A Guidance for Assembling, Programming, and Operating the Power interval Camera [and sensor] Automation Module (PiCAM)

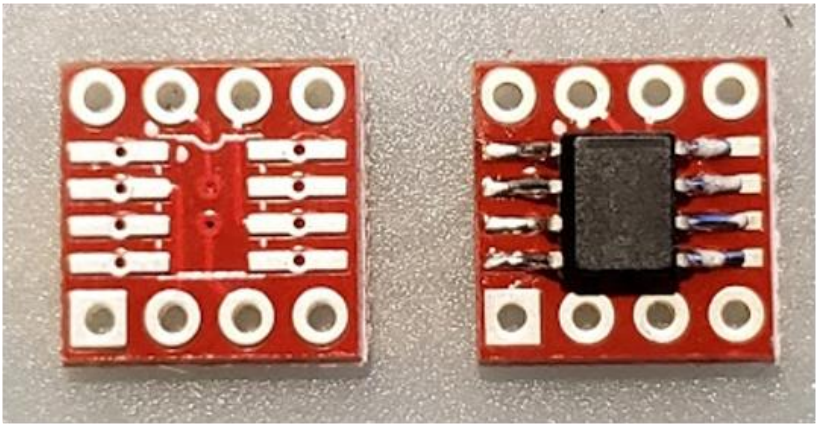
This document provides step-by-step guidance on how to assemble, program, and operate a Power interval Camera [and sensor] Automation Module (PiCAM) designed by the Terrestrial Ecosystem Science & Technology (TEST) Group at Brookhaven National Laboratory. This guidance is provided in alignment with a Wikipedia version that can be found on PiCAM GitHub: https://github.com/TESTgroup-BNL/PiCam/wiki. For any questions you have regarding this guidance, please contact:

Andrew McMahon (amcmahon@bnl.gov), Daryl Yang (dediyang@bnl.gov), Jeremiah Anderson (jeanderson@bnl.gov), Shawn Serbin (sserbin@bnl.gov)

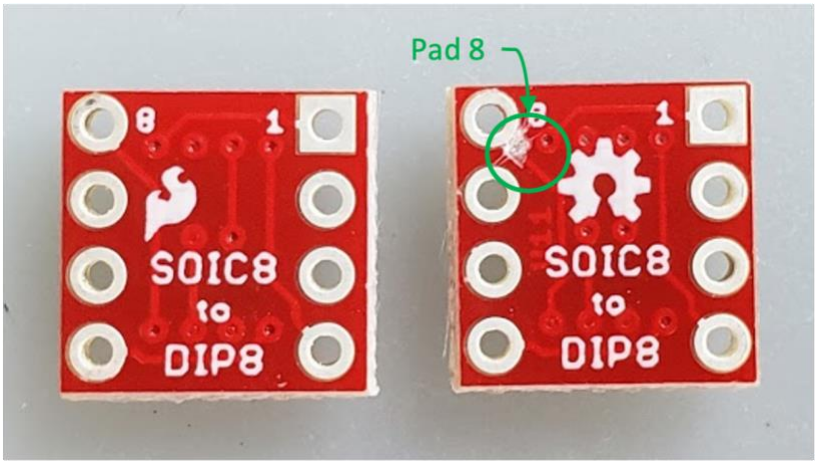
A. How to Assemble a PiCAM

1. Prepare power distribution board.

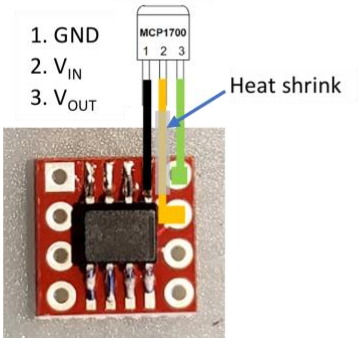
a. Attach TPS2080 to the breakout board.

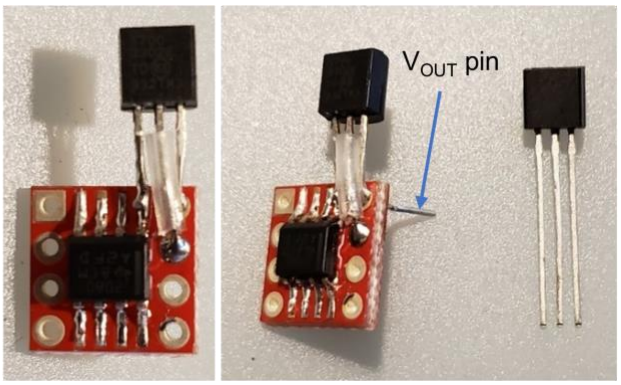


b. Cut trace to pad 8.

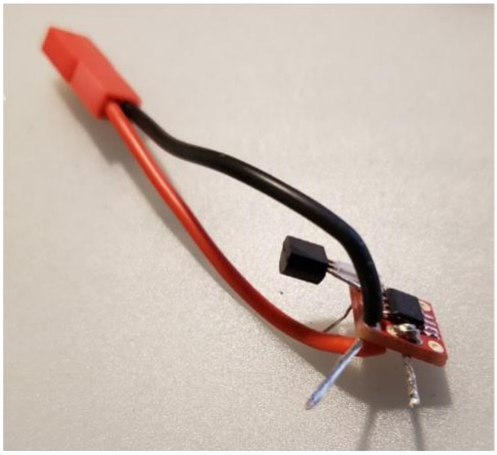


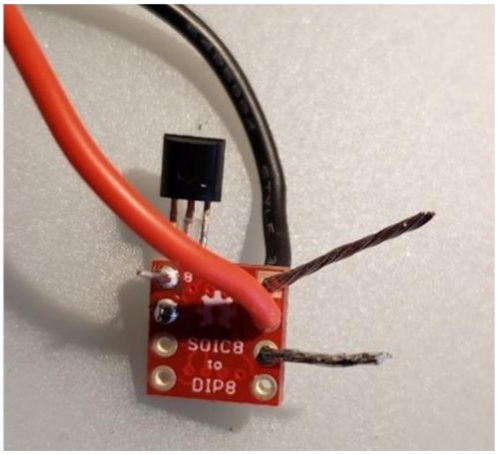
c. Attach MCP1700 (3.3V regulator). Leave V_{OUT} pin long to attach I^2C (GPS) power wire.



