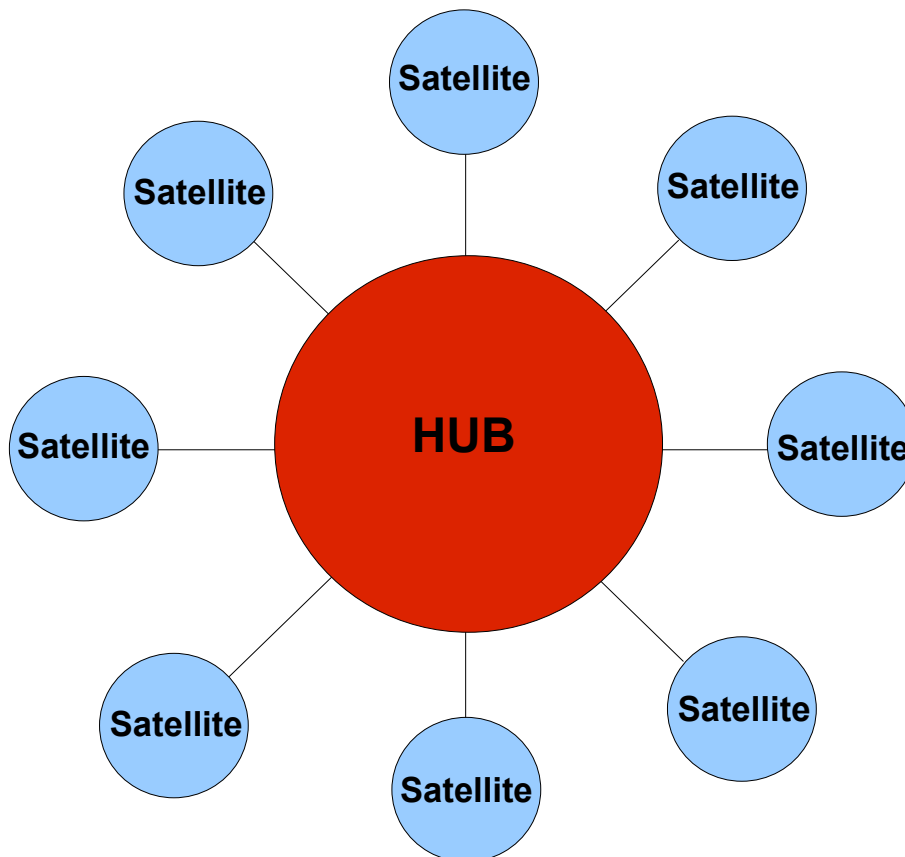


User Manual for Hub and Satellite



Feasa Enterprises Ltd,
Holland Road,
National Technology Park,
Castletroy,
Limerick,
Ireland.

www.feasa.ie

Email: sales@feasa.ie

Rev. 2.0

Date: August 2024

Feasa Legend LED Analyser



About this Manual

Feasa operates a policy of continuous development. Feasa reserves the right to make changes and improvements to any of the products described in this document without prior notice.

Feasa reserves the right to revise this document or withdraw it at any time without prior notice.

This manual is written for the Feasa Hub and Satellite LED Analyser. The model numbers are Hub, Feasa xL. (where x is the number of Fibers either 10 or 20)

The interface on the Hub units is USB or RS232-Serial and the Satellites are interfaced via a 6 pin connector.

© Copyright 2005-2024 by Feasa Enterprises Ltd. All rights reserved. No parts of this manual may be reproduced or retransmitted in any form or means, electronic or mechanical, including photocopying, recording, or any other storage and retrieval system without prior permission in writing from Feasa Enterprises Ltd. Every effort has been made to ensure that the information in this manual is accurate. Feasa Enterprises Limited is not responsible for printing or clerical errors.

Feasa Legend LED Analyser



Table of Contents

Introduction	5
IMPORTANT Information for Programmers	6
Colour and Saturation	7
Intensity	9
White LED's	11
Setting Tolerance Limits for Colour and Intensity	12
Product Description	13
Hub and Satellite Block Diagram	14
Mode of Operation	15
Hub Controller Module	16
External Input Port	17
External Input Logic Levels	18
Functional description	20
Limit group	20
Test	21
USB Port Control	31
Serial Port Control	31
LED Analyser Module (Satellite)	32
Physical Layout	32
Capture Commands	33
AUTO CAPTURE - Store LED Data	34
MANUAL CAPTURE - Store LED Data for a specific range	35
Get Data Commands	36
Under Range Condition	36
Over Range Condition	36
getRGBI### - Get RGB and Intensity for a LED	37
getRGBIall - Get RGB and Intensity for all LEDs	38
getHSI### - Get Hue, Saturation and Intensity	39
getHSIall - Get Hue, Saturation and Intensity	40
getWAVELENGTH### - Get the Dominant Wavelength	41
getWavelengthall - Get the Dominant Wavelength	42
getWAVELENGTHOFFSET@@ - Get the Wavelength Offset	43
getWI### - Get the Dominant Wavelength and Intensity	44
getWIall - Get the Dominant Wavelength	45
getxy### - Return the xy Chromaticity values	46
getxyall - Return the xy Chromaticity values	47
getxyi### - Return the xy Chromaticity & Intensity values	48
getxyiall - Return the xy Chromaticity values	49
getxoffset### - Return the x Chromaticity offset	50
getyoffset### - Return the y Chromaticity offset	51
getCCT - Get Correlated Color temperature	52
getCCTall - Get Correlated Color temperature	53
getCCTIall - Get Correlated Color temperature & Intensity	54

Feasa Legend LED Analyser



getuv### - Return the u'v' Chromaticity values	55
getuvall - Return the u'v' Chromaticity values	56
getIntensity### - Get the Intensity	57
getIntensityall - Get the Intensity	58
getIntGain####- Get the Intensity Gain Factor	59
getfactor - Get the exposure Factor	60
General Set Commands	61
SetIntGain - Set the Intensity GainFactor	62
SetFactor - Set the Exposure Factor	63
Setxoffset - Set the x Chromaticity Offset	64
setyoffset - Set the y Chromaticity Offset	65
setwavelengthoffset - Set the wavelength Offset	66
Hub Commands	67
GetStatus - Get the Status Information of the Hub	68
GetSerial - Get the Serial Number of the Hub	69
GetVersion - Get the Firmware Version	70
GetHW - Get the Hardware Version	71
Setbaud - Change the baud rate of the Serial Port	72
SetPrinterbaud- Set the Printer baud rate of the Serial Port	73
PrinterOn - Turn on the Printer function	74
PrinterOff - Turn off the Printer function	75
Automated Test Commands and Functions	76
Detect - detect LED Analysers	78
getPortinfo - list all connected LED Analysers	79
getPortDetails - list all connected Satellites with Serial No's	80
HubReset - Executes a Reset on the Hub	81
GetNFib - Get the total number of Fibers	82
setLimitGroup - set the active limits group	83
GetLimitGroup - get the active limits group	84
CaptureTest - Automatically Capture and Test LED's	85
CaptureMultiRange - Automatically Capture and Test LED's	86
Test - Automatically Test LED's	87
Reportlimit - Limit the number of reported failures	88
Reportfailures - Return failure data only	89
Testing Boards on a Panel	90
SetBoardEnable - Enable all Fibers on a Satellite Number	91
setBoardEnableAll - Enable all Fiber Channels	92
SetBoardDisable - Disable all Fibers on Satellite Number	93
SetBoardDisableAll - Disable all Fiber Channels	94
GetBoardEnableStatus### - Return the Status of a Board	95
GetTestResult - Return the Test Result of a Panel	96
Command Summary	97
Specifications	99
Warranty	100

Feasa Legend LED Analyser



Introduction

The **Feasa Legend system** is an instrument that tests the HSI (Hue, Saturation, Intensity), RGB (Red, Green, Blue Colour content of a single LED), xy Chromaticity, Dominant Wavelength and cct Colour temperature of Light Emitting Diodes (LEDs) in a test process. An individual Satellite can have up to 20* flexible Fiber-Optic Light Guides which are mounted individually over the LEDs to be tested.

The Satellite Analyser uses Optical Heads connected to flexible Optical fibers which are mounted individually over the LEDs to be tested.

Emitted Light from the LEDs is guided through the Optical fibers to the Satellite Analyser where the Colour and Intensity are tested. The test results can then read out using Feasa-supplied software programs or through User-generated software.

There are two component parts to the Legend LED Analyser:- A *Satellite LED Analyser module* which is mounted on the Test Fixture and a *Hub Module* which is used to control up to 8 Satellite LED Analysers.

The Feasa-supplied Programs communicate with the Hub module through a USB or Serial port on the Hub.

Colour, Saturation and Intensity values are output as ASCII strings through the Interfaces on the Hub Module.

All colours are derived from the three primary colors, Red, Green and Blue (RGB). The RGB values are used to identify the colour of LEDs.

Every LED tested by the LED Analyser will have a set of RGB values generated for analysis. The results for each LED to be tested can be read out through the USB and Serial Ports.

In addition, the system has the capability to 'learn' Test values from known good LED's and automatically assign Pass and Fail limits. The system can test the LED's automatically and generate Pass and Fail results.

This significantly reduces the amount of software generation required. One command is all that is required to test up to 160 LED's.

[Back to Index](#)

Feasa Legend LED Analyser



IMPORTANT Information for Programmers

The **set / put Commands** are used to adjust various settings in the LED Analyser such as Intensity, Exposure and offsets.

The **Set** commands are written to the on-board Flash.

These settings remain programmed in the Analyser even when the power is removed. The Led Analyser has a capacity limit of approximately 100,000 writes to the flash.

Use the Set command only to store relevant information on the Led Analyser. Please refrain from using Set commands in your high volume production programs as this constant writing to the Flash will eventually corrupt the Led Analyser. Use the Put command instead.

The **Put** commands are written to the on-board RAM. These settings will NOT remain programmed in the Analyser after the power is removed. Use the Put command as often as you need in your program. This will prolong the life of the Led Analyser particularly in high volume testing environment.

Commands are transmitted and received using ASCII characters and are **case-insensitive**. All commands must be terminated with a **<CR>** or **<LF>** character.

Feasa provides a set of programming examples in different languages (C++, C#, VB.NET, Python, Delphi/Free-pascal, Python and LabVIEW) and a DLL which ease the development process and reduces the development time and issues. These examples and DLL can be found in the CD/bundle provided along with the LED Analyser".

[Back to Index](#)

Colour and Saturation

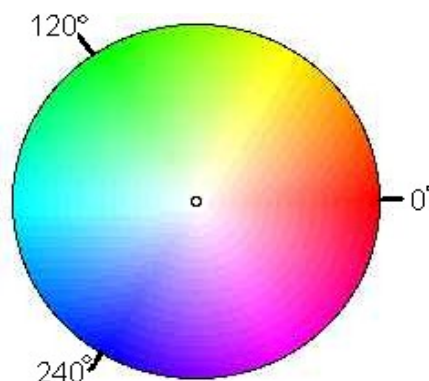


Figure 1: Hue (Color) Wheel

RED	=	0°
GREEN	=	120°
BLUE	=	240°

Colours can be represented by a 360° degree circular Colour wheel. The three Primary RGB values can also be represented as a single value called **Hue**.

Hue is a measured location on a Colour wheel and is expressed in degrees.

For example, Red will have a Hue value near 0°, Green will have a Hue value near 120° and Blue will have a value near 240°.

A pure Colour will be represented on the Colour wheel as a point near the outer edge. White will be represented by a point near the center of the wheel.

The degree of whiteness in a LED will affect its position on the wheel – the greater the amount of white the closer it will be to the center.

The Feasa Legend System output of **Saturation** is NOT part of the International System of Units and is not an absolute or traceable unit of colour measurement. It is part of the Hue Saturation and Relative intensity (HSI) system of describing colour. The degree of whiteness emitted by the LED is represented by the term **Saturation**. A Saturation value of 0% represents pure White. A Saturation value of 95 - 100% represents a pure Colour such as Red, Blue, Green, etc. It is a relative output much the same as the Relative Intensity output, and is used to compare similarly coloured LEDs under test. The Saturation values are calculated within the Feasa Legend system for each channel of the Satellite Analyser, relative to the LED under test.

There is a consistency in the Saturation readings between all Feasa Products, for LEDs of a similar colour, in a similarly constructed fixture.

Usually the user must determine the Hue and Saturation values by testing a number of LED's and recording the results.

[Back to Index](#)

Feasa Legend LED Analyser



The RGB and Hue values in *Figure 2* show how different Colour LEDs can be identified.

LED	R	G	B	HUE	SATURATION
Red	253	1	1	0	100%
Green	24	208	23	120	89%
Blue	2	13	240	238	99%
Yellow	76	171	8	95	96%
Orange	224	28	2	7	99%
White	71	72	112	See Page 13	21%

Figure 2.

The RGB or Hue values are used to identify different Colour LEDs.

Every LED tested by the LED Analyser will have a set of RGB values generated for analysis. These values are converted automatically to Hue and Saturation (whiteness) and can be read out for analysis.

[Back to Index](#)

Feasa Legend LED Analyser



Intensity

Intensity is a measure of the amount of light being emitted by the LED. The Analyser tests the Intensity of each LED and outputs this value to the Test System. The value is output as a number in the range 0-99,999. The Analyser is calibrated to a fixed standard and all measurements are relative. The Intensity output reading is NOT part of the *International System of Units* and is not an absolute or traceable unit of colour measurement. It is part of the Hue Saturation and Relative intensity (HSI) system of describing colour.

The Analyser can be set in either **Linear** mode or **Logarithmic** modes to measure Intensity of the LED. The units are shipped from the factory in Logarithmic mode. See [setlog](#) and [setlin](#) commands to determine how to set the mode of the analyser. Use the [getstatus](#) command to determine which intensity measure mode the Analyser is set to

Factors that influence Intensity Measurement:

- The position of the Fiber in relation to the LED.
- Offset from the Optical Centre of the LED.
- The Gap between the end of the Fiber Light Guide and LED to be measured.
- The condition of the Fiber end. It must be kept clean with a 90° Cleave.
- Is the LED Static or Flashing ?
- External Influences – Other LEDs in close proximity, Ambient Lighting.
- Is an Optical Head being used.

There are 6 capture ranges each with an intensity output range of 0 to 99,999. Feasa recommends that the UUT readings should be in the 55K to 85K range for the best stability.

