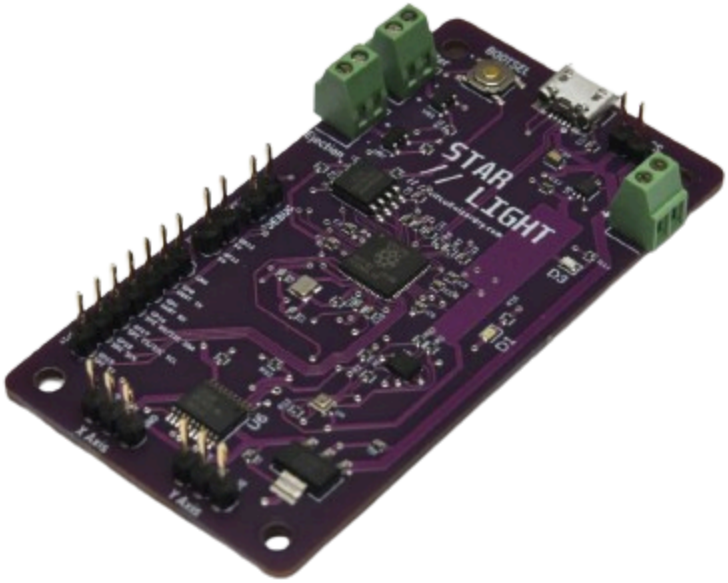


STARLIGHT

User Manual

V1.5

Circuit Wizardry - contact@circuitwizardry.com

Contents

1 **Basic Information**

1.1 **Specifications/Electrical Characteristics**

2 **Features**

2.1 **Board Features**

2.2 **Schematic & Layout**

3 **How do I use STARLIGHT?**

3.1 **MissionControl**

3.2 **Custom Firmware**

3.3 **Powering STARLIGHT**

3.3.1 **PWR_CS**

3.3.2 **Important Information**

4 **How does STARLIGHT 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**

4.2 **Custom Firmware**

4.2.1 **The Circuit Wizardry GitHub**

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 **0.0.3 vs 0.0.4**

5.6 **Thrust Vectoring**

5.7 **Audio/Visual Cues**

5.8 **Troubleshooting**

1 Basic Information

STARLIGHT is a flight computer designed for use in medium-to-high power model rockets. The board boasts altitude determination as well as a 6-axis inertial measurement unit (IMU) to allow the board to calculate its pitch and roll.

1.1 Specifications/Electrical Characteristics

Descriptor	**Value**
Dimensions (Length * Width * Height) (mm)	40 x 75 x 2.5
Weight (no headers) (g)	33
Input Voltage (V_{in}) (V)	5.5-18.5
Logic Voltage (V_{logic}) (V)	3.3
Servo Voltage (V_{servo}) (V)	5.0
Pyro Channel Current Draw (continuous) (I_{pyro}) (A)	2
Microcontroller Speed (MHz)	133

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

2 Features

STARLIGHT is designed with the model rocket enthusiast in mind. My goal with STARLIGHT was to pack as many features as possible onto a cost-friendly flight computer fit for use with medium and high-power rocketry.

2.1 Board Features

- **RP2040 dual-core MCU**
- STARLIGHT is built around the RP2040 dual-core microcontroller. It's an incredibly fast microcontroller and can be easily programmed if you want to write your own firmware.
- The RP2040 runs at 133MHz.

- **ICM-42605 6-axis IMU: Gyroscope and accelerometer**
- The ICM-42605 is a 6-axis IMU, meaning it has both a gyroscope and an accelerometer built-in. This does mean that STARLIGHT does not have a magnetometer, but the rocket's position can still be tracked easily with the accuracy of this IMU.
- After writing code for this IMU for the last few months, I've noticed its gyroscope and accelerometer are incredibly accurate and do more than enough to allow STARLIGHT to track its relative location and speed.

- **Level shifter to allow the use of 5V servos with the 3.3V RP2040**
- STARLIGHT has a built-in 5V voltage regulator to allow you to connect 5V accessories directly through the 5V GPIO pins & run 5V servos off the board.
- 5V is the typical operating voltage for most hobbyist servos.

