



User Manual for Feasa ICT Models



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

www.feasa.ie

Email: sales@feasa.ie

Rev. 10.2 ICT Version
Date: Dec 2024

FEASA LED ANALYSER ICT VERSION



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 In Circuit Test (ICT) Led Analysers. The model numbers are Feasa x-I and the Feasa x-IB (where x is the no. of fibers either 3, 5, 6, 10, 20).

The **Feasa 20-I** is a standard 20 channel unit which will test up to 20 LEDs while the **Feasa 3-I** is a standard 3 channel unit testing up to 3 LED's. This utilizes a Plastic Optical Fiber 1mm in Diameter.

The **Feasa 20-IB** is a High Brightness 20 channel unit which will test up to 20 LEDs while the **Feasa 3-IB** is a 3 channel High Brightness unit testing up to 3 LED's. This utilizes a Plastic Optical Fiber 1mm in Diameter.

The interface on these units is RS232-Serial and ICT Interface (Frequency Mode / Synchronous Serial Mode/ Continuous Frequency Mode).

© 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 LED ANALYSER ICT VERSION



Table of Contents

Introduction	7
IMPORTANT Information for Programmers	8
Colour/Hue and Saturation	9
Intensity / Signal Level	11
White LED's	13
Setting Tolerance Limits for Hue and Intensity	14
Physical Layout	15
Output Test Mode	16
In-Circuit Test Mode	17
Frequency Out Port Protocol	17
ICT Connector Pin-out	18
ICT Connector Pin Descriptions	19
ICT Logic Levels & Cycle Times	20
Frequency Responses	21
Hue vs Frequency	23
Saturation vs Frequency	24
Wavelength vs Frequency	25
CIE1931 vs Frequency	26
In-Circuit Test Frequency Mode	27
Initialize the LED Analyser for Capture	29
Set the Sensitivity Level of the Analyser (setfactor)	30
Capture LED Data (HSI or WSI or xyI) Static LED's	31
Capture LED Data (HSI or WSI or xyI) PWM LED's	32
Test LED Data (HSI or WSI or xyI)	33
Step-by Step Sequence for LED Analyser Auto Capture & Read HSI or WSI or xyI	34
Capture Times ICT Mode – Static Leds	36
In Circuit Test Synchronous Serial Mode	37
Synchronous Serial Pin Descriptions	38
Synchronous Serial Commands	39
Clear Comms Buffer	40
Set the Sensitivity Factor (setfactor)	41
Sync Serial Capture for Static LED's	42
Step-by Step Sequence for LED Analyser Capture Static LED's	43
Capture Times Synchronous Mode – Static Leds	44
Sync Serial Capture for PWM LED's	45
Step-by-Step Sequence for LED Analyser Capture for PWM LED's	46
Capture Times Synchronous Mode – PWM Leds	47
Capture / CPWM Command Timing Diagram	48
Read Hue, Saturation and Intensity (HSI)	49
Read RGB and Intensity (RGBI)	51
Dominant Wavelength (W)	53
Wavelength & Saturation & Intensity (WSI)	55
Colour Temperature (cct)	57
x-chromaticity, y-chromaticity & Intensity (xyi)	59
Absolute Intensity	61

FEASA LED ANALYSER ICT VERSION



Continuous Frequency Mode.....	63
SetContinuousMode - Enable the Continuous Frequency Mode.....	65
SetContinuousChannel## - Select which fiber you are using.....	65
SetContinuousCapture## - Select which capture range you are using.....	65
GetContinuousMode - Verify Continuous Frequency Mode On/Off.....	66
GetContinuousChannel - Verify which fiber you are using.....	66
GetContinuousCapture - Verify which capture range you are using.....	66
Serial Port Control (RS232).....	67
Capture Commands.....	68
Capture Times Static Leds.....	69
Capture Times PWM Leds.....	69
AUTO CAPTURE - Store LED Data.....	70
MANUAL CAPTURE# - Store LED Data for a specific range.....	71
AUTO CAPTUREPWM - Capture PWM LED Data.....	72
MANUAL CAPTURE - Capture PWM LED Data for a specific range.....	73
Get Data Commands.....	75
Under Range Condition.....	75
Over Range Condition.....	75
Incorrect Capture Mode.....	75
getRGBI## - Get RGB and Intensity for a LED.....	76
getHSI## - Get Hue, Saturation and Intensity.....	77
getxy## - Return the xy Chromaticity values.....	78
getxyi## - Return the xy Chromaticity & Relative Intensity values.....	79
getUV## - Return the u'v' Chromaticity values.....	80
getWAVELENGTH## - Get the Dominant Wavelength.....	81
getWI## - Get the Dominant Wavelength & Intensity.....	82
getWSI## - Get the Dominant Wavelength, Saturation & Intensity.....	83
getCCT## - Get the Correlated Colour Temperature.....	84
getINTENSITY## - Get the Intensity.....	85
getABSINT@@ - Get the Absolute Intensity Value of the Led under test.....	86
getCIEXYZ## - Return the CIE 1931 XYZ Colour values.....	87
getSIGNALLEVEL## - Get the Relative Intensity in a percentage of the Range.....	88
get7SEG# - Get the value of a 7 Segment Display.....	89
getFACTOR - Get the exposure Factor.....	91
getAUTOPWM - Get which Auto Capture Mode is active C or CPWM.....	92
getPHOTOPIC - Get which Photopic Mode is active.....	93
getintensitymode - Get which Intensity Mode is active Logarithmic or Linear.....	94
getSERIAL Get the Serial Number of the Analyser.....	95
getVERSION - Get the Firmware Version.....	96
getBAUD - Get the Baud Rate.....	97
getHW - Get the Hardware Version.....	98
getNFIB - Get the the Number of Channels on the Analyser.....	99
getSTATUS - Get a summary of the Led Analyser details.....	100
getICTSTATUS - Get the Led Analyser ICT Mode details.....	101
EOT - End of Transmission Character.....	102
General Set/Put Commands.....	103
set/putFACTOR## - Set the Exposure Factor.....	104

FEASA LED ANALYSER ICT VERSION



set/putLOG - Change the Intensity Response of the Analyser to Logarithmic mode	105
set/putLIN - Change the Intensity response of the Analyser to Linear Mode	106
setBAUD - Change the baud rate	108
setAUTOPWM# - Change the Auto Capture mode to Auto PWM Mode	109
setFREQUENCYxxx - Set the Frequency Output Mode of the Analyser	110
setSyncSerialx - Set the Output Mode of the Analyser to SyncSerial	111
setPHOTOPIC - Set which Photopic Mode you require	112
User Calibration Mode	113
Introduction:	113
set/putINTGAIN - Set the Intensity GainFactor	114
getINTGAIN## - Get the Intensity Gain Factor	115
set/putXOFFSET##0.xxx - Set the x Chromaticity Offset	116
set/putYOFFSET##0.yyy - Set the x Chromaticity Offset	117
getXOFFSET## - Return the x Chromaticity offset	118
getYOFFSET## - Return the y Chromaticity offset	119
set/putWAVELENGTHOFFSET@@±## - Set the wavelength Offset	120
getWAVELENGTHOFFSET@@ - Get the Dominant Wavelength Offset	121
set/putABSINTMULT@@ - Set the Absolute Intensity Multiplier Factor	122
getABSINTMULT@@ Get the Absolute Int Correction Factor of the Led under test	123
setcalibration# - Select the user cal setting	124
getCALIBRATIONset - Select the Calibration Set to use	125
ResetUserCal - Reset the usercal setting to factory default	126
SetCalibrationDateddmmyyyy - Set the user Cal Date	127
getcalibrationdate Get the Calibration Date set by the user	128
External Trigger Mode	129
Sequence Capture Mode:	131
Introduction	131
Capture Sequence Procedure:	132
Set/Put SequenceMode# - Set sequence mode for Fibers	133
Set/Put SequenceChannel@@ - Select sequence mode Fiber No.	134
Set/Put SequenceCapture##(##) - Set Sequence Capture Time	135
Set/Put SequenceWait##(##) - Set Sequence Wait Time	136
Set/Put SequenceNumber####(##) - Set the Number of Captures	137
Set/Put SequenceThreshold##### - Set the Intensity Threshold	138
Set/Put SequenceStartDelay### - Set delay in ms before sequencecapture starts	139
CaptureSequence - Start the Captures	140
StoreSequence## - Store the sequence pattern	141
IdentifySequence - Identify the current sequence pattern	142
ResetSequence## - Reset a sequence pattern	143
GetSequenceStored## - Get a sequence pattern	144
GetSequence@@ - Get the binary sequence of Intensities	145
GetSequenceTimes@@ - Get the Duty Cycle of the Sequence	146
GetSequenceTimesAll - Get the Duty Cycle of the Sequence for all fibers	147
GetFrequency## - Returns the frequency and duty cycle of Led	148
Get(Max/Min/Avg)Intensity## - Returns the Max/Min or Avg Intensity for the Seq Cap.	149
GetFlicker## - Returns the Flicker Index and frequency after a Seq Cap.	150
GetSequenceMode# - Get sequence mode for Fibers	151

FEASA LED ANALYSER ICT VERSION



GetSequenceChannel - Select sequence mode Fiber No.	152
GetSequenceCapture - Get Sequence Capture Time	153
GetSequenceWait - Get Sequence Wait Time	154
GetSequenceNumber - Get the Number of Captures	155
GetSequenceThreshold - Get the Intensity Threshold	156
GetSequenceStartDelay - Get the delay before the CS command	157
Capture Sequence Sync Serial Commands	158
Capture Sequence Setup	159
Typical CS Capture Times are as follows:	160
Step-by Step Sequence for LED Analyser CS Setup	161
Step-by Step Sequence for LED Analyser CS Capture	162
Step-by-Step approach to Testing a LED	163
Effect of PWM on Intensity Testing	164
Step-by-Step approach to Testing a PWM LED	165
Testing a 7-Segment Display	166
Command Summary:	167
Specifications	170
ECO:	171
Warranty	172

FEASA LED ANALYSER ICT VERSION



Introduction

The **Feasa LED Analyser** is an instrument that tests the following Led Parameters of Light Emitting Diodes (LEDs) in a test process.:

HSI (Hue, Saturation, Intensity),
RGB (Red, Green, Blue colour content),
xy Chromaticity,
Dominant Wavelength

cct colour temperature of Light Emitting Diodes (LEDs) in a test process. An individual Led Analyser can have up to 20* flexible Fiber-Optic Light Guides which are mounted individually over the LEDs to be tested.

The Feasa Led Analyser can be trained to read other Led Parameters like :

Absolute Intensity
CIEXYZ in Luminous flux (lm),
Luminous intensity (cd, mcd),
Luminance (cd/m²).

It can also be trained to interpret the data from an RGB Led. These outputs will require the use of the Feasa Led Spectrometer.

Emitted Light from the LEDs is guided through these Fiber-Optic Light Guides to the Analyser where the raw data is stored. The raw data can then be read out of the Led Analyser through the Serial or ICT Interfaces.

The ICT Interface is a 20-Pin Connector compatible with In-Circuit Test machines. There is a choice of two operating modes using the ICT Interface. The first mode uses frequencies to output the LED data and the second mode uses a high-speed Synchronous Serial Bus to output the Led data.

The Serial Interface is a 3-Pin Connector with a 3 pin to 9 pin dtype cable (supplied) compatible with the RS232 port on machines.

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

The raw data from the Led under test is stored on the Led Analyser and output in different formats as required through the Serial and ICT interfaces.

* Other options include 3,5,6,10.

[Back to Index](#)



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 download link provided along with the LED Analyser.

[Back to Index](#)



Colour/Hue and Saturation

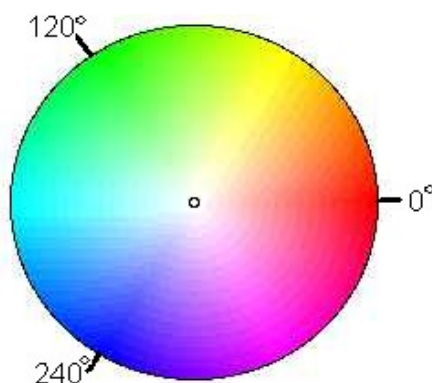


Figure 1: Hue (colour) Wheel

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 centre 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 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 100% represents a pure Colour such as Red, Green, Blue etc.

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

[Back to Index](#)



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	37%

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 through the Serial or ICT ports.

[Back to Index](#)



Intensity / Signal Level

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 Signal Level is represented as a percentage of the range **0 - 100%**. For example an Intensity reading of 35000 is equivalent to a Signal Level reading of 035%. 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.

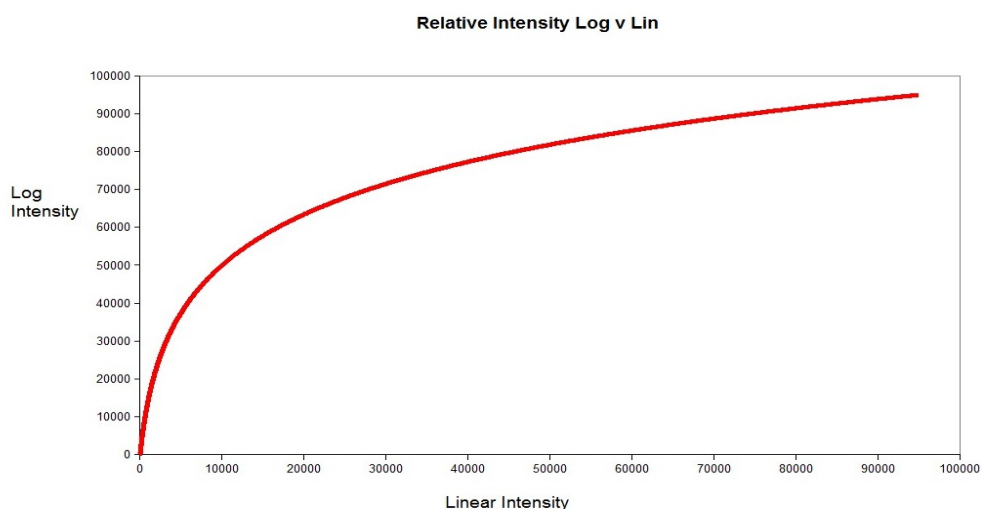


Figure 3a. Intensity for the LED Analyser in LOGARITHMIC Mode

[Back to Index](#)

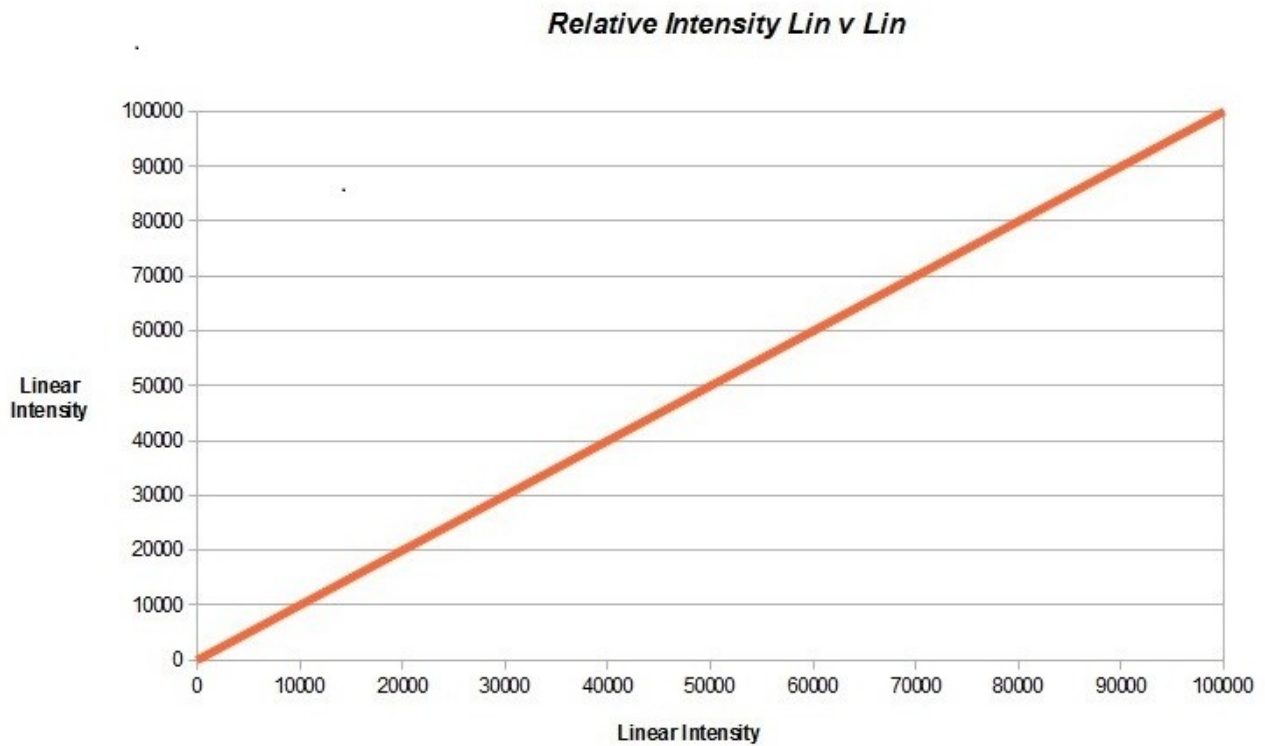


Figure 3b. Intensity for the LED Analyser in LINEAR Mode

[Back to Index](#)



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 values must 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 is the LED being tested. 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.

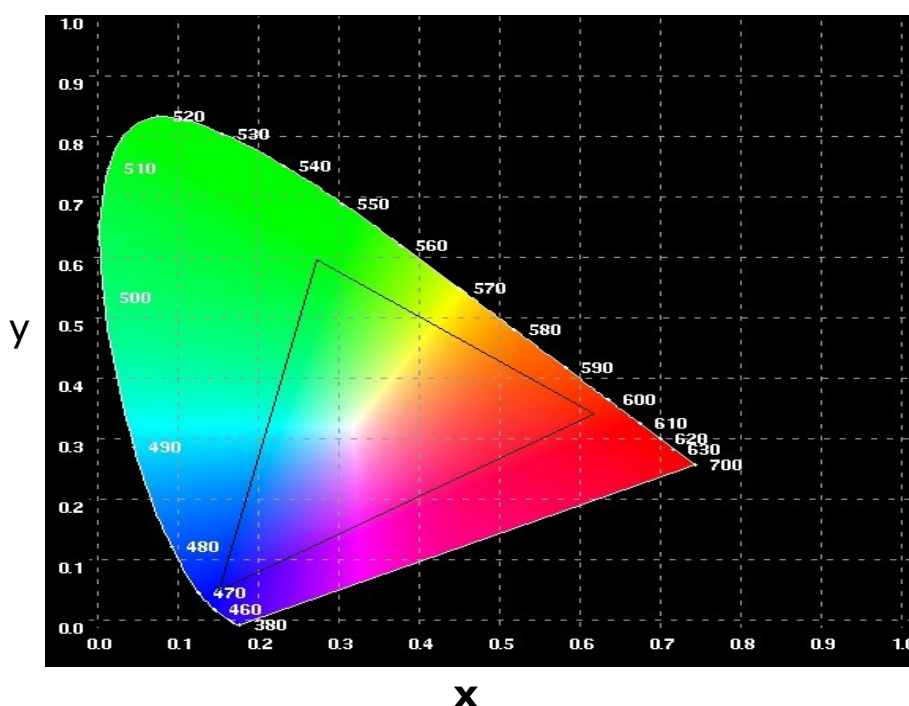


Figure 4. CIE 1931 Chromaticity Co-Ordinates

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 other LED's will have a Red tint (Warm White).

[Back to Index](#)



Setting Tolerance Limits for Hue and Intensity

The test procedure requires the user to set the Pass/Fail limits for Hue and Intensity for each LED which then become the standard against which LEDs to be tested are compared. The Pass/Fail limits for Hue 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 Hue in terms of their Hue value.

Sample Hue Pass/Fail Limits

LED	Minimum	Maximum
Red	0	2
Green	110	130
Blue	220	250
Amber	2	10
Yellow	80	110
Orange	10	20

The Saturation value must be taken into account when testing White LED's. The Saturation is a value between 0% and 100%. A pure white will have a value close to 0% while a pure colour will have a value close to 100%. The Pass/Fail limits for intensity are chosen from the average intensity values from a number of sample LEDs.

Sample Intensity Pass/Fail Limits in Auto Mode

Intensity Value	Upper Limit	Lower Limit	Comments
125	150	100	This represents a Very Dim LED, Recommend using Manual Capture Range C1, C2 or C1PWM, C2PWM
20000	25000	16000	This represents an Average LED, Recommend using Manual Capture Range, C2, C3 or C2PWM, C3PWM
70000	85000	56000	This represents a Bright LED, Auto capture use is OK but Manual Range C3, C4, C5 or C3PWM, C4PWM, C5PWM would be recommended.

Figure 5.

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

Note:- Feasa recommends setting the Capture Range to give Intensity Values of 55K to 85K for the best results..

[Back to Index](#)

FEASA LED ANALYSER ICT VERSION



Physical Layout

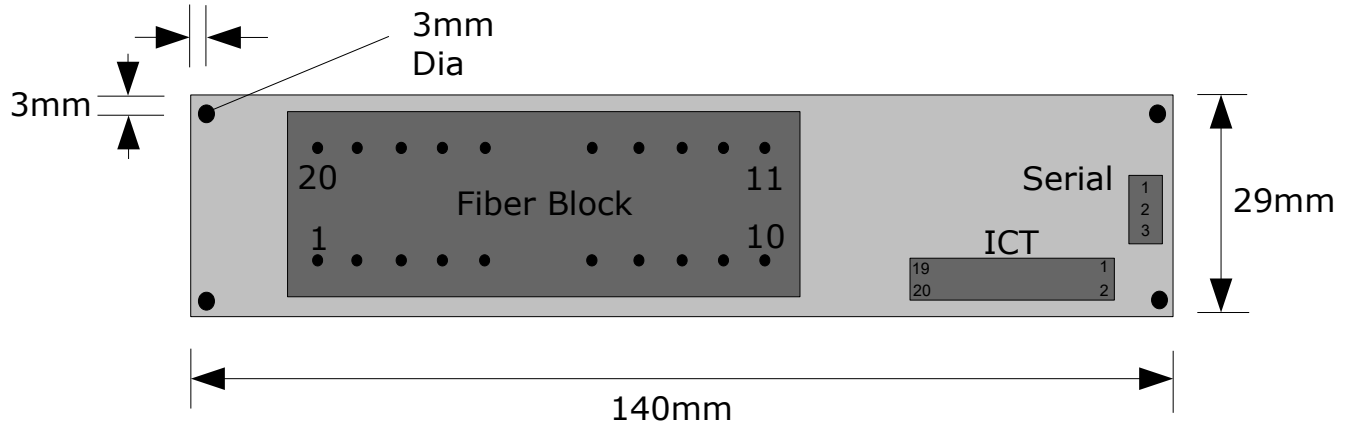


Figure 6a (Feasa 20-I, 20-IB).

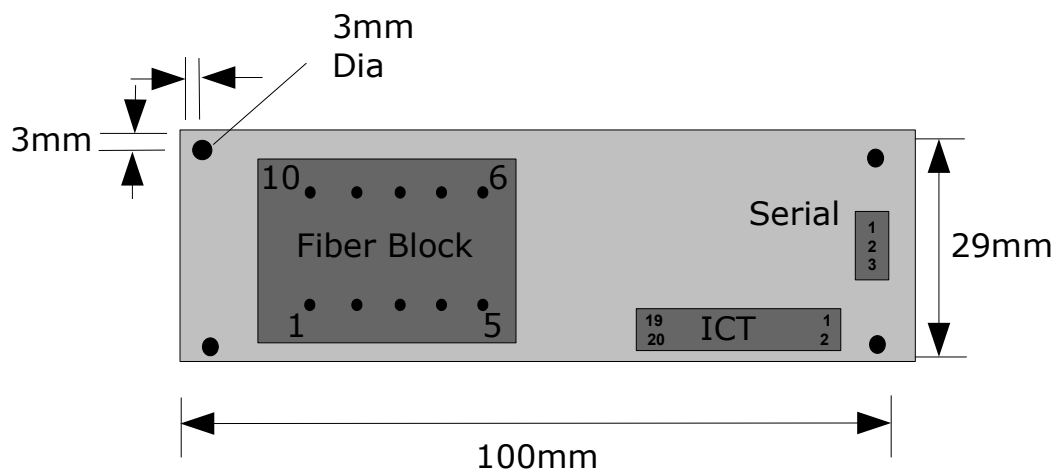


Figure 6b (Feasa 3-I, 5-I, 6-I, 10-I, 3-IB, 5-IB, 6-IB, 10-IB)

Figure 6a shows the physical layout of the Feasa 20-I & Feasa 20-IB Analyser's. The fibers are labelled 1-20.

Figure 6b shows the physical layout of the Feasa 10-I & 10-IB Analyser's.

The Fibers are labeled 1-10. Options include Feasa 3, 5, 6 channel

There are four 3mm diameter Mounting holes at the corners.

The ICT Port is a 20-pin connector while the Serial Port is a 3-pin connector.

The PCB has been designed so that it can easily be fitted in the top or bottom side of a Fixture.

Please refer to the Fixturing Guidelines Manual for connector pin out details.

[Back to Index](#)



Output Test Mode

Output Test Mode

In addition to the RS232 Serial Port Communications the Led Analyser provides a Port for In-Circuit Testing. There are two methods that can be used with this Interface. The first method use frequencies to represent the data for each LED. The second method uses a Synchronous Serial Digital mode to read the data.

The Frequency method is the easier to implement and debug while the Synchronous Serial method is a little more complex but is faster especially for large numbers of LED's, and it also uses less resources.

The Led analyser also has the ability to output continuous Frequency to represent the Led data HSI, WSI and XYI. This is called Continuous Frequency mode and is detailed in the relevant section of this manual.

