to the command.

"-2<CR>" → The module #5 did not answer.

## Intensity Control

The COBRA MultiSpec/HyperSpec offers the possibility to control the intensity of each individual wavelength, for each individual module (see MLIX command) or globally for all modules (GLIX command). In addition, each module has a Master intensity setting, which controls the overall light intensity, without affecting the relative intensities set up with the GLIX and MLIX commands.

By default, the Master intensity setting is set to 1023, allowing each channel to reach the highest intensity possible.

The resulting intensity setting for each channel can be expressed as:

[Master intensity value] * [Channel intensity (MLIX or GLIX) value] / 1023

- ### Light Intensity Range query:

Query the light intensity range.

Syntax: LIR?<CR>

Example: "LIR?<CR>" → Query the light intensity range.

Response: "1023<CR>" → The light intensity ranges is set to 0-1023.

- ### Global Light Intensity query:

Query the light intensity of all the modules. The light intensity is returned if all the modules are set to the same level. "-2" is returned if at least one module has a different light intensity or if a module did not answer (see 2.4.7 Answer array).

Syntax: GLI?<CR>

Example: "GLI?<CR>" → Query the light intensity of all the modules.

Response: "128<CR>" → All the light intensities are set to 128.

"-4<CR>" → At least one module has a different light intensity.

- Module Light Intensity query:

Query the light intensity of a specific module. The light intensity of the module is returned.

Syntax: MLI?<module#><CR>

Example: "MLI?10<CR>" → Query the light intensity of the module 10.

Response: "200<CR>" → The light intensity of the module #10 is set to 200.

"-2<CR>" → The module #10 did not answer.

- Global Master Light Intensity query:

Query the master light intensity of all the modules. The master light intensity is returned if all the modules are set to the same level. "-2" is returned if at least one module has a different master light intensity or if a module did not answer (see 2.4.8 Answer array).

Syntax: GMAS?<CR>

Example: "GMAS?<CR>" → Query the master light intensity of all the modules.

Response: "1023<CR>" → All the master light intensities are set to 1023 (maximum value, default).

Response: "-2<CR>" → Modules are set to different master light intensities, or at least one module did not answer.

- Module Master Light Intensity query:

Query the master light intensity of a specific module. The master light intensity of the module is returned.

Syntax: MMAS?<module#><CR>

Example: "MMAS?7<CR>" → Query the master light intensity of module 7.

Response: "600<CR>" → The light intensity of module #7 is set to 600.

"-2<CR>" → The module #7 did not answer.

- Global Master Light Intensity control:

Set the master light intensity of all the modules to the same value. "1" is returned if the command has succeeded for every module. "-2" is returned if the command did not succeed for at least one module (see 2.4.8 Answer array). The value entered must be in the range 0...1023.

Syntax: GMAS=<value><CR>

Example: "GMAS=950<CR>" → Set the light intensity of every module to 950.

Response: "1<CR>" → All the lights have been set according to the command.

"-2<CR>" → At least one module did not answer.

- Module Master Light Intensity control:

Set the master light intensity of a specific module. "1" is returned if the command has succeeded.

Syntax: MMAS=<module#>.<value#><CR>

Example: "MLI =5.750<CR>" → Set the light intensity of the module #5 to 750.

Response: "1<CR>" → The light has been set according to the command.

"-2<CR>" → The module #5 did not answer.

## 2.4.3 Individual Wavelength Control

Each wavelength in the COBRA MultiSpec and COBRA HyperSpec can be individually controlled at either a Global level, or at a Module level.

Use of the channel name "0" addresses all channels in the GLIX and MLIX commands and queries all channels simultaneously.

- **Global Wavelength Intensity control:**

Sets a single wavelength intensity on all modules. "1" is returned if the command succeeded for every module. "-2" is returned if the command did not succeed for at least one module (see 2.4.8 Answer array).

Syntax: GLIX=<Wavelength>.<value#><CR>
Example: GLIX=R.500<CR> Set the light intensity of Red across all modules to 500
Response: "1<CR>" → All the lights have been set according to the command.
Example: GLIX=R.500<CR> → Set the light intensity of Red across all modules to 500
Response: "-3<CR>" → The specified wavelength (channel) code doesn't exist in the unit's configuration.

- **Global Wavelength Channel Intensity control:**

Sets a single wavelength channel intensity on all modules. "1" is returned if the command succeeded for every module. "-2" is returned if the command did not succeed for at least one module (see 2.4.8 Answer array).