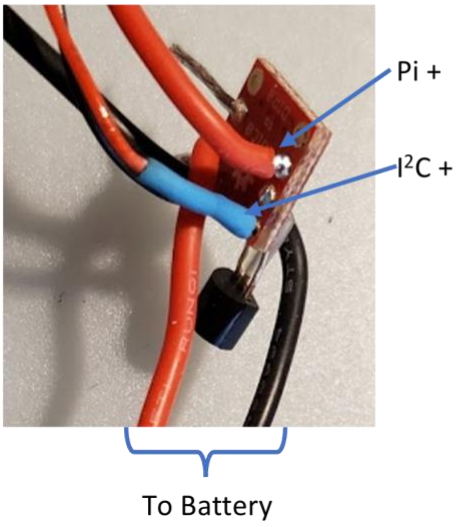


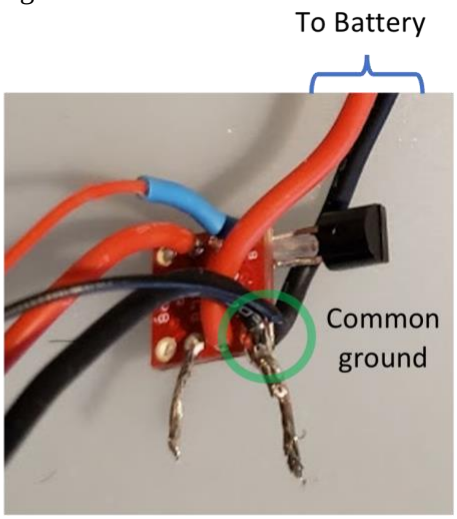
d. Attach the battery connector. If possible, pull about ½" of bare wire through the board, which will be used to attach to the USB pads on the Nordic dongle. The ground should go through from the chip side on pad 1 and the positive wire should be looped from the bottom through holes 2 and 3 and back down, so that both wires extend out of the bottom of the board.



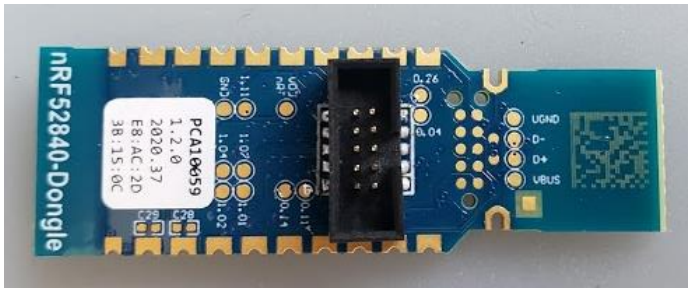


e. Connect the Pi and I^2C power wires; The Pi positive connects to pad 6 and the I^2C positive to pad 7, while both ground simply connect to the extended ground wire.



