

# Quabo Auto Test Manual-V1.0.0

Wei Liu

April 1, 2025

## Contents

<b>1 Overview</b>	<b>1</b>
<b>2 Preparations</b>	<b>2</b>
2.1 Computer Setup . . . . .	2
2.1.1 Required software . . . . .	2
2.1.2 IP configuration . . . . .	2
2.1.3 QuaboAutoTest Software . . . . .	3
2.2 Hardware Setup . . . . .	3
2.2.1 Quabo Auto Test Kit . . . . .	3
2.2.2 Hardware connection . . . . .	4
<b>3 Auto Test Steps</b>	<b>8</b>
<b>4 Test Reports</b>	<b>12</b>

## 1 Overview

This is the manual for Quabo auto test. We will know if the Quabo is good or not by running the following tests:

- 1) **Housekeeping test**
  - a. **hk\_vals**: check all of the values in the HK packets, including high voltage, FPGA voltage, IP address and so on;
  - b. **hk\_time**: check the HK packets interval time.
- 2) **MAROC chip configuration test**
  - a. **maroc\_config**: check the MAROC chip configuration by writing/reading back the values to/from all of the registers.
- 3) **MAC address**
  - a. **mac**: check if the Microblaze core gets the correct MAC address for PH and MOVIE packets.
- 4) **White Rabbit Timing/Movie Mode**
  - a. **wr\_timing**: check the timestamps in the MOVIE packets to see if White Rabbit works or not, which includes checking the max nanosec value and the nanosec difference.
- 5) **PH mode(SiPM Simulator board is required)**
  - a. **ph\_pulse\_height**: check the mean/std/max/min of the pulse height;
  - b. **ph\_pluse\_rate**: check the pulse rate, which is related to SiPM simulator board setting;

- c. **ph\_peaks**: check how many pulses are in each PH packet;
- d. **ph\_pattern**: check the pulse pattern in PH events.

Note: Currently, the software only works on Linux machines.

## 2 Preparations

To do the Quabo auto test, you need to prepare for a **Ubuntu(Linux)** computer/laptop, and you also need the hardware test kit.

### 2.1 Computer Setup

#### 2.1.1 Required software

There are some required software on the computer/laptop:

- 1) git
- 2) python3.12 ("miniconda"<sup>1</sup> environment is highly recommended.)
- 3) "vivado 2018.3" (Vivado Lab Solutions is enough<sup>2</sup>)

Here is the way to check if the required software is installed or not:

- 1) check git

```
~$ git --version
git version 2.25.1
```

- 2) check python

```
~$ python -V
Python 3.12.9
```

- 3) check vivado

Please download it from Xilinx website.

#### 2.1.2 IP configuration

As Quabo will send packets to the computer, we need to configure the IP address of the ethernet port on the computer, to make sure the computer and Quabo are in the same subnet. By default, Quabo's IP address is 192.168.3.248, so the IP address of the computer has to be in 192.168.3 subnet, and the netmask should be 255.255.255.0. Here, we set the IP address to 192.168.3.2.

IP setting is different on different machines, so we just show how to check if the IP address is set up correctly here.

```
~$ ifconfig
enp6s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9000
    inet 192.168.3.2 netmask 255.255.255.0 broadcast 192.168.3.255
    inet6 fe80::2e0:4cff:fe68:71d7 prefixlen 64 scopeid 0x20<link>
    ether 00:e0:4c:68:71:d7 txqueuelen 1000 (Ethernet)
    RX packets 179618 bytes 72256224 (72.2 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 61370 bytes 32570089 (32.5 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

<sup>1</sup><https://www.anaconda.com/docs/getting-started/miniconda/install>

<sup>2</sup><https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/archive.html>

### 2.1.3 QuaboAutoTest Software

Next is to get the software and install required python packages.

- 1) clone the software

```
git clone https://github.com/liuweiseu/Quabo-AutoTest.git
```

- 2) create and activate conda environment(optional)

```
conda create -n quabo python=3.12
conda activate quabo
```

- 3) install required python packages

