



# STARLIGHT MINI

User Manual

V1.0

# Contents

## 1 Basic Information

### 1.1 Specifications, Electrical Characteristics

## 2 Features

### 2.1 Board Features

### 2.2 Schematic & Labeled Board

## 3 Usage

### 3.1 MissionControl

### 3.2 Custom Firmware

### 3.3 Powering SL MINI

#### 3.3.1 **Important Information**

## 4 How does STARLIGHT MINI work?

### 4.1 MissionControl Firmware

#### 4.1.1 Altitude Determination

#### 4.1.2 Launch Detection

#### 4.1.3 Apogee Detection

#### 4.1.4 Motor Burnout Detection

#### 4.1.5 x Feet Reached Detection

#### 4.1.6 Landing Detection

## 5 MissionControl

### 5.1 Overview of Features

### 5.2 How does MissionControl interface with our flight computers?

### 5.3 Flight Mode vs Idle/Programming Mode

### 5.4 How to use MissionControl

### 5.5 Troubleshooting

# 1 Basic Information

STARLIGHT MINI is a flight computer designed for small-to-medium scale model rockets. The board has an incredibly small footprint while still packing tons of features and useful sensors.

## 1.1 Specifications/Electrical Characters

Descriptor	Value
Dimensions (L*W*H) (mm)	<b>34 x 20 x 1.6</b>
Weight (g)	<b>10</b>
Input Voltage ( $V_{in}$ ) (V)	<b>2.8-6 (1s LiPO)</b>
Logic Voltage ( $V_{logic}$ ) (V)	<b>3.3</b>
Pyro Channel Current Draw ( $I_{pyro}$ ) (A)	<b>2</b>
Microcontroller Speed (MHz)	<b>133</b>

The board is typically shipped in a purple color. Older or newer versions may be shipped in a green silkscreen color, but the electrical characteristics and layout are exactly the same.

# 2 Features

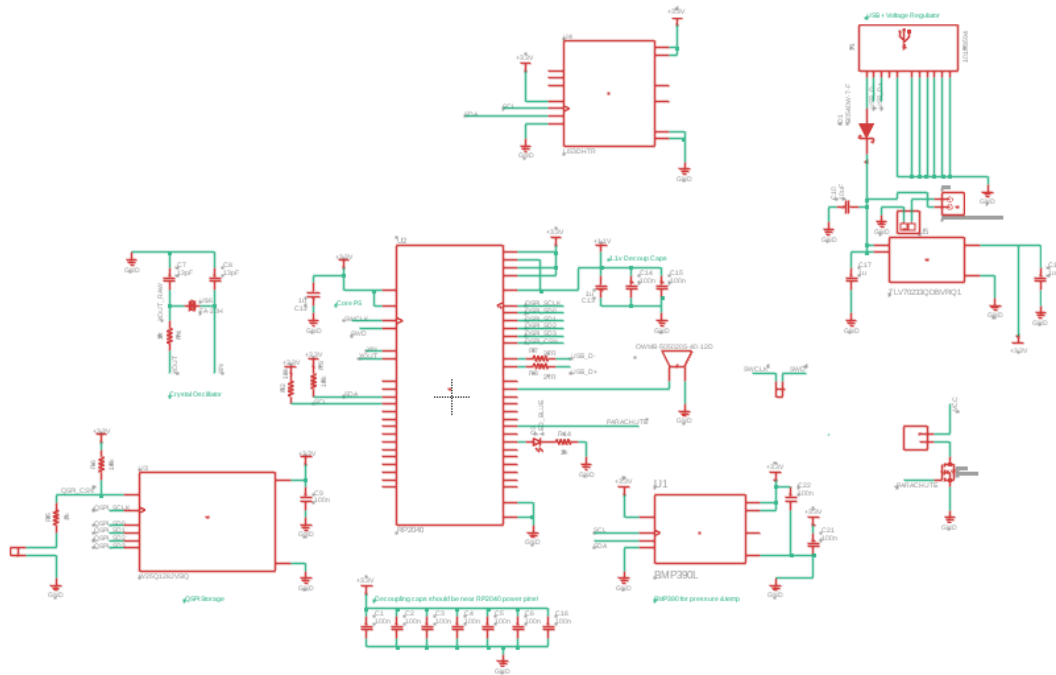
STARLIGHT MINI was designed as a cheaper alternative to more expensive MissionControl-powered boards, with a much smaller footprint.