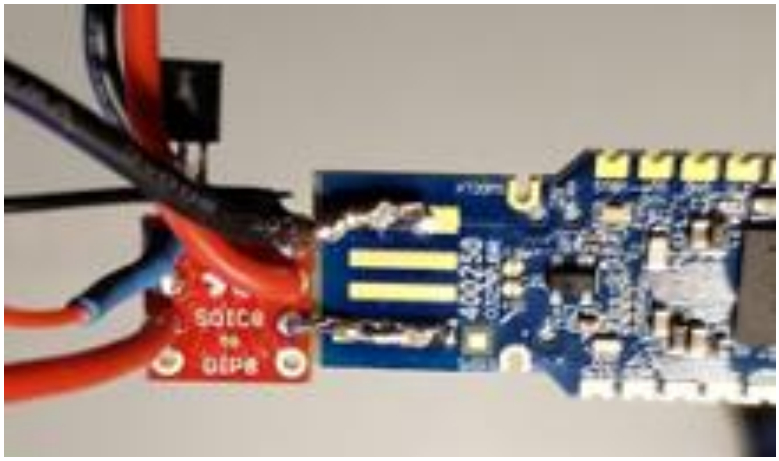


2. **Attach J-link header to the Nordic nrf52840 dongle.**

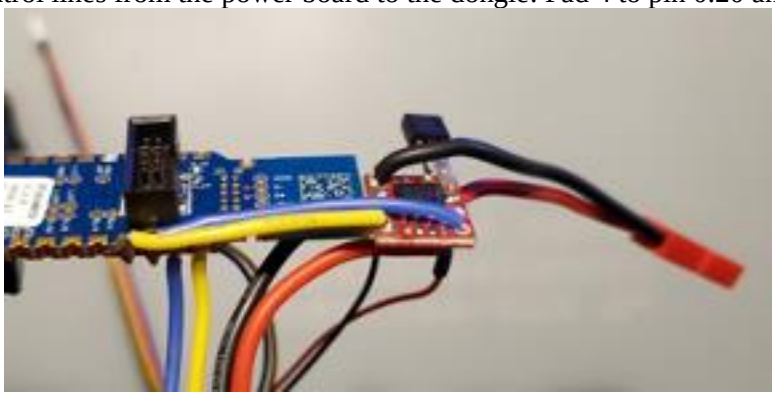


3. **Attach power board to the Nordic dongle.**

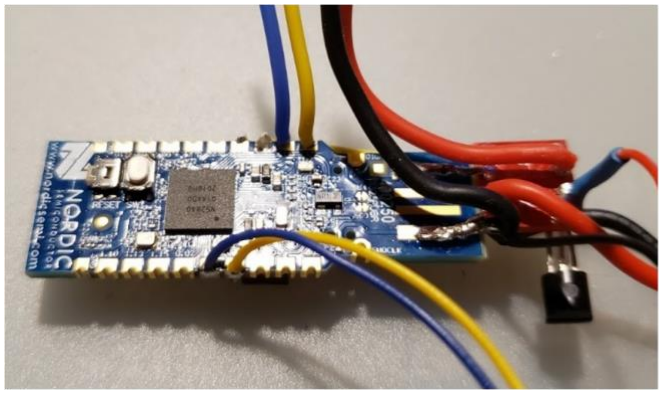
   a. Connect V+ and ground to the USB pads. Note for reference that the positive pad has the square test point next to it.

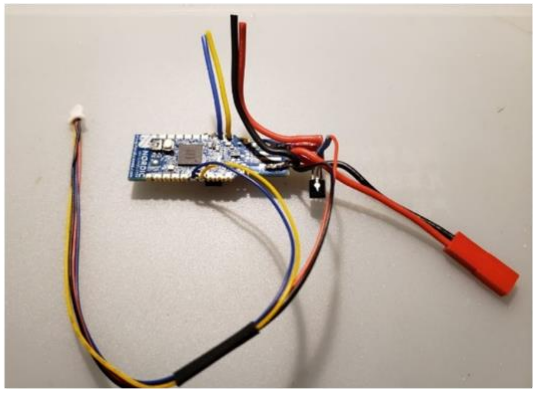


   b. Connect the control lines from the power board to the dongle: Pad 4 to pin 0.20 and Pad 5 to 0.17.

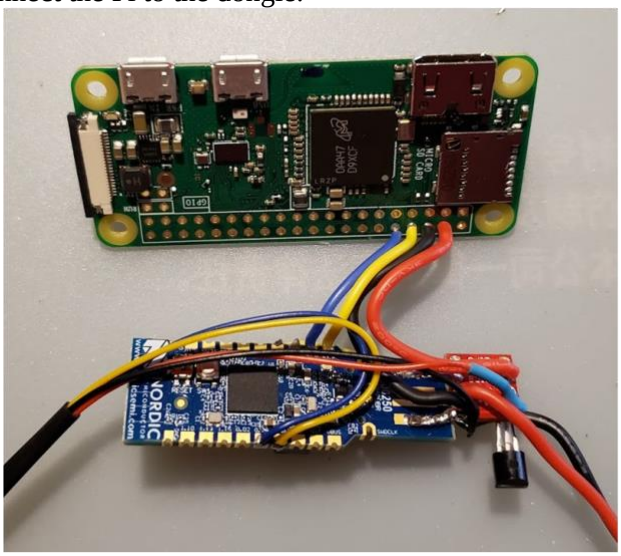


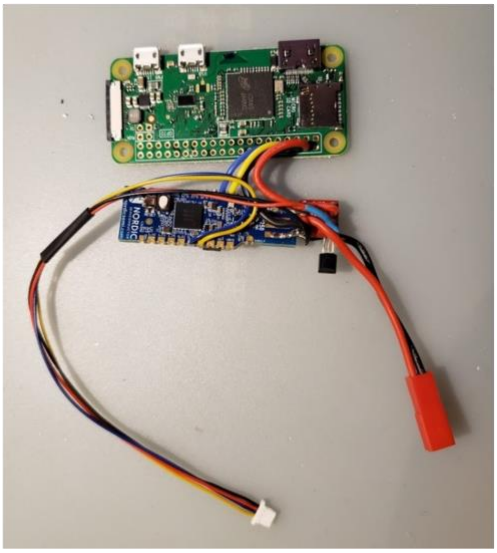
4. **Connect serial lines for the Pi to the dongle on 0.13 (yellow) and 0.15 (blue) and I^2C lines on 0.29 (blue) and 0.31 (yellow).**



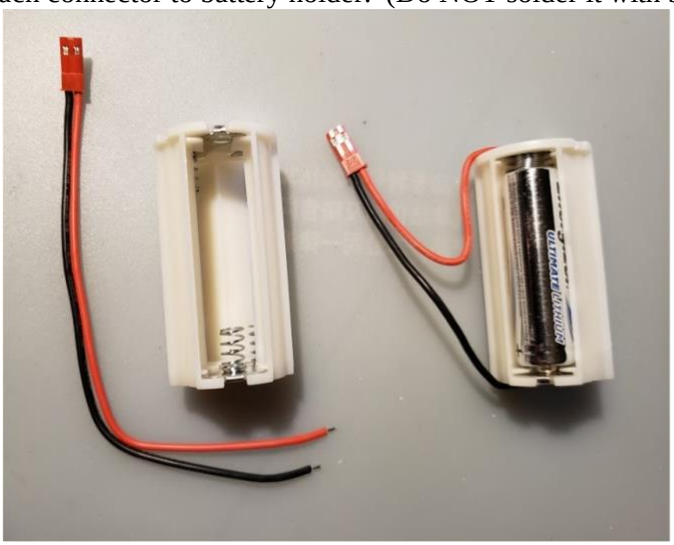


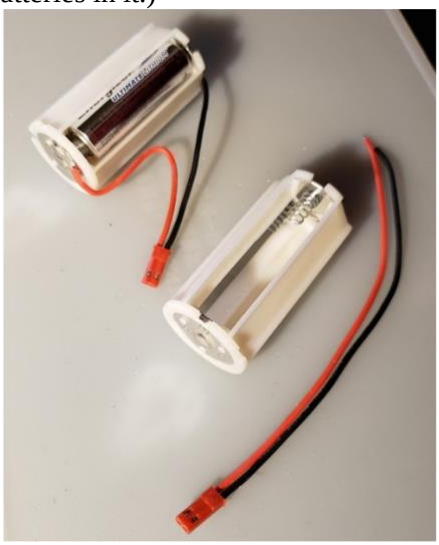
5. Connect the Pi to the dongle.





