v1.3.2 4. STEM Payload Board 2

Edit New page

Alan Johnston edited this page last week · 33 revisions

Pages 129

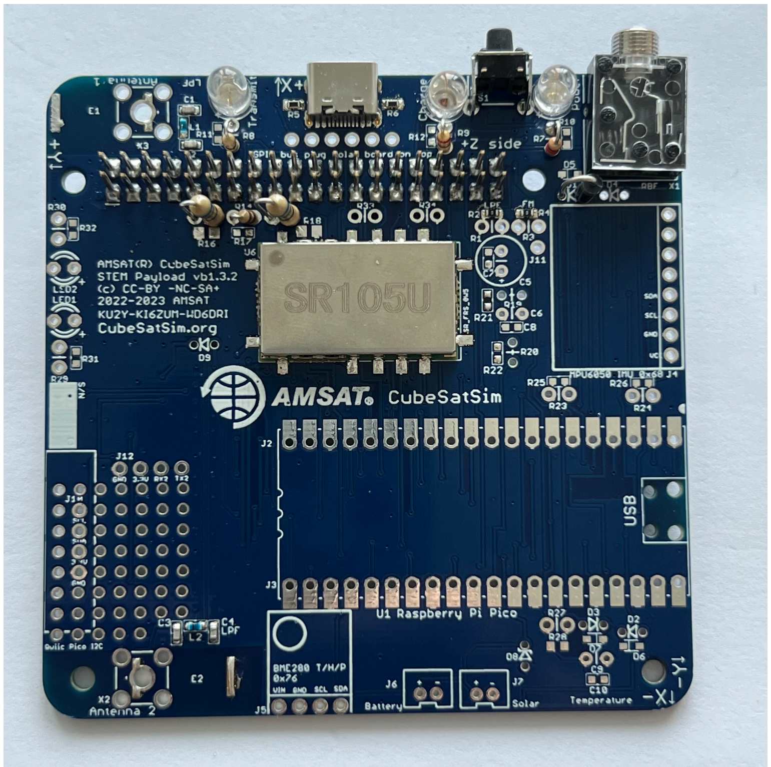
Find a page...

- 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
- 9. Board Stack
- 9. Index
- Creating the CubeSatSim Raspberr...
- CubeSatSim Lite
- CubeSatSim Loaner User Guide
- CubeSatSim Mini

Show 114 more pages...

4. CubeSatSim STEM Payload Board Assembly Part 2

After Part 1, here is the STEM Payload board:



We can now start doing some intermediate tests on the 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 parts)
- Side cutters (to trim leads)
- glue, such as super glue to attach the JST connectors to the PCB

Other tools that are helpful:

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

The first part of this step is testing while the second part continues the assembly

Checklist

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

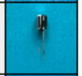
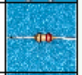
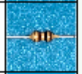
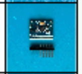
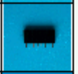
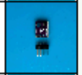
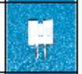
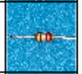
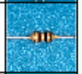
For example, here is the checklist for this step:

+ Add a custom sidebar

Clone this wiki locally

https://github.com/alanbjohnst

☐ Step 4. STEM Payload Board Part 2 https://github.com/alanbjoh

☑	Ref	Item		Qty	Location	Image
☐		STEM Payload Board from Step 1		1		
☐		Pi Zero with SD card from Step 2		1		
☐		USB-C cable and power plug		1		
☐		RBF keychain		1		
☐		3.5mm plug		1		
☐	C5	47uF electrolytic capacitor		1	Top	
☐	C6	100nF capacitor		1	Top	
☐	R20	220 Ohm resistor		1	Top	
☐	R19	100 Ohm resistor		1	Top	
☐	J2, J3	20 pin female socket for Pico		2	Top	
☐	R23,24	4.7k resistor		2	Top	
☐		MPU6050 9 Axis Gyro GY-521		1	Top	
☐	J4	Female 1x8 header		1	Top	
☐		BME280 board Temperature/Humidity/Press		1	Top	
☐	J5	Female 1x4 header		1	Top	
☐	R27	10k resistor		1	Top	
☐	C9	100nF capacitor		1	Top	
☐	D3	1N4148 diode		1	Top	
☐	J6	JST 2.0 connector		1	Top	
☐	LED1	Green LED		1	Top	
☐	R29	1k resistor		1	Top	
☐	LED2	Blue LED		1	Top	
☐	R30	100 Ohm resistor		1	Top	
☐	J10	QWICC connector		1	Top	
☐		1x4 male breakaway header		1	Top	

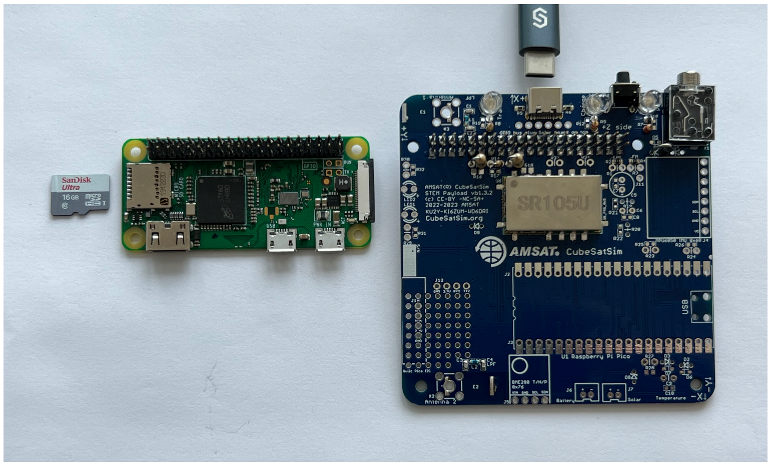
4.1 Testing

Video

Here is a video of the testing part of this step.

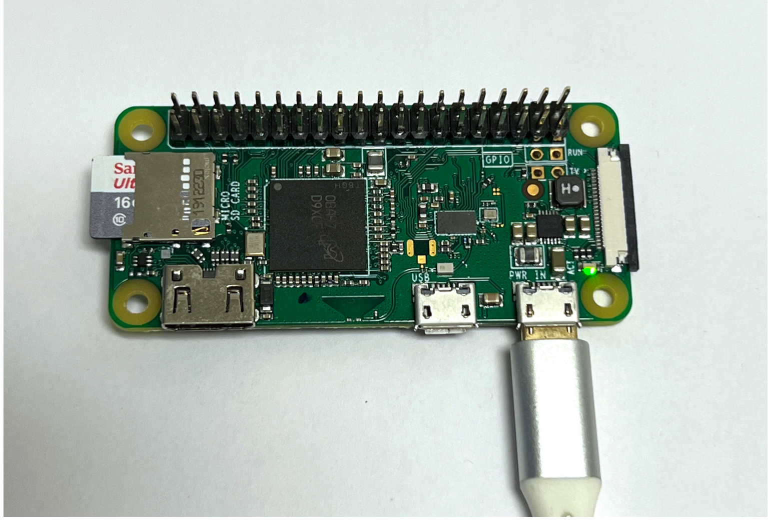
Pi Blink Test

For these tests, you will need your Raspberry Pi Zero W or Pi Zero, a micro SD card with the CubeSatSim software installed (see the Software Install instructions), and the USB-C power cable (or a micro USB power cable or adapter if you power it directly to the Pi).

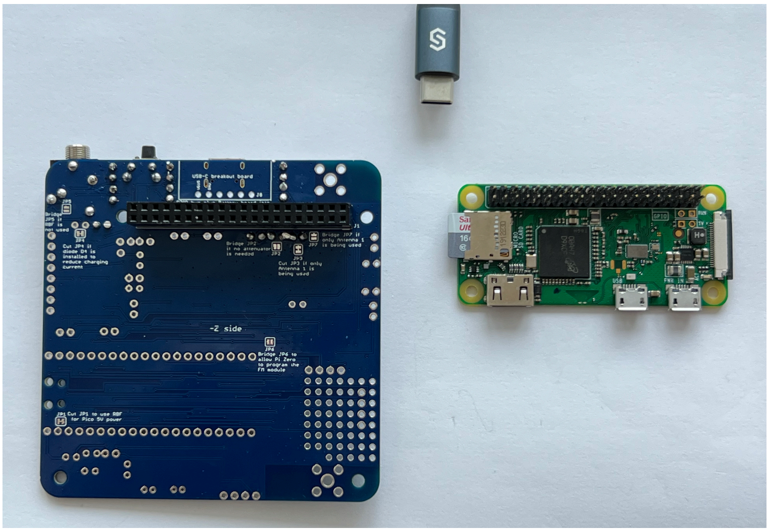


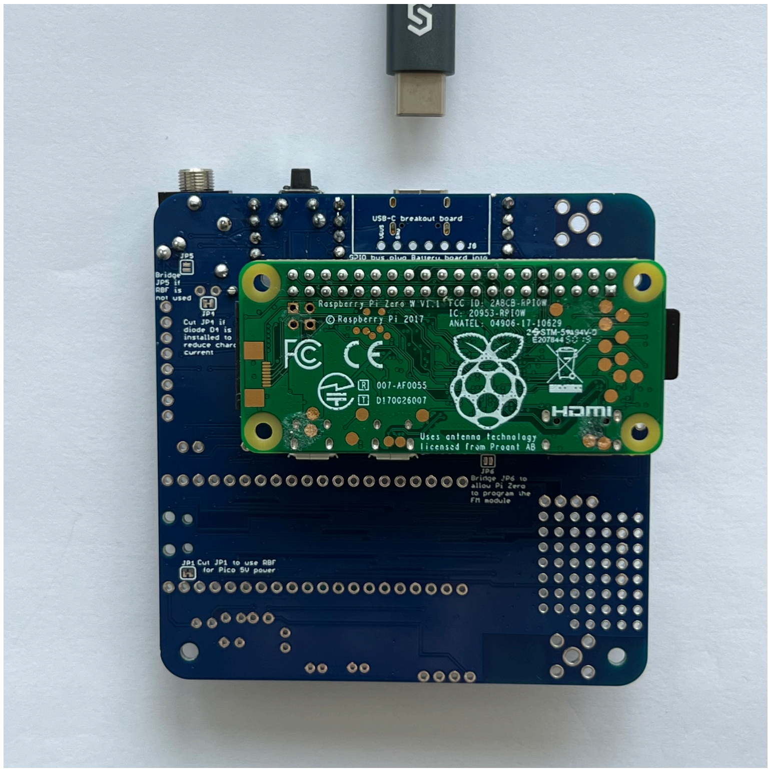
If your Pi Zero W does not have the 2x20 pin header installed, you will need to carefully solder it in. The pins should stick up on the side that has the ICs and the SD card holder.

If you don't have your STEM Payload board, you can still do some testing. Without the STEM Payload Board, you will need to power the Pi Zero with a micro USB cable plugged into the PWR IN micro USB port. When you do so with the programmed micro SD card, you will see the small LED on the Pi Zero board blink:

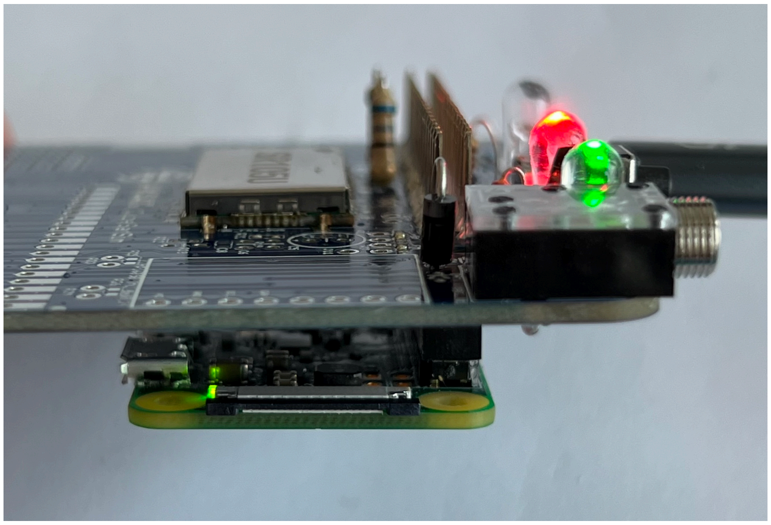


If you have the STEM Payload Board, flip the STEM Payload Board upside down and plug the Pi Zero into the bottom of the board so that it looks like this:



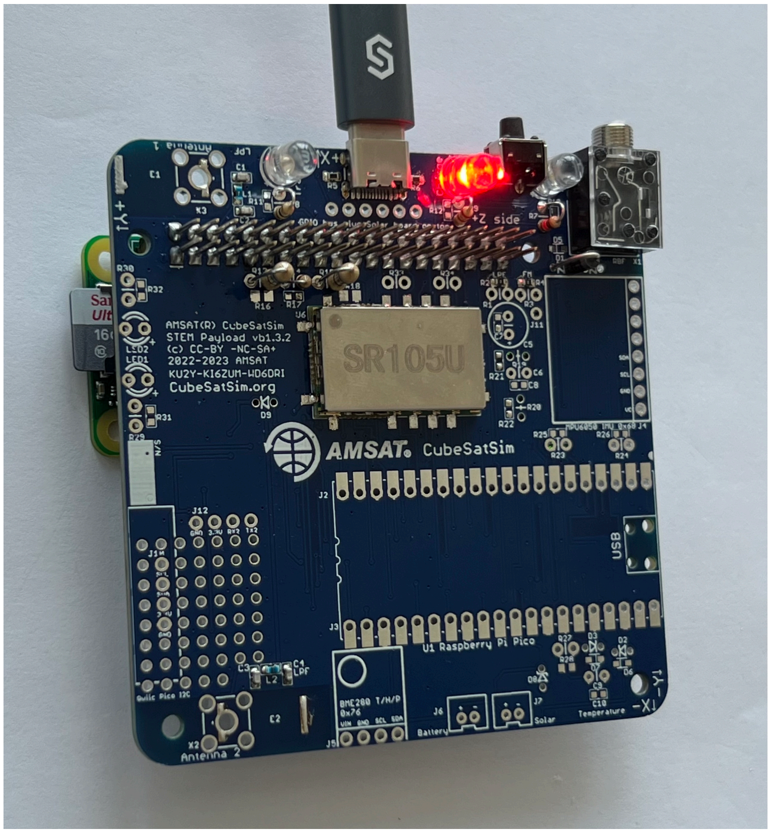

USB-C breakout board
VBUS
GND
J8
GPIO bus plug Battery board info
JP5
Bridge JP5 if RBF is not used
JP4
Cut JP4 if diode D4 is installed to reduce charg current
Raspberry Pi Zero W V1.1
© Raspberry Pi 2017
FCC ID: 2ABCB-RPI0W
IC: 20953-RPI0W
ANATEL: 04906-17-10629
E207844
007-AF0055
D170026007
HDMI
Uses antenna technology licensed from Proant AB
JP6
Bridge JP6 to allow Pi Zero to program the FM module
JP1 Cut JP1 to use RBF for Pico 5V power

Plug your USB-C power cable directly via the PCB USB-C port. The Pi will turn on and the tiny green LED on the Pi will be on or blinking. Note that the LED is on the top of the Pi Zero W board, but you can see it through the lower left mounting hole, or from the side:



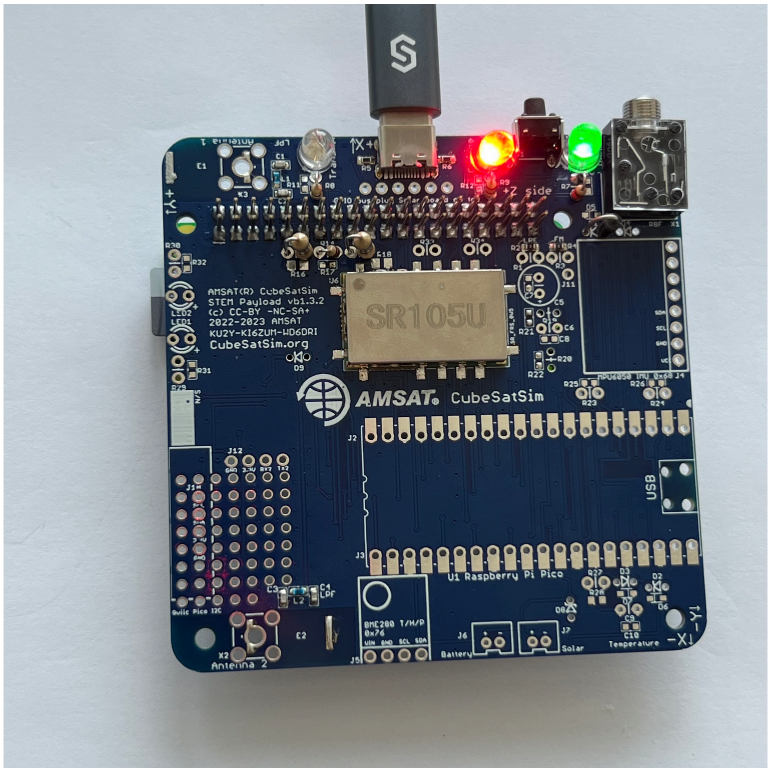
If you don't see this, make sure you have your micro SD card with the CubeSatSim software installed and the USB cable providing power to the Pi.