Syntax: GLIX=<Wavelength><WavelengthChannel#>.<value#><CR>
Example: GLIX=R1.500<CR> → Set the light intensity of the first Red channel across all modules to 500
Example: GLIX=0.500<CR> → Set the light intensity of all channels across all modules to 500.
→ This command is equivalent to "GLI=0<CR>"

- **Module Wavelength Intensity control:**

Sets a single wavelength intensity of a specific module. "1" is returned if the command succeeded. "-2" is returned if the command did not succeed for that specific module (see 2.4.8 Answer array).

Syntax: MLIX=<module#>.<Wavelength>.<value#><CR>
Example: MLIX=1.R.500<CR> → Set the light intensity of all channels of Red in module 1 to 500
Response: "1<CR>" → The module light intensity has been set to the command.
Response: "-2<CR" → The module #1 did not answer

- **Module Wavelength Channel Intensity control:**

Sets a single wavelength intensity of a specific channel in a specific module. "1" is returned if the command succeeded. "-2" is returned if the command did not succeed for that specific channel in that specific module (see 2.4.8 Answer array).

Syntax: MLIX=<module#>.<Wavelength><WavelengthChannel#>.<value#><CR>
Example: MLIX=1.R1.500<CR> → Set the light intensity of the first Red channel in module 1 to 500
Response: "1<CR>" → The module channel light intensity has been set to the command.
Response: "-2<CR" → The module #1 did not answer

## 2.4.4 Substrate Temperature

- **Global Temperature Value query:**

Query the substrate temperature of all the modules in degrees Celsius (°C). The highest substrate temperature among the modules is returned. "-2" is returned if at least one module did not answer (see 2.4.8 Answer array). Two query formats are possible:

Syntax: GTV?<CR>
Example: "GTV?<CR>" → Query highest substrate temperature.
Response: "48<CR>" → The highest substrate temperature is 48°C.
"-2<CR>" → At least one module did not answer.

Syntax: GHRTV?<CR>
Example: "GHRTV?<CR>" → Query highest high resolution substrate temperature.
Response: "4823<CR>" → The highest substrate temperature is 48.23°C.
"-2<CR>" → At least one module did not answer.

- Module Temperature Value query:

Query the substrate temperature of a specific module. The substrate temperature of the module is returned.

Syntax: MTV?<module#><CR>
Example: "MTV?28<CR>" Query highest substrate temperature.
Response: "45<CR>" → The module #28 substrate temperature is 45°C.
"-2<CR>" → At least one module did not answer.
→

Syntax: MHRTV?<module#><CR>
Example: "MHRTV?28<CR>" Query the module #28 substrate temperature.
Response: "4513<CR>" → The module #28 high resolution substrate temperature is 45.13°C.
"-2<CR>" → The module #28 did not answer.
→

## 2.4.5 Operating Status

The operating status register is used to latch the errors detected until read by the user via the Ethernet commands "GOS" or "MOS". Each reading clears the error flags. If the error is still active, the error flag is immediately set again.
The operating status is a bit mapped register where each bit is an error flag as described below:

| Operating Status Answer | Binary Representation | Error Flagged |
|---|---|---|
| 1 | 00000001 | No error detected.<br>The system is working properly. |
| 2 | 00000010 | Reset flag.<br>The module has been reset. This flag is set after each power-up or software reset and is cleared after a reading using the commands GOS? or MOS? |
| 8 | 00001000 | Over/under Temperature flag.<br>The LEDs have been shut down due to an over or under temperature. |

The error flags can be combined if several errors took place since the last GOS/MOS command.
Example:
00001010 (decimal 10) → Over/under temperature and reset flags (see 8 and 2).

- Global Operating Status query:

Query the operating status of all the modules and clear the error flags. A "bitwise-OR" compilation of all the modules operating statuses is returned. "-2" is returned if at least one module did not answer (see 2.4.8 Answer array).

Syntax: GOS?<CR>
Example: "GOS?<CR>" → Query the operating status of all the modules.
Response: "2<CR>" → At least 1 module has been reset since last status check.
"-2<CR>" → At least one module did not answer.

- Module Operating Status query:

Query the operating status of a specific module. The operating status of the module is returned.

Syntax: MOS?<module#><CR>
Example: "MOS?22<CR>" → Query the operating status of module 22.
Response: "1<CR>" → The module 22 is OK.
"-2<CR>" → The module 22 did not answer.

