# IGH SHIELD HARDWARE BRINGUP

## Boron Pinout

The following pin configurations are assumed for the shield bringup test.

| Pin Name | Pin Num |
|---|---|
| RFM69_CS | A2 |
| RFM69_G0 | D8 |
| RFM69_RST | D6 |
| RFM69_EN | N/C |
| SD_CARD_CS | A5 |
| VALVE_CTRL | D7 |
| IRR_BUTTON | A4 |
| SD_MISO | D11 |
| SD_MOSI | D12 |
| SD_SCK | D13 |
| RFM69_MISO | D4 |
| RFM69_MOSI | D3 |
| RFM69_SCK | D2 |

The bringup firmware considers all the peripherals and resources of the device under test and lets the development/engineering team test the hardware so that production may be done parallel to the firmware development.

## Programming the Shield

1. prepare a computer running Windows
2. Install the lates version of Particle CLI for windows by following the instruction at this link
3. Connect a fully built Shield to the computer over USB
4. Press both the `RESET` and `MODE` buttons on the Boron, release only the `RESET` button then and wait for the Boron to start blinking yellow after it blinks magenta then release the `MODE` button immediately the device starts blinking yellow
5. Open a `CMD terminal/PowerShell` at from the folder containing `igh_shield_bringup.bin`
6. Enter the following command `particle flash --usb igh_shield_bringup.bin` then press enter
7. After successfully programming the device, the terminal will display a `Flash success!` message and will reset the board automatically.

## Running Shield BringUp

8. Install CoolTerm, a terminal software from the following link to the production computer.
9. Open CoolTerm
10. Click on `Options` on the top menu bar and select the `COM` port number for the Boron
11. Set the Baudrate to `115200`
12. Click `OK`
13. Click on `Connect` to start a bringup session
14. Enter `?` into the terminal and wait for a resposnes with all possible tests.
15. To run a specific test, enter into the terminal the index value, you will not need to press enter. The device will start the test for the specified number entered. The test available include:
    \`

- ?. Print Options - 1. Get Device ID - 2. Test Device API - 3. Test EEPROM - 4. Test Irrigation Button - 5. Test Valve Control - 6. Test SD Card - 7. Test Radio \`

### 4. Test Irrigation Button

When testing the irrigation button, the device will give you instructions on how long to press the button. For the test to pass, you must hold the button for at least the number of seconds required.

### 5. Test Valve Control

In order to test the valve, the device will toggle the valve to the open state, wait for the valve to fully open, the closes the valve and waits for it to full close before passing or failing the test.

### 6. Test SD Card

In order to test the SD card, install an SD card into the device. It is assumed that once the SD card is installed, it will never be removed for the device's lifetime. Make sure the SD card is formated to `Fat16` or `Fat32` using a windows machine before installing it into the device. During this test, a file is written to the sd card then deleted. if this sequence is successful, then the result is true

### 7. Test Radio

To test the Radio, you will need a second device `(Feather M0)` with a radio running the `igh_spear_radio_test_hub.bin` . The second radio may be used for all the devices under test and does not need any further interruction other than keeping it on. You may, however, choose to connect the second device to c terminal app to read serial data. In this case, the device will always print out the radio ID of the device under test and confirm to you that the device is infact sending radio data or not. You may also use this to do a range test if necessary. For this test, the device under test sends a message to the radio device and expects a response back. The test passes if a valid response is sent back to the device under test.

## *Notes*

- Using more advanced terminals like DockLight in place of CoolTerm, it is possible to automate the test and generate test results for quality control purposes of the devices.
- A similar solution shall be presented for the Spear bringup
- The System will only successfully process one test at a time. Only send tests after the previous test is completed.

# IGH SPEAR HARDWARE BRINGUP AND RADIO TEST HUB

## Feather M0 Pinout

The following pin configurations are assumed for the Spear bringup test.

| Pin Name | Pin Num |
|---|---|
| VBAT_SENSE | A7 |
| SOIL_MOISTURE_SENSOR | A2 |
| SHT10_DATA | 10 |
| SHT10_CLOCK | 12 |
| DHT_DATA | 5 |
| MHZ19_RX | TX1 |
| MHZ19_TX | RX0 |
| RESERVED | 8 |
| RESERVED | 3 |
| RESERVED | 3 |
| RESERVED | 4 |
| RESERVED | 13 |

The spear bringup firmware may be used to test all peripherals and sensors using the drivers written as well as bare metal application code. The Feather M0 used for the spear has two different binaries that can be programmed to each.
The `igh_spear_bringup.bin` is the firmware to test all sensors and periphersals while `igh_spear_radio_test_hub.bin` converts the spear into a radio testing hub that enables validation of the Shield radio test routine.

## Programming the Spear

To simplify the devlopment process, we will be using the Arduino CLI feature to easily program the spears in bulk as they are already compatible with it. These instructions are only valid for Windows Computers

1. Install `Arduino 1.8.12` fro this link
2. Follow the instructions from the Arduino IDE setup page instructions to add the `package_adafruit_index.json` support to the IDE
3. Follow the instruction from Using With Arduino IDE page to add Feather M0 Radio support to the IDE.
4. Download `igh_spear_bringup.bin` or `igh_spear_radio_test_hub.bin` and save it in an easy to reach path, e.g. `Desktop/`

5. Connect each feather you would like to program ***one at a time*** to a computer via USB.
6. Double tap the `reset button` on the `Feather M0` and make sure the `red LED` next to the USB Micro port is `pulsing with a faded glow`.
7. Look in `Device Manager` to find the COM port at which the Feather M0 is attached. For isntance `COM4`.
8. Open windows Command Line of PowerShell in the directory with the downloaded binaries.
9. Use the following commands to program the device with the respective application

### *Program Feather as Radio Test Hub*

```
C:\Users\$name$\AppData\Local\Arduino15\packages\arduino\tools\bossac\1.7.0-arduino3/bossac.exe -i -d --port=COM8 -U true -i -e -w -v igh_spear_radio_test_hub.bin -R
```

### *Program Feather Spear for Bringup*

```
C:\Users\$name$\AppData\Local\Arduino15\packages\arduino\tools\bossac\1.7.0-arduino3/bossac.exe -i -d --port=COM8 -U true -i -e -w -v igh_spear_bringup.bin -R
```

**NOTE:** After programming the board, the COM Port number will change.

## Running The Spear Bringup Firmware

Due to the lack of non volatile addressable memory of the MCU on the Feather M0, it therefore adviced that the Bringup Test for each device is done as few times as possible as each test consumes a flash write.

The Spear Bringup Firmware will auto test all the peripherals and print a report out to a terminal through its COM port. The device will wait untill it is connected to a terminal software like CoolTerm or Putty before testing begins.

Testing for the `NPK sensor` is not included in this version of the bringup. This is due to the unavailability of the hardware at the time

A successfull test should look as follows with failed tests displaying `ERROR` instead of `OK`

```
   *** TEST IGH SPEAR ***
******************************
BATTERY................4361mV
SYSTEM.................OK
RFM69..................OK
SETTINGS...............OK
LUX METER..............OK
SOIL MOISTURE SENSOR...OK
SHT10..................OK
DHT22..................OK
MHZ19..................OK
```