By default the Analyser is supplied with Frequency Out HSI Mode **ENABLED** and Synchronous Serial Mode **DISABLED**. The Continuous Frequency mode is also **DISABLED** when shipped from the Factory.

The LED Analyser is set to the **Frequency Output - HSI** (Hue, Saturation, Intensity) mode at the Factory. There are two further frequency out options, **WSI** (Wavelength, Saturation, Intensity) and **xyI** (X chromaticity, Y chromaticity, Intensity). To place the analyser in WSI mode the command [setfrequencywsi](#) must be sent to the analyser. To place the analyser in xyI mode the command [setfrequencyxyi](#) must be sent to the analyser. The command [getICTStatus](#) will verify which modes have been selected. The commands must be sent using the Feasa Terminal window. The configuration is stored in Flash memory and need only be implemented once i.e. It will remain in the selected mode until instructed to change.

To Enable the Synchronous Serial Mode use the command [setsyncserial1](#). To disable the Synchronous Serial Mode use the command [setsyncserial0](#). These commands are available on Firmware Versions i210 upwards.

Summary of Commands:

getictstatus:	This will indicated which modes the analyser is set for.
setfrequencyhsi	Set the Analyser to Frequency Mode measuring HSI
setfrequencywsi	Set the Analyser to Frequency Mode measuring WSI
setfrequencyxyi	Set the Analyser to Frequency Mode measuring XYI
setsyncserial1	Enable the Analyser for Synchronous Serial Mode
setsyncserial0	Disable the Analyser for Synchronous Serial Mode
setcontinuousmode1	Enable the Analyser for Continuous Frequency Mode
setcontinuousmode0	Disable the Analyser for Continuous Frequency Mode

[Back to Index](#)



In-Circuit Test Mode

Frequency Out Port Protocol

This method is based on extracting LED data from the frequency of square wave output signals. There are three separate outputs and depending on the mode selected the frequency out represents – one each for Hue, Saturation and Intensity in HSI mode, one each for Dominant Wavelength, Saturation, Intensity in WSI mode and one each for x chromaticity, y chromaticity, Intensity in xyI mode. The frequency of the square wave (duty cycle is always at 50%) signal from the Led Analyser is in the range 1KHz to 100KHz. This is a fast and simple method of communication and is efficient in terms of tester resources where a frequency counter is usually available.

The Frequency value for Intensity (pin 15), Hue, Wavelength or x chromaticity (pin 16) and Saturation or y chromaticity (pin 17) of the LED are output simultaneously at the ICT Port.

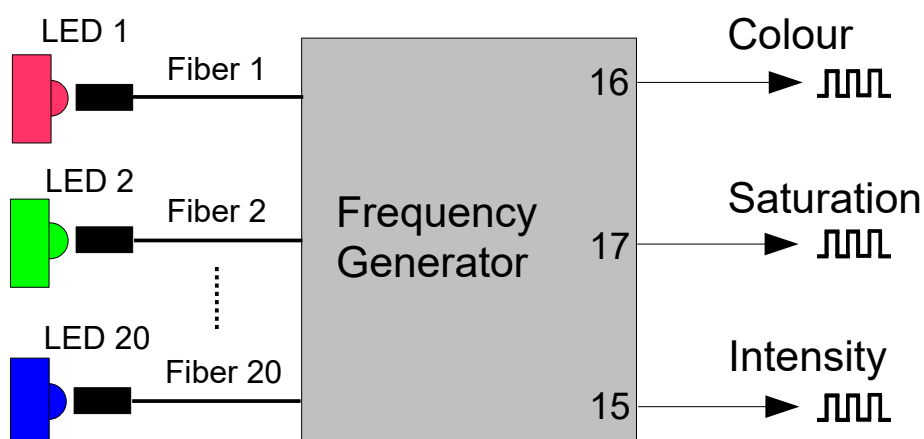


Figure 7.

[Back to Index](#)



In-Circuit Test Mode

ICT Connector Pin-out

The ICT Port consists of a 20-way connector which includes all the signals necessary to interface with the LED Analyser. These pins are shown in Figure 8.

Pin	Type	ICT Signal Name	Function
1	Output	Ser_Out	Synchronous Serial Output
2	Input	Ser_In	Synchronous_Serial Input
3	Input	Ser_Clk	Synchronous Serial Clock
4	Digital Ground	GND1	Led Analyser Digital Ground
5	Input (Active Low)	Reset_bar	Reset input
6	Input (Active Low)	OE_bar	Output Enable
7	Input	Ext_Trig	For External Capture
8	Input (Active Low)	PWM	Select PWM mode
9	Input	Addr_0 LA_Select	LSB of Fiber Address (also used to select Synchronous Serial mode)
10	Input	Addr_1	Fiber Address
11	Input	Addr_2	Fiber Address
12	Input	Addr_3	Fiber Address
13	Input	Addr_4	MSB of Fiber Address
14	Input (Active Low)	WE_bar	Write Enable
15	Output	Int	Frequency Out Square wave for Intensity
16	Output	Hue	Frequency Out Square wave for Hue, Wavelength, x Chromaticity
17	Output	Sat	Frequency Out Square wave for Saturation, y Chromaticity
18	Output	RyBy_bar	Ready Busy Output (Also Used for Synchronous Serial mode)
19	Power	VCC	Power Supply 5V DC at 180mA
20	Ground	GND	Power Supply Ground

- **NOTE:** **_bar & /** Indicate Active Low
- **NOTE:** Logic Level Voltages 3.3V

Figure 8.

[Back to Index](#)



In-Circuit Test Mode

ICT Connector Pin Descriptions

Pin 4 GND1

This is a Digital ground pin for the Analyser. Ensure this Digital ground pin is also tied to PSU Ground for correct operation.

Pin 5 Reset (/Reset)

This input resets the LED Analyser and is active low. When /Reset goes high the Analyser starts operating in normal mode.

Pin 6 Output Enable (/OE)

This input enables the Frequency Output Buffers when driven low.

Pin 7 Trigger

A Falling edge on the pin will trigger a capture on the Led Analyser. For this function to work an External Capture command must be activated. (see External Capture Section)

Pin 8 Select Output Mode (/PWM)

This input pin is used to select between Standard Capture (Static LED's) and PWM modes. If this pin is high Standard Capture mode is enabled and if this pin is low PWM mode is selected. This pin is pulled up internally by the Analyser so that Standard Capture mode is selected if left floating. This pin is active during the capture cycle.

Pin 9 – 13 Addrss Pins

The 5 Address inputs Addr_0, Addr_1, Addr_2, Addr_3 are used in the Frequency Output mode. A complete list of all the Addresses is shown in Figure 12. In Addition, Addr0 is used in the Synchronous Serial mode to select the Analyser (LA_Select).

Pin 14 Write Enable

This input is used to write commands into the Analyser. It is active low.

Pin 15 Intensity Out

This pin outputs the Intensity Frequency for the selected LED. It is a square-wave signal (0 – 3V) and will have a frequency in the range 1KHz – 100KHz.

Pin 16 Hue Out, Wavelength Out, X – Out

This pin outputs the Hue (Colour), Dominant Wavelength or x Chromaticity frequency for the selected LED. It is a square-wave signal (0 – 3V) and will have a frequency in the range 1KHz – 100KHz.

Pin 17 Saturation Out, Y- Out

This pin outputs the Saturation (Whiteness) or Y Chromaticity Frequency for the selected LED. It is a square-wave signal (0 – 3V) and will have a frequency in the range 1KHz – 100KHz.



In-Circuit Test Mode

ICT Connector Pin Descriptions Cont'd

Pin 18 Ready Busy Output (/RyBy)

This is the Ready/Busy output pin. It is used to indicate the status of the LED Analyser during operation. A high on this pin indicates the Analyser is busy and a low indicates the Analyser is ready for the next operation.

Pin 19 VCC (5.0V)

This is the Voltage supply pin to the Analyser. The power requirements are 5V DC at 180mA.

Pin 20 GND

This is Voltage Supply ground pin for the Analyser. Ensure this ground is tied to System Ground for correct operation.

Ensure Pins 4 & 20 GND are wired to Agilent System Ground and also ensure this pin is wired to the Agilent PSU GND that is used to power the Analyser.

ICT Logic Levels & Cycle Times

```
CMOS_3V3
Drive High  3.2;
Drive Low   0.1;
Receive High 2.4;
Receive Low  0.6;
Edge Speed  50;
Open Input Default X;
Load        NONE;
```

```
vector cycle 20u    ! minimum time 9u
receive delay 19u    ! minimum time 8u
```

```
test time 3        ! allow 3 seconds maximum time to capture the data
```

Agilent 3070 Model details are located on the Download Site at :
ICT/Agilent 3070/custom_Lib_20i

and Teradyne/GenRad details are located on the Download Site at :
ICT/Teradyne

[Back to Index](#)



In-Circuit Test Mode

Frequency Responses

A list of typical Hue and Saturation frequencies for different LED's is shown in *Figure 9a*. These values can be used initially to set up test limits in a Test Program. Adjustments should be made during the debug process to suit the actual LED's being tested.

LED Colour	Hue		Saturation		Intensity	
	Numeric	Freq.	Numeric	Freq.	Numeric	Freq.
Red	0.03	1,03kHz	100	100kHz	43871	44,9kHz
Green	121.56	44.8kHz	98	98kHz	52528	53,3kHz
Blue	235.9	71,8kHz	100	100kHz	68017	68,9kHz
Orange	3.08	4,08kHz	100	100kHz	3241	4,24kHz
Yellow	53.76	28,99kHz	100	100kHz	33488	34,48kHz
Warm White	61.74	30,85kHz	90	89,5kHz	63218	64,17kHz
Neutral White	100.36	40kHz	70	69,77kHz	73519	74,53kHz
Cool White	139.64	49,18kHz	28	28,04kHz	69852	70,59kHz

**Note: measurements done with Feasa 3-I + OH-3 @ 3mm, using Firmware version I209.*

***Note: tolerances assumed in the results due to Optical Head, fiber, frequency measurement and power supply noise.*

Figure 9a.

FEASA LED ANALYSER ICT VERSION



A list of typical XY & Wavelength frequencies for different LED's is shown in *Figure 9b*. These values can be used initially to set up test limits in a Test Program. Adjustments should be made during the debug process to suit the actual LED's being tested.

LED Colour	x		y		Wavelength	
	Numeric	Freq.	Numeric	Freq.	Numeric	Freq.
Red	0.7198	71,8kHz	0.2764	27,6kHz	646	64,5kHz
Green	0.2648	26,4kHz	0.6961	69,7kHz	549	54,8kHz
Blue	0.1324	13,2kHz	0.0648	6,5kHz	465	46,5kHz
Orange	0.6527	65,2kHz	0.3428	34,2kHz	607	60,6kHz
Yellow	0.5793	57,9kHz	0.4188	41,8kHz	589	58,8kHz
Warm White	0.4854	48,5kHz	0.44	44,1kHz	586	58,54kHz
Neutral White	0.3959	39,6kHz	0.3856	38,5kHz	576	57,6kHz
Cool White	0.3215	32,1kHz	0.3381	33,8kHz	514	51,5kHz

**Note: measurements done with Feasa 2-I + OH-3 @ 3mm, using Firmware version I209.*

***Note: tolerances assumed in the results due to Optical Head, fiber, frequency measurement and power supply noise.*

Figure 9b.

Figure 9c shows the under range, over range conditions that will be reported.

Condition	Hue	Saturation	Intensity
Good	1KHz- 100KHz	1KHz - 100KHz	1k - 96KHz
Over Range	500Hz	500Hz	100KHz
Under Range	500Hz	500Hz	1KHz
Error	500Hz	500Hz	500Hz

Figure 9c.

[Back to Index](#)



In-Circuit Test Mode

Hue vs Frequency

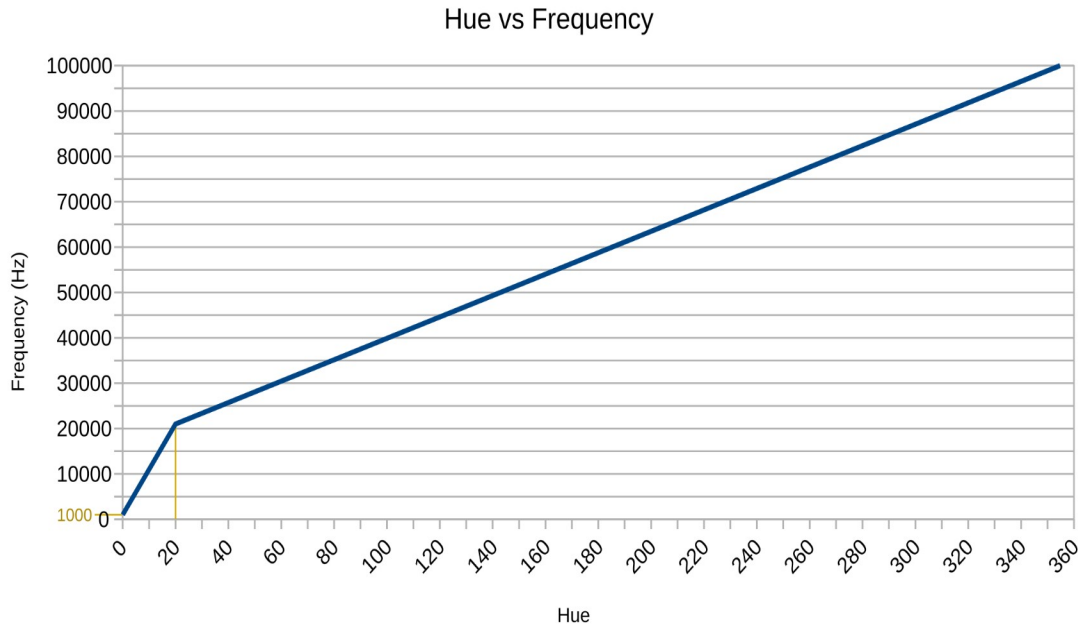


Figure 10a.

$$Hue = \begin{cases} \frac{Frequency_{Hue} - 1000}{1000} & , \text{for } Frequency_{Hue} < 21 \text{ kHz} \\ 20 + \frac{Frequency_{Hue} - 21000}{236} & , \text{for } Frequency_{Hue} \geq 21 \text{ kHz} \end{cases}$$

[Back to Index](#)



In-Circuit Test Mode

Saturation vs Frequency

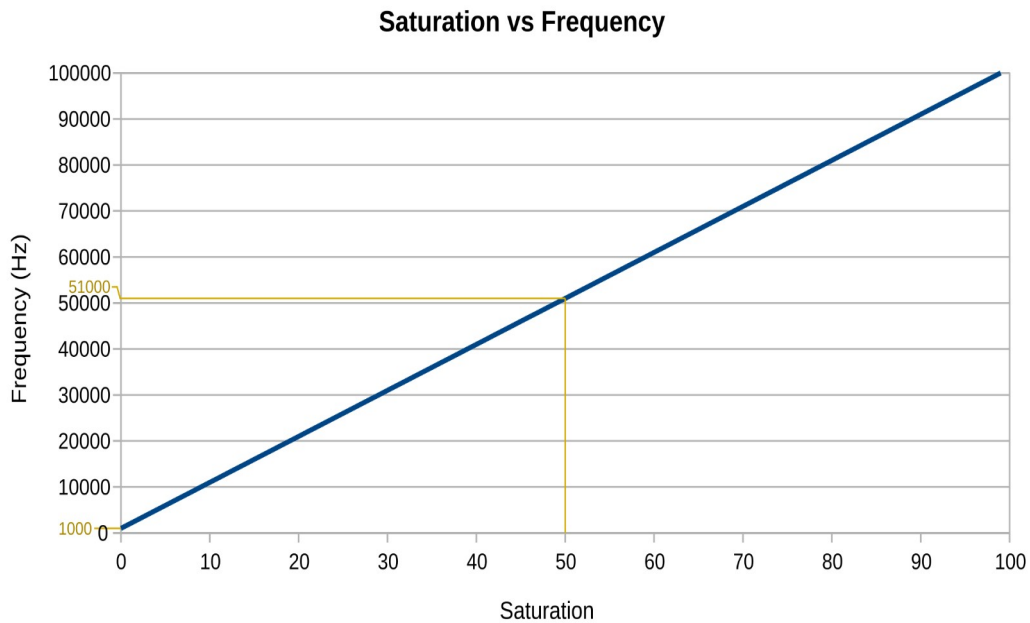


Figure 10b.

$$Saturation = \frac{Frequency_{Saturation} - 1000}{1000}$$

[Back to Index](#)



In-Circuit Test Mode

Wavelength vs Frequency

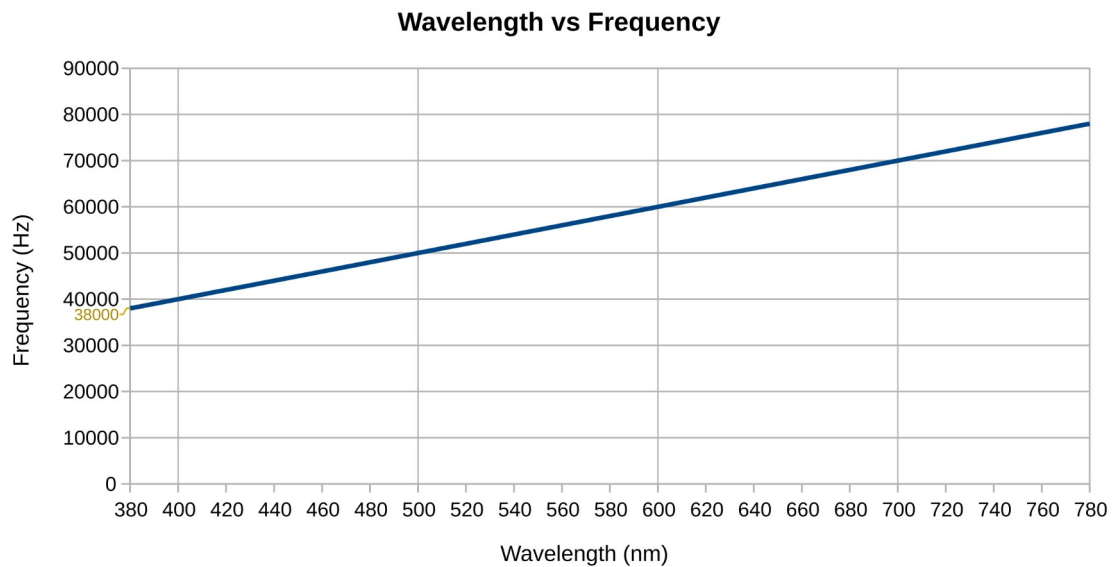


Figure 10c.

$$Wavelength = \frac{Frequency_{Wavelength}}{100}$$

[Back to Index](#)



In-Circuit Test Mode

CIE1931 vs Frequency

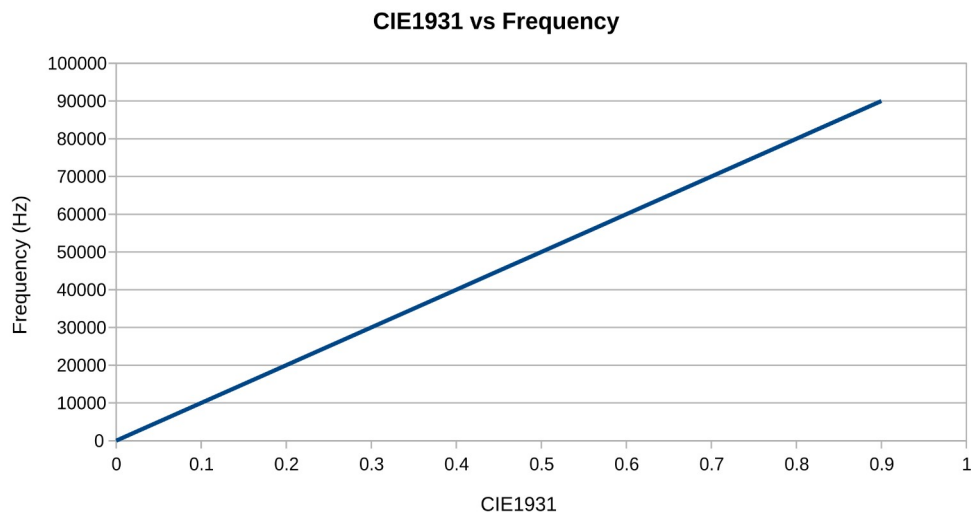


Figure 10d.

$$CIE_{xy} = \frac{Frequency_{CIE_{xy}}}{100000}$$

[Back to Index](#)



In-Circuit Test Mode

In-Circuit Test Frequency Mode

This mode is the easiest to implement at In-Circuit Test provided a method is available to measure the output frequencies. The frequencies range from 1KHz to 100KHz and the output is a square wave signal with a voltage swing from 0V to 3V.

The following sequence must be used to test the LED's successfully:-

Setup: Using Terminal Window and Serial port on the Led Analyser, Set the relevant Frequency mode, HSI or WSI or xyI and ensure Sync Serial Mode is disabled.

- 1/ Turn on the LED's to be tested.
- 2/ Apply power to the LED Analyser.
- 3/ Initialize the Analyser.
- 4/ Capture the LED Data.
- 5/ Read and Test the Hue (colour) or Dominant Wavelength or x Chromaticity Frequency.
- 6/ Read and Test the Saturation or y Chromaticity Frequency.
- 7/ Read and Test the Intensity Frequency.

The Analyser will measure the data from all the LED's in one capture operation. However, the data for each LED is read out one at the time. The data from last capture is stored in the Analyser until a new capture is done or power is removed.

Each Fiber or LED has an Address associated with it. Since there can be up to 20 LED's to be tested - Hexadecimal Addresses 0x01 – 0x14 are used to access the data for each individual LED. LED1 will have address 0x01, LED2 will have address 0x02, LED10 will have address 0x0A and LED20 will have address 0x14 etc.

Since the Analyser can test LED's that are dim to very bright - the correct intensity range should be used. There are five ranges available – *Low*, *Medium*, *High*, *Super* and *Ultra*. The low range is for dim LED's and the ultra is used for the very bright LED's. There is also an *auto* range and should only be used when both dim and bright LED's must be tested together. PWM Leds are captured in the same manner except pin 8 is driven Low.

For Very Bright Leds the Feasa x-IB High Brightness Analyser is recommended.

Figure 11 shows all the addresses that can be input to the LED Analyser and the function of each address.

These functions will be explained in the following sections.

FEASA LED ANALYSER ICT VERSION



ADDRESS	LED Analyser Function
0000	Capture & Capture PWM using the Automatic Range
0001	Address of Fiber 1
0010	Address of Fiber 2
0011	Address of Fiber 3
0100	Address of Fiber 4
0101	Address of Fiber 5
0110	Address of Fiber 6
0111	Address of Fiber 7
1000	Address of Fiber 8
1001	Address of Fiber 9
1010	Address of Fiber 10
1011	Address of Fiber 11
1100	Address of Fiber 12
1101	Address of Fiber 13
1110	Address of Fiber 14
1111	Address of Fiber 15
10000	Address of Fiber 16
10001	Address of Fiber 17
10010	Address of Fiber 18
10011	Address of Fiber 19
10100	Address of Fiber 20
10101	Capture & CapturePWM using Range1 (<i>Low Intensity</i>)
10110	Capture & CapturePWM using Range2 (<i>Medium Intensity</i>)
10111	Capture & CapturePWM using Range3 (<i>High Intensity</i>)
11000	Capture & CapturePWM using Range4 (<i>Super Intensity</i>)
11001	Capture & CapturePWM using Range5 (<i>Ultra Intensity</i>)
11010	reserved
11011	reserved
11100	Decrement the Exposure Factor by 1 (minimum = 1)
11101	Reset the Exposure Factor to 1
11110	Increment the Exposure Factor by 1 (Maximum = 15)
11111	Initialize the LED Analyser

Figure 11.

[Back to Index](#)



In-Circuit Test Mode

Initialize the LED Analyser for Capture

This function will initialize the LED Analyser and make it ready to capture new LED Test data. This function should be executed before each capture cycle.

Figure 12 shows the timing for this function.

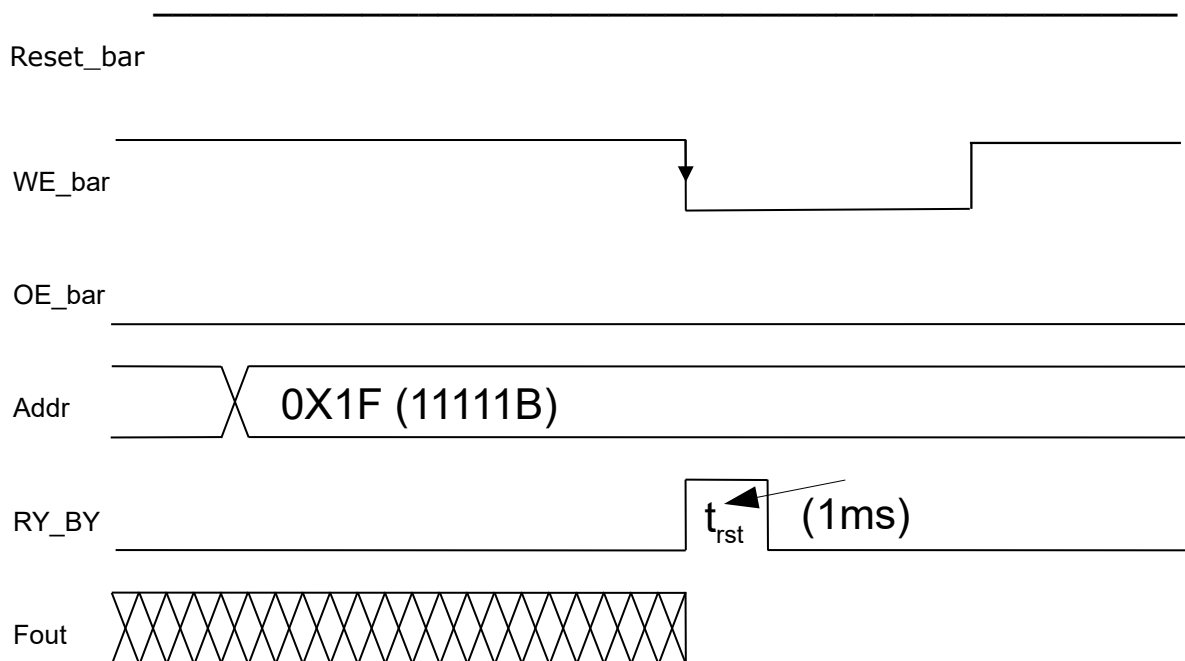


Figure 12.