## 2.4.6 Save

- SAVe:

Save the light intensity and light status for each module. Strobe mode is also stored with this command. Master settings as well as each chain's intensity settings are also saved. The saved settings are used at each light start-up. "-2" is returned if at least one module did not save the settings (see 2.4.8 Answer array). Note: the argument after "SAV=" must be "1" to validate the command, i.e.: "SAV=1".

| | | |
|---|---|---|
| Syntax: | SAV=<value><CR> | |
| Example: | "SAV=1<CR>" | → Saves the light intensity, status and fan mode of each module. |
| Response: | "1<CR>" | → Each module has saved its settings. |
| | "-2<CR>" | → At least one module did not save its settings. |

## 2.4.7 Reset

- ReSeT:

Reset the software of all the modules. This command can be used to activate new settings like the IP address, strobe mode, etc... "-2" is returned if at least one module did not answer (see 2.4.8 Answer array). Note: the argument after "RST=" must be "1" to validate the command, i.e.: "RST=1".

| | | |
|---|---|---|
| Syntax: | RST=<value><CR> | |
| Example: | "RST=1<CR>" | → Reset the software of each module. |
| Response: | "1<CR>" | → Each module is going through a reset process. |
| | "-2<CR>" | → At least one module did not answer. |

## 2.4.8 Answer Array

The array query command helps to further query the system state when the answer of a global command is insufficient.

For instance, if the "GOS" query returns "8" (over-temperature error), you can use the array command to find out which module exactly has generated this error. If the "GTV" query returns "55", it can be useful to know which module is at 55°C as well as the temperatures of all the other modules.

When the array query command is sent after a global command, the system returns the answer of each module separated by a comma (',').
If the previous command was not global, the array query returns "-1".

- Array:

| | | |
|---|---|---|
| Syntax: | ARR?<CR> | |
| Example1: | "GOS?<CR>" | → Query the operating status of all the modules. |
| Response: | "8<CR>" | → At least one module went in over-temperature. |
| Command: | "ARR?<CR>" | → Ask for the answer of every module. |
| Response: | "1,1,8,8,1<CR>" | → The modules 3 and 4 flagged an over-temperature error. |
| Example2: | "GLI?<CR>" | → Query the light intensity of all the modules. |
| Response: | "-2<CR>" | → Not all the modules have the same light intensity. |
| Command: | "ARR?<CR>" | → Ask for the answer of every module. |
| Response: | "20,20,0,20,20<CR>" | → The module 3 light intensity is set to 0. |
| Example3: | "GLI=0<CR>" | → Set the light intensity of every module to 0. |
| Response: | "-2<CR>" | → At least 1 module did not answer. |
| Command: | "ARR?<CR>" | → Ask for the answer of every module. |
| Response: | "1,1,1,1,-2<CR>" | → The module 5 did not answer. |

## 2.4.9 Strobe Commands (AL Version ONLY)

When the COBRA MultiSpec/HyperSpec is fitted with both Ethernet and Strobe options (TAL part numbers) it is possible to configure it in continuous mode or strobe mode via the "Global Strobe Mode" command "GSTM".

When in continuous mode (strobe mode = 0), the strobe input is ignored and the Ethernet "System Status" commands "GSS" and "MSS" are used to turn on and off the LEDs.

When in Strobe mode (strobe mode = 1), the Strobe input is used to turn on and off the LEDs. However, if the System Status of a module is set to 0 (off), the LEDs stay off regardless of the strobe signal state (see table below).

| Strobe Mode | System Status | Strobe Input | LEDs State |
|---|---|---|---|
| 0 → Continuous | 0 | Ignored | Off |
| | 1 | Ignored | On |
| 1 → Strobe | 0 | Ignored | Off |
| | 0 | Ignored | Off |
| | 1 | 0 | Off |
| | 0 | 1 | On |

- Global Strobe Mode query:

Query the strobe mode of all the modules. "0" is returned if the modules are in continuous mode and "1" if they are in strobe mode. "-2" is returned if at least one module has a different setting (see 2.4.8 Answer array). If a previous "GSTM= x" change has not been followed by a reset, then the response may not be valid.

Syntax: GSTM?<CR>
Example: "GSTM?<CR>" → Query strobe mode of all the modules.
Response: "1<CR>" → All the modules are in strobe mode.
"-2<CR>" → At least one module did not answer.

- Global Strobe Mode control:

Set the strobe mode of all the modules. "-2" is returned if at least one module did not answer (see 2.4.8 Answer array). The strobe mode will be activated at the next system power cycle or reset command.

Syntax: GSTM=<CR>
Example: "GSTM=0<CR>" → Set the strobe mode of all the modules to continuous.
Response: "1<CR>" → All the modules will be in continuous mode after reset.
"-2<CR>" → At least one module did not answer.