## 2.1 Board Features

- **RP2040 Dual-Core MCU**
- The RP2040 microcontroller is a very capable chip with 128kB of RAM and can be programmed in MicroPython, as well as C.
- [More information on the RP2040 can be found here.](#)
  
- **BMP388 Pressure & Temperature Sensor**
- The BMP388 is a temperature and pressure sensor.
- The pressure readings are used for the firmware's altitude determination as well as apogee detection.
  
- **LIS3DHTR Accelerometer**
- The LIS3DHTR accelerometer can measure +/-16g of acceleration.
- Rockets that fly with more acceleration than this are still supported, but accurate readings past +/-16g will not be possible.
  
- **16MB of flash storage**
- STARLIGHT MINI has a W25Q128JVS1Q storage chip.
- This chip has 16MB of flash storage
  
- **1x Pyro Channel**
- STARLIGHT MINI has one pyro channel for a single parachute deployment.
- While most rocket motors have a built-in ejection charge, it can be unreliable and has a very loose tolerance for timing.
- With STARLIGHT MINI, you can set parachute ejection timing to the foot.

## 2.2 Schematic & Labeled Board

The schematic is attached below. A larger version can be [downloaded here](#).



# 3 Usage

## 3.1 MissionControl

MissionControl is a software designed for use with Circuit Wizardry flight computers. MissionControl supports STARLIGHT MINI and programming the pyro channel on-board, as well as extracting data from previous flights.

## 3.2 Custom Firmware

STARLIGHT MINI supports the same kinds of custom firmware that our other STARLIGHT boards do! However, it may be less desirable to upload custom firmware to a STARLIGHT MINI board due to the lack of GPIO pins and servo outputs. MissionControl has almost all the customizability you'd ever need. However, it is still possible to upload custom firmware to STARLIGHT MINI.

If you're doing any sort of custom firmware work with STARLIGHT MINI, the Circuit Wizardry GitHub is going to be your best friend. Check it out at <https://github.com/Circuit-Wizardry/>

## 3.3 Powering STARLIGHT MINI

It's recommended to power STARLIGHT MINI with a 1s LiPo battery. STARLIGHT MINI can be powered with anywhere from 2.8 to 6V.

STARLIGHT MINI includes two headers, labeled PWR\_CS. When these two are shorted with either a jumper or a switch, the battery provides power to the board.

### 3.3.1 IMPORTANT INFORMATION

**STARLIGHT MINI DOES NOT INCLUDE A CHARGING CIRCUIT! MAKE SURE YOU NEVER HAVE PWR\_CS SHORTED WHILE USB IS PLUGGED IN. THIS COULD LEAD TO ISSUES ON THE USB SIDE!!**

# 4 How does our firmware work?

## 4.1 MissionControl Firmware

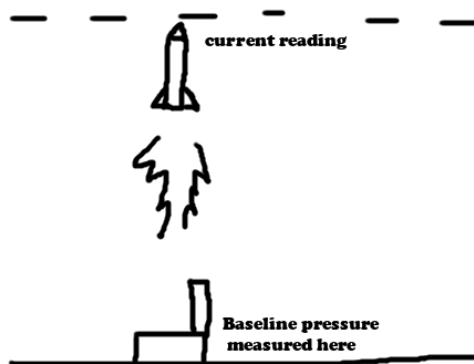
In order for STARLIGHT MINI to be able to collect data as well as detect apogee and other critical events during the rocket's flight, the board must run some computations that take this data and interpret it, turning it into something useful. This is what MissionControl is for. MissionControl is our end-to-end software solution that handles everything from programming the board to reading data and making inferences based on the collected data.

