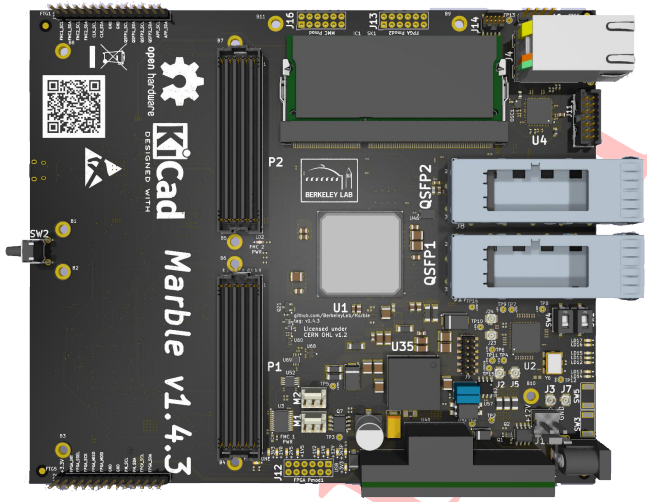
Marble – Factory Acceptance Tests

v1.0 2020

March 25, 2025



Introduction

This document presents a user–friendly guide on how to perform Factory Acceptance Tests (FAT) for the Marble module. Tests begin from a short description of where Marble's useful connector, LED indicators, and test points are placed on the board. It is highly recommended to perform the tests in the order presented in this document.

Contents

1 Overview

> Design files are open source and can be downloaded from GitHub: https://github.com/BerkeleyLab/Marble

> Each board is marked with a revision number. Make sure that the downloaded files correspond to your board.

1.1 Hardware requirements

To perform all tests the following hardware is required:

1. Lab bench-top power supply.
2. Micro USB cable.
3. QSFP loopback module.
4. FMC Tester module.
5. Multimeter.
6. MMC JTAG (example: SEGGER J-LINK mini).
7. FPGA JTAG (example: Digilent JTAG HS3).

1.2 Software requirements

To perform all tests the following software is required:

1. Vivado 19.1.
2. Serial port terminal. Example Miniterm, can be installed by:

```
$ sudo apt-get install python3-serial
```

3. OpenOCD, can be installed by:

```
$ sudo apt-get install openocd
```

2 Power connection

Before connecting the power supply for the first time, check that the main bus is not shorted. Using a multimeter set in resistance measurement mode, measure the resistance between metal pads of the J19 connector (Figure 1).

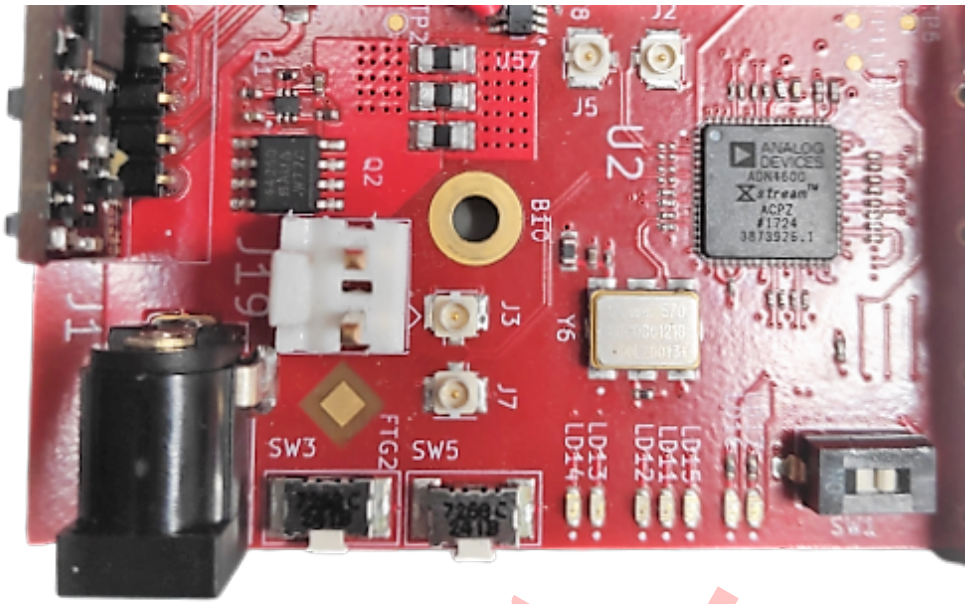


Figure 1: Connector's terminals.