## 2.4.10 Ethernet Settings

Use the latest COBRA MultiSpec GUI and input commands in the command window. For the latest version of the GUI visit this File sharing site:

https://www.prophotonix.com/wp-content/uploads/2022/09/Multispec-GUI-install-rev1.4.zip

- IP address query:

Query the last IP address set. This IP address can be different from the present IP address if it has just been changed and the system has not yet been reset.

Syntax: IPA?<CR>
Example: "IPA?<CR>" → Query the last IP address set.
Response: "10.0.0.10<CR>" → The last IP address set is 10.0.0.10.

- IP address control:

Set the system IP address. The IP address will be activated at the next system power cycle or reset command.

Syntax: IPA=<value>.<value>.<value>.<value><CR>
Example: "IPA=10.10.37.170<CR>" → Set new IP address to 10.10.37.170.
Response: "1<CR>" → The new IP address has been saved.

- SUBnet mask query:

Query the last subnet mask set. This subnet mask can be different from the present mask if it has just been changed and the system has not yet been reset.

Syntax: SUB?<CR>
Example: "SUB?<CR>" → Query the last subnet mask set.
Response: "255.255.255.0<CR>" → The last subnet mask set is 255.255.255.0.

- SUBnet mask control:

Set the system subnet mask. The subnet mask will be activated at the next system power cycle or reset command.

Syntax: SUB=<value>.<value>.<value>.<value><CR>
Example: "SUB=255.0.0.0<CR>" → Set the new subnet mask to 255.0.0.0.
Response: "1<CR>" → The new subnet mask has been saved.

- MAC address query:

Query the MAC address. The MAC address is in hexadecimal format.

Syntax: MAC?<CR>
Example: "MAC?<CR>" → Query the MAC address.

## 2.5 Ethernet Settings

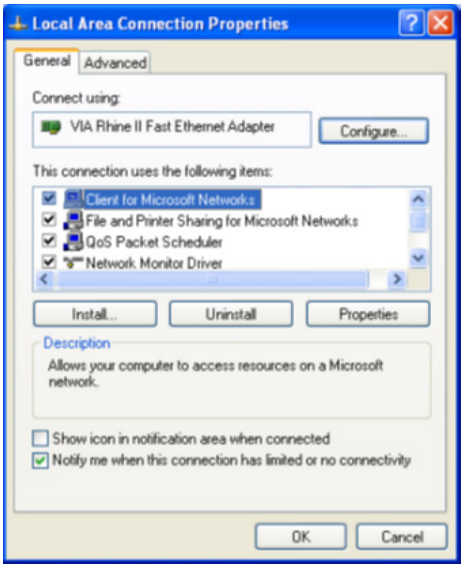
If the COBRA MultiSpec/HyperSpec is going to be used on a network, then a fixed IP address has to be reserved on that network. This fixed IP address has to be assigned to the COBRA so that computers on the network can communicate with it. This section describes how to use the graphical user interface (GUI) to set a new IP address and subnet mask on the COBRA.
For the following steps, the COBRA must be powered-up.

**********Consult you operating system provider to change the ip address**********

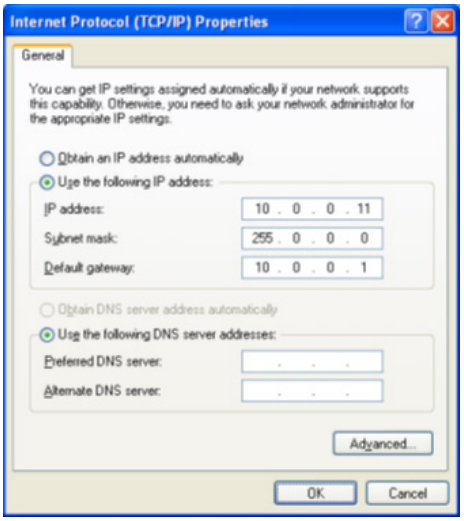
## Step 1: Temporarily reconfigure your computer network settings.

This process may very slightly depending on the Windows version you are currently using. Below is for Windows 10.

- Disconnect your computer Ethernet cable from the network.
- In the Network and Sharing Centre, click Change Adapter Settings
- Right click the proper connection (Local Area Connection) and select Properties (Note: administrative rights may be needed to change network adaptor settings)




- Scroll to the bottom and double click on Internet Protocol Version 4 (TCP/IPv4).
- NB: Make a note of your current Ethernet settings so that you can restore them after this procedure.
- Fill in the IP Address. The subnet mask should be automatically filled when clicked. Accept this value. Default gateway can be left blank. Click OK.




- Also select OK at the next menu.