- **Dual temperature sensing for added redundancy and protection from overheating**
- The ICM-42605 has a built-in temperature sensor. So does the BMP388. By combining these two measurements, we get a much more accurate temperature reading that is much less prone to failure.

- **BMP388 pressure sensor for altitude determination and flight tracking**

- The BMP388 sensor on STARLIGHT is used to read pressure data and make altitude estimations.
- The BMP388 is rated for an accuracy of +-0.5m, or +-1.64ft. After using the chip for a while, it's closer to +-2m or +-6.56ft.

- **6x 3.3V GPIO pins, plus exposed SPI, I2C, and UART interfaces for even more versatility**
- The board exposes more than enough GPIO pins to allow almost any customization to be made through both firmware and hardware.
- With the design files being open-source and free to use, it's totally up to you how you use STARLIGHT =)

- **2x 5V GPIO pins, made possible by the on-board level shifter**
- In addition to the six 3.3v GPIO pins, the board comes with an additional two 5V GPIO pins.
- Whether you're trying to trigger a 5V transistor or relay, or simply want to be able to communicate safely with a 5 volt device, these pins will come in handy.

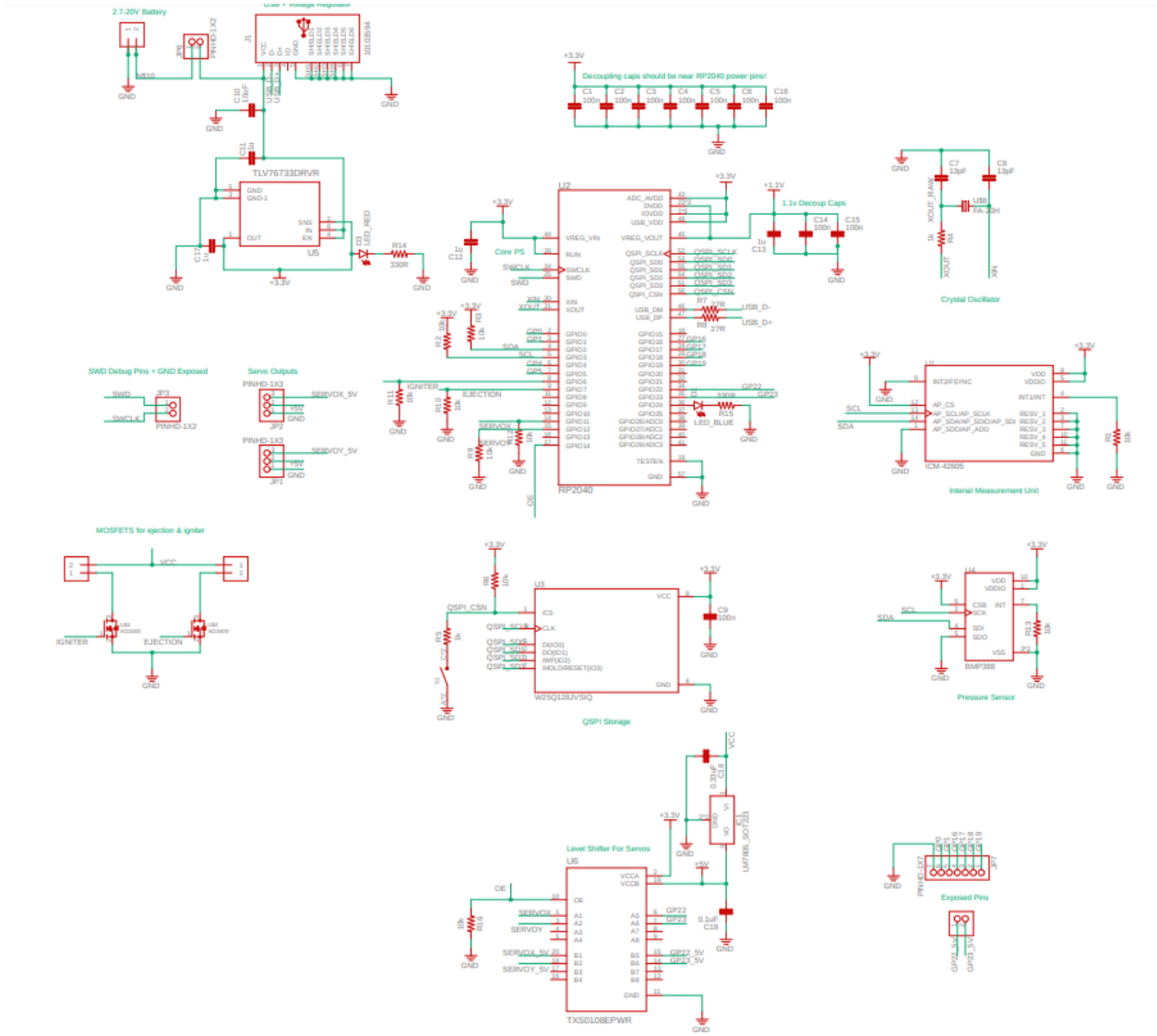
- **16MB of flash storage, for storing flight data and firmware**
- The boards comes with a built-in 16MB of flash storage.
- The MissionControl firmware runs at a default polling rate of 10Hz, which gives you over fifteen minutes of data collection before the flash storage fills up.

