

8. STEM Payload Board

[Edit](#) [New page](#)

alanjohnston edited this page on Apr 14, 2022 · 30 revisions

8.1 Building the STEM Payload Board

These are the instructions for building the STEM Payload board.



You will need these tools:

- Safety glasses (to protect eyes while soldering or trimming leads)
- Soldering iron and solder (I use lead-free solder, but leaded solder is easier to work with)
- Needle nose pliers (to bend leads and hold

Pages 129

Find a page...

▶ [Home](#)

▶ [1. Main Board 1](#)

▶ [2. Software Install](#)

▶ [3. Ground Station](#)

▶ [4. Main Board 2](#)

▶ [5. Battery Board](#)

▶ [6. Solar Panel and Frame](#)

▶ [7. Main Board 3](#)

▼ [8. STEM Payload Board](#)

8.1 Building the STEM Payload Board

Checklist

Assembly

Video

Pro Micro or STM32

8.2 Testing

Video

Blink Test

Payload Sensor Testing

Complete Payload Test

Payload Serial Connection

parts)

- Side cutters (to trim leads)

Other tools that are helpful:

- Blue mounting putty (to hold components in place while soldering)

Checklist

The BOM has a sheet "By Steps" which lists the parts needed for each step in order.

<http://cubesatsim.org/bom> If you have a Google account, you can make a copy of this spreadsheet ("File" then "Make a Copy") and check off each part as you install it.

For example, here is the checklist for this step:

Troubleshooting

8.3 STM32 Steps

Flashing the USB Bootloader

Arduino Programming IDE

Additional Sensor Info

▶ [9. Board Stack](#)

▶ [9. Index](#)

▶ [Creating the CubeSatSim Raspberr...](#)

▶ [CubeSatSim Lite](#)

▶ [CubeSatSim Loaner User Guide](#)

▶ [CubeSatSim Mini](#)



















Show 114 more pages...

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/alanbjohnst>



<input type="checkbox"/>	Step 8.STEM Payload Board				https://github.com/alanbjohr	
<input checked="" type="checkbox"/>	Ref	Item	Qty	Location	Image	
<input type="checkbox"/>		STEM Payload PCB	1	N/A		
<input type="checkbox"/>		Alt 5a: Sparkfun Pro Micro	<input checked="" type="checkbox"/>	1	Top	
<input type="checkbox"/>	U1	Alt 5a: 12 pin female header for Pro Micro	2	Top		
<input type="checkbox"/>		Alt 5a: 1x12 pin header for Pro Micro	2	Top		
<input type="checkbox"/>		Alt 5b: STM32F103C8T6 "Blue Pill" with head	<input type="checkbox"/>	1	Top	
<input type="checkbox"/>	U2	Alt 5b: 20 pin female header for STM32	2	Top		
<input type="checkbox"/>	D1-D3	ON Semiconductor 1N5817RLG	3	Top		
<input type="checkbox"/>	R1,R2	4.7k Ohm resistors, 1/8 Watt, 20% tolerance	2	Top		
<input type="checkbox"/>	LED1	Green LED	1	Top		
<input type="checkbox"/>	R3	1k Ohm resistor, 1/8 Watt, 20% tolerance	1	Top		
<input type="checkbox"/>	LED2	Blue LED	1	Top		
<input type="checkbox"/>	R4	100 Ohm resistor, 1/8 Watt, 20% tolerance	1	Top		
<input type="checkbox"/>	JP21	BME280 board Temperature/Humidity/Pressu	1	Top		
<input type="checkbox"/>	J5	MPU6050 9 Axis Gyro GY-521	1	Top		
<input type="checkbox"/>	J4	GPIO 20x2 female connector non-stacking	1	Bottom		
<input type="checkbox"/>	R9-R11	10k resistor	3	Top		
<input type="checkbox"/>	C1-C3	100nF capacitor	3	Top		
<input type="checkbox"/>	D4	1N4148 diode	1	Top		

Assembly

Video

Here is a video of the assembly part of this step:

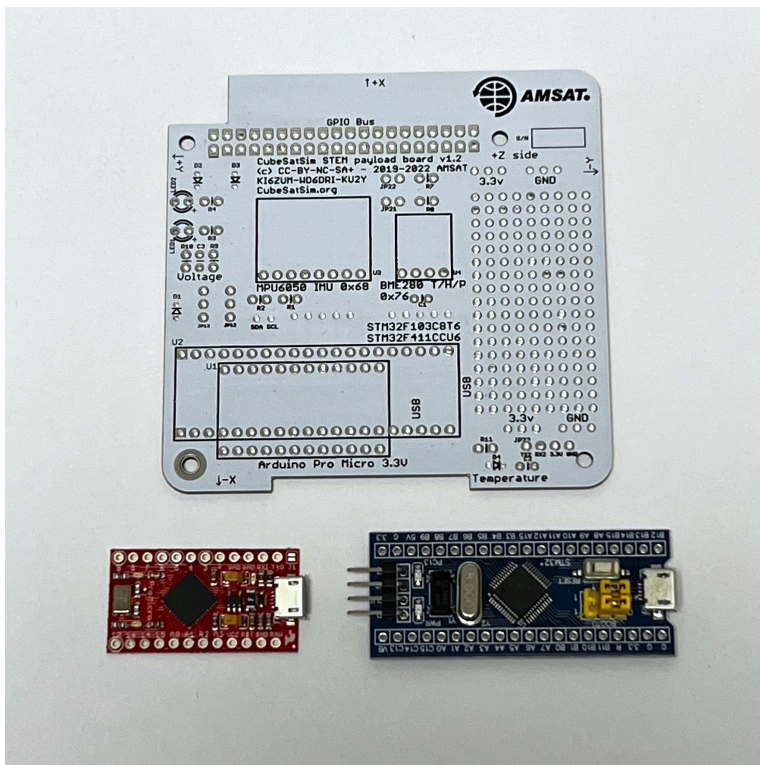
<https://youtu.be/cTxHojR-Z4Y>

Note: the video shows installing a 220 Ohm resistor for R3 instead of a 1k resistor as is now recommended - either will work, the green LED will be a little too bright with the 220 Ohm resistor.

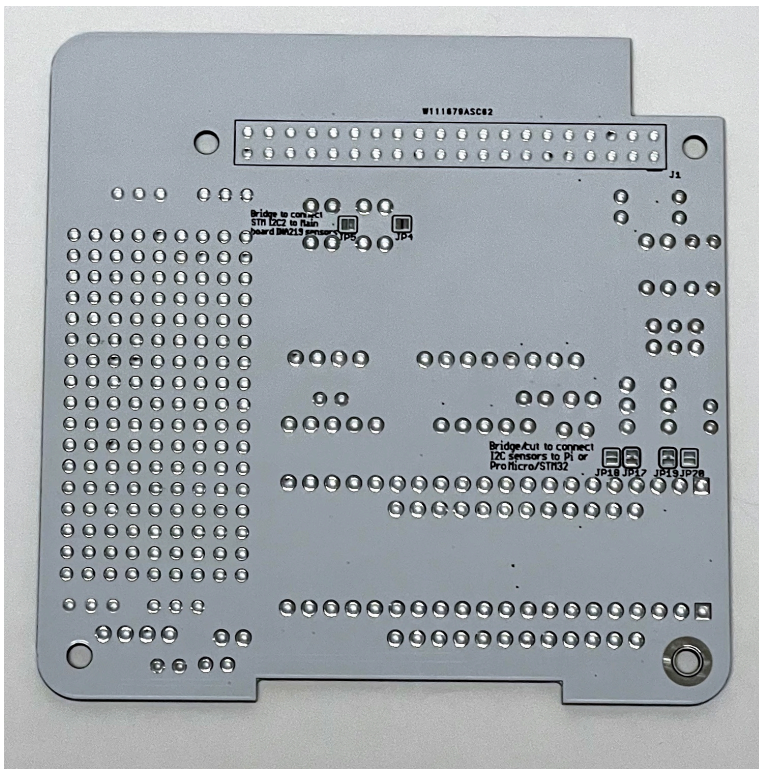
Pro Micro or STM32

This is the STEM Payload PCB Board along with the SparkFun Pro Micro board and the STM32 "blue pill" board. The STEM Payload Board can use either the Sparkfun Pro Micro microcontroller or the STM32F103C8T6 microcontroller, also known as a "blue pill". I recommend using the Sparkfun Pro Micro board since it is Arduino compatible and does not need to be flashed, making it a simpler alternative to the "blue pill". The "black pill" STM32 board can also be used, as described in later sections.

This photo shows the STM32 on the right, and the Pro Micro on the left. The Sparkfun Pro Micro is red, but clones are typically blue in color.



Here is the bottom of the PCB:



I strongly recommend using sockets for your Pro Micro or STM32. This way, you can program and test it unplugged from your board. This will help you figure out if any problems are a hardware or software issue.