## Step 2: Establishing connection with the COBRA.

- Connect the COBRA to your computer via an Ethernet cable. The COBRA must be powered.
- Run the "MultiSpecControl" application. Fill in the "IP Address" field with IP address of the COBRA you want to connect to, which is 10.0.0.10 by default.
- Click "Connect" and the GUI should open.

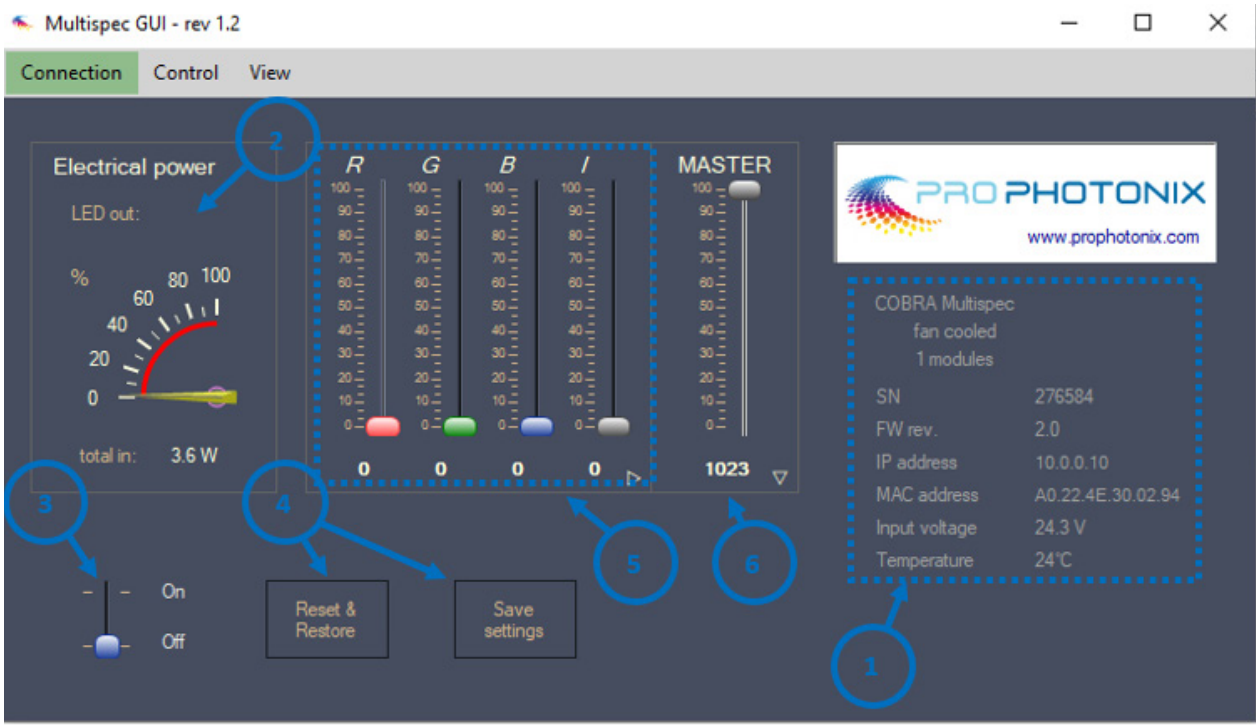
## 2.6 COBRA MultiSpec and COBRA HyperSpec GUI

The latest revision of the COBRA MultiSpec and COBRA HyperSpec graphical user interface (GUI) can be downloaded from the file sharing website listed in Section 2.4.10.

The COBRA MultiSpec and COBRA HyperSpec GUI has been designed to help the customer operate the COBRA MultiSpec and COBRA HyperSpec simply through the use of a number of individual intensity sliders that can control individual wavelengths and individual channels across the entire COBRA unit. Additionally, the GUI is accompanied by a separate command window which allows the user to input commands manually and provides even more detailed control of individual operating parameters for the COBRA unit. All relevant commands listed in Section 2.4 can be entered in this command window.

Follow the Setup instructions in Section 2.5 in order to connect to the COBRA MultiSpec or COBRA HyperSpec via the Ethernet.