- **NOTE: Logic Level Voltages 3.3V**

The timing sequence is as follows:-

- 1/ Drive Reset (pin 5) High, Drive WE_bar (pin 14) High and drive OE_bar (pin 6) Low. The RyBy_bar (pin18) will be driven low by the Analyser.
- 2/ Drive the address bus to 11111 (0x1F) (pins 9-13), This will Initialize the Analyser
- 3/ Drive WE_bar Low and the Analyser will drive RyBy_bar high for approx 1mS. All the frequencies (from the previous capture) will now be turned off at this time. After the time $T_{rst} = 1\text{mSec}$, RyBy_bar will be driven low by the Analyser to indicate it is ready for the next function. The Analyser is now Initialized.
- 4/ Drive WE_bar High any time after RyBy_bar has gone Low.

[Back to Index](#)



In-Circuit Test Mode

Set the Sensitivity Level of the Analyser (setfactor)

The Sensitivity of the Analyser can be adjusted to allow longer exposure times. This is useful when testing dim LED's. There are 15 adjustment steps numbered 1 – 15.

Number 1 has the shortest exposure and number 15 will have the longest. The Factory default value is 1.

Once programmed the Sensitivity Factor remains unchanged even when power is removed.

ONLY RUN THIS TEST DURING INITIAL DEBUG. WHEN SATISFIED WITH THE RESULTS, COMMENT OUT THIS TEST AS THERE IS A LIMIT TO THE NUMBER OF TIMES ONE CAN WRITE TO THE ON BOARD FLASH BEFORE YOU MAY CORRUPT THE DEVICE.



In-Circuit Test Mode

Capture LED Data (HSI or WSI or xyI) Static LED's

This function will capture the Hue, Saturation and Intensity of all the LED's and store this data in memory ready for output. The timing for this function is shown in Figure 13a.

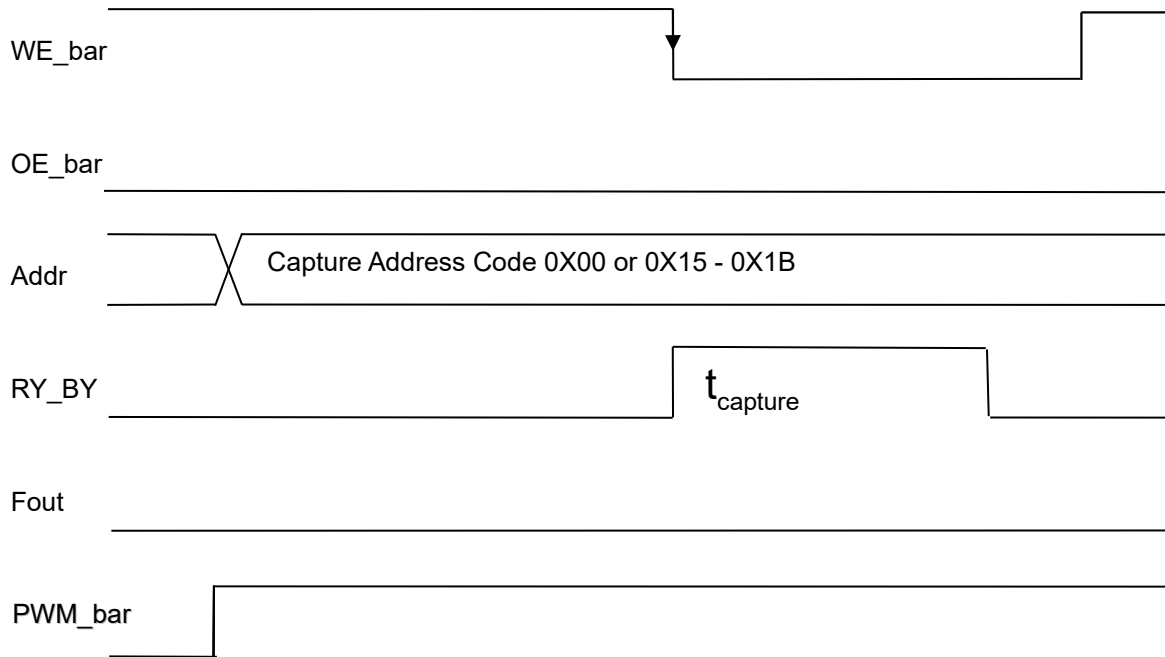


Figure 13a.

- **NOTE: Logic Level Voltages 3.3V**

The timing sequence is as follows:-

- 1/ Drive WE_bar High, PWM_bar High and OE_bar Low. RyBy_bar (pin18) will be driven low by the Analyser. (**note PWM_bar has a pullup Res on the board so its optional to drive pin 8 high or not)
- 2/ Select the Capture Range you require (Figure 11) and Drive the address bus to this code. *eg.(00000) for Auto Capture*
- 3/ Drive WE_bar is Low and the Analyser will drive RyBy_bar high. All the frequencies (from the previous capture) will be turned off at this time. After the time T_{capture} RyBy_bar will be driven low by the Analyser to indicate it is ready for the next function. The time for T_{capture} will vary depending on the range selected. The test system should loop on the RyBy_bar signal and wait for it to go low.
- 4/ Drive WE_bar High any time after RyBy_bar has gone Low.

[Back to Index](#)



In-Circuit Test Mode

Capture LED Data (HSI or WSI or xyI) PWM LED's

This function will capture the Frequency Data of all the LED's that are controlled by Pulse Width Modulation (PWM) and store this data in memory ready for output. PWM LED's are switched on and off rapidly to control the LED Intensity. The PWM_bar (pin 8) signal is pulled up on the Analyser. The timing for this function is shown in Figure 14b.

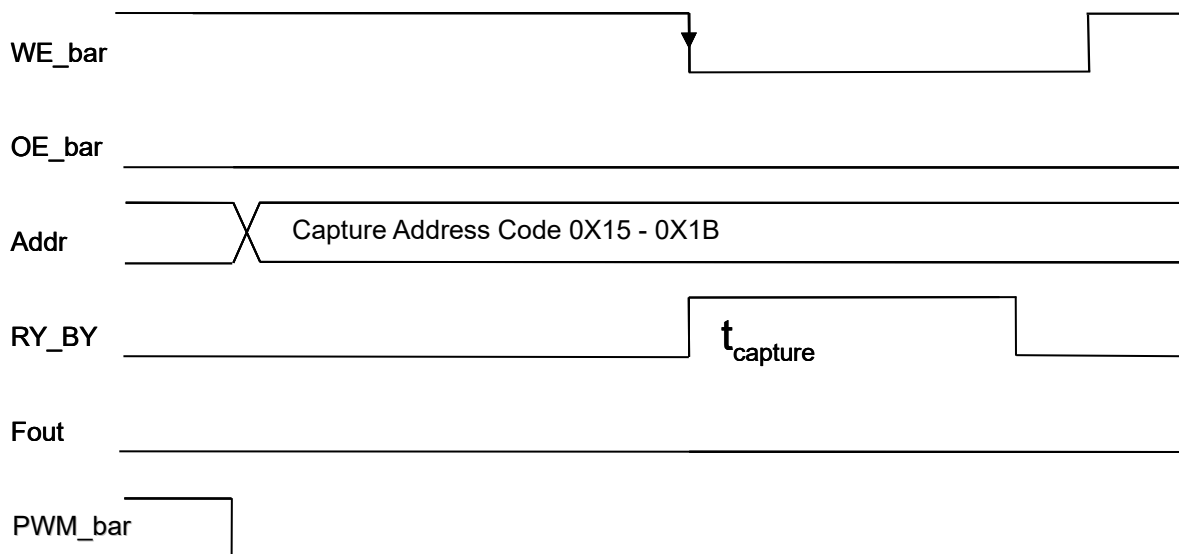


Figure 14b.

- **NOTE: Logic Level Voltages 3.3V**

The timing sequence is as follows:-

- 1/ Drive WE_bar High, Drive OE_bar and PWM_bar Low. RyBy_bar (pin18) will be driven low by the Analyser.
- 2/ Select the Capture Range you require (Figure 11) and Drive the address bus to this code. eg. (00000) for Auto Capture
- 3/ Drive WE_bar Low and the Analyser will drive RyBy_bar High. All the frequencies (from the previous capture) will be turned off at this time. After the time $T_{capture}$ RyBy_bar will be driven low by the Analyser to indicate it is ready for the next function. The time for $T_{capture}$ will vary depending on the range selected. The test system should loop on the RyBy_bar signal and wait for it to go low.
- 4/ Drive WE_bar High any time after RyBy_bar has gone Low.

[Back to Index](#)



In-Circuit Test Mode

Test LED Data (HSI or WSI or xyI)

This function is used to Read Back the Hue, Saturation, Dominant Wavelength, xy Chromaticity & Intensity of a LED. Three frequencies representing these properties are output. Each LED has a Fiber Address which must be input to the Analyser. A capture must be executed prior to this function to ensure valid data is stored in the Analyser.

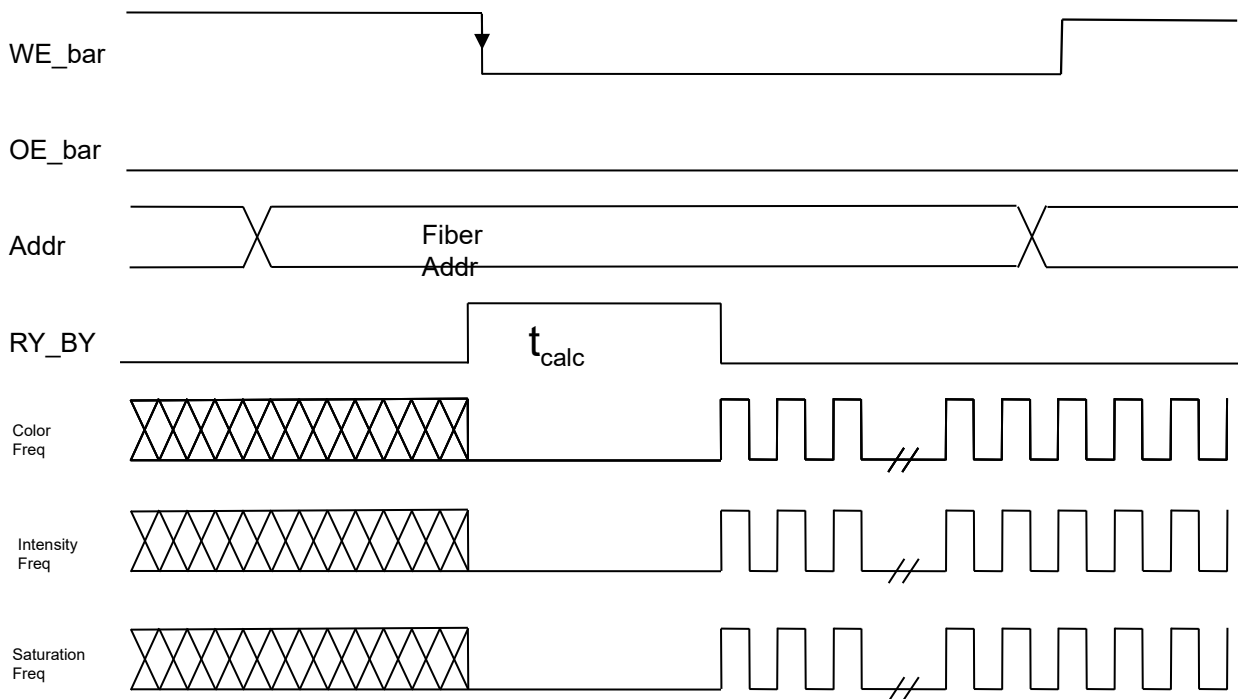


Figure 15.

- **NOTE: Logic Level Voltages 3.3V**

The timing sequence is as follows:-

- 1/ Drive WE_bar High and drive OE_bar Low. RyBy_bar (pin18) will be driven low by the Analyser.
- 2/ Select the Fiber Number you require (Figure 11) and Drive the address bus to this code. *eg.(0001) for Fiber 1*
- 3/ Drive WE_bar Low and the Analyser will drive RyBy_bar High. The Intensity (pin 15), Hue or Wavelength or x chromaticity (pin 16) and Saturation or y Chromaticity (pin 17) frequencies will be output on the ICT connector after the time T_{calc} . RyBy_bar will be driven low by the Analyser to indicate the outputs are valid. The time for T_{calc} is 200mSec.
- 4/ Drive WE_bar High any time after RyBy_bar has gone Low.

[Back to Index](#)



In-Circuit Test Mode

Step-by Step Sequence for LED Analyser Auto Capture & Read HSI or WSI or xyI

Ensure the Led Analyser is in the correct Frequency mode (HSI, WSI or xyI) prior to test. Use the command `getictstatus` to verify.

1. Initialize Function/Reset

Initialize all inputs to the following logic levels

```
set Reset to "1"  
set WE_bar to "1"  
set OE_bar to "0"  
set Addr_0 to "1"  
set Addr_1 to "1"  
set Addr_2 to "1"  
set Addr_3 to "1"  
set Addr_4 to "1"
```

set WE_bar to "0" ! This executes the reset command

Wait for RyBy_bar_bar to go to Logic "1"

Wait for RyBy_bar_bar to go to Logic "0" ! If RyBy_bar goes high & then low the **initialize** was good.

set WE_bar to "1"

2. /Auto Capture Function

! Auto Range Capture Address 00000 00 dec (350ms)

! Manual Range #1 Capture Address 10101 21 dec (650ms)

! Manual Range #2 Capture Address 10110 22 dec (200ms)

! Manual Range #3 Capture Address 10111 23 dec (22ms)

! Manual Range #4 Capture Address 11000 24 dec (4ms)

! Manual Range #5 Capture Address 11001 25 dec (2ms)

Hold all logic levels as per the end of the reset function

```
set Addr_0 to "0"  
set Addr_1 to "0"  
set Addr_2 to "0"  
set Addr_3 to "0"  
set Addr_4 to "0" ! Address 00000 is AUTO CAPTURE
```

set WE_bar to "0"

Wait for RyBy_bar to go to Logic "1"

Wait for RyBy_bar to go to Logic "0" ! If RyBy_bar goes high & then low the **capture** command was good.

set WE_bar to "1"

FEASA LED ANALYSER

ICT VERSION



3. Read the Hue(Colour), Saturation and Intensity or the Dominant Wavelength, Saturation and Intensity or the xy chromaticity, Intensity of Fiber Number 1

Hold all logic levels as per the end of the Capture function

set Addr_0 to "1"
set Addr_1 to "0"
set Addr_2 to "0"
set Addr_3 to "0"
set Addr_4 to "0" ! Select Fiber 01

set WE_bar to "0"

Wait for RyBy_bar to go to Logic "1"

Wait for RyBy_bar to go to Logic "0" ! If RyBy_bar goes high & then low the **Read** command was good.

Measure the frequencies on pins 15 (Intensity), 16 (Hue / Wavelength / X) and 17 (Saturation / Y)

set WE_bar to "1"

4. Read the Hue(Colour), Saturation and Intensity or the Dominant Wavelength, Saturation and Intensity or the xy chromaticity, Intensity of Fiber number 2

Hold all logic levels as per the end of the Capture function

set Addr_0 to "0"
set Addr_1 to "1"
set Addr_2 to "0"
set Addr_3 to "0"
set Addr_4 to "0" ! Select Fiber 02

set WE_bar to "0"

Wait for RyBy_bar to go to Logic "1"

Wait for RyBy_bar to go to Logic "0" ! If RyBy_bar goes high & then low the **Read** command was good.

Measure the frequencies on pins 15 (Intensity), 16 (Hue / Wavelength / X) and 17 (Saturation / Y)

set WE_bar to "1"

Repeat step 3 or 4 above steps for fibers 3 to 20

[Back to Index](#)



In-Circuit Test Mode

Capture Times ICT Mode – Static Leds

The Capture Time for each range is outlined on the following Table.

Range	Capture Time
Range 1 low	650ms
Range 2 medium	200ms
Range 3 high	22ms
Range 4 super	4ms
Range 5 ultra	2ms
Auto Range	350ms

Figure 16.

These are the maximum Capture Times for each range and verified using the Exposure Factor set at 01.

[Back to Index](#)



Synchronous Serial Mode

In Circuit Test Synchronous Serial Mode

The LED Analyser can be controlled using a Synchronous Serial bus, through which commands can be sent and the results read back for testing LED's. The Feasa ICT Analyser (Firmware Version i210 upwards) is shipped with Synchronous Serial Mode disabled. This mode can be ENABLED using the command [setsyncserial1](#).

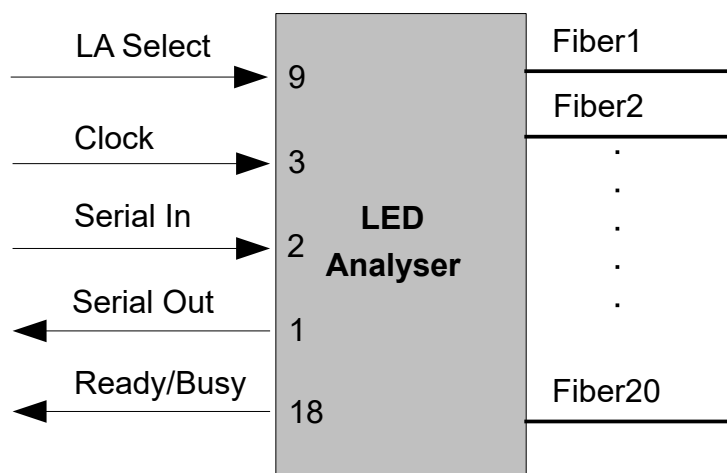


Figure 17.

Figure 17 shows a block diagram of the Synchronous serial bus. There are 5 lines used to implement the Synchronous serial mode. These signals are described in Figure 18.

Pin	Type	ICT Signal Name	Function
9	Input	Addr0/ LA_Select	Used to select the Sync Serial Mode on the Led Analyser
1	Output	Ser_Out	Synchronous Serial Output
2	Input	Ser_In	Synchronous_Serial Input
3	Input	Ser_Clk	Synchronous Serial Clock
18	Output	RyBy_bar	Ready Busy Status output

- **NOTE:** Logic Level Voltages 3.3V

Figure 18.

[Back to Index](#)



Synchronous Serial Mode

Synchronous Serial Pin Descriptions

Pin 1 Serial Out

This is the Synchronous Serial Data output pin. The Synchronous Serial is a 5-pin data bus designed for high-speed data transfer between the LED Analyser and the controlling computer. All data transfers are synchronized to the Serial Clock Pin 3.

Pin 2 Serial In

This is the Synchronous Serial Data input pin. The data on this pin is synchronized to the Serial Clock.

Pin 3 Serial Clock

This is the Synchronous Serial Clock input pin. All data transfers on pins 1 and 2 must be synchronized to this clock. Data is valid on the *falling* edge of this clock.

Pin 9 Addr0/LA_Select

Addr0 is used in the Synchronous Serial mode to select the Analyser (LA_Select). This signal is Active High.

Pin 18 RyBy_bar

This is the Ready/Busy output pin. It is used to indicate the status of the LED Analyser during operation. A high on this pin indicates the Analyser is busy and a low indicates the Analyser is ready for the next operation.

[Back to Index](#)



Synchronous Serial Mode

Synchronous Serial Commands

The following table lists the commands that can be executed through the Synchronous Serial Port. All data transmitted and received is in a binary format. The Most Significant Bit (MSB) is transmitted/received first. Another Table **Figure 29** contains other Sync Serial Commands for the Capture Sequence Testing Function.

Command	Data Transmitted	Data Received
Clear Comms Buffer	0xFF (1 byte)	
Initialize the LA	0xAA (1 byte)	
Capture Auto Mode	0x55	0xD5 or 0xEE (error condition)
Capture using Range 1	0xA1	0xD5 or 0xEE (error condition)
Capture using Range 2	0xA2	0xD5 or 0xEE (error condition)
Capture using Range 3	0xA3	0xD5 or 0xEE (error condition)
Capture using Range 4	0xA4	0xD5 or 0xEE (error condition)
Capture using Range 5	0xA5	0xD5 or 0xEE (error condition)
Capture Status	0xBB	0xD5 or 0xEE (error condition)
CapturePWM Auto Mode	0xB0	0xD5 or 0xEE (error condition)
CapturePWM using Range 1	0xB1	0xD5 or 0xEE (error condition)
CapturePWM using Range 2	0xB2	0xD5 or 0xEE (error condition)
CapturePWM using Range 3	0xB3	0xD5 or 0xEE (error condition)
CapturePWM using Range 4	0xB4	0xD5 or 0xEE (error condition)
CapturePWM using Range 5	0xB5	0xD5 or 0xEE (error condition)
Read HSI	0x20 + fiber num (1byte)	Hue(2bytes) + Sat(1byte) + Int (3byte)
Read RGB & I	0x30 + fiber num (1byte)	R(1byte) + G(1byte) +B(1byte) +Int (3byte)
Set Sensitivity Factor	0x40 + factor(0x01 - 0x0F)	factor(0x01 - 0x0F)
Read xy & I	0x50 + fiber num (1byte)	x(2bytes) + y(1byte) + Int (3byte)
Read cct	0x60 + fiber num (1byte)	cct (2bytes)
Read Dominant Wavelength	0x70 + fiber num (1byte)	Dominant Wavelength (2bytes)
Read Wavelength & Sat & Intensity	0x71 + fiber num (1byte)	Wavelength (2bytes) + Saturation (1byte) + Int (3bytes)
Read Absolute Intensity	0x80 + fiber num (1byte)	Int(3byte) + Exp(1byte)

Figure 19.

Note:- The code 0xD5 indicates the command completed correctly. If the code 0xEE is received then an error has occurred and the command should be re-transmitted.

[Back to Index](#)



Synchronous Serial Mode

Clear Comms Buffer

This command will clear the Communications Buffer of the Synchronous Serial Port of any data bits. The sequence for this command is shown in Figure 20.

Binary 11111111 (0xFF) is clocked into the LED Analyser while the Addr0/LA_Select signal is held low.

The command is clocked in on the falling edge of SCK.

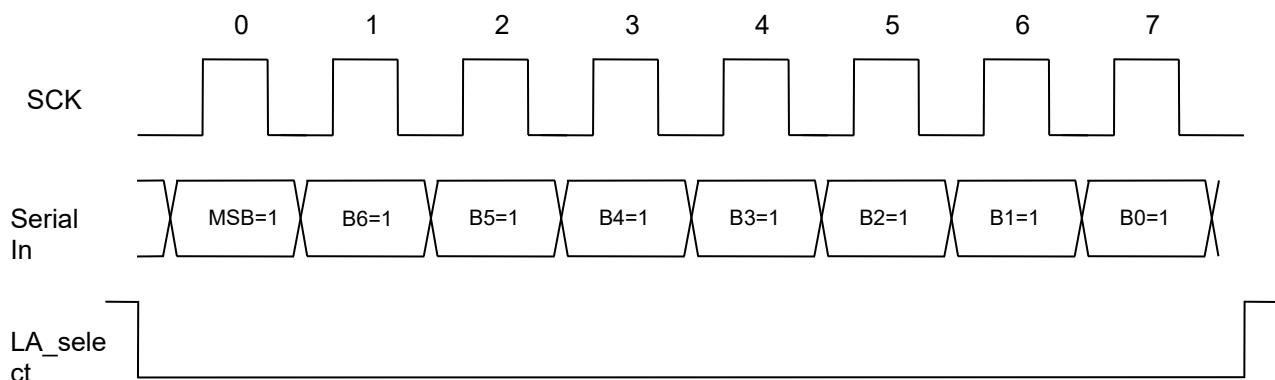


Figure 20.

The signal Addr0/LA_Select must be driven low while 0xFF is clocked into the Analyser.

For all other commands the signal Addr0/LA_Select should be driven high.

[Back to Index](#)



Synchronous Serial Mode

Set the Sensitivity Factor (setfactor)

The Sensitivity of the Analyser can be adjusted to allow longer exposure times. This is useful when testing dim LED's. There are 15 adjustment steps numbered 1 – 15.

Number 1 has the shortest exposure and number 15 will have the longest. The Factory default value is 1.

Once programmed the Sensitivity Factor remains unchanged even when power is removed.

ONLY RUN THIS TEST DURING INITIAL DEBUG. WHEN SATISFIED WITH THE RESULTS, COMMENT OUT THIS TEST AS THERE IS A LIMIT TO THE NUMBER OF TIMES ONE CAN WRITE TO THE ON BOARD FLASH BEFORE YOU MAY CORRUPT THE DEVICE.

Step-by Step Sequence to Set the Sensitivity Factor

1. **Initialize the board**
 - Set Addr0/LA_Select = 0
 - Set Clock_in = 0
 - Set Data_in to 0
2. **Clear the Comms Buffer**
 - Set Data_In to 1
 - Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
3. Wait 100u
4. **Send the Set Sensitivity Factor (0x40) command**
 - Set Data_In to 01000000 (0x40)
 - Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
5. **Send the Sensitivity Factor (0X01 to 0X0F) (Default 0x01)**
 - Set Data_In to [a number in the range 0x01-0x0F]
 - Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
 - RyBy_bar output goes high
6. Wait 50ms
 - RyBy_bar output goes low
7. **Read back the Sensitivity Factor set in Step 5**
 - Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
 - Data_Out = [[a number in the range 0x01-0x0F]
 - The value read back is the same as the Sensitivity factor programmed in step 5

[Back to Index](#)



Synchronous Serial Mode

Sync Serial Capture for Static LED's

Initialize the LED Analyser and make it ready for a *Capture* command. The sequence for this command is shown in Figure 21.

