



Feasa Legend ICT port

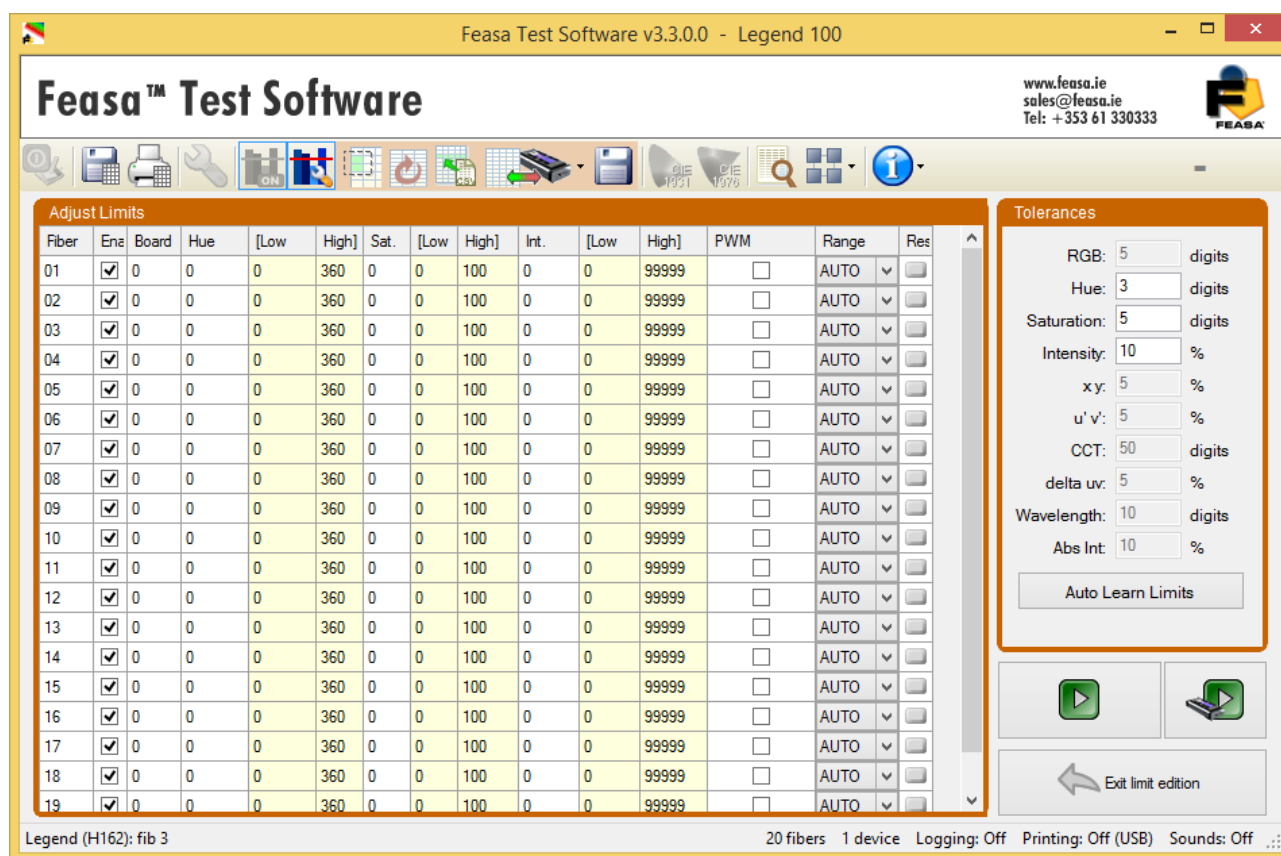
reference guide
rev. 3

1. Introduction

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

The behaviour of this port can be easily configured using a program called “Feasa ICTconfig”, provided in the Legend CD, but before start using the port from any external hardware, the embedded test has to be configured.

The Legend contains an **embedded test** through which we can indicate the test range to be used and also the limits for Hue, Saturation, Intensity, xy Chromaticity, Wavelength and CCT for each one of the fibers. The process of setting up this test is done through the **Feasa Test Software**, which is contained in the Feasa Legend CD.



Through this software, the user can decide which fibers to test as well as which data has to be checked for each fiber (Hue, Saturation...).

Once the test has been configured, it is mandatory to transfer the settings into the Legend Hub, and then it is recommended execute it in from the Test Software in order to be sure that everything behaves as expected.

1.1 Configure test limits with Test Software

In order to be able to perform a test using the embedded functional test of the Legend, we need to instruct the Legend to perform a test of LEDs based on some Low/High limits for the measured values. This used to be done through the Feasa Limits software, but now this functionality has been integrated into the Feasa Test Software.