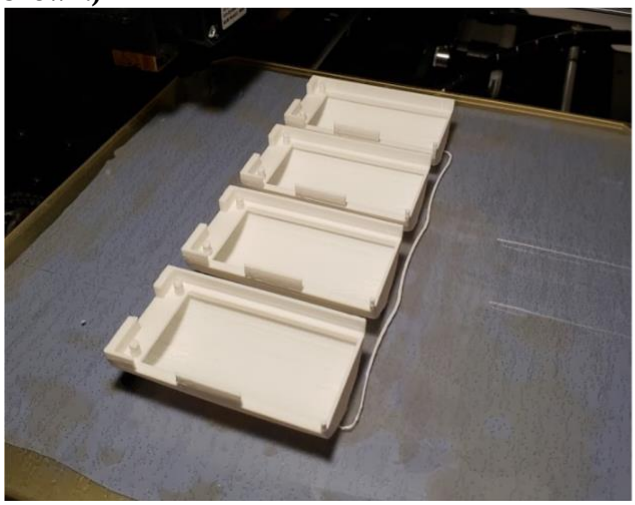
6. Attach connector to battery holder. (Do NOT solder it with batteries in it.)

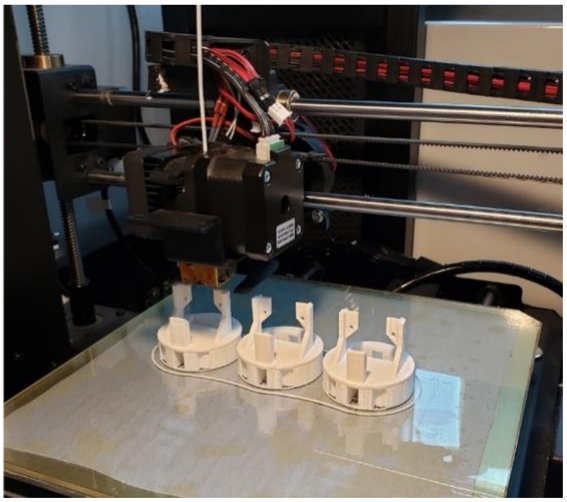




NOTE: This is the last of the soldering.

7. Print the "D-Cell" case and cam/GPS mount. (Pi-side of D-shell and partial cam/gps mount shown.)



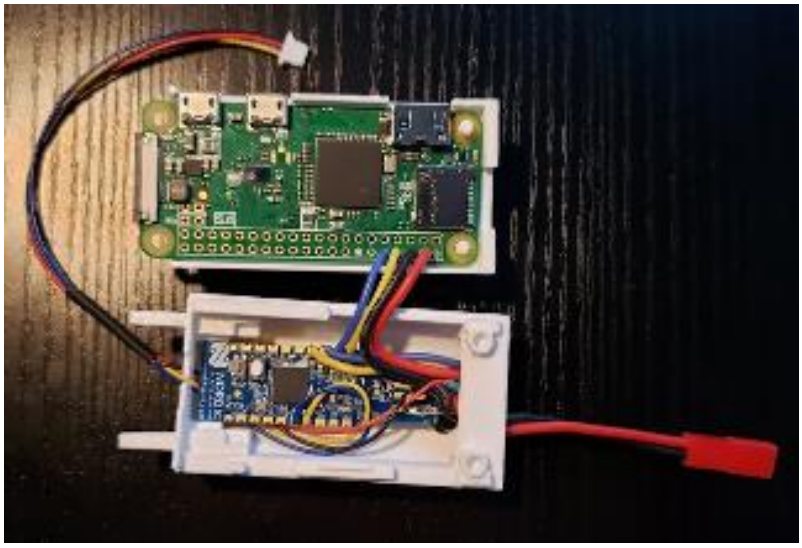


8. Put the dongle and Pi into the D-cell case:

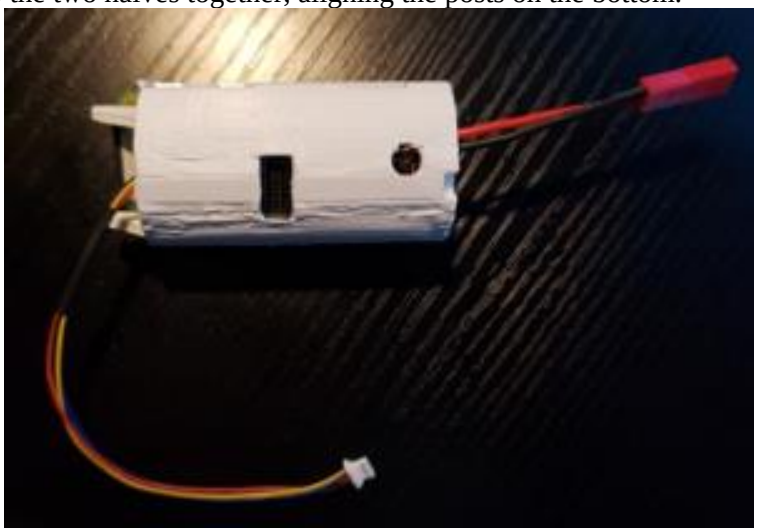
a. Feed the battery connector through the bottom hole and tuck the nrf52840 dongle into the half D-cell. The power/regulator board should be on the bottom, tucked underneath the overhang, with the battery wire running just above the regulator through the thinner section of the bottom opening. Tuck the I^2C cable behind the dongle as it is moved into place.

b. Gently, push the dongle down until the J-link connector can slide into the opening in the side of the case. It should pop in without much force; be careful not to crush the power board/regulator. Once the connector is in place, push down on the end of the dongle at the top of the case (where it says "Nordic") to snap it into place.