Binary 10101010 (0xAA) is clocked into the LED Analyser while the Addr0/LA_Select signal is held high.

The command is clocked in on the falling edge of SCK.

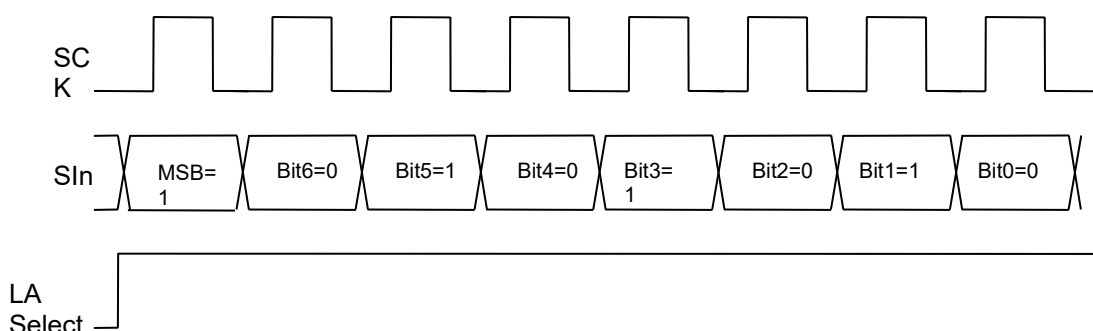


Figure 21.

The signal Addr0/LA_Select must be driven high while 0xAA is clocked into the Analyser.

This command will instruct the LED Analyser to capture and store all the LED Hue, Saturation and Intensity data. There are six Intensity ranges that can be specified to get the best results from the test. There are five manual ranges designated Range1 (Low), Range2 (Medium), Range3 (High), Range4 (Super) and Range5 (Ultra). Range1 has the highest sensitivity for low light conditions and the sensitivity is reduced progressively to Range5 which has the lowest sensitivity. This means that for brighter LED's the higher Ranges should be used.

The Sixth Range uses an automatic exposure and covers the full Intensity range of the Analyser. This should only be used when it is required to test very dim and very bright LED's together.

The command codes are shown in Figure 22.

Command	Data Transmitted	Data Received
Capture Auto Mode	0x55	0xD5 or 0xEE (error condition)
Capture using Range 1	0xA1	0xD5 or 0xEE (error condition)
Capture using Range 2	0xA2	0xD5 or 0xEE (error condition)
Capture using Range 3	0xA3	0xD5 or 0xEE (error condition)
Capture using Range 4	0xA4	0xD5 or 0xEE (error condition)
Capture using Range 5	0xA5	0xD5 or 0xEE (error condition)

Figure 22.

The code 0xD5 indicates a successful completion of the command. If an error occurs then the code 0xEE will be received. The command should then be re-transmitted.

[Back to Index](#)



Synchronous Serial Mode

Step-by Step Sequence for LED Analyser Capture Static LED's

1. Initialize the board

- Set Addr0/LA_Select = 0
- Set Clock_in = 0
- Set Data_in to 0

2. Clear Comms Buffer

- Set Data_In to 1
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit

3. Set Data_in to 0

4. Set Addr0/LA_Select to 1

- Wait until RyBy_bar goes to 0

5. Initialize the capture

- Set Data_in to 10101010 (0XAA)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
- Wait 1ms
- Wait until RyBy_bar goes to 0
- RyBy_bar goes to 1

6. Send one capture command from the following list

- (0X55) Auto Range
- (0XA1) Manual Range #1 (Low)
- (0XA2) Manual Range #2 (Medium)
- (0XA3) Manual Range #3 (High)
- (0XA4) Manual Range #4 (Super High)
- (0XA5) Manual Range #5 (Ultra)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit

7. Read back the Capture Acknowledge as follows

- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
- Read back 0xD5 (8 bits after each falling edge of the clock)

8. Check that RyBy_bar is high

9 Check that RyBy_bar goes low after the capture time.

10. Send the Capture Status command (0xBB)

- Set Data_in to (0xBB) (10111011)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit



Synchronous Serial Mode

Step-by Step Sequence for LED Analyser Capture con'td

11. Receive Response

- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
- Read back 8 bits after each falling edge of the clock
- (0xD5 is positive response and 0xEE is error response)

12. Check that RyBy_bar is low

Capture Times Synchronous Mode – Static Leds

Range	Capture Time
Range 1 low	650ms
Range 2 medium	200ms
Range 3 high	22ms
Range 4 super	4ms
Range 5 ultra	2ms
Auto Range	350ms

These are the maximum Capture Times for each range and verified using the Exposure Factor set at 01.

[Back to Index](#)



Synchronous Serial Mode

Sync Serial Capture for PWM LED's

Pulse-Width-Modulated(PWM) LED's are switched on and off rapidly to save power and to control Intensity. The Analyser contains a special set of commands for testing PWM LED's. There are five Intensity ranges that can be specified to get the best results from the test. The ranges are designated Range1, Range2, Range3, Range4 and Range5. Range1 has the highest sensitivity for low light conditions. The sensitivity is reduced progressively to Range5 which has the lowest sensitivity. This means that for brighter LED's the higher Ranges should be used.

The command codes are shown in Figure 23.

Command	Data Transmitted	Data Received
CPWM	0xB0 + byte2	0xD5 or 0xEE (error condition)
C1PWM01 - C1PWM15	0xB1 + byte2	0xD5 or 0xEE (error condition)
C2PWM01 - C2PWM15	0xB2 + byte2	0xD5 or 0xEE (error condition)
C3PWM01 - C3PWM15	0xB3 + byte2	0xD5 or 0xEE (error condition)
C4PWM01 - C4PWM15	0xB4 + byte2	0xD5 or 0xEE (error condition)
C5PWM01 - C5PWM15	0xB5 + byte2	0xD5 or 0xEE (error condition)

Figure 23.

The data in byte2 is a hexadecimal number in the range 0x01 – 0x0F. This number represents the *Averaging Factor* for PWM testing. The smaller the value the faster the test will be to execute. However, this value should be derived experimentally to determine the best compromise between speed and stability.

It is recommended to use an initial value of 0x07.

[Back to Index](#)



Synchronous Serial Mode

Step-by-Step Sequence for LED Analyser Capture for PWM LED's

1. **Initialize the board**
 - Set Addr0/LA_Select = 0
 - Set Clock_in = 0
 - Set Data_in to 0
2. **Clear Comms Buffer**
 - Set Data_In to 1
 - Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
3. **Set Data_in to 0**
4. **Set Addr0/LA_Select to 1**
 - Data_Out goes to 0
 - RyBy_bar goes to 0
5. **Wait 1m**
6. **Initialize the capture**
 - Set Data_in to 10101010 (0XAA)
 - Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
 - Data_out goes to 1
 - RyBy_bar goes to 1
7. **Send one capture command from the following list**
 - (0xB1) PWM Manual Range #1 (Low)
 - (0xB2) PWM Manual Range #2 (Medium)
 - (0xB3) PWM Manual Range #3 (High)
 - (0xB4) PWM Manual Range #4 (Super High)
 - (0xB5) PWM Manual Range #5 (Ultra)
 - Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
8. **Send the PWM Averaging Time as follows**
 - Set Data_in to one value between (0x01 to 0x0F) (Default 0x07)
 - Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
9. **Read back the Capture Acknowledge as follows**
 - Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
 - Read back 8 bits after each falling edge of the clock
 - (0xD5 is positive response) (0xEE is an error response)



Synchronous Serial Mode

10. Check that RyBy_bar is high
11. Check that RyBy_bar goes low after the capture time
12. **Send the Capture Status command (0xBB)**
 - Set Data_in to (0xBB) (10111011)
 - Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
13. **Receive Response**
 - Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
 - Read back 8 bits after each falling edge of the clock
 - (0xD5 is positive response and 0xEE is an error response)

Capture Times Synchronous Mode – PWM Leds

Range	Capture Time
Range 1 low	9.60Sec
Range 2 medium	6.4Sec
Range 3 high	3.3Sec
Range 4 super	932mSec
Range 5 ultra	740mSec

These are the maximum Capture Times for each range and verified using the Exposure Factor set at 01.

Note: These PWM Capture times are based on an averaging factor of 07.

[Back to Index](#)



Synchronous Serial Mode

Capture / CPWM Command Timing Diagram

Transmit Relevant Command Code to LED Analyser

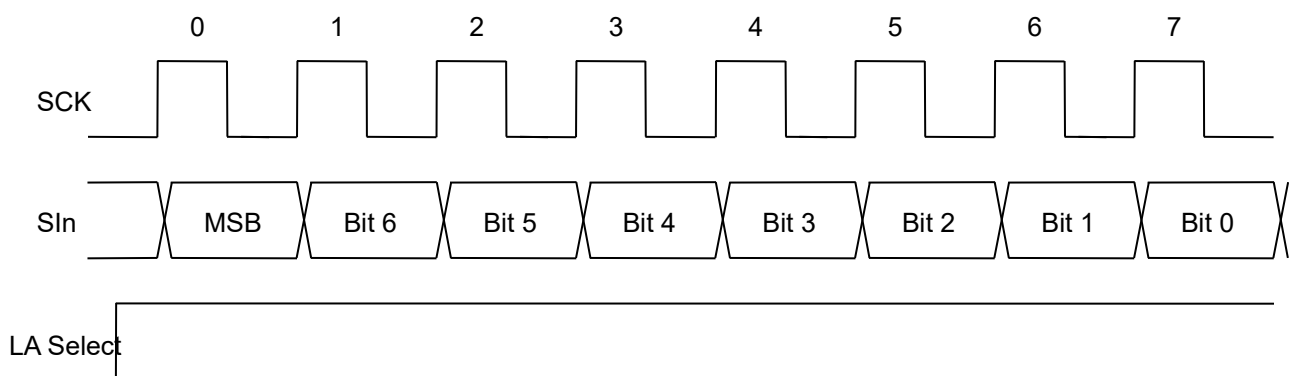


Figure 24.

Receive Status Code from LED Analyser

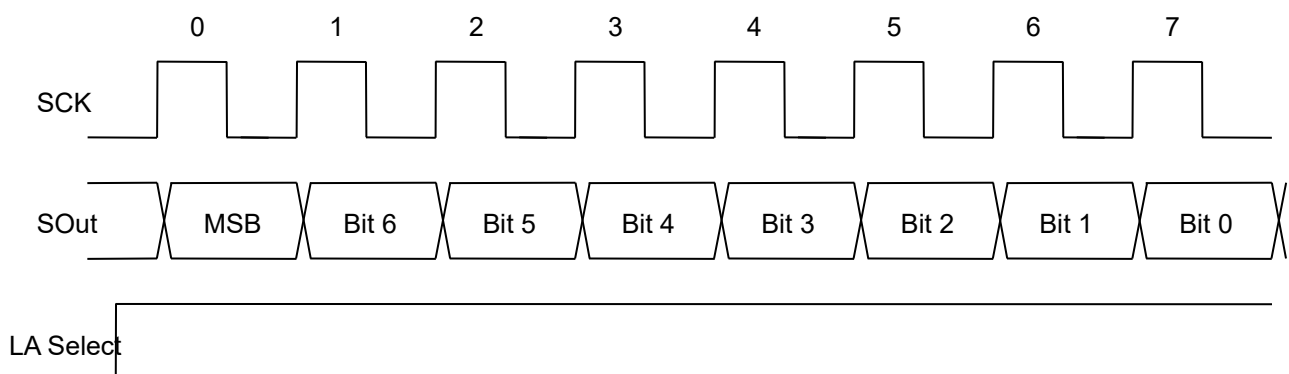


Figure 25.

[Back to Index](#)



Synchronous Serial Mode

Read Hue, Saturation and Intensity (HSI)

This command will instruct the LED Analyser to output the Hue, Saturation and Intensity of a LED. All the LED data should have been previously captured.

The code for this command is 0x20 – this must be transmitted to the LED Analyser followed by the Fiber Number of the LED to be tested. The Fiber Number is a hexadecimal number in the range 0x01 – 0x14 (in decimal this represents numbers 1 -20). The Analyser will return 6 bytes of data – 2 bytes for Hue, 1 byte for Saturation and 3 bytes for Intensity.

Command	Data Transmitted	Data Received
Read HSI	0x20 + fiber num (1byte)	Hue(2bytes) + Sat(1byte) + Int(3byte)

To convert the Hue data to a decimal number use the formula:-

$$\text{Hue} = (\text{byte1} * 256 + \text{byte2}) / 100$$

The Saturation value will be in the 3rd byte and will represent a number in the range 0 – 100.

$$\text{Saturation} = (\text{byte3})$$

The Intensity value is output in the last 3 bytes with the most significant byte first.

To convert the Intensity data to a decimal number use the formula:-

$$\text{Intensity} = (\text{byte4} * 65536) + (\text{byte5} * 256) + \text{byte6}$$

Note:- The data output in all the bytes are hexadecimal numbers.

[Back to Index](#)



Synchronous Serial Mode

Step-by Step Sequence to Read Hue, Saturation and Intensity (HSI)

1. Initialize the board

- Set Addr0/LA_Select = 0
- Set Clock_in = 0
- Set Data_in to 0

2. Clear the Comms Buffer

- Set Data_In to 1
- Set Clock_in to 1 and 0 [8] times to clock in each bit

3. Send the command to read HSI as follows

- Set Data_In to 0X20 (00100000)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit

4. Send Fiber Number to Read

- Send (0x01 to 0x14) (1-20 decimal)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit

5. Read back 6 bytes of data

- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Hue** high byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Hue** low byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Saturation** byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Intensity** high byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Intensity** middle byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Intensity** low byte]

[Back to Index](#)



Synchronous Serial Mode

Read RGB and Intensity (RGBI)

This command will instruct the LED Analyser to output the RGB colours and Intensity of a LED. All the LED data should have been previously captured.

The code for this command is 0x30 – this must be transmitted to the LED Analyser followed by the Fiber Number of the LED to be tested. The Fiber Number is a hexadecimal number in the range 0x01 – 0x14 (in decimal this represents numbers 1 -20). The Analyser will return 6 bytes of data – 1 byte each for Red, Green, Blue and 3 bytes for Intensity.

Command	Data Transmitted	Data Received
Read RGB + Intensity	0x30 + fiber num (1byte)	R(1byte)+G(1byte)+B(1byte) + Int(3byte)

Red = (byte1)

Green = (byte2)

Blue = (byte3)

The Intensity value is output in the last 3 bytes with the most significant byte first.

To convert the Intensity data to a decimal number use the formula:-

$$\text{Intensity} = (\text{byte4} * 65536) + (\text{byte5} * 256) + \text{byte6}$$

Note:- The data output in all the bytes are hexadecimal numbers.

[Back to Index](#)



Synchronous Serial Mode

Step-by Step Sequence to Read RGB and Intensity (RGBI)

1. Initialize the board

- Set Addr0/LA_Select = 0
- Set Clock_in = 0
- Set Data_in to 0

2. Clear the Comms Buffer

- Set Data_In to 1
- Set Clock_in to 1 and 0 [8] times to clock in each bit

3. Send the command to read RGBI as follows

- Set Data_In to 0X30 (00110000)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit

4. Send Fiber Number to Read

- Send (0x01 to 0x14) (1-20 decimal)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit

5. Read back 6 bytes of data

- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Red** byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Green** byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Blue** byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Intensity** high byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Intensity** middle byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Intensity** low byte]

[Back to Index](#)



Synchronous Serial Mode

Dominant Wavelength (W)

This command will instruct the LED Analyser to output the Dominant Wavelength of a LED. All the LED data should have been previously captured.

The code for this command is 0x70 – this must be transmitted to the LED Analyser followed by the Fiber Number of the LED to be tested. The Fiber Number is a hexadecimal number in the range 0x01 – 0x14 (in decimal this represents numbers 1 -20). The Analyser will return 2 bytes of data – for Dominant Wavelength.

Command	Data Transmitted	Data Received
Read Wavelength	0x70 + fiber num (1byte)	Dominant Wavelength (2bytes)

To convert the Dominant Wavelength data to a decimal number use the formula:-

$$\text{Wavelength} = (\text{byte1} * 256 + \text{byte2})$$

Note:- The data output in all the bytes are hexadecimal numbers.

[Back to Index](#)



Synchronous Serial Mode

Step-by Step Sequence to Read Dominant Wavelength (W)

Initialize the board

- Set Addr0/LA_Select = 0
- Set Clock_in = 0
- Set Data_in to 0

2. Clear the Comms Buffer

- Set Data_In to 1
- Set Clock_in to 1 and 0 [8] times to clock in each bit

3. Send the command to read Dominant Wavelength as follows

- Set Data_In to 0X70 (01110000)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit

4. Send Fiber Number to Read

- Send (0x01 to 0x14) (1-20 decimal)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit

5. Read back 2 bytes of data

- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Dominant Wavelength** high byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Dominant Wavelength** low byte]

[Back to Index](#)



Synchronous Serial Mode

Wavelength & Saturation & Intensity (WSI)

This command will instruct the LED Analyser to output the Wavelength, Saturation and Intensity of a LED. All the LED data should have been previously captured.

The code for this command is 0x71 – this must be transmitted to the LED Analyser followed by the Fiber Number of the LED to be tested. The Fiber Number is a hexadecimal number in the range 0x01 – 0x14 (in decimal this represents numbers 1 -20). The Analyser will return 2 bytes of data – for Dominant Wavelength.

Command	Data Transmitted	Data Received
Read Wave, Sat, Int	0x71 + fiber num (1byte)	Wavelength (2bytes), Saturation (1byte), Intensity (3bytes)

To convert the Dominant Wavelength data to a decimal number use the formula:-

$$\text{Wavelength} = (\text{byte1} * 256 + \text{byte2})$$

$$\text{Saturation} = (\text{byte3})$$

The Intensity value is output in the last 3 bytes with the most significant byte first.

To convert the Intensity data to a decimal number use the formula:-

$$\text{Intensity} = (\text{byte4} * 65536) + (\text{byte5} * 256) + \text{byte6}$$

Note:- The data output in all the bytes are hexadecimal numbers.

[Back to Index](#)



Synchronous Serial Mode

Step-by Step Sequence to Read Wavelength, Saturation, Intensity (WSI)

Initialize the board

- Set Addr0/LA_Select = 0
- Set Clock_in = 0
- Set Data_in to 0

2. Clear the Comms Buffer

- Set Data_In to 1
- Set Clock_in to 1 and 0 [8] times to clock in each bit

3. Send the command to read Wavelength, Saturation & Intensity as follows

- Set Data_In to 0X71 (01110000)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit

4. Send Fiber Number to Read

- Send (0x01 to 0x14) (1-20 decimal)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit

5. Read back 6 bytes of data

- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Dominant Wavelength** high byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Dominant Wavelength** low byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Saturation** byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Intensity** high byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Intensity** middle byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Intensity** low byte]

[Back to Index](#)



Synchronous Serial Mode

Colour Temperature (cct)

This command will instruct the LED Analyser to output the colour temperature of a LED. All the LED data should have been previously captured.

CCT data is NOT Valid when measuring pure colour Leds (Red, Green, Blue). The SyncSerial output from the Analyser will report 0 for CCT and delta uv is not reported.

The code for this command is 0x60 – this must be transmitted to the LED Analyser followed by the Fiber Number of the LED to be tested. The Fiber Number is a hexadecimal number in the range 0x01 – 0x14 (in decimal this represents numbers 1 -20). The Analyser will return 2 bytes of data – for cct.

Command	Data Transmitted	Data Received
Read cct	0x60 + fiber num (1byte)	Cct (2bytes)

To convert the cct data to a decimal number use the formula:-

$$\text{cct} = (\text{byte1} * 256 + \text{byte2})$$

Note:- The data output in all the bytes are hexadecimal numbers.

[Back to Index](#)



Synchronous Serial Mode

Step-by Step Sequence to Read colour Temperature (cct)

Initialize the board

- Set Addr0/LA_Select = 0
- Set Clock_in = 0
- Set Data_in to 0

2. Clear the Comms Buffer

- Set Data_In to 1
- Set Clock_in to 1 and 0 [8] times to clock in each bit

3. Send the command to read cct as follows

- Set Data_In to 0X60 (01100000)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit

4. Send Fiber Number to Read

- Send (0x01 to 0x14) (1-20 decimal)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit

5. Read back 2 bytes of data

- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [cct high byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [cct low byte]

[Back to Index](#)



Synchronous Serial Mode

x-chromaticity, y-chromaticity & Intensity (xyi)

This command will instruct the LED Analyser to output the xy and Intensity of a LED. All the LED data should have been previously captured.

The code for this command is 0x50 – this must be transmitted to the LED Analyser followed by the Fiber Number of the LED to be tested. The Fiber Number is a hexadecimal number in the range 0x01 – 0x14 (in decimal this represents numbers 1 -20). The Analyser will return 7 bytes of data – 2 bytes for x, 2 bytes for y and 3 bytes for Intensity.

Command	Data Transmitted	Data Received
Read xy an Intensity	0x50 + fiber num (1byte)	x(2bytes) + y(2byte) + Int(3byte)

To convert the x data to a decimal number use the formula:-

$$x = (\text{byte1} * 256 + \text{byte2}) / 10,000$$

To convert the y data to a decimal number use the formula:-

$$y = (\text{byte3} * 256 + \text{byte4}) / 10,000$$

The Intensity value is output in the last 3 bytes with the most significant byte first.

To convert the Intensity data to a decimal number use the formula:-

$$\text{Intensity} = (\text{byte5} * 65536) + (\text{byte6} * 256) + \text{byte7}$$

Note:- The data output in all the bytes are hexadecimal numbers.

[Back to Index](#)



Synchronous Serial Mode

Step-by Step Sequence to Read xy & Intensity (xyi)

Initialize the board

- Set Addr0/LA_Select = 0
- Set Clock_in = 0
- Set Data_in to 0

2. Clear the Comms Buffer

- Set Data_In to 1
- Set Clock_in to 1 and 0 [8] times to clock in each bit

3. Send the command to read xy & Intensity as follows

- Set Data_In to 0X50 (01010000)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit

4. Send Fiber Number to Read

- Send (0x01 to 0x14) (1-20 decimal)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit

5. Read back 7 bytes of data

- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**x** high byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**x** low byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**y** high byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**y** low byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Intensity** high byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Intensity** middle byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Intensity** low byte]

[Back to Index](#)



Synchronous Serial Mode

Absolute Intensity

This command will instruct the LED Analyser to output the Absolute Intensity of a LED. All the LED data should have been previously captured.

The code for this command is 0x80 – this must be transmitted to the LED Analyser followed by the Fiber Number of the LED to be tested. The Fiber Number is a hexadecimal number in the range 0x01 – 0x14 (in decimal this represents numbers 1 -20). The Analyser will return 4 bytes of data - 3 for Intensity and 1 for Exponent.

Command	Data Transmitted	Data Received
Read Absolute Intensity	0x80 + fiber num (1byte)	Int(3byte) + Exp(1byte)

To convert the Absolute Intensity data to a decimal number use the formula:-

$$\text{Absolute Intensity} = (\text{byte1} * 65536) + (\text{byte2} * 256) + \text{byte3}$$

To convert the Exponent data to a decimal number use the formula:-

$$\text{Exp} = \text{byte4}$$

Note:- The data output in all the bytes are hexadecimal numbers.

$$\text{Absolute Intensity} = ((\text{byte1} * 65536) + (\text{byte2} * 256) + \text{byte3}) \times \text{Exp } 10^{(+/- \text{byte4})}$$

byte 4 scientific notation 2's compliment.

If value of byte 4 > 127 the Exp is Negative, if value of byte 4 < 128 then Exp is Positive

[Back to Index](#)



Synchronous Serial Mode

Step-by Step Sequence to Read Absolute Intensity

Initialize the board

- Set Addr0/LA_Select = 0
- Set Clock_in = 0
- Set Data_in to 0

2. Clear the Comms Buffer

- Set Data_In to 1
- Set Clock_in to 1 and 0 [8] times to clock in each bit

3. Send the command to read Absolute Intensity as follows

- Set Data_In to 0X80 (10000000)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit

4. Send Fiber Number to Read

- Send (0x01 to 0x14) (1-20 decimal)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit

5. Read back 4 bytes of data

- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Intensity** high byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Intensity** middle byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Intensity** low byte]
- Set Clock_in to 1 and 0 repeat [8] times to clock out each bit
- Data_Out = [**Exp** byte]

[Back to Index](#)



Continuous Frequency Mode

Continuous Frequency Mode

The Led analyser has the ability to output continuous Frequency to represent the Led data HSI, WSI and XYI directly to a Frequency Counter without the need for an In-circuit tester machine. This is called Continuous Frequency mode and is detailed in this section of this manual. In Continuous Frequency Mode the Analyser will output Frequencies on ICT pins 15, 16, 17 to a frequency counter. It requires an initial setup in terminal window and there after it is free running.

This method is based on extracting LED data from the frequency of square wave output signals. There are three separate outputs and depending on the mode selected the frequency out represents – one each for Hue, Saturation and Intensity in HSI mode, one each for Dominant Wavelength, Saturation, Intensity in WSI mode and one each for x chromaticity, y chromaticity, Intensity in xyI mode. The frequency of the square wave signal from the Led Analyser is in the range 1KHz to 100KHz. This is a fast and simple method of testing the Led where a frequency counter is usually available. The Frequency value for Intensity (pin 15), Hue, Wavelength or x chromaticity (pin 16) and Saturation or y chromaticity (pin 17) of the LED are output simultaneously at the ICT Port.