With the USB-C cable plugged into the board, the Red LED will be illuminated:



Power LED Test

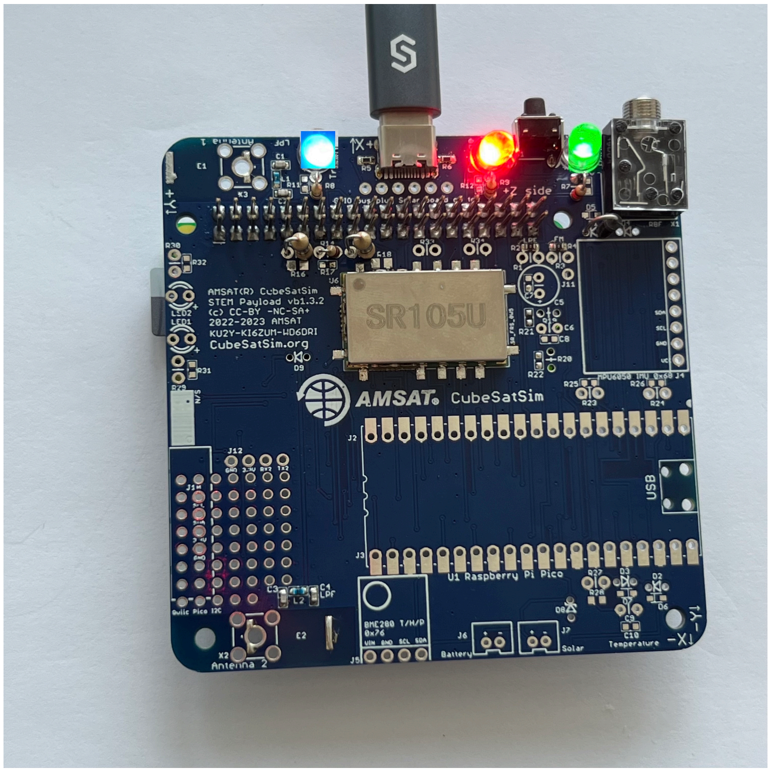
With the Pi plugged into the main board and the power cable plugged into the Pi micro USB port, about 30 seconds after booting, the CubeSatSim software will run and turn on the green Power LED.



If the LED doesn't illuminate, it might mean there is a problem with the green LED LED1 or resistor R11 or that the CubeSatSim software isn't running.

Transmit LED Test

With the Pi plugged into the main board and the power cable plugged into the Pi micro USB port, after booting, the CubeSatSim software will run. When the CubeSatSim is transmitting, the blue Transmit LED will be illuminated, occasionally blinking. This indicates that the CubeSatSim is transmitting.



If the blue LED doesn’t illuminate, it might mean there is a problem with the blue LED LED4 or resistor R8 or the CubeSatSim software. If it only illuminates once, it might mean that the Band-Pass Filter is not detected by the presence of either resistor R1 or R2 - check this resistor.

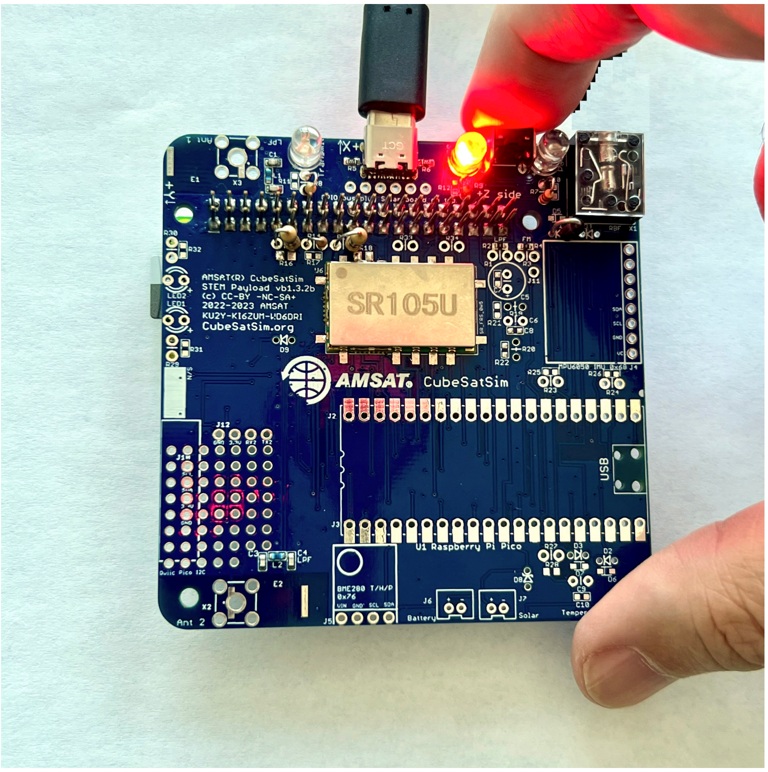
Transmit Test

If the blue LED is illuminated or blinking on and off, you can tune your radio or SDR to 434.9 MHz (+/- 15 kHz) to see and hear a carrier signal. You don't have an antenna yet, but a close by radio will pick it up. If you have your FoxTelem ground station working (the Ground Station instructions are here), you can try to decode frames in FSK mode.

If your blue Transmit LED is illuminated but you don't see a signal, check your antenna on your radio or SDR, and the gain and frequency on your SDR or radio. Or there may be a problem with the CubeSatSim software, in particular the rpitx library.

Push Button Test

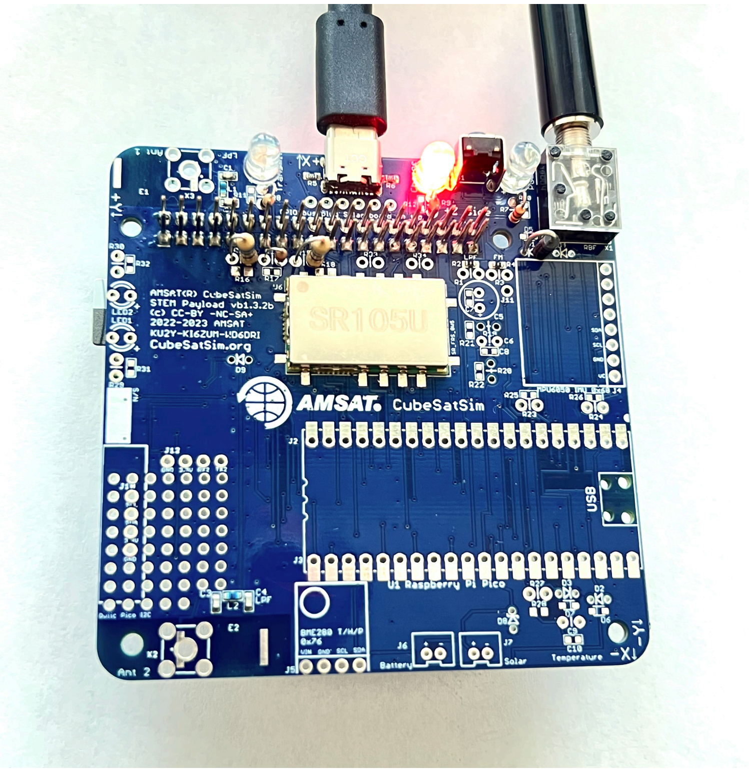
Press and hold the pushbutton. The green LED will blink once, then twice, three times, four times, then five times, then it will blink three times slowly. The Pi Zero computer will shut down and the transmissions and the blue LED will stop blinking.



RBF Switch Test

After completing the Push Button Test, insert the RBF pin. Only the red LED will be illuminated.

Unplug the RBF pin and the Pi Zero will boot up and eventually the green LED will turn on the blue LED will start blinking. Hold down the push button again until the Pi Zero shuts down. Then put the RBF pin back in.



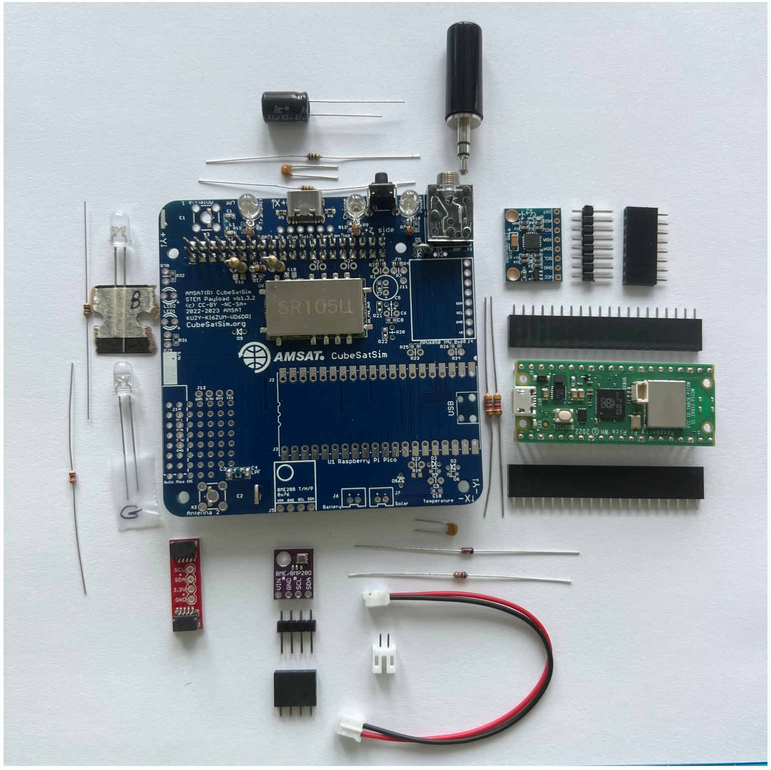
4.2 Build

Video

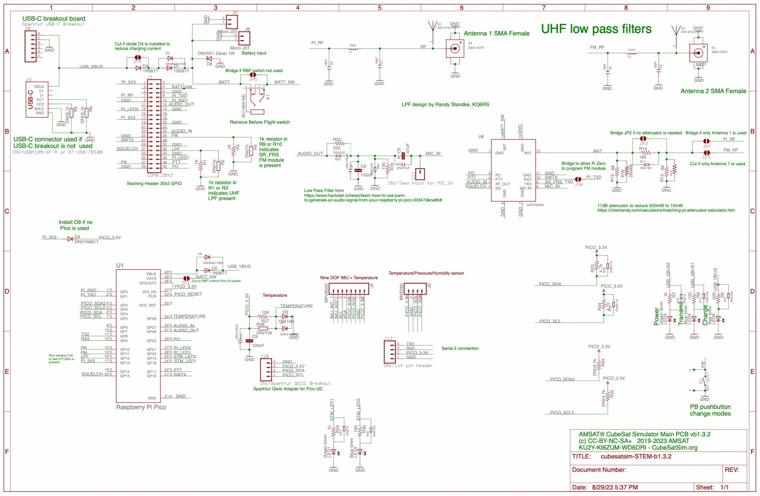
Here is a video of the testing part of this step (you can advance to 8:20 to skip the testing in the previous section).

If all your test results are "nominal" (expected results), you can continue your build after shutting down the Pi as described in the Power Button Test (by holding the push button until the green LED blinks slowly then releasing) and unplugging the Pi from the board.