```
cd Quabo-AutoTest
pip install -r requirements.txt
```

## 2.2 Hardware Setup

### 2.2.1 Quabo Auto Test Kit

To do the Quabo auto test, you need the hardware test kit, which is shown in figure1.



Figure 1: Quabo Auto Test Kit.

The Quabo Auto Test Kit includes:

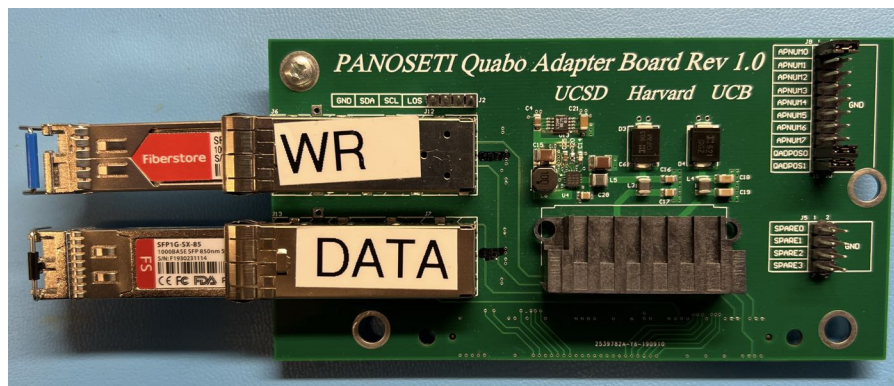
- 0) Quabo, which needs to be tested;
- 1) a power supply for powering the Quabo;
- 2) a WR-LEN for testing White Rabbit timing;
- 3) a Mini-Mobo, which will be attached to the Quabo;

- 4) a SiPM simulator board for testing all of the pixels;
- 5) a single mode, simplex fiber(the yellow one);
- 6) a multi mode, duplex fiber(the light blue one);
- 7) a media convertor for converting the optical signal to electric signal;
- 8) four 1Gb SFP transceivers(two multimode, duplex SFP transceivers and two single mode, simplex transceivers);