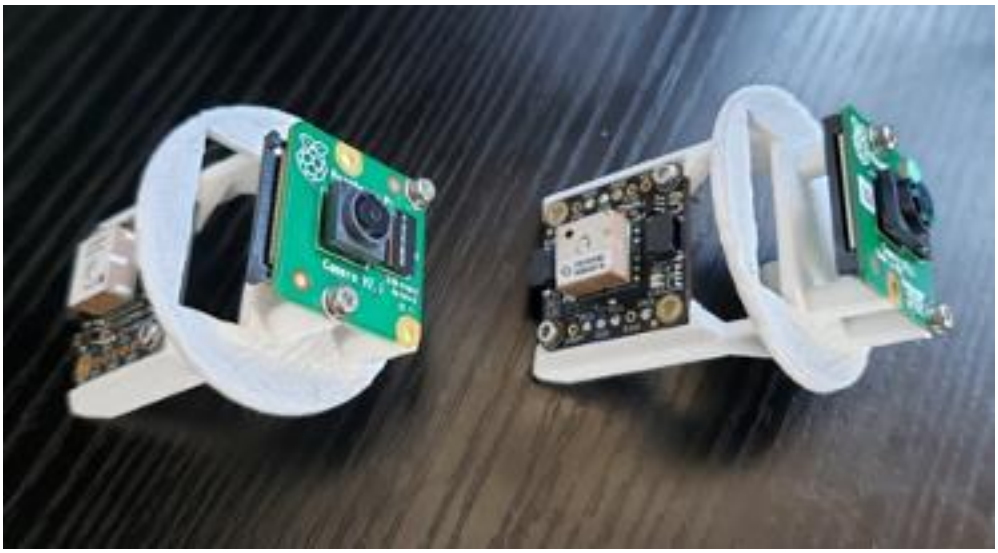
c. Snap the Pi into the other half of the case. It needs to go in flat to avoid catching on the edges. If it is not sitting flush, try to realign it rather than using more force.



d. Finally, snap the two halves together, aligning the posts on the bottom.



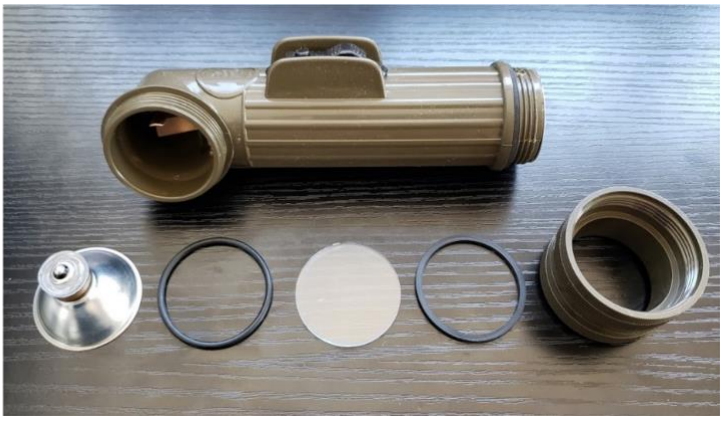
9. Attach the camera and GPS modules to the cam/GPS mount using M2 screws and washers on at the camera module. (The GPS module has pads very close to the screw holes, which could make washers risky depending on their diameter.)



10. Prepare the flashlight housing

a. Unscrew the bottom and remove the base. Pull out the battery spring (needle nose pliers help) and spare bulb.

b. Unscrew the top and remove the bulb, both o-rings and the plastic window.

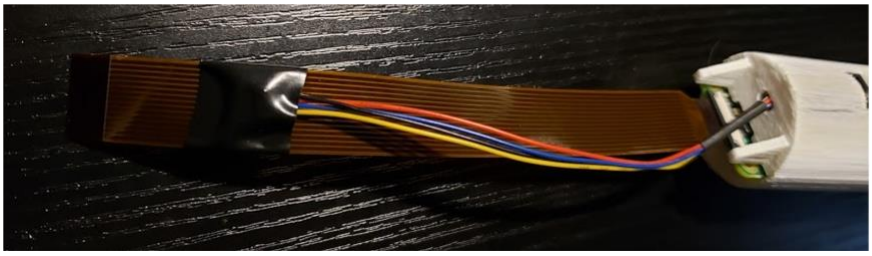
c. Place the new window into the front section of the top and put the round o-ring on top of it. Then, screw the back half on so that it seals against the o-ring.


d. Put the squared o-ring around the screw base on the front of the camera body.

e. Pull out the copper bulb contact.

11. Insert the ribbon and I²C cables

Option 1: If the camera housing has a large enough opening, it may be possible to slide the camera ribbon cable through while attached to the Pi. In this case, it's easy to feed the ribbon cable and I²C cable by taping them together. Having the I²C cable attached also provides some tension to make positioning the ribbon cable easier.