- Measured resistance should be around **200 kOhm**

If the resistance is correct, connect the main power. For this purpose, the current limitation on the power supply should be set to 100mA and the voltage to 12V. **Make sure that the current limit of the laboratory power supply is on.** Now the power cable can be connected to the board and the used laboratory power supply channel can be switched on. This should result in the *3.3V* and *12V* LEDs lighting up as shown on Figure 2. Now it is recommended to go to section Microcontroller programing.

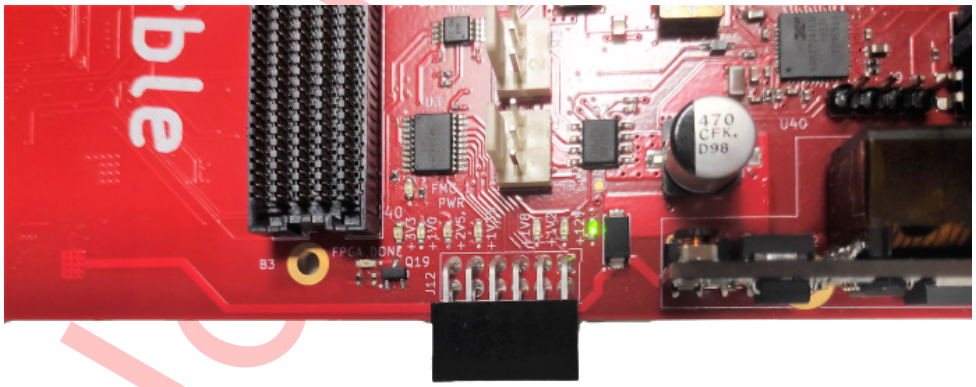


Figure 2: 12V indicator LED is on.

3 Microcontroller programing

Download the latest version of the microcontroller testing code from GitHub:

https://gitlab.lbl.gov/spaiagua/marble_mmc/-/tree/unified_marble

A recent version of OpenOCD (v0.10.0 or later) is required.

1. Connect JTAG module to **J14** as shown on Figure 3.
2. Connect the micro USB cable and using the serial terminal, connect to the last serial port for the new listed device in the operating system. Use 115200 baudrate.
3. Power up the board.
4. Program the microcontroller using the following commands:
   (a) Go to the main folder of the downloaded repository.
   (b) From the command line, run the following:

```
$ make marble_download
```