- 9) a JTAG;
- 10) an Ethernet cable (this is not shown in the figure).

### 2.2.2 Hardware connection

Here are the steps about how to connect the hardware parts in the Quabo Auto Test Kit:

- 1) Step 1: plug the two SFP transceivers to the Mini-Mobo



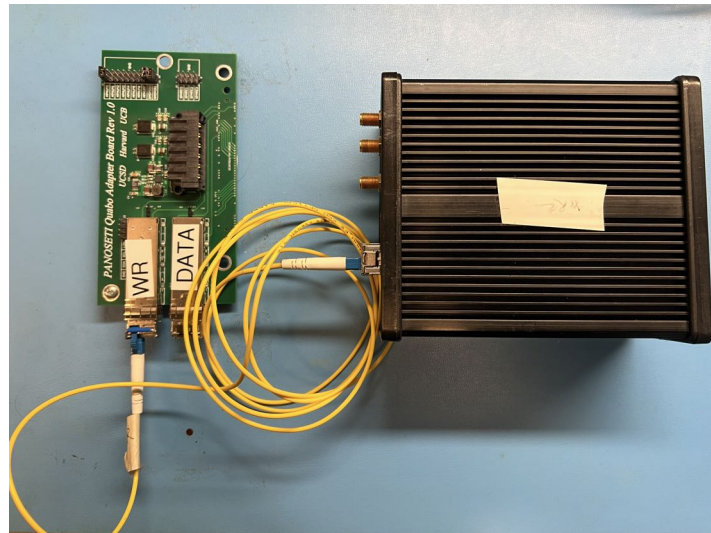
The single mode, simplex SFP transceiver with blue tab should be plugged into the *WR* cage, and a multimode, duplex mode SFP transceiver with black tab should be plugged into the *DATA* cage.

- 2) Step 2: plug the single mode, simplex SFP transceiver with purple tab into the SFP2 port of WR-LEN





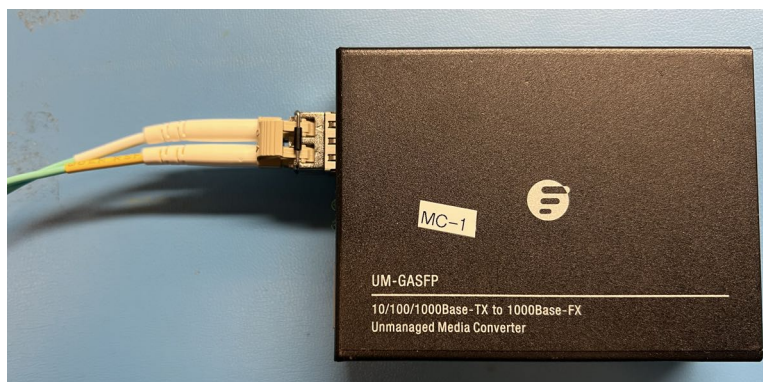
- 3) Step 3: connect the WR-LEN and Mini-Mobo through the single mode, simplex fiber(the yellow one)



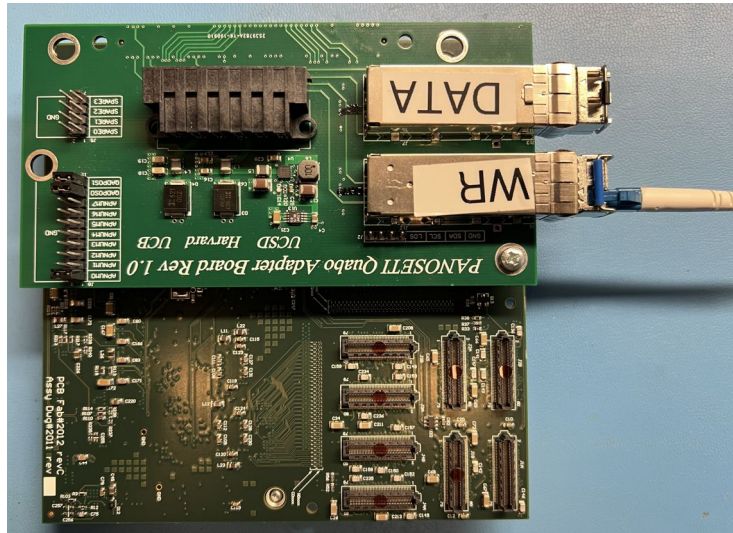
