PIC Test Control Software Customer User Guide

EVT6 – PTCS Customer User Guide

Version 3.0

|  |  |
| --- | --- |
|  | |
| Author | Date |
| Mark Nash, Owen Shriver | August 12, 2019 |
| Rohan Patil, Garrett DeLang | December 13, 2019 |
|  | |

|  |  |  |
| --- | --- | --- |
|  | | |
| Revision | Date | Description |
| 1.0 | 8/12/2019 | Initial release |
| 2.0 | 8/22/2019 | After Sidney’s Review |
| 3.0 | 12/13/2019 | Rohan/Garrett Review |
| 3.1 | 1/17/2020 | Added instructions to clear queue, added hardware tab screenshot |
| 3.2 | 2/11/2020 | Updated hardware tab screenshot again |
| 3.3 | 2/27/2020 | Updated pictures for the test parameter right panel since it now shows a test description and the instruments it will attempt to use |
| 3.4 | 3/5/2020 | Added description of test build reduction |

Table of Contents

[1.0 Introduction 4](#_Toc37929207)

[1.1 Premade Tests 4](#_Toc37929208)

[1.2 Instruments with Drivers Written 4](#_Toc37929209)

[2.0 Running a test that does not connect to instruments 5](#_Toc37929210)

[2.1 Run PTCS 5](#_Toc37929211)

[2.2 Create a simple queue 6](#_Toc37929212)

[2.3 Remove an experiment from the queue 7](#_Toc37929213)

[2.4 Change an experiment parameter 8](#_Toc37929214)

[2.5 Save a queue 9](#_Toc37929215)

[2.6 Clear the Current Queue 10](#_Toc37929216)

[2.7 Load a queue from startup 11](#_Toc37929217)

[2.8 Run the queue 12](#_Toc37929218)

[2.9 View the Results 13](#_Toc37929219)

[3.0 View instruments that can be connected to the application 14](#_Toc37929220)

[3.1 View the Hardware tab 14](#_Toc37929221)

[4.0 Using the Test Build tab 15](#_Toc37929222)

[4.1 Functionality 15](#_Toc37929223)

[4.1.1 Adding Commands 15](#_Toc37929224)

[4.1.2 Reduction Functionality 16](#_Toc37929225)

[4.2 Build a real Voltage Accuracy Test 17](#_Toc37929226)

[4.2.1 Enter all the commands 17](#_Toc37929227)

[4.2.2 Save the experiment 18](#_Toc37929228)

[4.2.3 Run the experiment 18](#_Toc37929229)

[4.2.4 View the results 18](#_Toc37929230)

[5.0 Running Premade Tests Using VCU108: BERT/Eyescan 19](#_Toc37929231)

[6.0 Running Premade Tests Using Instruments: Voltage Accuracy Test 20](#_Toc37929232)

[6.1 Running Premade Tests Using Instruments: Optical Power Sweep Test 22](#_Toc37929233)

[6.2 Running Premade Tests with Instruments: Optical Wavelength Test 24](#_Toc37929234)

[7.0 Appendix: PTCS Codebase README File 27](#_Toc37929235)

[7.1 What is *PTCS* ? 27](#_Toc37929236)

[7.2 User Guide 29](#_Toc37929237)

[7.3 Developer Documentation 31](#_Toc37929238)

EVT6 – PTCS Customer User Guide

# Introduction

This software allows a user to create and run tests on Photonic Integrated Circuits (PICs) using different off-the-shelf instruments and semi-custom platforms.

Before running the following tutorials, set up your computer with the correct software following document EVT9.