5. After successful programming, a menu in the serial terminal should appear and LEDs (LD15, LD11, LD12) should blink.

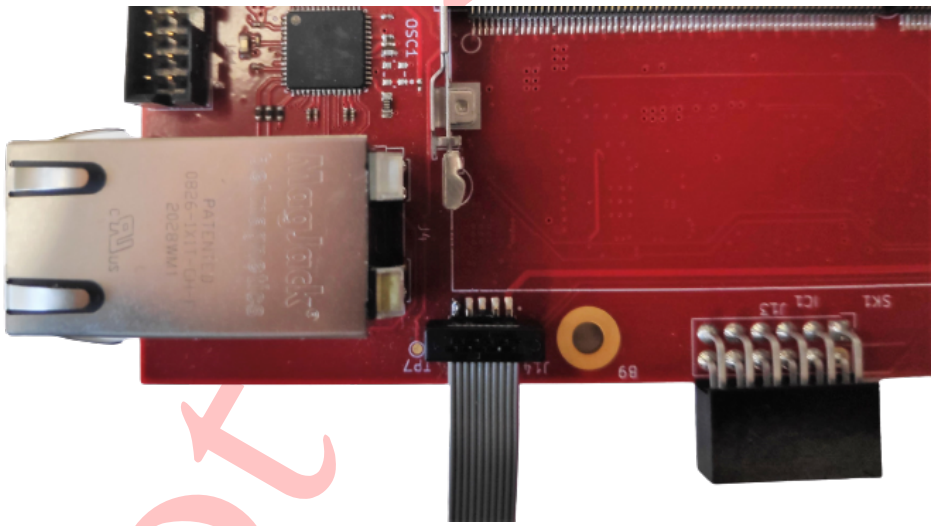


Figure 3: J14 connector.

4 Power Supply programing

Before doing the steps below it is highly recommended to verify that there are no shorts on the power rails. This can be done by measuring the resistance between the test points:

1. **TP12 (GND)** and **TP7 (+2V0)**.

2. **TP12 (GND)** and **TP9 (+1V0)**.
3. **TP12 (GND)** and **TP10 (+2V5)**.
4. **TP12 (GND)** and **TP8 (+1V8)**.
5. **TP12 (GND)** and **TP6 (+1V5)**.
6. **TP12 (GND)** and **TP4 (+1V2)**.
7. **TP12 (GND)** and **TP11 (+3V3)**.
8. **TP12 (GND)** and **TP15 (+3V3USB)**.
9. **TP12 (GND)** and **TP14 (+3V3P)**.
10. **TP12 (GND)** and **TP5 (+1V05)**.