Once you have connected to the COBRA, open the GUI and the following will be displayed (or similar, depending on the MultiSpec configuration:

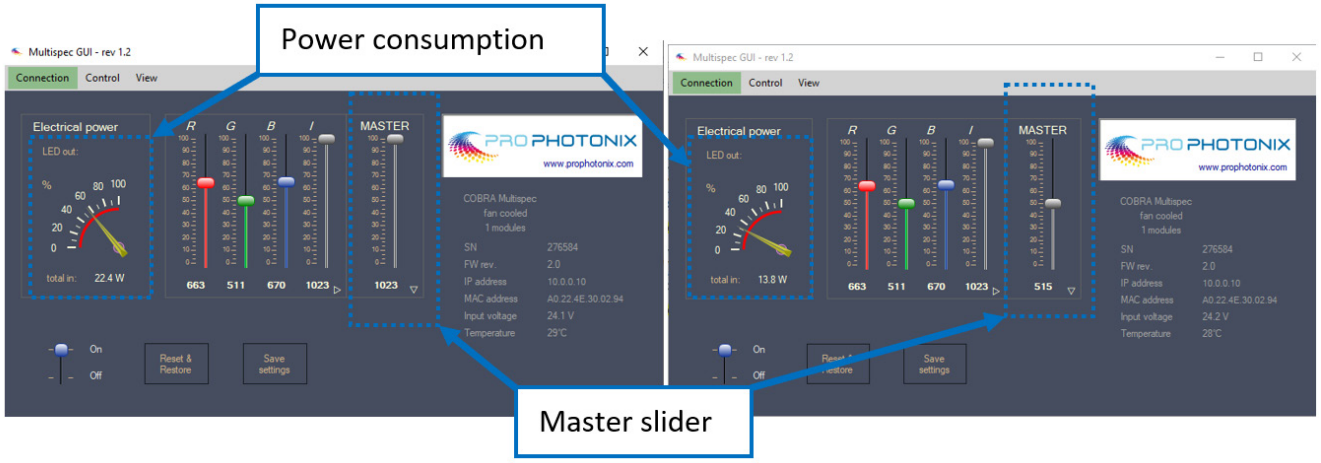



The above GUI represents a RGBIR COBRA MultiSpec. Each wavelength is represented by its own intensity slider.

| | |
|---|---|
| 1 | Product Details |
| 2 | Percentage Electrical Power. Value displayed indicates the highest percentage value across all modules. The "Total In" value indicates the total power across all modules |
| 3 | On/Off toggle |
| 4 | Settings Save, Reset & Restore (this will restart the unit) |
| 5 | Intensity sliders (0% to 100%) per wavelength |
| 6 | Master intensity slider (0% to 100%) |

## 2.6.1 Adjusting Individual Wavelength Intensity

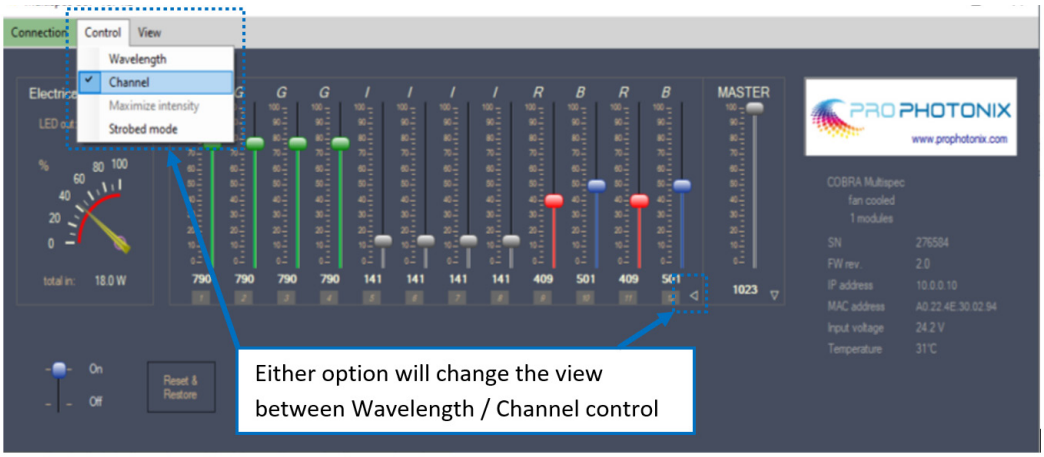
Each wavelength can be adjusted in intensity by moving the corresponding slider from 0% to 100%. The Master slider will affect each wavelength similarly. Therefore, adjusting the Master slider down from 100% to 50% (for e.g.), will maintain the same relative intensities between each wavelength, but reduce the overall power consumption.




Similarly, after adjusting individual wavelength intensities, the user can maximise the overall intensity of the COBRA (whilst maintaining the individual wavelength ratios selected) by selecting “Maximise intensity” from the “Control” drop-down bar.

## 2.6.2 Adjusting Individual Channel Intensity

By selecting Channel view under the “Control” drop-down bar; or by clicking on the small arrow on the lower, right-hand side of the Intensity slider box; you can change the level of intensity control from wavelength to channel and visa versa.



