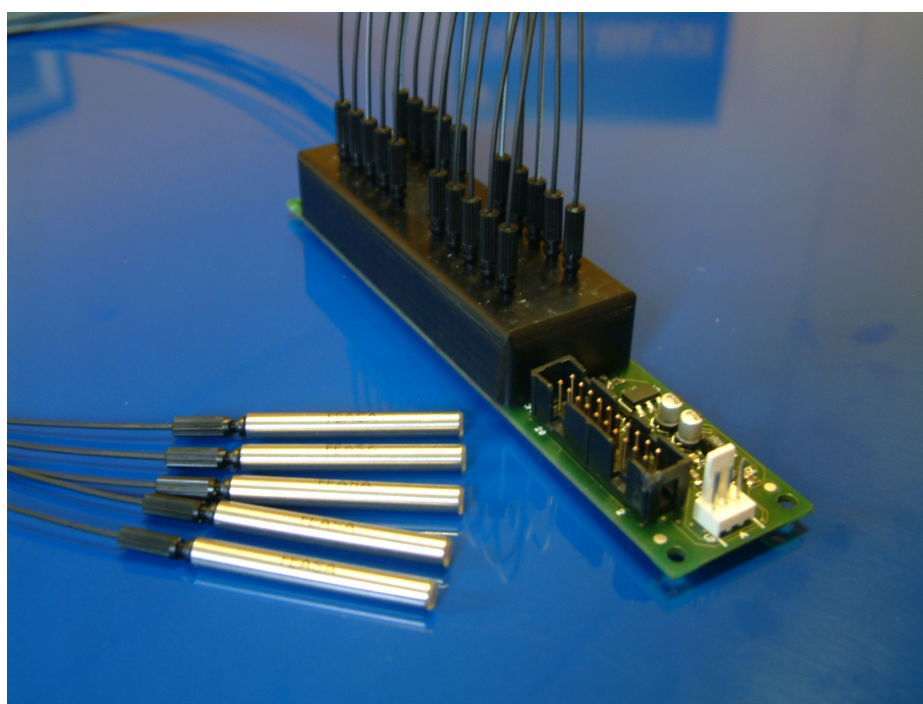


FEASA LED ANALYSER

Agilent 3070 Application Note



User Manual for Feasa-I Analysers installed on an Agilent 3070



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

www.feasa.ie

Email: sales@feasa.ie

Rev. 5.1

Date: October, 2020

FEASA LED ANALYSER

Agilent 3070 Application Note



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, x-IB (where x is the no. of fibers either 2, 3, 5, 6, 10, 20)

The Feasa 20-I is a 20 channel unit which will test up to 20 LEDs while the Feasa 2-I is a 2 channel unit testing up to 2 LED's.

The interface on these units is Serial or ICT Interface (Measuring Frequency).

© Copyright 2005-2020 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

Agilent 3070 Application Note



Table of Contents

Introduction.....	4
Preparation:.....	5
Decide NOW which test Mode you are going to use !.....	6
Create a board file:.....	7
LA_Frequency_Mode \ board.....	7
Create a board_xy file:.....	9
LA_Sync_Serial_Mode \ board.....	10
LA_Both_Modes \ board.....	12
Generate a Custom Lib.....	12
(FREQUENCY ONLY) custom_lib_20i.....	12
(SYNC SERIAL ONLY) custom_lib_20i.....	12
(Both_Modes) custom_lib_20i.....	12
Run IPG.....	12
Edit the testplan.....	13
LA_Frequency_Mode \ testplan.....	13
LA_Mode_Sync_Serial_Mode \ testplan.....	15
LA_Both_Modes \ testplan.....	31

FEASA LED ANALYSER

Agilent 3070 Application Note



Introduction

The purpose of this document is to aid the customer in the installation of the **Feasa LED Analyser** on an Agilent 3070 In Circuit Test platform. The Led Analyser is an instrument that tests the Colour and Intensity of Light Emitting Diodes (LEDs) in a test process. The Analyser can have up to 20 flexible Fiber-Optic Light Guides which are mounted individually over the LEDs to be tested. The models available are Feasa 2-I, Feasa 3-I, Feasa 5-I, Feasa 6-I, Feasa-10-I and Feasa 20-I, Feasa 2-IB, Feasa 3-IB, Feasa 5-IB, Feasa 6-IB, Feasa 10-IB, Feasa 20-IB.

Emitted Light from the LEDs is guided through these Fiber-Optic Light Guides to the Analyser where the data of the LED's are stored. The data can then be read out of the Analyser through Serial (3pin) or ICT Interfaces (20 pin) in either HSI or RGBI format.

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 in three formats,
HSI (Colour, Saturation and Intensity),
WSI (Wavelength, Saturation and Intensity) or
xyI (x-chromaticity, y-chromaticity and Intensity).

The second mode uses a high-speed Synchronous Serial Bus to output the LED Data in different formats.

RGBI (Red, Green, Blue & Intensity),
HSI (Colour, Saturation & Intensity),
xyI (xy & Intensity),
WSI (Wavelength, Saturation, Intensity)
WI (Dominant Wavelength),
cct (Colour Temperature),
ABSINT (Absolute Intensity)

Please refer to the Feasa ICT User Manual Vx.x for a more detailed explanation on the operation of the Led Analyser.

[*Back to Index*](#)

FEASA LED ANALYSER

Agilent 3070 Application Note



Preparation:

There is a choice of two operating modes using the Led analyser 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 results.