Here are components installed in this step, as described in the BOM https://docs.google.com/spreadsheets/d/1Ta5UaJcinGozcheROrkfwXdGSDUZrXvQ1_nbIBdIlOY/edit?usp=sharing



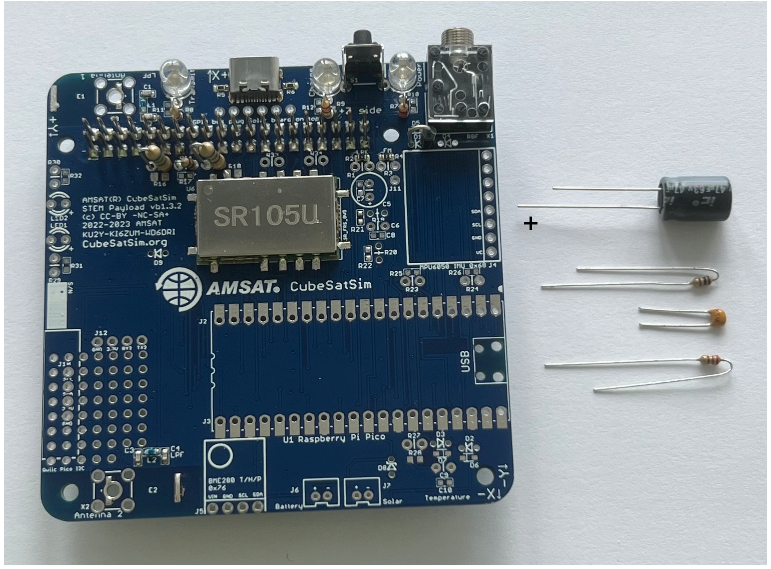
Here is the schematic


USB-C breakout board
UHF low pass filters
Antenna 1 SMA Female
Antenna 2 SMA Female
LPF design by Randy Standke, KQ6RS
Install D9 if no Pico is used
Temperature/Pressure/Humidity sensor
Raspberry Pi Pico
PB pushbutton change modes
AMSAT® CubeSat Simulator Main PCB v1.3.2
(c) CC-BY-NC-SA+ 2019-2023 AMSAT
KU2Y-KI6ZUM-WD6DRI - CubeSatSim.org
TITLE: cubesatsim-STEM-b1.3.2
Document Number:
REV:
Date: 8/29/23 5:37 PM
Sheet: 1/1

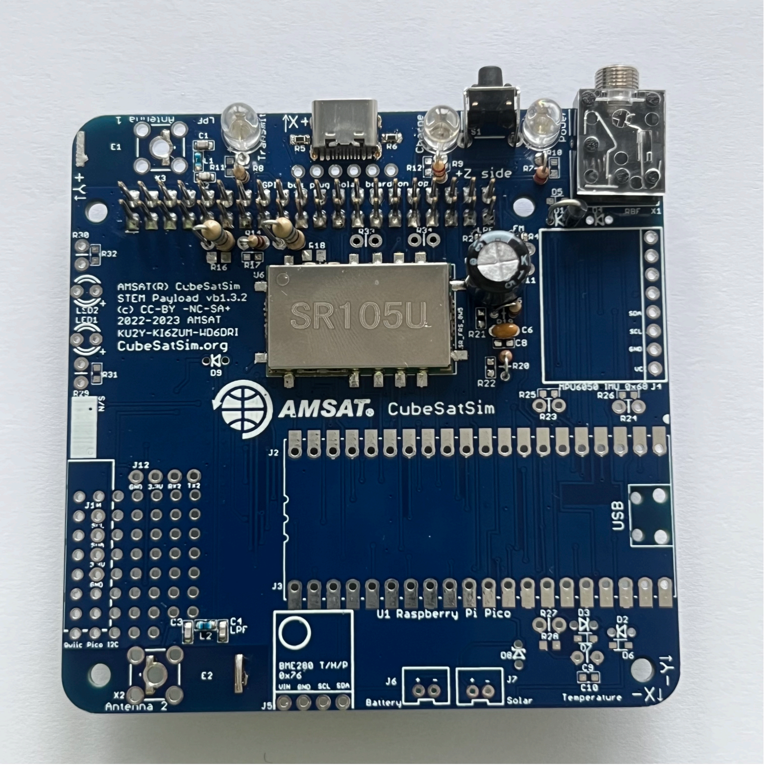
https://github.com/alanbjohnston/CubeSatSim/blob/beta-v1.3.1/hardware/beta-v1.3.2/cubesatsim-STEM-b1.3.2b_schematic.pdf

You should unplug the Pi Zero from the board before continuing with the assembly.

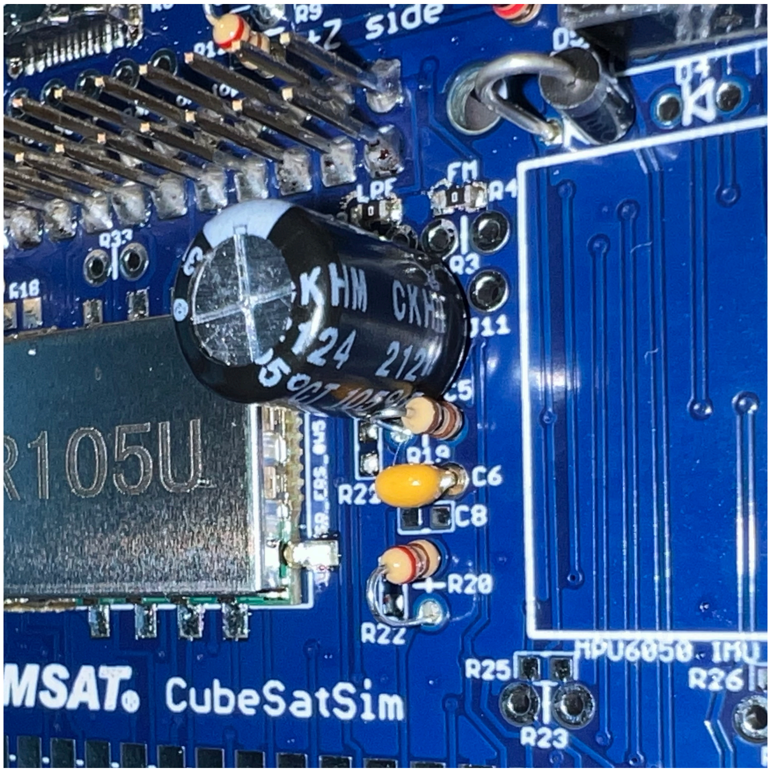
First, install the audio low pass filter that feeds the PWM (Pulse Width Modulation) audio signals into the FM Transceiver module with 4.7uF capacitor C5 (can), 100nF capacitor C6 (small yellow capacitor), 220 Ohm resistor R20 (color bands red red brown) and 100 Ohm resistor R19 (color bands brown black brown):



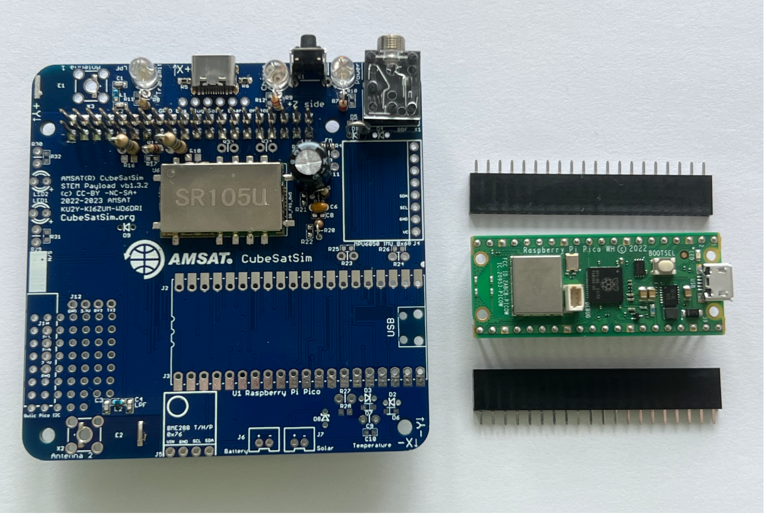
The electrolytic capacitor C5 has polarity - the negative lead (the shorter one in the photo) is marked with a white – . In the photo above and on the PCB, the positive side is marked with a + .



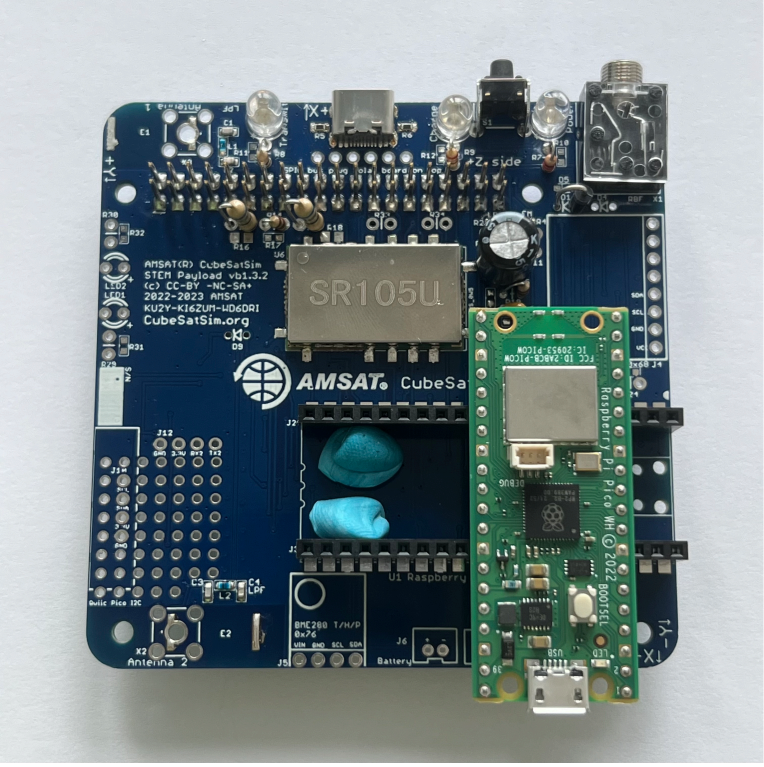
Here is a closeup showing the parts:

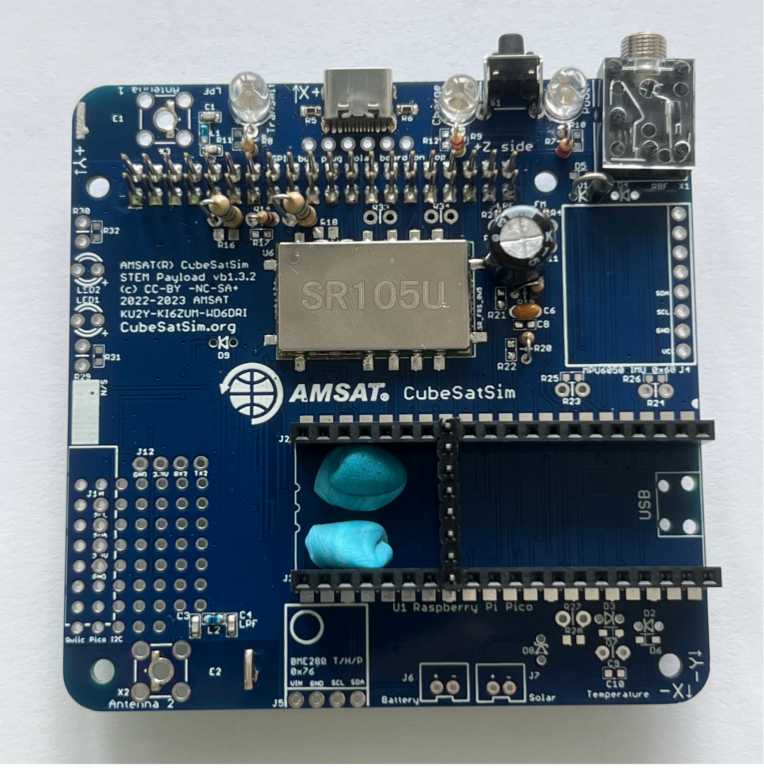


Next, we will install the female sockets J2 and J3 used to mount the Raspberry Pi Pico microcontroller U1. If your Pico doesn't have the male pin headers installed, you will need to solder them in.

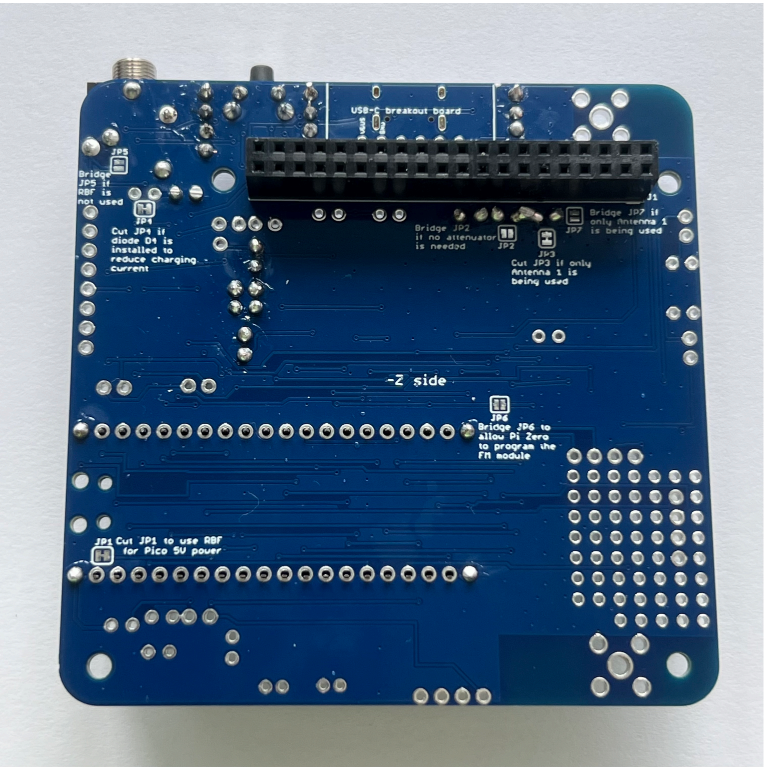


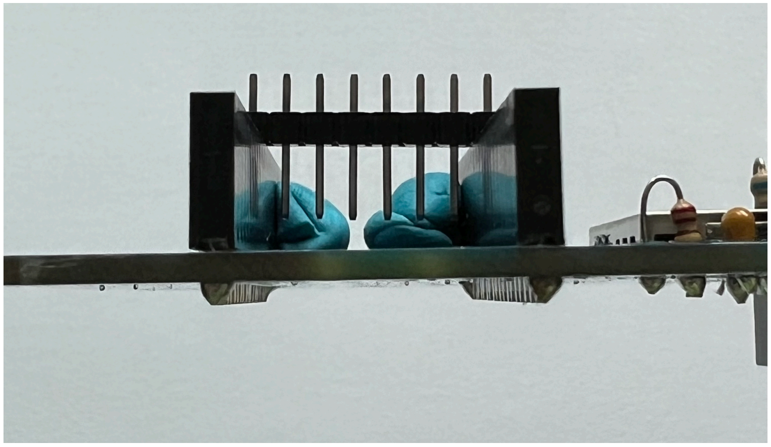
The spacing and alignment of the female sockets is important, otherwise the Pico won't plug in. You can use the Pico itself at right angles to ensure they are spaced correctly and at right angles to the PCB, or you could use the 1x8 pin header from the MPU 6050 board J4.