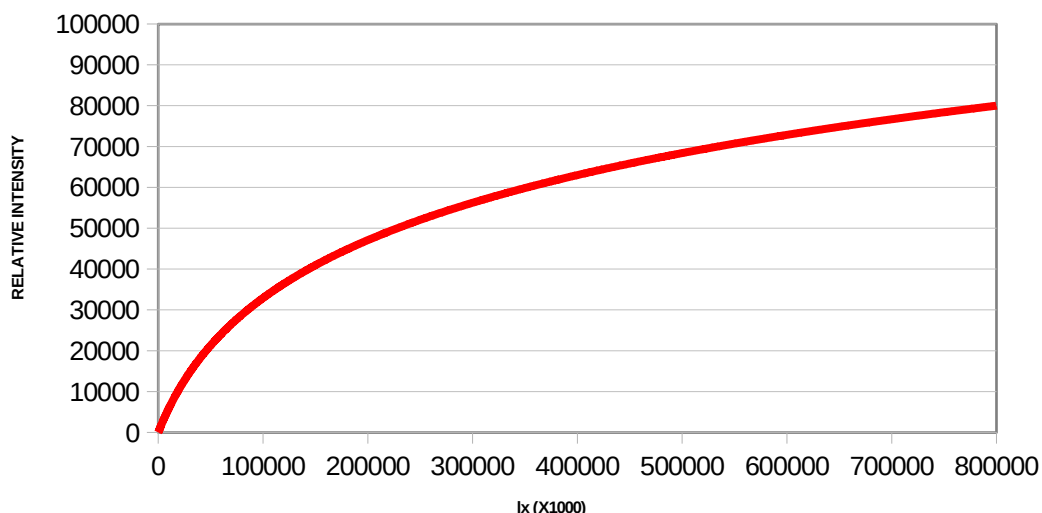


Figure 3a. Intensity for the LED Analyser in LOGARITHMIC Mode

[Back to Index](#)

Feasa Legend LED Analyser

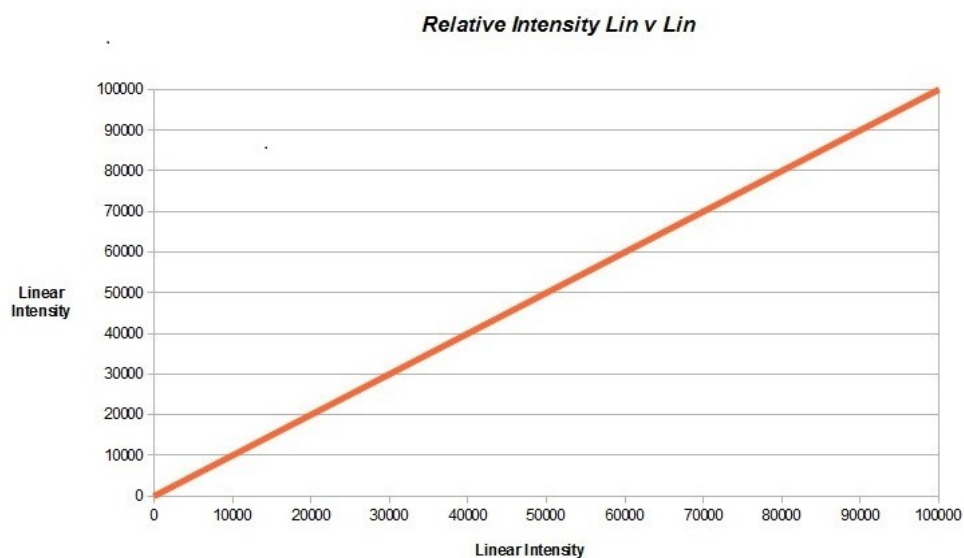


Figure 3b. Intensity for the LED Analyser in LINEAR Mode

[Back to Index](#)

Feasa Legend LED Analyser



White LED's

White LED's must be treated differently to coloured LED's when being tested. White is not a colour – it is a mix of all other colours. The three Primary colours Red, Green and Blue will be mixed in approximately equal proportions to display a White colour. The Saturation value should be used when testing White LED's. The Saturation is a value between 0% and 100%. A value of 0% indicates a pure White and a value of 100% indicates a pure Colour.

In reality, the Saturation value of white LED's vary significantly with values of 30% being typical. Remember, the Saturation value is an indication of how white the LED is. The correct values must be determined experimentally with the particular LED's to be tested.

Most LED manufacturers will specify their White LED's using Chromaticity co-ordinates xy . This is a two-dimensional Chart with x on the horizontal axis and y on the vertical axis. The range of x and y lies between 0 and 1.

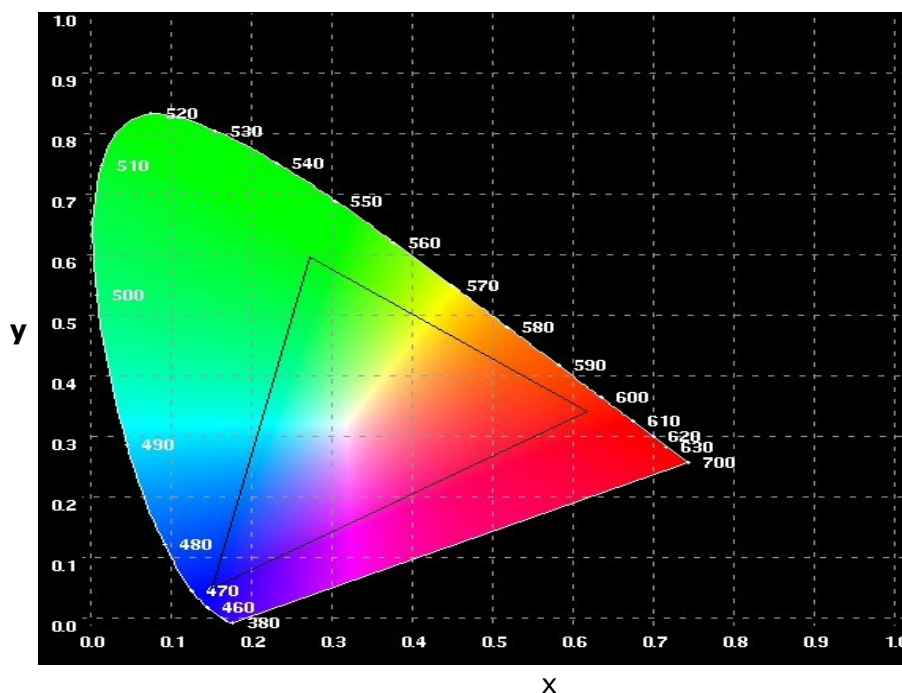


Figure 4. CIE 1931 Chromaticity Co-Ordinates

White LED's will have approximate co-ordinates of 0.33, 0.33. This may vary depending on the manufacturer of the LED where some LED's will have a Blue tint (cool white) and the other LED's will have a Red tint (Warm White).

[Back to Index](#)

Feasa Legend LED Analyser



Setting Tolerance Limits for Colour and Intensity

The test procedure requires the user to set the Pass/Fail limits for Colour and Intensity for each LED which then become the standard against which LEDs to be tested are compared. The Pass/Fail limits for Colour are chosen, in conjunction with manufacturer's specifications, from measurements taken from a sample number of typical LEDs. Because Hue is expressed as a single number it is more convenient to set the limits for the colour in terms of their Hue value.

Sample Hue Pass/Fail Limits

LED	Minimum	Maximum
Red	0	5
Green	110	130
Blue	220	250
Amber	5	10
Yellow	60	80
Orange	10	20

Figure 5.

White LED's should be identified in xy Chromaticity co-ordinates and should approximately be 0.3,0.3 in value. The *Saturation* value should be determined experimentally and typically should be less than 60% for a White LED.

The Pass/Fail limits for intensity are chosen from the average intensity values from a number of sample LEDs.

Sample Intensity Pass/Fail Limits

Relative Intensity Value	Upper Limit	Lower Limit	Comments
125	150	100	This represents a Very Dim LED, Recommend using Manual Capture Range C1 or C2
20000	25000	16000	This represents an Average LED, Recommend using Manual Capture Range, C2 or C3
70000	85000	55000	This represents a Bright LED, Auto capture use is OK but Manual Range C3, C4 or C5 would be recommended

Figure 6.

Note:- the Relative Intensity Value will depend on the Range selected.

Note:- Feasa recommends selecting a Capture Range to achieve Relative Intensity Reading of 55K to 85K for the best results.

[Back to Index](#)

Feasa Legend LED Analyser



Product Description

The product comprises two modules: a **Hub Controller** and at least one **LED Analyser Module** (Satellite).

The Hub Controller can be mounted internally or externally from the Test Fixture and can control up to 8 Satellite LED Analyser Modules at a time.

The LED Analyser Modules are mounted in the Test Fixture and are used to test the LED's on the PCB being tested.

Each LED Analyser Module can test up to 20 LED's simultaneously.

Each Hub Controller can control up to 8 LED Analyser Modules and can test up to 160 LED's. All tests are executed at the same time.

The Feasa Legend provides of an external port which allows to perform simple PASS/FAIL functional tests from external In-circuit Test machines, PLC controllers, micro-controllers or any other kind of test computer with I/O capabilities.

The behaviour of this port can be easily configured using a program called "Feasa ICTconfig", provided in the Legend CD.

Software Tools are provided to allow full integration into the Test System.

A block diagram of the system is shown in Figure 7.

[Back to Index](#)

Feasa Legend LED Analyser



Hub and Satellite Block Diagram

The function of the Hub is to control up to 8 Satellite LED Analyser Modules on the Test Fixture and to present a single Interface to the User Test System.

The Hub can be controlled by a PC through the USB or Serial Interfaces.

There is also Port that allows the Hub to be controlled using external inputs from Switches or Relays. This allows the Hub to be used in a standalone mode without a controlling PC.

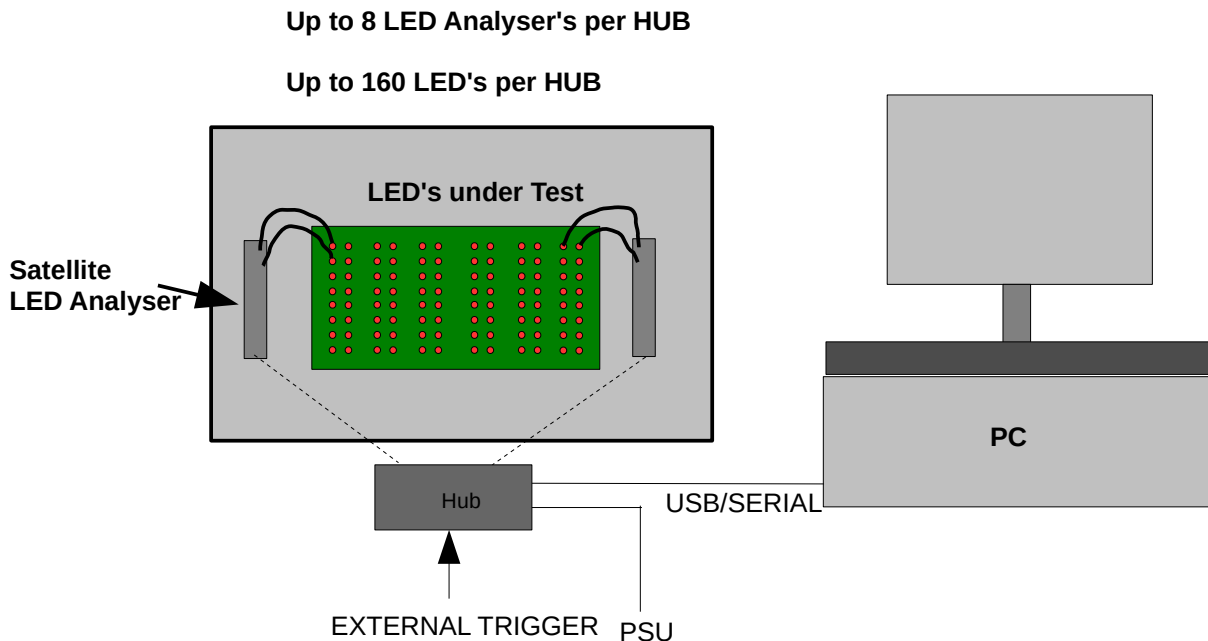


Figure 7.

[Back to Index](#)

Feasa Legend LED Analyser



Mode of Operation

All the Satellite Analysers connected to the Hub are grouped together into one large virtual Analyser. The Fiber channels of each Analyser are added together to give one large Analyser. For example, if a 20 Channel and a 10 Channel Analyser are connected to the Hub then these Analysers will behave as if there is one 30 Channel Analyser. If there are eight 20 Channel Analysers connected to the Hub then they will behave like one single 160 Channel Analyser.

All the Fiber Channels will be addressable in the range 001-160. The FeasaLimits software is used to manage the LED's and set the Pass and Fail limits for each LED.

[*Back to Index*](#)

Feasa Legend LED Analyser



Hub Controller Module

The layout of the Hub Controller module is shown in Figure 8.

Ports 1 to 8 are 6-pin connectors so that a flat 6-way ribbon cable can be used to connect to each Satellite LED Analyser module.

The USB or Serial ports can be connected to a host PC.

Note that only the Serial or USB can be used at any one time.

The *Power In* Jack Connector is used to supply +5V to each Satellite through the Ports 1 – 8.

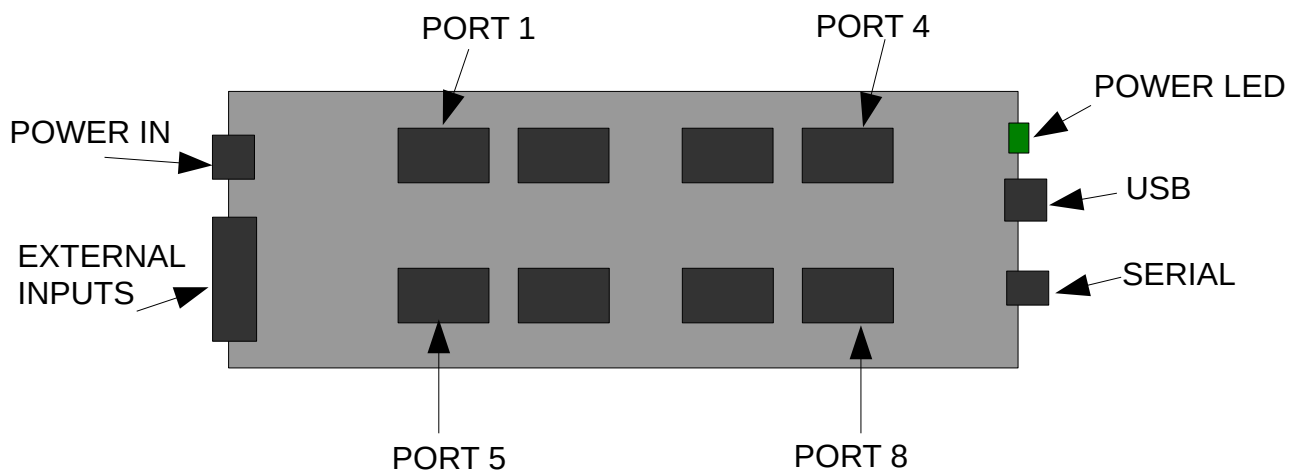


Figure 8.