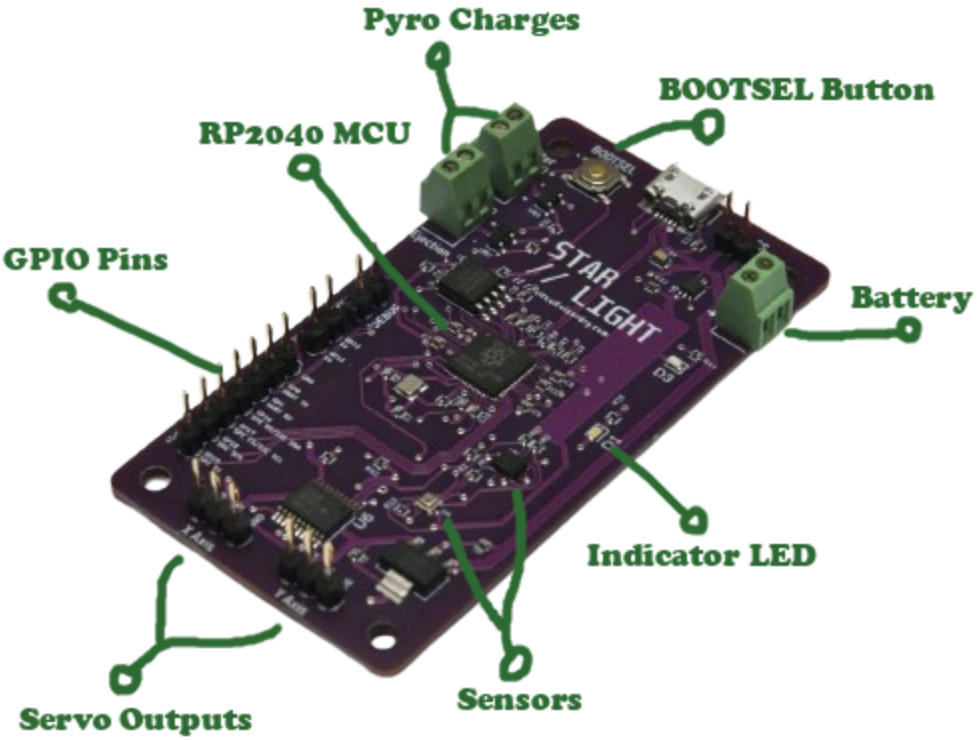
- **Indicator LEDs for power and runtime**
- STARLIGHT includes two LEDs - one red LED that is driven directly by V_{in} and one blue LED connected to GPIO pin 24.
- Check out **5.4 How to use MissionControl with STARLIGHT** for more info on the LED when using the default firmware.

- **Two servo outputs to allow for an optional thrust vector control interface**
- These two servo outputs can be used for thrust vectoring, parachute deployment, or anything else you could dream of.