The **Frequency mode** is by far the simplest to implement and debug, however it will require Agilent resources for 16 pins.

The **Synchronous Serial Mode** is easy to implement but a little more complicated to debug and only requires Agilent resources for 5 pins.

Both modes can be implemented up front and decide during debug which to use and will only require Agilent resources for 20 pins.

Feasa have included full examples of all the required files on the CD.

[*Back to Index*](#)

FEASA LED ANALYSER

Agilent 3070 Application Note



Decide NOW which test Mode you are going to use !

Frequency Mode:

or

Synchronous Serial Mode:

or

Frequency Mode & Synchronous Serial Mode

Your decision will dictate which of the following Methods you will select.

[Back to Index](#)

FEASA LED ANALYSER

Agilent 3070 Application Note



Create a board file:

Follow the Agilent guidelines for generating a PANEL board file. We have included here the board file for a Feasa 20-I which can be cut and paste into your board file. Full version available on the CD at :

LA_Frequency_Mode \ board

LIBRARY OPTIONS

"custom_lib_20i";

UNCONNECTED PIN NC;

BOARDS

- 1 Led_Analyser "Feasa LA fail";*
- 2 Led_Demo "Led Demo fail";*

BOARD Led_Analyser

FIXED NODE OPTIONS

GND Family ALL is 0;
VCC Family ALL is 1;
GND GROUND;
VCC Supply 3 At 5 Volts, 1 Amps;

PIN LIBRARY

Feasa_20i PN"20i";

CONNECTIONS

Addr_0
Feasa_20i.9;
Addr_1
Feasa_20i.10;
Addr_2
Feasa_20i.11;
Addr_3
Feasa_20i.12;
Addr_4
Feasa_20i.13;
Color
Feasa_20i.16;
Ext_trig
Feasa_20i.7;
GND
Feasa_20i.4
Feasa_20i.20;
Intensity
Feasa_20i.15;
OE_bar
Feasa_20i.6;
PWM
Feasa_20i.8;
RESET
Feasa_20i.5;
Ry_By_bar
Feasa_20i.18;
Saturation
Feasa_20i.17;

FEASA LED ANALYSER

Agilent 3070 Application Note



```
Serial_clk
Feasa_20i.3;
Serial_in
Feasa_20i.2;
Serial_out
Feasa_20i.1;
VCC
Feasa_20i.19;
WE_bar
Feasa_20i.14;
```

DEVICES

```
Feasa_20i
1.Serial_out
2.Serial_in
3.Serial_clk
4.GND
5.RESET
6.OE_bar
7.Ext_trig
8.PWM
9.Addr_0
10.Addr_1
11.Addr_2
12.Addr_3
13.Addr_4
14.WE_bar
15.Intensity
16.Color
17.Saturation
18.Ry_By_bar
19.VCC
20.GND;
```

END BOARD

BOARD Led_Demo

FIXED NODE OPTIONS

```
ERDE Family ALL is 0;
VCC Family ALL is 1;
ERDE GROUND;
PA Supply 1 At 5 Volts, 1 Amps;
```

CAPACITOR

```
C1 1u 30 30 f PN"123";
C2 100n 30 30 f PN"940-6387";
```

CONNECTOR

```
J1 NT "MOLEX_2PIN";
J2 NT "USB_MINI_B";
```

PIN LIBRARY

```
D1 PN"Red_LED";
D2 PN"Green_LED";
D3 PN"Blue_LED";
D4 PN"White_LED";
D5 PN"Yellow_LED";
```

END BOARD

[Back to Index](#)

FEASA LED ANALYSER

Agilent 3070 Application Note



Create a board_xy File

Create a board_xy file:

Follow the Agilent guidelines for generating a **PANEL** board file. We have included here the board file for a Feasa 20-I which can be cut and paste into your board file. Full version available on the CD at **ICT/Agilent 3070/ Fixture Files/board_xy**

```
UNITS MILS;  
SCALE 0.1;  
PLACEMENT 38838, -12067 0.0;
```

```
OUTLINE  
-5000, -5000  
85000, -5000  
85000, 70000  
50000, 70000  
50000, 77000  
32000, 77000  
32000, 70000  
-5000, 70000;
```

```
BOARDS  
1 Led_Demo 10000 50000 90.00;  
2 Led_Analyser 43000 55000 0.00;
```

BOARD Led_Analyser

```
OUTLINE  
-9000, 20000  
5000, 20000  
5000, 16000  
-9000, 16000  
-9000, 20000;
```

```
OTHER  
ALTERNATES  
-6500, 17000 Feasa_20i.20 PREFERRED;  
-6500, 19000 Feasa_20i.19 PREFERRED;  
-5500, 17000 Feasa_20i.18 PREFERRED;  
-5500, 19000 Feasa_20i.17 PREFERRED;  
-4500, 17000 Feasa_20i.16 PREFERRED;  
-4500, 19000 Feasa_20i.15 PREFERRED;  
-3500, 17000 Feasa_20i.14 PREFERRED;  
500, 17000 Feasa_20i.6 PREFERRED;  
500, 19000 Feasa_20i.5 PREFERRED;  
1500, 17000 Feasa_20i.4 PREFERRED;  
1500, 19000 Feasa_20i.3 PREFERRED;  
2500, 17000 Feasa_20i.2 PREFERRED;  
2500, 19000 Feasa_20i.1 PREFERRED;
```