The *External Inputs* is a 10-pin Connector which is used to trigger a Test Cycle and to report Pass or Fail status independently of a PC. See Figure 9 for a description of this Connector.

[Back to Index](#)

Feasa Legend LED Analyser



External Input Port

Pin	Name	Pin type	Function	Description
1	$\overline{\text{DATA0}}$	Input	Limit group (MSB)	4-bit binary code to select a Limit Group
2	$\overline{\text{DATA1}}$	Input	Limit group	
3	$\overline{\text{DATA2}}$	Input	Limit group	
4	$\overline{\text{DATA3}}$	Input	Limit group (LSB)	
5	TRIGGER	Input	Test	Trigger Input to start Automated Test cycle.
6	BUSY	Output	Test	Indicates Hub is busy with Test cycle.
7	PASS	Output	Test	Indicates Test cycle has Passed.
8	FAIL	Output	Test	Indicates Test cycle has Failed
9	$\overline{\text{RESET}}$	Input	Hardware Reset	Reset the Hub.
10	GND	Power	Power	Reference Ground

Note 1: all inputs are **active low**, so they follow an inverted logic.

Note 2: all inputs have **internal pull-up resistors**, so only need to be set to 0V for signaling a logic "1", since having the pin floating or connected to a positive voltage V_{IH} will indicate a logic "0".

Note 3: confusing input with outputs or vice versa can lead to damage the unit permanently.

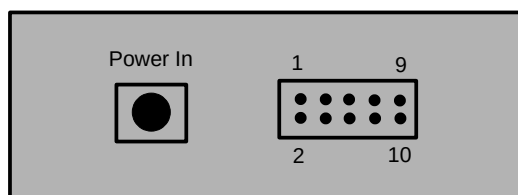


Figure 9.

[Back to Index](#)

Feasa Legend LED Analyser



External Input Logic Levels:

Minimum input voltage: 0.0V

Maximum input voltage: 5V

Minimum output voltage: 0.0V

Maximum output voltage: 3.3V

Maximum output current per pin: 10mA

$V_{IL} < 0.8V$

$V_{IH} \geq 2.0V$

$V_{OL} \leq 0.6V$

$V_{OH} \geq 2.2V$

Important:

- ensure that GND is properly connected and its voltage is actually 0V.
- ensure that no more than 5V are applied to any input pin
- ensure that output lines are never connected to GND. Mind the maximum current limits.

[Back to Index](#)

External Input Port

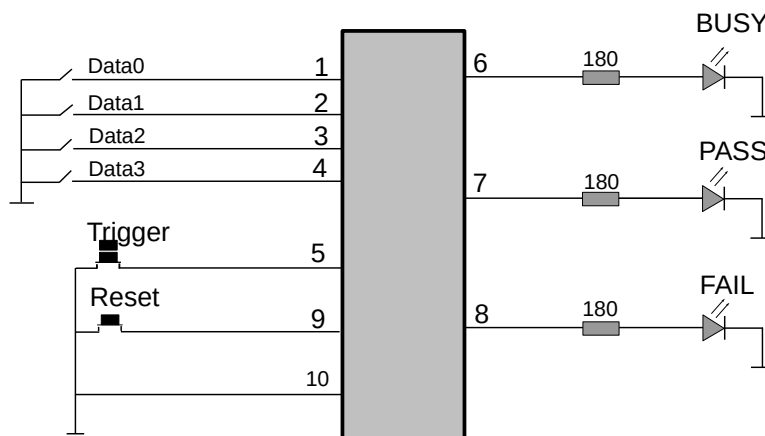


Figure 10.

Figure 10 illustrates how the External Input Port may be used. The Four Data lines are used to set the Limit Group to be used for the test. The Inputs must be set to a binary value in the range 0001 – 1111 (1 – 15). The binary value 0000 is not used.

The Pushbutton switches labelled *Trigger* and *Reset* momentarily pull the corresponding Input Low. The *Trigger* is used to initiate a Test Cycle. *Reset* will cause the Hub to be reset and enter an ideal state waiting for an input.

The two Outputs *Pass* and *Fail* indicate the result of the Test Cycle. If the Test passed then Pin 7 will be driven High and the *Pass LED* will light. If the Test Failed then Pin 8 will be driven High and the *Fail LED* will light. The *Busy* Pin 6 will be driven High during the Test Cycle to indicate that the Hub is executing a test cycle. Pin 6 will be driven Low at the end of the test.

Failure messages are output through either the USB or Serial Port. Only one output is available at a time.

[Back to Index](#)

Feasa Legend LED Analyser



Functional description

Limit group

A limit group is a set of values that define the low and high tolerable limits for a value under test. For example, when testing Intensity, we can set a lower limit of 65000 and an upper limit of 70000, so an Intensity value of 68000 will PASS, but a value of 75000 or 52344 will FAIL.

Each limit group can store a different set of lower and upper limits for all the LEDs under test, having the possibility to configure **up to 15 limit groups**, since only 4 bits are provided.

The **limit group** that will be used for the functional test **has to be signaled** using the 4 DATA lines provided in the external port, **before triggering a new measurement** or capture.

All inputs are active low, so in order to indicate the Limit group, a negative logic has to be used. Let's see some examples:

Desired Limit Group	DATA0	DATA1	DATA2	DATA3
00	NOT USED			
01	0	1	1	1
02	1	0	1	1
03	0	0	1	1
04	1	1	0	1
05	0	1	0	1
06	1	0	0	1
07	0	0	0	1
08	1	1	1	0
09	0	1	1	0
10	1	0	1	0
11	0	0	1	0
12	1	1	0	0
13	0	1	0	0
14	1	0	0	0
15	0	0	0	0

* in this table, logic is already inverted, so "0"=0V and "1"=floating/3.3

Feasa Legend LED Analyser



Test

The test lines allow to perform a functional measurement and retrieve the result.

The basic procedure to perform a measurement is to trigger a capture/measurement using the TRIGGER line (pin 5), then monitor the BUSY line (pin 6) and finally check for the result of the test in the PASS line (pin 7).

The behaviour of these lines can be configured through the "Feasa ICTconfig" program.

The Legend hub provides of a highly configurable Test, since the user can decide what kind of data will be given in the PASS/FAIL line:

- simple PASS/FAIL results through a HIGH/LOW status respectively
- PASS/FAIL results through HIGH/LOW + fiber/LED numbers failing, through an asynchronous communication
- PASS/FAIL results through HIGH/LOW + fiber/LED numbers failing + lower/upper limits, through an asynchronous communication
- PASS/FAIL results through HIGH/LOW + fiber/LED numbers failing, through a synchronous communication
- PASS/FAIL results through HIGH/LOW + fiber/LED numbers failing + lower/upper limits, through a synchronous communication

As could have been noticed error output data logging could be output Asynchronous or Synchronously. Following, a detailed description will be given for the different communication variants.

Capture + no output data

For triggering a capture, TRIGGER line (Capture in the diagram) has to be set to LOW for a T_{TRG} time, then set it back to HIGH and finally wait until BUSY line goes LOW.

Then, the result of the test will be given in the PASS line (pin 7) and FAIL line (pin 8); when the result of the test is **PASS**, the PASS line will be set to HIGH and the FAIL to LOW; when the result is **FAIL**, the PASS line will go to LOW while the FAIL will go to HIGH, being HIGH (3.3V) and LOW (0V).

Feasa Legend LED Analyser

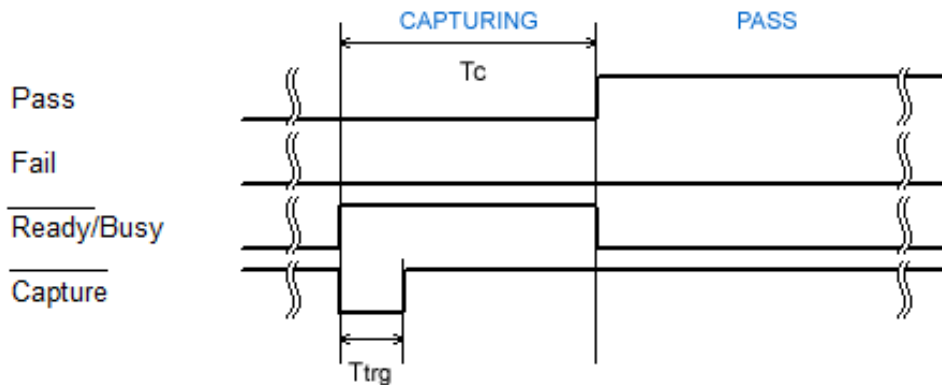


Diagram 1: Capture sequence with PASS result

Note: it is recommended to wait for at least 20us before testing the PASS line.

$T_{TRG} \geq 20\mu s$

$T_C \rightarrow$ Capture Time. Depends on the capture range selected.

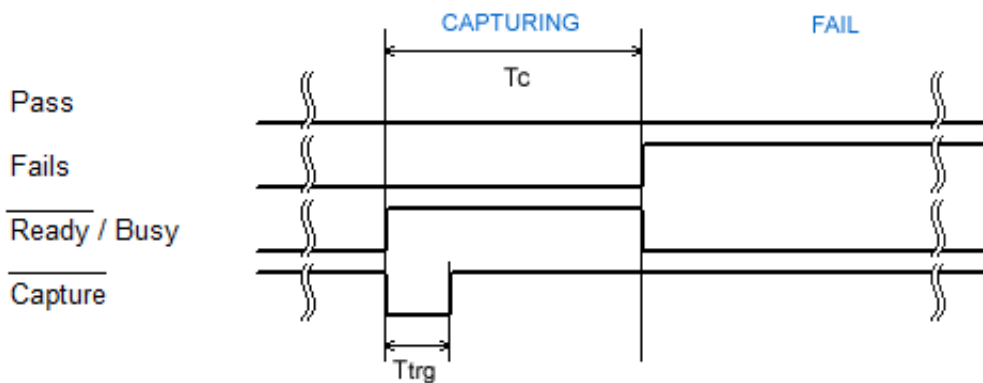


Diagram 2: Capture sequence with FAIL result

In case that **any error occur**, both PASS and FAIL lines will go to HIGH. The most common errors are:

- over-range
- under-range
- communication failed with one or more satellites
- satellites have not been detected
- Limit group has not been set
- test Limits have not been stored in the current group

Feasa Legend LED Analyser

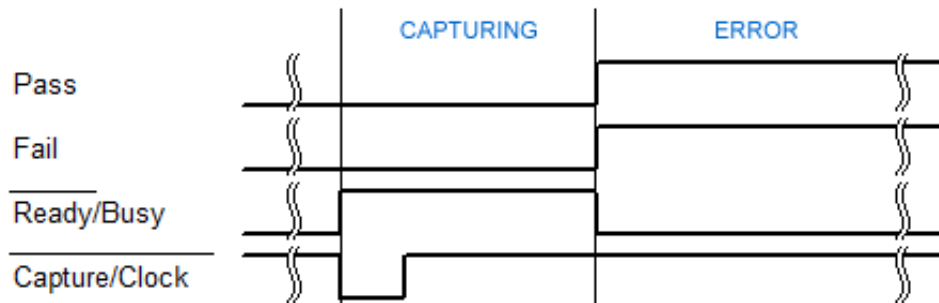


Diagram 3: Capture sequence with ERROR result

Set-up:

Despite the format for the output data logged can be easily configured through the ICTconfig program, this type of error report output can be configured using the following serial commands when **Asynchronous** communication mode is desired:

```
SETICTOUTMODEASYNC  
SETICTOUTDATANONE
```

For **Synchronous** communication mode, the commands needed are:

```
SETICTOUTMODESYNC  
SETICTOUTDATANONE
```

Capture + failing fibers output (Asynchronous)

As explained before, this extends the basic Capture output behaviour, adding the functionality to get data output of the fiber numbers/LEDs failing, through 8-bit values from 001 to 160.

The extra data provided is transmitted asynchronously through the FAIL line, after the fail condition has been signalled.

Each byte is transmitted using the typical *asynchronous* pattern, with a **start bit + 8 data bit + stop bit**, always starting with the Least significant bit LSB. **Positive logic** is used here.

For PASS and error conditions, the output is identical to what was previously described.

The communication output structure when a FAIL occurs is as follows:

1. Capture/trigger
2. FAIL status
3. Number of errors (8 bit)
4. Fiber number (8 bit) for error number 1
5. Fiber number (8 bit) for error number 2
6. ...
7. Fiber number (8 bit) for error number n

Feasa Legend LED Analyser

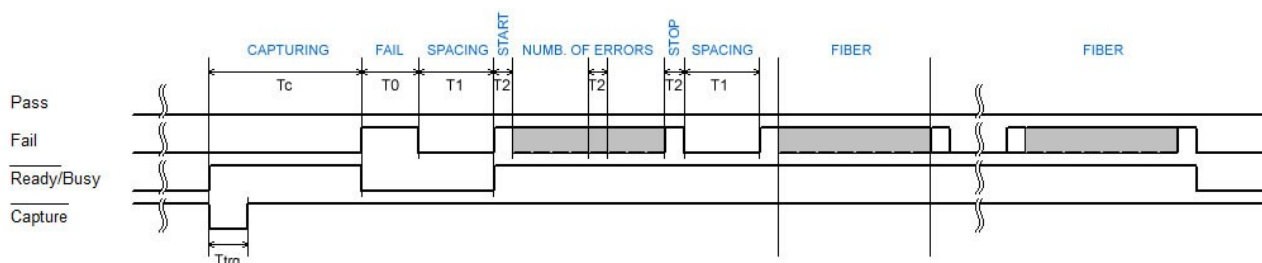


Diagram 4: Capture sequence with FAIL result + error output

$T_{trg} \geq 20\mu s$

$10\mu s \leq T0 \leq 999\mu s$ (user configurable)

$10\mu s \leq T1 \leq 999\mu s$ (user configurable)

$10\mu s \leq T2 \leq 999\mu s$ (user configurable)

For example, if fiber 3 and fiber 12 are failing, three bytes will be transmitted in total: the first for the number of errors (2), the second containing the number of the first fiber failing (3) and the third one containing the number of the seconds fiber failing (12).