The components, as described in the BOM

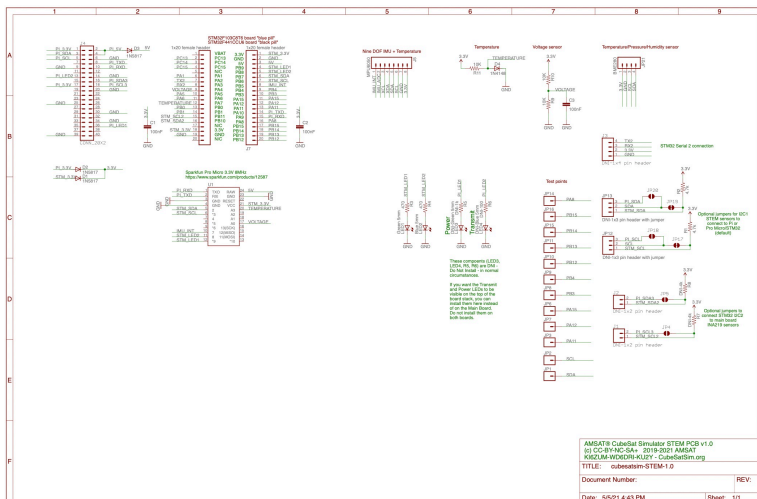
<https://cubesatsim.org/bom> are:

- Sparkfun Pro Micro microcontroller (3.3V, 8 MHz version) with two breakaway pin headers (alternatively a STM32F103C8T6 microcontroller mounted with two breakaway headers).
- 1N5817 diodes D1, D2, and D3 to supply power to the board (black with a grey band on one side)
- 4.7k Resistors R1 and R2 as pullups for the I2C bus
- BME280 Temperature Humidity Barometric Pressure Sensor (small purple board) with 1x4 pin header
- MPU6050 (GY-521) 3-Axis Accelerometer and

Gyro (larger blue board) with 1x8 pin header

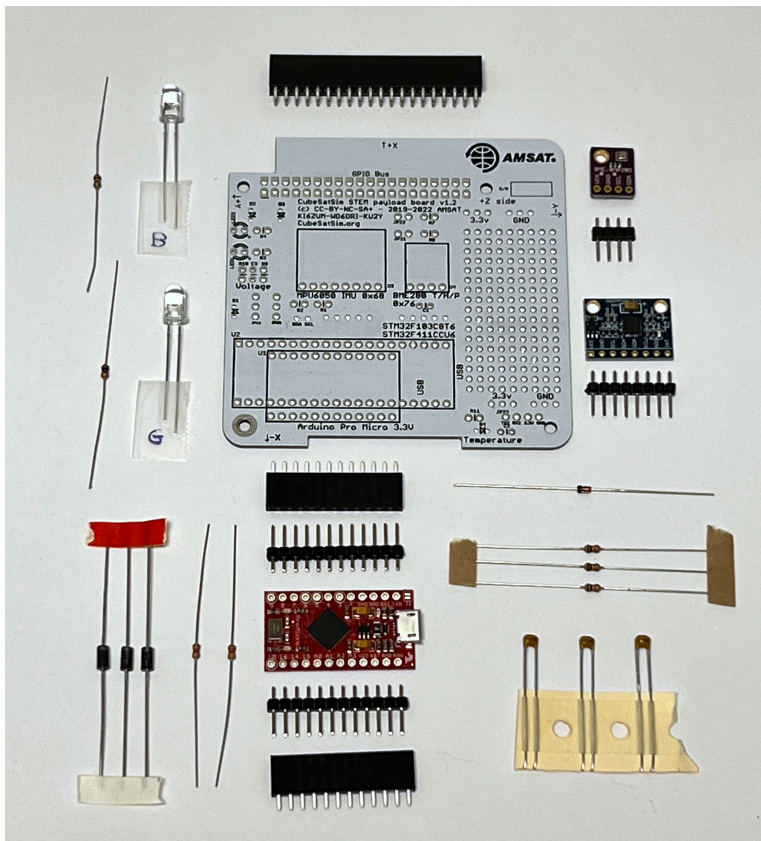
- GPIO female non-stacking connector
- 100nF capacitors C1, C2, and C3
- Diode D4 1N4148 (has "41" and "48" printed on it and a black band on one side)
- Green LED 5mm clear lens LED1
- Blue LED 5mm clear lens LED2
- 1k Ohm resistor R3
- 100 Ohm resistor R4
- 10k resistors R9, R10, and R11

Here is the schematic

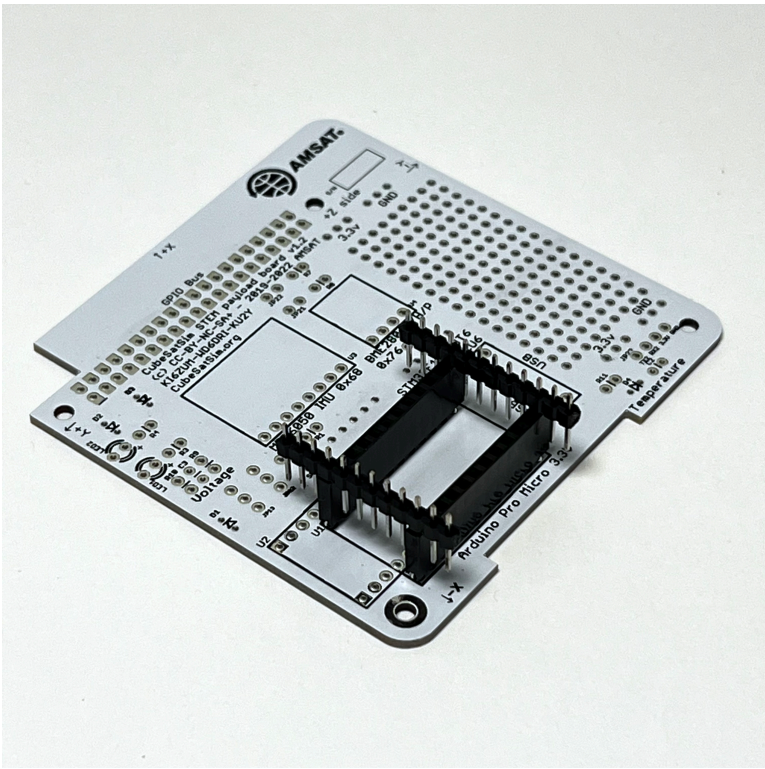


https://github.com/alanbjohnston/CubeSatSim/blob/master/hardware/v1.2/cubesatsim-STEM-1.2_schematic.pdf

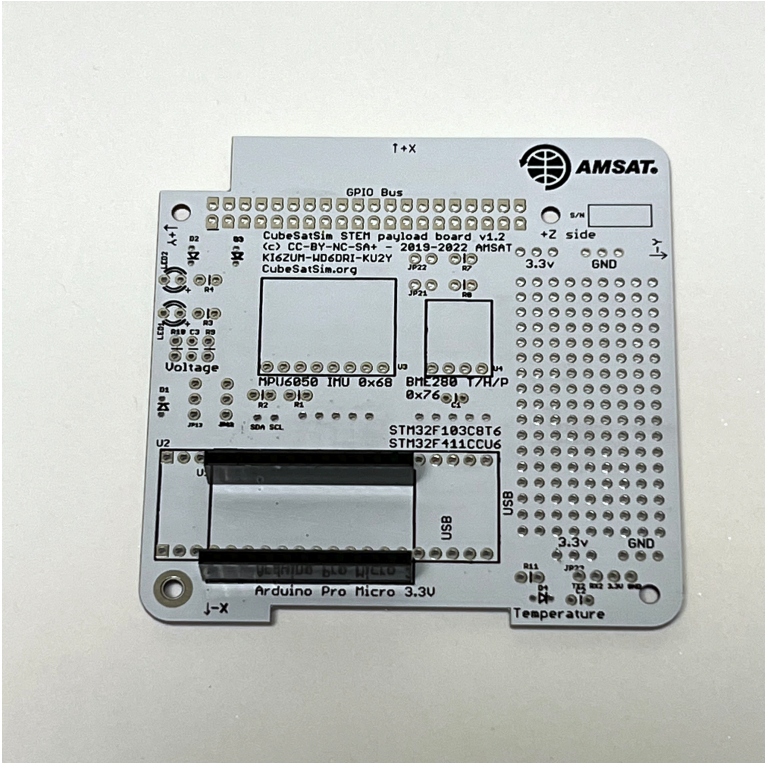
The complete set of parts is shown here:



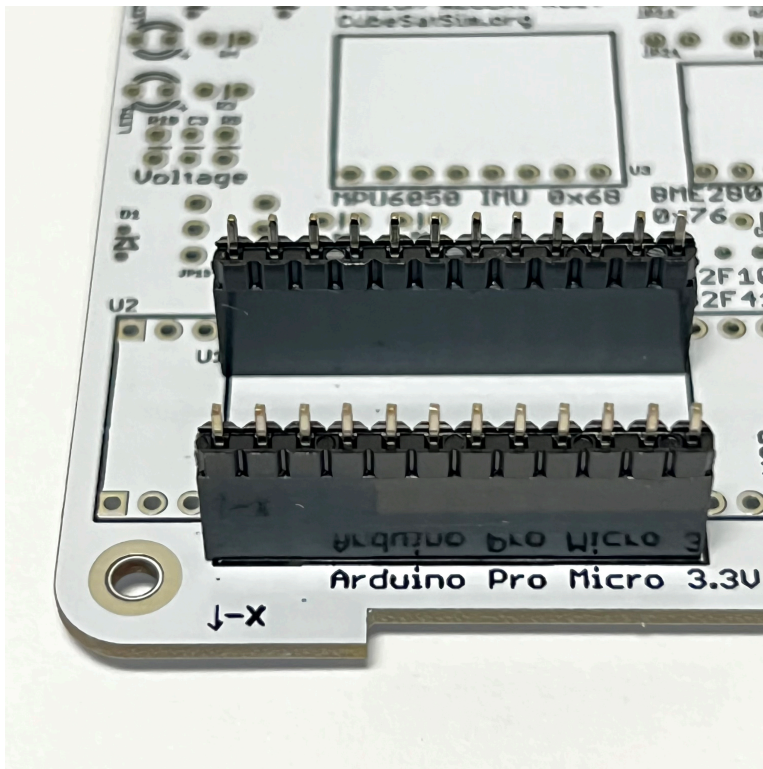
Start by mounting the female pin headers for the Pro Micro or STM32. It is important that the spacing and the alignment of the two headers is correct or you won't be able to plug in and unplug it. One way to do this is to use the male pin headers that come with the Pro Micro or STM3s to set the distance, as shown here:



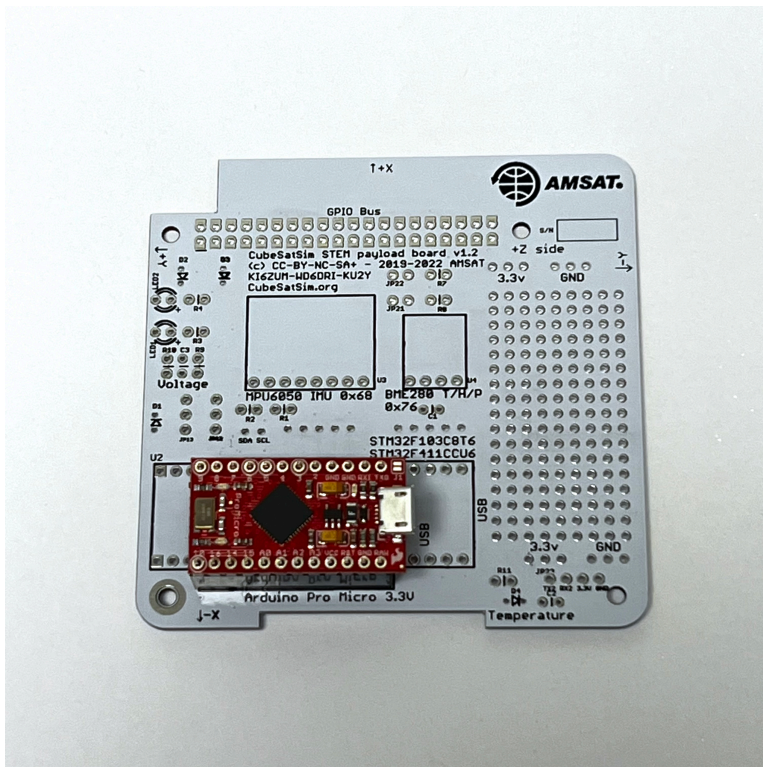
Once the headers are soldered in, they will look like this:



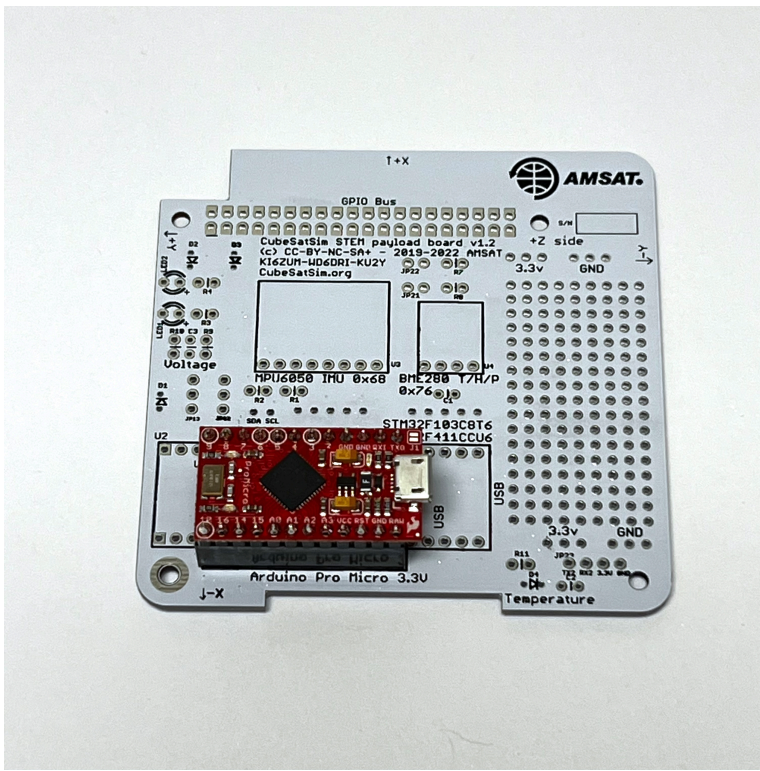
Next, plug the headers for the Pro Micro or STM32 into the sockets,



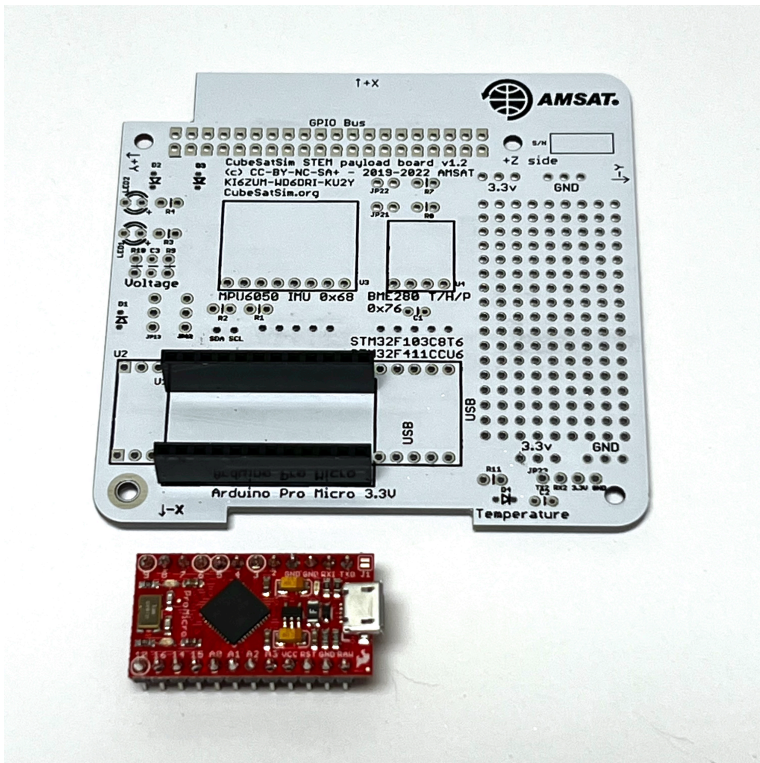
Then place the Pro Micro or STM32 on top as shown here and the pin headers then can be soldered onto the Pro Micro or STM32 board. Note that the micro USB port is to the right.



Solder the pin headers to the Pro Micro or STM32:

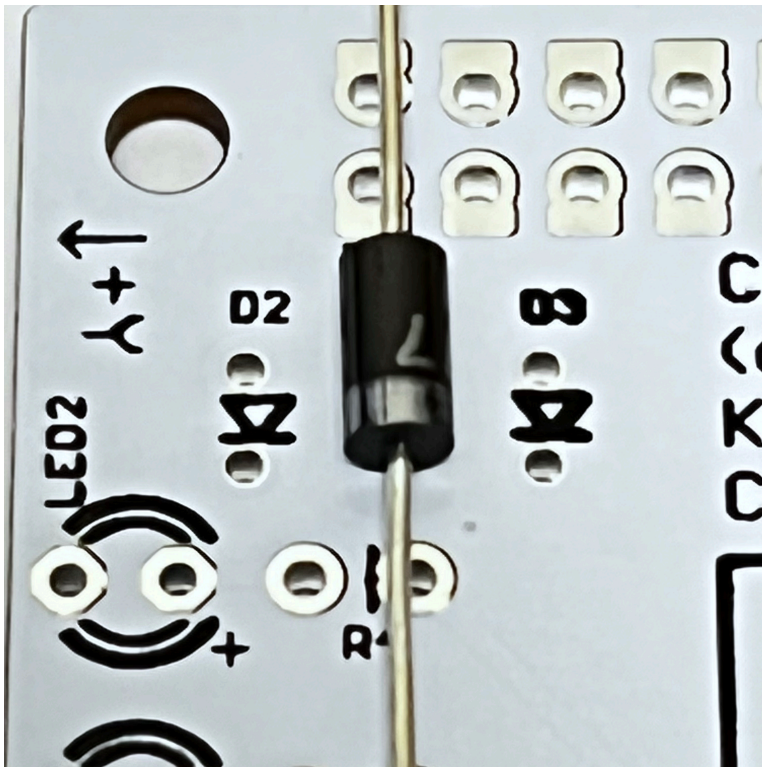


Then, the micro controller can be unplugged while the rest of the components are soldered in:



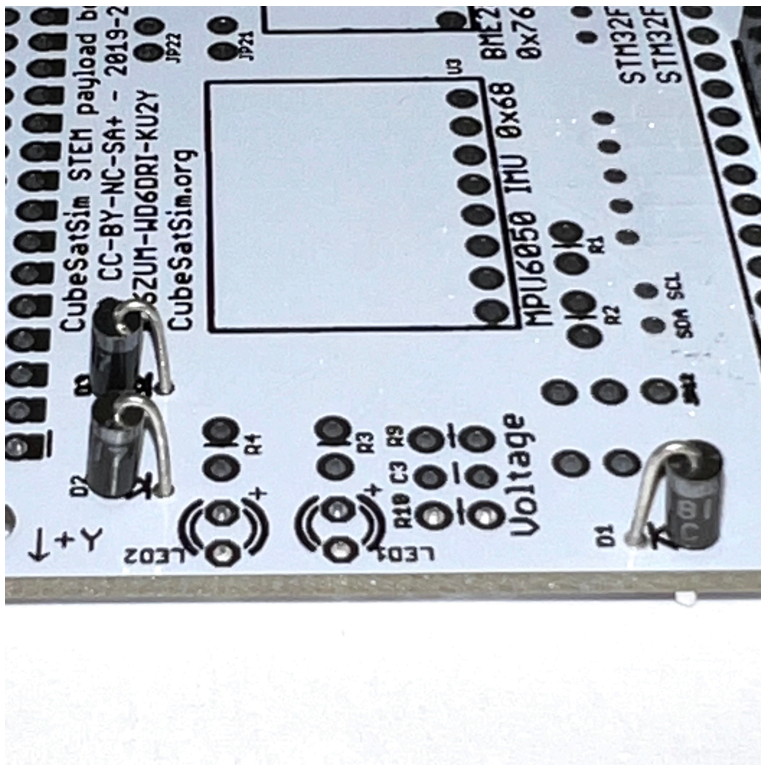
Next, solder the 1N5817 diodes D1, D2, and D3. The diodes allow the STEM Payload board to be powered by the Pi over the GPIO bus, but prevents the Pi from being powered accidentally from the microcontroller USB port. This allows you to safely plug a micro USB cable into your microcontroller even when the Pi is running.

Note that diodes have a polarity indicated by a grey band on one side. This image shows the PCB printing for D2 and D3, and the marking on the 1N5817 diode that it corresponds to.

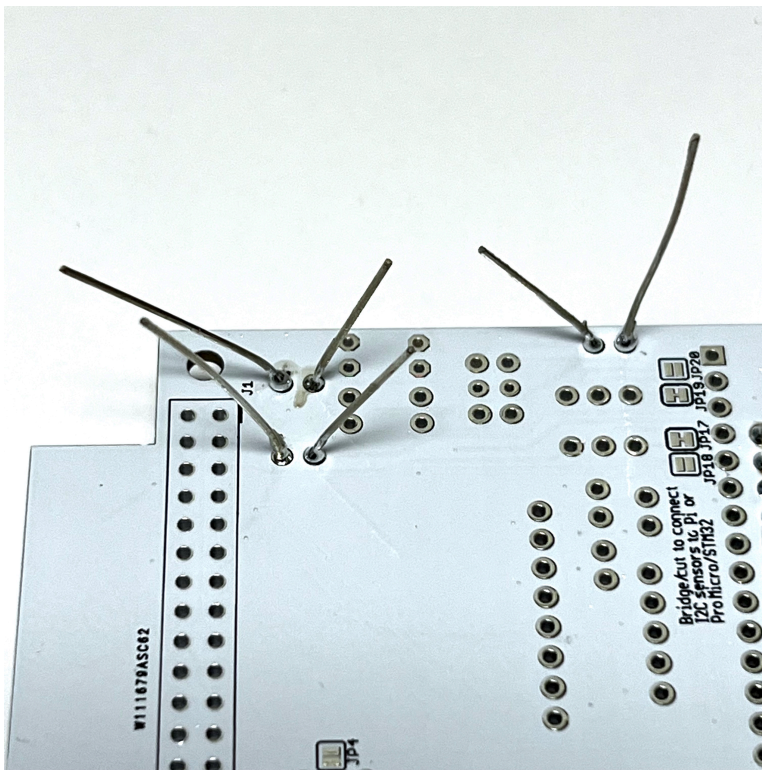




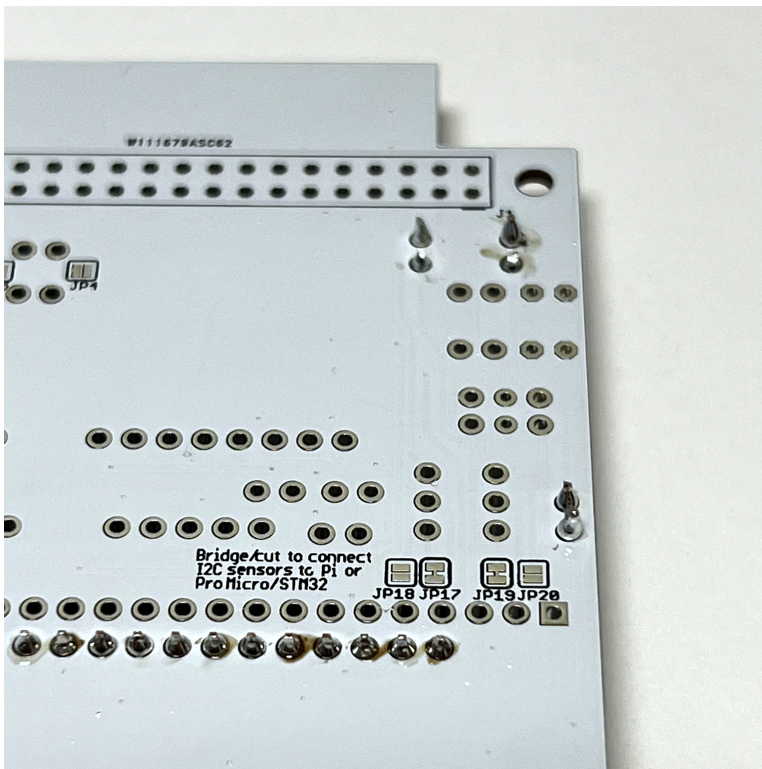
All the diodes, along with all the resistors on the board are mounted vertically. Bend over one lead on the diode and insert, making sure the polarity of the diodes D1, D2, and D3 is correct before soldering them:



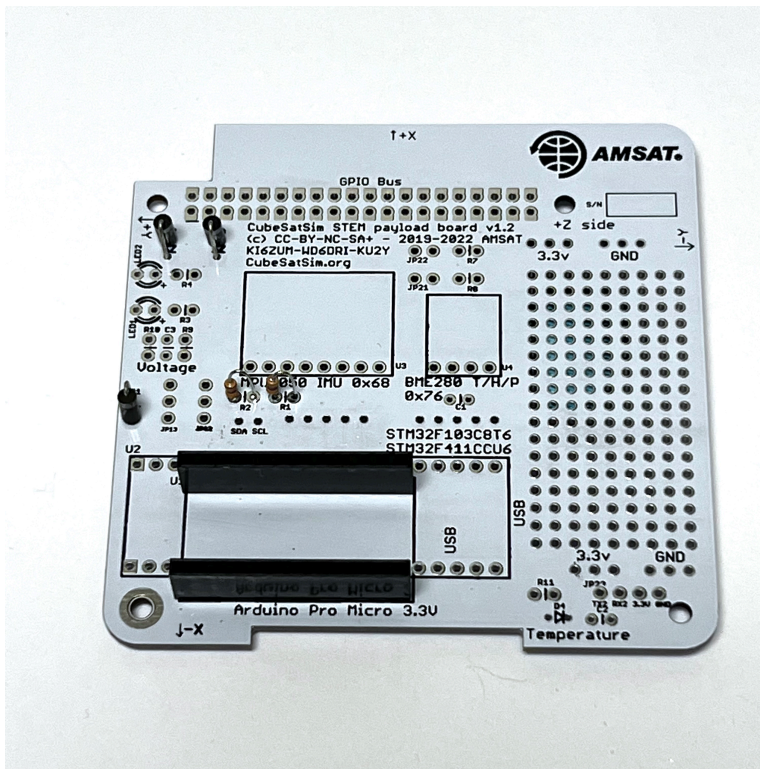
Solder the leads on the bottom of the PCB:



Then trim the leads, being careful to wear safety glasses and putting your finger over the end so it doesn't go flying:

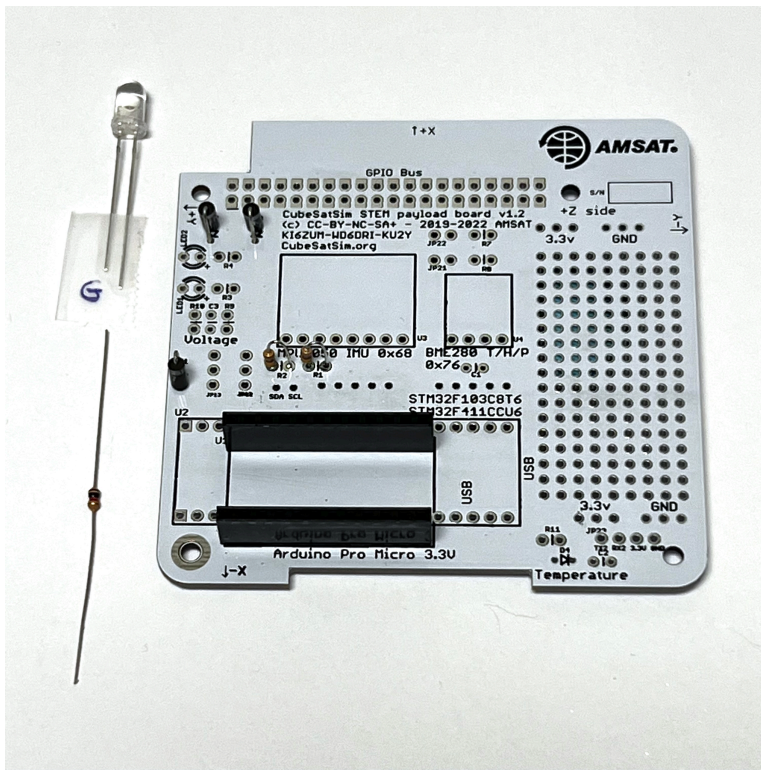


Then mount the two 4.7k resistors (yellow violet red color bands) R1 and R2:



Then install the green LED1 and 1k Ohm R3 (brown black red color bands). LEDs need to be installed with the correct polarity (one lead is + polarity and the other lead is - polarity) or it will not illuminate. The longer leg on the LED is the '+' lead and should be away from the edge of the PCB. Also, if you look at the LED lens from the top, it is circular but there is a flat side that marks the '-' side. So the flat side of the LED should be towards the edge of the PCB.

