Testing procedure - Yardbird-0.4.0

Hardware needed to run the testing procedure:

- Yardbird 0.4.0 Board
- External Antenna
- 3.3V Power Supply
- Cable connection
    i. JST Connector Header SMD R/A 2POS 1mm
    ii. Jumper wire
- Launchpad CC1352R and ISP Programming Cable
- Computer or Raspberry Pi

Functionalities to be tested:

1. Flashing the Yardbird
2. Buzzer beep
3. Resetting process through push button
4. Low power mode current/Deep sleep MCU
5. Debug LEDs
6. Sub1GHz communication
7. Over the Air Download programming
8. Test Seismic Gain, temperature and supply voltage
9. Geophone Signal behavior
10. Conditioning Circuit transfer function
11. Basic mechanical strength/robustness

Setting up the Ground Sensor Testing Environment

In order to perform the tests described below, it is first necessary to prepare the physical environment. This is composed by two main devices:

- The Ground Sensor, which we will call the node device;
- The network CoProcessor, which is a functionality of the Homebase in the Perimeter System, implemented for tests using a development board and the computer/RPi.

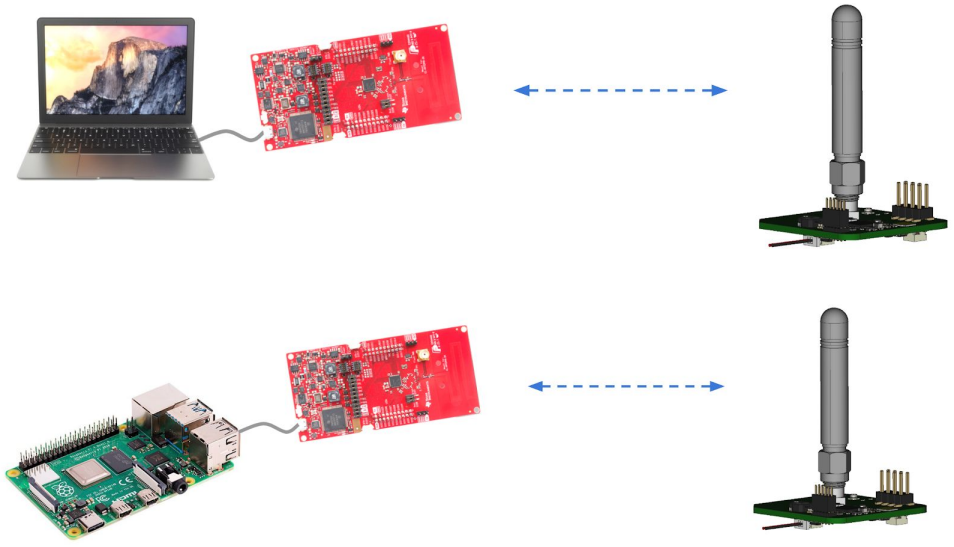
After setting up the peripherals through UART communication, the computer will run test scripts controlling the Launchpad and exhibiting the results in order to verify the proper functioning of the Ground Sensor features.

1) Hardware Connections

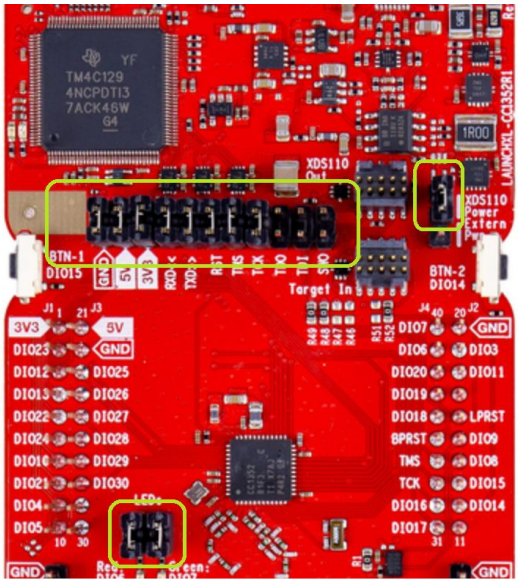
The test procedure is based on a sequence of commands and the user can run and evaluate the output and can be run on a regular computer or even a Raspberry Pi.

The following hardware connection should be done:

- Connect the LAUNCHXL-CC1352R in a USB Port (from computer or RPi).