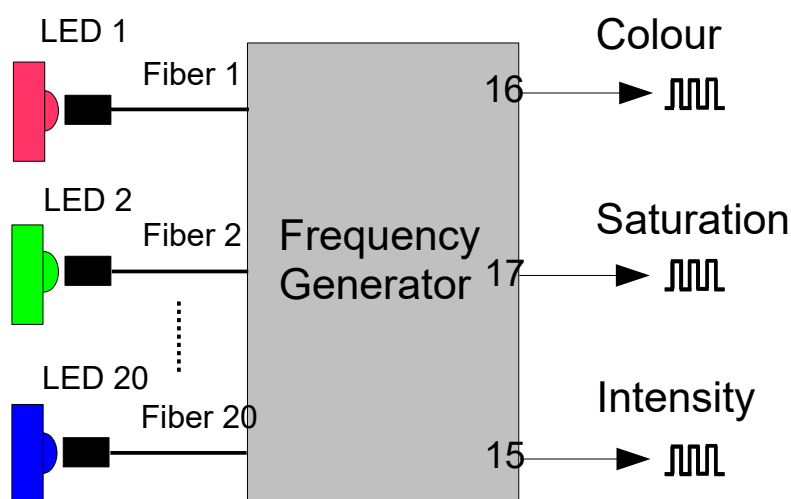


Figure 26.

The ICT Port consists of a 20-way connector which includes all three signals necessary to give a frequency output.

15	Output	Int	Frequency Out Square wave for Intensity
16	Output	Hue	Frequency Out Square wave for Hue, Wavelength, x Chromaticity
17	Output	Sat	Frequency Out Square wave for Saturation, y Chromaticity

FEASA LED ANALYSER

ICT VERSION



By default the Analyser is supplied with the Continuous Frequency mode **DISABLED** when shipped from the Factory.

To enable this mode use the command [setcontinuousmode1](#). The command [getICTStatus](#) will verify which modes have been selected. When this mode is selected SyncSerial and Frequency Out Modes are both disabled automatically.

The LED Analyser is set to the **Frequency Output - HSI** (Hue, Saturation, Intensity) mode at the Factory. There are two further frequency out options, **WSI** (Wavelength, Saturation, Intensity) and **xyI** (X chromaticity, Y chromaticity, Intensity). To place the analyser in WSI mode the command [setfrequencywsi](#) must be sent to the analyser. To place the analyser in xyI mode the command [setfrequencyxyi](#) must be sent to the analyser. The command [getICTStatus](#) will verify which modes have been selected. The commands must be sent using the Feasa Terminal window. The configuration is stored in Flash memory and need only be implemented once i.e. It will remain in the selected mode until instructed to change.

Summary of Commands:

setfrequencyhsi	Set the Analyser to Frequency Mode measuring HSI
setfrequencywsi	Set the Analyser to Frequency Mode measuring WSI
setfrequencyxyi	Set the Analyser to Frequency Mode measuring XYI
setcontinuousmode1	Enable the Analyser for Continuous Frequency Mode
setcontinuousmode0	Disable the Analyser for Continuous Frequency Mode
getcontinuousmode	This will indicated which modes the analyser is set for.
SetContinuousChannel##	Select which Fiber Number you are using.
SetContinuousCapture##	Select which capture range you require.
GetContinuousChannel	This will indicated which Fiber Number the analyser is set for.
GetContinuousCapture	This will indicated which Capture Range the analyser is set for.
getictstatus:	This will indicated which modes the analyser is set for.

Figure 9a-c (Pages 21/22) above show the typical Frequency outputs, the under range and over range conditions that will be reported.

[Back to Index](#)



Continuous Frequency Mode

SetContinuousMode - Enable the Continuous Frequency Mode

Transmit	Receive
setcontinuousmodeX	OK

Where:

- X** represents the mode
- 1** Enables the mode
- 0** Disables the mode

To determine which mode the analyser is set for use the command **getcontinuousmode** or **getictstatus**.

SetContinuousChannel## - Select which fiber you are using

Transmit	Receive
setcontinuouschannel##	OK

Where:

- ##** represents fiber 01 to 20

To determine which Channel the analyser is set for use the command **getcontinuouschannel** or **getictstatus**.

SetContinuousCapture## - Select which capture range you are using

Transmit	Receive
setcontinuouscapture##	OK

Where:

- ##** 01 – 05 represents the manual capture ranges Low(c1), Medium(c2), High(c3), Super(c4), Ultra(c5)
- ##** 06 represents the auto capture range (c)
- ##** 11 – 15 represents the manual pwm capture ranges Low(c1pwm), Medium, High, Super, Ultra(c5pwm)
- ##** 16 represents the auto pwm capture range (cpwm)

To determine which Capture range the analyser is set for use the command **getcontinuouscapture**.

[Back to Index](#)



Continuous Frequency Mode

GetContinuousMode - Verify Continuous Frequency Mode On/Off

Transmit	Receive
getcontinuousmode	#

Where:

1 *Enabled*
0 *Disabled*

To determine which mode the analyser is set for use the command **getcontinuousmode** or **getictstatus**.

GetContinuousChannel - Verify which fiber you are using

Transmit	Receive
getcontinuouschannel	##

Where:

*represents fiber 01 to 20*

To determine which Channel the analyser is set for use the command **getcontinuouschannel** or **getictstatus**.

GetContinuousCapture - Verify which capture range you are using

Transmit	Receive
getcontinuouscapture	##

Where:

*01 – 05 represents the manual capture ranges Low(c1), Medium(c2), High(c3), Super(c4), Ultra(c5)*
*06 represents the auto capture range (c)*
*11 – 15 represents the manual pwm capture ranges Low(c1pwm), Medium, High, Super, Ultra(c5pwm)*
*16 represents the auto pwm capture range (cpwm)*

To determine which Capture range the analyser is set for use the command **getcontinuouscapture**.

[Back to Index](#)



Serial Port Control (RS232)

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

5V DC @ 200mA Power must be supplied to Pins 19 and 20 of the 20-way Connector. See Figure 27. **Prolonged Voltages Greater than 6V will damage the Analyser.**

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. These can be selected using the [setbaud](#) command.

Serial Connector (RS232C)

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

20-Way Connector

Pin No	Signal
19	Power (5V 200mA DC)
20	GND

Figure 27.

Please Ensure the RS232 GND (pin3) and the PSU GND are connected together.

[Back to Index](#)



Capture Commands

The Capture commands are used to capture the parameters (Hue, Saturation, Intensity, xy, uv, dominant 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 Download Site 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 LED Analyser User** 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 raw data from the Analyser can be observed in the Window.

The **Feasa LED Analyser Test** Program allows the User to test all the LED's together. Pass and Fail limits can be set and results can be printed and logged.

Alternately, the User may generate a customised Program that sends commands and receives data through the RS232 Serial Port.

[Back to Index](#)



Capture Times Static Leds

The Capture Time for each range is outlined on the following Table.

Range	Capture Time
Range 1 low	610ms
Range 2 medium	210ms
Range 3 high	25ms
Range 4 super	6ms
Range 5 ultra	5ms
Auto Range	230ms

Notes:

* Values measured using a baud-rate of 57600baud, through the serial port. USB port may introduce additional latencies. Values can increase if baud-rate is reduced. RS232 to USB adapters may introduce additional latencies.

* An application note about how to reduce the test times is provided in the software bundle.

Capture Times PWM Leds

The Capture Time for each range is outlined on the following Table.

Range	Capture Time
Range 1 low C1PWM	4.75s
Range 2 med C2PWM	3.2s
Range 3 high C3PWM	1.74s
Range 4 super C4PWM	570ms
Range 5 ultra C5PWM	530ms
Auto Range CPWM	3.45s

Notes:

* Values measured using a baud-rate of 57600baud, through the serial port. USB port may introduce additional latencies. Values can increase if baud-rate is reduced. RS232 to USB adapters may introduce additional latencies.

* An application note about how to reduce the test times is provided in the software bundle.

[Back to Index](#)



Capture Mode

AUTO CAPTURE - Store LED Data

Transmit	Receive
capture c	OK OK

Description

This Auto Range Capture [capture](#) or [c](#) command instructs the LED Analyser to capture and store the data of all the LED's positioned under the fibers. The Analyser automatically determines the correct settings to capture the LED data based on the Intensity. In the case of a 20 channel unit the data for all 20 LED's are captured simultaneously and stored in internal memory of the LED Analyser. The data is stored until the power is removed or another [capture](#) or [c](#) command is issued. When completed the Analyser 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 manual [Capture#](#) command described on the next page.

Example:

The PC transmits [capture](#) to the LED Analyser and the LED Analyser sends **OK** to the PC to acknowledge that the command is completed.

capture
OK

or

C
OK

[Back to Index](#)



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 the LED Analyser 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 **capture#** command is issued. When completed the Analyser will transmit the character **OK** on the receive line to the transmitting device (i.e. the PC).

Example:

The PC transmits **capture#** to the LED Analyser and the LED Analyser 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](#)



Capture Mode

AUTO CAPTUREPWM - Capture PWM LED Data

Transmit	Receive
capturepwm cpwm	OK OK

Description

Pulse-Width-Modulated(PWM) LED's are switched on and off rapidly to save power and to control Intensity. The Analyser automatically determines the correct settings required to execute the test.

This command uses the *auto-ranging* feature and a pre-set *averaging factor* to capture the LED data. This command is useful if it is required to test very dim and very bright PWM LED's together.

The command instructs the LED Analyser to read and store the colour and Intensity of all the LED's positioned under the fibers. The data is stored until the power is removed or another **capture** command is issued. When completed the Analyser will transmit the character **OK** on the receive line to the transmitting device (i.e. the PC).

Example:

The PC transmits **capturepwm** to the LED Analyser. The Analyser sends **OK** to the PC to acknowledge that the command is completed.

capturepwm
OK

or

cpwm
OK

[Back to Index](#)



Capture Mode

MANUAL CAPTURE - Capture PWM LED Data for a specific range

Transmit	Receive
capture#pwm@@	OK
c#pwm@@	OK
capture#pwm	OK
c#pwm	OK

Where:

represents the exposure Range 1 – 5.

@@ represents an averaging factor in the range 1 - 15.

Note: If the @@ digits are omitted then a default setting of 07 is used.

Description

This command allows the User to specify the *exposure range #* and an *averaging factor @@* when testing PWM LED's. Select the *exposure range # (1-5)* to match the **Intensity** of the LED's. The Analyser tests these LED's by taking a number of readings and averaging the results. A larger Averaging factor will lead to more stable results but increased Test Times. The *averaging factor @@* is a number in the range 1-15. If this number is omitted from the command the default value is 07.

This command instructs the LED Analyser to read and store the colour and Intensity of all the LED's positioned under the fibers. The data is stored until the power is removed or another **capture** command is issued. When completed the Analyser will transmit the character **OK** on the receive line to the transmitting device (i.e. the PC).

capture1pwm10

OK

or

c1pwm10

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](#)



Capture Mode

[Back to Index](#)



Get Data Commands

The *get data* commands are used to read out the Hue, 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 Analyser.

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

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.

Incorrect Capture Mode

An incorrect capture mode condition will occur when a blinking (pwm) light from the LED reaches the sensor when the Autorange C capture is selected. This will be indicated by **XXX.XX XXX XXXXX** for **HSI**, **XXX XXX XXX XXXXX** for **RGBI** and **X.XXXX X.XXXX** for xy and uv.

If this condition occurs select the **CPWM** command and test again. In the later versions of software this CPWM mode is automatically deployed and can be verified using the *getstatus* or *getautopwm* commands.

The manual Capture Ranges C1 – C5 are unaffected by this.

[Back to Index](#)



Get Data Mode

getRGBI## - Get RGB and Intensity for a LED

Transmit	Receive
getrgbi## getrgbiall	Rrr ggg bbb iiii

Where:

represents the Fiber Number and is a number in the range 01 – 20.

all represents the 20 Fibers in a Feasa 20i, or 3 Fibers in a 3i.

rrr , **ggg** and **bbb** are the **red**, **green** and **blue** components of the LED colour.
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 0000 – 99999. 0000 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 LED Analyser to return RGB and Intensity data for fiber **##** (01-20) in format **rrr ggg bbb iiii** where **rrr**, **ggg** and **bbb** are the **red**, **green** and **blue** components of the colour. 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 the LED Analyser. The data for each LED can only be read out one-at-the-time, in any order. Note: it is important for the PC to use 01, 02, 03, etc to indicate Fibers 1, 2 and 3, etc.

Example:

The PC transmits **getrgbi05** to the LED Analyser to instruct it to send the stored colour and Intensity data for the LED positioned under Fiber No 5. The LED Analyser will return a string **rrr ggg bbb iiii** to the PC.

getrgbi05
006 230 018 06383

[Back to Index](#)

FEASA LED ANALYSER ICT VERSION



Get Data Mode

getHSI## - Get Hue, Saturation and Intensity

Transmit	Receive
gethsi## gethsiall	hhh.hh sss iiii

Where:

represents the Fiber Number and is a number in the range 01 – 20.

all represents the 20 Fibers in a Feasa 20i, or 3 Fibers in a 3i

hhh.hh represents the Hue (colour) 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 0000 – 99999. 0000 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 LED Analyser to return Hue, Saturation and Intensity data for fiber **##** (01-20) in format **hhh.hh sss iiii** where **hhh.hh** represents the Hue(colour), **sss** represents the Saturation(whiteness) of the LED under Fiber **##**. 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 the LED Analyser. The data for each LED can only be read out one-at-the-time, in any order. Note: it is important for the PC to use 01, 02, 03, etc to indicate Fibers 1, 2 and 3, etc.

Alternatively use the *gethsiall* command to read back the LED data for all 20 leds in one command.

Example:

The PC transmits **gethsi05** to the LED Analyser to instruct it to send the stored colour and Intensity data for the LED positioned under Fiber No 5. The LED Analyser will return a string **hhh.hh sss iiii** to the PC.

gethsi05
123.47 098 06383

[Back to Index](#)



Get Data Mode

getxy## - Return the xy Chromaticity values

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

Where:

represents the Fiber Number and is a number in the range 01 – 20.
all represents the 20 Fibers in a Feasa 20i, or 3 Fibers in a 3i.

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 Analyser. The data for each LED can only be read out one-at-the-time, in any order. Note: it is important for the PC to use 01, 02, 03, etc to indicate Fibers 1, 2 and 3, etc.

Alternatively use the *getxyall* command to read back the LED data for all 20 leds in one command.

Example:

The PC transmits **getxy01** to the LED Analyser to instruct it to send the stored xy Chromaticity data for the LED positioned under Fiber No 1. The LED Analyser will return a string **0.xxxx 0.yyyy** to the PC.

getxy01
0.6461 0.3436

[Back to Index](#)



Get Data Mode

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

Transmit	Receive
getxyi## getxyiall	0.xxxx 0.yyyy IIIII

Where:

represents the Fiber Number and is a number in the range 01 – 20.
all represents the 20 Fibers in a Feasa 20i, or 3 Fibers in a 3i.

0.xxxx represents the x Chromaticity value
0.yyyy represents the y Chromaticity value
IIIII represents the Relative Intensity 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 Analyser. The data for each LED can only be read out one-at-the-time, in any order. Note: it is important for the PC to use 01, 02, 03, etc to indicate Fibers 1, 2 and 3, etc.

Alternatively use the *getxyiall* command to read back the LED data for all 20 leds in one command.

Example:

The PC transmits **getxyi01** to the LED Analyser to instruct it to send the stored xy Chromaticity data for the LED positioned under Fiber No 1. The LED Analyser will return a string **0.xxxx 0.yyyy 12345** to the PC.

getxyi01
0.6461 0.3436 03456

[Back to Index](#)



Get Data Mode

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

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

Where:

@@ represents the Fiber Number and is a number in the range 01 – 20.

all represents the 20 Fibers in a Feasa 20i, or 3 Fibers in a 3i.

0.uuuu represents the u Chromaticity value

0.vvvv represents the v Chromaticity value

Description

This command is used to return the uv 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 Analyser. The data for each LED can only be read out one-at-the-time, in any order. Note: it is important for the PC to use 01, 02, 03, etc to indicate Fibers 1, 2 and 3, etc.

Alternatively use the *getuvall* command to read back the LED data for all 20 leds in one command.

Example:

The PC transmits **getuv01** to the LED Analyser to instruct it to send the stored xy Chromaticity data for the LED positioned under Fiber No 1. The LED Analyser will return a string **0.0000 0.0000** to the PC.

**getuv01
0.1809 0.4414**

[Back to Index](#)



Get Data Mode

getWAVELENGTH## - Get the Dominant Wavelength

Transmit	Receive
getwavelength## getwavelengthall	XXX

Where:

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

all represents the 20 Fibers in a Feasa 20i, or 3 Fibers in a 3i.

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 the **capture C** command to ensure valid LED data is stored in the memory of the LED Analyser. The **C1-C5** commands are available but are not as accurate. The data for each LED can only be read out one-at-the-time, in any order.

Alternatively use the *getwavelengthall* command to read back the LED data for all 20 leds in one command.

Example:

The PC transmits **getwavelength01** to the LED Analyser to instruct it to send the stored dominant wavelength for the LED positioned under Fiber No 1. The LED Analyser will return a string **xxx** to the PC.

**getwavelength01
513**

[Back to Index](#)



Get Data Mode

getWI## - Get the Dominant Wavelength & Intensity

Transmit	Receive
getwi## getwiall	XXX YYYYY

Where:

represents the Fiber Number and is a number in the range 1 – 20.
all represents the 20 Fibers in a Feasa 20i, or 3 Fibers in a 3i.

xxx represents the dominant wavelength of the LED in nanometers.
YYYYY represents the intensity of the LED.

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 the **capture C** command to ensure valid LED data is stored in the memory of the LED Analyser. The **C1-C5** commands are available but are not as accurate. The data for each LED can only be read out one-at-the-time, in any order.

Alternatively use the *getwiall* command to read back the LED data for all 20 leds in one command.

Example:

The PC transmits **getwi01** to the LED Analyser to instruct it to send the stored dominant wavelength & Intensity for the LED positioned under Fiber No 1. The LED Analyser will return a string **xxx yyyyy** to the PC.

getwi01
513 12345

[Back to Index](#)



Get Data Mode

getWSI## - Get the Dominant Wavelength, Saturation & Intensity

Transmit	Receive
getwsi## getwsiall	XXX SSS YYYYY

Where:

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

all represents the 20 Fibers in a Feasa 20i, or 3 Fibers in a 3i.

XXX represents the dominant wavelength of the LED in nanometers.

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

YYYYY represents the intensity of the LED.

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 the **capture C** command to ensure valid LED data is stored in the memory of the LED Analyser. The **C1-C5** commands are available but are not as accurate. The data for each LED can only be read out one-at-the-time, in any order.

Alternatively use the *getwsiall* command to read back the LED data for all 20 leds in one command.

Example:

The PC transmits **getwsi01** to the LED Analyser to instruct it to send the stored dominant wavelength & Intensity for the LED positioned under Fiber No 1. The LED Analyser will return a string **xxx yyyyy** to the PC.

getwsi01
513 100 12345

[Back to Index](#)



Get Data Mode

getCCT## - Get the Correlated Colour Temperature

Transmit	Receive
getcct## getcctall	cccc + /-d.ddd

Where:

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

all represents the 20 Fibers in a Feasa 20i, or 3 Fibers in a 3i.

cccc represents the Correlated Colour Temperature of the LED.

d.ddd represents the perpendicular distance the LED is from the Plankian locus (*Delta E*). This result can be + or-

Description

This command is used to get the value of the Correlated Colour Temperature (CCT) for the LED under the Fiber number #.

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

Alternatively use the *getcctall* command to read back the LED data for all 20 leds in one command.

Example:

The PC transmits **getcct01** to the LED Analyser to instruct it to send the stored Correlated Colour Temperature for the LED positioned under Fiber No 1. The LED Analyser will return a string **cccc d.ddd** to the PC.

getcct1

04621 +0.0340

getCCT is NOT Valid when measuring pure colour Leds (Red, Green, Blue). The output from the Analyser will report 0 for CCT and +0.5555 for delta uv.

[Back to Index](#)



Get Data Mode

getINTENSITY## - Get the Intensity

Transmit	Receive
getintensity## getintensityall	iiii

Where:

represents the Fiber Number and is a number in the range 01 – 20.

all represents the 20 Fibers in a Feasa 20i, or 3 Fibers in a 3i.

iiii represents the intensity value of the LED under fiber **##**. This 5-digit number is in the range 0000 – 99999. 0000 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 Intensity value for the LED under the Fiber number **##**.

This command should be preceded by a **capture** command to ensure valid LED data is stored in the memory of the LED Analyser. The data for each LED can be read out one-at-the-time, in any order using the **getintensity##** command or all data can be read simultaneously using the **getintensityall** command. Note: it is important for the PC to use 01, 02, 03, etc to indicate Fibers 1, 2 and 3, etc.

Example:

The PC transmits **getintensity01** to the LED Analyser to instruct it to send the stored Intensity data for the LED positioned under Fiber No 1. The LED Analyser will return a string **iiii** to the PC.

getintensity01
06734

[Back to Index](#)

FEASA LED ANALYSER ICT VERSION



Get Data Mode

getABSINT@@ - Get the Absolute Intensity Value of the Led under test.

Transmit	Receive
getabsint@@	#.####e±##

Where:

@@ represents the Fiber Number and is a number in the range 01 – 20.

all represents the 20 Fibers in a Feasa 20i, or 3 Fibers in a 3i.

#.####e±## represents the Absolute Intensity reading

Description:

This command will allow the user to interrogate the Led Analyser to return the absolute Intensity Value for the Led under test at Fiber@@.

This command should only be used in conjunction with the Feasa LED Spectrometer. Absolute Intensity values must first be measured with the Spectrometer. The LED Analyser must then be calibrated with these values using the **usercal** program supplied with the Analyser.

Example:

Place fiber 01 over the Led to be tested. Using any of the **Capture** commands, capture the data for that Led. Type the command **getabsint01** and the analyser will respond with the intensity reading in the format **#.####e±##**

getabsint01
2.5000e-02

[Back to Index](#)



Get Data Mode

getCIEXYZ## - Return the CIE 1931 XYZ Colour values

Transmit	Receive
getCIEXYZ##	0.0000e+00 0.0000e+00 0.0000e+00

Where:

represents the Fiber Number and is a number in the range 01 – 20.

0.0000e+00 represents the X value

0.0000e+00 represents the Y value (Absolute Intensity *)

0.0000e+00 represents the Z value

Description

The [getciexyz##](#) Led Analyser is capable of measuring the CIE 1931 XYZ components. The **Y component** is the photometric magnitude corresponding to the brightness of the source and corresponds to the photometric quantity that was used to calibrate the LED Analyser with the Feasa Spectrometer and:

- the Integrating Sphere (SPxx) was used then the Photometric Quantity is - Luminous flux (in lm)
- or the Millicandela Head (CDxx) was used then the Photometric Quantity is - Luminous intensity (in cd)
- or the Luminance Head (LUxx) was used then the Photometric Quantity is - Luminance (in cd/m²)

This can also be referred to as the Absolute Intensity. The command [getCIEXYZ@@](#) is used to return the CIE XYZ Colour value for the LED under the Fiber number @@ after a Capture.

To achieve the XYZ values from the Led Analyser there are a couple of steps to follow:

- * Calibrate each Channel/Fiber of the Led Analyser for **Absolute Intensity**.
- * When Calibration is complete, do a **Capture** to ensure valid Data is stored in the memory of the LED Analyser
- * The data for each LED can be read out one-at-the-time, in any order using the command [getciexyz##](#). Note: it is important for the PC to use 01, 02, 03, etc to indicate Fibers 1, 2 and 3, etc.

Example:

The PC transmits [getciexyz01](#) to the LED Analyser to instruct it to send the stored CIEXYZ Colour data for the LED positioned under Fiber No 1. The LED Analyser will return a string in the format **0.0000e+00 0.0000e+00 0.0000e+00** to the PC.

getciexyz01

1.2345e+04 9.8765e-01 5.6789e+03

[Back to Index](#)

FEASA LED ANALYSER ICT VERSION



Get Data Mode

getSIGNALLEVEL## - Get the Relative Intensity in a percentage of the Range

Transmit	Receive
getsignallevel## getsignallevelall	iii%

Where:

represents the Fiber Number and is a number in the range 01 – 20.

all represents the 20 Fibers in a Feasa 20i, or 3 Fibers in a 3i.

iii% represents the intensity value in percentage terms of the LED under fiber ##. This 3-digit number is in the range 001 – 100. A reading of 000% represents no Intensity or under range (i.e. the LED is off) and a reading of 999% will represent over range or the LED is too bright.

Note: it is important to verify the Led Analyser is set to the correct Capture Range.

Description

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

Note: it is important for the PC to use 01, 02, 03, etc to indicate Fibers 1, 2 and 3, etc.

Alternatively use the **getsignallevelall command to read back the LED data for all 20 leds in one command.**

Example:

The PC transmits **getsignallevel01** to the LED Analyser to instruct it to send the stored Intensity data for the LED positioned under Fiber No 1. The LED Analyser will return **iii%** to the PC.

getsignallevel01
050 %

[Back to Index](#)



Get Data Mode

get7SEG# - Get the value of a 7 Segment Display

Transmit	Receive
get7seg#	x

Where:

represents the Number 1 or 2
x represents the value of the display 0 - 9

Description

The LED Analyser can be used to test LED-based 7-Segment displays. To set up the LED Analyser to interrogate a single 7-Segment display, fit fibers labelled 1 to 7 over segments a-g on the 7-Segment display. See *Figure 28*.

To set up the LED Analyser to interrogate an additional 7-Segment display, fit fibers labelled 11 to 17 over segments a-g on the additional display.

To interrogate the digit displayed on the first 7-Segment Display send the command **get7seg1** to the LED Analyser. The LED Analyser will return the digit displayed. The LED Analyser will return the character **X** when the displayed value is not recognized (**0 to 9**).

To interrogate the digit displayed on the second 7-Segment display send the command **get7seg2** to the LED Analyser. The LED Analyser will return the digit displayed. Again, the LED Analyser will return the character **X** when the displayed value is not recognized (**0 to 9**).

Note:- it is not necessary to send any *capture* commands prior to using the **get7seg1** or **get7seg2** commands.

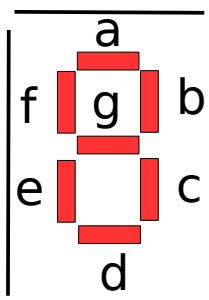


Figure 28.