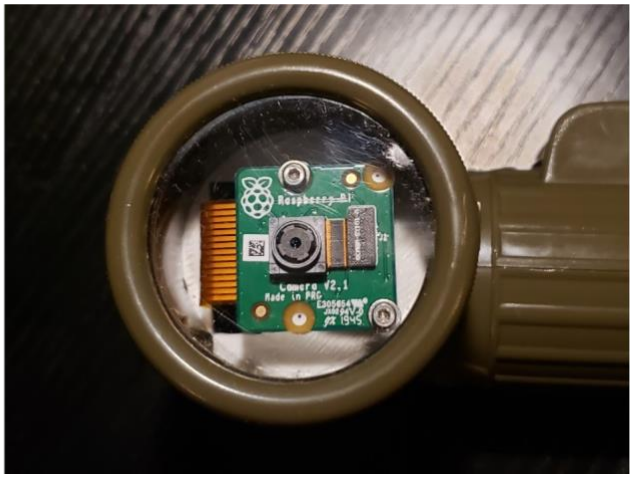
Option 2: If the opening is too narrow to easily slide the cable though, the ribbon cable can be pushed through from the top, then attached to the Pi.

Note: This is the last opportunity to access the J-link connector and upload firmware without disconnecting the ribbon and I²C cables!

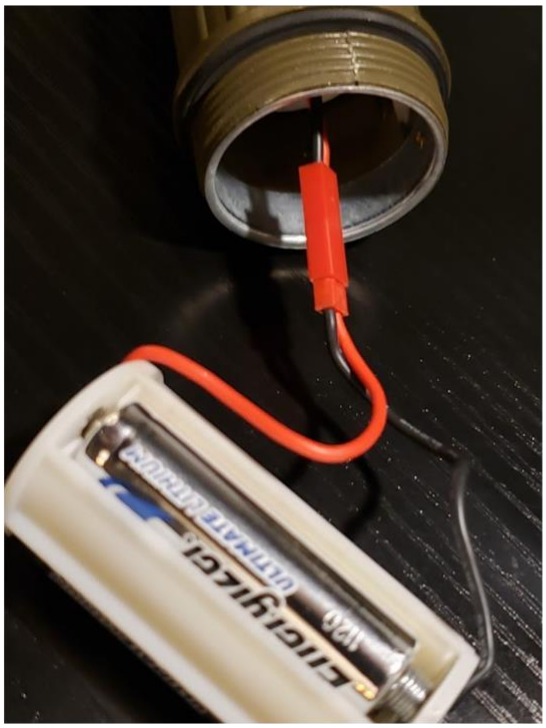
12. **Slide the D-cell case into the camera body. Gently "pull" (more so allow them to move freely) the two cables from the front of the camera to ensure that they do not get tangled/crumpled above the D-cell case.**

13. **Attach the I2C cable to the GPS module and the ribbon cable to the camera module.**

14. **Insert the camera/GPS assembly. Place a desiccant packet under the GPS module and slide the assembly into place. (The bottom is keyed to lock into the bottom of the body opening and keep the assembly straight.) Then, screw on the front.**



15. **Finally, connect the battery pack and slide it into the camera body below the D-cell case. There should be space for the battery connector and any excess wiring around the sides of the battery pack. Avoid bunching up wiring above or below the pack as damaged insulation there could allow a short to one of the battery terminals.**

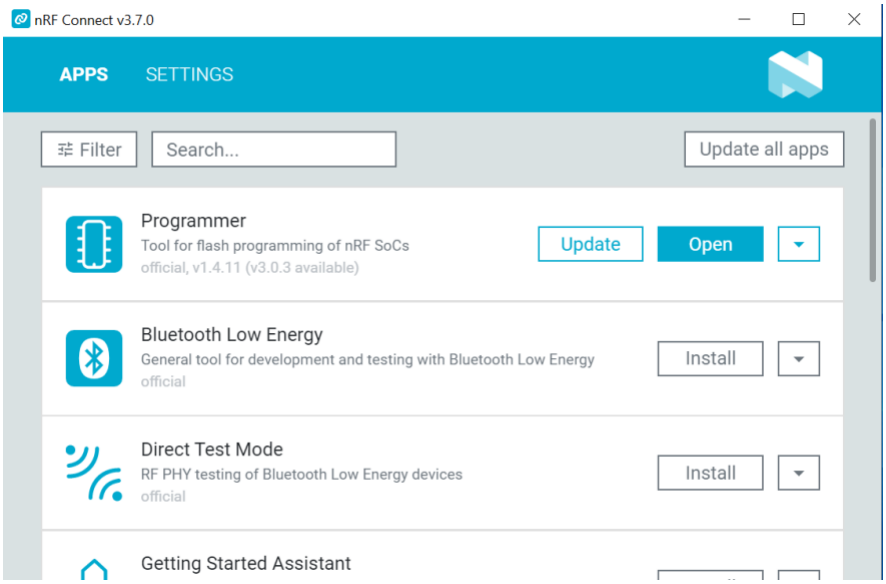


B. How to Program a PiCAM

1. Download the nRF firmware hex file (SoftDevice_S140_6_1_1-App_2_1_1.hex) from PiCAM GitHub (https://github.com/TESTgroup-BNL/PiCam/tree/main/ARM_Firmware)

2. Download nRF Connect for Desktop (https://www.nordicsemi.com/Products/Development-tools/nrf-connect-for-desktop/download) and install it on a PC or Mac.