### 4.1.1 Altitude Determination

STARLIGHT MINI determines its altitude with a simple loop - the board takes the last five pressure measurements and averages them in order to get a more accurate picture on altitude (the reason for this is because sometimes there will be spikes or drops in pressure & averaging can help mitigate these spikes). This averaged pressure value is simply put through a function that determines the altitude at a certain pressure.

Source: <https://www.weather.gov/media/epz/wxcalc/pressureAltitude.pdf>

$p$  = current pressure;  $a$  = altitude



$$a = (1 - p/1013.25^{0.190284}) * 145366.45$$

### **4.1.2 Launch Detection**

Using STARLIGHT MINI's built-in accelerometer, we are able to detect how fast the rocket is accelerating. In order to detect launch, STARLIGHT MINI needs to register at least 1.5g of acceleration on the Y axis. This allows STARLIGHT MINI to register launch much faster than pressure readings would allow it to.

### **4.1.3 Apogee Detection**

STARLIGHT MINI detects apogee solely with pressure. The reason for this is because when the rocket is in free fall, there will be no forces applied other than  $F_g$ , which means there's no way of detecting when the rocket is at apogee with an accelerometer (drag could be used, but pressure is significantly simpler). STARLIGHT MINI uses the averaged pressure and saves the lowest average pressure experienced (low pressure = high altitude). If the current average pressure reading is not within 0.1hPa of the lowest average pressure experienced, a counter ticks up. The counter is reset once the average pressure reading returns to within 0.1hPa of the lowest average pressure experienced. If the counter reaches three AND launch has been detected AND the rocket is more than 20 feet in the air (to prevent false positives on the ground), apogee is detected.

### **4.1.4 Motor Burnout Detection**

Motor burnout detection is fairly straightforward. Once all accelerations are zeroed out (rocket is in free fall), motor burnout is detected.

### **4.1.5 x Feet Reached Detection**

x feet reached is also fairly straightforward. We already have altitude data, all we have to do is check if our rocket is above/below the threshold, and whether its reached apogee or not.

- If we are measuring feet on the way down, this event will be triggered only if apogee has already been detected and altitude is less than the threshold.
- If we are measuring feet on the way up, this event will be triggered only if apogee has not been detected and altitude is more than the threshold.

### **4.1.6 Landing Detection**

Landing is detected when the board is below 10 feet of altitude and has already reached apogee.



# 5 MissionControl

## 5.1 Disclaimer

MissionControl is still under **HEAVY DEVELOPMENT**. Not all features work yet. I'm working as hard as I can getting MissionControl to a full release-ready state. **This section is up to date as of 4/10/2024**. If you are reading a paper copy of this manual, it may be worth checking out our digital MissionControl guide that is continuously updated. Check it out at <https://circuitwizardry.com/missioncontrol>.

## 5.2 How does MissionControl interface with our flight computers?

MissionControl is built around two pieces of software - firmware that runs on the flight computer and accepts connections and data, and MissionControl itself. Circuit Wizardry flight computers come pre-installed with MissionControl-compatible firmware. However, if you own a legacy board that did NOT come with the firmware, or you removed the firmware or need to update it, follow the below steps:

In order to flash new firmware to your board, it is highly recommended to install the latest UF2 as well.

**Find the STARLIGHT MINI UF2 at**

<https://github.com/Circuit-Wizardry/slmini-missioncontrol>

Hold down the "BOOTSEL" button on your flight computer and plug it into USB. After a few seconds, you should notice it register as a mass storage device. Drag the appropriate UF2 into this storage device, and allow the board to reboot.

## 5.3 Flight Mode vs Idle/Programming Mode:

The firmware on our flight computers is structured around two main modes - Flight Mode, and Idle/Programming Mode. All boards are by default in Idle/Programming mode. In this mode, your board is able to connect to MissionControl and you can program it or extract flight data from the board. **In MissionControl, when flashing your code, there's an option to set the board to Flight Mode.** If you select this option, the next time the board is plugged in it will boot in flight mode.