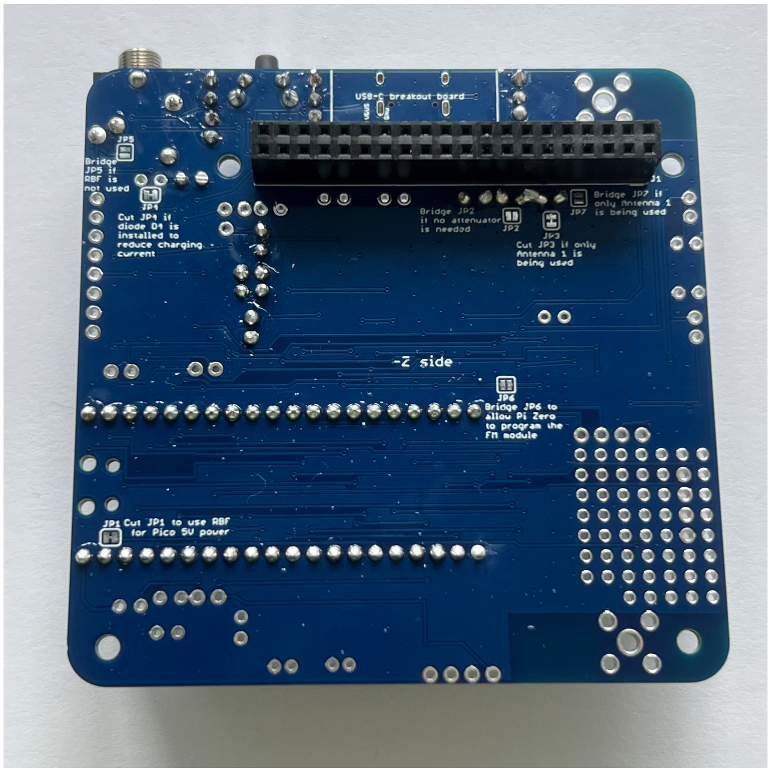


First solder only one pin on each header and double check to make sure it is installed correctly. Look at it sideways to make sure there is no gap - this photo shows a tiny gap which is fine:

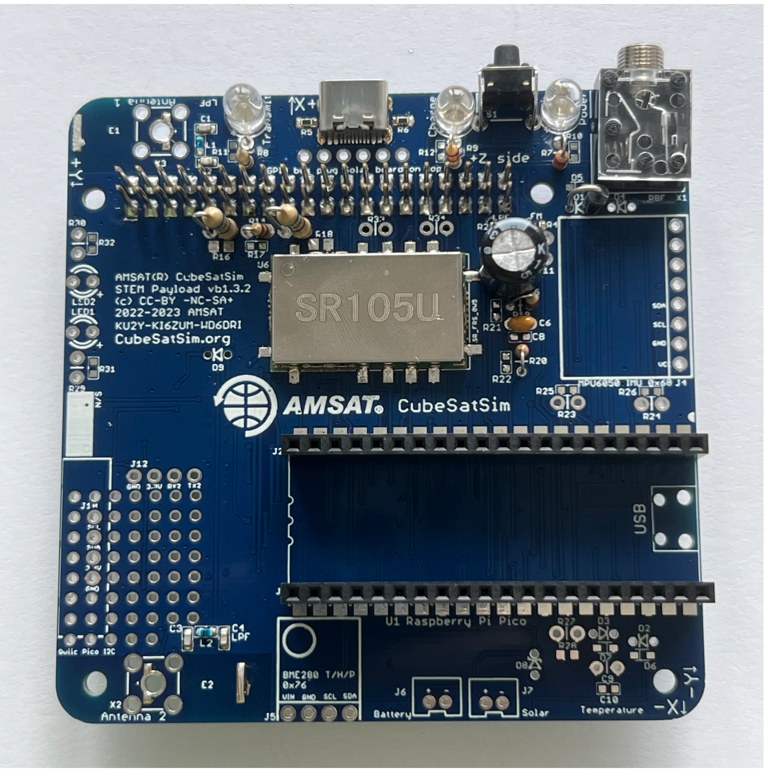

USB-C breakout board
Bridge JP5 if RBF is not used
Cut JP4 if diode D4 is installed to reduce charging current
Bridge JP2 if no attenuator is needed
Cut JP3 if only Antenna 1 is being used
Bridge JP7 if only Antenna 1 is being used
-Z side
Bridge JP6 to allow Pi Zero to program the FM module
Cut JP1 to use RBF for Pico 5V power



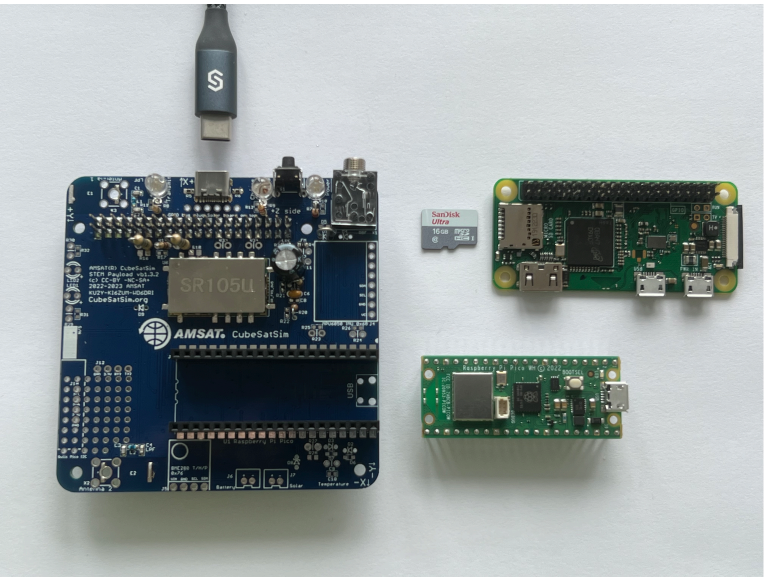
Here is the bottom of the PCB with all the pins soldered:


USB-C breakout board
JP5
Bridge JP5 if RBF is not used
JP4
Cut JP4 if diode D1 is installed to reduce charging current
Bridge JP2 if no attenuator is needed
JP2
J1
Bridge JP7 if only Antenna 1 is being used
JP7
JP3
Cut JP3 if only Antenna 1 is being used
-Z side
JP6
Bridge JP6 to allow Pi Zero to program the FM module
JP1 Cut JP1 to use RBF for Pico 5V power

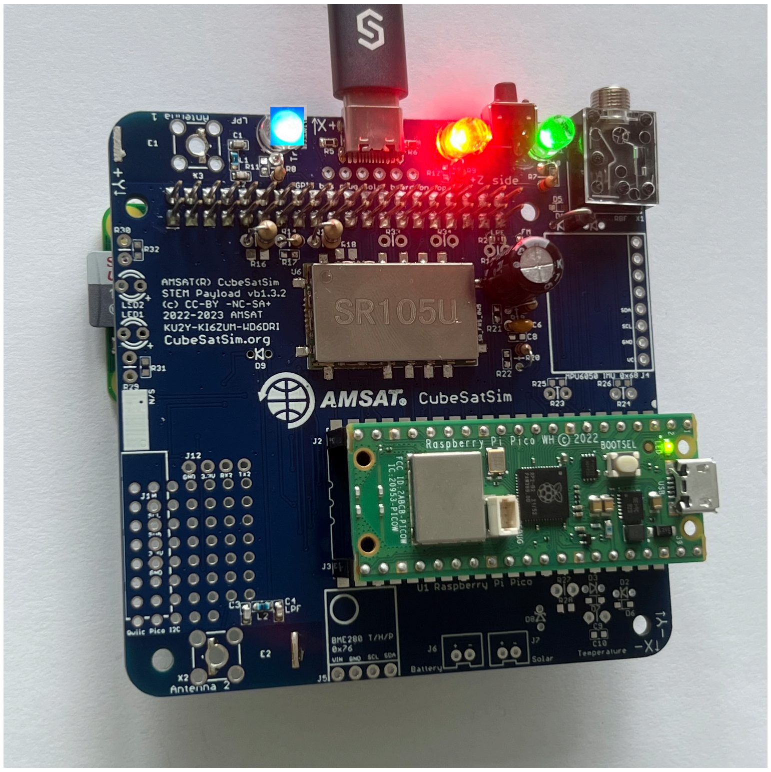
Here is the board with the headers installed ready to plug in the Pico:



The FM modes can now be tested with the help of the Pi Zero WH with a programmed SD card and the Pico which has been flashed with the software. If you haven't completed these steps, you can find them here: Pi Zero and Pico Software Install Instructions



Plug the Pi Zero into the bottom of the PCB and the Pico into the top - make sure the micro USB connector is at the edge of the board as shown below. Plug in the powered USB-C power cable and the Red charging LED should illuminate. The Green Power LED will illuminate after a few seconds. Then the Blue LED will illuminate and go on and off:



If the Pico is programmed and functioning correctly, it should flash once then twice, then repeat the pattern every few seconds.

Push Button Test

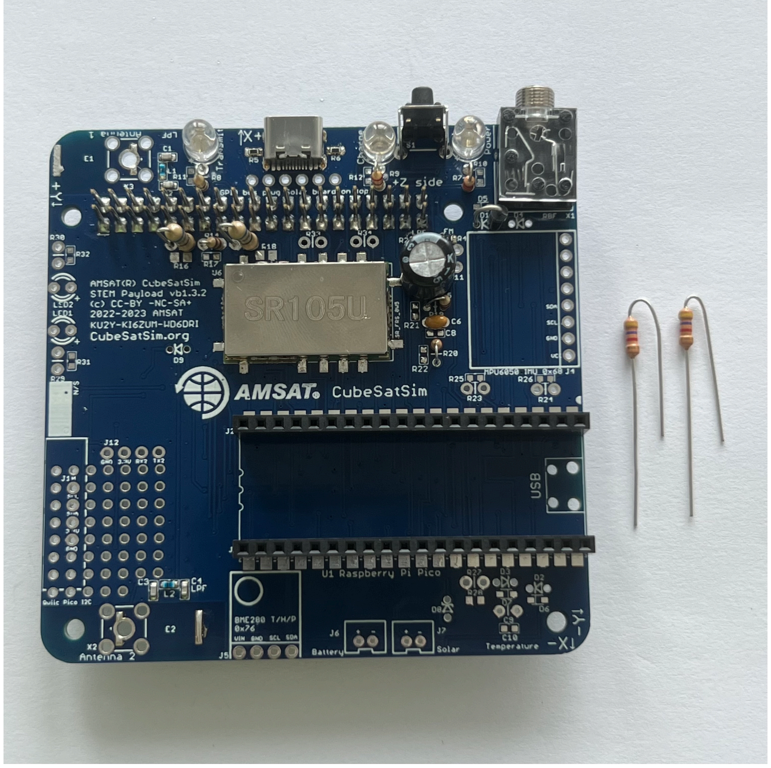
With the Pi plugged into the main board and the power cable plugged into the Pi micro USB port, after booting, the CubeSatSim software will run. When the push button is pressed and held, the green LED should blink rapidly in this sequence: 1 quick blink, 2 quick blinks, 3 quick blinks, 4 quick blinks, 3 slow blinks. Releasing the button after one of the quick blink sequences will put the CubeSatSim in that mode, and the green LED will stop blinking. For example, in this video, the push button is released after the 3 blinks, so the CubeSatSim is in Mode 3.
http://countingfromzero.net/amsat/v1/blink.MOV
The modes are: 1: APRS, 2: FSK, 3: BPSK, 4: SSTV, 5:CW.

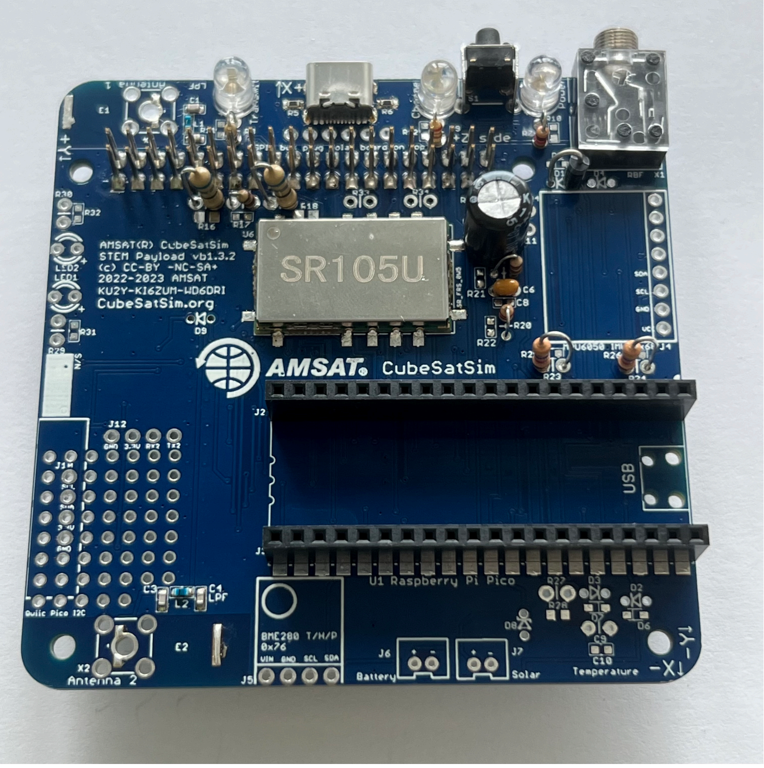
- When pressed and immediately released, the Pi should reboot after about 30 seconds.
- When pressed and held until the green LED starts to blink slowly (see above for the LED blink sequence), the green LED will then blink slowly three times, then the Pi will shutdown (the green Power and blue Transmit LEDs will go out, then the tiny green LED on the Pi will flash rapidly for about 10 seconds then go out). You can release the button as soon as the LED starts blinking slowly. This video shows the button shutting down the Pi: http://countingfromzero.net/amsat/v1/shutdown.MOV After the Pi is shutdown (no LEDs are on or blinking), you can safely disconnect the USB power cable and unplug the Pi from the board to continue the build.
- When the Pi is off, pressing and releasing the push button will startup the Pi, as this video shows: http://countingfromzero.net/amsat/v1/power_up.MOV

If the green Power LED doesn’t blink, there might be a problem with the push button switch S1 or with the pi-power-button software.

If you don't hear anything or see anything for the FM modes (APRS, SSTV, and CW), try listening on 450.000 MHz for the signal. If you hear the signal there, it means the FM Transceiver module hasn't had its frequency set to 434.9 MHz by the Pi Zero software. Try powering down everything and then powering up again.