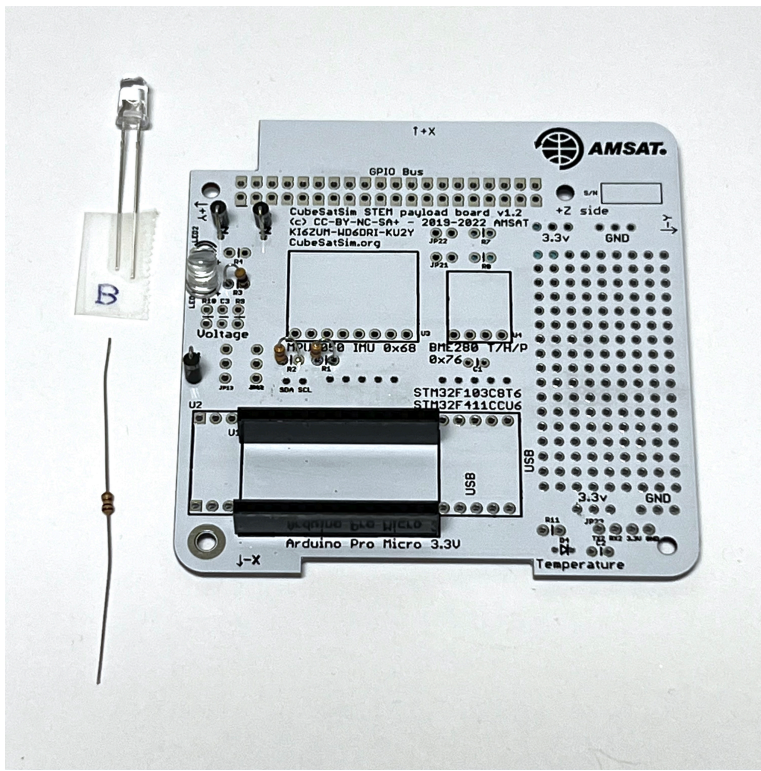
The longer lead of the LED is the positive and matches the '+' on the PCB. This LED and the blue one next to it are turned on and off by the microcontroller.



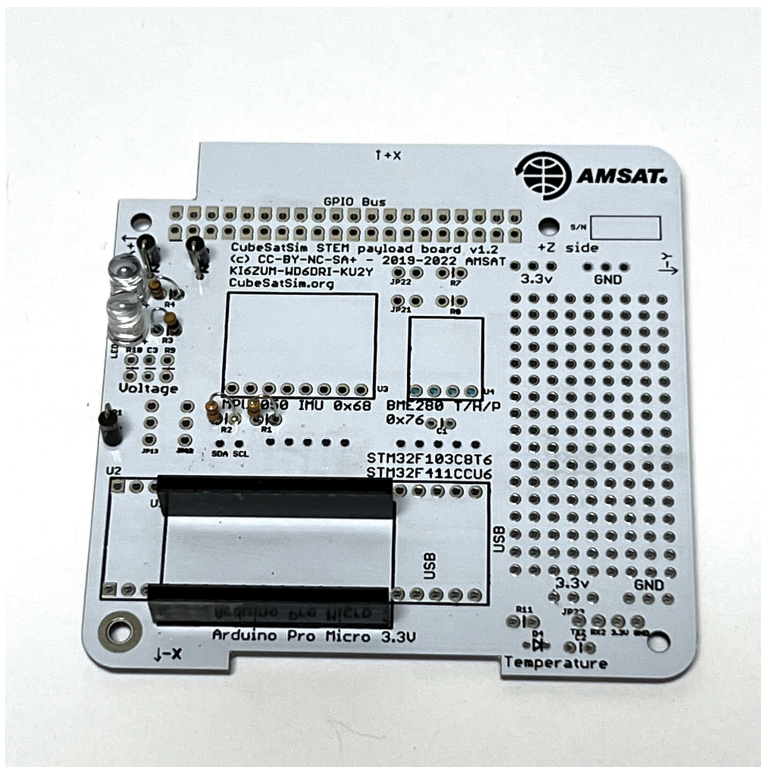
Here is the LED and resistor installed:



Then install the blue LED2 and 100 Ohm R4 (brown black brown color bands). The longer lead of the LED is the positive and matches the '+' on the PCB.



Here is the LED and resistor installed:



Then, mount the BME280 sensor (small purple board) and the MPU6050 Gyro and Accelerometer sensor (larger blue board) on the pin header.



Flip the PCB upside down and insert the GPIO header.

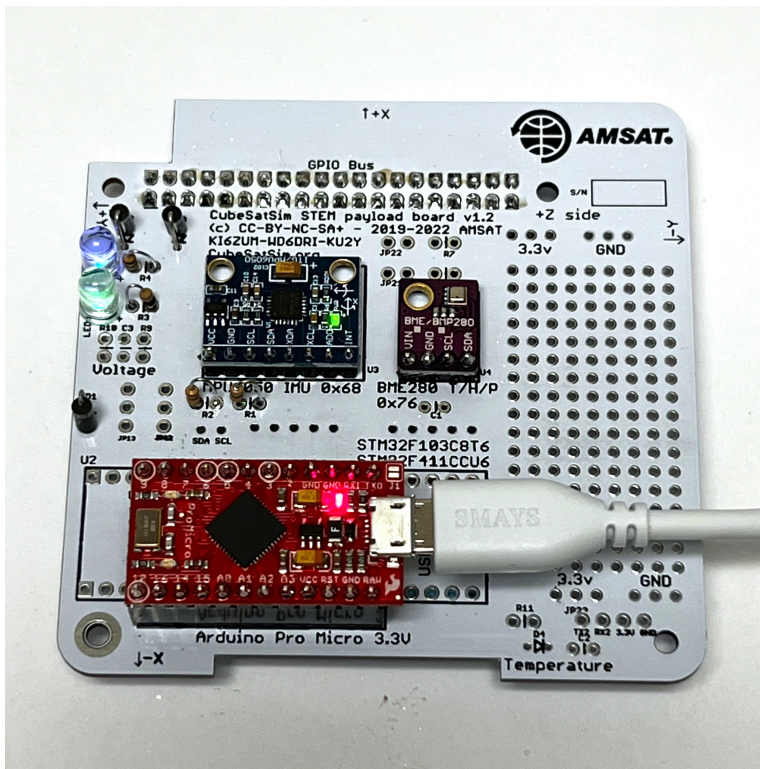


Flip the PCB right side up. This GPIO should be a non-stacking header in order to fit inside the CubeSatSim frame. If you aren't building it to fit inside the frame, you could use a stacking GPIO header.

Solder one pin on each side of the connector, then check to make sure it is flat and straight. If it is, solder the rest of the pins.



Connecting the micro USB cable to your PC will power the board and you should see a power LED on the Pro Micro and a power LED on the MPU6050 gyro:



You are now ready to test the board.

8.2 Testing

Video

Here is a video of testing the STEM Payload Board:

<https://youtu.be/ddvi4S12JKM>

Blink Test

To do the blink test, you need to program your microcontroller. Here are the basic instructions for the Pro Micro board, while the steps for the STM32 are in later sections. In Windows, the drivers should automatically install the first time you plug the Pro Micro board in using a micro USB cable. If not, there are instructions here

<https://learn.sparkfun.com/tutorials/pro-micro--fio-v3-hookup-guide> to download the driver and update it. This link also has MacOS and Linux instructions, too.