3. Open nRF Connect, and install Programmer. Once installed, open Programmer.

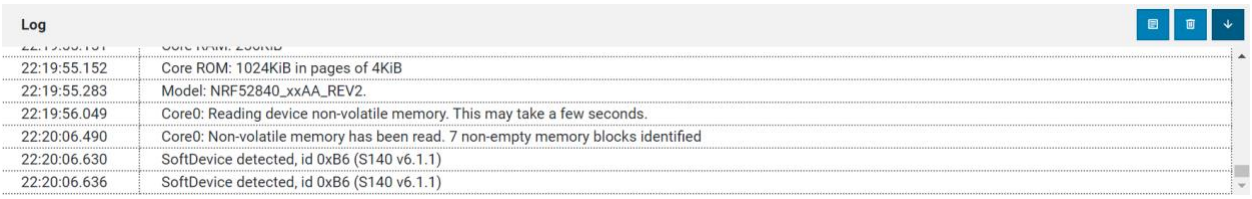


4. Connect the PiCAM to your PC or MacOS through a USB port using J-Link GTAG/SWD debugger (https://www.segger.com/products/debug-probes/j-link/models/j-link-edu-mini/).

5. In nRF Connect, click “SELECT DEVICE”, and choose your PiCAM from the drop-down list. Once your PiCAM is connected, its memory layout will be shown in the right-hand panel.

6. In nRF, click on ‘Add File’ and locate your .hex file downloaded from PiCAM GitHub. Or you can directly drag and drop your .hex file to the left-hand panel “File memory layout”.

7. Once both the ‘File memory layout” and “Device memory layout” are loaded, click “Erase and write” to overwrite the Device memory using .hex file. Once this step is completed, you will see a printout in the log like below.



8. Disconnect your PiCAM from nRF Connect.
9. Download Pi Zero firmware folder from GitHub:

   https://github.com/TESTgroup-BNL/PiCam/tree/main/Pi_Firmware
10. Prepare an empty microSD card and format it.
11. Copy and paste the Pi Zero firmware into the main folder on your microSD card.
12. Open the ‘cam_cfg.txt” file and modify image capture frequency, time zone (UTM+), and other necessary parameters based on user need.
13. Plug the SD card into Pi Zero

C. First Power Up and Operation Loop

When first powered on, the PiCAM will first turn on the nRF supervisory microcontroller. The nRF will then attempt to:

1. Power on the Pi Zero,
2. Read the configuration file stored on microSD card,
3. Power on the Pi Camera module and take a picture,
4. Power on the GPS module and get a GPS lock for time and location (the GPS module has a green LED that is visible through the camera window). This step might take one to a couple of minutes to complete.

At this point, the camera will show up as a Bluetooth device and can be scanned by any device with Bluetooth Lower Energy (BLE) capacity, but will not yet show accurate information (ID, date, time) in the advertising string. Once a GPS lock is acquired (or times out), the camera will enter "normal" operation mode, where the scheduled event loop takes over. Once in the main loop, the Bluetooth advertising information will be updated every 5 seconds, and it will respond normally to Bluetooth connections.

Normal Operation Loop

Once operating, the PiCAM will capture images at the interval specified in the configuration file (https://github.com/TESTgroup-BNL/PiCam/blob/main/Pi_Firmware/cam_cfg.txt). The image interval is synchronized to hours/minutes, so that offsets are evenly spaced throughout the day starting from 12 AM, rather than a timed interval between images. For example, if a camera is set to a 15-minute interval and powered on at 11:52 AM, its first image will be at 12:00 PM and the next at 12:15 PM, rather than a direct offset of 15 minutes from power on, which would give uneven times of 12:07 PM, 12:22 PM.

Turning on/off: The camera starts when the batteries are connected and stops when they are disconnected; there is no other start/stop mechanism. Disconnecting power while an image is being saved should be avoided as much as possible as this could cause SD card corruption. (Users can look at the UART terminal or dashboard to check if an image is currently being captured). There is no other "safe power down" requirements.

D. Bluetooth Communication with External Devices

The Bluetooth function of a PiCAM stays on at all times and can provide information in a variety of ways:

- **The Advertising String**
  An advertising string is visible without connecting or any specific software. Any device (phone, computer, etc.) that can scan for Bluetooth Low Energy (BLE) devices should be able to "see" it. The format of the advertising string is shown below:

  Format: PiCam <ID> <Battery Level as %> <images captured in current run> <date> <time>
  Example: PiCam001 95 0512 08/07 13:08

- **Serial UART Connection**
  This connection provides a terminal-like text interface. The Adafruit Bluefruit Connect app works great for this and is available for both Android and iOS.

  Usage: Once connected to a PiCAM, the app will receive periodic strings from the camera with basic status information. Sending any character will trigger the app to print out a menu with several options:

  'c': Print cfg
  'g': Get GPS fix
  'i': Capture image now
  'r': Reload cfg
  't [YYYY,MM,DD,hh,mm,ss]': Set time

  Sending a specified character listed above will trigger the command. If users do not wish to print the menu, sending a double character of any command above will execute it without displaying the menu. For example, to quickly trigger getting a GPS fix, sending 'gg' will immediately start that command.

- **Web Dashboard**
  The web dashboard provides the most remote functionality including all the items from the UART menu as well as getting preview images and syncing time to the host. It displays the UART data as a text feed and additionally parses it to better display the data.
  The dashboard utilizes Web Bluetooth, which is currently an experimental API but has reasonable support for Chrome (the progress of adoption can be checked here). Nevertheless, some bugs and changes should be expected as this continues to develop. The current site being used for testing is hosted at https://web-picam.glitch.me/, but should be cached offline for use in the field. Tip: Connecting can take 10-15 seconds; if the Bluetooth icon hasn't appeared on the tab yet, clicking connect again sometimes does the trick. Other times it causes a disconnect or a "double connection" where everything is printed twice (though commands still work fine).