END BOARD

[Back to Index](#)

FEASA LED ANALYSER

Agilent 3070 Application Note



LA_Sync_Serial_Mode \ board

BOARD Led_Analyser

FIXED NODE OPTIONS

*GND Family ALL is 0;
VCC Family ALL is 1;
GND GROUND;
VCC Supply 3 At 5 Volts, 1 Amps;*

PIN LIBRARY

Feasa_20i PN"20i";

CONNECTIONS

*Addr_0
Feasa_20i.9;
Addr_1
Feasa_20i.10;
Addr_2
Feasa_20i.11;
Addr_3
Feasa_20i.12;
Addr_4
Feasa_20i.13;
Color
Feasa_20i.16;
Ext_trig
Feasa_20i.7;
GND
Feasa_20i.4
Feasa_20i.20;
Intensity
Feasa_20i.15;
OE_bar
Feasa_20i.6;
PWM
Feasa_20i.8;
RESET
Feasa_20i.5;
Ry_By_bar
Feasa_20i.18;
Saturation
Feasa_20i.17;
Serial_clk
Feasa_20i.3;
Serial_in
Feasa_20i.2;
Serial_out
Feasa_20i.1;
VCC
Feasa_20i.19;
WE_bar
Feasa_20i.14;*

DEVICES

*Feasa_20i
1.Serial_out
2.Serial_in
3.Serial_clk
4.GND
5.RESET*

FEASA LED ANALYSER

Agilent 3070 Application Note



6.OE_bar
7.Ext_trig
8.PWM
9.Addr_0
10.Addr_1
11.Addr_2
12.Addr_3
13.Addr_4
14.WE_bar
15.Intensity
16.Color
17.Saturation
18.Ry_By_bar
19.VCC
20.GND;

END BOARD

[Back to Index](#)

FEASA LED ANALYSER

Agilent 3070 Application Note



LA_Both_Modes \ board

This board file is a combination of the two methods described above. Please refer to the documentation on the CD provided for an example.

Generate a Custom_Lib

Feasa have created a suite of pin Libraries for you to use. Please select the option that meets your requirements from the CD at :

(FREQUENCY ONLY) custom_lib_20i

20i_la_cap_n_read_f_hsi
20i_la_cap_n_read_f_wsi
20i_la_cap_n_read_f_xyi

(SYNC SERIAL ONLY) custom_lib_20i

20i
20i_la_capture_ss
20i_la_capture_pwm_ss

20i_la_read_absint_ss
20i_la_read_cct_ss
20i_la_read_hsi_ss
20i_la_read_rgb_ss
20i_la_read_wave_ss
20i_la_read_wsi_ss
20i_la_read_xyi_ss
20i_la_set_setfactor_ss

la_seq_capt_setup_ss
la_seq_capture_ss
la_seq_rd_fib_ca_num_ss
la_seq_rd_fib_times_ss
la_seq_set_threshold_ss

(Both_Modes) custom_lib_20i

All of the above

Run IPG

Run IPG to generate all your necessary files.

[Back to Index](#)

FEASA LED ANALYSER

Agilent 3070 Application Note



Edit the testplan

Feasa have included all the necessary changes to the testplan on the examples provided on the CD. Please read all the comments sections added as they will attempt to explain the process.

LA_Frequency_Mode \ testplan

```
sub Analog_Functional_Tests (Status_Code, Message$)
global All_Failed
global BoardSet_boards_2_to_2(*)
global BoardSet_boards_1_to_1(*)
global BoardSet(*)
dim Fibernum(0:7), Data(0:8), Sen(0:7) !Define arrays needed for Led Testing
```

```
if Message$ <> "" then print tab(5);Message$
call Panel_Enable_Boards
```

Frequency_mode:

```
!*****
! NOTE: Edit all the Led Analyser Tests *
! to remove the assign Floating Inputs *
! in the disable section and recompile *
! - see Example in the documentation CD *
!*****

!*! -----
!*! The are 3 possible Frequency Modes available to test with.
!*! Hue, Saturation, Intensity (HSI)
!*! Wavelength, Saturation, Intensity (WSI)
!*! xChromaticity, yChromaticity, Intensity (xyI)
!*! To use the Led Analyser in either of these FREQUENCY MODEs
!*! Connect the Led analyser to a PC using the Serial port - cable supplied
!*! Launch the Terminal program provided on the CD
!*! type getictstatus
!*! It will respond with Frequency out mode : HSI (factory preset)
!*! Select the mode you require by typing
!*! setfrequencyxxx where xxx is the required mode
!*! Disconnect from the PC and proceed to Fixture/Program debug
!*! -----
```

```
!***** Select the SENSITIVITY / SETFACTOR of the Analyser
!*!
!*! The sensitivity level or Exposure level of the Analyser has a range 1-15
!*! The default setting is 1. Setting the sensitivity to 2 doubles the
!*! measurement test time and increases the intensity values etc..
```

FEASA LED ANALYSER

Agilent 3070 Application Note



!! The need to change from the default is for testing dim Leds
!!
!! RUN THIS TEST in DEBUG mode ONLY as running constantly during production
!! will eventually corrupt the Analyser after 100,000 writes approx.

! test "4%feasa_20i%la_setfactor_f"

!***** **Capture Command** *****

!