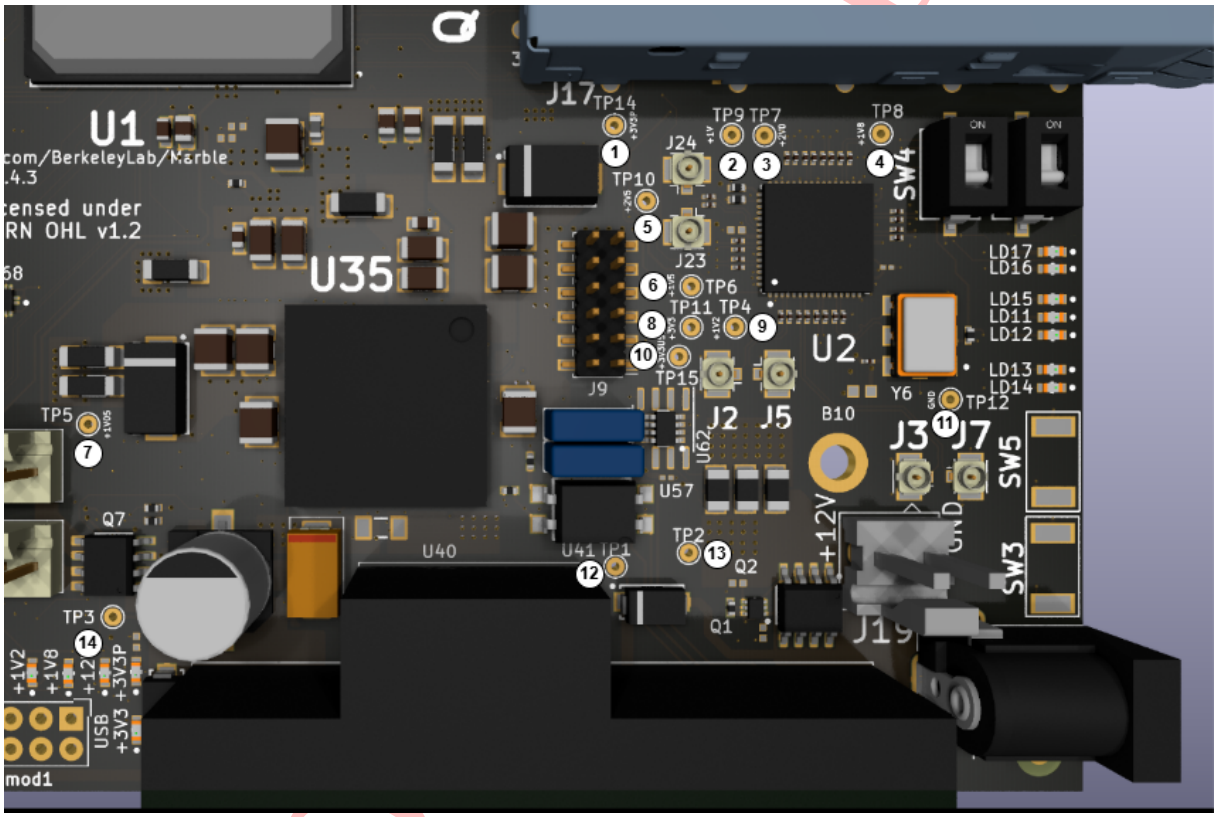


Figure 4: Test points.

Warning: Only perform the following steps if there are no shorts on the power rails.

1. Connect the micro USB cable and, using the serial terminal, connect to the last serial port for the new device listed in the operating system. Use 115200 baudrate.
2. Set up voltage to 12V and the current limit to 1A on a lab power supply.
3. Power up the board.

4. From the menu displayed in the serial terminal, select the option `g) XRP7724 go`.
5. Power cycle by turning the lab power supply Off and then On.
6. All power LED indicators should be On (Figure 5).
7. Using multimeter measure the voltage between the test points:
   (a) **TP12 (GND)** and **TP7 (+2V0)** - expected voltage: +2.0V.
   (b) **TP12 (GND)** and **TP9 (+1V0)** - expected voltage: +1.0V.
   (c) **TP12 (GND)** and **TP10 (+2V5)** - expected voltage: +2.5V.
   (d) **TP12 (GND)** and **TP8 (+1V8)** - expected voltage: +1.8V.
   (e) **TP12 (GND)** and **TP6 (+1V5)** - expected voltage: +1.5V.
   (f) **TP12 (GND)** and **TP4 (+1V2)** - expected voltage: +1.2V.
   (g) **TP12 (GND)** and **TP11 (+3V3)** - expected voltage: +3.3V.
   (h) **TP12 (GND)** and **TP15 (+3V3USB)** - expected voltage: +3.3V.
   (i) **TP12 (GND)** and **TP14 (+3V3P)** - expected voltage: +3.3V.
   (j) **TP12 (GND)** and **TP5 (+1V05)** - expected voltage: +1.05V.

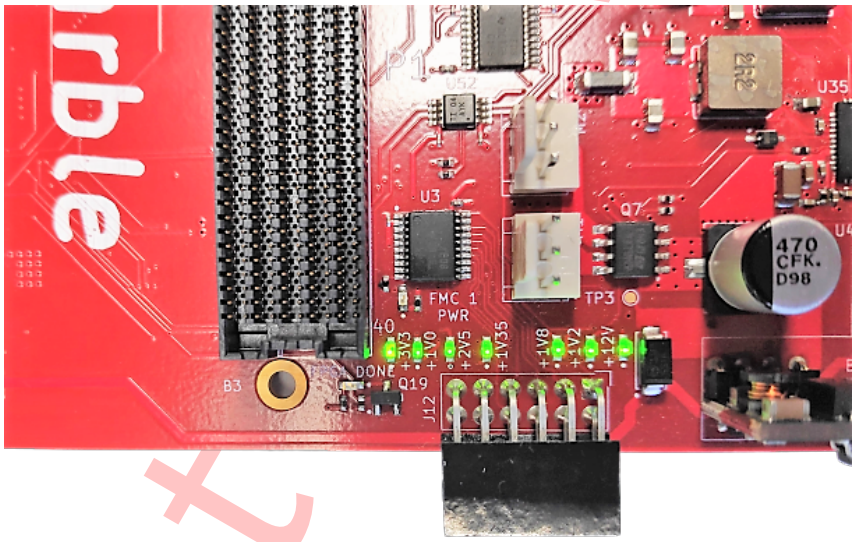


Figure 5: Power LED indicators after a successful power cycle.