In order to work as the CoProcessor board, the LAUNCHXL-CC1352R needs to be connected as marked below (in yellow). Make sure that the XDS110 and LEDs jumpers are connected, and the XDS110 Power is selected on the header on the right.



- The Ground Sensor PCB is connected to a 3V3 source (CR123A battery/USB Dongle/ other) through the connectors J1 (BAT_PWR) or J6 (3V3 and GND).
- The geophone (seismic sensor) connects to the connector J4 (GEOPH)

2) Command line interface

Follow the installation instructions:

1. Install Rust;
2. Clone *hardware-abstraction-layer* repository on GitHub:

```
git clone https://github.com/perimeter-inc/hardware-abstraction-layer
```

3. Clone the tools repository on GitHub:

```
git clone https://github.com/perimeter-inc/tools.git
```

4. Switch to the test procedure branch:

```
git checkout test-procedure-ground-sensor
```

5. Inside the `test_procedure_ground_sensor/` folder can be found a series of folders containing the scripts for the specific tests.
6. Run the scripts according to each description from the following sections. Please consider using the Yardbird Testing Routine Table in order to record the progress. In case of questions, feel free to reach out our team.

Functionalities and Tests Description

1) Flashing Process

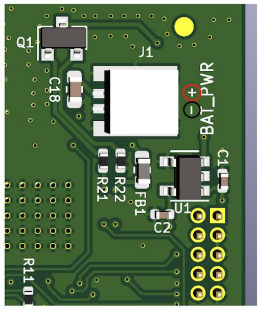
Hardware under test:

- Connections between the JTAG and the MCU
- MCU CC1352R

Testing procedure:

1.a) Power-up the board by connecting 3V3 (red) and GND (black) wires. Use **one** of the options below

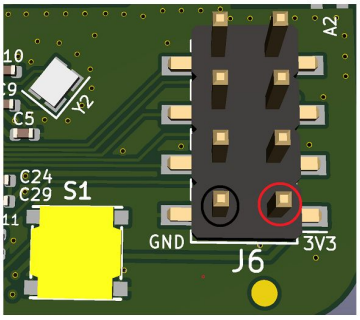
- Option A



+

JST Connector Header
SMD R/A 2POS 1mm

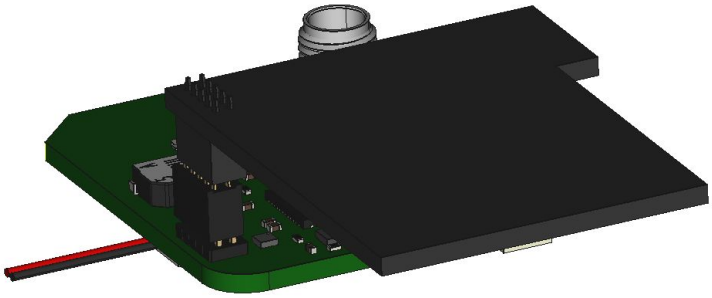
- Option B



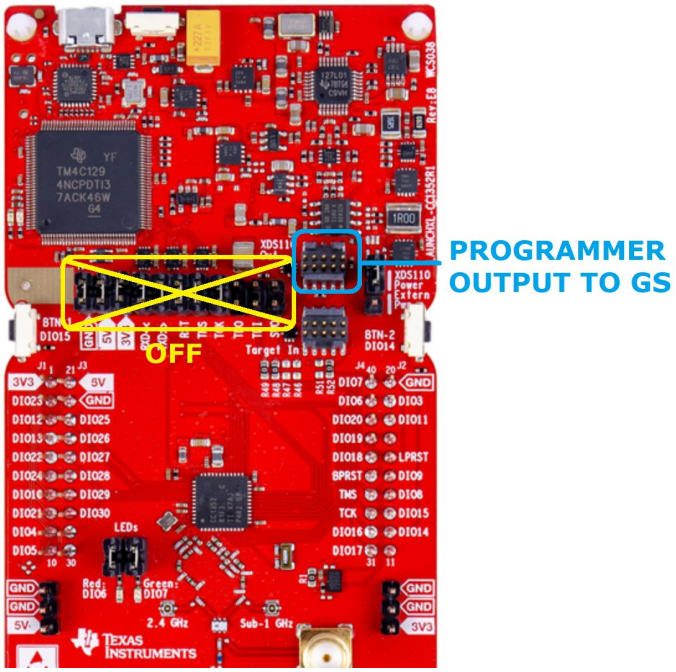
+

Jumper wire

1.b) Connect the programmer device to the pcb JTAG connector (J5). The programmer device can be a separate programming board, as in the example below (Debugger DevPack).