! There are many different capture modes which are possible in Frequency Mode
! C(Auto Range), C1-C5(Manual Ranges Low, Medium, High, Super, Ultra) or PWM.
! Select the range & type you want from the following:
! la_cap_n read_f_hsi, la_cap_n read_f_wsi or la_cap_n read_f_xyi models

! To adjust your HSI (Hue, Saturation & Intensity) readings please edit the
! analog portion of the la_read_hsi_f mixed test

! To adjust your WSI (Wavelength, Saturation & Intensity) readings please
! edit the analog portion of the la_read_wsi_f mixed test

! To adjust your xyI (xyChromaticity & Intensity) readings please edit the
! analog portion of the la_read_xyi_f mixed test

!***** **SELECT THE FREQUENCY TEST MODE you require** *****

```
print tab(5);"      Capture & Read"  
print tab(5);"      Frequency Mode HSI"  
test "4%feasa_20i%la_cap_n_read_f_hsi"
```

```
! print tab(5);"      Capture & Read"  
! print tab(5);"      Frequency Mode WSI"  
! test "4%feasa_20i%la_cap_n_read_f_wsi"
```

```
! print tab(5);"      Capture & Read"  
! print tab(5);"      Frequency Mode XYI"  
! test "4%feasa_20i%la_cap_n_read_f_xyi"
```

goto End_LA

!! *****

[Back to Index](#)

FEASA LED ANALYSER

Agilent 3070 Application Note



LA_Mode_Sync_Serial_Mode \ testplan

```
sub Functional_Tests (Status_Code, Message$)
global All_Failed
global BoardSet_boards_2_to_2(*)
global BoardSet_boards_1_to_1(*)
```

```
    if Message$ <> "" then print tab(5);Message$
    call Panel_Enable_Boards
```

```
!*****
! NOTE: Edit all the Led Analyser Tests *
!       to remove the assign Floating Inputs *
!       in the disable section and recompile *
!       - see Example in the documentation CD *
!*****
```

SyncSerial_mode:

```
!*/ *****
```

```
print tab(5);"    Synchronous Serial Mode"
```

```
!*** Select the SENSITIVITY / SETFACTOR of the Analyser in ss mode
!*/
!*/ The sensitivity level or Exposure level of the Analyser has a range 1-15
!*/ The default setting is 1. Setting the sensitivity to 2 doubles the
!*/ measurement test time and increases the intensity values etc..
!*/ The need to change from the default is for testing very dim Leds
```

```
!*/ RUN THIS TEST in DEBUG mode ONLY as running constantly during production
!*/ will eventually corrupt the Analyser after 100,000 writes approx.
```

```
!*/ To verify or change the sensitivity level of the Led Analyser
!*/ Uncomment the 9 lines below.
```

```
! Sensitivity = 1      ! Test Sensitivity Default = 1 max = 15
```

```
! for Data_bit = 0 to 7 ! Convert Decimal number to binary number
! Sen(Data_bit) = bit(Sensitivity,(7-Data_bit)) ! extract each binary bit
! next Data_bit
```

```
! test "4%feasa_20i%la_setfactor_ss" ! Set Sensitivity OK
```

```
! print "Sensitivity Level:"
```

```
! print Data(0) ! This is the sensitivity read back from the test
```

FEASA LED ANALYSER

Agilent 3070 Application Note



!***** **CAPTURE COMMAND** *****

!*

!! There are many different capture modes which are possible through the Sync

!! Serial port, Auto Range, Manual Range #1 to #5 and PWM Capture

!! Each of these capture modes is very similar except that the command send to

!! the Analyser is changed as listed after each test

!! *** Capture Commands ***

!! (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)

!! (0X90) Get Intensity for Fiber ##

!! (0X91) Sequence Capture Fiber No + Capture No ## + ###

!! (0XC0) Capture Time (2 Bytes) + Wait Time (2 Bytes) + Sequence Number (2 Bytes)

!! (0X45) Capture Sequence #

Command\$ = "55" ! Command to send to analyser

CommandInt = hti(Command\$) ! Convert to Integer

CommandBin\$ = itb\$(CommandInt) ! Convert to Binary

CommandBin\$ = "00000000" & CommandBin\$! Add leading zeros

CommandBin\$ = CommandBin\$(len(CommandBin\$)-7;8) ! Extract the last 8 bits

! print CommandBin\$

for Data_bit = 1 to 8 ! Convet the fiber number to binary

Data(Data_bit) = val(CommandBin\$(Data_bit;1))

next Data_bit

! print Data(*)

!! Choose either Standard or PWM capture mode here

!!

print tab(5);" Synchronous Serial Mode Capture"

test "4%feasa_20i%la_capture_ss"; Data(*)

! or

! test "4%feasa_20i%la_capture_pwm_ss"; Data(*)

!***** **RESULTS FORMAT** *****

FEASA LED ANALYSER

Agilent 3070 Application Note



!! Select the results format you want by uncommenting the goto statement

```
goto SyncSerial_HSI_results
! goto SyncSerial_RGB_results
! goto SyncSerial_WSI_results
! goto SyncSerial_Wave_results
! goto SyncSerial_xyI_results
! goto SyncSerial_cct_results
! goto SyncSerial_absint_results
```

```
!*****
!! -----
!! To use the Led Analyser in SYNCHRONOUS MODE
!! Connect the Led analyser to a PC using the Serial port - cable supplied
!! Launch the Terminal program provided on the CD
!! type  getictstatus
!! type  setsyncserial1
!! type  getictstatus
!! it should reply Sync serial mode enabled
!! Disconnect from the PC and proceed to Fixture/Program debug
!! -----
```

FEASA LED ANALYSER

Agilent 3070 Application Note



!***** READ BACK HSI RESULTS *****

SyncSerial_HSI_results:

print tab(5);" Synchronous Serial Mode HSI"

! *! The Feasa 20-I is shipped with 20 fibers, the Feasa 10-I with 10 fibers etc.