START	LSB	B1	B2	B3	B4	B5	B6	MSB	STOP
1	0	1	0	0	0	0	0	0	1
1	1	1	0	0	0	0	0	0	1
1	0	0	1	1	0	0	0	0	1

Set-up:

Despite the format for the output data logged can be easily configured through the ICTconfig program, this type of error report output can be configured using the following serial commands:

SETICTOUTMODEASYNC

SETICTOUTDATAFIBER

SETICTTIMINGT0###, where ### is a value in microseconds from 10 to 999

SETICTTIMINGT1###, where ### is a value in microseconds from 10 to 999

SETICTTIMINGT2###, where ### is a value in microseconds from 10 to 999

Capture + failing fibers + limits output (Asynchronous)

This is another extension for the previous case, in which additional information is provided for what limits are failing and whether it is failing the upper or the lower one.

The extra data (2 extra bytes) provided is transmitted *asynchronously* through the FAIL line too, after the error number.

Each byte is transmitted using the typical asynchronous pattern, with a **start bit + 8 data bits (fibre number) + 8 data bit (data type failing) + 8 data bit (upper/lower flags) + stop bit**, always starting with the Least significant bit LSB.

For PASS and error conditions, the output is identical to what was previously described.

The communication output structure when a FAIL occurs is as follows:

Feasa Legend LED Analyser



1. Capture/trigger
2. FAIL status
3. Number of errors (8 bit)
4. Fiber number + flags for data type failing + flags for hi/lo status (3 bytes) for 1st error
5. Fiber number + flags for data type failing + flags for hi/lo status (3 bytes) for 2nd error
6. ...
7. Fiber number + flags for data type failing + flags for hi/lo status (3 bytes) for nth error

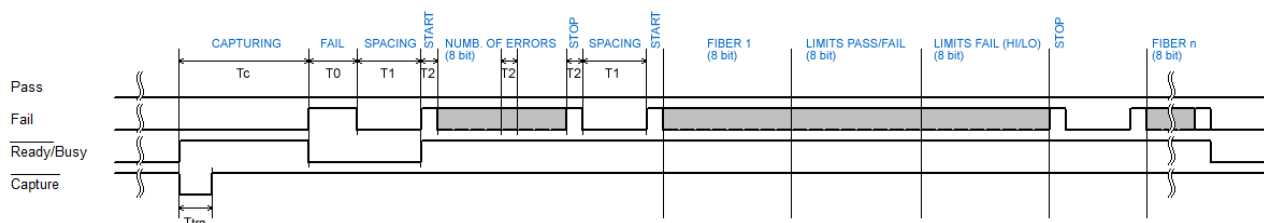


Diagram 5: Capture sequence with FAIL result + error + limits output

$T_{trg} \geq 20\mu s$

$10\mu s \leq T0 \leq 999\mu s$ (user configurable)

$10\mu s \leq T1 \leq 999\mu s$ (user configurable)

$10\mu s \leq T2 \leq 999\mu s$ (user configurable)

Here is the format for the second and third byte, containing information about what limit is failing and whether it is an upper (1) or a lower limit (0), respectively

W	d	C	Y	X	I	S	H
---	---	---	---	---	---	---	---

H: Hue
S: Saturation
I: Intensity
x: x (CIE 1931)
y: y (CIE 1931)
C: CCT
d: delta uv (CCT)
W: Wavelength

Table 1: Fail flags byte format

The flag bit for Hue is the LSB, while the flag bit for Wavelength is the MSB.

For example, if fiber 3 (Hue too low and x too high) and fiber 12 (Intensity too high) are failing, seven bytes will be transmitted in total: the first for the number of errors (2), the second containing the number of the first fiber failing (3), the third one containing the Fail flag with the bit for Hue set to high, the fourth one containing the Fail flag again with the bit for Hue set to low, etc.

Feasa Legend LED Analyser



START	LSB	B1	B2	B3	B4	B5	B6	MSB	STOP
1	0	1	0	0	0	0	0	0	1
1	1	1	0	0	0	0	0	0	1
1	1	0	0	1	0	0	0	0	1
1	0	0	0	1	0	0	0	0	1
1	0	0	1	1	0	0	0	0	1
1	0	0	1	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	1

Table 2: Data received: n. of errors (blue), data for 1st err (yellow); data for 2nd (red)

Set-up:

Despite the format for the output data logged can be easily configured through the ICTconfig program, this type of error report output can be configured using the following serial commands:

SETICTOUTMODEASYNC

SETICTOUTDATALIMITS

SETICTTIMINGT0###, where ### is a value in microseconds from 10 to 999

SETICTTIMINGT1###, where ### is a value in microseconds from 10 to 999

SETICTTIMINGT2###, where ### is a value in microseconds from 10 to 999

Capture + failing fibers output (Synchronous)

This extends the basic Capture output behaviour, adding the functionality to get data output of the fiber numbers/LEDs failing, through 8-bit values from 001 to 160.

The extra data provided is transmitted *synchronously* through the FAIL line, after the fail condition has been signalled. In this mode, TRIGGER line (pin 5) serves as a clock line too.

Each byte is transmitted synchronously, for which each bit is output on every falling edge in the TRIGGER line and should be read after the rising edge, while the clock signal is HIGH. Data is provided in bytes, this is **8 data bit**, always starting with the Least significant bit LSB.

Positive logic is used here.

For PASS and error conditions, the output is identical to what was previously described.

The communication output structure when a FAIL occurs is as follows:

1. Capture/trigger
2. FAIL status
3. Number of errors (8 bit)
4. Fiber number (8 bit) for error number 1
5. Fiber number (8 bit) for error number 2
6. ...
7. Fiber number (8 bit) for error number n

FEASA™



Feasa Legend LED Analyser



Capture + failing fibers + limits output (Synchronous)

This is another extension for the previous case, in which additional information is provided for what limits are failing and whether it is failing the upper or the lower one.

The extra data (2 extra bytes) provided is transmitted *synchronously* through the FAIL line too, after the error number, under the guidance of the clock signal (done through the TRIGGER line).

Each byte is transmitted synchronously, for which each bit is output on every falling edge in the TRIGGER line and should be read after the rising edge, while the clock signal is HIGH. Data is provided in bytes, this is **8 data bit**, always starting with the Least significant bit LSB.

Positive logic is used here.

For PASS and error conditions, the output is identical to what was previously described.

The communication output structure when a FAIL occurs is as follows:

1. Capture/trigger
2. FAIL status
3. Number of errors (8 bit)
4. Fiber number + flags for data type failing + flags for hi/lo status (3 bytes) for 1st error
5. Fiber number + flags for data type failing + flags for hi/lo status (3 bytes) for 2nd error
6. ...
7. Fiber number + flags for data type failing + flags for hi/lo status (3 bytes) for nth error

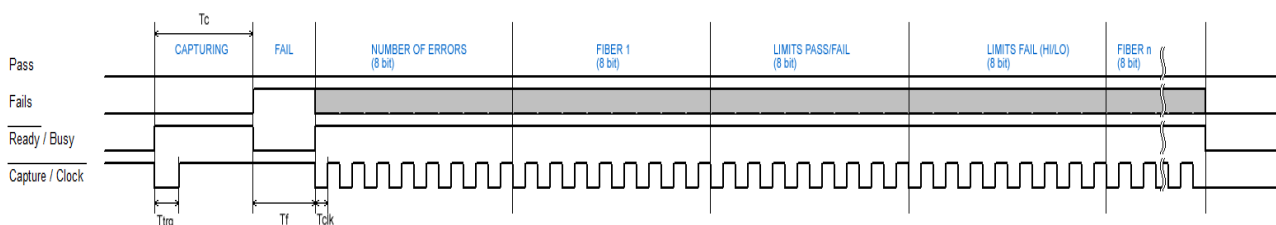


Diagram 7: Capture sequence with FAIL result + error + limits output

$$\begin{aligned} T_{\text{trg}} &\geq 20\mu\text{s} \\ 100\mu\text{s} &\leq T_{\text{F}} < T_{\text{err}} \\ T_{\text{clk}} &\geq 20\mu\text{s} \\ T_{\text{err}} &\geq 65\text{ms} \end{aligned}$$

Here is the format for the second and third byte, containing information about what limit is failing and whether it is an upper (1) or a lower limit (0), respectively

Feasa Legend LED Analyser



W	d	C	Y	X	I	S	H
---	---	---	---	---	---	---	---

H: Hue
 S: Saturation
 I: Intensity
 x: x (CIE 1931)
 y: y (CIE 1931)
 C: CCT
 d: delta uv (CCT)
 W: Wavelength

Table 3: Fail flags byte format

The flag bit for Hue is the LSB, while the flag bit for Wavelength is the MSB.

For example, if fiber 3 (Hue too low and x too high) and fiber 12 (Intensity too high) are failing, seven bytes will be transmitted in total: the first for the number of errors (2), the second containing the number of the first fiber failing (3), the third one containing the Fail flag with the bit for Hue set to high, the fourth one containing the Fail flag again with the bit for Hue set to low, etc.

LSB	B1	B2	B3	B4	B5	B6	MSB
0	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0

Table 4: Data received: n. of errors (blue), data for 1st err (yellow); data for 2nd (red)

Set-up:

Despite the format for the output data logged can be easily configured through the ICTconfig program, this type of error report output can be configured using the following serial commands:

SETICTOUTMODESYNC
 SETICTOUTDATALIMITS

Handling clocking errors (Synchronous)

In case that the clock signal has been inactive for a period equal or greater than a T_{err} time, then a timeout is triggered and communication will be aborted.

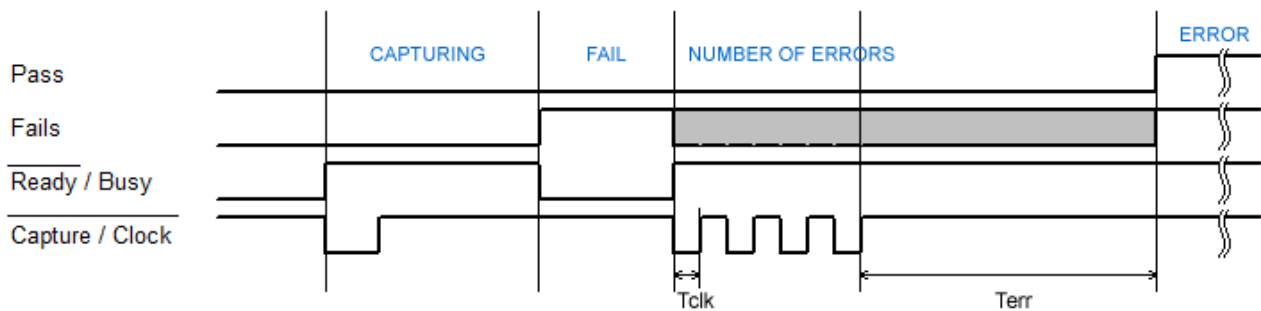


Diagram 8: clock interrupted before the data transmission ends

$$T_{clk} \geq 20\mu s$$

$$T_{err} \geq 65ms$$

This method could also be used to interrupt the communication.

[Back to Index](#)

Feasa Legend LED Analyser



USB Port Control

Connect the Hub to the PC using the supplied USB cable. USB Power is supplied through the USB Cable.

The installed Software Driver will configure the USB Port automatically.

The USB Port is configured as a Virtual Com Port and will be designated a name such as COM5, COM6, etc.

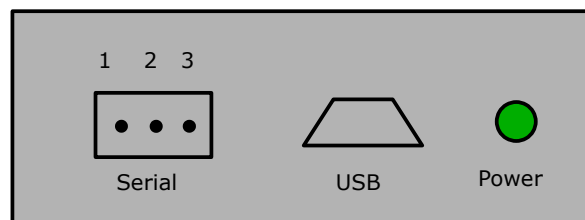


Figure 11A

Serial Port Control

For serial communications the Hub must be connected from the 3-pin Serial Connector to the PC or Controller using the supplied serial cable (LA-SER-02).

The Green LED should turn on to indicate the Analyser is ready for use.

The default serial communications settings are **57,600 Baud, 8 Data bits, 1 Stop bit and No Parity.**

The baud rate can be changed to any of the following:- 9600, 19200, 38400, 57600, 115200. See the [setbaud](#) command for more details.

Serial Connector (RS232C)

<i>Pin</i>	<i>Signal</i>	<i>Pin on 9-Pin D-type</i>
1	Tx from LED Analyser	2
2	Rx from LED Analyser	3
3	GND	5

Figure 11B.

[Back to Index](#)

LED Analyser Module (Satellite)

Physical Layout

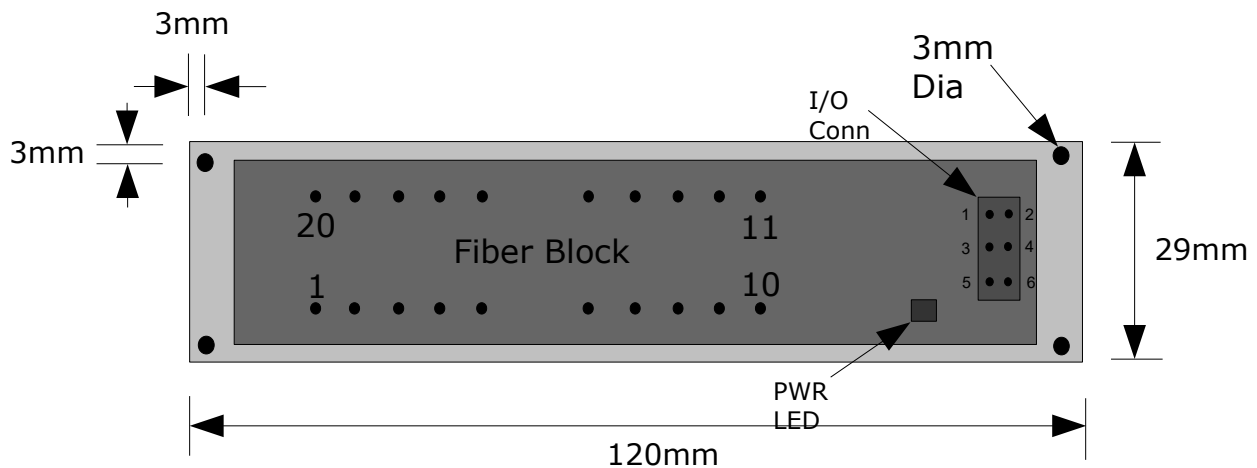


Figure 12.