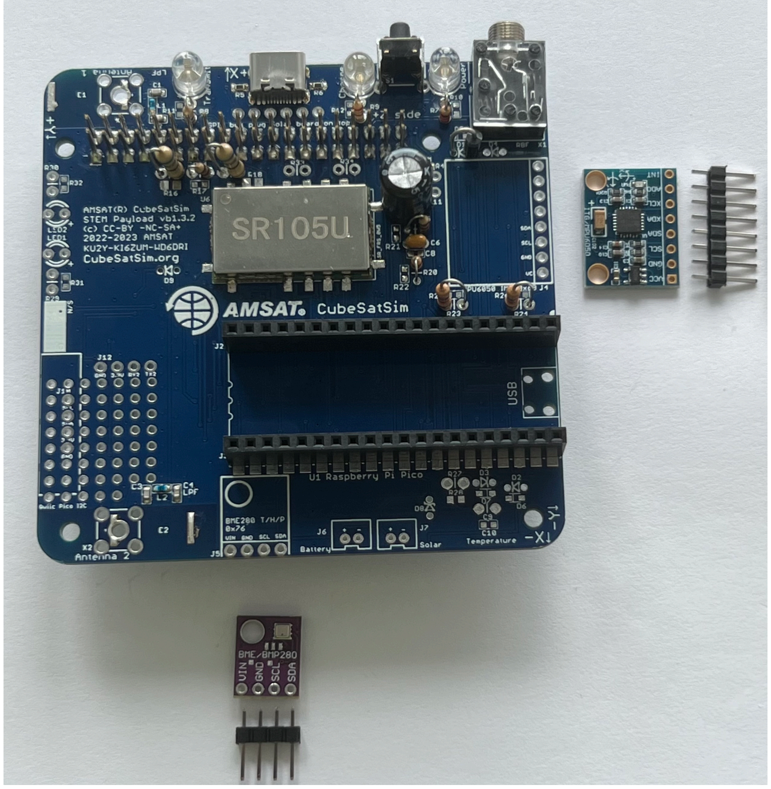
Then mount the two 4.7k Ohm resistors R23 and R24 (yellow violet red color bands):



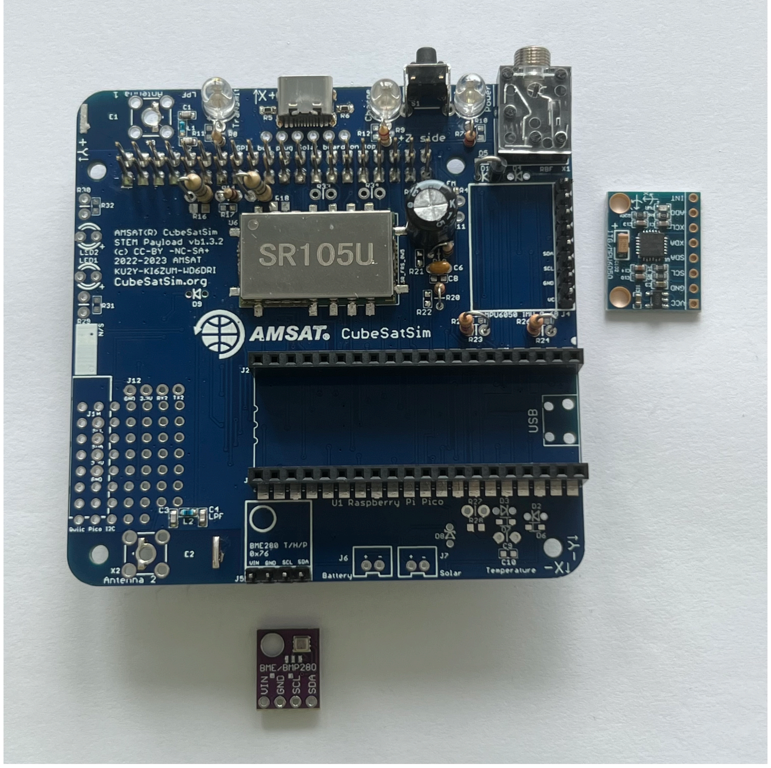


Next we will mount the BME280 sensor (small purple board) J5 and the MPU6050 Gyro and Accelerometer sensor (larger blue board) J4. They can be mounted directly on the board or on female sockets for more flexibility.

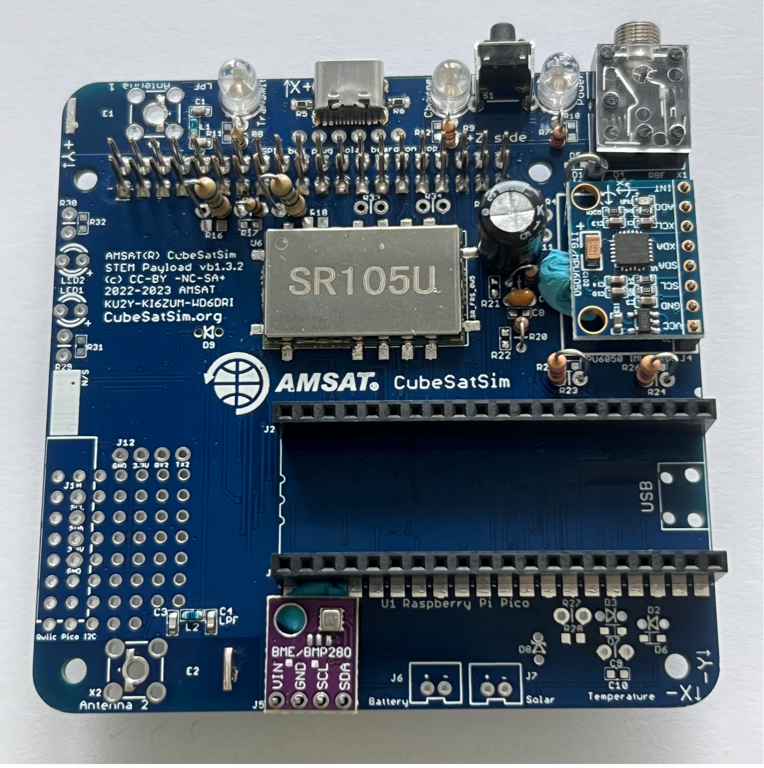
If mounting them directly on the board:



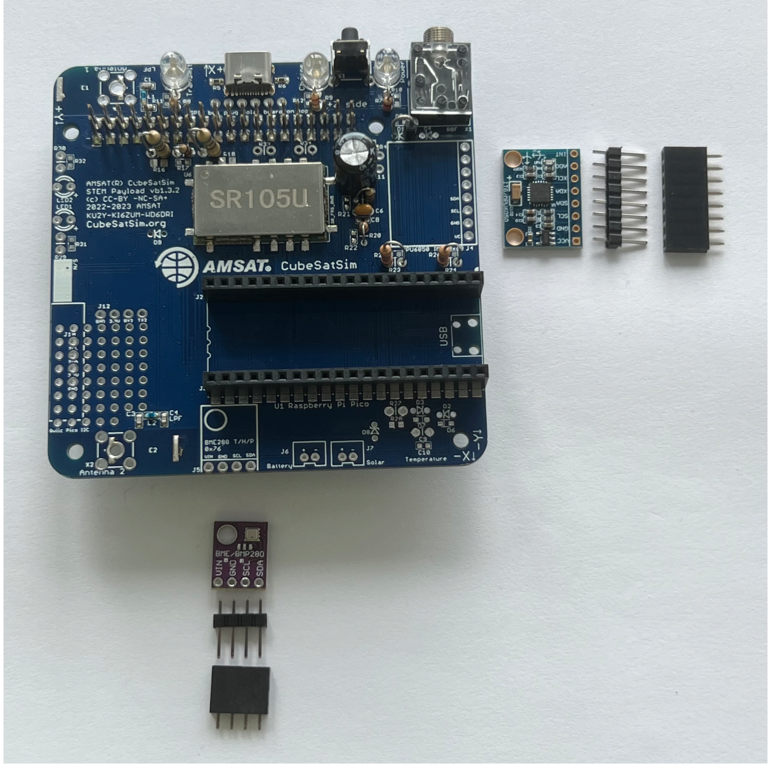
You will need to solder the pin headers directly onto the PCB, long pins down:



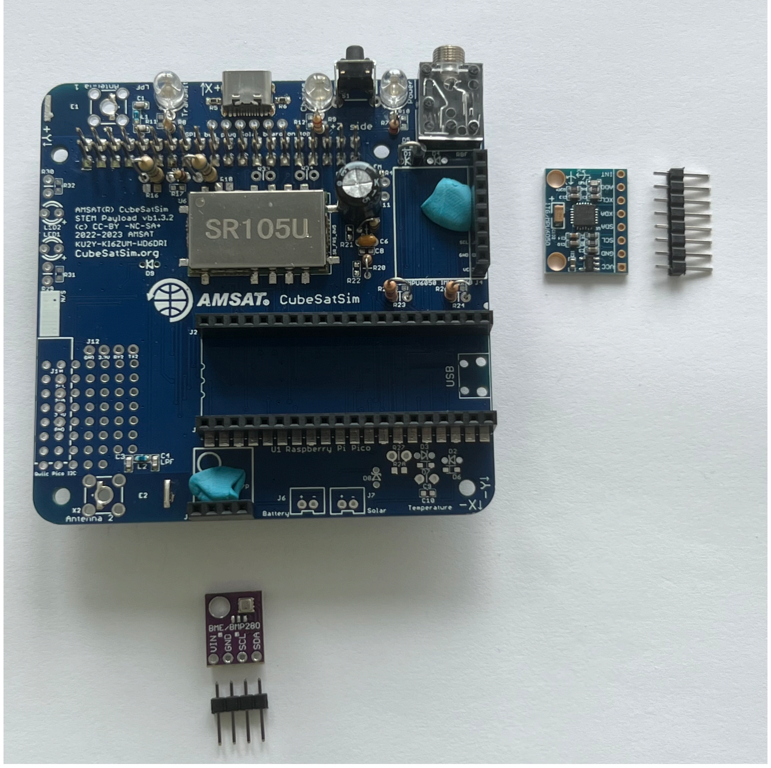
Then, solder the boards to the pin headers while keeping them level with blue putty:



If you are mounting them with female sockets:

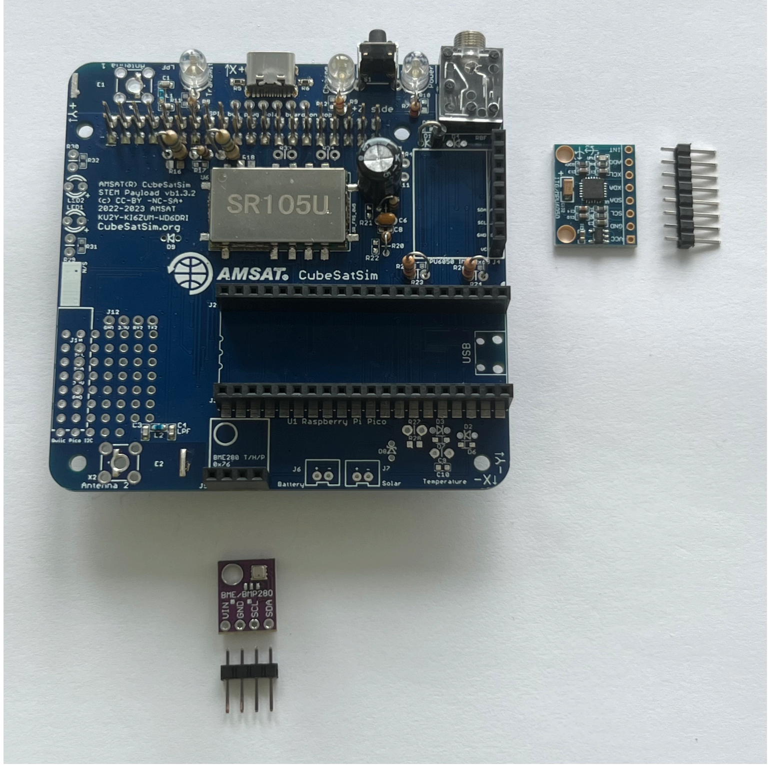


First solder the female sockets, making sure they aren't tilted. Blue putty can be used to hold them in place during soldering.

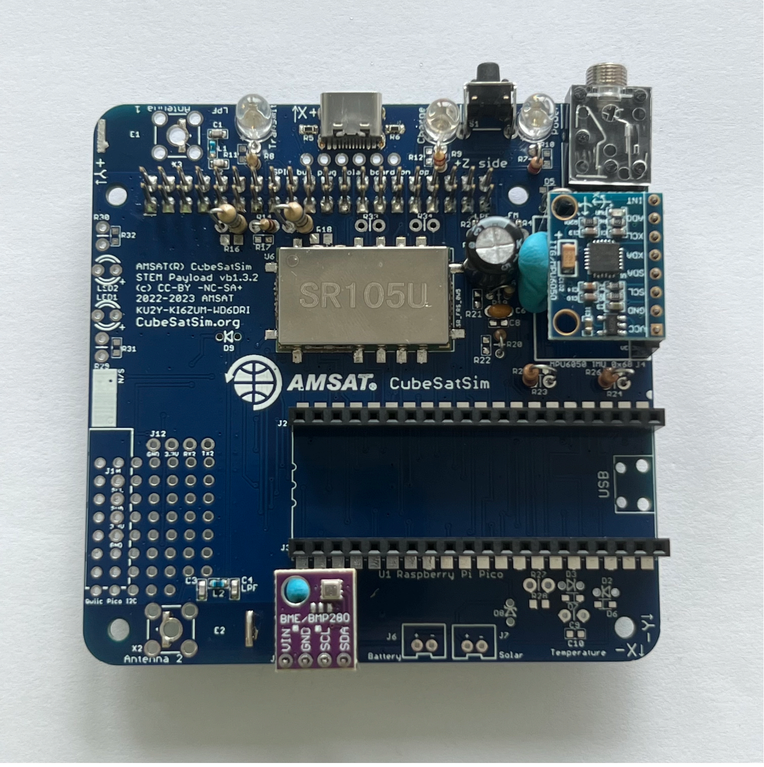


Solder one pin on each socket, then double check.

Here's how they look soldered in:



Insert the pin headers into the sockets, long pins into the sockets, then place the sensor boards on top. Blue putty can be used to hold them horizontal to the board.

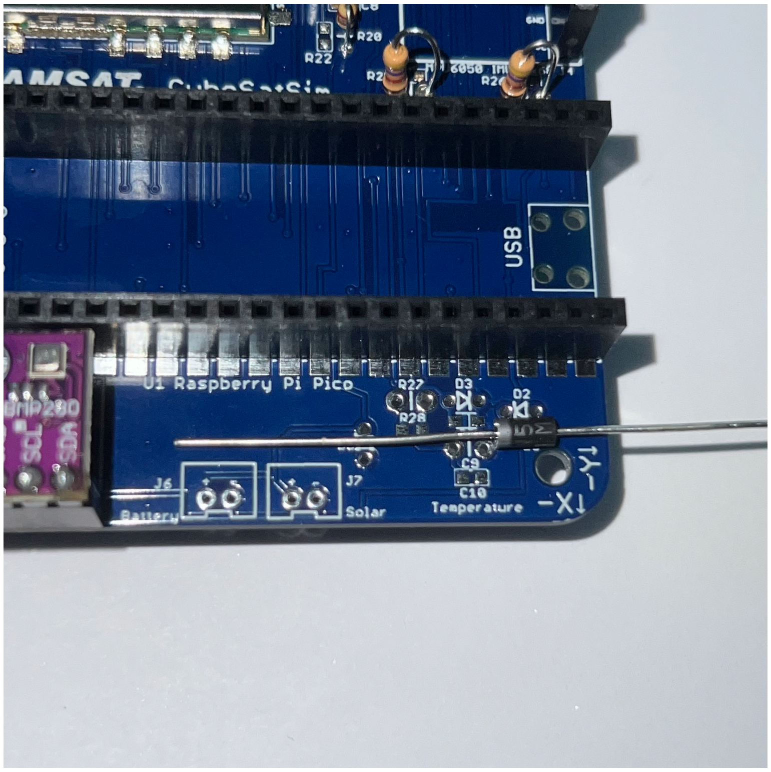


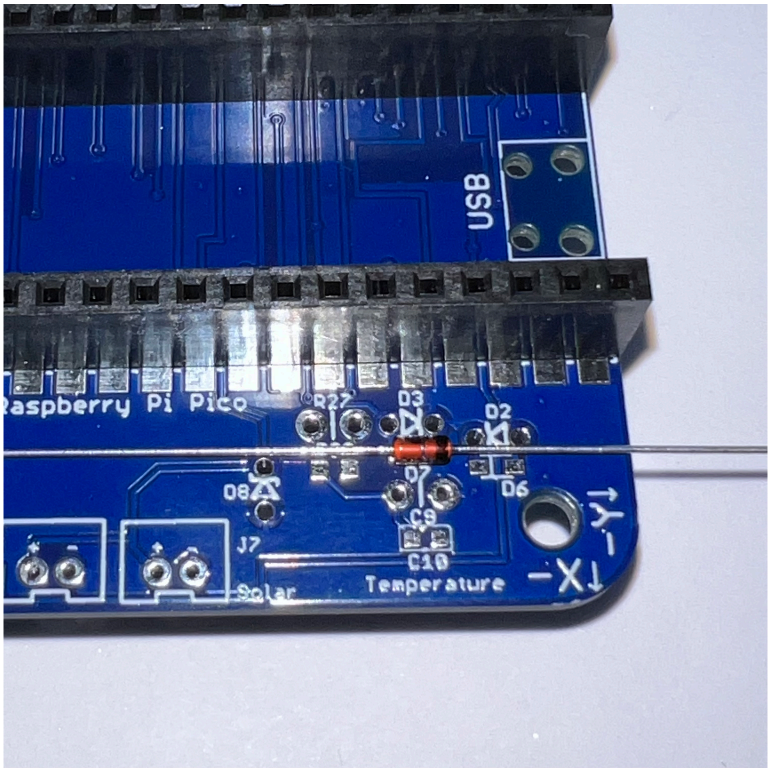
IMPORTANT: make sure the purple BME280 is not upside down. The top has the BME sensor chip (small metal can). The circular hole in the board should be to the left.