- 4) Step 4: plug another multimode, duplex SFP transceiver into the media convertor



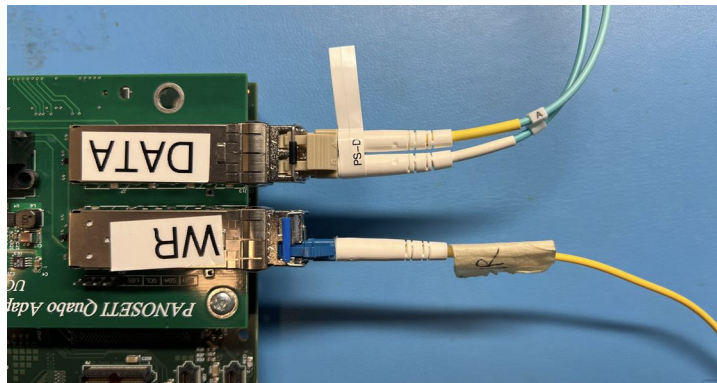
- 5) Step 5: connect the multimode, duplex mode fiber to the transceiver on the media convertor



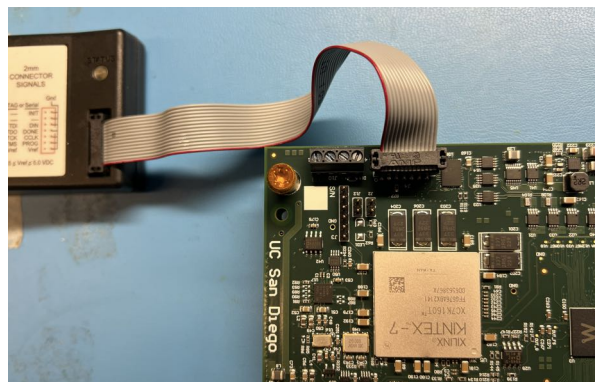
6) Step 6: attach the Mini-Mobo to the Quabo



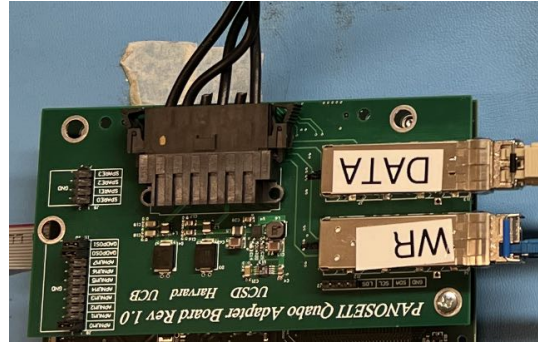
7) Step 7: connect the another end of the multimode, duplex fiber to the transceiver on the Mini-Mobo



8) Step 8: connect the JTAG to the connector J4 on the Quabo



9) Step 9: connect the power supply to the Mini-Mobo



Please make sure the power status is off at this moment.



(a) power on



(b) power off

The full setup is shown in figure 2. The SiPM simulator board is not attached to the Quabo for now. It will be used later.