**WARNING: DO NOT BOOT THE BOARD IN FLIGHT MODE UNTIL YOU ARE READY TO FLY AND THE BOARD IS POSITIONED IN THE ROCKET TO AVOID FALSE DETECTIONS**

It is recommended to have a switch that can be toggled right before flight so the board doesn't have to be jostled or moved once it's booted into flight mode. STARLIGHT MINI includes two headers labeled PWR\_CS that, when shorted, provide power to the board from the battery. Check out **3.3 Powering STARLIGHT MINI** for more information.

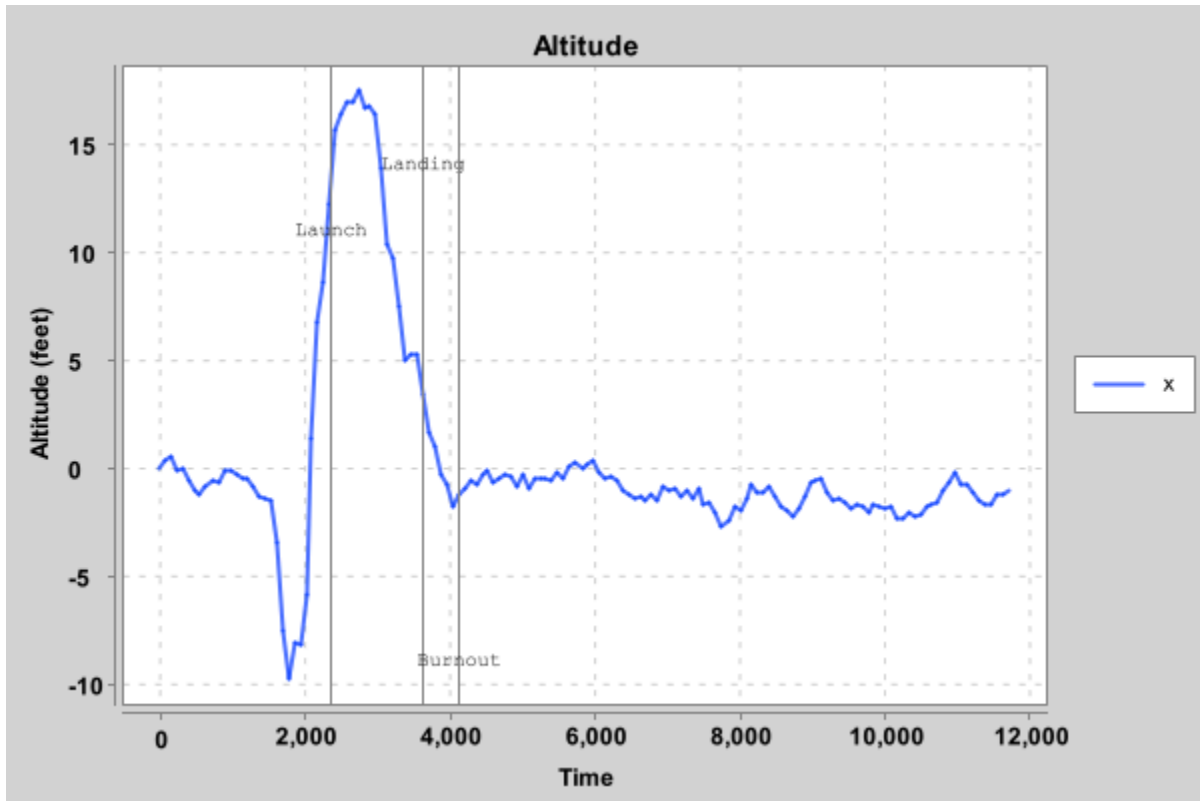
In flight mode, the board will be collecting data and is ready to fly. Simply launch your rocket with the board in flight mode, and data will be collected and events be triggered!