Next, solder these four components on the bottom right of the PCB:

- 10k resistor (black brown orange color bands) R27
- 100nF capacitor C9 (small yellow capacitor)
- 1N5817 diode D2 (black device with a white band on one side to mark polarity)
- 1N4148 diode D3 (has a glass case with the numbers "41" and "48" written on it and a black band to indicate polarity

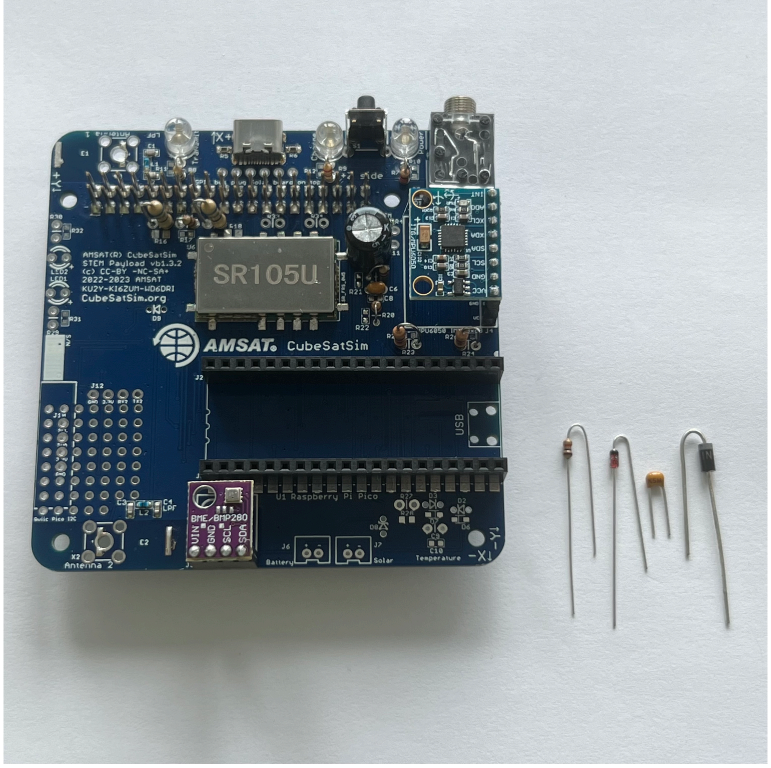
Here's the polarity of the diodes:





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 is correct before soldering them.

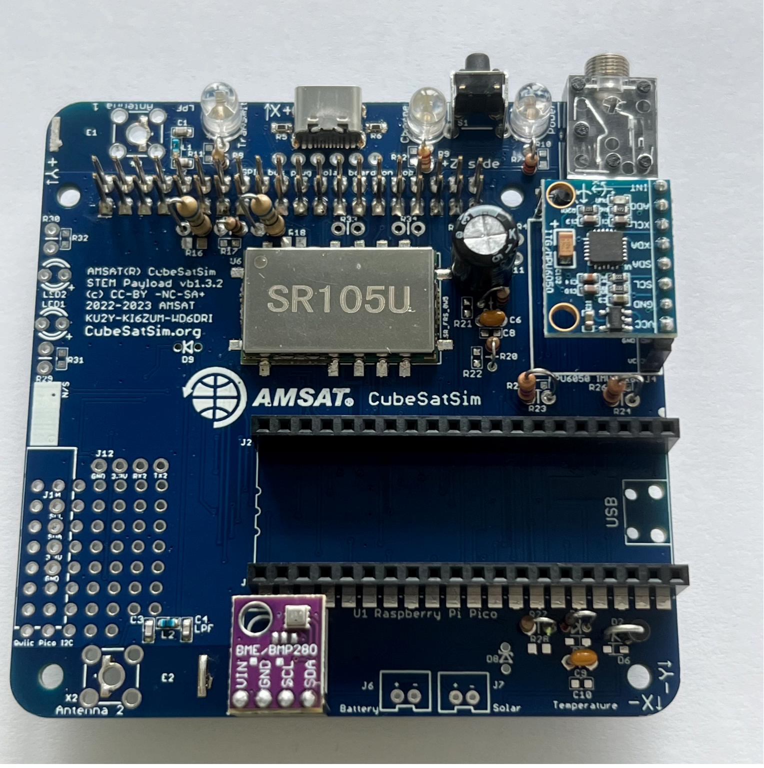
Here's all four parts ready to install:



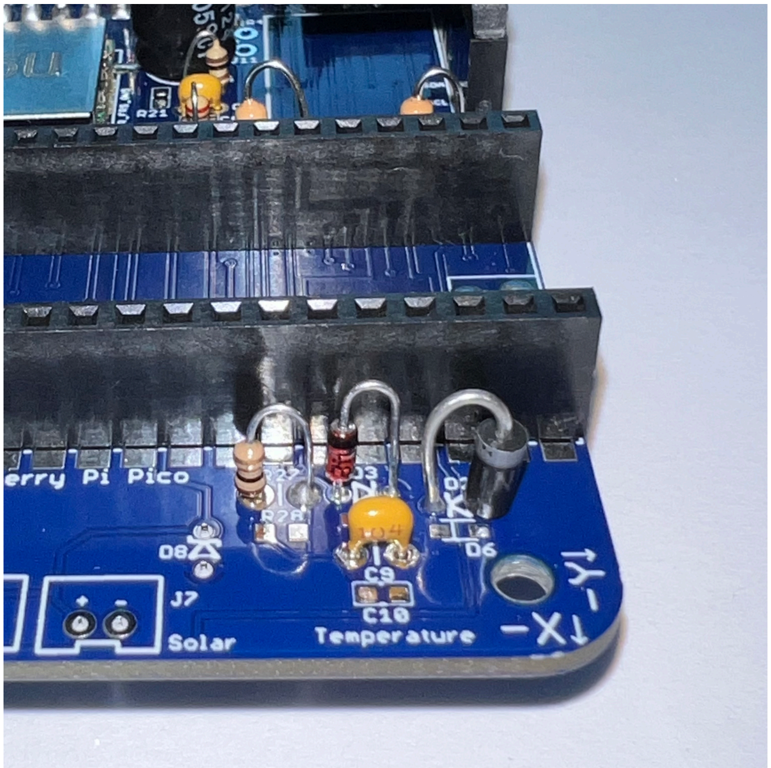
Solder the leads on the bottom of the PCB.

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

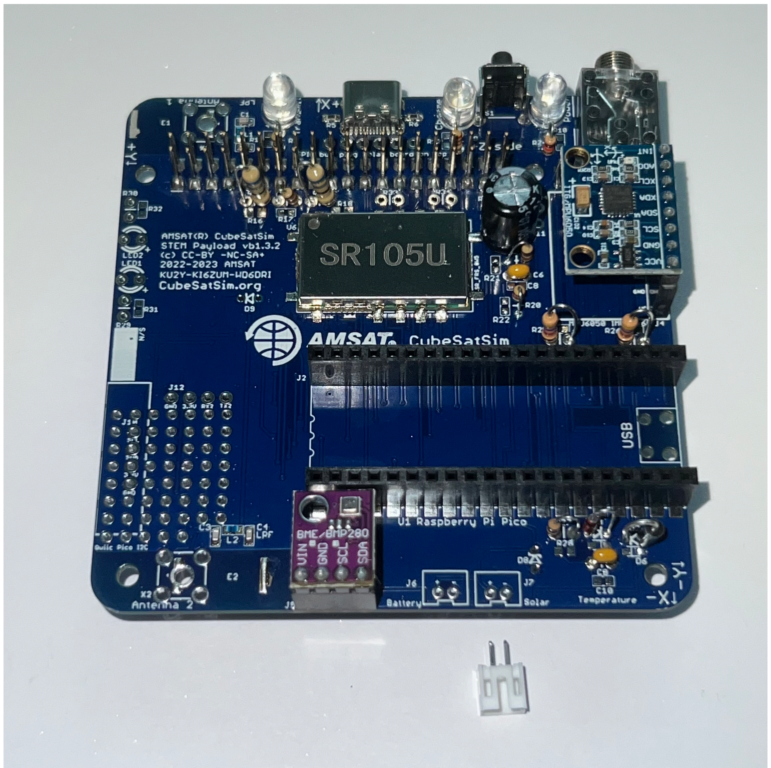
Here's after they are installed:



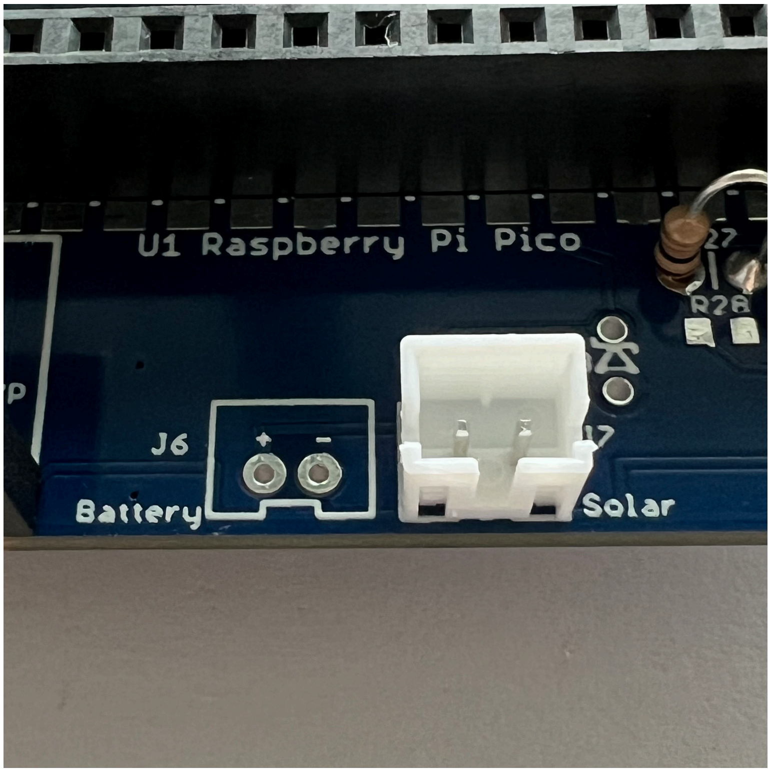
Here is a closeup:



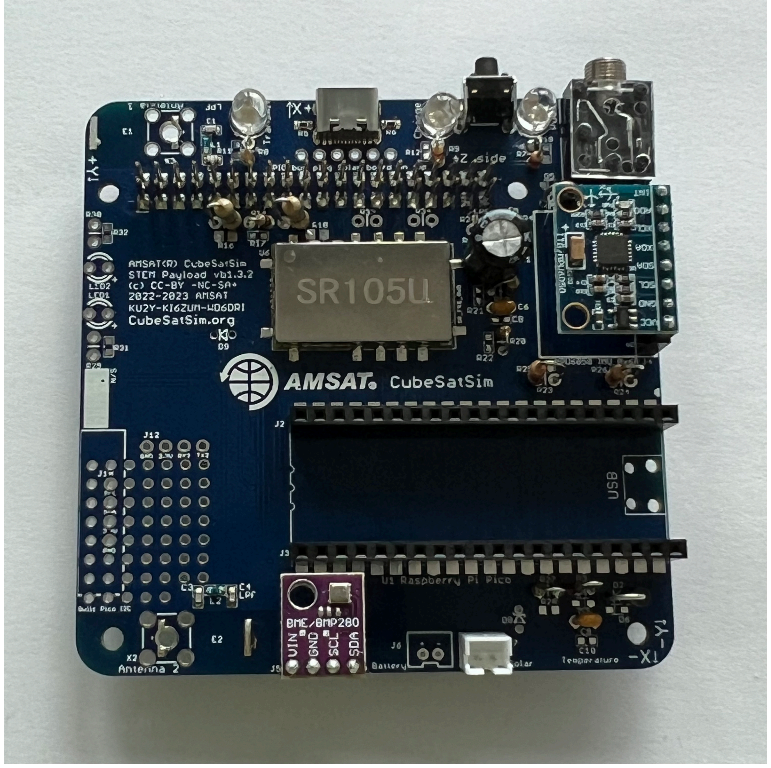
Install the JST connector J7 which is labeled "Solar" since the JST jumper cable will connect it to the Solar Board. The other JST connector J6 labeled "Battery" is used if no solar panels are used and the Battery Board will be connected here.