Another option is to use the TI LAUNCHXL-CC1352R programmer. In such case, the jumpers connecting the upper part (debugger) to the lower part (microcontroller) must be removed, and the ISP programming cable is connected to the programmer output connector (in blue).

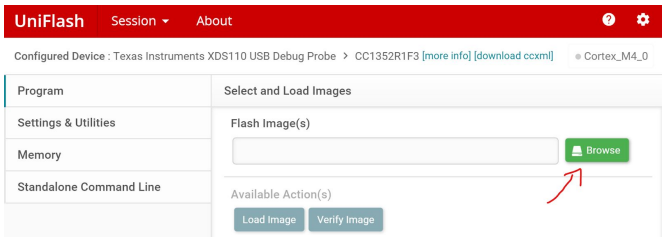


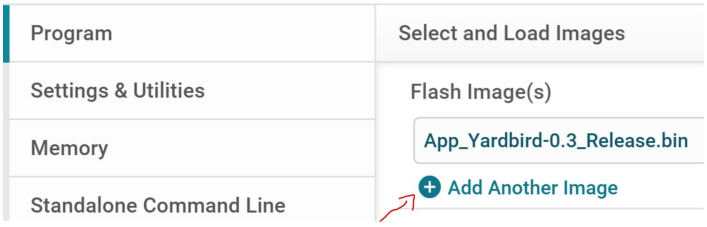
1.c) Open Uniflash Software and connect the programmer device to one of the computer USB Ports.

- If the software succeeds in automatically recognizing the programmer device, its name and serial number will be shown in the *Detected Devices* tab - just click *Start* to open a *Session*.

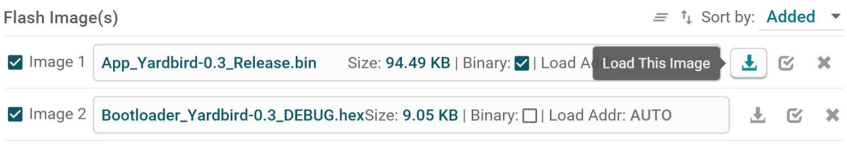
- If autodetection doesn't succeed, select the microcontroller CC1352R (the MCU used on the Yardbird PCB) in the *New Configuration* tab. Then, click *Start* to open a *Session.*

1.d) In the new *Session*, go to Program -> Select and Load Images, browse for the Application file and open it. Then, click on *Add Another Image,* browse for the Bootloader file and open it.





1.e) First, flash the no_channel_application binary by clicking on *Load This Image*. Then flash the bootloader file the same way (the files below are being used as examples).



Test evaluation:

If no error shows up when loading the Application and Bootloader images, the Flashing Process was done successfully.

2) Buzzer

Hardware under test:

- Buzzer BZ1
- Transistor Q1
- Resistors and connections between those parts

Testing procedure:

The buzzer must chime when the Application and Bootloader images are correctly loaded onto the MCU. If the board has already been programmed, you can reset the microcontroller by pressing and releasing the push button *S1*, and the buzzer should chime subsequently.

Test evaluation:

If you can easily hear the buzzer's *beep*, then the test is successful.
Otherwise, this may need further investigation:

- the application or the bootloader may not have been programmed correctly
- the buzzer and/or its circuitry may be defective.

3) Resets when pressing push button

Hardware under test:

- Push button S1 and its circuitry

Testing procedure:

- Approach 1: press and release the push button *S1*. The buzzer should chime, meaning that the reset operation was performed successfully (this test needs the buzzer BZ1 to be working correctly);
- Approach 2 (to be used if BZ1 is not working): with the Ground Sensor connected to the HAL, press and release the push button *S1*. HAL log messages called *Association Request* should start coming from this sensor - if this happens, the reset operation was performed successfully (this test needs the Ground Sensor RF circuitry to be working correctly).

Test evaluation:

Described in the *Testing Procedure* session.

4) Low power mode current

Hardware under test:

- 32.678 kHz Crystal and loading capacitors
- Stand by capability of the CC1352 MCU

Testing procedure:

The Ground Sensor needs to be running the *no_channel_application* firmware.

Instructions on how to measure the current:
1 - Connect a jumper wire to the +3.3Vdc of a power supply. Then, connect the other side (needs to be female type) to the 3v3 pin of connector J6.