**The board's on-board LED will flash when the board is in flight mode!**

## 5.4 How to use MissionControl with STARLIGHT MINI

1. Plug STARLIGHT MINI into your computer with a Micro-USB cable. Make sure you're using a data cable!
2. Run MissionControl and click the "Program" button.
3. Select the appropriate COM port and click "Connect". After a few seconds, you should see a message that says "Connected to STARLIGHT MINI board".
4. Customize your STARLIGHT MINI board how you see fit! MissionControl is fairly self-explanatory. If you are still running into confusion or issues using MissionControl, check out the full online guide at <https://circuitwizardry.com/missioncontrol>.
5. Once you're ready to fly, click "BURN TO BOARD" and check "Set to flight mode". Once you've sent the data to the board, unplug it. **DO NOT POWER THE BOARD AGAIN UNTIL YOU'RE READY TO LAUNCH!**
6. Once you're ready to launch, power the board and wait a few seconds. Make sure that the red LED is flashing. This indicates the board is in flight mode and is in standby, ready to fly. The board must be upright and NOT MOVING once powered.
7. Launch your rocket!
8. Once the rocket lands, cut off power to the board. The board is now back in idle mode. Flight data will not be overwritten in idle mode, and you are free to plug it and unplug it as many times as you want without risk of losing your flight data.

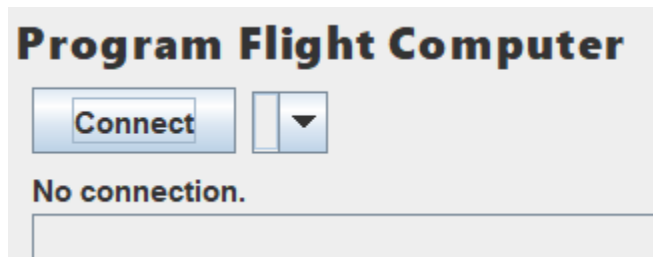
9. Run MissionControl and click “Debrief”.
10. Connect to STARLIGHT MINI and press “View”. Give MissionControl some time! This may take upwards of one or two minutes, depending on how long your flight was.
11. You can view any of the graphs as PDFs in the ‘graphs’ folder that MissionControl creates in the directory its run in!



This is an example of the kind of graph that is generated. This graph is from me throwing STARLIGHT about twenty feet into the air. Obviously, the graph will be much smoother when it's over a longer period of time and the rocket flies higher. Graphs will be similar for STARLIGHT MINI, as it uses the same pressure sensor. In this graph, burnout is detected late because the board was in free fall the entire time, with no forces being applied.

## 5.5 Troubleshooting

**My board doesn't show up in the list of COM ports!**



Make sure you're using a data cable to connect the board to your computer. If you're not sure, check your Device Manager and see if the COM port is showing up there.

**The "Program Flight Computer" screen won't even open! I click the button and nothing happens!**

This is almost always due to using the wrong Java version. Using the wrong Java version makes jSerialComm (the library used for communicating over serial ports) get all wonky and sometimes won't even register as a dependency!

Make sure you're running the program with Java 17, and your JAVA\_HOME is set to Java 17 as well.

If this doesn't solve your problem, run the jar file through the terminal with `java -jar missioncontrol.jar` and check out the error message. E-mail me at [contact@circuitwizardry.com](mailto:contact@circuitwizardry.com) with the error message you received and I'll help you get everything sorted out, as well as push an update to MissionControl hopefully solving the problem.

## **“There was an error connecting to your board”**

This issue can crop up for a variety of reasons. The error is thrown when an unrecognized board is connected, or MissionControl isn't able to receive a response from the board.

Try:

- **Make sure no other program is accessing the COM port!**
- **Make sure you have the right COM port selected!**
- **Unplug your STARLIGHT MINI board and plug it back in**
- **Update MissionControl and the firmware on your STARLIGHT MINI board (see 5.2 *How does MissionControl interface with our flight computers?* and <https://github.com/Circuit-Wizardry/missioncontrol>)**
- **Restart your computer**

If none of this works, e-mail me at [contact@circuitwizardry.com](mailto:contact@circuitwizardry.com) and I can provide further guidance.

## **The program won't even run!**

Check your Java version and your JAVA\_HOME environment variable. Make sure MissionControl is being run with Java 17.