## Premade Tests

* BERT using the Xilinx VCU108 semi-custom test platform
* Eyescan using the Xilinx VCU108 semi-custom test platform
* BERT and Eyescan using the BERTWave test instrument
* Voltage Accuracy Test using the power source and the oscilloscope
* Power Sweep Test using the Laser and the Optical Power Meter
* Wavelength Sweep Test using the Laser and the Optical Power Meter

## Instruments with Drivers Written

* Voltage/power source (Agilent E364XA)
* Logic Analyzer (Agilent 16802A)
* Oscilloscope (Agilent DSO/MSO 7XXXA)
* BERTWave (Anritsu MP2100A)
* Newport Optical Power Meter (through Prologix GPIB to USB Adapter)
* Laser Source (ANDO AQ4321D) (through the Prologix GPIB to USB Adapter)

There are more drivers written, but they have not been tested yet.

# Running a test that does not connect to instruments

This procedure guides a user in exploring the Queue and Results tabs of the user interface. This same procedure can be used to run tests that do use actual instruments. All that is necessary is the instrument(s) to be attached to the computer on the correct port(s), but for complexity sake, this procedure will not demonstrate instrument connection.

## Run PTCS

Double-click on the desktop icon labeled ***PTCS.*** The following window should appear

|  |
| --- |
|  |
|  |
| Figure 1: The Queue tab allows a user to build, run, save and load a queue of experiments. |
|  |

## Create a simple queue

Select the ***Fake Voltage Accuracy Test*** from the drop-down and then select the ***Add*** button 3 times. The queue will now contain 3 experiment as shown. The Fake Voltage Accuracy test is a pre-made void experiment.

|  |
| --- |
|  |
|  |
| Figure 2: The queue has 3 loaded Fake Voltage Accuracy Test experiments in it. |
|  |

## Remove an experiment from the queue

If you decide you only want to run two Fake Voltage Accuracy Tests, select any of the entries in the queue and select the ***Remove*** button.

|  |
| --- |
|  |
|  |
| Figure 3: An experiment in the queue is selected and a remove button appears |
|  |

## Change an experiment parameter

Experiments can be made to expose parameters to the user. If you decide you only want to test 5 levels on the second voltage accuracy test, select the second voltage accuracy test from the queue and change the configurable parameter named **Levels** to 5 instead of 10.

|  |
| --- |
|  |
|  |
| Figure 4: A parameter of an experiment has been changed to the value 5 |

|  |
| --- |
|  |

## Save a queue

To save the queue for later use: Type “my queue” into the text box to the right of the ***Save Queue*** button, and then click said button.

|  |
| --- |
|  |
|  |
| Figure 5: The queue is about to be saved under the name “my queue” |
|  |

## Clear the Current Queue

To remove all the experiments in the current queue without having to select each one and then selecting the remove button, simply select the ***Clear Queue*** button.

|  |
| --- |
|  |
|  |
| Figure 6: The queue has been cleared by the **Clear Queue** button. |
|  |

## Load a queue from startup

Close the program by selecting the X in the top right corner. You come back sometime later and you are ready to run the experiments in your queue. Reopen the program. From the drop-down just to the right of the ***Load Queue*** button, select the entry “my queue”

|  |
| --- |
|  |
|  |
| Figure 7: The queue named “my queue” is selected and is about to be loaded into the current queue |
|  |

## Run the queue

Select the button ***Run Queue***. Note that the info box shown on the right displays the status output of the queue and its experiments. The first experiment should have applied from 0 to 9 fake volts. The second test should have applied from 0 to 4 fake volts. Select the button ***Back to Test List***.

|  |
| --- |
|  |
|  |
| Figure 8: The current queue has been run |
|  |

## View the Results

Navigate to the ***Results*** tab. On the left panel, the bottom most entry should contain two experiments in order of the time run. Select the first of the two entries.