2.2 Schematic & Layout

A larger version of the schematic is attached to the very back of this manual. If you're reading the digital version of this manual, you can download the schematic here.

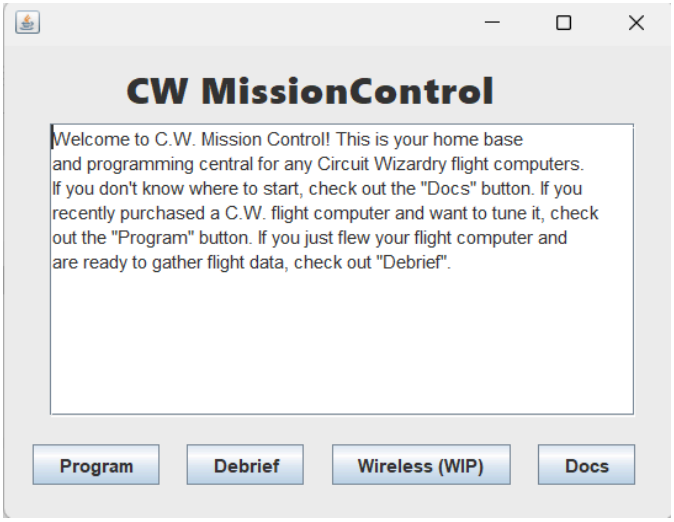



Pyro Charges
BOOTSEL Button
RP2040 MCU
GPIO Pins
Battery
Indicator LED
Servo Outputs
Sensors

3 How do I use STARLIGHT?

3.1 MissionControl

MissionControl is a software designed by Circuit Wizardry for use with Circuit Wizardry flight computers. STARLIGHT supports MissionControl programming and is easily programmed with the press of a button. Check out **Section 5, MissionControl** for information on how to fly your rocket using the MissionControl software.



3.2 Custom Firmware

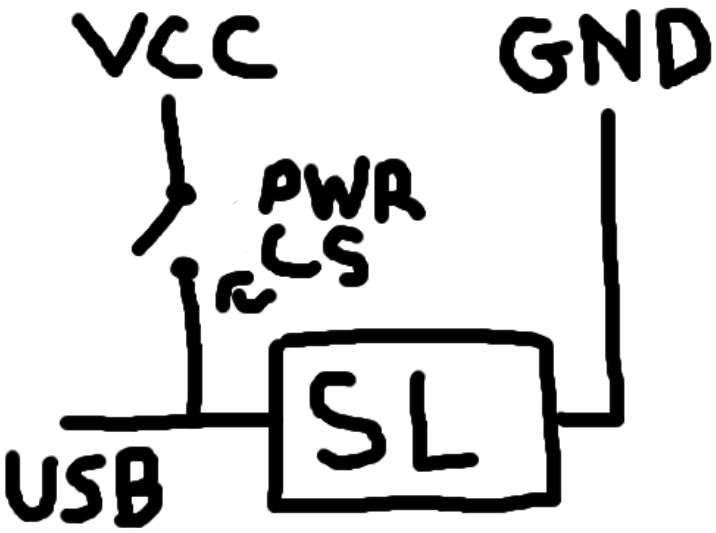
Of course, STARLIGHT supports the writing of custom firmware as well. Install the developer UF2 for STARLIGHT and you'll be off to the races programming your very own firmware for STARLIGHT! This task may be intimidating, but we have a whole guide at https://github.com/Circuit-Wizardry/starlight-python/tree/main on how to write custom firmware for your Circuit Wizardry flight computers.

3.3 Powering STARLIGHT

It is recommended to power STARLIGHT with a 2s LiPo battery if you're trying to use the pyro channels on-board (as LiPos have a very high max current draw). If you're simply looking for data collection and are using another system for parachute deployment, STARLIGHT can be powered from anywhere between 5.5-18.5 volts.

3.3.1 PWR_CS

STARLIGHT 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. A simplified schematic can be seen below.



3.3.2 Important Information

STARLIGHT DOES NOT INCLUDE A CHARGING CIRCUIT! Make sure you NEVER have PWR_CS shorted while STARLIGHT is plugged into USB. This could lead to an overvoltage on the USB side!