2- Connect a jumper wire on the GND pin of connector J6 (needs to be female type). Then, connect the other side of the wire to a breadboard hole, letting some metal surface exposed.

3- Connect a jumper wire to the GND of the power supply. Then, connect the other side of the wire (needs to be male type) to another hole of the breadboard.

4- Measure the current between the two connections in the breadboard. As soon as you place the two probes and close the circuit, the buzzer should *beep* indicating that the application has re-started.

Test evaluation:

Current values between 70~200uA (direct current) indicate that the Low Power Mode is working correctly.

Based on past experiences, current values higher than 1mA usually indicate that the Ground Sensor is not entering in Low Power Mode. The 32kHz crystal and its loading capacitors must be investigated.

After the end of this test, please flash the Ground Sensor with the *application_v0_debug* binary. The bootloader needs to be re-loaded right after that.

5) Debug LEDs

Hardware under test:

- Red and green debug LEDs (D1 and D2, respectively)
- LEDs circuitry

Testing procedure:

With the Ground Sensor running the firmware *application_v0_debug*, one can check if the LEDs are working properly simply by looking at them and watching them blink.

When are they supposed to blink?

- The green LED toggles every time a RF packet is sent.
- The red LED is ON when the microcontroller is sleeping, and OFF otherwise. To turn ON the red LED, press the reset button S1.

Test evaluation:

In the following steps, *RF Communication* and *OAD* tests will be performed. When the Ground Sensor connects to the Homebase, the red LED must turn OFF and the green one must start blinking.

6) RF communication

Hardware under test:

- 48MHz crystal and RF circuitry

Testing procedure:

The Homebase needs to be ready according to the < Setting Up a Homebase > section.

The Ground Sensor should be positioned ~ 20cm far from the Homebase communication board, and both must be on the same horizontal plane.

Inside the folder rf_communication_test:
1 - run the command **cargo build --release** and wait until the end of the compilation process
2 - run the command **bash rf_test.sh**

The system will start configuring the coprocessor board. When this process finishes, it will wait until receiving a *connection request* from the sensor. This might take some seconds.

As soon as the necessary parameters are configured, the RSSI information will be logged on the screen, at the rate of one value per second.

Test evaluation:

1 - New RSSI values should be constantly logging on the screen, meaning that the RF communication is working. A message saying "**Test finished**" shows up in the next minute, the **Evaluation Point 1 is achieved.**

2- In the end of the test, the ***Average RSSI*** will be printed on the screen. If this value is **between -5 and -50** (dBm), the **Evaluation Point 2 is achieved and we assume RF communication is working properly.**

- ➔ This value can vary mainly due to destructive interferences. If *Average RSSI* is below -50 (for instance -55, -60, -70), move the Ground Sensor some centimeters and repeat the test, running the command **bash rf_test.sh**. If the RSSI level keeps lower than expected, there might be a problem in the Ground Sensor RF circuitry (contact myPerimeter staff).

7) OAD

Hardware under test:

- External flash and its circuitry
- Bootloader correctly flashed

Testing procedure:

The procedure aims to change the Ground Sensor firmware from *Application_v0_debug* to *Application_v1.* To perform this test, one needs to ensure that the firmware being sent over the air is different from the current Ground Sensor firmware.

Start the OAD tool by running
`cargo run -- -y 00:11:22:33:44:55:66:77 -f application_v1.bin`,
where `00:11:22:33:44:55:66:77` is the Ground Sensor MAC address.

The tool will wait for a *Connection Request* from the sensor, and then start transferring the new firmware blocks. At the end, it will check for failed updates.

Test evaluation:

- In the case of a successful update, a massage will show up saying “Ground Sensors were successfully updated”. OAD was tested and is working correctly

- In the case of a failed update, a Warning message will state that the firmware version and commit hash of the Ground Sensor didn’t change. There may be a problem with the firmware being downloaded or with the circuit related to the flash memory on the pcb.

8) Seismic Gain

Hardware under test:

- U6 digital potentiometer

Testing procedure:

Go to the *gain_test/* folder and run the command
cargo test -- --nocapture test_write_read_seismic_gain

Wait a little bit until the experiment finishes.

Test evaluation:

The test results will appear on the screen whether as

[Work in progress]

9) Geophone signal acceptable

Capture idle-impact-idle states and transitions between them, for:

- default position
- upside down