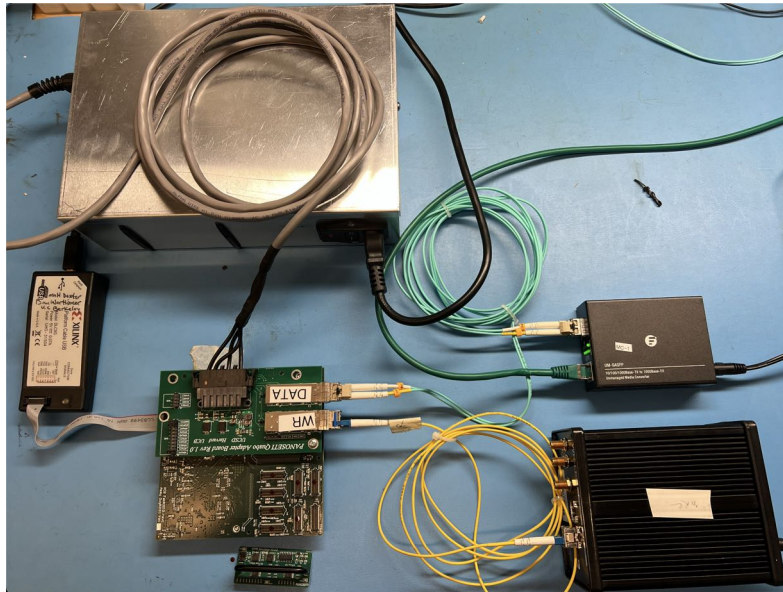


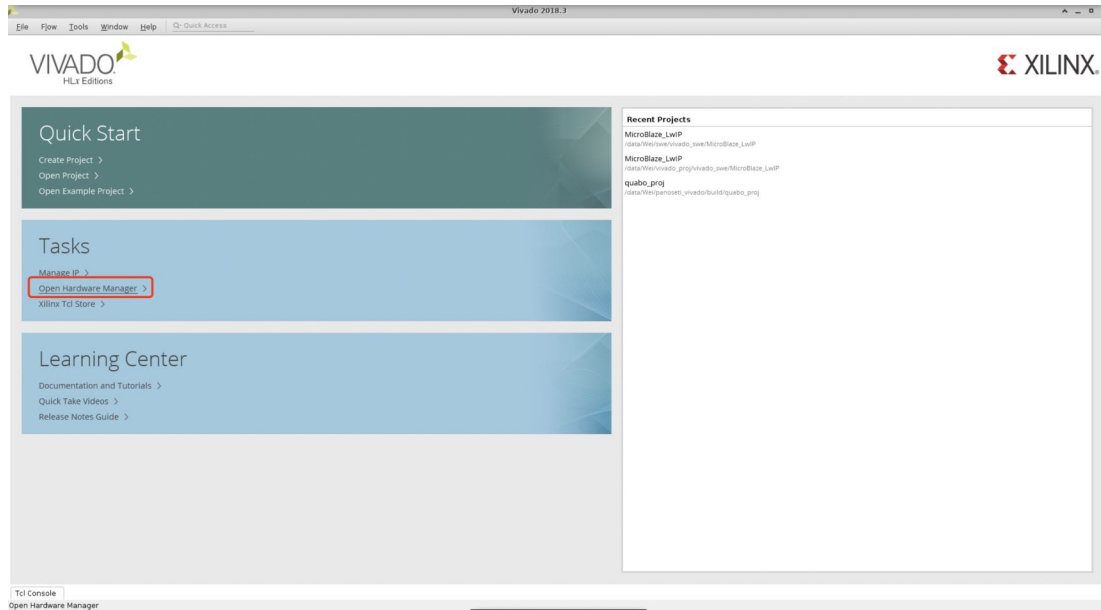
Figure 2: The full setup of the test environment.



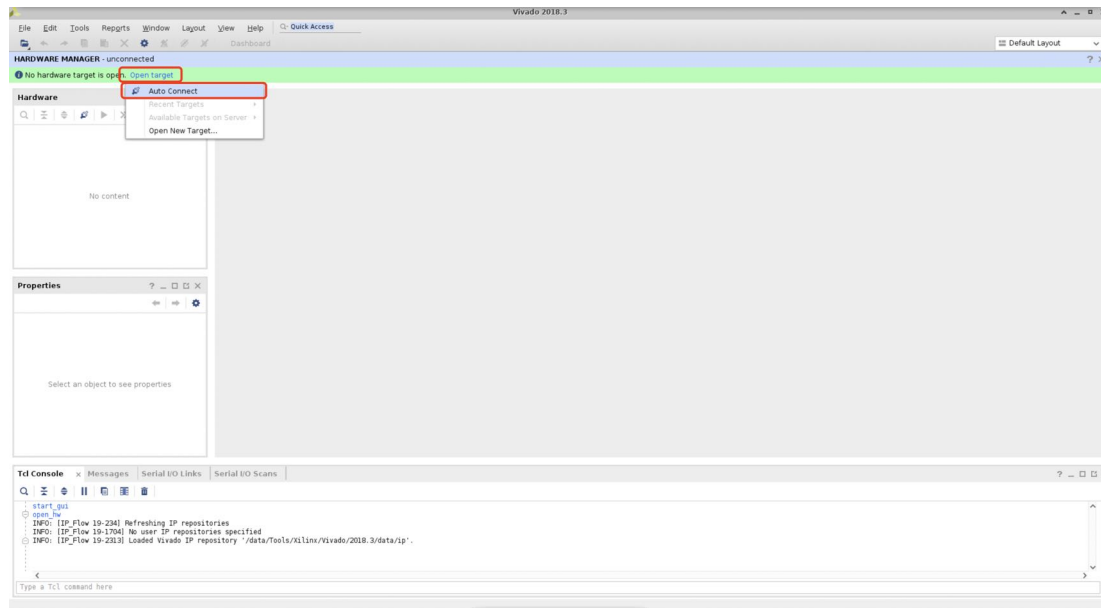
### 3 Auto Test Steps

Now, the computer, software, hardware should be fully setup. We will start to do the Quabo auto tests.

- 1) open Vivado hardware manager

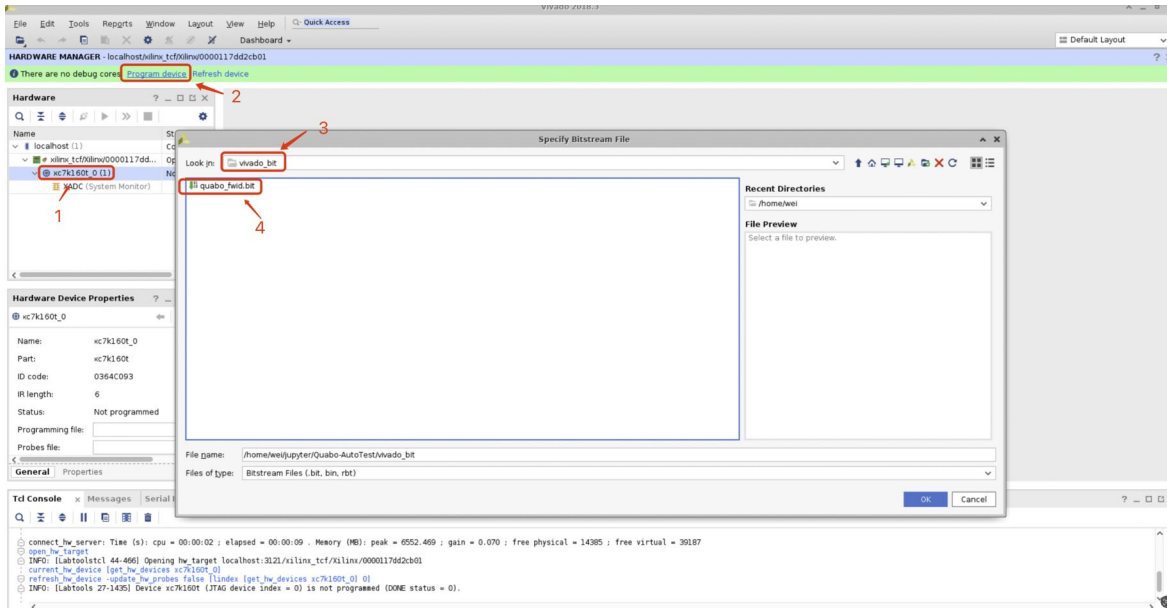


- 2) connect to the Quabo through JTAG

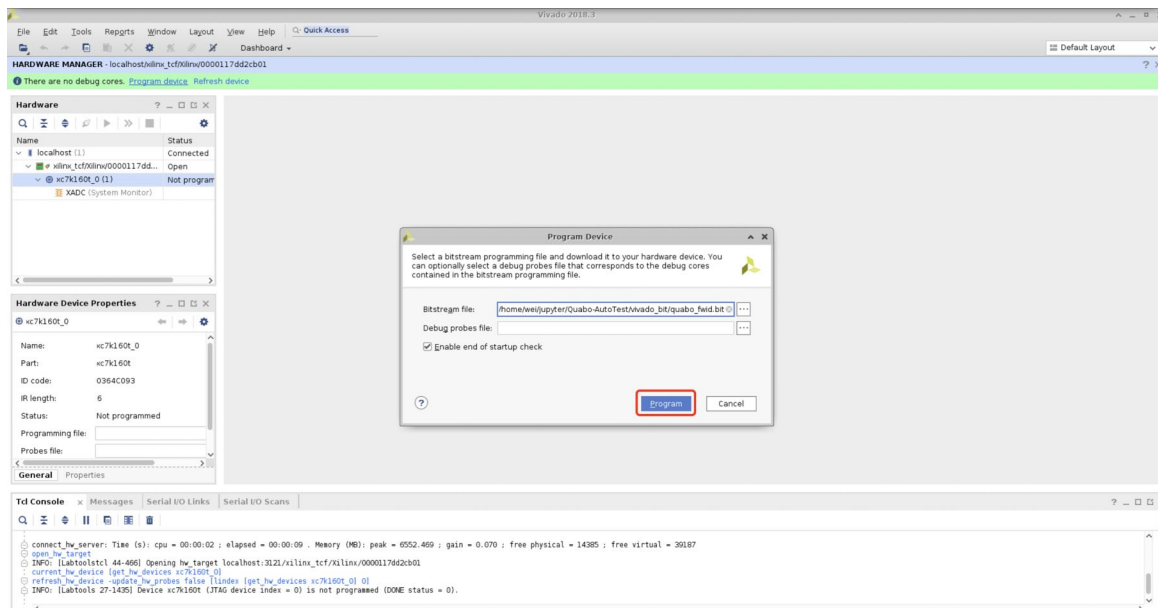




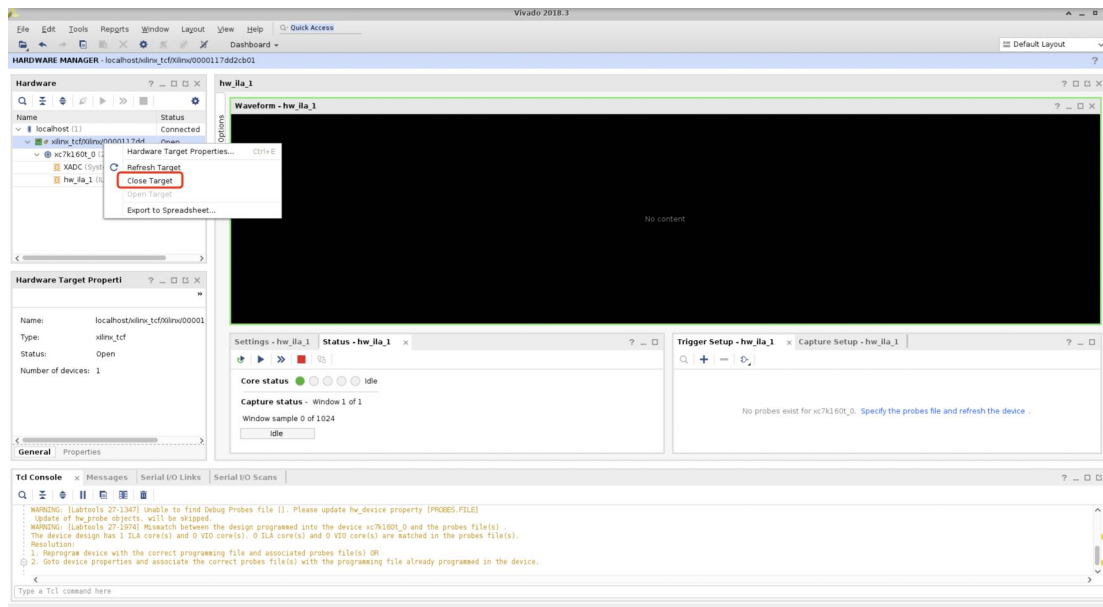
3) select the bit file from "Quabo-AutoTest/vivado\_bit"



4) program the Quabo



- 5) close connection between the JTAG and Quabo  
 Right click "xilinx\_tcf/xilinx...", and then click "Close Target".



- 6) upload firmware to the Quabo  
 Open a terminal, and run the following commands:

```
~$ cd Quabo-AutoTest
~$ python upload_firmware.py
```

If everything goes well, you should see the firmware is uploaded to the Quabo in **3min**.

```
(py312) wei@panoseti:~/jupyter/Quabo-AutoTest$ python upload_firmware.py
192.168.3.248 responded in 0.46 ms
Quabo is up and running at 192.168.3.248.
Quabo IP address: 192.168.3.248
Uploading golden firmware...
Golden firmware uploaded successfully.
Uploading silver firmware...
Silver firmware uploaded successfully.
Uploading wrpc filesystems...
Wrpc filesystems uploaded successfully.
Rebooting the Quabo...
192.168.3.248 responded in 376.13 ms
Quabo is up and running at 192.168.3.248.
Done.
```

- 7) test quabo  
 Run the following command to test the Quabo automatically:

```
python run_tests.py -t quabo
```

If everything goes well, you should see all of the tests are passed, and all of the tests should be done in **30 seconds**.

```

(py312) wei@panoseti:~/jupyter/Quabo-AutoTest$ python run_tests.py -t quabo
ping quabo: 192.168.3.248
192.168.3.248 responded in 0.27 ms
Reboot is not required.
ping quabo: 192.168.3.248
192.168.3.248 responded in 0.25 ms
===== test session starts =====
platform linux -- Python 3.12.9, pytest-8.3.5, pluggy-1.5.0 -- /home/wei/.conda/envs/py312/bin/python
cachedir: .pytest_cache
metadata: {'Python': '3.12.9', 'Platform': 'Linux-5.4.0-204-generic-x86_64-with-glibc2.31', 'Packages': {'pytest': '8.3.5', 'pluggy': '1.5.0'}, 'Plugins': {'metadata': '3.1.1', 'anyio': '4.9.0', 'html': '4.1.1'}}
rootdir: /home/wei/jupyter/Quabo-AutoTest/test_scripts
configfile: pytest.ini
plugins: metadata-3.1.1, anyio-4.9.0, html-4.1.1
collected 4 items

test_scripts/test_quabo.py::test_hk_vals PASSED [ 25%]
test_scripts/test_quabo.py::test_hk_time PASSED [ 50%]
test_scripts/test_quabo.py::test_maroc_config PASSED [ 75%]
test_scripts/test_quabo.py::test_wr_timing PASSED [100%]

----- Generated html report: file:///home/wei/jupyter/Quabo-AutoTest/reports/2d98d0002318206/reports_quabo.html -----
===== 4 passed in 29.51s =====

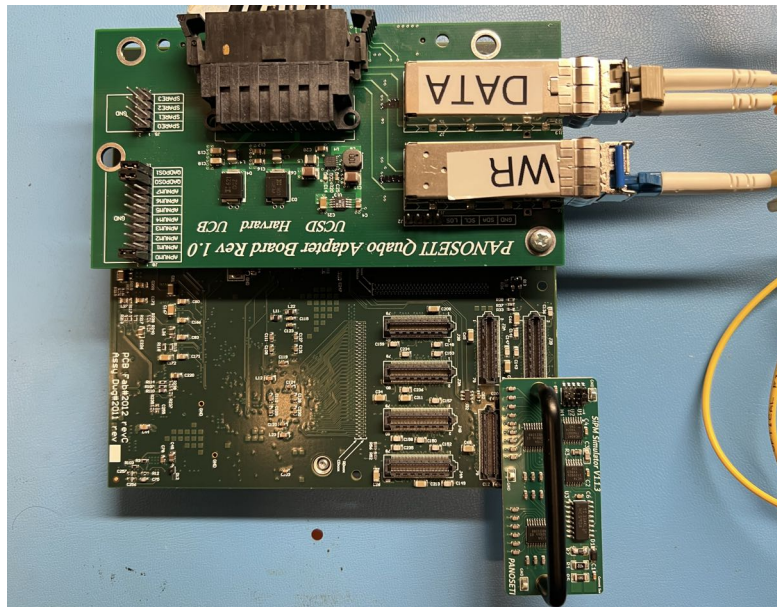
```

8) test all of the pixels with the SiPM simulator board

**Before doing this test, please make sure you turn off the power for the Quabo.**

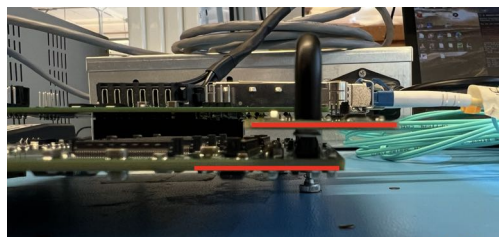
a. attach the SiPM simulator board to the Quabo

There are 8 connectors on the Quabo, and we need to test all of the connectors. It's recommended to start from J1A, then J1B, ..., J4A, J4B.



b. check the side view of the attached board

The SiPM simulator board should be parallel to the Quabo from the side view.





c. **power on the Quabo**

d. run the following command to test the pixels automatically