5 FPGA programing

Download the latest version of the FPGA testing code from GitHub:

https://github.com/BerkeleyLab/Bedrock/

Before testing the FPGA it is recommended to set up the current limit to 2A on the lab power supply.

5.1 FMC test

Board power should be turned off when inserting and removing the FMC module.

1. Plug FMC Tester module to one of the FMC connectors as shown on Figure **??**.
2. Connect the micro USB cable and, using the serial terminal, connect to the last serial port for the new device listed in the operating system. Use 115200 baudrate.
3. Change the network adapter settings to connect with static IP address **192.168.9.10**.
4. Connect Marble to the computer using an Ethernet cable.
5. Power up the board.
6. In the serial terminal menu choose 4) GPIO control > a) FMC power to turn on power to the FMCs.
7. Program the FPGA using the following steps:
   (a) Go to the folder **Bedrock/projects/test_marble_family/**
   (b) Open command terminal and run command:

```
$ mutil usb
```

8. To run a test do the following steps:
   (a) Go to the folder **Bedrock/projects/test_marble_family/**
   (b) To test P1 FMC connector run the following command:

```
$ PYTHONPATH=../../badger:../../peripheral_drivers
/i2cbridge python fmc_test_c.py --ip $IP --fmc 0
```

   (c) To test P2 FMC connector run the following command:

```
$ PYTHONPATH=../../badger:../../peripheral_drivers
/i2cbridge python fmc_test_c.py --ip $IP --fmc 1
```

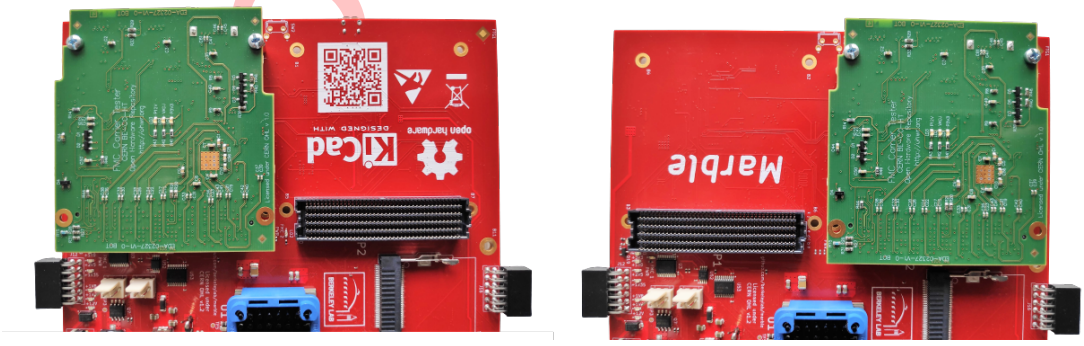
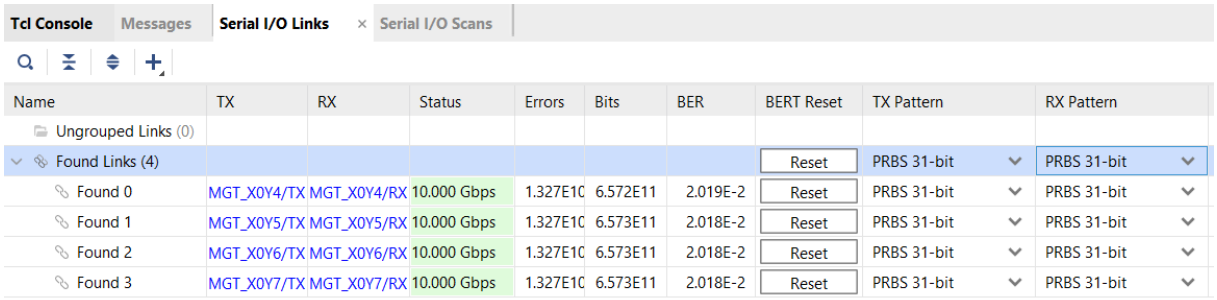


Figure 6: FMC connectors FMC Tester board plugged in.