The right panel should display files that were generated by that experiment. Double click on the .png file to open it in the operating system’s default photo viewer and verify it has 10 data points plotted. Next, select the second of the two entries in the left panel. Open that experiment’s .png file. Verify it has 5 points.

|  |
| --- |
|  |
|  |
| Figure 9: The first experiment of the last queue run has been selected and its generated files are shown on the right. |
|  |

# View instruments that can be connected to the application

## View the Hardware tab

Open the application and navigate to the hardware tab. On the left are the names of instruments that have been configured to interact with the application, if these are connected to the computer in the right configuration. Additional instruments can be connected by adding instrument specific drivers – see the PTCS Developers Guide EVT11.

|  |
| --- |
|  |
|  |
| Figure 10: The devices the software knows how to connect with |
|  |

# Using the Test Build tab

The final tab is the Test Build tab, which allows the creation of experiments without the knowledge of a specific programing language, though it is useful to have some programing background.

|  |
| --- |
|  |
|  |
| Figure 11: The Test Build tab |
|  |

## Functionality

### Adding Commands

To build a test, it is necessary to add a series of lines that provide instructions for how the test should be carried out. The following table shows the different instructions that can be used by selecting different buttons:

|  |  |  |  |
| --- | --- | --- | --- |
| Keyword | Effect | Parameters required | Example |
| IF | Executes the next block of instructions only if a certain condition is met. | Two values and an equals/inequality sign to compare them. | IF x < 5 |
| ELSE | Executes the next block of instructions only if the condition in an IF was not met. | None | ELSE |
| END | Ends a block of instructions to be executed by IF, ELSE, or LOOP. | None | END |
| LOOP | Executes the next block of instruction as long as a certain condition is met. | Two values and an equals/inequality sign to compare them. | LOOP voltage >= 0 |
| PARAMETER | Adds a parameter to the test that can be set in the Queue Page (see Figure 3). | Parameter name and default parameter value. | PARAMETER start 1 |
| SET | Sets a variable value. This can be a newly defined variable, or one already used in this test. | A variable name and a variable value (which could be another variable, or an expression containing one). | SET max = 100 |
| PRINT | Prints out a string, which must be in quotes. | A string, or multiple strings added together by + signs | PRINT “Time taken: “ + time + “ seconds” |
| SAVE | Save an ordered pair of values to be displayed in the results directory. | Two values to save | SAVE time, voltage |
| START TIMER | Starts the timer. | None | START TIMER |
| GET TIMER | Gets the time since the timer was started. | A variable to read this time as. | GET TIMER AS time |
| DEVICE CALL | Call a function from a connected device. | The device to call from, the function to call, and whatever parameters the function requires (separated by spaces). | FROM Laser\_Source CALL run\_sweep\_step 1520 1530 1 2 |
| DEVICE READ | Read a value from a function from a connected device. | The device to call from, the function to call, whatever parameters the function requires (separated by spaces), and a variable name to read the result as. | FROM Newport OPM READ get\_power\_reading AS power |

### Reduction Functionality

If a test is utilizing the SAVE command specified above, to make it easier to parse through data from a test, the Test Build Tab has functionality to screen out invalid data in a range. For example, if a test has to take power measurements from a laser, there may be initial readings that are totally off of what they should be because the laser was still initializing when the power meter was starting to read. Since we know this is going to happen, we will remove all values from Ymin to Ymax. This can be done by inputting specific Ymin and Ymax values in the text boxes to the left and right of ***< Y <*** before the test is saved. This can also be done for the X values.

## Build a real Voltage Accuracy Test

Pre-requisite: Have a developer connect the necessary devices to the computer and verify that they work with a computer.