Figure 12 shows the physical layout of the Analyser. The fibers are labeled 1-20. There are four 3mm diameter Mounting holes at the corners. The I/O Port is a 6-pin connector (See Figure 13 for the Pin connections). The I/O Connector must be connected to a Port on the Hub Module using a 6-way Ribbon Cable. The PCB has been designed so that it can easily be fitted in the top or bottom side of a Fixture.

I/O Connector

<i>Pin No</i>	<i>Signal</i>
1	Not Used
2	Rx
3	Tx
4	GND
5	Power (5V DC)
6	GND

Figure 13.

[Back to Index](#)

Feasa Legend LED Analyser



Capture Commands

The Capture commands are used to capture the parameters (colour, saturation, intensity, xy, uv, wavelength, cct) of the LED's to be tested and store the results in the memory of the Analyser.

These results can be read out later using the **GET DATA** commands.

Commands are transmitted and received using ASCII characters and are **NOT case-sensitive**. All commands must be terminated with a **<CR>** or **<LF>** character. All responses from the Led Analyser are also terminated with **<CR>** **<LF>**

The **Terminal Program** supplied on the CD is used to send/receive commands to/from the Analyser. This program is also available as a drop down box in the **User** program and the **Test** Software program.

The **Feasa Limits** Program is a graphical tool that can be used to send commands and receive results from the Analyser. It allows one LED to be tested at a time. This Program also allows a Terminal Window to be opened so that the User can type the commands directly and send them to the Analyser. The responses from the Analyser can be observed in the Window.

[Back to Index](#)

Feasa Legend LED Analyser



Capture Mode

AUTO CAPTURE - Store LED Data

Transmit	Receive
Capture c	OK OK

Description

This Auto Range Capture command instructs all LED Analysers connected to the Hub to capture and store the Colour and Intensity of all the LED's positioned under the fibers. The Analyser automatically determines the correct settings to capture the LED data. The data is stored until the power is removed or another **capture** command is issued. When completed the Hub will transmit the character **OK** on the receive line to the transmitting device (i.e. the PC).

This command uses a wide Intensity range to be able to test dim and bright LED's simultaneously. However, if the LED's to be tested are of similar Intensity then better results will be obtained by using the [Capture#](#) command described on the next page.

Example:

The PC transmits **capture** to the Hub and the Hub sends **OK** to the PC to acknowledge that the command is completed.

capture
OK

or

C
OK

[Back to Index](#)

Feasa Legend LED Analyser



Capture Mode

MANUAL CAPTURE - Store LED Data for a specific range

Transmit	Receive
capture# c#	OK OK

Where:

represents the ranges 1, 2, 3, 4, 5

The LED brightness level for each range is as follows:-

Range 1 = Low
Range 2 = Medium
Range 3 = High
Range 4 = Super
Range 5 = Ultra

Description

This command uses a pre-selected exposure time designated Range1, Range2 etc. For low light or dim LED's use Range 1 and for brighter LED's use higher ranges. The higher ranges lead to faster test times because the exposure time is shorter.

This command instructs all LED Analysers to read and store the Colour and Intensity of all the LED's positioned under the fibers using a fixed range.

The range setting must be specified. The data is stored until the power is removed or another **manual capture** command is issued. When completed the Hub will transmit the character **OK** on the receive line to the transmitting device (i.e. the PC).

Example:

The PC transmits **capture** to the Hub and the Hub sends **OK** to the PC to acknowledge that the command is completed.

capture2

OK

or

c2

OK

There are 5 manual capture ranges each with an intensity output range of 0 to 99,999. Feasa recommends that the UUT readings should be in the 55K to 85K range for the best stability.

[Back to Index](#)

Feasa Legend LED Analyser



Get Data Commands

The *get data* commands are used to read out the Colour, Saturation and Intensity data stored by the *capture* commands.

The data from the last *capture* command remains in memory until a new *capture* command is issued or the power is removed from the Analysers.

Commands are transmitted and received using ASCII characters and are **case-insensitive**. All commands must be terminated with a **<CR>** or **<LF>** character.

All references to fiber numbers must use 3 digits i.e. 046 for fiber number 046.

Under Range Condition

An under range condition will occur when insufficient light from the LED reaches the sensor for the range selected. This will be indicated by

999.99 999 00000 for **HSI**, **000 000 000 00000** for **RGBI** and **0.0000 0.0000** for xy and uv.

If this condition occurs select the next **lower** range and test again.

Over Range Condition

An over range condition will occur when too much light from the LED reaches the sensor for the range selected by the switch. This will be indicated by

999.99 999 99999 for **HSI**, **255 255 255 99999** for **RGBI** and **0.0000 0.0000** for xy and uv.

If this condition occurs select the next **higher** range and test again.

[Back to Index](#)

Feasa Legend LED Analyser



Get Data Mode

getRGBI### - Get RGB and Intensity for a LED

Transmit	Receive
getrgbi###	Rrr ggg bbb iiii

Where:

represents the Fiber Number in the range 001 – 160.

rrr , **ggg** and **bbb** are the **red**, **green** and **blue** components of the LED color. These values are normalized and are in the range 0 – 255.

iiii represents the intensity value of the LED under Fiber ##(#). This 5-digit number is in the range 00000 – 99999. 00000 represents no Intensity or under range(i.e. the LED is off) and 99999 will represent over range or the LED is too bright.

Description

This command instructs the Hub to return RGB and Intensity data for fiber ## (01-20) or **###** (001 – 160) in format **rrr ggg bbb iiii** where **rrr**, **ggg** and **bbb** are the **red**, **green** and **blue** components of the color. The **iiii** value indicates the intensity value.

This command should be preceded by a **capture** command to ensure valid LED data is stored in the memory of each LED Analyser. The data for each LED can be read out one-at-the-time, in any order.

Example:

Check a RED Led under Fiber No 5.

Capture followed by **getrgbi005** to the Hub to instruct it to capture the Led Data and send the stored RGBI data for the LED positioned under Fiber No 5. The Hub will return a string **rrr ggg bbb iiii** to the PC.

getrgbi005
244 008 003 06383

[Back to Index](#)

Feasa Legend LED Analyser



Get Data Mode

getRGBIall - Get RGB and Intensity for all LEDs

Transmit	Receive
getRGBIall	FFF rrr ggg bbb iiii

Where:

FFF represents the Fiber Number in the range 001-160

rrr , **ggg** and **bbb** are the **red**, **green** and **blue** components of the LED color. These values are normalized and are in the range 0 – 255.

iiii represents the intensity value of the LED under Fiber ###. This 5-digit number is in the range 00000 – 99999. 00000 represents no Intensity or under range(i.e. the LED is off) and 99999 will represent over range or the LED is too bright.

Description

This command instructs the Hub to return the RGB and Intensity data for all fibers in format **FFF rrr ggg bbb iiii** where **FFF** represents the Fiber Number in the range 001-160, **rrr**, **ggg** and **bbb** are the **red**, **green** and **blue** components of the color. The **iiii** value indicates the intensity value.

This command should be preceded by a **capture** command to ensure valid LED data is stored in the memory of each LED Analyser. The data for each LED will be returned starting with the lowest fiber number.

Example:

Check a RED Led under all Fibers.

Capture followed by **getrgbi** to the Hub to instruct it to capture the Led Data and send the stored RGBI data for all LEDs positioned under Fibers. The Hub will return a string **FFF rrr ggg bbb iiii** to the PC.

getrgbi

001 244 008 003 06383

002 243 009 003 06385

....

160 245 007 003 06387

[Back to Index](#)

Get Data Mode

getHSI### - Get Hue, Saturation and Intensity

Transmit	Receive
getHSI###	hhh.hh sss iiii

Where:

represents the Fiber Number in the range 001 – 160.

hhh.hh represents the Hue (color) and is a number in the range 0.00 – 360.00.

sss represents the Saturation (whiteness) and is a number in the range 0–100.

iiii represents the intensity value of the LED under fiber **###**. This 5-digit number is in the range 00000 – 99999. 00000 represents no Intensity or under range(i.e. the LED is off) and 99999 will represent over range or the LED is too bright.

Description

This command instructs the Hub to return Hue, Saturation and Intensity data for fiber **###** (001 – 160) in format **hhh.hh sss iiii** where **hhh.hh** represents the Hue(Color), **sss** represents the Saturation (whiteness) of the LED under Fiber **###**. The **iiii** value indicates the intensity value.

This command should be preceded by one of the **capture** commands to ensure valid LED data is stored in the memory of each LED Analyser. The data for each LED can be read out one-at-the-time, in any order.

Example:

The PC transmits **gethsi005** to the Hub to instruct it to send the stored Color and Intensity data for the LED positioned under Fiber No 5. The Hub will return a string **hhh.hh sss iiii** to the PC.

gethsi005
123.47 098 06383

[Back to Index](#)

Feasa Legend LED Analyser



Get Data Mode

getHSIall - Get Hue, Saturation and Intensity

Transmit	Receive
getHSIall	FFF hhh.hh sss iiii (multiple)

Where:

FFF represents the Fiber Number in the range 001 – 160.

hhh.hh represents the Hue (color) and is a number in the range 0.00 – 360.00.

sss represents the Saturation (whiteness) and is a number in the range 0–100.

iiii represents the intensity value of the LED under each fiber. This 5-digit number is in the range 00000 – 99999. 00000 represents no Intensity or under range(i.e. the LED is off) and 99999 will represent over range or the LED is too bright.

Description

This command instructs the Hub to return Hue, Saturation and Intensity data for all fibers in the format **hhh.hh sss iiii** where **hhh.hh** represents the Hue(Color), **sss** represents the Saturation(whiteness) and **iiii** indicates the Intensity value.

This command should be preceded by one of the **capture** commands to ensure valid LED data is stored in the memory of each LED Analyser.

Example:

The PC transmits **gethsiall** to the Hub to instruct it to send the stored Color and Intensity data for all Fibers.

gethsiall

001 123.47 098 66383

002 120.53 099 65124

160 120.53 099 65124

[Back to Index](#)

Get Data Mode

getWAVELENGTH### - Get the Dominant Wavelength

Transmit	Receive
getwavelength###	XXX

Where:

represents the Fiber Number and is a number in the range 001 – 160.
xxx represents the dominant wavelength of the LED in nanometers.

Description

This command is used to get the value of the Dominant Wavelength for the LED under the Fiber number ###.

This command should be preceded by one of the **capture** commands to ensure valid LED data is stored in the memory of each LED Analyser. The data for each LED can be read out one-at-the-time, in any order.

Example:

The PC transmits **getwavelength001** to the Hub to instruct it to send the stored dominant wavelength for the LED positioned under Fiber No 1. The Hub will return a string **xxx** to the PC.

getwavelength001
513

[Back to Index](#)

Get Data Mode

getWavelengthall - Get the Dominant Wavelength

Transmit	Receive
getwavelengthall	FFF www

Where:

FFF represents the Fiber Number in the range 001 – 160.

www represents the dominant wavelength of the LED in nanometers.

Description

This command is used to get the Dominant Wavelength and Intensity data for all Fibers.

This command should be preceded by one of the **capture** commands to ensure valid LED data is stored in the memory of each LED Analyser.

Example:

The PC transmits **getwavelengthall** to the Hub to instruct it to send the stored data all Fibers.

getwavelengthall

001 513

002 514

.

160 520

[Back to Index](#)

Get Data Mode

getWAVELENGTHOFFSET@@ - Get the Wavelength Offset

Transmit	Receive
getwavelengthoffset@@	+/-XX

Where:

@@ represents the Fiber Number and is a number in the range 1 – 20.

+/- xx represents the wavelength offset set of the LED in nanometers.

Description

This command is used to get the value of the Dominant Wavelength Offset set for Fiber number @@.

The range of value that can be programmed is +/-99, however setting a value of greater than +/-10 would probably indicate an error in your measurement setup

Example:

The PC transmits **getwavelengthoffset01** to the LED Analyser to instruct it to send the programmed dominant wavelength offset for the LED positioned under Fiber No 1. The LED Analyser will return a string **+/- xx** to the PC.

getwavelengthoffset01
+07

[Back to Index](#)

Get Data Mode

getWI### - Get the Dominant Wavelength and Intensity

Transmit	Receive
getwi###	www iiii

Where:

www represents the dominant wavelength of the LED in nanometers.

iiii represents the intensity value of the LED under each fiber. This 5-digit number is in the range 00000 – 99999. 00000 represents no Intensity or under range(i.e. the LED is off) and 99999 will represent over range or the LED is too bright.

Description

This command is used to get the Dominant Wavelength and Intensity data for all Fibers. This command should be preceded by one of the **capture** commands to ensure valid LED data is stored in the memory of each LED Analyser.

Example:

The PC transmits **getwi005** to the Hub to instruct it to send the stored data all Fibers.

getwi005

513 46122

[Back to Index](#)

Feasa Legend LED Analyser



Get Data Mode

getWIall - Get the Dominant Wavelength

Transmit	Receive
getwiall	Www iiii

Where:

www represents the dominant wavelength of the LED in nanometers.

iiii represents the intensity value of the LED under each fiber. This 5-digit number is in the range 00000 – 99999. 00000 represents no Intensity or under range(i.e. the LED is off) and 99999 will represent over range or the LED is too bright.

Description

This command is used to get the Dominant Wavelength and Intensity data for all Fibers.

This command should be preceded by one of the **capture** commands to ensure valid LED data is stored in the memory of each LED Analyser.

Example:

The PC transmits **getwiall** to the Hub to instruct it to send the stored data all Fibers.

getwiall

001 513 34567

002 514 23456

.

160 520 12345

[Back to Index](#)

Get Data Mode

getxy### - Return the xy Chromaticity values

Transmit	Receive
getxy###	0.xxxx 0.yyyy

Where:

represents the Fiber Number in the range 001 – 160.

0.xxxx represents the x Chromaticity value

0.yyyy represents the y Chromaticity value

Description

This command is used to return the xy Chromaticity value for the LED under the Fiber number ###. This command is used for testing White LED's.