! *! Select here the number of Fibers you are using.

for I = 1 to 20 ! 20 Fibers

for Data_bit = 0 to 7 !Convet the fiber number to binary

Fibernum(Data_bit) = bit(I,(7-Data_bit))

next Data_bit

test "4%feasa_20i%la_read_hsi_ss" ; Fibernum(*), Data(*) ! Read H,S,I

! pause

Hue = ((Data(0) * 256) + Data(1))/100 ! Hue

Saturation = Data(2) ! Saturation

Intensity = (Data(3) * 65536) + (Data(4) * 256) + Data(5) ! Intensity

! *! Uncomment the following 2 lines for debug purposes only

! *!

! print Data(0);Data(1);Data(2);Data(3);Data(4);Data(5) ! print Data

! print I; Hue; Saturation; Intensity ! print HSI

!*****

! *!----- Fiber 1

if Hue < 0.01 or Hue > 3 or Saturation < 90 or Saturation > 101 or Intensity < 2k or Intensity > 10k then

if I = 1 then

fail device

report "----- "

report " Failed HSI for D1 "

report " Hue High 003 Low .01 "; Hue

report " Sat High 101 Low 090 "; Saturation

report " Int High 10k Low 2k "; Intensity

report "----- "

end if

end if

next I

goto End_LA

FEASA LED ANALYSER

Agilent 3070 Application Note



! ***** **READ BACK RGB RESULTS** *****

SyncSerial_RGB_results:

```
print tab(5);"      Synchronous Serial Mode RGBI"
```

! *! The Feasa 20-I is shipped with 20 fibers, the Feasa 10-I with 10 fibers etc.

! *! Select here the number of Fibers you are using.

for I = 1 to 20 ! 20 Fibers

```
for Data_bit = 0 to 7 !Convrt the fiber number to binary
```

```
  Fibernum(Data_bit) = bit(I,(7-Data_bit))
```

```
next Data_bit
```

```
test "4%feasa_20i%la_read_rgb_ss" ; Fibernum(*), Data(*) ! Read R, G, B
```

```
Red = Data(0) ! Red
```

```
Green = Data(1) ! Green
```

```
Blue = Data(2) ! Blue
```

```
Intensity = (Data(3) * 65536) + (Data(4) * 256) + Data(5) ! Intensity
```

! *! Uncomment the following 2 lines for debug purposes

! *!

```
! print Data(0);Data(1);Data(2);Data(3);Data(4);Data(5) ! print Data
```

```
! print I; Red; Green; Blue; Intensity ! print results
```

! *****

! *!----- Fiber 1

if Red < 253 or Red > 255 or Green < 0 or Green > 3 or Blue < 0 or Blue > 3 or Intensity < 2k or Intensity > 10k then

if I = 1 then

fail device

```
report "----- "
```

```
report " Failed RGBI for D1 "
```

```
report " Red High 255 Low 253 "; Red
```

```
report " Green High 003 Low 000 "; Green
```

```
report " Blue High 003 Low 000 "; Blue
```

```
report " Int High 10k Low 2k "; Intensity
```

```
report "-----"
```

end if

end if

next I

goto End_LA

FEASA LED ANALYSER

Agilent 3070 Application Note



!***** READ BACK WSI RESULTS *****

SyncSerial_WSI_results:

```
print tab(5);"      Synchronous Serial Mode WSI"
```

! *! The Feasa 20-I is shipped with 20 fibers, the Feasa 10-I with 10 fibers etc.

! *! Select here the number of Fibers you are using.

for I = 1 to 20 ! 20 Fibers

```
for Data_bit = 0 to 7 !Convet the fiber number to binary
```

```
Fibernum(Data_bit) = bit(I,(7-Data_bit))
```

```
next Data_bit
```

```
test "4%feasa_20i%la_read_wsi_ss" ; Fibernum(*), Data(*) ! Read WSI
```

```
Wavelength = ((Data(0) * 256) + Data(1)) ! Wavelength data
```

```
Wavelength = Wavelength ! Wavelength data
```

```
Saturation = Data(2) ! Saturation
```

```
Intensity = (Data(3) * 65536) + (Data(4) * 256) + Data(5) ! Intensity
```

! *! Uncomment the following 2 lines for debug purposes

!

```
! print Data(0);Data(1);Data(2);Data(3);Data(4);Data(5) ! print Data
```

```
! print I; Wavelength; Saturation; Intensity ! print wavelength
```

!*****

! *!----- Fiber 1

if Wavelength < 625 or Wavelength > 635 or Saturation < 90 or Saturation > 101 or
Intensity < 2k or Intensity > 10k then

if I = 1 then

fail device

```
report "----- "
```

```
report " Failed WSI for D1 Red "
```

```
report " Wav High 635 Low 625 "; Wavelength
```

```
report " Sat High 101 Low 090 "; Saturation
```

```
report " Int High 10k Low 2k "; Intensity
```

```
report "-----"
```

end if

end if

next I

goto End_LA

FEASA LED ANALYSER

Agilent 3070 Application Note



!***** READ BACK WAVELENGTH RESULTS *****

SyncSerial_Wave_results:

```
print tab(5);"      Synchronous Serial Mode Dominant Wavelength"
```

!*! The Feasa 20-I is shipped with 20 fibers, the Feasa 10-I with 10 fibers etc.