```
python run_tests.py -t sipmsim -c J1A -r
```

The `-c` option indicates which connector you're using for the test.

If everything goes well, you should see all of the tests passed in **2.5min**.

```
(py312) wei@panoseti:~/jupyter/Quabo-AutoTest$ python run_tests.py -t sipmsim -c J1A -r
ping quabo: 192.168.3.248
192.168.3.248 responded in 1.19 ms
192.168.3.248 responded in 0.32 ms
ping quabo: 192.168.3.248
192.168.3.248 responded in 0.27 ms
===== test session starts =====
platform linux -- Python 3.12.9, pytest-8.3.5, pluggy-1.5.0 -- /home/wei/.conda/envs/py312/bin/python
cachedir: .pytest_cache
metadata: {'Python': '3.12.9', 'Platform': 'Linux-5.4.0-204-generic-x86_64-with-glibc2.31', 'Packages': {'pytest': '8.3.5', 'pluggy': '1.5.0'}, 'Plugins': {'metadata': '3.1.1', 'anyio': '4.9.0', 'html': '4.1.1'}}
rootdir: /home/wei/jupyter/Quabo-AutoTest/test_scripts
configfile: pytest.ini
plugins: metadata-3.1.1, anyio-4.9.0, html-4.1.1
collected 5 items

test_scripts/test_sipmsim.py::test_mac PASSED [ 20%]
test_scripts/test_sipmsim.py::test_ph_peaks PASSED [ 40%]
test_scripts/test_sipmsim.py::test_ph_data PASSED [ 60%]
test_scripts/test_sipmsim.py::test_ph_timing PASSED [ 80%]
test_scripts/test_sipmsim.py::test_ph_pattern PASSED [100%]

----- Generated html report: file:///home/wei/jupyter/Quabo-AutoTest/reports/2dd0d00002318206/reports_sipmsim_J1A.html -----
===== 5 passed in 96.12s (0:01:36) =====
```

e. repeat step *a.* to *d.*, to make sure all of the 8 connectors are all tested.

All of the test will be done in  $3 + 0.5 + 2.5 \times 8 = 23.5\text{min}$ .

## 4 Test Reports

After finishing all of the tests, you will get the test report in the *reports* directory. For each board, you will see a sub directory created, and the directory name is the board's UID.

```
(py312) wei@panoseti:~/jupyter/Quabo-AutoTest/reports$ ll
total 28
drwxrwxr-x 7 wei wei 4096 Mar 31 19:58 ./
drwxrwxr-x 12 wei wei 4096 Mar 31 22:29 ../
drwxrwxr-x 4 wei wei 4096 Mar 31 14:05 1a90e00002318206/
drwxrwxr-x 5 wei wei 4096 Mar 31 17:50 2d98d00002318206/
drwxrwxr-x 5 wei wei 4096 Mar 31 16:04 2dd0d00002318206/
drwxrwxr-x 5 wei wei 4096 Mar 31 12:30 2e30d00002318206/
drwxrwxr-x 5 wei wei 4096 Mar 31 16:59 30b8d00002318206/
```

Figure 3: 5 boards are tested, so we have 5 sub directories.

In each directory, you will see the test reports in *html*, and the log files.