This command should be preceded by a **capture** command to ensure valid LED data is stored in the memory of the LED Analysers. The data for each LED can only be read out one-at-the-time, in any order.

Example:

The PC transmits **getxy001** to the Hub to instruct it to send the stored xy Chromaticity data for the LED positioned under Fiber No 1. The Hub will return a string **0.xxxx 0.yyyy** to the PC.

Getxy001

0.3161 0.3436

[Back to Index](#)

Get Data Mode

getxyall - Return the xy Chromaticity values

Transmit	Receive
getxyall	FFF 0.xxxx 0.yyyy

Where:

FFF represents the Fiber Number in the range 001 – 160.

0.xxxx represents the x Chromaticity value

0.yyyy represents the y Chromaticity value

Description

This command is used to return the xy Chromaticity for all Fibers. This command is used for testing White LED's.

This command should be preceded by a **capture** command to ensure valid LED data is stored in the memory of the LED Analysers.

Example:

The PC transmits **getxyall** to the Hub to instruct it to send the stored xy Chromaticity and Intensity data for all fibers.

Getxyall

001 0.3141 0.3436

002 0.3152 0.3445

.

160 0.3300 0.3300

[Back to Index](#)

Feasa Legend LED Analyser



Get Data Mode

getxyi### - Return the xy Chromaticity & Intensity values

Transmit	Receive
getxyi###	0.xxxx 0.yyyy iiii

Where:

represents the Fiber Number in the range 001 – 160.

0.xxxx represents the x Chromaticity value

0.yyyy represents the y Chromaticity value

iiii represents the intensity value of the LED under each fiber. This 5-digit number is in the range 00000 – 99999. 00000 represents no Intensity or under range(i.e. the LED is off) and 99999 will represent over range or the LED is too bright.

Description

This command is used to return the xy Chromaticity and Intensity value for the LED under the Fiber number **###**. This command is used for testing White LED's.

This command should be preceded by a **capture** command to ensure valid LED data is stored in the memory of the LED Analysers. The data for each LED can only be read out one-at-the-time, in any order.

Example:

The PC transmits **getxyi001** to the Hub to instruct it to send the stored xy Chromaticity data and Intensity for the LED positioned under Fiber No 1. The Hub will return a string **0.xxxx 0.yyyy iiii** to the PC.

Getxyi001

0.3161 0.3436 12345

[Back to Index](#)

Feasa Legend LED Analyser



Get Data Mode

getxyiall - Return the xy Chromaticity values

Transmit	Receive
getxyiall	FFF 0.xxxx 0.yyyy iiii

Where:

FFF represents the Fiber Number in the range 001 – 160.

0.xxxx represents the x Chromaticity value

0.yyyy represents the y Chromaticity value

iiii represents the intensity value of the LED under each fiber. This 5-digit number is in the range 00000 – 99999. 00000 represents no Intensity or under range(i.e. the LED is off) and 99999 will represent over range or the LED is too bright.

Description

This command is used to return the xy Chromaticity and Intensity for all Fibers. This command is used for testing White LED's.

This command should be preceded by a **capture** command to ensure valid LED data is stored in the memory of the LED Analysers.

Example:

The PC transmits **getxyiall** to the Hub to instruct it to send the stored xy Chromaticity and Intensity data for all fibers.

Getxyiall

001 0.3141 0.3436 12345

002 0.3152 0.3445 23456

.

160 0.3300 0.3300 34567

[Back to Index](#)

Get Data Mode

getxoffset### - Return the x Chromaticity offset

Transmit	Receive
getxoffset###	±0.xxx

Where:

represents the Fiber Number in the range 001 – 160.

±0.xxx represents the x Chromaticity offset

Description

This command is used to return the x Chromaticity offset for the LED under the Fiber number **###**. The value of this offset must be set by the [setxoffset](#) command. The default value is 0.000.

Example:

The PC transmits **getxoffset001** to the Hub to instruct it to send the stored x Chromaticity offset for the LED positioned under Fiber No 1. The Hub will return a string **±0.xxx** to the PC.

getxoffset001
+0.155

[Back to Index](#)

Get Data Mode

getyoffset### - Return the y Chromaticity offset

Transmit	Receive
getyoffset###	±0.yyy

Where:

represents the Fiber Number in the range 001 – 160.

±0.yyy represents the y Chromaticity offset

Description

This command is used to return the y Chromaticity offset for the LED under the Fiber number **###**. The value of this offset must be set by the [setyoffset](#) command. The default value is 0.000.

Example:

The PC transmits **getyoffset001** to the Hub to instruct it to send the stored y Chromaticity offset for the LED positioned under Fiber No 1. The Hub will return a string **±0.yyy** to the PC.

getyoffset001
-0.025

[Back to Index](#)

Feasa Legend LED Analyser



Get Data Mode

getCCT - Get Correlated Color temperature

Transmit	Receive
getcct###	XXXXX ±0.yyyy

Where:

represents the Fiber Number and is a number in the range 001 – 160.

xxxxx represents the correlated color temperature of the LED.

± 0.yyyy represents the distance the LED is from the Plankian locus (Delta E)

Description

This command is used to get the Correlated Color Temperature (CCT) for each Fiber ###. The command must be preceded by a *Capture* **C** command to ensure valid LED data is stored in the memory of the Led Analysers.

The CCT value is only valid for white LED's and the maximum value allowed for Delta E is +/- 0.0500.

Example:

The PC transmits **getcct001** to the Hub to instruct it to send the stored CCT data for Fiber No 001.

getcct001
04621 +0.0340

[Back to Index](#)

Get Data Mode

getCCTall - Get Correlated Color temperature

Transmit	Receive
getcctall	FFF xxxx ±0.yyyy iiii

Where:

FFF represents the Fiber Number and is a number in the range 001 – 160.

xxxxx represents the correlated color temperature of the LED.

± 0.yyyy represents the distance the LED is from the Plankian locus (Delta E)

Description

This command is used to get the Correlated Color Temperature (CCT) and Intensity values for all fibers.

The command must be preceded by a *Capture C* command to ensure valid LED data is stored in the memory of the Led Analysers. The CCT value is only valid for white LED's and the maximum value allowed for Delta E is +/- 0.0500.

Example:

The PC transmits **getcctall** to the Hub to instruct it to send the stored data for all fibers.

getcctall

001 04621 +0.0340

002 04526 +0.0140

.

002 04526 +0.0140

[Back to Index](#)

Feasa Legend LED Analyser



Get Data Mode

getCCTIall - Get Correlated Color temperature & Intensity

Transmit	Receive
getcctiall	FFF xxxx $\pm$0.yyyy iiii

Where:

FFF represents the Fiber Number and is a number in the range 001 – 160.

xxxxx represents the correlated color temperature of the LED.

$\pm$ 0.yyyy represents the distance the LED is from the Plankian locus (Delta E)

iiii represents the intensity value of the LED under each fiber. This 5-digit number is in the range 00000 – 99999. 00000 represents no Intensity or under range (i.e. the LED is off) and 99999 will represent over range or the LED is too bright.

Description

This command is used to get the Correlated Color Temperature (CCT) and Intensity values for all fibers.

The command must be preceded by a *Capture* **C** command to ensure valid LED data is stored in the memory of the Led Analysers. The CCT value is only valid for white LED's and the maximum value allowed for Delta E is +/- 0.0500.

Example:

The PC transmits **getcctiall** to the Hub to instruct it to send the stored data for all fibers.

```
getcctiall
001 04621 +0.0340 67124
002 04526 +0.0140 67367
.
160 04526 +0.0140 67367
```

[Back to Index](#)

Get Data Mode

getuv### - Return the u'v' Chromaticity values

Transmit	Receive
getuv###	0.uuuu 0.vvvv

Where:

represents the Fiber Number in the range 001 – 160.

0.uuuu represents the u Chromaticity value

0.vvvv represents the v Chromaticity value

Description

This command is used to return the u'v' Chromaticity value for the LED under the Fiber number ###. This command is used for testing White LED's.

The u'v' values are derived from the xy Chromaticity co-ordinates including any xy offsets that may be applied.

This command should be preceded by a **capture** command to ensure valid LED data is stored in the memory of the LED Analysers. The data for each LED can only be read out one-at-the-time, in any order.

Example:

The PC transmits **getuv001** to the Hub to instruct it to send the stored u'v' Chromaticity data for the LED positioned under Fiber No 1. The Hub will return a string **0.uuuu 0.vvvv** to the PC.

getuv001
0.1809 0.4414

[Back to Index](#)

Feasa Legend LED Analyser



Get Data Mode

getuvall - Return the u'v' Chromaticity values

Transmit	Receive
getuvall	FFF 0.0000 0.0000

Where:

FFF represents all the Fiber Number in the range 001 – 160.

0.0000 represents the u Chromaticity value

0.0000 represents the v Chromaticity value

Description

This command is used to return the u'v' Chromaticity value for all fibers. This command is used for testing White LED's.

The u'v' values are derived from the xy Chromaticity co-ordinates including any xy offsets that may be applied.

This command should be preceded by a **capture** command to ensure valid LED data is stored in the memory of the LED Analysers.

Example:

The PC transmits **getuvall** to the Hub to instruct it to send the stored u'v' Chromaticity data for all fibers.

Getuvall

001 0.1809 0.4414

002 0.1821 0.4423

.

160 0.1821 0.4423

[Back to Index](#)

Get Data Mode

getIntensity### - Get the Intensity

Transmit	Receive
getIntensity###	IIIII

Where:

represents the Fiber Number in the range 001 – 160.

IIIII represents the Intensity value.

Description

This command is used to get the Intensity value for the LED under the Fiber number **###**. This command should be preceded by any **capture** command to ensure valid LED data is stored in the memory of the LED Analysers. The data for each LED can only be read out one-at-the-time, in any order.

Example:

The PC transmits **getintensity001** to the Hub to instruct it to send the stored Intensity data for the LED positioned under Fiber No 1. The Hub will return a string **IIIII** to the PC.

getintensity001

06734

[Back to Index](#)

Get Data Mode

getIntensityall - Get the Intensity

Transmit	Receive
getIntensityall	FFF IIIII

Where:

FFF represents the Fiber Number in the range 001 – 160.

IIIII represents the Intensity value.

Description

This command is used to get the Intensity value all fibers.

This command should be preceded by any **capture** command to ensure valid LED data is stored in the memory of the LED Analysers.

Example:

The PC transmits **getintensityall** to the Hub to instruct it to send the stored Intensity data for all fibers.

getintensityall

001 76734

002 64734

.

160 64000

[Back to Index](#)

General Commands

getIntGain###- Get the Intensity Gain Factor

Transmit	Receive
getIntGain###	xxx

Where:

represents the Fiber Number in the range 001 – 160.

xxx represents the Intensity gain value. Default 100.

Description

This command is used to get the Intensity gain value for each Fiber.

The default values set at the factory are 100 i.e. 100% of nominal. The values can be adjusted by the [SetIntGain](#) command.

Example:

The PC transmits **getintgain141** to the Hub to instruct it to send the stored intensity gain data for Fiber No 141.

getIntGain141
100

[Back to Index](#)

General Commands

getfactor - Get the exposure Factor

Transmit	Receive
Getfactor getfactor##	HH xx xx

Where:

HH represents the Hub port Number.

xx represents the exposure factor value 01 to 15. The default value is 01.

represents No Satellite connected to this port

Description

This command is used to get the exposure factor value for all Fibers.

The default value set at the factory are 01. The values can be adjusted by the [SetFactor](#) command.

Example No 1:

The PC transmits **getfactor** to the LED Analyser to instruct it to send the stored exposure factor for all Analysers.

The following example lists the response for 3 Satellite Analysers connected to a Hub. **##** indicates that no Satellite Analysers are connected to ports 04 – 08.

getfactor

```
01 01
02 01
03 01
04 ##
05 ##
06 ##
07 ##
08 ##
```

Example No 2:

The PC transmits **getfactor01** to the LED Analyser to instruct it to send the stored exposure factor for Satellite Analysers connect to Hub port No 1.

```
getfactor01
01
```

[Back to Index](#)

General Set Commands

The *Set* commands are used to adjust various settings in the LED Analyser such as Intensity and Exposure.

These settings remain programmed in the Analyser even when the power is removed.

Commands are transmitted and received using ASCII characters and are **case-insensitive**. All commands must be terminated with a **<CR>** or **<LF>** character.

The *Set* commands are written to the on-board Flash.

Use the Set command only to store relevant information on the Led Analyser. Please refrain from using Set commands in your high volume production programs as this constant writing to the Flash will eventually corrupt the Led Analyser.

[*Back to Index*](#)

Set Data Mode

SetIntGain - Set the Intensity GainFactor

Transmit	Receive
SetIntGain###xxx	OK

Where:

represents the Fiber Number in the range 001 – 160.

xxx represents a 3 digit gain factor, default 100.

Description

This command allows the user to adjust the Intensity Gain Factor for each Fiber. This is useful when it is required to balance all or some of the Fibers to give the same Intensity when Testing similar LED's. The Factory default setting is 100. The value for each Fiber can be adjusted from 050 – 200.

These values are stored permanently in memory and can only be changed by using the *setIntGain* command again.

The command [getIntGain](#) will display the current stored gain setting.

Note: A program called 'User Gains' is supplied that will execute this function automatically for all Fiber Channels.

Example:

Set the Intensity gain for Fiber 124 to 095.

setIntGain124095

OK

[Back to Index](#)

Set Data Mode

SetFactor - Set the Exposure Factor

Transmit	Receive
SetFactor##	OK

Where:

represents the factor value (01 to 15).

Description

This command allows the user to adjust the *Exposure Factor* for **all** Fibers. This is useful when it is required to test very dim LED's. The Factory default setting is 01. The value can be adjusted from 01 to 15. The exposure time will be increased when the factor is increased which will lead to longer test times. Try the low range(*capture1*) first before adjusting the Exposure Factor.

All fibers on all connected Analysers are set to the same value.

These values are stored permanently in memory and can only be changed by using the *setFactor* command again.

The current value can be read out using the [getfactor](#) command.

Example No 1:

Set the Exposure Factor for **all** Satellites connected to the Hub to Factor05.

```
setport00
OK
setfactor05
OK
```

Example No 1:

Set the Exposure Factor for the Satellite connected to the Hub Port No 2 to Exposure Factor 08.

```
setport02
OK
setfactor08
OK
```

[Back to Index](#)

Set Data Mode

Setxoffset - Set the x Chromaticity Offset

Transmit	Receive
Setxoffset###±0.xxx	OK

Where:

represents the Fiber Number in the range 001 – 160.