I'd recommend using the Arduino Integrated Development Environment (IDE) application to program your Pro Micro or STM32, although you can use other IDEs too. Download the latest version here: <https://www.arduino.cc/en/software>

You can even install the Arduino IDE on your Raspberry Pi Ground Station. The steps are listed here

https://create.arduino.cc/projecthub/techno_z/program-your-arduino-from-your-raspberry-pi-3407d4

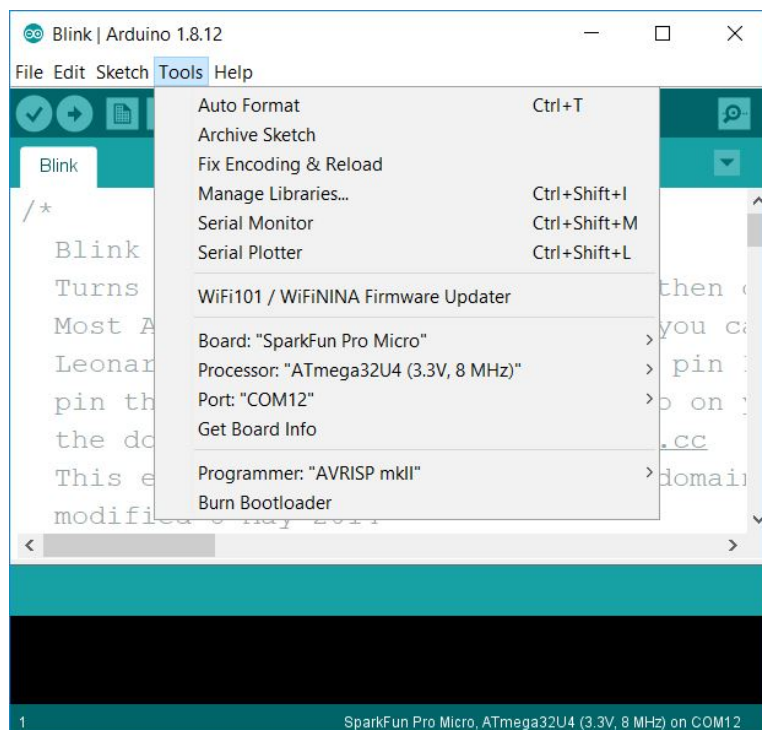
On my Pi, the Port was named /dev/ttyACM0.

Before you can program your Pro Micro using the Arduino IDE application, you will need to install the board files for the Pro Micro. First open the Preferences under File/Preferences and you will see a "Additional Board Manager URLs" text box near the bottom. Copy and paste this URL into the box:

https://raw.githubusercontent.com/sparkfun/Arduino_Boards/master/IDE_Board_Manager/package_sparkfun_index.json

Click OK to save it and exit the Preferences. Next under the Tools menu, mouse over the Board menu item and you should see the Boards Manager listed at the top. Click on it and it will open after a few minutes. At the top, there will be a search bar. Type in Sparkfun and search and you will see a listing for Sparkfun AVR Boards by Sparkfun Electronics. Click Install. When it finished the installation, click Close.

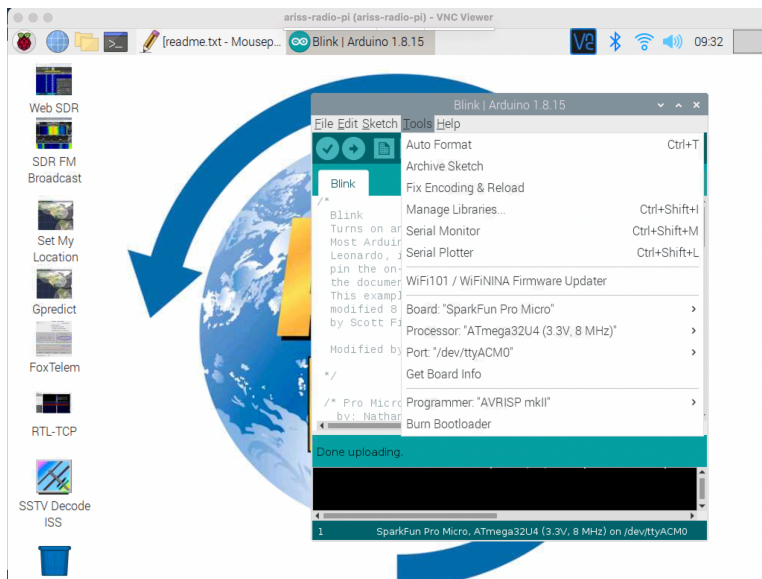
Now when you go under the Tools menu and select Board, you can select Sparkfun Pro Micro, possibly under SparkFun AVR Boards. You also need to select the correct version of the board. We use the 3.3V version of the board - if you look at the bottom of the Pro Micro board, the 3.3V option should be checked. If you accidentally purchased the more common 5V version of the board, you will need to get the 3.3V version. Under the Tools Menu, the Processor option should be set to ATmega32U4 (3.3V, 8 MHz). You will also have to select the Port for the Pro Micro. Here's what it will look like in the Arduino IDE when you are ready to program your Pro Micro (your COM port will likely have a different number):



Here is the Blink code (works for both the Pro Micro and STM32):

<https://github.com/alanjohnston/CubeSatSim/tree/master/stempayload/Blink>

If you are using the Arduino IDE on your Pi Ground Station, this is the configuration:



After copying this code and uploading to your board (the upload button has the arrow pointing to the right), on the Pro Micro board, the Transmit LED and Receive LEDs will alternatively blink. For the STM32, the LED should blink.

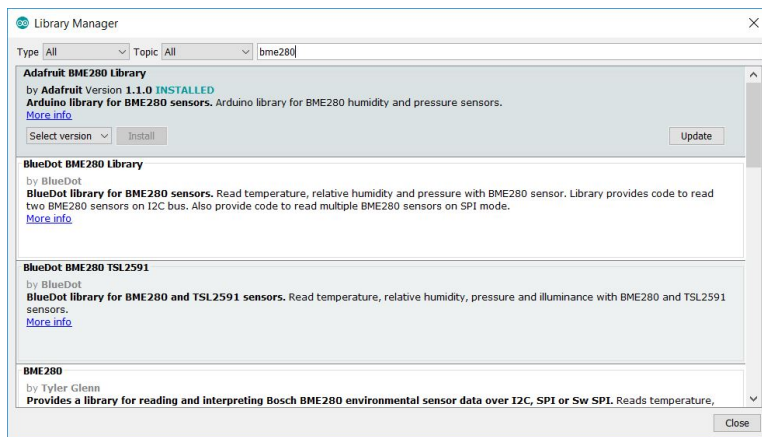
Payload Sensor Testing

You can now test that the Pro Micro can communicate with the two sensor boards. Here is the code that runs on both the Pro Micro and the STM32:

https://github.com/alanbjohnston/CubeSatSim/tree/master/stempayload/Payload_BME280_MPU6050_XS

This code requires several libraries which you will need to install. If you forget to install one or more libraries, you will get error messages saying "No such file or directory".

Open Tools/Manage Libraries and the Library Manager will open after a few moments. In the search box, type bme280 then return and you will see the Adafruit BME280 library. When you mouse over it, it will give a version pulldown menu and an Install button. Select version 1.1.0 and click on Install to install it. It should look like this:



Now type adafruit then return in the search box. A long list of Adafruit libraries will appear. Scroll down to find the Adafruit Unified Sensor library. Mouse over and select version 1.0.3 and click Install to install it. You will be prompted to install a few other library dependencies as well. Finally, type tockn then return in the search box and install the MPU6050_tockn library and click Install to install it.

Click Close to close the Library Manager. In the sketch window, click on the Verify icon (checkbox) and it should compile without errors. If you get an error saying "No such file or directory", go back to the Library Manager and make sure you have installed the correct version of all three libraries. Click on the Upload button (right arrow) and the Pro Micro should be programmed to read the sensors.

Once it is done uploading (indicated by the Done Uploading) message at the bottom of the window, open the Serial Monitor by clicking on the magnifier icon in the top right of the window. In the Serial Monitor window, make sure 9600 baud is set in the lower right. Click on the text box at the top and type a ? then click on Send. You should get a response displayed similar to this:

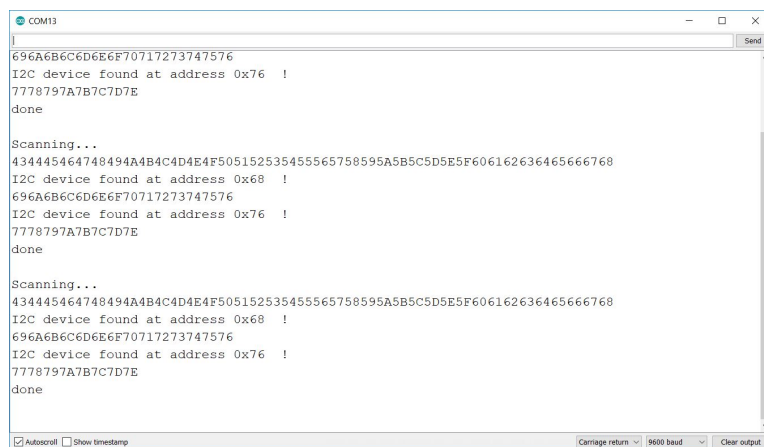
```
OK BME280 24.89 1003.96 77.61 21.13 MPU6050
-0.85 -2.48 -1.09 0.19 -0.15 1.02 XS 0 0
0.00
```

The OK is the status response. The four numbers after the BME280 are the temperature in Celsius, pressure in hPa, altitude in meters, and humidity in percentage read from the purple BME280 sensor. The six numbers after the MPU6050 are the X, Y, and Z axis angular rotation in degrees per second and the X, Y, and Z axis acceleration in g. If you get a series of zeros after a sensor, it means it was not successfully read by the Pro Micro. The three numbers after XS are extension sensor fields that you can set in the code by setting the Sensor1, Sensor2, and Sensor 3 variables.