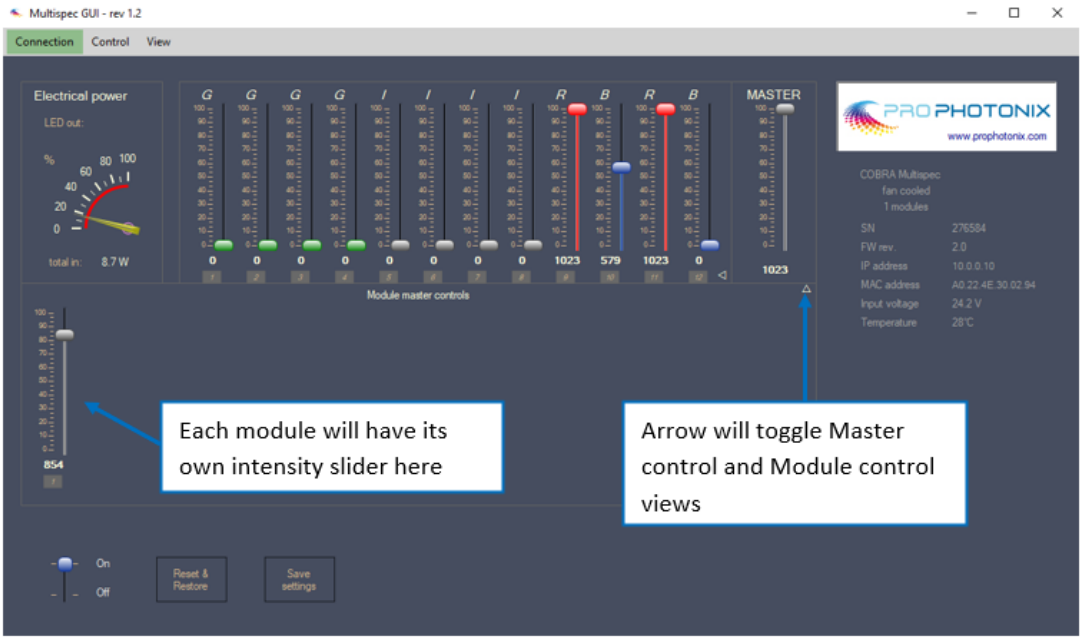

Each channel (there are a total of twelve on every COBRA module) can be adjusted in intensity by moving the corresponding slider from 0% to 100%. If fewer than 12 wavelengths are present in the product, then more than one channel may be assigned to a particular wavelength.

Adjusting the Master Slider at any time, will adjust each channel proportionately in order to maintain the same channel intensity ratio.

## 2.6.3 Adjusting Individual Module Intensity

By selecting the small arrow depicted on the lower, right-hand side of the Master Slider box, the view will switch to Module control.

This enables the user to adjust the overall intensity of each module, but the balance of wavelengths is assumed to be the same across all modules. It's important to note that the Module intensity slider overrides the Master Intensity Slider. For e.g.: Selecting 100% on the Master slider after adjusting Module 1 down to 50% intensity, will result in Module 1 remaining at 50% Intensity.



# 3 Strobe (AL Version Only)

## 3.1 Overview

Strobing control is used to pulse LED illuminators in a variety of machine vision applications and allows for current-overdriving of the LEDs, and hence, a much higher light intensity. It is possible to then synchronise the light pulse with the camera exposure to optimise signal strength. Shorter pulses lead to higher instantaneous light output. This approach is commonly used in areas such as high-speed motion detection.