4 How does STARLIGHT work?

4.1 MissionControl Firmware

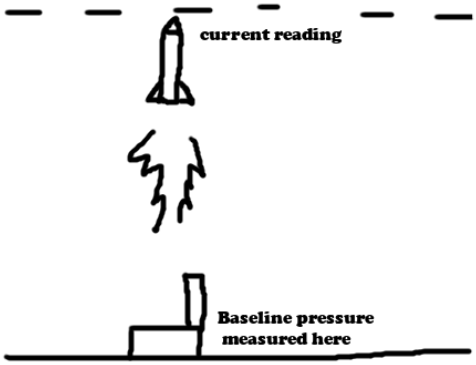
In order for STARLIGHT to detect launch, apogee, staging, etc, it requires fast collection of data. Not only does it need to be able to collect data, it needs to be able to run computations in order to turn the data into something usable. MissionControl firmware handles all of this for you, allowing you to focus more on other parts of your rocket. If you're interested in how the MissionControl firmware detects apogee, launch, and more, the details are outlined below.

4.1.1 Altitude Determination

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

4.1.3 Apogee Detection

STARLIGHT 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. STARLIGHT 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 five, 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

(0.0.3) Detecting landing is a bit tricky. In order to detect landing as fast as possible, we use sensor fusion and some trigonometry to figure out whether the board is moving or not. Using sensor fusion, we are able to cancel out gravity from our accelerometer readings. With this, it's simply a question of figuring out when these readings equal zero. Once they equal zero and apogee has already been detected, we detect landing.
(0.0.4) Landing is detected when the board is below 10 feet of altitude and has already reached apogee.

4.2 Custom Firmware

Obviously, with custom firmware, it's totally up to you to choose how STARLIGHT detects things such as apogee, landing, etc. We do provide some libraries with the STARLIGHT development UF2, however. These libraries make your life a little bit easier when interfacing with the ICM-42605 and the BMP388.

4.2.1 The Circuit Wizardry GitHub

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

The **starlight-python** repository includes the libraries and UF2 necessary for DIY firmware tasks with STARLIGHT, as well as step-by-step instructions on getting the development UF2 working on STARLIGHT. Check it out here.

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 UF2 at https://github.com/Circuit-Wizardry/starlight-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 includes two headers labeled PWR_CS that, when shorted, provide power to the board from the battery. Check out **3.3 Powering STARLIGHT** 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

1. Plug STARLIGHT 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 board".

4. Customize your STARLIGHT 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 blue LED is flashing (post 0.0.3, solid blue light indicates this). 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.