Example:

The PC transmits **get7seg1** to the LED Analyser and the Analyser will return the value of the display.

get7seg1

[Back to Index](#)



Get Data Mode

[*Back to Index*](#)



Get Data Mode

getFACTOR - Get the exposure Factor

Transmit	Receive
getfactor	xx

Where:

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

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:

The PC transmits **getfactor** to the LED Analyser to instruct it to send the stored exposure factor for all Fibers. The default value is 01.

getfactor
01

[Back to Index](#)



Get Data Mode

getAUTOPWM - Get which Auto Capture Mode is active C or CPWM

Transmit	Receive
getautopwm	0 or 1

Where: **0** is Standard Auto Capture (c) and **1** is PWM Auto Capture (cpwm)

Description

This command will verify the Capture mode of the Analyser between Auto Capture (C) and Auto PWM Capture (CPWM).

Example:

Connect the Led Analyser to the Feasa Terminal Window and type `getstatus`. The Analyser will report Auto PWM Mode: OFF or type `getautopwm` and the Analyser will report 0.

Set a fiber over a Led to test and do an Auto Capture (C). If its a non-pwm Led the Analyser will report useable data for HSI, RGB or XY as requested. However if the LED is actually a PWM Led the Analyser will detect un-useable data in the form XXX. Type the command `setautopwm1` to change the capture mode. Repeat the test, The analyser will now mimic the c capture as a cpwm capture and report useable data. Doing a `getstatus` will now show Auto PWM Mode: ON or `getautopwm` will report a 1. It will remain in this mode until the user changes the mode by typing the command `setautopwm0`.

[Back to Index](#)



Get Data Mode

getPHOTOPIC - Get which Photopic Mode is active

Transmit	Receive
getphotopic	0 or 1 or 2

Where: **0 is Photopic Response OFF**
 1 is Response Photometric
 2 is Response Radiometric

Description

The Led Analyser is shipped from the factory in photopic mode 1. This corrects the Absolute Intensity to reproduce the photometric response (for lm, mcd). This can be verified by typing the command [getphotopic](#) with the answer 1.

The photopic mode can be set to a radiometric response by sending the command [setphotopic2](#). This corrects the Absolute Intensity to reproduce the radiometric response (for mW)

Example:

To get the Photopic Mode of the Analyser transmit the command [getphotopic](#) to the Analyser.

Getphotopic
1

[Back to Index](#)



Get Data Mode

getintensitymode - Get which Intensity Mode is active Logarithmic or Linear

Transmit	Receive
getintensitymode	Log or Lin

Where: **Log** is the Logarithmic Mode and **Lin** is the Linear Mode

Description

The Intensity level of the led under test can be measured in two formats, Logarithmic or Linear Refer to pages 10,11 of this document for more details.

The Led Analyser is shipped from the factory in Logarithmic Mode. However it is possible to change the mode to Linear by using the commands set/putlin. To reset the analyser back to log mode use the command set/putlog.

Example:

Connect the Led Analyser to the Feasa Terminal Window and type [getintensitymode](#). The Analyser will report the mode the Intensity output of the Analyser is set to.

getintensitymode

Linear

[Back to Index](#)



General Commands

getSERIAL Get the Serial Number of the Analyser

Transmit	Receive
getSerial	xxxx

Where: xxxx is an alphanumeric value.

Description

This command will return the Serial Number of the Analyser. This is a unique number and is useful if multiple LED Analysers are used in a System. The Controlling Software can query each LED Analyser for it's Serial Number to ensure the correct Analyser is being controlled.

Example:

The PC transmits **getserial** to the LED Analyser an it will return **xxxx** to the PC.

getserial
75A6

[Back to Index](#)



General 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 Analyser.

Example:

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

getversion
F122

[Back to Index](#)



General Commands

getBAUD- Get the Baud Rate

Transmit	Receive
getBAUD	X bps

Where: **X** = 9600, 19200, 38400, 57600, 115200 for **Serial RS232 Port**

Where: **X** = 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600 for **USB Port**

Description

This command will get the baud rate of the Serial and USB Port's that the Analyser is communication through.

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

It can also be seen on the comment section of the Feasa terminal window.

Example:

To get the baud rate of the Analyser transmit the command **getbaud** to the Analyser.

getbaud
57600 bps

[Back to Index](#)



General Commands

getHW - Get the Hardware Version

Transmit	Receive
getHW	Feasa XX-YY

Where: XX-YY is an alphanumeric value.

Description

This command will return the hardware version the Analyser.

Example:

The PC transmits **gethw** to the LED Analyser and it will return **Feasa XX-YY** to the PC.

gethw
Feasa 20-I

[Back to Index](#)



General Commands

getNFIB - Get the the Number of Channels on the Analyser

Transmit	Receive
getNFIB	xx

Where: XX is an alphanumeric value in the range 01 to 20. (Always 2 characters)

Description

This command will return the Nunber of Channels on the Analyser.

Example:

The PC transmits `getnfib` to the LED Analyser an it will return **xx** to the PC.

getnfib
06

[Back to Index](#)



General Commands

getSTATUS - Get a summary of the Led Analyser details

Transmit	Receive
getstatus	See details below

Information Received back from the Analyser:

Hardware: Feasa xx-y
Serial number : Xyyy
Firmware Version : Iyyy
Intensity Mode : Logarithmic
Exposure Factor : 01
Capture : Manuel 04
Auto PWM mode: OFF
Number of Fibers : 06
External Trigger status: Disabled
Colour Gains: xxx xxx xxx
End of Transmission Character: Disabled
Selected calibration set: 00
user calibration date: dd/mm/yyyy
Factory calibration date: dd/mm/yyyy

Description

This command will return a summary of the Led Analyser details.

Example:

The PC transmits **getstatus** to the LED Analyser and it will return to the PC.

Led Analyser STATUS

Hardware: Feasa 20-I
Serial Number : J001
Firmware Version : F123
Intensity Mode : Logarithmic
Exposure Factor: 01
Last Capture : Manuel 04
Auto PWM Mode: OFF
Number of Fibers : 20
External Trigger Status: Disabled
Colour Gains: 100 150 100
End of Transmission Character: Disabled
Selected calibration set: 0
User Calibration Date: 01/01/2015
Factory calibration date: 23/10/2016

[Back to Index](#)



General Commands

getICTSTATUS - Get the Led Analyser ICT Mode details

Transmit	Receive
getictstatus	See details below

The Led Analyser is shipped from the factory with the Frequency out mode set to HSI and Enabled while the Sync Serial Mode is disabled. The Continuous Frequency Mode is OFF.

Information Received back from the Analyser:

Frequency out mode: xxx (where xxx is HSI, WSI or xyI)
Sync Serial Mode: ????? (where ????? is Enabled or Disabled)
Continuous Mode: YY ## @@ (where yy is ON/OFF, ## is Fiber No. @@ Capture Range)

Description

This command will return a summary of the Led Analyser ICT mode details.

To change the Frequency Out mode refer to the **setfrequencyxxx** commands.

To enable the Synchronous Serial Mode refer to the **setsyncserialx** commands.

To enable the Continuous Frequency Mode refer to the **setcontinuousmodeX** commands.

Example:

The PC transmits **getictstatus** to the LED Analyser and it will return to the PC.

Frequency out mode: HSI
Sync Serial Mode disabled
Continuous Mode: ON 01 01

[Back to Index](#)



General Commands

EOT - End of Transmission Character

Transmit	Receive
enableeot disableeot	See details below

Description

Every command sent to the LED Analyser or response received from the Led Analyser, ends with a set of characters that indicates that the command or response has finished. Those characters are CR + LF, which are ASCII characters 13 and 10 respectively. Alternatively, responses from the LED Analyser can also have an end-of-transmission format like CR + LF + EOT, which are the ASCII characters 13, 10 and 04 respectively. CR + LF are always present, but character EOT can be enabled or disabled using the commands ENABLEEOT and DISABLEEOT. Having EOT character enabled can be extremely useful in multi-line commands, since this allows to differentiate between the end of a line and the end of a transmission.

The EOT command is disabled by default. To enable EOT simply send the command enableeot using Feasa Terminal software. The Analyser will maintain the EOT state even when powered off. To disable the state just send the command disableeot.

Information Received back from the Analyser using the getstatus command is as follows:

Hardware: Feasa xx-y
Serial number : Xyyy
Firmware Version : Iyyy
Intensity Mode : Logarithmic
Exposure Factor : 01
Capture : Manuel 04
Number of Fibers : 06
External Trigger status: Disabled
Colour Gains: xxx xxx xxx
End of Transmission Character: **Disabled**
Selected calibration set: 0
user calibration date: dd/mm/yyyy
Factory calibration date: dd/mm/yyyy

[Back to Index](#)



General Set/Put Commands

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.

[Back to Index](#)



Set Data Mode

set/putFACTOR## - Set the Exposure Factor

Transmit	Receive
setfactor## putfactor##	OK OK

Where:

represents the Factor Number and is in the range 01 – 15 (default 01).

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 and higher Intensity readings, For Example.

Capture Range	Factor 01 / Intensity	Factor 05 / Intensity	Factor 10 / Intensity	Factor 15 / Intensity
C3 (High)	22ms / 23K	110ms / 50K	220ms / 64K	330ms / 72K

Try using the lowest capture range first (c1) before adjusting the Exposure Factor.

These set values are stored permanently in memory and can only be changed by using the *setfactor* command again, power on/off will NOT effect the set Factor.

The put values are stored in RAM memory and must be changed by using the *putfactor* command each time after Power On/Off.

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

Example:

Set the Factor Number for all Fibers to 05.

setfactor05
OK

Example:

Set the Factor Number for all Fibers to 05.

putfactor05
OK

[Back to Index](#)



Set Data Mode

set/putLOG - Change the Intensity Response of the Analyser to Logarithmic mode

Transmit	Receive
setlog putlog	OK OK

Description

This command will change the Intensity Response of the Analyser to measure in Logarithmic mode. All Analysers shipped from the factory are set in Log mode.

Please refer to the Intensity Section of this document page 8 [Intensity](#)

The set command is stored permanently in memory and can only be changed by using the *setlog* command again, power on/off will NOT effect the setlog

The put values are stored in RAM memory and must be changed by using the *putlog* command each time after Power On/Off.

Example:

setlog
OK

Example:

putlog
OK

[Back to Index](#)



Set Data Mode

set/putLIN - Change the Intensity response of the Analyser to Linear Mode

Transmit	Receive
setlin putlin	OK OK

Description

This command will change the Intensity Response of the Analyser to measure in Linear mode. All Analysers shipped from the factory are set in Log mode.

Please refer to the Intensity Section of this document page 8 [Intensity](#)

The set command is stored permanently in memory and can only be changed by using the *setlin* command again, power on/off will NOT effect the setlin

The put values are stored in RAM memory and must be changed by using the *putlin* command each time after Power On/Off.

Example:

setlin
OK

Example:

putlin
OK

[Back to Index](#)



General set/put Commands

[Back to Index](#)



General set/put Commands

setBAUD - Change the baud rate

Transmit	Receive
Setbaudxxxxx	OK

Where: **X** = 9600, 19200, 38400, 57600, 115200 for **Serial RS232 Port**

Description

This command will change the baud rate of the Serial and USB Port's in the Analyser.

The default Port settings of the Analyser are 57,600, 8 Data bits, 1 Stop bit and No Parity. If an incompatible baud rate is selected the Analyser will not respond. Please disconnect and reconnect the Analyser to return the Analyser to the default 57600 bps. A correct baud rate selected will remain stored in memory even after power off.

The command getbaud will return the set baud rate. It can also be seen on the comment section of the Feasa terminal window.

Example:

To change the baud rate to 19,200 transmit the command **setbaud19200** to the Analyser .

setbaud19200

OK

[Back to Index](#)



General set/put Commands

setAUTOPWM# - Change the Auto Capture mode to Auto PWM Mode

Transmit	Receive
setautopwm#	OK

Where: # = **0** for **OFF** and **1** for **ON**,

Description

This command will change the Auto Capture mode of the Analyser from an Auto Capture to an Auto PWM Capture should the Led Analyser detect a blinking Led.

Example:

Connect the Led Analyser to the Feasa Terminal Window and type `getstatus`. The Analyser will report Auto PWM Mode: OFF. Set your fiber over a Led and do an Auto Capture (C). If its a standard Led the Analyser will report useable data for HSI, RGB or XY as requested. However if the LED is actually a PWM Led the Analyser will detect un-useable data in the form XXX. To compensate for for this type the command `setautopwm1` to change the capture mode. Repeat the test, The analyser will now mimic the c capture as a cpwm capture and report useable data. Doing a `getstatus` will now show Auto PWM Mode: ON or `getautopwm` will report a 1. It will remain in this mode until the user changes the mode by typing the command `setautopwm0`.

[Back to Index](#)



Set Data Mode

setFREQUENCYxxx - Set the Frequency Output Mode of the Analyser

Transmit	Receive
SetFrequencyxxx	OK

Where:

xxx represents the Frequency Measuring mode of the Analyser at ICT
HSI represents the Hue, Saturation & Intensity mode.
XyI represents the Xchromaticity, yChromaticity & Intensity mode.
WSI represents the Dominant Wavelength, Saturation & Intensity mode.

Description

The Analyser shipped from the factory is defaulted to FrequencyOut-HSI (Hue, Saturation and Intensity) Mode, There are 2 other frequency out modes available – WSI (Dominant Wavelength, Saturation, Intensity) and xyI (xy chromaticity and Intensity) These commands are used to set the Analyser ICT Interface in *Frequency Output* mode of your choice. It must be issued through the Serial Port. The unit will remain in this mode until instructed otherwise. The status of the ICT Interface can be determined using the [getICTStatus](#) command.

Example:

The PC transmits **setfrequencyhsi** to the LED Analyser to instruct it to set Frequency Output Hue, Saturation & Intensity mode.

The PC transmits **setfrequencywsi** to the LED Analyser to instruct it to set Frequency Output Wavelength, Saturation & Intensity mode.

The PC transmits **setfrequencyxyi** to the LED Analyser to instruct it to set Frequency Output xChromaticity, yChromaticity & Intensity mode.

Setfrequencyhsi
OK

Setfrequencywsi
OK

Setfrequencyxyi
OK

[Back to Index](#)



Set Data Mode

setSyncSerialx - Set the Output Mode of the Analyser to SyncSerial

Transmit	Receive
Setsyncserialx	OK

Where:

x	Represents the Sync Serial Enable / Disable option for the Analyser at ICT
1	Enables the Synchronous Serial Mode
0	Disables the Synchronous Serial Mode

Description

The Analyser shipped from the factory is defaulted to FrequencyOut-HSI (Hue, Saturation and Intensity) Mode, The Synchronous Serial Mode is Disabled by default. The [setsyncserial1](#) command which must be issued through the Serial Port to activate the Synchronous Serial Mode. The unit will remain in this mode until instructed otherwise. The status of the ICT Interface can be determined using the [getICTStatus](#) command.

It is **NOT recommended** to have the Synchronous Serial Mode Enabled if you are using the Frequency out mode.

On the other hand Having the Frequency mode active during Synchronous Serial Mode will have no effect on the results.

Example:

The PC transmits **setsyncserial1** to the LED Analyser to instruct it to Enable the Synchronous Serial Mode.

The PC transmits **setsyncserial0** to the LED Analyser to instruct it to Disable the Synchronous Serial Mode.

Setsyncserial1
OK

Setsyncserial0
OK

[Back to Index](#)



Set Data Mode

setPHOTOPIC - Set which Photopic Mode you require

Transmit	Receive
setphotopic#	OK

Where #: **0 is Photopic Response OFF**
 1 is Response Photometric (Default from the Factory)
 2 is Response Radiometric

Description

The Led Analyser is shipped from the factory in photopic mode 1. This corrects the Absolute Intensity to reproduce the photometric response (for lm, mcd). This can be verified by typing the command [getphotopic](#) with the answer 1.

The photopic mode can be set to a radiometric response by sending the command [setphotopic2](#). This corrects the Absolute Intensity to reproduce the radiometric response (for mW)

The photopic mode can be turned OFF by sending the command [setphotopic0](#). With the Analyser in this mode there is no effect on Absolute Intensity.

To return the Led Analyser to default mode send the command [setphotopic1](#)

[Back to Index](#)



User Calibration Mode

User Calibration Mode

Introduction:

The user calibration mode function allows the up to 9 (1 to 9) different calibration sets to be simultaneously stored on the Led Analyser. The Analyser is shipped from the factory in set mode 1. [setcalibration#](#) (# is 1 to 9) will enable which set you are working in. While in this mode you can vary the xy-offset, the wavelength offset or the intensity gains and then save them in that *version* or *set*. The date of this user calibration can also be saved using the command [setcalibrationdatedd/mm/yyyy](#). Using the [getstatus](#) or [getcalibrationset](#) commands the Led Analyser will display which user calibration set you are enabled in and also the date you set it up.

For example if you are testing a board which has 2 different led Colour options you could [setcalibration1](#), adjust your gains or offsets as necessary and proceed to test your version 1 board. [setcalibration2](#) and now adjust you gains or offsets as necessary for your other version of the board and proceed to test your version 2 of the board. The user calibration date for each set can be saved also.

Before doing the Calibration, we recommend you select the proper capture mode and range for the Leds you are testing. **Standard Capture** (C, C1, C2....) mode is recommended for static Leds whereas **Capture PWM** (eg. c2pwm05) is recommended for RGB LEDs that are pulsing.

Select the relevant **Range** (Low, Medium High, Super & Ultra) by measuring the Led color to be measured in the production. Ensure the **signal level is > 30 %** in order to maximize the repeatability.

For PWM mode you also need to select the best **average factor** (00-15) by doing a repeatability analysis on the final colour. The higher the average number, the more repeatable the results will be, However the higher the average the longer the capture time will be required. The Feasa User software has a repeatability test analysis to help you on this task.

When this data is accumulated then proceed with the calibration. Feasa provide a program called UserCal on the Download Site to accomplish this process. However should you wish to undertake the process manually please refer to the procedure below.

[Back to Index](#)



User Calibration Mode

set/putINTGAIN - Set the Intensity GainFactor

Transmit	Receive
Setintgain##xxx putintgain##xxx	OK OK

Where:

represents the Fiber Number and is a number in the range 01 – 20.
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 and the value for each Fiber can be adjusted from 050 – 200.

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

The put values are stored in RAM memory and must be changed by using the *putIntGain* command each time after Power On/Off.

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

Example:

Set the Intensity gain for Fiber 1 to 095.

setintgain01095
OK

Example:

Set the Intensity gain for Fiber 1 to 095.

putintgain01095
OK

[Back to Index](#)



User Calibration Mode

getINTGAIN## - Get the Intensity Gain Factor

Transmit	Receive
getIntGain##	xxx

Where:

represents the Fiber Number and is a number in the range 01 – 20.

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 **getintgain01** to the LED Analyser to instruct it to send the stored intensity gain data for Fiber No 1.

getintgain01
100

[Back to Index](#)



User Calibration Mode

set/putXOFFSET##0.xxx - Set the x Chromaticity Offset

Transmit	Receive
setxoffset##±0.xxx putxoffset##±0.xxx	OK OK

Where:

represents the Fiber Number and is a number in the range 01 – 20.
0.xxx represents the x Chromaticity offset value ($\pm 0.000 - 0.300$).

Description

This set 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 $\pm 0.000 - 0.300$. 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.

This put 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 $\pm 0.000 - 0.300$. 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 RAM memory and will remain at the programmed setting until changed by a new *Setxoffset* command or power OFF.

Example:

The PC transmits **setxoffset01+0.050** to the LED Analyser to instruct it to set the x offset on Fiber 01 to +0.050.

setxoffset01+0.050
OK

Example:

The PC transmits **setxoffset01+0.050** to the LED Analyser to instruct it to set the x offset on Fiber 01 to +0.050.

putxoffset01+0.050
OK

[Back to Index](#)



User Calibration Mode

set/putyOFFSET##0.yyy - Set the x Chromaticity Offset

Transmit	Receive
setyoffset##±0.yyy putyoffset##±0.yyy	OK OK

Where:

represents the Fiber Number and is a number in the range 01 – 20.
0.yyy represents the y Chromaticity offset value (±0.000 – 0.300).

Description

This set 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 – 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.

This put 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 – 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 RAM memory and will remain at the programmed setting until changed by a new *Setyoffset* command or power OFF.

Example:

The PC transmits **setyoffset01+0.050** to the LED Analyser to instruct it to set the y offset on Fiber 01 to +0.050.

setyoffset01+0.050
OK

Example:

The PC transmits **putyoffset01+0.050** to the LED Analyser to instruct it to set the y offset on Fiber 01 to +0.050.

putyoffset01+0.050
OK

[Back to Index](#)



User Calibration Mode

getXOFFSET## - Return the x Chromaticity offset

Transmit	Receive
getxoffset##	±0.xxx

Where:

represents the Fiber Number and is a number in the range 01 – 20.

±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 **getxoffset01** to the LED Analyser to instruct it to send the stored x Chromaticity offset for the LED positioned under Fiber No 1. The LED Analyser will return a string **±0.xxx** to the PC.

getxoffset01
+0.155

[Back to Index](#)



User Calibration Mode

getYOFFSET## - Return the y Chromaticity offset

Transmit	Receive
getyoffset##	±0.yyy

Where:

represents the Fiber Number and is a number in the range 01 – 20.

±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 **getyoffset01** to the LED Analyser to instruct it to send the stored y Chromaticity offset for the LED positioned under Fiber No 1. The LED Analyser will return a string **±0.yyy** to the PC.

getyoffset01
-0.025

[Back to Index](#)



User Calibration Mode

set/putWAVELENGTHOFFSET@@±## - Set the wavelength Offset

Transmit	Receive
setwavelengthoffset##±xx putwavelengthoffset##±xx	OK OK

Where:

@@ represents the Fiber Number and is a number in the range 01 – 20.
xx 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.

This put 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 RAM memory and will remain at the programmed setting until changed by a new *setwavelengthoffset* command or power OFF.

Example:

The PC transmits **setwavelengthoffset01-05** to the LED Analyser to instruct it to set the wavelength offset on Fiber 01 to -05.

setwavelengthoffset01-05
OK

Example:

The PC transmits **putwavelengthoffset01-05** to the LED Analyser to instruct it to set the wavelength offset on Fiber 01 to -05.

putwavelengthoffset01-05
OK

Back to Index



User Calibration Mode

getWAVELENGTHOFFSET@@ - *Get the Dominant 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](#)



User Calibration Mode

set/putABSINTMULT@@ - Set the Absolute Intensity Multiplier Factor

Transmit	Receive
setabsintmult@@m.mmmmmme+mm putabsintmult@@m.mmmmmme+mm	OK OK

Where:

@@ represents the Fiber Number and is a number in the range 01 – 20.
m.mmmmmme+mm represents the Absolute Intensity Multiplier Factor

Description

The command **setabsintmult@@m.mmmmmme+mm** directs the LED Analyser to set the Absolute Intensity Multiplier Factor on Fiber @@ to m.mmmmmme+mm

Example:

send the command **setabsintmult@@m.mmmmmme+mm** to the LED Analyser to instruct it to set the Absolute Intensity Multiplier Factor on Fiber 3 to 1

setabsintmult031.00000e+00
OK

Back to Index



User Calibration Mode

getABSINTMULT@@ Get the Absolute Int Correction Factor of the Led under test.

Transmit	Receive
getabsintmult@@	#.####e±##

Where:

@@ represents the Fiber Number and is a number in the range 01 – 20.
#.####e±## represents the Absolute Intensity Correction Factor

Description:

This command will allow the user to interrogate the Led Analyser to return the absolute Intensity Correction Factor or Multiplier Value for the Led under test at Fiber@.

Example:

Type the command **getabsintmult01** and the analyser will respond with the Multiplier in the format **0.00000e+00**

getabsintmult01
0.00000e+00

[Back to Index](#)



User Calibration Mode

setcalibration# - Select the user cal setting

Transmit	Receive
setcalibration# putcalibration#	OK OK

Where:

represents the set value (## can be 1 to 9).

Description

This command will select which calibration set you are accessing. The Led analyser is capable of storing nine (9) separate user Calibration sets on the board.

The command **SETCALIBRATION#** stores the Calibration sets both on RAM and on EEPROM (stays after power cycle but limits the number of set command operations to ~ 100K).

The command **PUTCALIBRATION#** only stores them on RAM and the analyser will default back to what values were stored in the EEPROM after a power cycle.

The default calibration set is 1

Example:

If you want to select set No.3

setcalibration3
OK

Example:

If you want to select set No.3

putcalibration3
OK

The command **GETCALIBRATIONSET** will return the number of which stored Calibration set of values are currently selected.

Example:

calibrationset3

Back to Index



User Calibration Mode

getCALIBRATIONset - Select the Calibration Set to use

Transmit	Receive
getcalibrationset	##

Where:

represents the set value (## can be 01 to 09).

Description

This command will retrieve which calibration set you are accessing. The Led analyser is capable of storing fifty one (09) separate user Calibration sets on the board.

The command **SETCALIBRATION##** stores the Calibration sets both on RAM and on EEPROM (stays after power cycle but limits the number of set command operations to ~ 100K).

The command **PUTCALIBRATION##** only stores them on RAM and the analyser will default back to what values were stored in the EEPROM after a power cycle.

The default calibration set is 00

Example:

If you want to know what set is active

getcalibrationset

03

Back to Index



User Calibration Mode

ResetUserCal - Reset the usercal setting to factory default

Transmit	Receive
resetusercal	OK

Description

This command will reset the calibration set back to factory default values. The Led analyser is capable of storing ten (00-09) separate user Calibration sets on the board. Each one must be reset separately.

The command **resetusercal** resets the active Calibration set to factory default.

The command resets:

- Absolute intensity factors to 0
- Relative intensity gains to 100
- xy and dominant wavelength offsets to 0
- User calibration date to 00/00/0000
- Reset RGB mode if selected and come back to standard mode
- Exposure factor to 1
- Log mode

of all the fibers in the user calibration set selected.