  Preview image: When triggered, this powers up the Pi, captures a low-resolution image, and transfers it to the nrf52840, which finally transfers it to the dashboard. Transferring from the Pi to the nrf52840 can be very slow (it's 230400 baud serial), so it may take 15-30 seconds even for a small image. The Bluetooth transfer is generally much faster, taking only a few seconds. Chrome can be picky about rendering the image once it's received; if it isn't displayed immediately, try opening/switching to a different tab and then back to force it to redraw.

Important: While connected to Bluetooth (UART or dashboard), the camera is consuming about 10 times as much power as in "normal" standby, so it is important to not stay connected indefinitely and to remember to disconnect! (If a connection is dropped by device moving out of range or interference, the camera will automatically drop the connection rather than being stuck in connected mode.)

E. Configuration File

Most aspects of the camera configuration are determined by the "cam_cfg.txt" file on the SD card. (If there is no file/SD card, the defaults will be used.) Each parameter should be on its own line in the format <Parameter> <Value> (one space between parameter and value). This file can include short comments after a parameter, separated by at least one space from the parameter value.

Parameter	Description	Default Value	Units	Format	Usable Range
ID	Arbitrary Camera ID	0	None	Unsigned Integer	0-999
IMAGE_INTERVAL	Frequency to capture images	30	Minutes	Unsigned Integer	1-1440
BLE_INTERVAL	Frequency to update advertising string	1	Minutes	Unsigned Integer	1-1440
GPS_INTERVAL	Frequency to update GPS info (location and time)	24	Hours	Unsigned Integer	1-720
CLK_SYNC_INT	Minimum time before next GPS clock sync	60000	Milliseconds	Unsigned Long	1000-86400000
CLK_RESOLUTION	Main loop delay time; all other timers are checked at this interval. Significantly impacts battery life.	5000	Milliseconds	Unsigned Long	1000-60000
PI_ON_TIMEOUT	Time to wait for communication from Pi after powering it on.	10000	Milliseconds	Unsigned Integer	1000-60000
PI_SAVE_TIMEOUT	Time to wait for a saved image confirmation from Pi. Setting too low can cause corrupted images.	30000	Milliseconds	Unsigned Integer	1000-60000
GPS_TIMEOUT	Time to wait for GPS fix	120000	Milliseconds	Unsigned Long	1000-1800000
MAX_BATT_VOLTAGE	Maximum expected battery voltage. Used to scale battery %.	5.2	Volts	Float	3.3-5.5
LOW_BATT_CUTOFF	Minimum battery voltage. Once the voltage falls below this threshold, the camera will no longer attempt to capture images. Also used to scale battery %.	3.4	Volts	Float	1.2-MAX_BATT_VOLTAGE
TZ_OFFSET	Time offset from UTC	-5	Hours	Signed Integer	-12-12

For most cases, a user should only need to change **IMG_INTERVAL**, **GPS_INTERVAL,** and **TZ_OFFSET**. Changing any parameter, especially CLK_RESOLTUION, IMG_INTERVAL and GPS_INTERVAL, can have a significant impact on battery life.

F. Image and Log Data

Images are saved to the SD card in the Pi. Each time the camera is powered on (or explicitly told to), it starts a new "run". Each run has a numbered folder in the images folder. This prevents any accidental overwrites and a degree of metadata if clock information is missing or inaccurate. Images are named in the following format:

"img_<cam ID>_<cam MAC>_<date>_<time>.jpg"
(Files for preview images are saved with the prefix "prv".)

Each time an image is saved, a CSV record is added to "log.txt" with the following fields:

Example record: *00:00:15, 0\img_5_0C7732331A02_2021-04-15_142000.jpg, 5.18,0.00,40.748938,N,-73.942496,W,77.400002,6,1*

Field	Description	Example
Time	Time since Pi powered on; NOT time of day. This is a useful diagnostic to determine how long it's taking the Pi to capture and save images. A slow or corrupt SD card can cause very long save times, impacting battery life.	00:00:15
Filename	Image filename. ("images" folder is implied as the root.)	0\img_5_0C7732331A02_2021-04-15_142000.jpg
Battery Voltage	Battery voltage before Pi is powered on.	5.18
Temperature	Internal temperature of nrf52840. *Currently not implemented.*	0.00
Latitude	Latitude from last GPS fix. Note that the sign and direction (N/S) can be redundant.	40.748938 N
Longitude	Longitude from last GPS fix. Note that the sign and direction (E/W) can be redundant.	-73.942496 W
Altitude	Altitude from last GPS fix. Note that it is possible to have a lat/long but no altitude because a 3D fix takes longer than a 2D fix.	77.400002
Satellites	Satellites used for last GPS fix	6
Fix Quality	GPS fix quality (0, 1, 2 = Invalid, GPS, DGPS). DGPS does not apply with the current hardware.	1

The "images" folder and "log.txt" can be safely deleted at any time; they will be automatically recreated. However, the following files should always be present on the SD card and should not be modified (other than "cam_cfg.txt"):

bootcode.bin
cam_cfg.txt
config.txt
fixup.dat
fixup_x.dat
kernel.img
libraspistill.a
start.elf
start_x.elf

Other Tips/Notes:

- The SD card can be safely removed or inserted without powering down the camera provided it's not actively trying to save an image. This allows checking data without restarting a run, but should still be done with caution.
- It is highly recommended to use tweezers to pull out the SD card.
  The D-cell case containing the Pi and nrf52840 **cannot** slide out of the camera body with the camera and GPS modules connected. Trying to pull it will likely pull the ribbon cable out of the Pi, requiring the camera and GPS modules to be disconnected and removed, the cable reattached to the Pi, and then fed back through the body. (In other words, an annoying amount of work.)