5.2 QSFP test

Board power should be turned off when inserting and removing the QSFP module.

1. Plug QSFP loopback module to one of the ports.
2. Connect FPGA JTAG programmer.
3. Configure FPGA using `marble_ibert.bit` bit file.
   (a) Run Vivado
   (b) Go to Flow » Open Hardware Manager and then Tools » Auto Connect
   (c) Click Tools » Program Device » xc7k160t_0 to open the programming window.
   (d) Choose the *bitstream file* and click Program
4. After successful programming, the Dashboard should start automatically.
5. Detect links by clicking Serial I/O Links » Auto-detect links
6. Correctly detected and working links should appear as shown in figure 7.
7. Connect the QSFP loopback module to the other QSFP connector and repeat the steps above.



Name	TX	RX	Status	Errors	Bits	BER	BERT Reset	TX Pattern	RX Pattern
Ungrouped Links (0)									
Found Links (4)							Reset	PRBS 31-bit	PRBS 31-bit
Found 0	MGT_X0Y4/TX	MGT_X0Y4/RX	10.000 Gbps	1.327E10	6.572E11	2.019E-2	Reset	PRBS 31-bit	PRBS 31-bit
Found 1	MGT_X0Y5/TX	MGT_X0Y5/RX	10.000 Gbps	1.327E10	6.573E11	2.018E-2	Reset	PRBS 31-bit	PRBS 31-bit
Found 2	MGT_X0Y6/TX	MGT_X0Y6/RX	10.000 Gbps	1.327E10	6.573E11	2.018E-2	Reset	PRBS 31-bit	PRBS 31-bit
Found 3	MGT_X0Y7/TX	MGT_X0Y7/RX	10.000 Gbps	1.327E10	6.573E11	2.018E-2	Reset	PRBS 31-bit	PRBS 31-bit

Figure 7: Working links.

5.3 Ethernet test

The following steps should be performed to test the Ethernet:

1. Program the FPGA using the following steps:
   (a) Go to the folder **Bedrock/projects/test_marble_family/**
   (b) Open command terminal and run command:

```
$ mutil usb
```

2. Change the network adapter settings to connect with static IP address **192.168.9.10**.
3. Connect Marble to the computer using an Ethernet cable.

4. Ping the board:

```
$ ping 192.168.9.10
```

5. Expected result:

```
Pinging [192.168.9.10] with 32 bytes of data:
Reply from 192.168.9.10: bytes=32 time=17ms TTL=55
Reply from 192.168.9.10: bytes=32 time=152ms TTL=55
Reply from 192.168.9.10: bytes=32 time=12ms TTL=55
Reply from 192.168.9.10: bytes=32 time=14ms TTL=55

Ping statistics for 192.168.9.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 12ms, Maximum = 152ms, Average = 48ms
```

6. For data traffic test use *iperf*. Run command:

```
$ iperf -c 192.168.9.10 -u -b 1000M -p 802
```

6 Appendix

6.1 An alternative way to program FPGA

Programming FPGA using Vivado and Digilent JTAG HS3:

1. Run Vivado
2. Go to Flow > Open Hardware Manager and then Tools > Auto Connect
3. Click Tools > Program Device > xc7k160t_0 to open the programming window.
4. Choose the *bitstream file* and click Program
5. After successful programming, the Dashboard should start automatically.