The command **GETCALIBRATIONSET** will return the number of which stored Calibration set of values are currently selected.

Example:

calibrationset03

If you want to reset calibration this set No.03

setcalibration03

OK

resetusercal

OK

[Back to Index](#)



User Calibration Mode

SetCalibrationDatedddmmyyyy - Set the user Cal Date

Transmit	Receive
Setcalibrationdatedddmmyyyy	OK

Where:

dd represents the 2 digit Day (01, 02....21, 30)
mm represents the 2 digit Month (01, 02...12)
yyyy represents a 4 digit Year. (2039)

Description

This command allows the user to input the Calibration Date for the Led Analyser. This command is used in conjunction with the user calibration set mode.

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

The command [getcalibrationdate](#) will display the current stored Calibration Date setting and calibration mode settings.

Example:

Set the user Calibration Date for August 1st 2015.

setcalibrationdate01082015
OK

The commands [getcalibrationdate](#) or [getstatus](#) will display the current stored user Calibration Date and calibration set mode.

Example:

```
.  
. calibrationset3  
user calibration date 01/08/2015
```

[Back to Index](#)



User Calibration Mode

getcalibrationdate **Get the Calibration Date set by the user.**

Description:

This command will allow the user to interrogate the Led Analyser to return the actual USER Calibration Date in the format dd/mm/yyyy. This is the date set by the user for a specific cal version.

Example:

Connect the Analyser to the Terminal Window and using the getcalibrationdate command, read the data for that Led. Type the command **getcalibrationdate** and the analyser will respond with with the user calibration date in the format **dd/mm/yyyy**

getcalibrationdate
01/01/2015

[*Back to Index*](#)



External Capture Mode

External Trigger Mode

There is an external capture function triggered by an external transition on the Feasa ICT units. A falling edge on the **pin 7** of the ICT connector will trigger a capture on the Led Analyser.

Set up the external capture

The following commands will activate the external capture.

Transmit	Receive
Setexternalcapture## Putexternalcapture## getexternalcapture	See details below

Where **##** is as follows:

- 00 Disable Trigger (Factory Default)
- 01 Capture1 or C1 (Range Low)
- 02 Capture2 or C2 (Range Medium)
- 03 Capture3 or C3 (Range High)
- 04 Capture4 or C4 (Range Super)
- 05 Capture5 or C5 (Range Ultra)
- 06 Capture or C (Range Auto)
- 10 Capturepwm or CPWM (Range Auto)
- 11 Capturepwm1 or CPWM1
- 12 Capturepwm2 or CPWM2
- 13 Capturepwm3 or CPWM3
- 14 Capturepwm4 or CPWM4
- 15 Capturepwm5 or CPWM5



Set up the data report

Transmit	Receive
Setexternaloutput#### Putexternaloutput#### getexternaloutput	See details below

Once the external capture has been finished, the analyser can output the data corresponding to a particular parameter (HSI, RGBI...). The commands to configure the data report are:

Where **####** is as follows:

No output **setexternaloutputnone** (Factory Default)

HSI **setexternaloutputhsi**

XYI **setexternaloutputxyi**

RGBI **setexternaloutputrgbi**

CCTI **setexternaloutputcct**

ABSINT **setexternaloutputabsint**

WSI **setexternaloutputwsi**

Once the external capture has been finished, the analyser can output the data corresponding to a particular parameter (HSI, RGBI...). The commands to configure the data report are:

Examples:

"**SETEXTERNALOUTPUTRGBI**": Red, Green, Blue and Intensity are reported for all the fibres automatically after an external trigger has been applied:

01 rrr ggg bbb iiii

02 rrr ggg bbb iiii

...

20 rrr ggg bbb iiii

The command "**GETEXTERNALOUTPUT**" returns the report mode selected.

[Back to Index](#)



Capture Sequence Mode

Sequence Capture Mode:

Introduction

The Sequence Capture functionality of the LED Analyser allows the user to take multiple captures in rapid succession and store them in the LED Analyser. The results of these captures can be downloaded at any time after the capture sequence has been completed.

This feature is useful when testing LEDs, that when combined, are generating a visual pattern that the user can see. An example of this would be 'soft start' lighting when entering or exiting a vehicle. Another example is when testing the 'visual sweep' on sequential Indicators in vehicles.

The number of captures that can be stored in memory depends on the number of fibers used during the test. If all 20 fibers of a 20 Channel Analyser are used the maximum number of captures that can be stored is 400. If a single fiber is used then the number of captures increases to 8000.

There are two ways to store commands in the Analyser – one way stores the command in RAM memory and the second way stores the command in Flash memory. When a command is stored in Flash memory it is not lost when power is removed from the Analyser. When the command is stored in RAM it will be lost when power is removed.

However, there is a limit to the number of write cycles to Flash memory so it is recommended to store commands that are being changed under program control in RAM memory.

Commands that begin with "**put**" will be stored in RAM while commands that begin with "**set**" will be stored in Flash memory.

All commands can use upper or lower case characters.



Capture Sequence Mode

This is a simple procedure to use the Capture Sequence Mode in the Led Analyser. Each command is detailed in the following section.

Capture Sequence Procedure:

Assumption: Blinking Led sequence to be tested here is blink 1 second on 1 second off.

Setup: Place the Led Analyser Fiber and Optical Head over the Leds to be tested.

1. Open Terminal window
2. Connect the Led Analyser.
3. Setup the sequence parameters, wait for the OK reply in each case..

Type -	putsequencemode#	(0 for All fibers, 1 for single fiber)
	putsequencechannel##	(## is Fiber Number, Use ONLY if 1 above)
-	putsequencecapture100	(100ms, Capture Time depends on the Led Intensity)
-	putsequencewait100	(100ms, Wait time between captures)
-	putsequencenumber0050	(50, The number of captures or samples to store)
4. Perform a capture

Type -	CS	(Capture Sequence, Based on the info above the total capture time here is 10seconds, (100ms+100ms)x 50captures))
--------	----	--
5. Retrieve the Data.

(If putsequencemode1)		
Type -	geths##	(## represents the fiber number) (This will return the Hue, Saturation, Intensity for Fiber No ##, getxyi##, getwi## etc can be used as required.)
(If putsequencemode0)		
Type -	gethsiall	(This will return the Hue, Saturation, Intensity for ALL fibers, getxyiall, getwiiall etc can be used as required.)
6. Interpret the data.
The results will be listed in the terminal window in columns for each fiber showing the ON/OFF data.
7. Type - getfrequency## (## represents the fiber number)
8. Interpret the data.
The result will show 000.5Hz 050.0% (Frequency in Hertz, Dutycycle in %)

Note1: There are other data interrogation commands listed below.

[Back to Index](#)



Capture Sequence Mode

Set/Put SequenceMode# - Set sequence mode for Fibers

Transmit	Receive
setsequencemode#	OK
putsequencemode#	OK

Where:

represents the sequence mode and is a number in the range 0 – 1.

Description

This command is used to set the number of fibers that will be used in the capture sequence command. PutSequencemode**0** will set **all** Fiber Channels for the Capture Sequence. PutSequencemode**1** sets a **single** Fiber Channel for the capture sequence.

If **putsequencemode1** is selected the Fiber Channel must be set using the **putSequenceChannel** command. Also, when this mode is selected up to 8000 captures can be stored in memory.

If putsequencemode**0** is selected the number of captures is limited to 400. **putsequencemode** commands are stored in RAM memory while **setsequencemode** commands are stored in Flash memory.

Example 1:

The PC transmits **putsequencemode0** to the LED Analyser to instruct it to do sequence capture on all Fibers. The LED Analyser will return a string **OK** to the PC.

putsequencemode0
OK

Example 2:

The PC transmits **putsequencemode1** to the LED Analyser followed by **putsequencechannel05** to instruct it to do sequence capture on Fiber No. 5. The LED Analyser will return a string **OK** to the PC.

putsequencemode1
OK
putsequencechannel05
OK

To determine which sequence mode is set, use the command **getsequencemode**. It will return a 1 or a 0.

getsequencemode
0 (All Fibers)

[Back to Index](#)



Capture Sequence Mode

Set/Put SequenceChannel@@ - Select sequence mode Fiber No.

Transmit	Receive
setsequencechannel@@	OK
putsequencechannel@@	OK

Where:

@@ represents the Fiber and is a number in the range 01 -20.

Description

This command is used to set the Fiber number associated with the **Capture Sequence** command. The command **putSequencemode1** should be executed before this command.

In addition, the commands **putsequencecapture**, **putsequencewait** and **putsequencenumber** should be issued before this command.

The Analyser can store up to 8000 captures in memory when this mode is selected.

putsequencechannel commands are stored in RAM memory while **setsequencechannel** commands are stored in Flash memory.

Example:

The PC transmits **putsequencechannel04** to the LED Analyser to instruct it to do sequence capture on Fiber 04 only. The LED Analyser will return a string **OK** to the PC.

putsequencechannel04
OK

To determine which Fiber Number is set, use the command **getsequencechannel**. It will return a number in the range 01 to 20.

getsequencechannel
05 (Fiber Number 5)

[Back to Index](#)



Capture Sequence Mode

Set/Put SequenceCapture##(#) - Set Sequence Capture Time

Transmit	Receive
setsequencecapture##	OK
putsequencecapture##	OK
setsequencecapture###	OK
putsequencecapture###	OK

Where:

represents the capture time in milliseconds (01 – 99 msec)

represents the capture time in milliseconds (001 – 999 msec)

Description

This command is used to set the Capture Time. The Capture Time can be set to any value in the range 001 – 999 milliseconds. This capture time is set for all Fiber Channels enabled by the **setSequencemode** or **putSequencemode** command.

Some experimentation may be required to determine the correct Capture Time because it depends on the brightness level of the LED(s) being tested. If the Capture Time is too short the Analyser may indicate an under-range condition and if the time is too long it may cause an over-range to be reported.

PutSequencecapture commands are stored in RAM memory while **SetSequencecapture** commands are stored in Flash memory.

Example:

The PC transmits **putsequencecapture05** or **putsequencecapture005** to the LED Analyser to instruct it to set the Capture Time to 5 msec. The LED Analyser will return a string **OK** to the PC.

putsequencecapture005

OK

To determine which Capture Time is set, use the command **getsequencecapture** It will return a time in milliseconds.

getsequencecapture

005 ms

(5 milliseconds)

[Back to Index](#)



Capture Sequence Mode

Set/Put SequenceWait##(##) - Set Sequence Wait Time

Transmit	Receive
setsequencewait##	OK
putsequencewait##	OK
setsequencewait###	OK
putsequencewait###	OK

Where:

represents the wait time in milliseconds (00 – 99 msec)

represents the wait time in milliseconds (000 – 999 msec)

Description

This command is used to set the *Wait* time between captures. The *Wait* time can be set to any value in the range 000 – 999 milliseconds. This *Wait* time is set for all Fiber Channels enabled by the **setSequencemode** or **putSequencemode** command.

PutSequencewait commands are stored in RAM memory while **SetSequencewait** commands are stored in Flash memory.

Example:

The PC transmits **putsequencewait10** to the LED Analyser to instruct it to set the wait time between captures to 10 msec. The LED Analyser will return a string **OK** to the PC.

putsequencewait10
OK

To determine what Wait Time between Sequence Captures is set to, use the command **getsequencewait**. It will return a time in milliseconds.

getsequencewait

005 ms (5 milliseconds)

[Back to Index](#)



Capture Sequence Mode

Set/Put SequenceNumber###(##) - Set the Number of Captures

Transmit	Receive
putsequencenumber###	OK
setsequencenumber###	OK
putsequencenumber####	OK
setsequencenumber####	OK

Where:

represents the number of captures to store in the range 000 - 999.

represents the number of captures to store in the range 000 - 9999.

Description

This command is used to set the number of captures or samples to store. If the **setsequencemode** or **putsequencemode** is set to 0 then the number of samples that can be stored is 400.

If the **setsequencemode** or **putsequencemode** is set to 1 then the number of stored samples increases to 8000.

PutSequencenumber commands are stored in RAM memory while **SetSequencenumber** commands are stored in Flash memory.

Example:

The PC transmits **putsequencenumber050** to the LED Analyser to instruct it to store 50 captures when the **CaptureSequence** command is issued. The LED Analyser will return a string **OK** to the PC.

putsequencenumber050

OK

To determine the number of Sequence Captures stored, use the command **getsequencenumber** It will return a number in the range 0 - 9999

getsequencenumber

0250 (250 Captures)

[Back to Index](#)



Capture Sequence Mode

Set/Put SequenceThreshold##### - Set the Intensity Threshold

Transmit	Receive
putsequencethreshold#####	OK
setsequencethreshold#####	OK

Where:

represents the Intensity threshold.

Description

This command is used to set an Intensity threshold so that Intensities above the threshold will be converted to binary 1's and Intensities below the threshold will be converted to binary 0's.

This binary data is used to generate a timing sequence for LEDs that are being switched on and off such as in an automotive Indicator. The timing sequence can be read out using the command **getsequencetimes**.

PutSequencethreshold commands are stored in RAM memory while
SetSequencethreshold commands are stored in Flash memory.

Example:

The PC transmits **setsequencethreshold09500** or **putsequencethreshold09500** to the LED Analyser to instruct it to treat intensities greater than 9,500 as binary 1 and intensities less than 9,500 as binary 0. The LED Analyser will return a string **OK** to the PC.

putsequencethreshold09500
OK

To determine the minimum Intensity level acceptable as a PASS, use the command **getsequencethreshold**. It will return an Intensity Value in the range 0 - 99999

getsequencethreshold
9500 (Min Intensity Level 9500)

[Back to Index](#)



Capture Sequence Mode

Set/Put SequenceStartDelay### - Set delay in ms before sequencecapture starts

Transmit	Receive
putsequencestartdelay###	OK
setsequencestartdelay###	OK

Where:

represents the delay in milliseconds before executing a **capturesequence** and is in the range 001-999ms.

Description

This command is used to set a delay in milliseconds before the capturesequence starts. This can be useful when using the external trigger mode.

PutSequencestartdelay commands are stored in RAM memory while **SetSequenceStartDelay** commands are stored in Flash memory.

Example:

The PC transmits **setsequencestartdelay100** or **putsequencestartdelay100** to the LED Analyser to instruct it to wait 100ms before starting the capturesequence. The LED Analyser will return a string **OK** to the PC.

putsequencestartdelay100
OK

To determine the delay in milliseconds before the sequence capture is initiated use the command **getsequencestartdelay** It will return an time in the range 000 - 999ms

getsequencestartdelay

123 ms (Delay before Sequence Capture is 123ms)

[Back to Index](#)



Capture Sequence Mode

CaptureSequence - Start the Captures

Transmit	Receive
Capturesequence or CS	OK

Description

This command starts the sequence capture function. The LED Analyser will capture and store the Intensity results of each capture up to the number of captures specified by the **setsequencenumber or putsequencenumber** command.

The execution time for this command will depend on the time set for each capture, the wait time between captures and the number of captures.

Example:

The PC transmits **capturesequence or cs** to the LED Analyser to instruct it to begin capturing data. The LED Analyser will return a string **OK** to the PC.

CS
OK

[Back to Index](#)



Capture Sequence Mode

StoreSequence## - Store the sequence pattern

Transmit	Receive
storesequence##	OK

Where:

represents a sequence pattern number and is a value in the range 00 - 20.

Description

This command stores the sequence pattern from the last **capturesequence** command. The pattern is identified by a number in the range 00 – 20. Up to 21 patterns can be stored. This feature allows the Analyser to identify patterns by comparing them against patterns stored in memory.

The **setsequencemode#**, **setsequencechannel@@**, **setsequencenumber###**, **sequencecapture###**, **sequencenumber**, **sequencethreshold#####**, **sequencestartdelay###** and **sequencewait###** commands must be executed before this command takes effect

The **StoreSequence** commands are stored in Flash memory.

Example:

The PC transmits **storesequence01** to the LED Analyser to instruct it to store the current pattern under the pattern number 01. The LED Analyser will return **OK** to the PC.

storesequence01
OK

[Back to Index](#)



Capture Sequence Mode

IdentifySequence - Identify the current sequence pattern

Transmit	Receive
identifysequence	##

Where:

represents the number of a stored sequence pattern and is a value in the range 00 – 20 which compares to the current sequence capture executed.

Description

This command compares the current sequence pattern with a stored list of patterns and identifies it. The pattern is identified by a number in the range 00 – 20. Up to 21 patterns can be stored. This feature allows the Analyser to identify patterns by comparing them against patterns stored in memory.

The **setsequencemode#**, **setsequencechannel@@**, **setsequencenumber###**, **sequencecapture###**, **sequencenumber**, **sequencethreshold####**, **sequencestartdelay###** and **sequencewait###** commands must be executed before this command takes effect. Also, a **capturesequence** command must be issued before the **identifysequence** command.

Example:

The PC transmits **identifysequence** to the LED Analyser to instruct it to identify the current pattern and return its identity number.

identifysequence
04

[Back to Index](#)



Capture Sequence Mode

ResetSequence## - Reset a sequence pattern

Transmit	Receive
Resetsequence## resetsequenceall	OK

Where:

represents the number of a sequence pattern and is a value in the range 00 - 20.
all represents the all sequence patterns stores in the Analyser. (max 21 patterns)

Description

This command resets the sequence pattern number **##**. The pattern is identified by a number in the range 00 - 20. Up to 21 patterns can be stored.

The sequence pattern **##** will be deleted from memory.

Example:

The PC transmits **resetsequence01** to the LED Analyser to instruct it to delete sequence pattern 01.

resetsequence01
OK

[Back to Index](#)



Capture Sequence Mode

GetSequenceStored## - Get a sequence pattern

Transmit	Receive
getsequencestored##	OK

Where:

Represents the Sequence pattern stored previously.

Description

The pattern is identified by a number in the range 00 – 20. Up to 21 patterns can be stored. This feature allows the user to recall a stored pattern from memory. If you have stored x different patterns you can recall any one if you know the pattern number.

Example:

getsequencestored05 will recall the sequencecapture number 05.

[Back to Index](#)



Capture Sequence Mode

GetSequence@@ - Get the binary sequence of Intensities

Transmit	Receive
getsequence@@	Binary data 0101..

Where:

@@ represents the Fiber number to be interrogated and is a value in the range 01 - 20.

Description

This command returns the Intensity values for Fiber# # represented as a binary number. Intensities greater than the threshold value are converted to binary 1's while Intensities less than the threshold are converted to binary 0's.

The threshold value is set by the command **putsequencethreshold** or **setsequencethreshold**. The number of Intensities returned is set by the **putsequencenumber** or **setsequencenumber** command.

Example:

The PC transmits **getsequence01** to the LED Analyser to instruct it to output a binary sequence representing the Intensities that have been captured. The LED Analyser will return a **binary sequence** to the PC.

getsequence01

0
1
1
.
.

[Back to Index](#)



Capture Sequence Mode

GetSequenceTimes@@ - Get the Duty Cycle of the Sequence

Transmit	Receive
getsequencetimes@@	Toff1 Ton Toff2

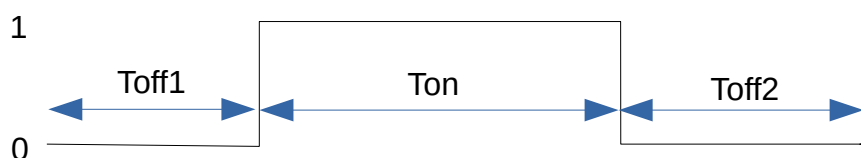
Where:

@@ represents the Fiber number and is a value in the range 01 – 20.

Toff1 represents the time the LED is off at the beginning of the sequence.

Ton represents the time the LED is on during the sequence.

Toff2 represents the time the LED is off at the end of the sequence.



Description

This command returns three time intervals for Fiber##. The times are in milliseconds and represent the time the LED is 'off' at the beginning, the 'on' time and the 'off' time at the end of the sequence. A valid **capturesequence** command must be issued before this command in order to get valid data. The resolution of the times Toff1, Ton and Toff2 depends on the sequence capture and sequence wait time settings.

Example:

The PC transmits **getsequencetimes01** to the LED Analyser to instruct it to output the time intervals for Fiber 01.

getsequencetimes01

Toff1 = 00350ms Ton = 00340 ms Toff2 = 76910ms



Capture Sequence Mode

GetSequenceTimesAll - Get the Duty Cycle of the Sequence for all fibers

Transmit	Receive
getsequencetimesall	Toff1 Ton Toff2

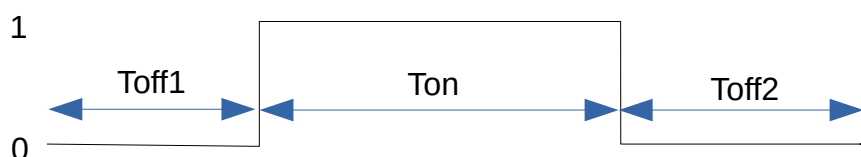
Where:

ALL represents the total number of Fibers used in the sequence capture.

Toff1 represents the time the LED is off at the beginning of the sequence.

Ton represents the time the LED is on during the sequence.

Toff2 represents the time the LED is off at the end of the sequence.



Description

This command returns three time intervals for each Fiber. The times are in milliseconds and represent the time the LED is 'off' at the beginning, the 'on' time and the 'off' time at the end of the sequence. A valid **capturesequence** command must be issued before this command in order to get valid data. The resolution of the times Toff1, Ton and Toff2 depends on the sequence capture and sequence wait time settings.

Example:

The PC transmits **getsequencetimesall** to the LED Analyser to instruct it to output the time intervals for Fiber 01, Fiber 02, Fiber 03, Fiber 04,

getsequencetimesall

Toff1 = 00350ms Ton = 00340 ms Toff2 = 76910ms

Toff1 = 00350ms Ton = 00340 ms Toff2 = 76910ms

Toff1 = 00350ms Ton = 00340 ms Toff2 = 76910ms

Toff1 = 00350ms Ton = 00340 ms Toff2 = 76910ms

.
.
.
.

[Back to Index](#)



Capture Sequence Mode

GetFrequency## - Returns the frequency and duty cycle of Led

Transmit	Receive
getfrequency##	Fff.f Hz xxx.x %

Where:

represents the Fiber Number and is a value in the range 01 - 20.
fff.f Hz represents the Frequency in Hertz.
Xxx.x % represents the Duty Cycle in Percentage.

Description

This command reports the Frequency in Hertz and the Duty Cycle in % of the Led under test after the capture is complete.

Example:

The PC transmits **getfrequency01** to the LED Analyser to instruct it to report the Frequency and duty cycle of fiber 1

getfrequency01
123.4Hz 067.8%

[Back to Index](#)



Capture Sequence Mode

Get(Max/Min/Avg)Intensity## - Returns the Max/Min or Avg Intensity for the Seq Cap.

Transmit	Receive
getmaxintensity## getminintensity## getavgintensity##	Iiii @xxxxx ms iiii @xxxxx ms iiii

Where:

represents the Fiber Number and is a value in the range 01 – 20.
iiii represents the intensity value of the LED under fiber ##. This 5-digit number is in the range 0000 – 99999. 0000 represents no Intensity or under range (i.e. the LED is off) and 99999 will represent over range or the LED is too bright.
xxxxx ms Represents the time in ms after the sequence capture has started, when the maximum (or minimum) intensity measurement has been taken.

Description

This command reports the Maximum, Minimum or Average Intensity of the Led under test after the sequence capture is complete.

Example:

The PC transmits **getmaxintensity01** or **getminintensity01** or **getavgintensity** to the LED Analyser to instruct it to report the Maximum or Minimum or Average Intensity of fiber No.1

getmaxintensity01
99800 @ 00600 ms

getminintensity01
00200 @ 00300 ms

getavgintensity01
05650

[Back to Index](#)



Capture Sequence Mode

GetFlicker## - Returns the Flicker Index and frequency after a Seq Cap.

Transmit	Receive
getflicker##	xxx.x %, y.yyyyyy, fff.f Hz

Where:

represents the Fiber Number and is a value in the range 01 – 20.
xxx.x represents the Flicker Duty Cycle in Percentage.
y.yyyyyy represents the Flicker Index.
fff.f Hz represents the Flicker Frequency in Hertz.

Description

This command reports the Flicker Duty Cycle, Flicker Index and Frequency of the Led under test after the sequence capture is complete.

Example:

The PC transmits **getflicker01** to the LED Analyser to instruct it to report the Flicker Index Frequency of fiber No.1

getflicker01

Flicker : 100.0% Flicker index: 0.283214 Flicker Frequency: 100.0 Hz

[Back to Index](#)



Capture Sequence Mode

GetSequenceMode# - Get sequence mode for Fibers

Transmit	Receive
getsequencemode#	0 or 1

Where:

represents the sequence mode and is a number in the range 0 – 1.

To determine which sequence mode is set, use the command **getsequencemode**. It will return a 1 or a 0.

getsequencemode

1 (All Fibers)

getsequencemode

0 (Individual Fibers)

[Back to Index](#)



Capture Sequence Mode

GetSequenceChannel - Select sequence mode Fiber No.

Transmit	Receive
getsequencechannel	##

Where:

represents the Fiber Number and is in the range 01 -20.

To determine which Fiber Number is set, use the command **getsequencechannel**. It will return a number in the range 01 to 20.

getsequencechannel

05 (Fiber Number 5)

[Back to Index](#)



Capture Sequence Mode

GetSequenceCapture - Get Sequence Capture Time

Transmit	Receive
getsequencecapture	xx ms or xxx ms

Where:

represents the capture time in milliseconds (01 – 99)

represents the capture time in milliseconds (001 – 999)

To determine which Capture Time is set, use the command **getsequencecapture** It will return a time in milliseconds.

getsequencecapture

005 ms (5 milliseconds)

[Back to Index](#)



Capture Sequence Mode

GetSequenceWait - Get Sequence Wait Time

Transmit	Receive
getsequencewait	xx ms or xxx ms

Where:

xx represents the wait time in milliseconds (00 – 99)

xxx represents the wait time in milliseconds (000 – 999)

To determine what Wait Time between Sequence Captures is set to, use the command **getsequencewait** . It will return a time in milliseconds.

getsequencewait

005 ms (5 milliseconds)

[Back to Index](#)



Capture Sequence Mode

GetSequenceNumber - Get the Number of Captures

Transmit	Receive
getsequencenumber	xxxx

Where:

xxxx represents the number of captures stored.