0.xxx represents the x Chromaticity offset value (± 0.000 to ± 0.300).

Description

This command is used to set an offset to the displayed x Chromaticity value. The limit of the offset is ± 0.300 which means values must be in the range ± 0.000 to ± 0.300 . This command is useful when the user wishes to set the x Chromaticity to be the same as that specified by the LED Manufacturer. The default value of the offset is 0.000. The offset is stored in non-volatile memory and will remain at the programmed setting until changed by a new *setxoffset* command.

Example:

The PC transmits **setxoffset121+0.050** to the Hub to instruct it to set the x offset on Fiber 121 to +0.050.

setxoffset121+0.050

OK

[Back to Index](#)

Set Data Mode

setyoffset - Set the y Chromaticity Offset

Transmit	Receive
setyoffset###±0.yyy	OK

Where:

represents the Fiber Number in the range 001 – 160.

0.yyy represents the y Chromaticity offset value (±0.000 to ±0.300).

Description

This command is used to set an offset to the displayed y Chromaticity value. The limit of the offset is ±0.300 which means values must be in the range ±0.000 to ±0.300. This command is useful when the user wishes to set the y Chromaticity to be the same as that specified by the LED Manufacturer. The default value of the offset is 0.000. The offset is stored in non-volatile memory and will remain at the programmed setting until changed by a new *setyoffset* command.

Example:

The PC transmits **setyoffset097-0.050** to the Hub to instruct it to set the y offset on Fiber 097 to -0.050.

setyoffset097-0.050
OK

[Back to Index](#)

Set Data Mode

setwavelengthoffset - Set the wavelength Offset

Transmit	Receive
setwavelengthoffset###±ww	OK

Where:

represents the Fiber Number in the range 001 – 160.

ww represents the wavelength offset value (±99).

Description

This set command is used to set an offset to the wavelength value. The limit of the offset is ±99. This command is useful when the user wishes to set the wavelength to be the same as that specified by the LED Manufacturer. The default value of the offset is 00. The offset is stored in non-volatile memory and will remain at the programmed setting until changed by a new *setwavelengthoffset* command.

Example:

The PC transmits **setwavelengthoffset097-50** to the Hub to instruct it to set the wavelength offset on Fiber 097 by -50.

setwavelengthoffset097-50
OK

[Back to Index](#)

Hub Commands

Hub Commands

There are a number of commands which identify the Hub and provide details about its configuration. These commands are described in the following section.

[*Back to Index*](#)

Hub Commands

GetStatus - Get the Status Information of the Hub

Transmit	Receive
getStatus	Multiple Lines

Description

This command will return the status of the Hub. This includes the Serial Number, Firmware Version and the number of Fibers available. Also included is the status of the Printer Output function and the Printer Baudrate.

Example:

getstatus

Hub Info

Serial Number : C012
Firmware Version : HB44
Total Fibers : 160
Printer Output : Enabled
Printer Baudrate : 57600

[Back to Index](#)

Hub Commands

GetSerial - Get the Serial Number of the Hub

Transmit	Receive
getSerial	xxxx

Where: xxxx is an alphanumeric value.

Description

This command will return the Serial Number of the Hub. This is a unique number and is useful if multiple Hubs are used in a System.

Example:

The PC transmits **getserial** to the Hub and it will return **xxxx** to the PC.

getserial
75A6

[Back to Index](#)

Hub Commands

GetVersion - Get the Firmware Version

Transmit	Receive
getVersion	xxxx

Where: xxxx is an alphanumeric value.

Description

This command will return the Version Number of the firmware in the Hub.

Example:

The PC transmits **getversion** to the Hub and it will return **xxxx** to the PC.

getversion
F002

[Back to Index](#)

Hub Commands

GetHW - Get the Hardware Version

Transmit	Receive
getHW	xxxxxx

Where: xxxxx is an alphanumeric value.

Description

This command will return the hardware version of the Hub.

Example:

The PC transmits **gethw** to the Hub and it will return **xxxxxx** to the PC.

gethw
Hub-8

[Back to Index](#)

Hub Commands

Setbaud - Change the baud rate of the Serial Port

Transmit	Receive
SetbaudX	OK

Where: **X** = 9600,19200,38400,57600,115200

Description

This command will change the baud rate of the Serial Port in the Analyser.

The default Port settings port are 57,600, 8 Data bits, 1 Stop bit and No Parity.

Example:

To change the baud rate to 9600 transmit the command **setbaud9600** to the Analyser .

Setbaud9600

OK

[Back to Index](#)

Hub Commands

SetPrinterbaud- Set the Printer baud rate of the Serial Port

Transmit	Receive
SetPrinterbaudX	OK

Where: **X** = 9600,19200,38400,57600,115200

Description

A Printer may be connected to the Serial Port to report failures when the external Input Trigger is used. The baud rate can be set for this function to allow for slow printers. This baud rate only applies when the external Trigger is used.

Also, the Printer function must be enabled with the [PrinterOn](#) command if it has been previously disabled with the [PrinterOff](#) command. The default setting is to have the printer function enabled.

This command will change the baud rate of the Serial Port in the Analyser.

The default Port settings port are 57,600, 8 Data bits, 1 Stop bit and No Parity.

Example:

To change the baud rate to 9600 transmit the command **setbaud9600** to the Analyser .

Setbaud9600
OK

[Back to Index](#)

Hub Commands

PrinterOn - Turn on the Printer function

Transmit	Receive
PrinterOn	OK

Description

This command will turn on the Printer function through the Serial Port. An external printer can be used to report failures. This function only works with the external Trigger function. The baud rate for the Printer can be set with the [SetPrinterBaud](#) command.

Note: This is an unbuffered output without handshaking.

Example:

printeron
OK

[Back to Index](#)

Hub Commands

PrinterOff - Turn off the Printer function

Transmit	Receive
PrinterOff	OK

Description

This command will turn off the Printer function through the Serial Port. An external printer can be used to report failures. This function only works with the external Trigger function. The baud rate for the Printer can be set with the *SetPrinterBaud* command.

Note: This is an unbuffered output without handshaking.

Example:

printeroff
OK

[Back to Index](#)

Automated Test Commands and Functions

The Automated Test feature has been designed to simplify the task of setting up the Led Analysers for testing. In addition, the test time and programming effort is reduced because the Analysers can be programmed with Pass/Fail limits. This allows the Analysers to decide if the LED's have passed the tests thus greatly reducing the amount of data to be transferred to a program.

A graphical program is supplied on the CD called *Feasalimits* which should be used to set up the Hub and LED Analysers for testing.

[Back to Index](#)

Feasa Legend LED Analyser



Feasa Limits v0.0.35, Project: C:\Program Files\FEASA Hub\FeasaLimits\projects\13 Dec 2011

File Boards Edit Limits Range Execute Statistics Select Help

Feasa™ Limits

www.feasa.com
sales@feasa.com
Tel: +353 61 330333

Current Limit Group: (no description); Limit Group 1

✓ Glob#	Board	Designator	Hue Min	Hue Max	Sat Min	Sat Max	Int Min	Int Max	x Min	x Max	y Min	y Max	VI Min	VI Max	CCT Min	CCT Max	d Min	d Max	Range
✓ 001	1	1	28	45	90	100	31817	54984	0	0	0	0	552	605	0	0	-0.075	-0.075	MEDIUM
✓ 002	1	2	28	98	90	100	11817	84984	0	0	0	0	552	605	0	0	-0.075	-0.075	MEDIUM
✓ 003	1	3	28	98	90	100	11817	84984	0	0	0	0	552	605	0	0	-0.075	-0.075	MEDIUM
✓ 004	1	4	28	98	90	100	11817	84984	0	0	0	0	552	605	0	0	-0.075	-0.075	MEDIUM
✓ 005	1	5	28	98	90	100	11817	84984	0	0	0	0	552	605	0	0	-0.075	-0.075	MEDIUM
✓ 006	1	6	28	98	90	100	11817	84984	0	0	0	0	552	605	0	0	-0.075	-0.075	MEDIUM
✓ 007	1	7	28	98	90	100	11817	84984	0	0	0	0	552	605	0	0	-0.075	-0.075	MEDIUM
✓ 008	1	8	28	98	90	100	11817	84984	0	0	0	0	552	605	0	0	-0.075	-0.075	MEDIUM
✓ 009	1	9	28	98	90	100	11817	84984	0	0	0	0	552	605	0	0	-0.075	-0.075	MEDIUM
✓ 010	1	10	28	98	90	100	11817	84984	0	0	0	0	552	605	0	0	-0.075	-0.075	MEDIUM
✓ 011	1	11	28	98	90	100	11817	84984	0	0	0	0	552	605	0	0	-0.075	-0.075	MEDIUM
✓ 012	1	12	28	98	90	100	11817	84984	0	0	0	0	552	605	0	0	-0.075	-0.075	MEDIUM
✓ 013	1	13	28	98	90	100	11817	84984	0	0	0	0	552	605	0	0	-0.075	-0.075	MEDIUM
✓ 014	1	14	28	98	90	100	11817	84984	0	0	0	0	552	605	0	0	-0.075	-0.075	MEDIUM
✓ 015	1	15	28	98	90	100	11817	84984	0	0	0	0	552	605	0	0	-0.075	-0.075	MEDIUM
✓ 016	1	16	28	98	90	100	11817	84984	0	0	0	0	552	605	0	0	-0.075	-0.075	MEDIUM
✓ 017	1	17	28	98	90	100	11817	84984	0	0	0	0	552	605	0	0	-0.075	-0.075	MEDIUM
✓ 018	1	18	28	98	90	100	11817	84984	0	0	0	0	552	605	0	0	-0.075	-0.075	MEDIUM
✓ 019	1	19	28	98	90	100	11817	84984	0	0	0	0	552	605	0	0	-0.075	-0.075	MEDIUM
✓ 020	1	20	28	98	90	100	11817	84984	0	0	0	0	552	605	0	0	-0.075	-0.075	MEDIUM
✓ 021	1	1	0	1	00	100	17770	64640	0	0	0	0	510	600	0	0	-0.075	-0.075	HIGH

Figure 14.

Figure 14 shows one of the setup screens from Feasalimits. Refer the the *Feasalimits User Guide* for more information on how to use this program.

Automated Test Commands

Detect - detect LED Analysers

Transmit	Receive
detect	OK

Description

This command will instruct the Hub to detect all the LED Analyser Modules connected to the Hub. A list of these LED Analysers can be output using the [getportinfo](#) command.

Example:

To detect all connected LED Analysers .

detect
OK

[Back to Index](#)

Automated Test Commands

getPortinfo - list all connected LED Analysers

Transmit	Receive
getportinfo	See example below

Description

This command will instruct the Hub to list all the LED Analyser Modules connected to the Hub. The first column lists the Hub Port Number and the second column lists the number of fibers available. If no LED Analyser is detected on a Port then the number of fibers will be listed as 00.

Example:

To list all connected LED Analysers .

getportinfo

```
01    20
02    10
03    10
04    00
05    00
06    00
07    00
08    20
```

[Back to Index](#)

Automated Test Commands

getPortDetails - list all connected Satellites with Serial No's

Transmit	Receive
getportdetails	See example below

Description

This command will instruct the Hub to list all the LED Analyser Modules connected to the Hub including the Serial Number of each Analyser. The first column lists the Hub Port Number and the second column lists the number of fibers available and the third column lists the Serial Number. If no LED Analyser is detected on a Port then the number of fibers will be listed as 00.

Example:

To list all connected LED Analysers .

getportdetails

01 20 C126	
02 10 C128	
03 10 C134	
04 00 ????	!not connected
05 00 ????	!not connected
06 00 ????	!not connected
07 00 ????	!not connected
08 20 C141	

[Back to Index](#)

Automated Test Commands

HubReset - Executes a Reset on the Hub

Transmit	Receive
hubreset	OK

Description

This command will instruct the Hub to perform a reset. The Detect Command must be executed after the reset to establish connections to all connected LED Analysers.

Example:

hubreset
OK

detect
OK

[Back to Index](#)

Automated Test Commands

GetNFib - Get the total number of Fibers

Transmit	Receive
GetNFib	###

Where:

represents the total number of fibers connected to the Hub

Description

This command instructs the Hub to return the total number of fiber channels connected to the Hub. The maximum number of fiber channels is 160.

This command is used to verify that all fiber channels are available before a test begins.

Example:

getnfib	Get the number of available fibers
030	here are 030 fibers connected to the Hub

[Back to Index](#)

Automated Test Commands

setLimitGroup - set the active limits group

Transmit	Receive
setlimitgroup##	OK

Where:

represents the limit group number (01 - 31)

Description

A limitgroup is a set of Pass and Fail values for each LED to be tested. There can be up to 31 limitgroups stored in the LED Analysers. This allows the testing of up to 31 different versions of the same board or 31 different board types. If additional limit groups are required they can be downloaded into the LED Analysers.

The values of each limitgroup are set by the *FeasaLimits* program during the debug phase. This command allows the controlling software to select a limitgroup to test a particular version of a board.

Example:

setlimitgroup01
OK

[Back to Index](#)

Automated Test Commands

GetLimitGroup - get the active limits group

Transmit	Receive
getlimitgroup	##

Where:

represents the limit group number (01 - 31)

Description

This command returns the current active limitgroup.

Example:

getlimitgroup
01

[Back to Index](#)

Automated Test Commands

CaptureTest - Automatically Capture and Test LED's

Transmit	Receive
CaptureTest	Pass/Fail String

Description

This command combines a *capture* and *test* in one command. The capture range to be used must be previously set by the *Feasalimits* Program.

The Hub will initiate a Capture followed by a Test Sequence that will test all *enabled* LED's using the Analysers connected to the Hub. The Pass/Fail limits for each LED must be previously uploaded to the Analyser(s) using the *FeasaLimits Program*. If all LED's pass the test then the string *Pass* will be returned. If there are failures then a list of failures will be returned.

Example:

capturetest
[Pass]

capturetest
list of failures.
[END FAIL]

[Back to Index](#)

Automated Test Commands

CaptureMultiRange - Automatically Capture and Test LED's

Transmit	Receive
CaptureMultiRange	OK

Description

This command instructs the LED Analyser to capture and store the Colour and Intensity of all the LED's positioned under the fibers using a pre-programmed *multiple range* feature. This allows LED's of different Intensities to be tested simultaneously using different ranges for individual LED's. The capture ranges to be used must be previously set by the *FeasaLimits* Program.