!*! Select here the number of Fibers you are using.

for I = 1 to 20 ! 20 Fibers

```
for Data_bit = 0 to 7 !Convet the fiber number to binary
```

```
Fibernum(Data_bit) = bit(I,(7-Data_bit))
```

```
next Data_bit
```

```
test "4%feasa_20i%la_read_wave_ss"; Fibernum(*), Data(*) ! Read Wavelength
```

```
Wavelength = ((Data(0) * 256) + Data(1))
```

```
! Wavelength data
```

```
Wavelength = Wavelength
```

```
! Wavelength data
```

!*! Uncomment the following 2 lines for debug purposes

!

```
! print Data(0);Data(1)
```

```
! print Data
```

```
! print I; Wavelength
```

```
! print wavelength
```

!

!*****

!*!----- Fiber 1

```
if Wavelength < 630 or Wavelength > 640 then
```

```
if I = 1 then
```

```
fail device
```

```
report "----- "
```

```
report " Failed WSI for D1 Red "
```

```
report " Wav High 640 Low 630 "; Wavelength
```

```
report "-----"
```

```
end if
```

```
end if
```

```
next I
```

```
goto End_LA
```

FEASA LED ANALYSER

Agilent 3070 Application Note



!***** READ BACK XYI RESULTS *****

SyncSerial_xyI_results:

```
print tab(5);"      Synchronous Serial Mode xyI"
```

!! The Feasa 20-I is shipped with 20 fibers, the Feasa 10-I with 10 fibers etc.

!! Select here the number of Fibers you are using.

for I = 1 to 20 ! 20 Fibers

```
for Data_bit = 0 to 7 !Convet the fiber number to binary
```

```
Fibernum(Data_bit) = bit(I,(7-Data_bit))
```

```
next Data_bit
```

```
test "4%feasa_20i%la_read_xyi_ss" ; Fibernum(*), Data(*) ! Read xy,I
```

```
X = ((Data(0) * 256) + Data(1))/10000 ! x- data
```

```
Y = ((Data(2) * 256) + Data(3))/10000 ! y- data
```

```
Intensity = (Data(4) * 65536) + (Data(5) * 256) + Data(6) ! Intensity
```

!! Uncomment the following 2 lines for debug purposes

!!

```
! print Data(0);Data(1);Data(2);Data(3);Data(4);Data(5);Data(6) ! print Data
```

```
! print I; X; Y; Intensity ! print xyI
```

!

!*****

!!----- Fiber 1

```
if X < .30 or X > .360 or Y < .30 or Y > .360 or Intensity < 1k or Intensity > 10k then
```

```
if I = 1 then
```

```
fail device
```

```
report "----- "
```

```
report " Failed xyI for D1 White "
```

```
report " X High .360 Low .30 "; X
```

```
report " Y High .360 Low .30 "; Y
```

```
report " Int High 10k Low 1k "; Intensity
```

```
report "-----"
```

```
end if
```

```
end if
```

```
next I
```

```
goto End_LA
```

FEASA LED ANALYSER

Agilent 3070 Application Note



!***** **READ BACK CCT RESULTS** *****

SyncSerial_cct_results:

```
print tab(5);"      Synchronous Serial Mode cct"
```

! *! The Feasa 20-I is shipped with 20 fibers, the Feasa 10-I with 10 fibers etc.

! *! Select here the number of Fibers you are using.

for I = 1 to 20 ! 20 Fibers

```
for Data_bit = 0 to 7 !Convet the fiber number to binary
  Fibernum(Data_bit) = bit(I,(7-Data_bit))
next Data_bit
```

test "4%feasa_20i%la_read_cct_ss" ; Fibernum(*), Data(*) ! Read CCT

```
CCT = ((Data(0) * 256) + Data(1)) ! CCT data
```

! *! Uncomment the following 2 lines for debug purposes

```
!
! print Data(0);Data(1) ! print Data
! print I; CCT ! print CCT
!
```

!*****

! *!----- Fiber 1

if CCT < 3k or CCT > 8k then

if I = 1 then

fail device

report "----- "

report " Failed cct for D1 "

report " Red High 8k Low 3k "; CCT

report "----- "

end if

end if

next I

goto End_LA

FEASA LED ANALYSER

Agilent 3070 Application Note



!***** **READ ABSINT RESULTS** *****

SyncSerial_absint_results:

!! The Feasa 20-I is shipped with 20 fibers, the Feasa 10-I with 10 fibers etc.