If you get all zeros for a sensor, you need to determine if it is a hardware or software problem. Running an I2C bus scanner program will tell you if the sensor can be accessed on the I2C bus. The program is:

https://github.com/alanbjohnston/CubeSatSim/tree/master/stempayload/i2c_scanner

Upload this to your Pro Micro (or STM32) and then open the Serial Monitor. You should see:



```
COM13
|
696A6B6C6D6E6F70717273747576
I2C device found at address 0x76 !
7778797A7B7C7D7E
done

Scanning...
434445464748494A4B4C4D4E4F505152535455565758595A5B5C5D5E5F606162636465666768
I2C device found at address 0x68 !
696A6B6C6D6E6F70717273747576
I2C device found at address 0x76 !
7778797A7B7C7D7E
done

Scanning...
434445464748494A4B4C4D4E4F505152535455565758595A5B5C5D5E5F606162636465666768
I2C device found at address 0x68 !
696A6B6C6D6E6F70717273747576
I2C device found at address 0x76 !
7778797A7B7C7D7E
done

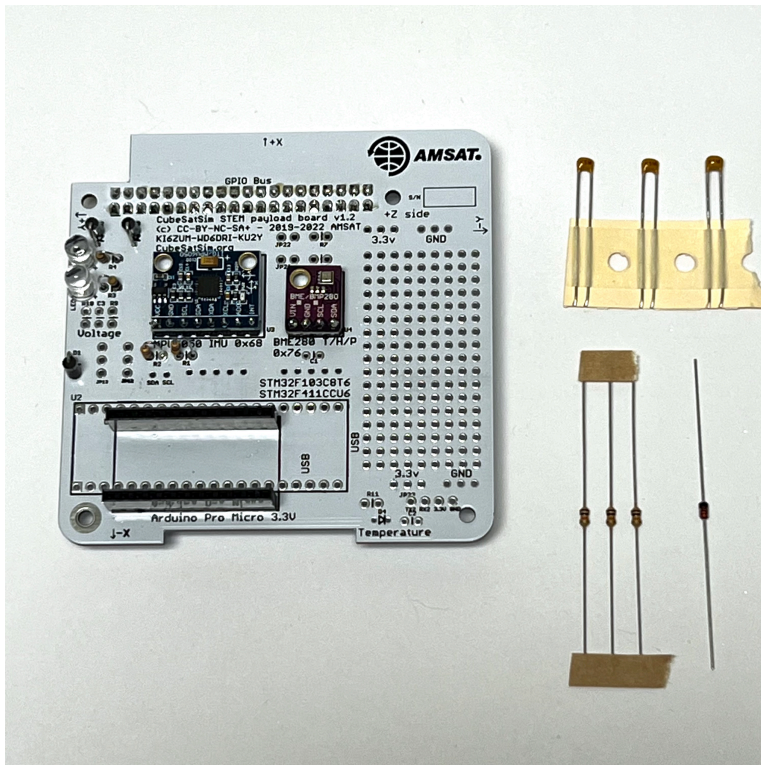
Autoscroll Show timestamp Carriage return 9600 baud Clear output
```

The blue MPU6050 should be at address 0x68 while the purple BME280 should be at address 0x76. If neither device is present on the I2C bus, make sure resistors R1 and R2 are soldered in and are 4.7k in value. If one sensor shows up but not the other, it might be due to soldering on the sensor pins or due to a bad sensor board.

If both sensors show up on the I2C bus but you get zeros in the Serial Monitor, this indicates a software problem.

Next, you can complete the build, as shown here by adding:

- Three 10k resistors (black brown orange color bands) R9, R10, and R11
- Three 100nF capacitors C1, C2, and C3
- 1N4148 diode D4 (has a glass case with the numbers "41" and "48" written on it and a black band to indicate polarity).



Diode D4 also has a black band on one lead to indicate polarity:

https://github.com/alanjohnston/CubeSatSim/tree/master/stempayload/STEM_Payload_Test

The green and blue LEDs should blink alternately, as shown here

https://countingfromzero.net/amsat/stemv1/STEM_Payload_Test.MOV

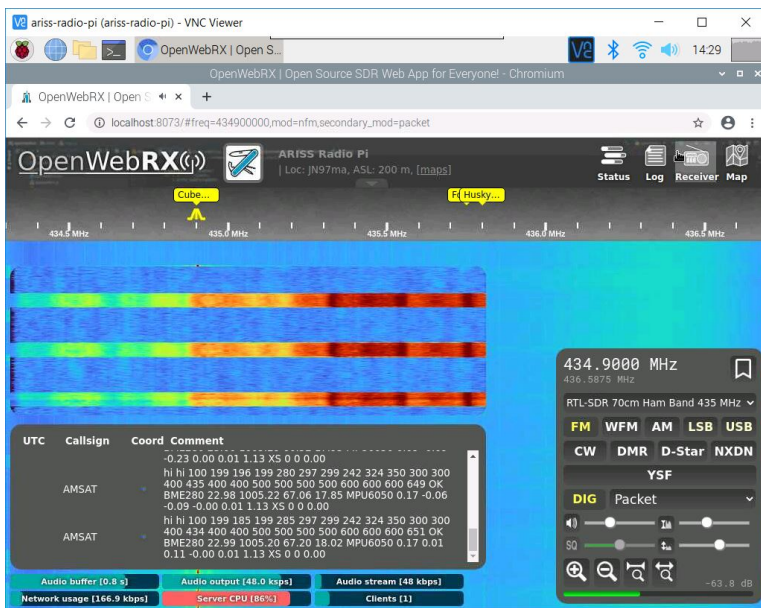
You will then need to re-upload the Payload OK sketch:

https://github.com/alanjohnston/CubeSatSim/tree/master/stempayload/Payload_BME280_MPU6050_XS

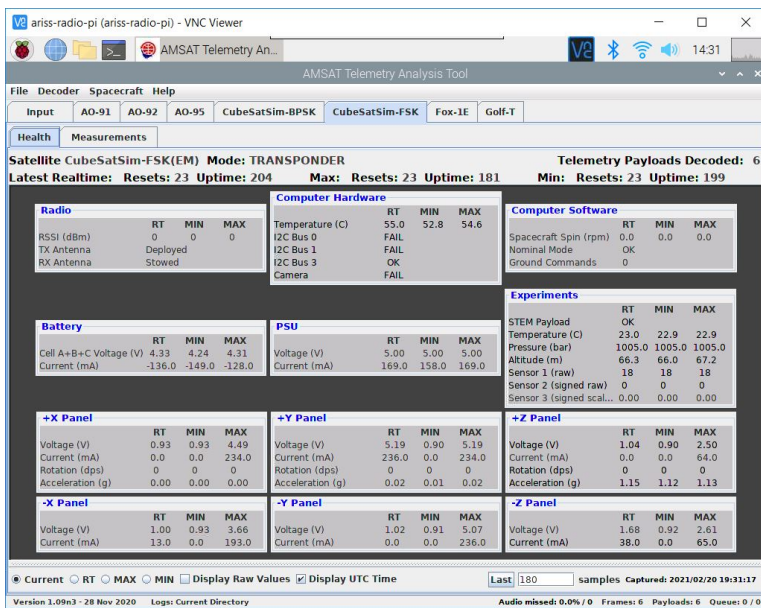
When the STEM Payload board is plugged into the the three board stack, the Pi will read the sensor data over the UART and report the data in telemetry. When the CubeSatSim software is running, you should see the built-in LED on the Pro Micro or STM32 blink every 4 seconds or so as the Pi reads the telemetry data.



In APRS (AFSK 1200) mode, the sensor string will be added to the end of the data.



In FoxTelem, in DUV/FSK mode, under the CubeSatSim-FSK tab Health tab, Experiment box, you should see STEM Payload OK and numbers for the other fields. In addition, the rotation and acceleration will also be displayed under the +X, +Y, and +Z boxes.



Using the pushbutton, you can switch the CubeSatSim to BPSK mode by pressing and holding the push button until it blinks rapidly three times. Then, if you run FoxTelem in BPSK mode, you can look under the CubeSatSim-BPSK tab and see:

The screenshot shows the AMSAT Telemetry Analysis Tool interface. At the top, it displays 'ariss-radio-pi (ariss-radio-pi) - VNC Viewer' and 'AMSAT Telemetry Analysis Tool'. The main window is titled 'AMSAT Telemetry Analysis Tool' and has a menu bar with 'File', 'Decoder', 'Spacecraft', and 'Help'. Below the menu bar, there are tabs for 'Input', 'AO-91', 'AO-92', 'AO-95', 'CubeSatSim-BPSK', 'CubeSatSim-FSK', 'Fox-1E', and 'Golf-T'. The 'CubeSatSim-BPSK' tab is selected. The interface shows 'Satellite CubeSatSim-BPSK(EM) Mode: TRANSPONDER' and 'Telemetry Payloads Decoded: 1'. Below this, there are several data tables:

Radio				Computer Hardware				Computer Software			
	RT	MIN	MAX		RT	MIN	MAX		RT	MIN	MAX
RSSI (dBm)	0	0000	0000	Temperature (C)	37.8	0000	0000	Spacecraft Spin (rpm)	0.0	0000	0000
TX Antenna	Deployed			I2C Bus 0	FAIL			Nominal Mode	OK		
RX Antenna	Stowed			I2C Bus 1	FAIL			Ground Commands	0		
				I2C Bus 3	OK						
				Camera	FAIL						

Other tables include Battery, PSU, Experiments, and various panels (+X, +Y, +Z, -X, -Y, -Z) showing voltage, current, rotation, and acceleration data. The interface also has a status bar at the bottom with version information and logs.