### Enter all the commands

- Create a PARAMETER called **start\_voltage** with a value of 1

- Create a PARAMETER called **end\_voltage** with a value of 5

- Create a PARAMETER called **step\_voltage** with a value of 1

- Add a DEVICE CALL from the **Voltage\_Source** call **set\_output\_switch** with the argument of 1

- START the TIMER

- Create a new variable by SETting **voltage** to **start\_voltage**

- Start a LOOP WHILE **voltage** <= **end\_voltage**

- Add a DEVICE CALL from the **Voltage\_Source** call **set\_voltage** with the parameter of **voltage**

**-** Add a PRINT statement with the value of: "should be reading " + voltage + " volts from the oscilloscope"

**-** Add a DEVICE CALL from the **Oscilloscope** call **autoscale** with no parameters

**-** Add a DEVICE READ from the **Oscilloscope** call **measure\_vaverage** with no paramaters and set that value equal to **reading**

- SAVE (will save to a file) said reading with a name of **voltage** and a value of **reading**

- Add a PRINT statement with the value of: "voltage read from osciliscope: " + **reading**

- Increment the voltage by SETting **voltage** to **voltage** + **step\_voltage**

- END the loop

- Turn the output off on the voltage source by DEVICE CALLing **set\_output\_switch** on the **Voltage\_Source** with a value of 0

- GET the TIMER as the name **tmr**

- Add a PRINT statement with the value of: "measurements took: " + tmr + " seconds"

At the end, the UI will show the following:

|  |
| --- |
|  |
|  |
| Figure 12: The build experiment from following the steps above. |
|  |

### Save the experiment

- Check the box to the right of ***CSV?***

- Click the ***Save As*** button and enter the test name as **Built VAT**

### Run the experiment

- Navigate to the ***Queue*** tab. Add a ***Built VAT*** experiment to the queue

- Change some of the experiment’s parameters if you want to

- Select the ***Run Queue*** button

- Watch the experiment run through the given voltage values and watch the queue finish

### View the results

- Close and reopen the application

- Navigate to the ***Results*** tab

- Select the only experiment in the latest run queue (all the way at the bottom)

- Double click on the .csv file. This will open with your default csv viewer. Observe how the first column has the set voltages from the voltage source, and the second column has the values taken from the oscilloscope.

# Running Premade Tests Using VCU108: BERT/Eyescan

1. Run the file ‘PICTestControlSoftware.pyw’ found in the PTCS3 directory or as a shortcut on the desktop.
2. The tests used here are all based on TCL files. Add to the Queue:
   1. Initialize VCU108
   2. Program VCU108 with IBERT *or*Program VCU108 with PETB
   3. BERT (tcl)
   4. Eyescan (tcl)
   5. Close VCU108
3. Click on the test named Program IBERT or Program PETB to change the speed at which it runs. The acceptable values are 2.5 (needs to be typed as 2\_5), 5, 10, 15, and 25 Gb/s.

|  |
| --- |
|  |
|  |
| Figure 13: The queue named “my queue” is selected and is about to be loaded into the current queue |
|  |

1. Make sure the USBs are connected to the VCU108 for JTAG and UART communication, the VCU108 is powered on, and a QSFP device is plugged in.
2. Optionally, you can save your queue at this point to easily run it again.
3. Click Run Queue. The results will be found in the Results tab.

# Running Premade Tests Using Instruments: Voltage Accuracy Test

1. Ensure connection of both instruments DSO7054A (oscilloscope) and Agilent E3649A (power supply) to the PC running the PTCS application. Turn both on.
2. Hook up a set of probes up to both the oscilloscope and power supply.
3. Hook up black probes from both instruments together. Hook up the red probe from power supply to the remaining gray probe from the oscilloscope.
4. Run PTCS
5. Select dropdown under Pre-defined tests, and select “Voltage Accuracy Test”. Select “Add”.

|  |
| --- |
|  |
|  |
| Figure 14: The Voltage Accuracy Test added to the queue |
|  |

1. Select “Run Queue”.

|  |
| --- |
|  |
|  |
| Figure 15: The Voltage Accuracy Test has been run |
|  |
|  |
|  |
| Figure 16: Expected Voltage Accuracy Results |
|  |

## Running Premade Tests Using Instruments: Optical Power Sweep Test

1. Connect Newport OPM to ANDO laser with GPIB - GPIB cable.
2. Plug GPIB-USB adapter into reverse end of GPIB-GPIB cable.
3. Plug in USB A to USB B card into computer and adapter.