To determine the number of Sequence Captures stored, use the command **getsequencenumber** It will return a number in the range 0 - 9999

getsequencenumber

0250 (250 Captures)

[Back to Index](#)



Capture Sequence Mode

GetSequenceThreshold - Get the Intensity Threshold

Transmit	Receive
getsequencethreshold	xxxxxx

Where:

xxxxxx *represents the Intensity threshold.*

To determine the minimum Intensity level acceptable as a PASS, use the command **getsequencethreshold** . It will return an Intensity Value in the range 0 - 99999

getsequencethreshold

9500 (Min Intensity Level 9500)

[Back to Index](#)



Capture Sequence Mode

GetSequenceStartDelay - Get the delay before the CS command

Transmit	Receive
getsequencestartdelay	xxx

Where:

xxx represents the delay in milliseconds

To determine the delay in milliseconds before the sequence capture is initiated use the command **getsequencestartdelay** It will return a time in the range 000 - 999ms

getsequencestartdelay

123 ms (Delay before Sequence Capture is 123ms)

[Back to Index](#)



Capture Sequence Mode

Capture Sequence Sync Serial Commands

The following table lists the sequence capture commands that can be executed through the Synchronous Serial Port.

These commands are in addition to the Synchronous Serial commands described above in **Figure 19** in the Synchronous Serial Section

All data transmitted and received is in a binary format. The Most Significant Bit (MSB) is transmitted/received first.

Command	Data Transmitted	Data Received
Clear Comms Buffer	0xFF (1 byte)	
Initialize the LA	0xAA (1 byte)	RyBy_bar goes high then low
Capture Sequence setup	0xC0+Capture Time (2 byte)+ Wait Time (2 byte) + No of captures (2 bytes)	RyBy_bar goes high then low
Capture Sequence Intensity Threshold	0xC1+ Intensity (3 bytes)	RyBy_bar goes high then low
Execute Capture Sequence	0x45	RyBy_bar goes high then low
Read Intensity Capture Sequence	0x91+Fiber num (1 byte)+ Sequence Capture num (2bytes)	Intensity (3 bytes)
Read Pulse Times	0x92+Fiber num (1 byte)	Time Off Beginning (2 bytes) Time On (2 bytes) Time Off End (2 bytes)

Figure 29.

Note:- The code 0xD5 indicates the command completed correctly. If the code 0xEE is received then an error has occurred and the command should be re-transmitted.

[Back to Index](#)



Capture Sequence Mode

Capture Sequence Setup.

Initialize the LED Analyser and make it ready for a *Capture* command. The sequence for this command is shown in Figure 30.

Binary 10101010 (0xAA) is clocked into the LED Analyser while the Addr0/LA_Select signal is held high. The command is clocked in on the falling edge of *SCK*.

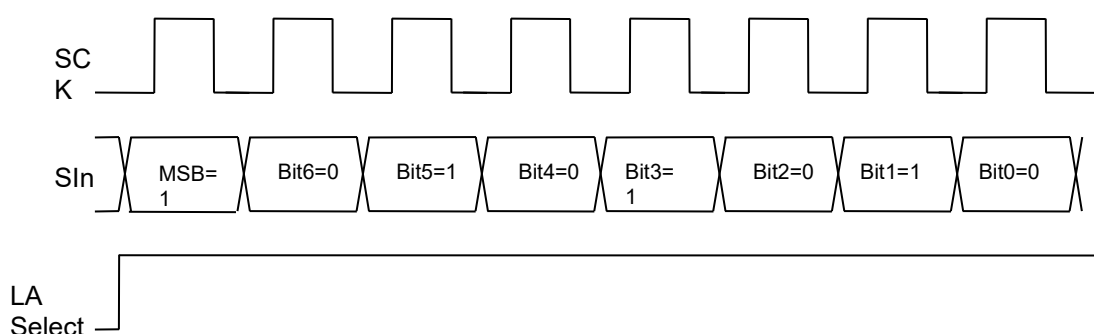


Figure 30.

The signal Addr0/LA_Select must be driven high while 0xAA is clocked into the Analyser.

The relevant data is then clocked into the Analyser, Capture Time, Capture Wait, Number of Captures.

The code for this command is 0xC0 – this must be transmitted to the LED Analyser followed by the Required Capture Time in mSec + the Required Wait Time in mSec + the Number of captures. Each of these requires 2 Bytes. The Capture Time is a time from 1mS to 999mS. The Wait Time is a time from 1mS to 999mS. The Number of Captures possible for all 20 fibers is 400, while the number of captures available for a single fiber is 8000.

Command	Data Transmitted	Data Received
Capture Sequence Setup	0xC0 + Capture Time (2bytes) + Wait time (2Bytes) + No of Captures (2 Bytes)	

Setup Command: **00xC0** (byte0)

Capture Time: **001 – 999 mSec** (byte1 + byte2)

Wait Time: **001 – 999 mSec** (byte3 + byte4)

Number of Captures: **400 max(20 fibers) or 8000 max (1 Fiber)** (byte5 + byte6)



Capture Sequence Mode

This command will instruct the LED Analyser to capture and store all the LED Intensity data. Setting the Capture Time in the previous setup section will determine the best Intensity results from the test. In RS232 mode there are five manual ranges designated Range1 (Low), Range2 (Medium), Range3 (High), Range4 (Super) and Range5 (Ultra). Range1 has the highest sensitivity for low light conditions and the sensitivity is reduced progressively to Range5 which has the lowest sensitivity. This means that for brighter LED's the higher Ranges should be used. In sequence capture mode you set the time in mSec to achieve this.

Typical CS Capture Times are as follows:

Range	Capture Time
Range 1 low	650ms
Range 2 medium	200ms
Range 3 high	22ms
Range 4 super	4ms
Range 5 ultra	2ms
Auto Range	350ms

The code 0xD5 indicates a successful completion of the command. If an error occurs then the code 0xEE will be received. The command should then be re-transmitted.

[Back to Index](#)



Capture Sequence Mode

Step-by Step Sequence for LED Analyser CS Setup

1. Initialize the board

- Set Addr0 = 0
- Set Clock_in = 0
- Set Data_in to 0

2. Clear Comms Buffer

- Set Data_In to 1
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit

3. Initialize the capture

- Set Data_in to 10101010 (0xAA)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
- Wait 1ms
- Wait until RyBy_bar goes to 0
- RyBy_bar goes to 1
- Set Addr0 = 0

4. Send setup data to the Analyser

- (0xC0) Sequence Capture
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
- (0xCapture Time High Byte)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
- (0xCapture Time Low Byte)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
- (0xWait Time High Byte)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
- (0xWait Time Low Byte)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
- (0xSequence Number High Byte)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
- (0xSequence Number High Byte)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
- Wait until RyBy_bar goes to 0
- Set Addr0 = 0

[Back to Index](#)



Capture Sequence Mode

Step-by Step Sequence for LED Analyser CS Capture

1. Initialize the board

- Set Addr0 = 0
- Set Clock_in = 0
- Set Data_in to 0

2. Clear Comms Buffer

- Set Data_In to 1
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit

3. Initialize the capture

- Set Data_in to 10101010 (0XAA)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
- Wait 1ms
- Wait until RyBy_bar goes to 0
- RyBy_bar goes to 1
- Set Addr0 = 0

4. Initialize to capture Intensity

- Set Data_in to 01000101 (0x45)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit

5. Read back the Capture Acknowledge as follows

- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
- Read back 0xD5 (8 bits after each falling edge of the clock)
- (0xD5 is positive response and 0xEE is error response)

6. Check that RyBy_bar is high

7. Check that RyBy_bar goes low after the capture time.

8. Send the Capture Status command (0xBB)

- Set Data_in to (0xBB) (10111011)
- Set Clock_in to 1 and 0 repeat [8] times to clock in each bit
- Set Addr0 = 0



Step-by-Step approach to Testing a LED

Standard Capture Mode

To Test the colour and Intensity of up to 20 LEDs simultaneously carry out the following:

1. Ensure that the LEDs to be Tested are turned on and that the fibers are centered over the LEDs.
2. To Test the colour and Intensity in **RGB** first send the command **capture** to the LED Analyser. The LED Analyser will return the characters **OK** indicating that the colour and Intensity data for the LEDs has been stored in the internal memory.
3. Transmit the command **getrgbi01** to retrieve the results for Fiber 1. Any fiber can be queried by sending the command **getrgbixx** to the LED Analyser, where **xx** is the fiber number in the range **01 to 20**. The LED Analyser will return the results in the format **rrr ggg bbb iiiiiE** where **rrr**, **ggg** and **bbb** are the **red**, **green** and **blue** components of the light in decimal format in the range **000 to 255**. The **iiii** value indicates the intensity value of the light in the range **00000 to 99,999**.
4. Alternatively, the Led Analyser may also be queried to retrieve the Hue, Saturation and Intensity results for the LEDs under test. After step 2 send the command **gethsixx** to the LED Analyser where **xx** is the fiber number. The LED Analyser will return the results for that fiber in the format **hhh.hh sss iiiiiE** where **hhh.hh** is the Hue, **sss** is the Saturation(whiteness) and **iiii** are the Intensity results for that Fiber.
5. An Intensity value of 0 will indicate that the LED under test is not bright enough. To compensate for this move the fiber closer to the LED or increase the LED intensity. An Intensity value of 99,999 will indicate that the LED is too bright and the LED Analyser has an over-range condition. In this case increase the distance of the LED to the Fiber or decrease the LED Intensity.
6. In general, try to keep the Intensity less than 80,000 and greater than 20,000.

[Back to Index](#)



Effect of PWM on Intensity Testing

The effect of Pulse Width Modulation (PWM) of a typical LED on the Analyser Intensity can be seen in Figure 31. In this graph the LED will always be on at 100% modulation. The Intensity has been normalised to 100% at 100% modulation. When the modulation has been reduced to 50% (i.e. the LED is off 50% of the time) the relative Intensity drops to approximately 80%.

At 20% modulation (the LED is off 80% to the time) the relative Intensity drops to approximately 50%.

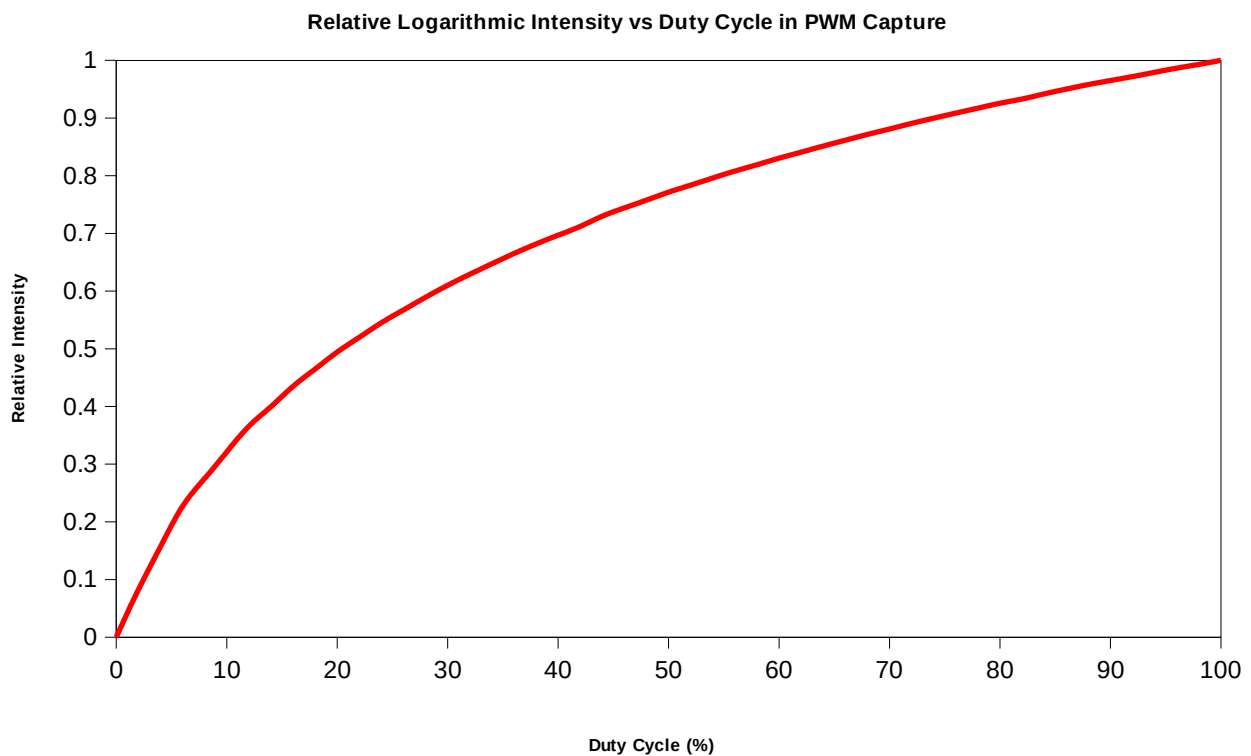


Figure 31: Normalized Intensity versus Modulation

[Back to Index](#)



Step-by-Step approach to Testing a PWM LED

PWM LED Mode

To test PWM LEDs use the following commands on the LED Analyser for Fiber 1:

1. Decide how many readings are required to test the PWM LED's. The Analyser can be programmed to take between 1 and 15 readings. The more readings that are taken the greater the stability of the results.
2. Send the command **CAPTURE#PWM@@**, where @@ is the number of readings to take and # is the Intensity range. The LED Analyser will capture and store the colour, Saturation and Intensity data for all fibers. The LED Analyser will respond with the Characters **OK** indicating that the command has been completed.
3. To read back the RGB and Intensity of the LED under fiber **xx** send the command **GETRGBIxx** to the LED Analyser. The LED Analyser will return the data in the format **rrr ggg bbb iiii** where **rrr**, **ggg** and **bbb** are the **red**, **green** and **blue** components of the colour. The **iiii** value indicates the intensity value.
4. To read back the Hue, Saturation and Intensity of the LED under fiber **xx** send the command **GETHSIxx** to the LED Analyser. The LED Analyser will return the data in the format **hhh.hh sss iiii** where **hhh.hh** represents the Hue value, **sss** represents the Saturation(whiteness) and **iiii** indicates the Intensity value.

[Back to Index](#)



Testing a 7-Segment Display

The LED Analyser can be used to test LED-based 7-Segment displays. To set up the LED Analyser to interrogate a single 7-Segment display, fit fibers labeled 1 to 7 over segments a-g on the 7-Segment display.

To set up the LED Analyser to interrogate an additional 7-Segment display, fit fibers labeled 11 to 17 over segments a-g on the additional display.

To interrogate the digit displayed on the first 7-Segment Display send the command **get7seg1** to the LED Analyser. The LED Analyser will return the digit displayed. The LED Analyser will return the character **X** when the displayed value is not recognized (**0 to 9**).

To interrogate the digit displayed on the second 7-Segment display send the command **get7seg2** to the LED Analyser. The LED Analyser will return the digit displayed. Again, the LED Analyser will return the character **X** when the displayed value is not recognized (**0 to 9**).

Note:- it is not necessary to send any *capture* commands prior to using the **get7seg1** or **get7seg2** commands.

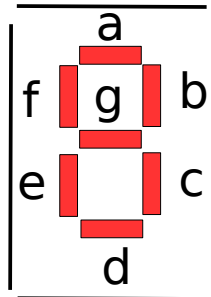


Figure 32. 7 Segment Display

[Back to Index](#)

FEASA LED ANALYSER ICT VERSION



Command Summary:

Capture Commands	Page	DESCRIPTION
capture or c	70	Measure Static LED Data - Auto Range
capture# or c#	71	Measure Static LED Data - Manual Range(1-5)
capturepwm or cpwm	72	Measure PWM LED Data - Auto Range
capture#pwm or c#pwm	73	Measure PWM LED - Manual Range(1-5), Auto Average (7)
capture#pwm@@ or c#pwm@@	73	Measure PWM LED - Manual Range(1-5), Manual Average(1-15)
capturesequence or cs	140	Measure LED Data - Sequence Mode

Get Led Data Commands	Page	DESCRIPTION
getrgbi## or getrgbiall	76	Get RGB, Saturation and Intensity for LED/LEDS
gethsi## or gethsiall	77	Get Hue, Saturation and Intensity for LED/LEDS
getxy## or getxyall	78	Get the xy Chromaticity values for LED/LEDS
getxyi## or getxyiall	79	Get the xy Chromaticity and Intensity values for LED/LEDS
getcixyz##	87	Get the XYZ data for LED/LEDS
getuv## or getuval	80	Get the u'v' Chromaticity values for LED/LEDS
getwavelength## or getwavelengthall	81	Get Dominant Wavelength for LED/LEDS
getwi## or getwiall	82	Get Wavelength & Intensity for LED/LEDS
getwsi## or getwsiall	83	Get Wavelength, Saturation & Intensity for LED/LEDS
getcct## or getcctall	84	Get Colour Temperature for LED/LEDS
getintensity## or getintensityall	85	Get the Intensity for LED/LEDS
getabsint@@ or getabsintall	86	Get the Absolute Intensity for a LED / LEDS
getsignallevel## or getsignallevelall	88	Get the Intensity of the LED/LEDS in % mode
get7seg#	89	Get the value of a 7segment display

Continuous Frequency Mode	Page	DESCRIPTION
setcontinuousmode	65	Enable Continuous Frequency Mode
setcontinuouschannel##	65	Select which Channel / Fiber No. you require
setcontinuouscapture##	65	Select which capture range you require
getcontinuousmode	66	Enable Continuous Frequency Mode
getcontinuouschannel	66	Select which Channel / Fiber No. you require
getcontinuouscapture	66	Select which capture range you require

General Control Commands	Page	DESCRIPTION
setfactor / getfactor	104/91	Set / Get the Sensitivity Level (Exposure Factor) of the Analyser
setautopwm / getautopwm	109/92	Set / Verify the Analyser is in Auto capture or Autopwm capture mode
setphotopic / getphotopic	112/93	Set / Verify which photopic mode is set 0, 1 or 2
setLOG / setLIN	105/106	Set the Intensity Response to Logarithmic Mode
setfrequencyxxx	110	Set the desired Frequency output mode of the Led Analyser
Setsyncserialx	111	Set the Synchronous Serial Mode on the Led Analyser ON/OFF
getintensitymode	94	Verify which Intensity mode is active Logarithmic or Linear

[Back to Index](#)

FEASA LED ANALYSER

ICT VERSION



System Commands	Page	DESCRIPTION
gethw	98	Get the Hardware Version of the Analyser
getstatus	100	Get the summary details of a Led Analyser
getICTstatus	101	Get the ICT summary details of a Led Analyser
getserial	95	Get the Serial Number of the Analyser
getversion	96	Get the Firmware Revision of the Analyser
getnfb	99	Get the Number of Channels on a Led Analyser
setbaud / getbaud	108/97	Set / Get the Baud Rate the Analyser is set to
EnableEOT	102	Enable End of line transmission character
DisableEOT	102	Disable End of line transmission character

UserCal Commands	Page	DESCRIPTION
set / getINTGAIN	114/115	Set / Get the Intensity Gain of the Led Analyser
set / getXOFFSET	116/118	Set / Get the X Chromaticity Offset of the Led Analyser
set / getYOFFSET	117/119	Set / Get the Y Chromaticity Offset of the Led Analyser
Set / getwavelengthOFFSET# #	120/121	Set / Get the Wavelength Offset of the Led Analyser
Set / getabsintmult@@m.mmmme+mm	122/123	Set / Get the Absolute Intensity Correction Factor of the Led under test.
setcalibration# #	124	Set the Calibration version of the USER
getcalibrationset	125	Return the set number of the activated calibration
resetusercal	126	Reset the active calset usercal parameters to the factory defaults
setCalibrationDateddmmYYYY	127	Set the Calibration Date of the USER
getcalibrationdate	128	Get the Calibration Date set by the user

External Trigger Commands	Page	DESCRIPTION
setexternalcapture# # # #	129	Set the External Trigger Capture mode
getexternalcapture	129	Verify what capture mode the external trigger is set at.
setexternaloutput# # # #	130	Set the External Trigger Output mode
getexternaloutput	130	Verify what capture mode the external trigger output is set at.

[Back to Index](#)

FEASA LED ANALYSER ICT VERSION



Sequence Capture Mode Commands	Page	DESCRIPTION
setSequenceMode#	133	Set the No. of Fibers in the CS Mode -0 is ALL -1 is individual Channel
getsequencemode#	151	Determine which sequence mode is set -0 is ALL -1 is individual Channel
setSequenceChannel@@	134	Set the Channel No's for the Capture Sequence Mode
getsequencechannel	152	Determine which Channel No's for the Capture Sequence Mode
setSequenceCapture###	135	Set the Capture Time in Capture Sequence Mode (CS)
getsequencecapture	153	Determine which capture time is set in CS
setSequenceWait###	136	Set the WAIT time between Captures in CS
getsequencewait	154	Determine what is the wait time between captures in CS
setSequenceNumber####	137	Determine the NUMBER of Captures to store in CS
getsequencenumber	155	Determine the number of captures to store in CS
setSequenceThreshold#####	138	Set the Intensity Threshold in CS
getsequencethreshold	156	Determine a Minimal Intensity level acceptable as a PASS in CS
getSequence@@	145	Get the Intensity Value in 1's or 0's based on threshold set by user
setsequencestartdelay###	139	Set a delay time in mSec before executing a CS
getsequencestartdelay	157	Get the delay time set before executing a CS
getSequenceTimes@@	146	Get the OFF.. ON.. OFF times of the Fiber for sequence measured
getSequenceTimesALL	147	Get the OFF.. ON.. OFF times of ALL Fibers for sequence measured
StoreSequence##	141	Stores the parameters of the active CS into memory
getsequencestored##	144	Retrieve a previously stored measure sequence
identifysequence	142	Compare the current CS pattern to already Stored user CS patterns
resetsequence##	143	Delete the designated CS Pattern from memory
resetsequenceALL	143	Delete ALL CS Patterns from memory
GetMaxIntensity##	149	Get the Maximum Intensity recorded in Sequence Capture Mode
GetMinIntensity##	149	Get the Minimum Intensity recorded in Sequence Capture Mode
getavgIntensity##	149	Get the Average Intensity recorded in Sequence Capture Mode
GetFlicker##	150	Get the Flicker Index and Frequency after a Sequence Capture
Getfrequency##	148	Get the Frequency and Duty Cycle of the Led under test in Sequence Capt.

[Back to Index](#)

FEASA LED ANALYSER ICT VERSION



Specifications

Part Number(s)

Feasa 20-I	LED Analyser with 20 Fibers
Feasa 10-I	LED Analyser with 10 Fibers
Feasa 6-I	LED analyser with 6 Fibers
Feasa 5-I	LED Analyser with 5 Fibers
Feasa 3-I	LED Analyser with 3 Fibers

Feasa 20-IB	LED Analyser with 20 Fibers
Feasa 10-IB	LED Analyser with 10 Fibers
Feasa 6-IB	LED analyser with 6 Fibers
Feasa 5-IB	LED Analyser with 5 Fibers
Feasa 3-IB	LED Analyser with 3 Fibers

Physical -I, -IB

Dimensions 140mm x 29mm x 50mm (L x W x H) – 20-I, 20-IB
Dimensions 100mm x 29mm x 50mm (L x W x H) – 3-I - 10-I, 3-IB - 10-IB
Fiber Length 0.6m
Fiber Diameter 1.0mm
Fiber Core Diameter 0.5mm
Fiber Bend Radius 15mm
Plastic Fiber Temperature Range -40°C to +70°C
Number of Fibers 3, 5, 6, 10, 20
Operating Temperature Range 0°C to +50°C

Electrical

Supply Voltage 5.0V +/- 0.5V
Supply Current 200 mA max
ICT Interface, Serial RS232 Interface
Output Data Format RGB, HSI, xy, uv, wavelength, cct
Hue (Frequency): 1-100 Khz
Wavelength (Frequency): 1-100KHz
Saturation (Frequency): 1-100 Khz
x Chromaticity (Frequency): 1-100KHz
y Chromaticity (Frequency): 1-100KHz
Intensity (Frequency): 1-100 KHz

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=± 0.01, Y = ± 0.01 @ x=0.33, y=0.33 using OH-3 Optical Head
Red (630nm) ± 4nm
Green (520nm) ± 2nm
Blue (450nm) ± 5nm
Yellow (590nm) ± 2nm

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

[Back to Index](#)



ECO:



Engineering Change Notice: Date of Release August 10th 2020

Feasa Led Analyser Models: **20i, 10i, 6i, 5i, 3i**

In the Feasa ICT Led Analyser released prior to July 2020, the ICT Frequency Mode and Sync Serial Mode were both active. However when using the ICT Led analyser in Frequency Out mode the SyncSerial Mode must be disabled. There are two hardware steps and one software step to do this on these older models.

Hardware Steps:

1. In the ICT Fixture please wire ICT_pin 2 (Serial_IN) through a 4.7Kohm resistor to ICT_pin 19 (VCC).
2. In the ICT Fixture please wire ICT_pin 3 (Serial_CLK) through a 4.7Kohm resistor to ICT_pin 19 (VCC).

Software Step:

1. On the relevant ICT models please remove the SyncSerial pins 1,2,3 from you digital model.

There are 3 models affected

20i_la_cap_n_read_f_hsi

20i_la_cap_n_read_f_wsi

20i_la_cap_n_read_f_xyi

A full suite of new models is available to download.

On Led Analysers released after this date no changes are required.

*Engineering change Notice Issued August 10th 2020 by Feasa Enterprises
Ver:1.0*

[Back to Index](#)



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](#)