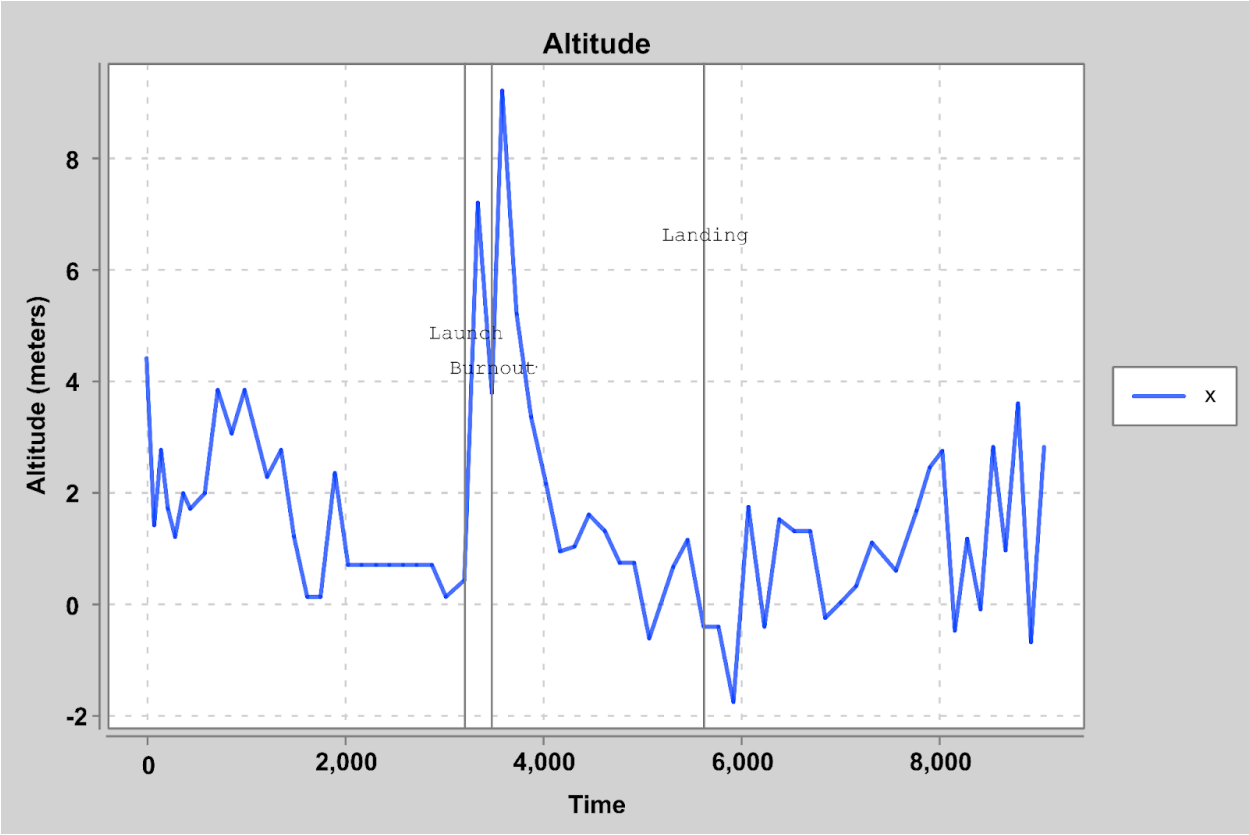
7. Launch your rocket!

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 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 thirty 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.

5.5 0.0.3 vs 0.0.4

There are many key differences between STARLIGHT's 0.0.3 version and the newer, more refined 0.0.4 version.

- **Refresh Rate:** 0.0.3 would collect data at 12.5Hz. 0.0.4 will collect data as fast as the board will allow.
- **Thrust Vectoring:** 0.0.4 allows thrust vectoring, while 0.0.3 does not.
- **Threaded Data Collection:** 0.0.4 utilizes both of the RP2040's cores in order to streamline collecting data and running thrust vector control calculations.

5.6 Thrust Vectoring

As of update 0.0.4, STARLIGHT and MissionControl both support thrust vectoring on the X and Z axes.

Check MissionControl for a “TVC” option when burning flight data to the board to see if your board is updated.

5.7 Audio/Visual Cues

In order for you to know exactly what's going on in STARLIGHT, the board has certain audio/visual cues that help you understand what's going on. **Note that you'll need a buzzer connected for audio cues!**

STARLIGHT Audio/Visual Cues:

Booting into Idle mode: 2 short beeps

Misformed JSON/First boot: 2 short flashes of the LED on bootup

Reading/Writing data when connected to computer: LED flashing

Connected to computer: one short beep

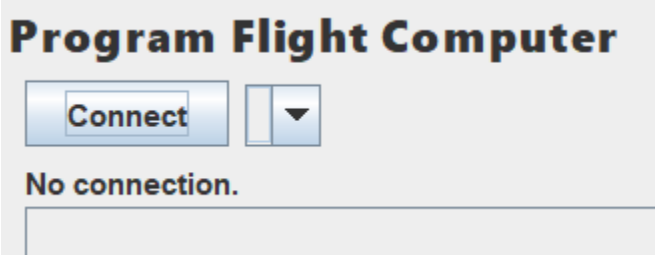
Flight Mode Cues:

Startup: 2 short beeps followed by 3 long beeps

Ready to fly: Solid blue LED

5.8 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 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 board and plug it back in**
- **Update MissionControl and the firmware on your STARLIGHT 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 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.