Here is a closeup showing that the slot should face the edge of the PCB for correct polarity. Some JST connectors snap into the board. If yours do not, then I recommend a drop of super glue gel or other adhesive prior to soldering:

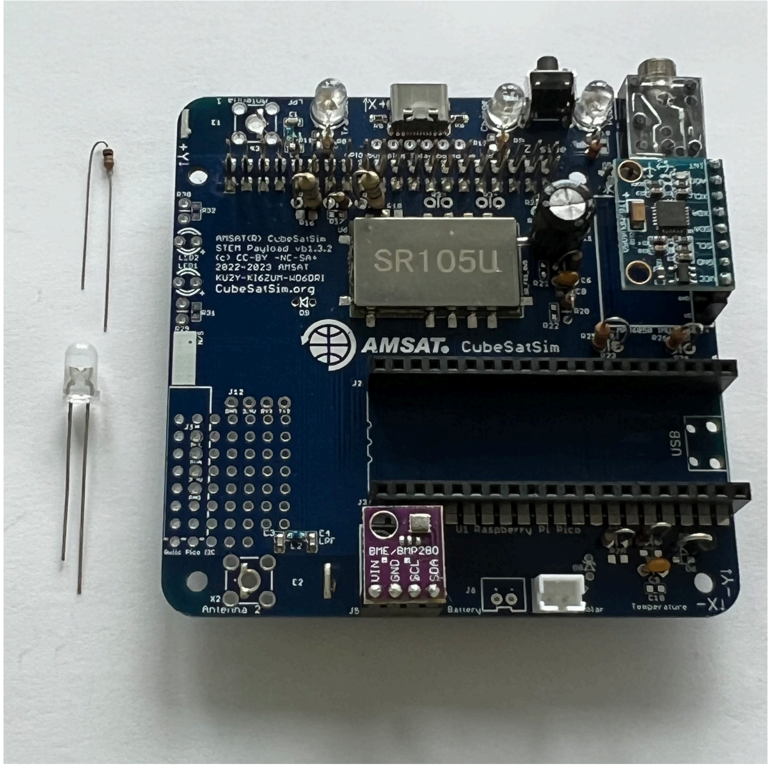


Here's how it looks installed:

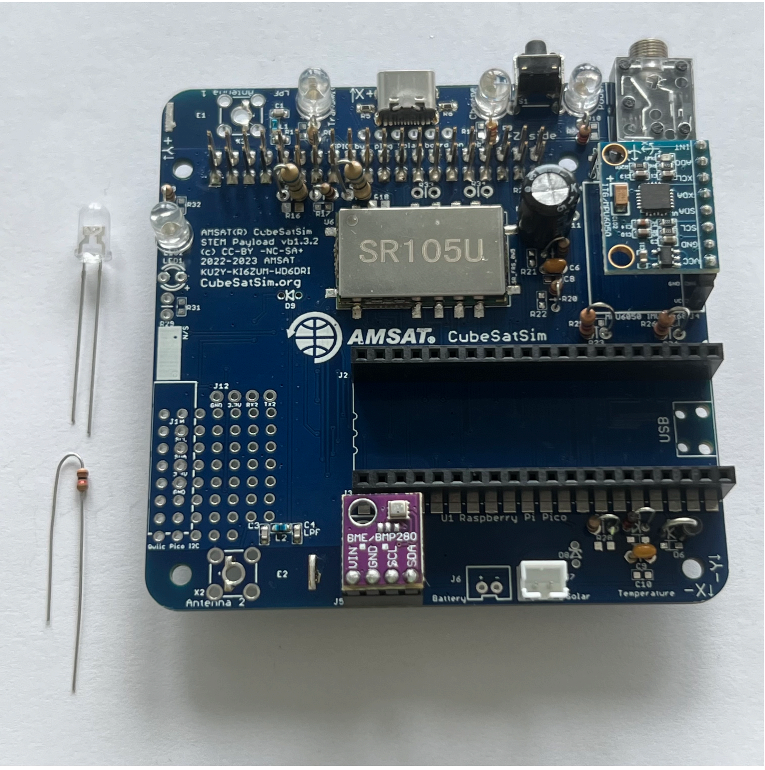


Then install the green LED1 and 1k Ohm R29 (brown black red color bands). LEDs need to be installed with the correct polarity (one lead is + polarity and the other lead is - polarity) or it will not illuminate. The longer leg on the LED is the '+' lead and should be away from the edge of the PCB. Also, 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.

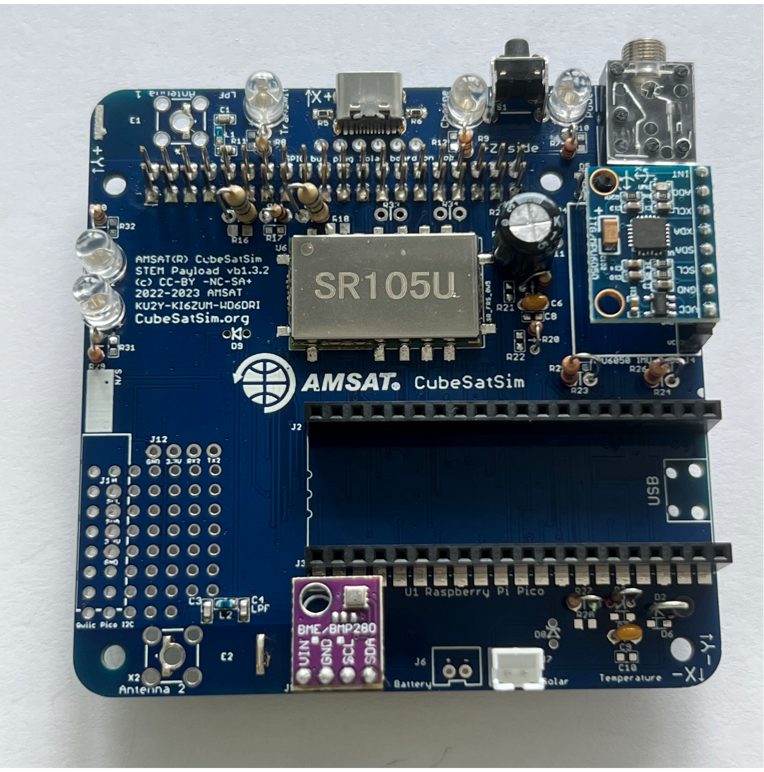
The longer lead of the LED is the positive and matches the '+' on the PCB. This LED and the blue one next to it are turned on and off by the microcontroller.



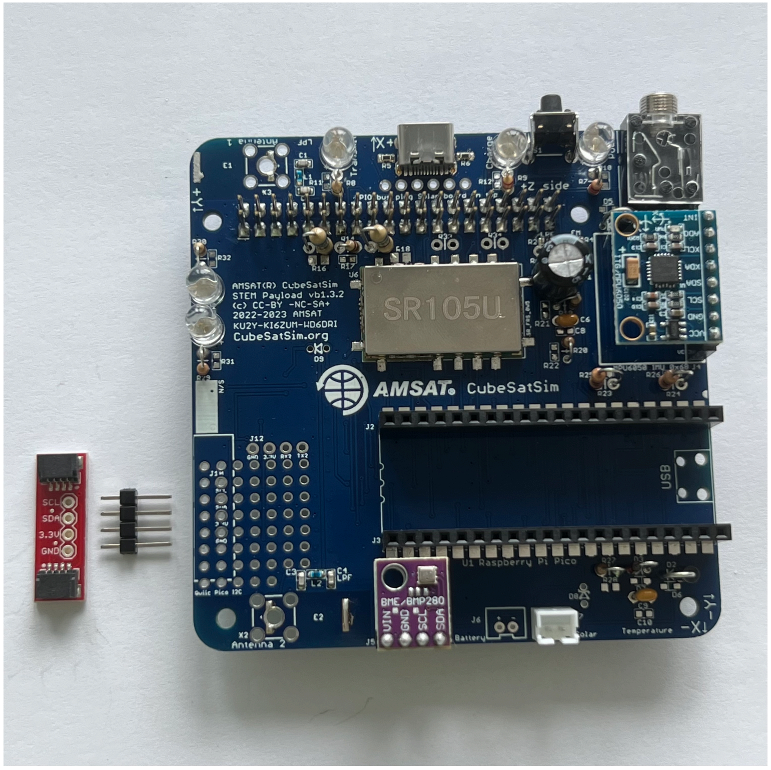
Then install the blue LED2 and 100 Ohm R30 (brown black brown color bands). The longer lead of the LED is the positive and matches the '+' on the PCB.



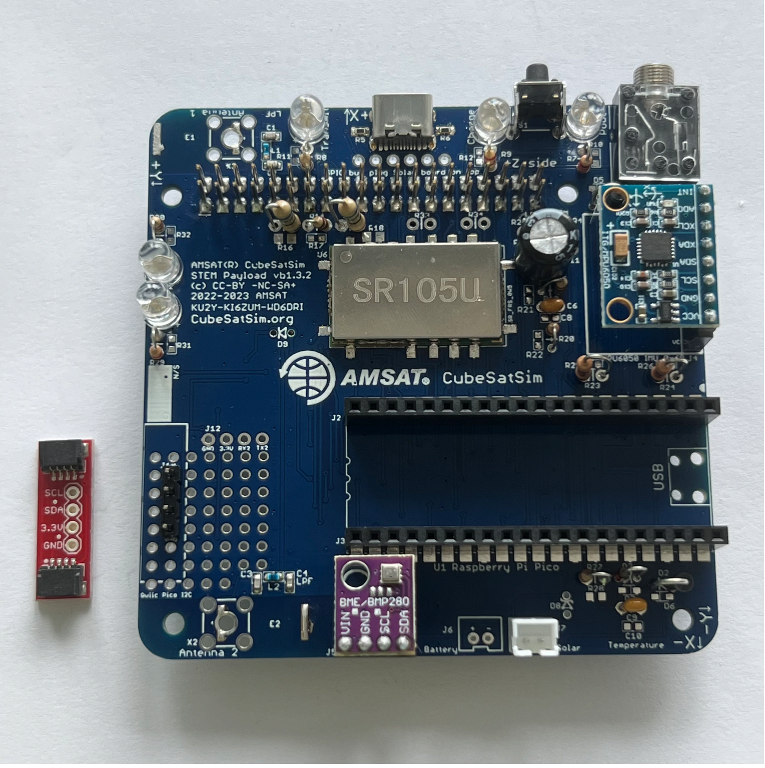
Here are all the resistors and LEDs installed:



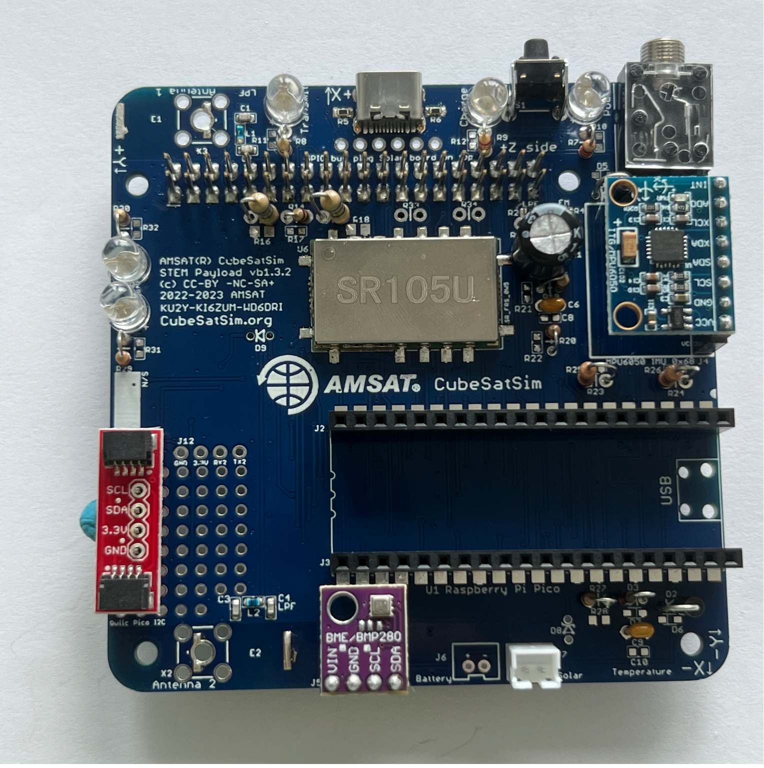
Next, install the Qwiic connector (red board) and 1x4 male pin header J10 for adding more I2C sensors to the board:



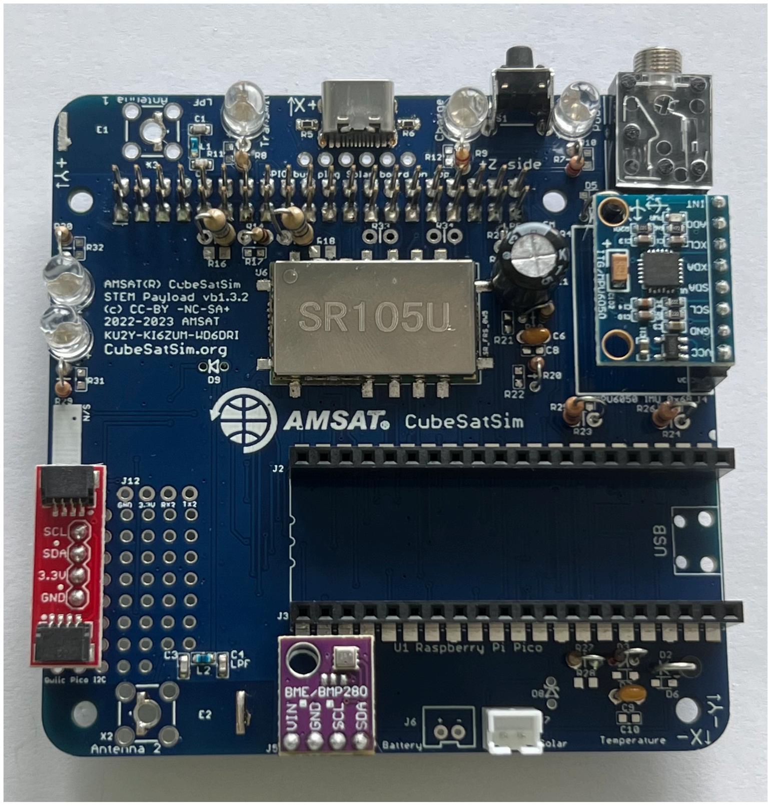
First install and solder the four pin male header J10, long pins down:



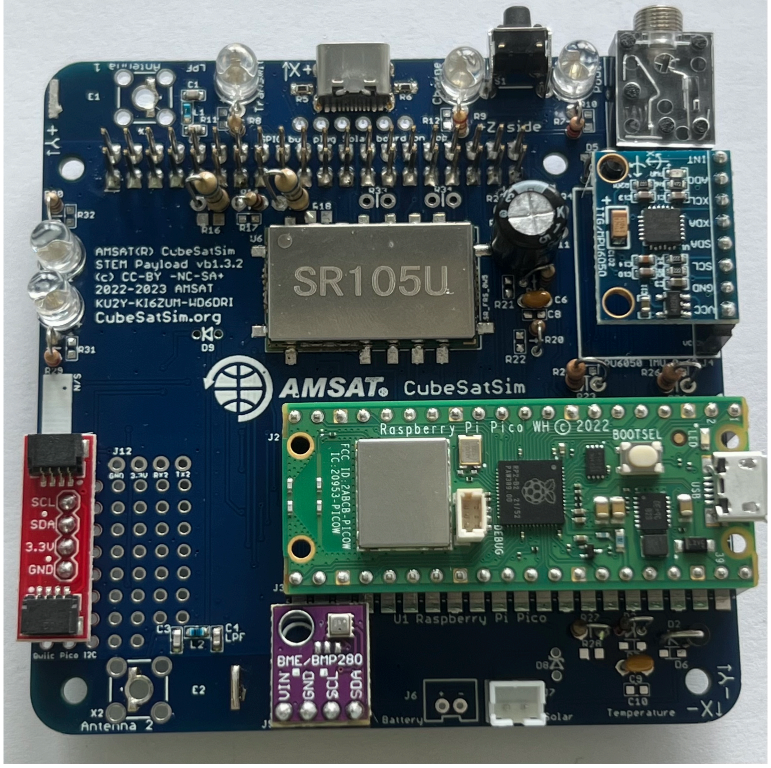
Then, install the connector, using blue putty to hold it level:



Here's how it looks after soldering:



Now, insert the Pico and Pi Zero, and you are now ready to test the board.