On the spinning turntable, you should see 7-10 degrees per second of rotation and the blue LED on the STEM Payload board illuminated. If the CubeSatSim is accelerated greater than 1.2g, the green LED will illuminate on the STEM Payload board.

Payload Serial Connection Troubleshooting

If you get the right output from typing a ? the Serial Monitor in the Arduino IDE, but you don't see the payload data in the APRS packet or FoxTelem says STEM Payload FAIL, here are some things you can check.

Login to your Pi and type these commands:

```
cd
CubeSatSim/log | grep ayload
```

The use of `ayload` instead of `payload` will give you the log entries for `Payload` and `payload`.

You can also send commands over the serial port like you did with the Arduino IDE using these commands:


```
sudo apt-get install -y minicom
```

```
sudo systemctl stop cubesatsim
```

```
minicom -b 115200 -o -D /dev/serial0
```

Now, you can type ? or R and you should see the same output you saw in the Arduino Serial Monitor.

To exit, you have to type Control-A then z then x then hit Return.

If you suspect your Pi serial port isn't working, you can unplug your Pro Micro and put a jumper between pins 1 and 2 to make a transmit-to-receive loopback. Now when you run minicom, you should see every character you type echoed back to you.

8.3 STM32 Steps

The following steps are specific to the STM32 - you do not need to do them if you are using the Sparkfun Pro Micro.

Flashing the USB Bootloader

These instructions are for configuring the STM32F103C8T6 "blue pill" in the STEM Payload Board so that it can be programmed using the Arduino IDE. Other programming methods and IDEs can be used as well. For the STM32F411CCU6 "black pill", I have successfully followed these instructions: https://www.sgbotic.com/index.php?dispatch=pages.view&page_id=49 Before you can program the STM32, you need to flash a bootloader in it so you can push code using the micro USB interface.

This section describes how to flash a USB bootloader into the STM 32.

Follow the instructions

<https://alselectro.wordpress.com/tag/stm32-usb-upload/> For the Serial USB-TTL adapter, I used the Moyina adapter

<https://www.amazon.com/gp/product/B075N82CDL/>

Some of the lower cost ones just don't work.

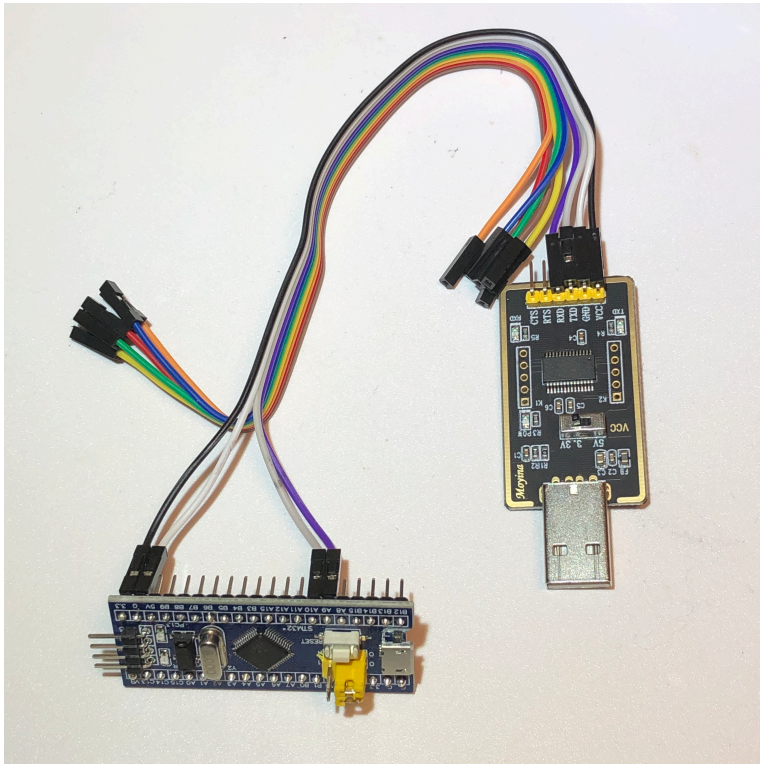
Wired up the STM32 to the USB-TTL adapter set to 3.3V

PA9 Tx to Rx of USB-TTL

PA10 Rx to Tx of USB-TTL

3v3 to 3v3 , GND to GND

This is shown here:



Download the programming tool

<https://www.st.com/en/development-tools/flasher-stm32.html> You will need to sign up and receive email before you get a link to download. Download and install.

Download the firmware generic_boot20_pc13.bin
https://github.com/rogerclarkmelbourne/STM32duino-bootloader/raw/master/binaries/generic_boot20_pc13.bin I used this version since the LED on my STM32 board is labeled PC13.

Plug USB-TTL in to USB of computer.

Set jumper Boot 0 to position 1 (closest to edge).

Run flash_loader_demo_v2.8.0.exe to install app in
Program Files (x86)/STMicroelectronics/Software
STMFlashLoader Demo.exe

Select a Port say Next. Get Target is readable
message. Select Next.

2nd option is Download to device/Download from
file. Click on the box to get the menu. Change file
type to *.bin and select the .bin file downloaded
earlier.

Select Next and it will start and you should see a
green "Download operation completed
successfully" message.

BEFORE YOU UNPLUG ANYTHING, move the
jumper back to position 0.

Now, when you plug in the STM32 using the micro
USB connector, it will show up as "Maple Mini". If
you run the Arduino IDE, select the right port for the
STM32 and open the Serial Monitor set to 9600
baud, you will see this message displayed:

```
Congratulations, you have installed the  
STM32duino bootloader
```

See
<https://github.com/rogerclarkmelbourne/STM32duino-bootloader>

For more information about Arduino on STM32

and <http://www.stm32duino.com>

Arduino Programming IDE

To use the Arduino IDE to program the STM32; plug the STM32 to your computer using a right angle micro USB cable:



Follow the STM32duino installation instructions here:

https://github.com/rogerclarkmelbourne/Arduino_STM32/wiki/Installation

Run the Arduino IDE. The settings under Tools should be:

Board: "Generic STM32F103C series" under STM32F1 Boards (STM32duino.com)

Variant: "STM32F103C8 (20k RAM, 64k Flash)"

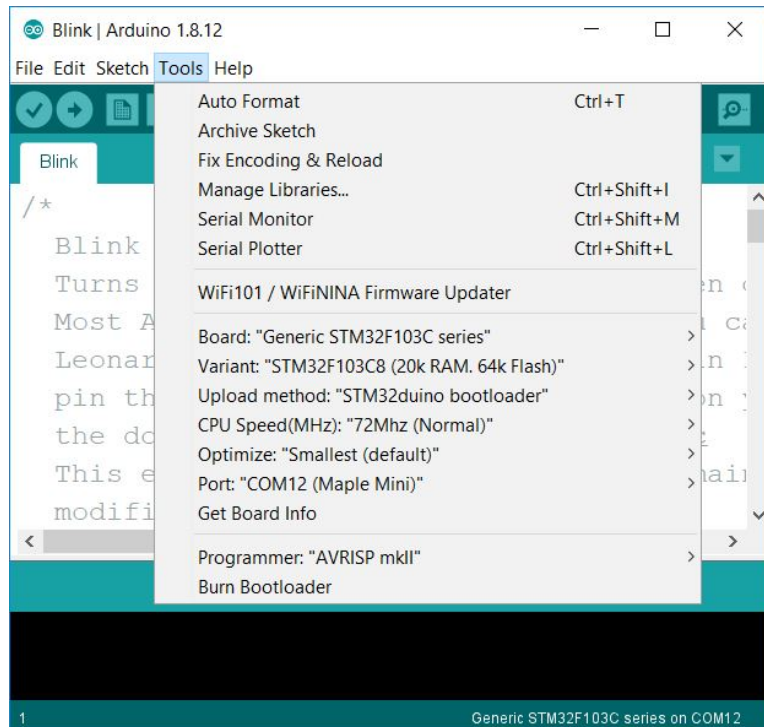
CPU Speed(MHz): "72MHz (Normal)"

Upload method: "STM32duino bootloader"

Optimize: "Smallest (default)"

Port: Some COM port here, labeled Maple Mini should be listed.

It should look like this (although your port number will likely be different):



Open Examples/A_STM32_Examples/Digital/Blink and edit LED pin from PB1 to PC13 and compile. You should see blinking! Note that when I use my Mac, I have to reset the STM32 after each push to get it to run the new software.

If you have soldered in the other board components, you can [test the STM32 and sensors](#).

Additional Sensor Info

The BME-280 Temperature Humidity Barometric Pressure Sensor (small purple board):

<https://lastminuteengineers.com/bme280-arduino-tutorial/>

Here is some code that displays all the values:

<https://github.com/alanbjohnston/CubeSatSim/tree/master/stempayload/bme280test>

The MPU-6050 (GY-521) 3-Axis Accelerometer and Gyro (larger blue board with green LED):

<https://howtomechatronics.com/tutorials/arduino/arduino-and-mpu6050-accelerometer-and-gyroscope-tutorial/>

Here is some code that displays all the values:

<https://github.com/alanbjohnston/CubeSatSim/tree/master/stempayload/GetAllData>

The final step is to [build the board stack and install it in the frame.](#)

+ Add a custom footer