Follow this instructions for a basic setup:

1. Start program
2. Create a new project
3. From the setup window, add the Legend to the hardware list and then select the data types to be tested from the Grid settings tab.
4. Click the button to enable the functional test if it appears as “Off” and then click the “Set limits” button.
5. Disable the fibers that won’t be used, as well as the data types that won’t be tested on the specific fibers.
6. Set the limits of the test either manually or learn them using the button “Auto Learn Limits”. A measurement will be done here, so ensure that all LEDs are on and the fibers or optical heads are placed on top of the LEDs.
7. Upload the limits on the Legend using the button “Transfer Limits” from the toolbar at the top.
8. Perform a test on the Legend using the test button on the right (green play button with Legend picture on it) to ensure that everything works as expected.
9. Exit limits edition and save the project data.
10. Close the Test Software

2. Voltage ratings

Minimum input voltage: 0.0V

Maximum input voltage: 5V

Minimum output voltage: 0.0V

Maximum output voltage: 3.3V

Maximum output current per pin: 10mA

$V_{IL} < 0.8V$

$V_{IH} \geq 2.0V$

$V_{OL} \leq 0.6V$

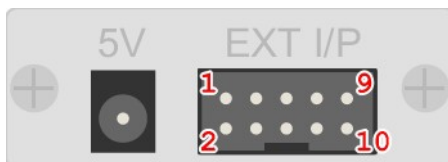
$V_{OH} \geq 2.2V$

Important:

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

3. Pinout

The testing functionality explained in this document is provided through the ICT port, which can be found on one side of the Legend Hub unit, as seen in the picture below.



Picture 1: ICT and power connector in the Legend Hub

The ICT connector has 10 pins which function is described in the table below:

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

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

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

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

4. Functional description

4.1 Limit group

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

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

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

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

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

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

4.2 Test

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

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

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

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

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

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

Capture + no output data

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

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

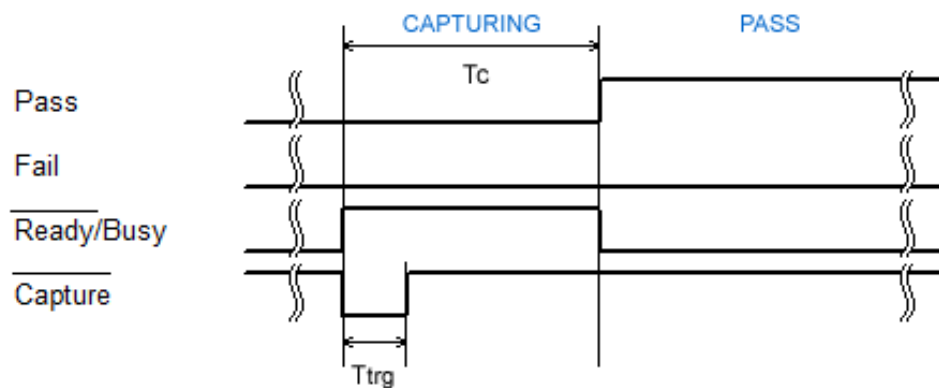


Diagram 1: Capture sequence with PASS result

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

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

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

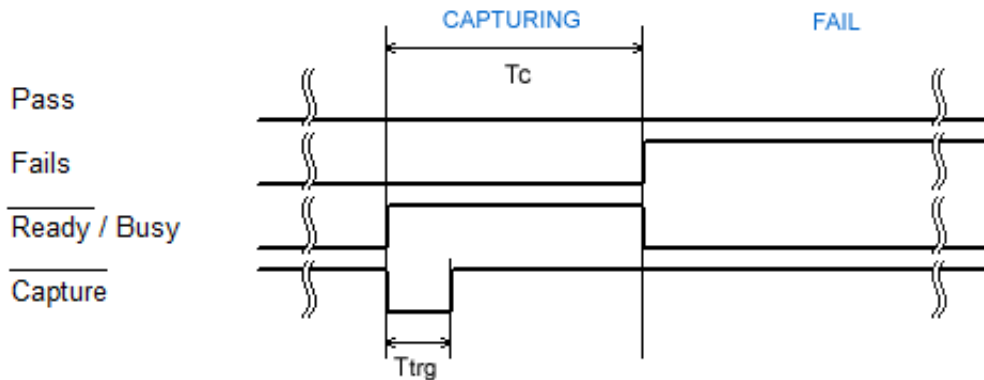


Diagram 2: Capture sequence with FAIL result

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

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

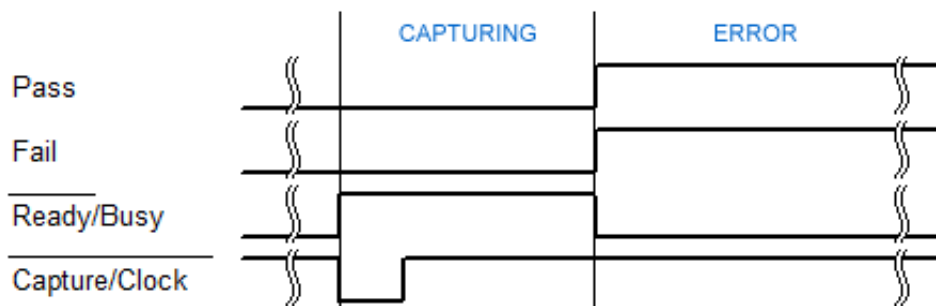


Diagram 3: Capture sequence with ERROR result

Set-up:

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

```
SETICTOUTMODEASYNC  
SETICTOUTDATANONE
```

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

SETICTOUTMODESYNC
SETICTOUTDATANONE

Capture + failing fibers output (Asynchronous)

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

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

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

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

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

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

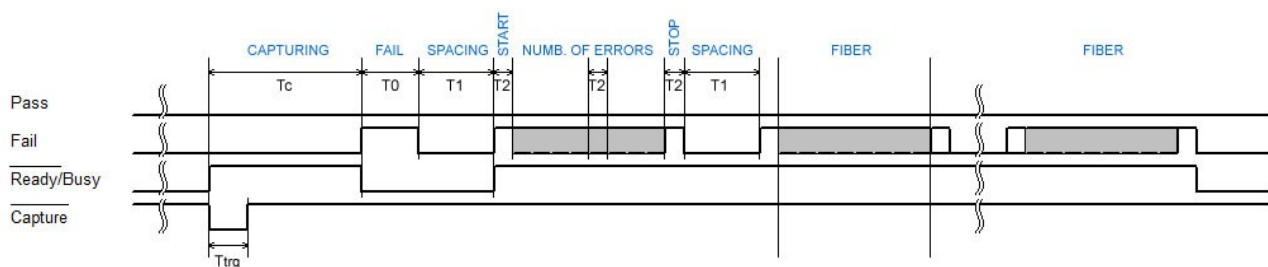


Diagram 4: Capture sequence with FAIL result + error output

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

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

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

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

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

START	LSB	B1	B2	B3	B4	B5	B6	MSB	STOP
-------	-----	----	----	----	----	----	----	-----	------

1	0	1	0	0	0	0	0	0	1
1	1	1	0	0	0	0	0	0	1
1	0	0	1	1	0	0	0	0	1

Set-up:

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

SETICTOUTMODEASYNC

SETICTOUTDATAFIBER

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

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

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

Capture + failing fibers + limits output (Asynchronous)

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

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

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

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

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

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

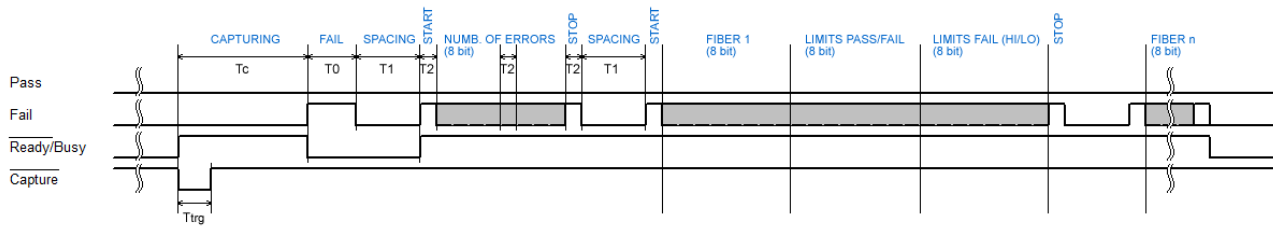


Diagram 5: Capture sequence with FAIL result + error + limits output

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

$$10\mu\text{s} \leq T_0 \leq 999\mu\text{s} \text{ (user configurable)}$$

$$10\mu\text{s} \leq T_1 \leq 999\mu\text{s} \text{ (user configurable)}$$

$$10\mu\text{s} \leq T_2 \leq 999\mu\text{s} \text{ (user configurable)}$$

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

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

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

Table 1: Fail flags byte format

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

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

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

1	0	0	1	0	0	0	0	0	1
---	---	---	---	---	---	---	---	---	---

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

Set-up:

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

```
SETICTOUTMODEASYNC
```

```
SETICTOUTDATALIMITS
```

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

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

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

Capture + failing fibers output (Synchronous)