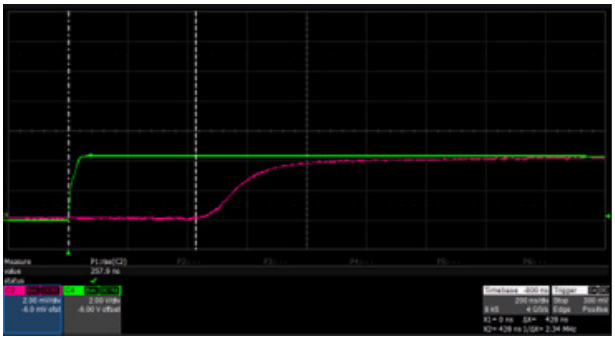
## 3 .2 Optical Output Gain

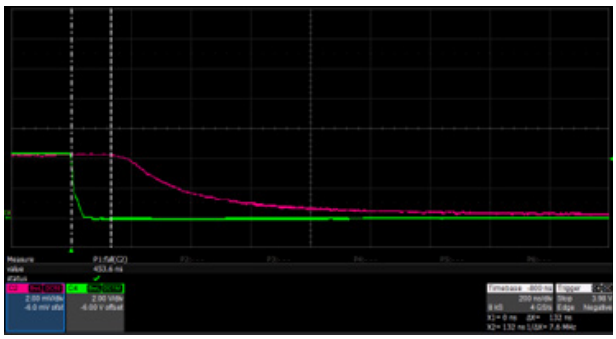
During the pulse, the operating current can be increased well above normal operations. This leads to an increase in optical output power. Samples of the ratios for a few specific wavelengths are listed in the table below. This highlights the variation between individual wavelengths and individual LED chip technologies. This needs to be taken into account when strobing multiple different wavelengths simultaneously.

| Wavelength (nm) | Convection Gain | Fan Gain |
|---|---|---|
| White | 5 | 2 |
| 365 | 6 | 2.5 |
| 470 | 4.7 | 2 |
| 630 | 6 | |
| MultiSpec/HyperSpec | Variable | Variable |

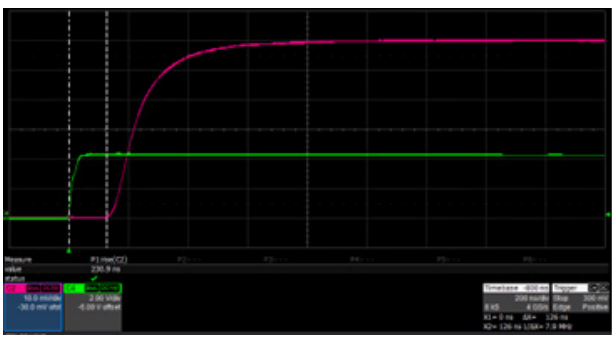
## 3.3 Turn-on, Turn-off Time

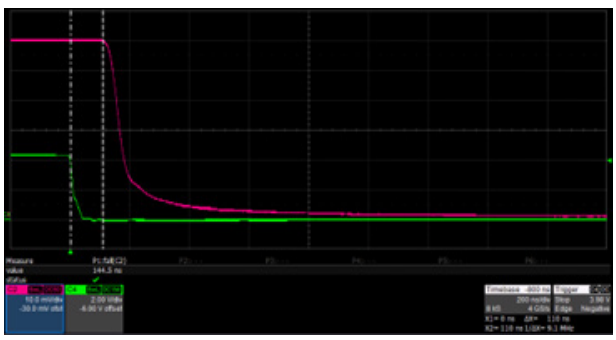
Turn-on and turn-off times (which include $T_{Delay}$ + $T_{Rise}$/Fall) are typically less than 1µ s in total. This is achieved using a two stage DCDC converter + High Speed Linear Amplifier. Typical waveforms for 100% and 5-10% optical output are shown as follows:





A typical waveform trace running at 10% is shown above
(Green = Strobe signal, Red = PhotoDiode Optical Output).





A typical waveform trace running at 100% optical output is shown above
(Green = Strobe signal, Red = PhotoDiode Optical Output).

## 3.4 COBRA MultiSpec and COBRA HyperSpec Strobe Operation

The COBRA MultiSpec and COBRA HyperSpec has the capability to operate up to 4 individual strobe groups. Users need to specify before a unit is manufactured, which of the following conditions are required:

- No strobe is required. All wavelengths are operated in CW mode.
- All wavelengths are strobed simultaneously.
- Three separate strobe groups (of up to 4 wavelengths in each group) are required. Each of these groups can be simultaneously or sequentially strobed with remaining strobe groups.
- Four separate strobe groups (of up to 3 wavelengths in each group) are required. Each of these groups can be simultaneously or sequentially strobed with remaining strobe groups.

If users wish to benefit from overdriving gains (as specified in Section 3.2 above), then the expected strobe conditions need to be specified prior to manufacture. In order to maintain the spectral balance defined by the end user, these gains will be limited by the wavelength that is working the hardest.

### 3.4.1 COBRA MultiSpec and COBRA HyperSpec Signal Cable

The signal cable supplied with the product (TAL- only) differs from the COBRA Slim and COBRA Max signal cable. The COBRA MultiSpec and COBRA HyperSpec signal cable is a 8-core (4 twisted pairs) cable allowing for four separate signals to operate up to 4 strobe groups independently or together.

**Corporate**
Tel: +1 603-893-8778
Fax: +1 603-898-8851

**LED Solutions**
Tel: +353-21-5001313
Fax: +353-21-4297749

**Laser Solutions**
Tel: +44-1279-717170
Fax: +44-1279-717171