STELLA-1.2 Programming Instructions

Overview

These programming instructions follow the exact testing sequence performed during the STELLA-1.2 build process. Each test is performed in order to verify proper operation before proceeding to the next step.

Required Software and Tools

Software Requirements

- **Mu Editor** (Python code editor)

Hardware Requirements

- USB-C cable
- Computer with USB port

Initial Setup

1. **Extract Files**: Unzip the STELLA-1.2 file to access its contents.
2. **Enter Bootloader Mode**:
    - Hold down the **BOOT** button on the microprocessor
    - Connect the USB-C data cable from your computer to the microprocessor
    - A drive labeled **FEATHERBOOT** should appear in your file explorer
    - **If the drive doesn't appear**: Press the **RESET** button twice quickly, then press the **BOOT** button
3. **Install Firmware**:
    - Navigate to the **UF2** folder
    - Drag and drop the **.uf2** file onto the **FEATHERBOOT** drive
    - Wait for the copy to complete
    - The **FEATHERBOOT** drive will automatically eject and be replaced by **CIRCUITPY**
4. **Install Software**:
    - Open the **STELLA-1.2-code-and-libraries** folder
    - Drag and drop the following items onto the **CIRCUITPY** drive:
        - code.py file
        - setting.toml file

- configuration_files folder
- lib folder
- software_modules folder

Test A. Testing the Microcontroller

Prerequisites:

- Microcontroller connected via USB-C data cable
- Mu Editor installed and open

Test 1: Unique Identifier

1. In Mu Editor, click **Serial** at the top
2. Navigate to: S12_test_codes → main unit test codes → A0._microcontroller_unique_identifier
3. Drag and drop code.py into the **CIRCUITPY** drive
4. Check the serial output in Mu Editor for the Unique Identifier

Test 2: Blink Indicator

1. Navigate to: S12_test_codes → main unit test codes → A1._microcontroller_blink_indicator
2. Drag and drop code.py into the **CIRCUITPY** drive
3. Verify the microprocessor LED is blinking

Note: Each time you drag a new code.py file to the CIRCUITPY drive, it will replace the previous version and automatically restart the microcontroller.

Test B. Testing the Real Time Clock

Hardware Setup

1. Connect the RTC clock module to the microprocessor using the Qwiic cable
2. Connect the microprocessor to your computer via USB-C data cable

Testing and Configuration

I2C Bus Scan

1. Open **Serial** in Mu Editor
2. Navigate to folder: B0._i2c_bus_scan
3. Drag and drop code.py onto the **CIRCUITPY** drive
4. Check the serial output in Mu Editor for the I2C device response

Real-Time Clock Setup

1. Navigate to folder: B1._real_time_clock_and_clock_set
2. Drag and drop code.py onto the **CIRCUITPY** drive
3. Go to **time.is/utc** to get the current UTC time
4. **Add approximately 5 seconds** to the time to allow buffer for entering all prompts
5. In Mu Editor's serial window, enter the adjusted date and time in **UTC format** when prompted

Test C. Rotary Encoder

1. Open **Serial** in Mu Editor
2. Navigate to folder: C._rotary_encoder
3. Drag and drop code.py onto the **CIRCUITPY** drive
4. Twist and push the rotary encoder and observe the results in the serial output

Test D. Display Testing

Hardware Setup: Connect the screen to the EYESPI module using the cable

TFT Display

1. Navigate to folder: D._2.8in_tft_display
2. Drag and drop code.py onto the **CIRCUITPY** drive
3. Check the display screen for visual output

Test E. Capacitive Touch Screen

1. Open **Serial** in Mu Editor
2. Navigate to folder: E._capacitive_touch_screen
3. Drag and drop code.py onto the **CIRCUITPY** drive
4. Tap on the display with your finger and observe touch coordinates in the serial output

---

Note: Each test replaces the previous code.py file on the CIRCUITPY drive and automatically restarts the microcontroller.

Test F. 5V Boost Converter

1. Open **Serial** in Mu Editor
2. Navigate to folder: F0._5V_boost
3. Drag and drop code.py onto the **CIRCUITPY** drive
4. Check the serial output for results

Analog Input

1. Open **Serial** in Mu Editor
2. Navigate to folder: F1._analog_in
3. Drag and drop code.py onto the **CIRCUITPY** drive
4. Check the serial output for results

Test B. RTC Clock Verification (Post-Soldering)

After soldering the clock module:

1. Repeat **Test 1** (I2C Bus Scan) and **Test 2** (Real-Time Clock Setup) from the RTC Clock Module Setup section to verify proper installation.

Note: If the battery was installed and you already set the clock you should not need to reset the time. Just verify that it is working.

Test G.SD Card

1. Open **Serial** in Mu Editor
2. Navigate to folder: G._sdcard
3. Drag and drop code.py onto the **CIRCUITPY** drive
4. Check the serial output for results

Test H. GPS Module

1. Open **Serial** in Mu Editor
2. Navigate to folder: H._GPS
3. Drag and drop code.py onto the **CIRCUITPY** drive
4. Check the serial output for results

Final Setup: Install Main STELLA Code

1. Navigate back to the **STELLA-1.2-code-and-libraries** folder
2. Drag and drop code.py onto the **CIRCUITPY** drive
3. Coding installation is now complete

Final Hardware Test: Power Button

1. Test the power button by cycling the power on and off
2. Verify the button is working properly

Congratulations! You now have a fully functional STELLA-1.2 Base module.