|  |
| --- |
|  |
|  |
| Figure 17: Newport OPM and ANDO Laser Connections |
|  |

1. Ensure optical power detector is plugged into Newport OPM and ensure attenuator is NOT screwed onto detector.
2. Prop the optical power detector vertically and bring it as close to the laser as possible.
3. Turn laser and Newport OPM on, and adjust power detector until the detector detects the highest power. Change prop mechanism to hold detector in place.

|  |
| --- |
|  |
|  |
| Figure 18: Optical Power Detector Propped |
|  |

1. Enter laser password found on top of the ANDO laser.
2. Run PTCS
3. Select dropdown under Pre-defined tests and select “Optical Power Sweep Test”. Hit “Add” to add to queue.
4. Select “Run Queue”.

|  |
| --- |
|  |
|  |
| Figure 19: Similar Plot Expected |
|  |

## Running Premade Tests with Instruments: Optical Wavelength Test

1. Connect Newport OPM to ANDO laser with GPIB - GPIB cable.
2. Plug GPIB-USB adapter into reverse end of GPIB-GPIB cable.
3. Plug in USB A to USB B card into computer and adapter.

|  |
| --- |
|  |
|  |
| Figure 20: Newport OPM and ANDO Laser Connections |
|  |

1. Ensure optical power detector is plugged into Newport OPM and ensure attenuator is NOT screwed onto detector.
2. Prop the optical power detector vertically and bring it as close to the laser as possible.
3. Turn laser and Newport OPM on, and adjust power detector until the detector detects the highest power. Change prop mechanism to hold detector in place.

|  |
| --- |
|  |
|  |
| Figure 21: Optical Power Detector Propped |
|  |

1. Enter laser password found on top of the ANDO laser.
2. Run PTCS
3. Select dropdown under Pre-defined tests and select “Optical Wavelength Sweep Test”. Hit “Add” to add to queue.
4. Select “Run Queue”.

|  |
| --- |
|  |
|  |
| Figure 22: Expected Results |
|  |

# Appendix: PTCS Codebase README File

## What is *PTCS* ?

PTCS is a software tool that lets a user create and run tests by remotely connecting to measurement instruments and integrated circuits.

PTCS is being developed by the Center for Detectors at Rochester Institute of Technology

#### Features of PTCS

* Create, save and load a queue of pre-built tests
* Open result files through the UI
* Build a test through a GUI

#### Where Can I get PTCS?

PTCS is currently available from the [Future Photon Initiative public GitHub repository.](https://github.com/FuturePhotonInitiative/PTCS)

## User Guide

The user guide for the project is EVT6 located in the EVT project formal documentation folder: \hawkdocuments. Start by reading this document as it gives an overview of the functionality of the software.

#### Requirements to Run

Reading this document will point you to EVT9 which details the requirements to run the software. You should read this as well, but you do not need to do anything found in it because Blackdog (the computer that is used for PTCS development) is already set up.

#### Running TCL tests

EVT8 has some documentation about creating TCL based tests. Currently most use cases for this software have been implemented by writing and auto-generating python scripts. Sometimes it is useful to run a test that communicates with Vivado. Vivado can operate in headless mode by giving it a TCL script.

## Developer Documentation

After you read the above user guides but before you start reading the developer documentation below, you should read the New Developer Startup guide located in the supporting developer documentation folder: \hawkdocumentsDocumentationFall2019

#### Main Developer Document

The first developer document you should read is EVT11. It is the document that gives context to and glues all the files in the supporting developer documentation folder together. The supporting developer documentation folder is pretty much a copy of the folder we (Owen and Mark) used in Summer, but it is cleaned up. Feel free to edit these documents in the folder as the software changes. If there are any files in EVT11 that are not mentioned (other than the remote instrument communication data sheets) make sure you read over and understand them also.