```
(py312) wei@panoseti:~/jupyter/Quabo-AutoTest/reports/2d98d0002318206$ ll
total 380
drwxrwxr-x 5 wei wei 4096 Mar 31 17:50 ./
drwxrwxr-x 7 wei wei 4096 Mar 31 19:58 ../
drwxrwxr-x 2 wei wei 4096 Mar 31 17:22 assets/
drwxrwxr-x 2 wei wei 4096 Mar 31 17:22 quabo/
-rw-rw-r-- 1 wei wei 32532 Mar 31 22:30 reports_quabo.html
-rw-rw-r-- 1 wei wei 4144 Mar 31 22:30 reports_quabo.log
-rw-rw-r-- 1 wei wei 3898 Mar 31 17:26 reports_sipm_J1A.log
-rw-rw-r-- 1 wei wei 3898 Mar 31 17:29 reports_sipm_J1B.log
-rw-rw-r-- 1 wei wei 3898 Mar 31 17:33 reports_sipm_J2A.log
-rw-rw-r-- 1 wei wei 3898 Mar 31 17:38 reports_sipm_J2B.log
-rw-rw-r-- 1 wei wei 3898 Mar 31 17:43 reports_sipm_J3A.log
-rw-rw-r-- 1 wei wei 3898 Mar 31 17:46 reports_sipm_J3B.log
-rw-rw-r-- 1 wei wei 3898 Mar 31 17:49 reports_sipm_J4A.log
-rw-rw-r-- 1 wei wei 3898 Mar 31 17:52 reports_sipm_J4B.log
-rw-rw-r-- 1 wei wei 32971 Mar 31 17:26 reports_sipmsim_J1A.html
-rw-rw-r-- 1 wei wei 32971 Mar 31 17:29 reports_sipmsim_J1B.html
-rw-rw-r-- 1 wei wei 32971 Mar 31 17:33 reports_sipmsim_J2A.html
-rw-rw-r-- 1 wei wei 32971 Mar 31 17:38 reports_sipmsim_J2B.html
-rw-rw-r-- 1 wei wei 32971 Mar 31 17:43 reports_sipmsim_J3A.html
-rw-rw-r-- 1 wei wei 32971 Mar 31 17:46 reports_sipmsim_J3B.html
-rw-rw-r-- 1 wei wei 32971 Mar 31 17:49 reports_sipmsim_J4A.html
-rw-rw-r-- 1 wei wei 32971 Mar 31 17:52 reports_sipmsim_J4B.html
drwxrwxr-x 2 wei wei 4096 Mar 31 17:52 sipmsim/
```

Figure 4: Test reports for one board

If you open the *html* file, and click on the test function, you will see the detailed log information about this test.

## reports\_quabo.html

Report generated on 31-Mar-2025 at 17:22:54 by pytest-html v4.1.1

### Environment

Python	3.12.9
Platform	Linux-5.4.0-204-generic-x86_64-with-glibc2.31
Packages	<ul style="list-style-type: none"> <li>pytest: 8.3.5</li> <li>pluggy: 1.5.0</li> </ul>
Plugins	<ul style="list-style-type: none"> <li>metadata: 3.1.1</li> <li>anyio: 4.9.0</li> <li>html: 4.1.1</li> </ul>

### Summary

4 tests took 00:00:27.

(Un)check the boxes to filter the results.

☒ 0 Failed, ☒ 4 Passed, ☒ 0 Skipped, ☒ 0 Expected failures, ☒ 0 Unexpected passes, ☒ 0 Errors, ☒ 0 Reruns

Result	Test
Passed	test_quabo.py::test_hk_vals
<pre> ----- Captured log call ----- INFO QuaboAutoTest:QuaboAutoTest.py:2194 ----- INFO QuaboAutoTest:QuaboAutoTest.py:2195 Checking HK Values INFO QuaboAutoTest:QuaboAutoTest.py:2196 ----- INFO QuaboAutoTest:QuaboAutoTest.py:2177 Info: tag - expected val(32) is equal to actual val(32) INFO QuaboAutoTest:QuaboAutoTest.py:2160 Info: boardloc - expected val(192.168.3.248) is equal to actual val(192.168.3.248) INFO QuaboAutoTest:QuaboAutoTest.py:2184 Info: hvmn0 - expected val(-55.00)/deviation(2.00) is equal to actual val(-53.29) INFO QuaboAutoTest:QuaboAutoTest.py:2184 Info: hvmn1 - expected val(-55.00)/deviation(2.00) is equal to actual val(-55.07) INFO QuaboAutoTest:QuaboAutoTest.py:2184 Info: hvmn2 - expected val(-55.00)/deviation(2.00) is equal to actual val(-55.11) INFO QuaboAutoTest:QuaboAutoTest.py:2184 Info: hvmn3 - expected val(-55.00)/deviation(2.00) is equal to actual val(-55.10) INFO QuaboAutoTest:QuaboAutoTest.py:2184 Info: rawhvmn - expected val(-70.00)/deviation(1.00) is equal to actual val(-70.36) INFO QuaboAutoTest:QuaboAutoTest.py:2184 Info: v12mon - expected val(1.20)/deviation(0.03) is equal to actual val(1.18) INFO QuaboAutoTest:QuaboAutoTest.py:2184 Info: v18mon - expected val(1.80)/deviation(0.02) is equal to actual val(1.79) INFO QuaboAutoTest:QuaboAutoTest.py:2184 Info: v33mon - expected val(3.30)/deviation(0.10) is equal to actual val(3.28) INFO QuaboAutoTest:QuaboAutoTest.py:2184 Info: v37mon - expected val(3.70)/deviation(0.05) is equal to actual val(3.73) INFO QuaboAutoTest:QuaboAutoTest.py:2184 Info: i10mon - expected val(0.76)/deviation(0.10) is equal to actual val(0.77) </pre>	
Passed	test_quabo.py::test_hk_time
Passed	test_quabo.py::test_maroc_config
Passed	test_quabo.py::test_wr_timing

Figure 5: Get detailed test information from the html file