4.3 Testing

Video

Here is a video of testing the STEM Payload Board

Payload Sensor Testing

You can now test that the Raspberry Pi Pico can communicate with the two sensor boards. Your Raspberry Pi Pico should already have the software installed from Step 2. If not, go back to Step 2 and install it.

You will connect a micro USB to the Pico and to your computer which has the Arduino IDE installed.

In the Arduino software, 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. You should see 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
```

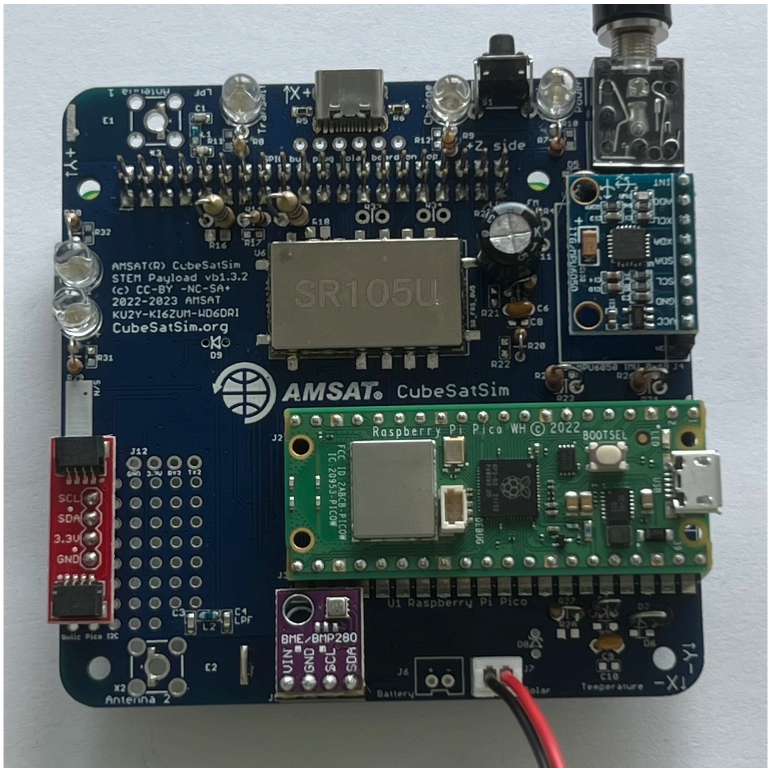
The `OK` is the status response. The four numbers after the `BME280` are the temperature in Celsius, pressure in hPa, altitude in meters, and humidity in percentage read from the purple BME280 sensor. The six numbers after the `MPU6050` are the X, Y, and Z axis angular rotation in degrees per second 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 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 indicate a software problem.

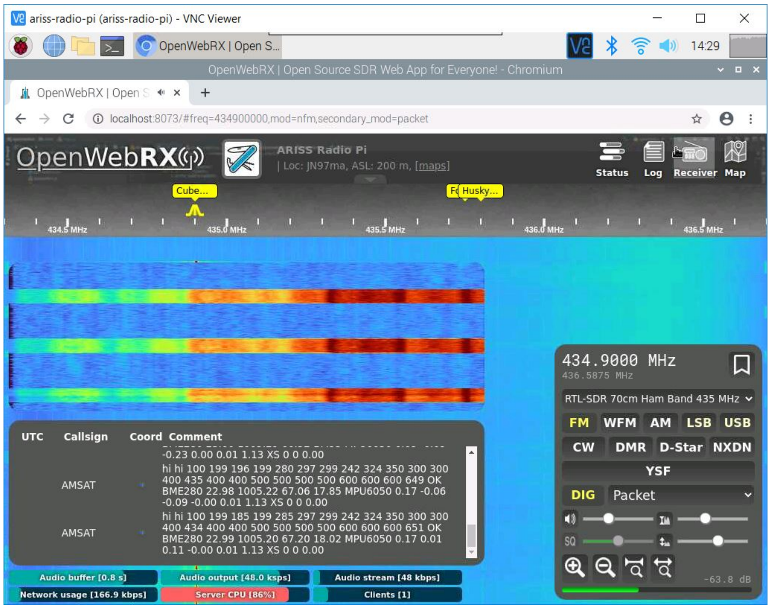
Here is the complete STEM Payload Board:



Complete Payload Test

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 Pico blink once, then twice, then repeat.

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



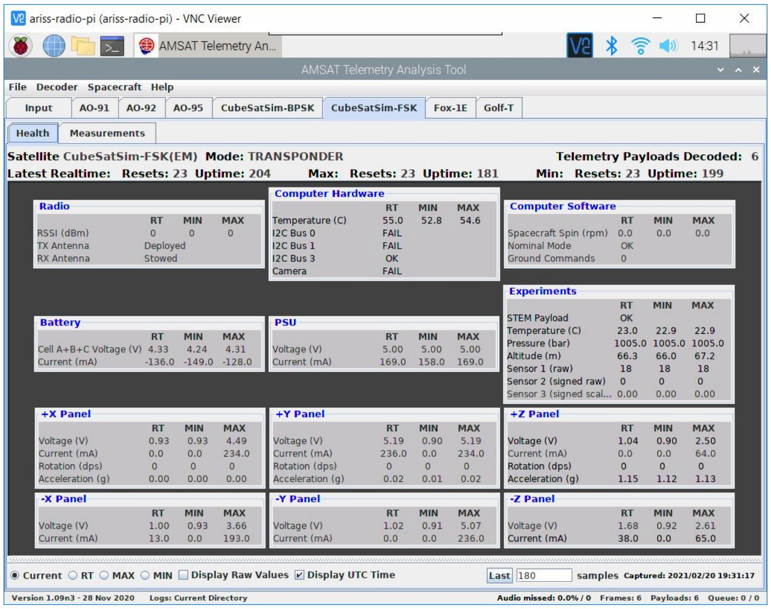
localhost:8073/#freq=434900000,mod=nfm,secondary_mod=packet

434.9000 MHz
436.5875 MHz
RTL-SDR 70cm Ham Band 435 MHz
DIG Packet

```
UTC   Callsign   Coord  Comment
                        -0.23 0.00 0.01 1.13 XS 0 0 0.00
                        hi hi 100 199 196 199 280 297 299 242 324 350 300 300
      AMSAT             400 435 400 400 500 500 500 500 600 600 600 649 OK
                        BME280 22.98 1005.22 67.06 17.85 MPU6050 0.17 -0.06
                        -0.09 -0.00 0.01 1.13 XS 0 0 0.00
                        hi hi 100 199 185 199 285 297 299 242 324 350 300 300
      AMSAT             400 434 400 400 500 500 500 500 600 600 600 651 OK
                        BME280 22.99 1005.20 67.20 18.02 MPU6050 0.17 0.01
                        0.11 -0.00 0.01 1.13 XS 0 0 0.00
```

Audio buffer [0.8 s] | Audio output [48.0 ksps] | Audio stream [48 kbps]
Network usage [166.9 kbps] | Server CPU [86%] | Clients [1]

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.



Satellite CubeSatSim-FSK(EM) Mode: TRANSPONDER   Telemetry Payloads Decoded: 6
Latest Realtime: Resets: 23 Uptime: 204   Max: Resets: 23 Uptime: 181   Min: Resets: 23 Uptime: 199

Radio

	RT	MIN	MAX
RSSI (dBm)	0	0	0
TX Antenna	Deployed		
RX Antenna	Stowed		

Computer Hardware

	RT	MIN	MAX
Temperature (C)	55.0	52.8	54.6
I2C Bus 0	FAIL		
I2C Bus 1	FAIL		
I2C Bus 3	OK		
Camera	FAIL		

Computer Software

	RT	MIN	MAX
Spacecraft Spin (rpm)	0.0	0.0	0.0
Nominal Mode	OK		
Ground Commands	0		

Experiments

	RT	MIN	MAX
STEM Payload	OK		
Temperature (C)	23.0	22.9	22.9
Pressure (bar)	1005.0	1005.0	1005.0
Altitude (m)	66.3	66.0	67.2
Sensor 1 (raw)	18	18	18
Sensor 2 (signed raw)	0	0	0
Sensor 3 (signed scal...	0.00	0.00	0.00

Battery

	RT	MIN	MAX
Cell A+B+C Voltage (V)	4.33	4.24	4.31
Current (mA)	-136.0	-149.0	-128.0

PSU

	RT	MIN	MAX
Voltage (V)	5.00	5.00	5.00
Current (mA)	169.0	158.0	169.0

+X Panel

	RT	MIN	MAX
Voltage (V)	0.93	0.93	4.49
Current (mA)	0.0	0.0	234.0
Rotation (dps)	0	0	0
Acceleration (g)	0.00	0.00	0.00

+Y Panel

	RT	MIN	MAX
Voltage (V)	5.19	0.90	5.19
Current (mA)	236.0	0.0	234.0
Rotation (dps)	0	0	0
Acceleration (g)	0.02	0.01	0.02

+Z Panel

	RT	MIN	MAX
Voltage (V)	1.04	0.90	2.50
Current (mA)	0.0	0.0	64.0
Rotation (dps)	0	0	0
Acceleration (g)	1.15	1.12	1.13

-X Panel

	RT	MIN	MAX
Voltage (V)	1.00	0.93	3.66
Current (mA)	13.0	0.0	193.0

-Y Panel

	RT	MIN	MAX
Voltage (V)	1.02	0.91	5.07
Current (mA)	0.0	0.0	236.0

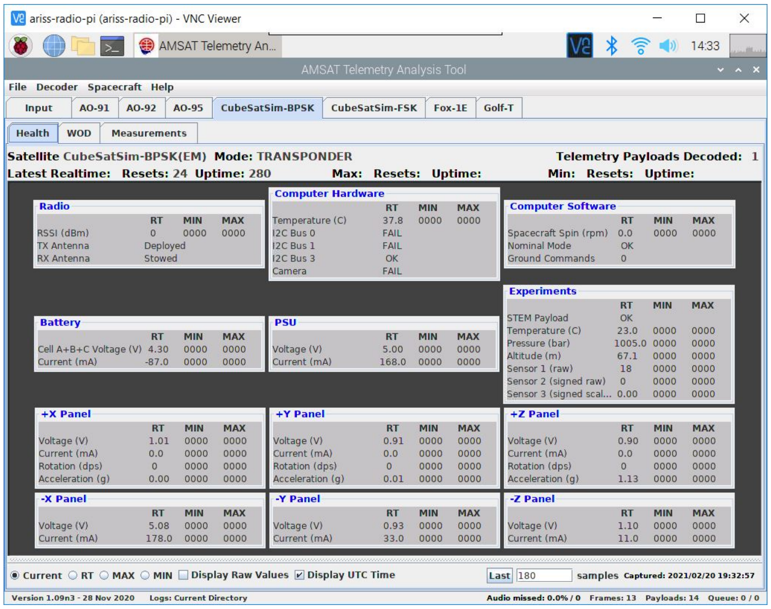
-Z Panel

	RT	MIN	MAX
Voltage (V)	1.68	0.92	2.61
Current (mA)	38.0	0.0	65.0

● Current ○ RT ○ MAX ○ MIN □ Display Raw Values ☑ Display UTC Time   Last 180 samples Captured: 2021/02/20 19:31:17

Version 1.09n3 - 28 Nov 2020   Logs: Current Directory   Audio missed: 0.0% / 0   Frames: 6   Payloads: 6   Queue: 0 / 0

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:


ariss-radio-pi (ariss-radio-pi) - VNC Viewer
AMSAT Telemetry An...
14:33
AMSAT Telemetry Analysis Tool
File Decoder Spacecraft Help
Input AO-91 AO-92 AO-95 CubeSatSim-BPSK CubeSatSim-FSK Fox-1E Golf-T
Health WOD Measurements

Satellite CubeSatSim-BPSK(EM) Mode: TRANSPONDER Telemetry Payloads Decoded: 1

Latest Realtime: Resets: 24 Uptime: 280 Max: Resets: Uptime: Min: Resets: Uptime:

Radio	RT	MIN	MAX
RSSI (dBm)	0	0000	0000
TX Antenna	Deployed		
RX Antenna	Stowed		

Computer Hardware	RT	MIN	MAX
Temperature (C)	37.8	0000	0000
I2C Bus 0	FAIL		
I2C Bus 1	FAIL		
I2C Bus 3	OK		
Camera	FAIL		

Computer Software	RT	MIN	MAX
Spacecraft Spin (rpm)	0.0	0000	0000
Nominal Mode	OK		
Ground Commands	0		

Experiments	RT	MIN	MAX
STEM Payload	OK		
Temperature (C)	23.0	0000	0000
Pressure (bar)	1005.0	0000	0000
Altitude (m)	67.1	0000	0000
Sensor 1 (raw)	18	0000	0000
Sensor 2 (signed raw)	0	0000	0000
Sensor 3 (signed scal...	0.00	0000	0000

Battery	RT	MIN	MAX
Cell A+B+C Voltage (V)	4.30	0000	0000
Current (mA)	-87.0	0000	0000

PSU	RT	MIN	MAX
Voltage (V)	5.00	0000	0000
Current (mA)	168.0	0000	0000

+X Panel	RT	MIN	MAX
Voltage (V)	1.01	0000	0000
Current (mA)	0.0	0000	0000
Rotation (dps)	0	0000	0000
Acceleration (g)	0.00	0000	0000

+Y Panel	RT	MIN	MAX
Voltage (V)	0.91	0000	0000
Current (mA)	0.0	0000	0000
Rotation (dps)	0	0000	0000
Acceleration (g)	0.01	0000	0000

+Z Panel	RT	MIN	MAX
Voltage (V)	0.90	0000	0000
Current (mA)	0.0	0000	0000
Rotation (dps)	0	0000	0000
Acceleration (g)	1.13	0000	0000

-X Panel	RT	MIN	MAX
Voltage (V)	5.08	0000	0000
Current (mA)	178.0	0000	0000

-Y Panel	RT	MIN	MAX
Voltage (V)	0.93	0000	0000
Current (mA)	33.0	0000	0000

-Z Panel	RT	MIN	MAX
Voltage (V)	1.10	0000	0000
Current (mA)	11.0	0000	0000

● Current ○ RT ○ MAX ○ MIN □ Display Raw Values ☑ Display UTC Time Last 180 samples Captured: 2021/02/20 19:32:57

Version 1.09n3 - 28 Nov 2020 Logs: Current Directory Audio missed: 0.0% / 0 Frames: 13 Payloads: 14 Queue: 0 / 0

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 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.

Adding Additional Sensors

Here's how to add more sensors to your STEM Payload board.

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 next step is to build the Battery board.

+ Add a custom footer