When completed the Hub will transmit the character **OK** to indicate completion of the command.

Example:

capturemultirange
OK

[Back to Index](#)

Automated Test Commands

Test - Automatically Test LED's

Transmit	Receive
Test	Pass/Fail String

Description

This command will instruct the Hub to initiate a Test Sequence that will test all *enabled* LED's using the Analysers connected to the Hub. The Pass/Fail limits for each LED must be previously uploaded to the Analyser(s) using the *FeasaLimits* Program. If all LED's pass the test then the string *Pass* will be returned. If there are failures then a list of failures will be returned.

To prevent long strings of failures being returned there are a number of commands to control the amount of data. These commands begin with the keyword *report* and are described in the next sections.

This command must be preceeded by one of the *capture* commands.

Example:

Test
[Pass]

Test
list of failures.
[END FAIL]

[Back to Index](#)

Automated Test Commands

Reportlimit - Limit the number of reported failures

Transmit	Receive
Reportlimit##	OK

Where:

represents the a number in the range 01 – 99.

Description

This command will instruct the Hub to limit the number of failed LED's (Fibers). The number of failed LED's can be set from 1 – 99. The default setting is to report all failures.

This command should be issued before the *CaptureTest* Command.

Example:

reportlimit10
OK

[Back to Index](#)

Automated Test Commands

Reportfailures - Return failure data only

Transmit	Receive
Reportfailures	OK

Description

This command will instruct the Hub to return *failure data only* resulting from the *Test* Command. This is the default setting for the system.

This command should be issued before the *Test* Command.

Example:

reportfailures
OK

[Back to Index](#)

Automated Test Commands

Testing Boards on a Panel

The LED Analyser can be used to test Panelized Boards and contains built-in commands to support Panel operations.

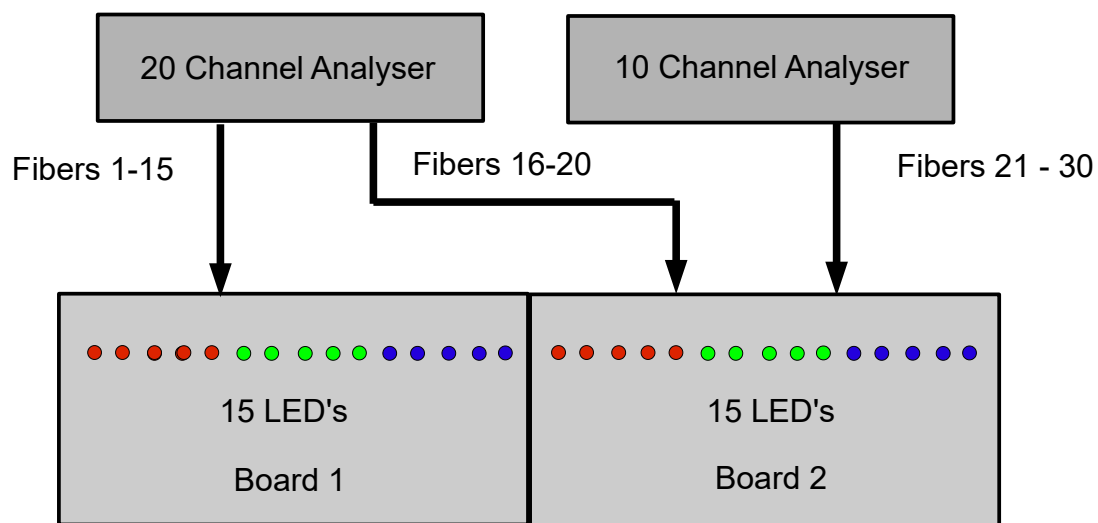


Figure 15

Figure 15 shows a simple two-board Panel with each board containing 15 LED's. Two LED Analyser's have been selected to test all 30 LED's – a 20 Channel and a 10 Channel.

On the 20 Channel Analyser 15 Fiber Channels have been assigned to *Board 1* and the remaining 5 Channels have been assigned to 5 LED's on *Board 2*. All the 10 Channels of the the 10 Channel Analyser have been assigned to *Board 2*. The *FeasaLimits* program should be used to configure LED Analyser boards on a panel.

The following section describes commands used specifically for Boards on a Panel.

[Back to Index](#)

Automated Test Commands

SetBoardEnable - Enable all Fibers on a Satellite Number

Transmit	Receive
SetBoardEnable###	OK

Where:

represents the Board Number.

Description

This command instructs the Hub to enable all the Fiber Channels assigned to a specific board on a Panel for Test. The default setting is that all Fiber channels are enabled.

The range of values allowed for the Board Number is 001 – 160.

Example:

setboardenable008 Enable all the fibers on board number 008
OK

[Back to Index](#)

Automated Test Commands

setBoardEnableAll - Enable all Fiber Channels

Transmit	Receive
SetBoardEnableAll	OK

Description

This command instructs the Hub to enable all the Fiber Channels connected to all available Satellites for a test.

Example:

setboardenableall !Enable all the fibers for test
OK

[Back to Index](#)

Automated Test Commands

SetBoardDisable - Disable all Fibers on Satellite Number

Transmit	Receive
SetBoardDisable###	OK

Where:

represents the Board Number.

Description

This command instructs the Hub to disable all the Fiber Channels assigned to a specific board on a Panel for Test. The Fibers will remain disabled until enabled by the [setboardenable](#) command.

The range of values allowed for the Board Number is 001 – 160.

Example:

setboarddisable008 disable all the fibers on board number 008
OK

[Back to Index](#)

Automated Test Commands

SetBoardDisableAll - Disable all Fiber Channels

Transmit	Receive
SetBoardDisableAll	OK

Description

This command instructs the Hub to disable all the Fiber Channels connected to Satellites. The Fibers will remain disabled until enabled by the [setboardenableall](#) command.

Example:

setboarddisableall !disable all the fibers on all available satellites
OK

[Back to Index](#)

Automated Test Commands

GetBoardEnableStatus### - Return the Status of a Board

Transmit	Receive
GetBoardEnableStatus###	ENABLE/DISABLE

Where:

represents the Board Number in the Panel.

Description

This command instructs the Hub to return the Status of a Board in the Panel. The Board status can be either ENABLE or DISABLE. If the status is ENABLE this means that the board is enabled for testing LED's. If the status is DISABLE this means that all the LED's on this board will not be tested.

The range of values allowed for the Board Number is 001 – 160.

Example:

getboardenablestatus016	get the status of board number 016
ENABLE	the board is enabled

[Back to Index](#)

Automated Test Commands

GetTestResult - Return the Test Result of a Panel

Transmit	Receive
GetTestResult	[PASS] or [FAIL]

Description

This command instructs the Hub to return the Test Result of a Panel.

The result can be either [PASS] or [FAIL]. If the result is [PASS] this means that the panel has passed the last test. If the result is [FAIL] then the panel has failed the test.

Example:

```
gettestresult      has the panel passed or failed ?  
[PASS]            the panel has passed
```

[Back to Index](#)

Feasa Legend LED Analyser



Command Summary

COMMAND	DESCRIPTION
<i>Capture</i>	Capture LED Color and Intensity Data - Auto Range
<i>Capture#</i>	Capture LED Color and Intensity Data - Manual Range
<i>CaptureTest</i>	Combined capture and test
<i>CaptureMultiRange</i>	Capture using pre-programmed multiple ranges
<i>Detect</i>	Detect LED Analysers connected to a Hub
<i>getBoardEnableStatus</i>	Return the Enable or Disable status of a board
<i>getNFib</i>	Get the number of fibers available
<i>getRGBI###</i>	Get RGB, Saturation and Intensity for a LED
<i>getRGBIall</i>	Get RGB, Saturation and Intensity for all LEDs
<i>getHSI###</i>	Get Hue, Saturation and Intensity for a LED
<i>getHSIall</i>	Get Hue, Saturation and Intensity for all LEDs
<i>getwavelength###</i>	Get Dominant Wavelength for a LED
<i>getwavelengthall</i>	Get Dominant Wavelength for all LEDs
<i>getwi###</i>	Get Dominant Wavelength and Intensity for a LED
<i>getwiall</i>	Get Dominant Wavelength and Intensity for all LEDs
<i>getxy##</i>	Get the xy Chromaticity values for a LED
<i>getxyall</i>	Get the xy Chromaticity values for all LEDs
<i>getxyI###</i>	Get the xy and Intensity values for a LED
<i>getxyIall</i>	Get the xy and Intensity values for all LEDs
<i>getuv##</i>	Get the u'v' Chromaticity values for a LED
<i>getcct###</i>	Get CCT values for a LED
<i>getcctall</i>	Get CCT values for all LEDs
<i>getcctiall</i>	Get CCT and Intensity values for all LEDs
<i>getuv###</i>	Get the u'v' Chromaticity values for a LEDs
<i>getuvall</i>	Get the u'v' Chromaticity values for all LEDs
<i>getIntensity###</i>	Get the Intensity for a LED

Feasa Legend LED Analyser



<i>getIntensityall</i>	Get the Intensity for all LEDs
<i>getxoffset</i>	Get the x offset value for a Fiber
<i>getyoffset</i>	Get the y offset value for a Fiber
<i>getIntGain</i>	Get the Intensity gain factor for a Fiber
<i>getfactor</i>	Get the exposure factor for all fibers
<i>getPortInfo</i>	List all the connected LED Analysers
<i>getPortDetails</i>	List all the connected LED Analysers
<i>getLimitGroup</i>	Get the current active limits group
<i>getSerial</i>	Get the Serial Number of the Hub
<i>getVersion</i>	Get the Firmware Revision of the Hub
<i>getHW</i>	Get the Hardware Version of the Analyser
<i>Test</i>	Test LED's using the active limits group
<i>ReportLimit</i>	Limit the number of failures from the Test command
<i>setBoardEnable</i>	Enable a board in a Panel
<i>setBoardDisable</i>	Disable a board in a Panel
<i>setIntGain</i>	Set the Intensity Gain Factor
<i>setfactor</i>	Set the exposure factor
<i>setLimitGroup</i>	Set the active limits group
<i>setxoffset</i>	Set the x offset for a fiber
<i>setyoffset</i>	Set the y offset for a fiber
<i>PrinterOn</i>	Turn on the Printer Function
<i>PrinterOff</i>	Turn off the Printer Function
<i>Getxoffset# #</i>	Get x Offset
<i>getyoffset# #</i>	Get y Offset
<i>getwavelengthoffset# #</i>	Get Wavelength Offset

Feasa Legend LED Analyser



Specifications

Part Number(s)

Feasa 20-L Legend Analyser with 20 Fibers
Feasa 10-L Legend Analyser with 10 Fibers
Feasa Hub-8 Hub Controller with 8 Ports

Physical

Analyser dimensions 120mm x 29mm x 50mm (L x W x H)
Hub-8 dimensions 127mm x 57mm x 37mm
Fiber Length 0.6m
Fiber Diameter 1.0mm
Fiber Core Diameter 0.5mm
Number of Fibers 20 or 10
Operating Temperature Range 0°C to +50°C
** does NOT include Fiber bend Radius*

Electrical

Hub Supply Voltage 5.0V
Hub Supply Current 2.5A (Power Supply is supplied with Hub)
Satellite Supply Current 200mA (Supplied through Hub using External PSU)
Output Data Format RGB, HSI, XY, UV, Wavelength, CCT & Intensity

Optical

Red Peak Efficiency Wavelength	615 nm
Green Peak Efficiency Wavelength	540 nm
Blue Peak Efficiency Wavelength	465 nm
Total Operating Wavelength Range	450 nm to 650 nm

Accuracy

White $X = \pm 0.01$, $Y = \pm 0.01$ @ $x=0.33$, $y=0.33$ using OH-3 Optical Head
Red (630nm) $\pm 4\text{nm}$
Green (520nm) $\pm 2\text{nm}$
Blue (450nm) $\pm 5\text{nm}$
Yellow (590nm) $\pm 2\text{nm}$

These accuracy's are based on the Satellite Analyser being calibrated at a point in time against known set values.

[Back to Index](#)

Feasa Legend LED Analyser



Warranty

1. Feasa Enterprises Limited (herein referred to as Feasa) warrants Feasa hardware, accessories and supplies against defects in materials and workmanship for the period of one year. If Feasa receives notice of such defects during the warranty period, Feasa will, at its option, either repair or replace products which prove to be defective. Replacement products may be either new or like-new.
2. Feasa warrants that Feasa software will not fail to execute its programming instructions, for the period of one year, due to defects in material or workmanship when properly installed and used. If Feasa receives notice of defects during the warranty period, Feasa will replace software media which does not execute its programming instructions due to such defects.
3. Feasa does not warrant that the operation of Feasa products will be uninterrupted or error free. If Feasa is unable, within a reasonable time, to repair or replace any product to a condition as warranted, customer will be entitled to a refund of the purchase price upon prompt return of the product to Feasa.
4. Feasa products may contain re-manufactured parts equivalent to new in performance or may have been subject to incidental use
5. The warranty period begins on the date of deliver
6. Warranty does not apply to defects resulting from:
 - (a) improper or inadequate maintenance or calibration,
 - (b) software, interfacing, parts or supplies not supplied by Feasa,
 - (c) unauthorized modification or misuse,
 - (d) operation outside the published environmental specifications for the product, or
 - (e) improper site preparation or maintenance.
7. TO THE EXTENT ALLOWED BY LOCAL LAW, THE ABOVE WARRANTIES ARE EXCLUSIVE AND NO OTHER WARRANTY OR CONDITION, WHETHER WRITTEN OR ORAL, IS EXPRESSED OR IMPLIED AND FEASA SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, SATISFACTORY QUALITY, AND FITNESS FOR A PARTICULAR PURPOSE.
8. TO THE EXTENT ALLOWED BY LOCAL LAW, THE REMEDIES IN THIS WARRANTY STATEMENT ARE CUSTOMER'S SOLE AND EXCLUSIVE REMEDIES. EXCEPT AS INDICATED ABOVE, IN NO EVENT WILL FEASA OR ITS SUPPLIERS BE LIABLE FOR LOSS OF DATA OR FOR DIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL (INCLUDING LOST PROFIT OR DATA), OR OTHER DAMAGE WHETHER BASED IN CONTRACT, TORT, OR OTHERWISE.

[Back to Index](#)