!! Select here the number of Fibers you are using.

for I = 1 to 20 ! 20 Fibers

for Data_bit = 0 to 7 !Convet the fiber number to binary

Fibernum(Data_bit) = bit(I,(7-Data_bit))

next Data_bit

test "4%feasa_20i%la_read_absint_ss" ; Fibernum(*), Data(*) ! Read absint

!! Uncomment the following lines for debug purposes

! print Data(0);Data(1);Data(2);Data(3) ! print Data

Intensity = (Data(0) * 65536) + (Data(1) * 256) + Data(2) ! Intensity Data

Intensity = Intensity * .0001

Exponent\$ = ith\$(Data(3))

Val = hti(Exponent\$, Err)

if Err<> 0 then print "Data Error"

if Val > 127 then

! If Val > 127 Exponent is negative

Exp = Val -256

! If Val < 128 Exp is Positive 2's Comp

Intensity = Intensity * 10

end if

if Val < 128 then

! If Val > 127 Exponent is negative

Intensity = Intensity * 10

end if

Intensity = Intensity * 10^(Exp)

! print I, Intensity

next I

goto End_LA

FEASA LED ANALYSER

Agilent 3070 Application Note



SequenceCapture_mode:

```
!*****
! Capture Sequence setup command begins with 0xC0
! 0xC0 + capture time (2Bytes) + Wait time (2Bytes) + Sequence number (2Bytes)
! capture time and wait time are specified in msec 001 - 999
! Sequence Number is the number of captures to store.
! Maximum Sequence number is 400 per fiber channel.8000 if only 1 fiber selected
!-----
! Execute Capture Sequence using command 0x45
! Send 0x45 to the Analyser to initiate the capture sequence
!-----
! Send 0x91 + Fiber number + sequence capture number to read back 3
! bytes of Intensity
!-----
! Send 0xC1 to set a threshold value to convert Intensities to binary 1 or 0.
! Intensities > the threshold value will be converted to 1
! Intensities < the threshold value will be converted to 0
! This command must be sent before the 0x92 command.
!-----
! Send 0x92 + fiber number to read 6 bytes of duty cycle times
! for Progressive Indicator timing.
! 6 bytes are returned.
! Bytes 0,1 returns the initial OFF time in msec.
! Bytes 2,3 returns the ON time of the LED in msec.
! Bytes 4,5 returns the OFF time after the LED is switched off.
!-----

Command$ = "C0"                                ! Command to send to analyser

CommandInt = hti(Command$)                      ! Convert to Integer
CommandBin$ = itb$(CommandInt)                  ! Convert to Binary
CommandBin$ = "00000000" & CommandBin$          ! Add leading zeros
CommandBin$ = CommandBin$(len(CommandBin$)-7;8) ! Extract the last 8 bits
! print CommandBin$

for Data_bit = 1 to 8    ! Convet the fiber number to binary
    Data1(Data_bit) = val(CommandBin$(Data_bit;1))
next Data_bit

!print Data1(*)
```

FEASA LED ANALYSER

Agilent 3070 Application Note



!*****

Capture_Time\$ = "00" ! Capture Time in mSec - 5mSec

```
CommandInt = hti(Capture_Time$)           ! Convert to Integer
CommandBin$ = itb$(CommandInt)            ! Convert to Binary
CommandBin$ = "00000000" & CommandBin$    ! Add leading zeros
CommandBin$ = CommandBin$[len(CommandBin$)-7;8] ! Extract the last 8 bits
! print CommandBin$
```

```
for Data_bit = 1 to 8      ! Convet the fiber number to binary
  Data2(Data_bit) = val(CommandBin$[Data_bit;1])
next Data_bit
```

!print Data2()*

Capture_Time\$ = "05" ! Capture Time in mSec - 5mSec

```
CommandInt = hti(Capture_Time$)           ! Convert to Integer
CommandBin$ = itb$(CommandInt)            ! Convert to Binary
CommandBin$ = "00000000" & CommandBin$    ! Add leading zeros
CommandBin$ = CommandBin$[len(CommandBin$)-7;8] ! Extract the last 8 bits
! print CommandBin$
```

```
for Data_bit = 1 to 8      ! Convet the fiber number to binary
  Data3(Data_bit) = val(CommandBin$[Data_bit;1])
next Data_bit
```

!print Data3()*

FEASA LED ANALYSER

Agilent 3070 Application Note



!*****

Wait_Time\$ = "00" ! Wait Time in mSec - 5mSec

```
CommandInt = hti(Wait_Time$)           ! Convert to Integer
CommandBin$ = itb$(CommandInt)         ! Convert to Binary
CommandBin$ = "00000000" & CommandBin$ ! Add leading zeros
CommandBin$ = CommandBin$[len(CommandBin$)-7;8] ! Extract the last 8 bits
! print CommandBin$
```

```
for Data_bit = 1 to 8    ! Convet the fiber number to binary
  Data4(Data_bit) = val(CommandBin$[Data_bit;1])
next Data_bit
```

!print Data4()*

Wait_Time\$ = "05" ! Wait Time in mSec - 5mSec

```
CommandInt = hti(Wait_Time$)           ! Convert to Integer
CommandBin$ = itb$(CommandInt)         ! Convert to Binary
CommandBin$ = "00000000" & CommandBin$ ! Add leading zeros
CommandBin$ = CommandBin$[len(CommandBin$)-7;8] ! Extract the last 8 bits
! print CommandBin$
```

```
for Data_bit = 1 to 8    ! Convet the fiber number to binary
  Data5(Data_bit) = val(CommandBin$[Data_bit;1])
next Data_bit
```

!print Data5()*

FEASA LED ANALYSER

Agilent 3070 Application Note



```
!*****
!! Sequence Number is 2 Bytes : High Byte/256 + Low Byte
!*****
```