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

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

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

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

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

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

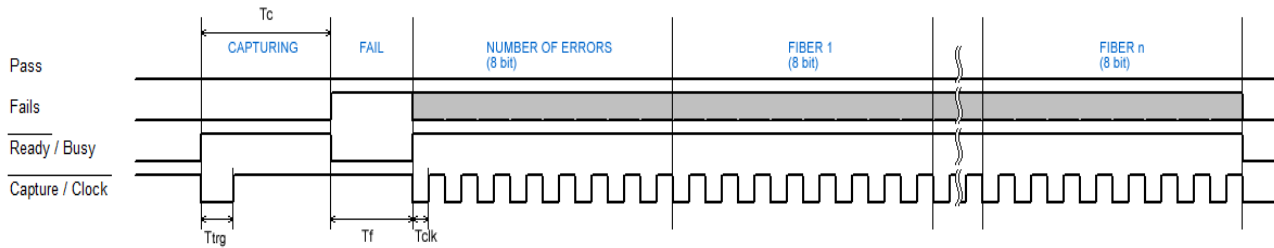


Diagram 6: Capture sequence with FAIL result + error output

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

$$100\mu s \leq T_F < T_{err}$$

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

$$T_{err} \geq 65ms \text{ (Timeout)}$$

Clocking for data retrieval should start in an interval higher than 100μs and lower than T_{err} .

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

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

Set-up:

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

SETICTOUTMODESYNC

SETICTOUTDATAFIBER

Capture + failing fibers + limits output (Synchronous)

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

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

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

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

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

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

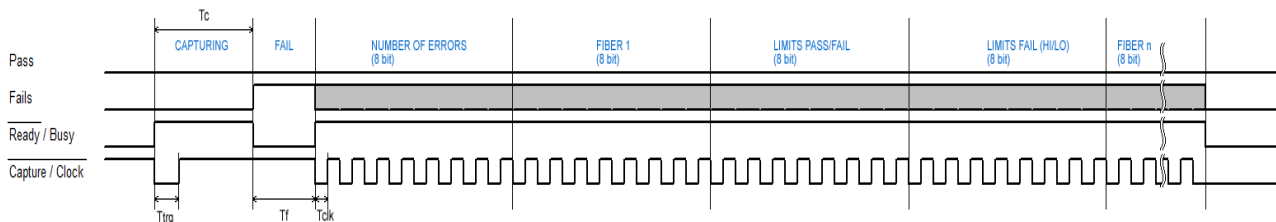


Diagram 7: Capture sequence with FAIL result + error + limits output

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

$$100\mu s \leq T_F < T_{err}$$

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

$$T_{err} \geq 65ms \text{ (Timeout)}$$

Clocking for data retrieval should start in an interval higher than 100μs and lower than T_{err}.

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

Feasa Legend · ICT port reference guide

by R&D Engineer Carles Martínez Rius, rev 3, © Feasa Enterprises Ltd.

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

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

Table 3: Fail flags byte format

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

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

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

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

Set-up:

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

SETICTOUTMODESYNC
 SETICTOUTDATALIMITS

Handling clocking errors (Synchronous)

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

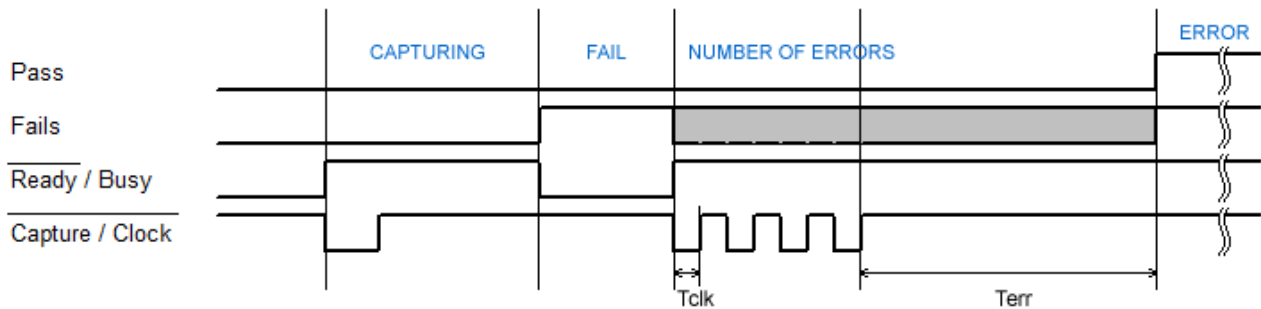


Diagram 8: clock interrupted before the data transmission ends

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

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

This method could also be used to interrupt the communication.