Sequence_High\$ = "00" ! High Byte of Sequence Number in HEX

```
CommandInt = hti(Sequence_High$)      ! Convert to Integer
CommandBin$ = itb$(CommandInt)         ! Convert to Binary
CommandBin$ = "00000000" & CommandBin$ ! Add leading zeros
CommandBin$ = CommandBin$(len(CommandBin$)-7;8) ! Extract the last 8 bits
! print CommandBin$
```

```
for Data_bit = 1 to 8    ! Convet the fiber number to binary
  Data6(Data_bit) = val(CommandBin$(Data_bit;1))
next Data_bit
```

!print Data6()*

Sequence_Low\$ = "40" ! Low Byte of Sequence Number in HEX

```
CommandInt = hti(Sequence_Low$)      ! Convert to Integer
CommandBin$ = itb$(CommandInt)         ! Convert to Binary
CommandBin$ = "00000000" & CommandBin$ ! Add leading zeros
CommandBin$ = CommandBin$(len(CommandBin$)-7;8) ! Extract the last 8 bits
! print CommandBin$
```

```
for Data_bit = 1 to 8    ! Convet the fiber number to binary
  Data7(Data_bit) = val(CommandBin$(Data_bit;1))
next Data_bit
```

!print Data7()*

```
!*****
! Now have 7 Bytes Data1 + (Data2+Data3) + (Data4+Data5) + (Data6 + Data7)
! Set Sequence Capture + CaptureTime + WaitTime + (No of Samples)
!*****
```

*! Send the Sequence Capture Setup command 0xC0 + Capture Time + Wait Time +
! Number of captures*

test"4%feasa_20i%la_seq_capt_setup_ss";Data1(),Data2(*),Data3(*),Data4(*),Data5(*)*

FEASA LED ANALYSER

Agilent 3070 Application Note



```

!*****

!Send the CaptureSequence command CS 0x45

test "4%feasa_20i%la_seq_capture_ss"          ! Do the Capture Sequence

!*****

SequenceCapture_results:

for F = 1 to 20  ! 20 Fibers used only in Feasa example indicator

    for Data_bit = 0 to 7  !Convet the fiber number to binary
        Fibernum(Data_bit) = bit(F,(7-Data_bit))
    next Data_bit

!print Fibernum(*)

No_of_Captures = 050  ! This number cannot be greater that detailed above in the setup

    for Sequence_Num = 1 to No_of_Captures

CommandBin$ = itb$(Sequence_Num)                ! Convert to Binary
CommandBin$ = "0000000000000000" & CommandBin$  ! Add leading zeros
CommandBin$ = CommandBin$[len(CommandBin$)-15;16] ! Extract the last 16 bits

!print CommandBin$

    for Data_bit = 1 to 16    ! Convet the fiber number to binary
        Data(Data_bit-1) = val(CommandBin$[Data_bit;1])
    next Data_bit

test "4%feasa_20i%la_seq_rd_fib_ca_num_ss" ; Fibernum(*), Data(*) ! Read Int

!*! Uncomment the following 5 lines for debug purposes
!*!
! print Data(0);Data(1);Data(2);Data(3);Data(4);Data(5)      ! print Data
Intensity = ((Data(0) * 65536) + (Data(1) * 256) + Data(2)) ! Intensity

! print F;" ";Sequence_Num;" "; Intensity                ! print HSI
! output @Logdata; "Fiber = ";F;" "; "Capture No = ";Sequence_Num;" ";Intensity
! output @Logdata; "-----"
next Sequence_Num

```

FEASA LED ANALYSER

Agilent 3070 Application Note



```
!=====
!send the Intensity threshold - 3 bytes for intensity
!get the sequence times for Fiber F
```

```
Intensity_threshold = 3000      ! Threshold for Intensity level which will
                                ! give a 1 for > than and a 0 for < than
```

```
CommandBin$ = itb$(Intensity_threshold)      ! Convert to Binary
CommandBin$ = "000000000000000000000000" & CommandBin$ ! Add leading zeros
CommandBin$ = CommandBin$[len(CommandBin$)-23;24] ! Extract the last 16 bits
```

```
!print CommandBin$
```

```
for Data_bit = 1 to 24      ! Convet the fiber number to binary
  Data(Data_bit-1) = val(CommandBin$[Data_bit;1])
next Data_bit
```

```
! print Data(*)
```

```
test "4%feasa_20i%la_seq_set_threshold_ss";Data(*) ! Set Intensity Threshold
```

```
!*****
! get the sequence times for Fiber F
```

```
test "4%feasa_20i%la_seq_rd_fib_times_ss";Fibernum(*),Data(*)
```

```
Time_off_begin = ((Data(0) * 256) + Data(1)) ! Time off at the beginning
Time_on = ((Data(2) * 256) + Data(3))        ! Time on
Time_off_end = ((Data(4) * 256) + Data(5))    ! Time off at the end
```

```
print "Fiber = ";F;" Time off Beg = "; Time_off_begin; "Time on = "; Time_on;"Time off end
= "; Time_off_end
```

```
output @Logdata; "Fiber = ";F;" "; "Toff Beg = ";Time_off_begin;"Ton =";Time_on;" Toff End
= "; Time_off_end
```

```
!=====
```

```
next F
```

```
End_LA:
```

```
end if
```

[Back to Index](#)

FEASA LED ANALYSER

Agilent 3070 Application Note



LA_Both_Modes \ testplan

This testplan is a combination of the two methods described above. Please refer to the documentation on the CD provided. Please read all the comments sections included in the testplan as they will explain the process.

[*Back to Index*](#)