

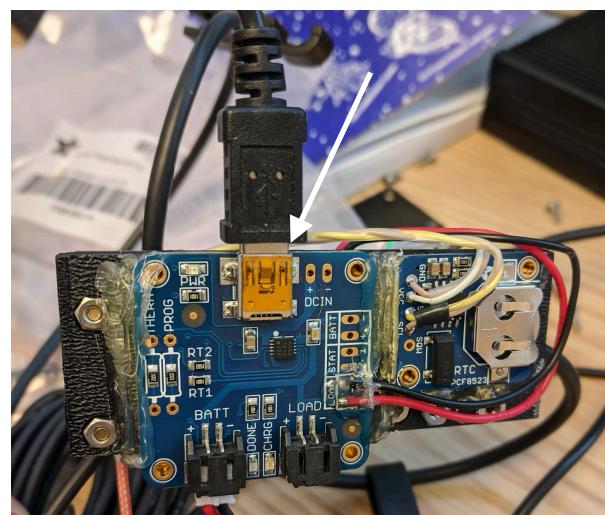
## Reading and Parsing the Tracker Data

1. Open the container by unscrewing the four screws on each end. Push out the main board enough to reveal the micro SD card. You may have to open both ends and push from the other side.
2. Remove micro SD card (Pictured). **You will need a micro SD card reader in order to retrieve the data.**
3. Insert the micro SD card into a card reader and retrieve the data file entitled “DATALOG.txt”.
4. Copy the following script on Github: <https://github.com/mrsolt/Mu2e/blob/master/GPSTracker/GPSParser.py>
5. Run the with the following command: **python GPSParser.py <output base name> DATALOG.TXT**
6. This will create 5 output files. <>.acc.png, <>.hum.png, <>.temp.png, and <>.speed.png are the output histograms for acceleration, humidity, temperature, and speed, respectively. <>.csv is the file to be used to track the GPS coordinates in the next step.
7. Upload the output CSV file into Google Earth Pro (free to download if you don’t have it). Make sure the check mark next to the data file in the “Places” list is checked so the data appears on the map. Screen shot the map and save the image.



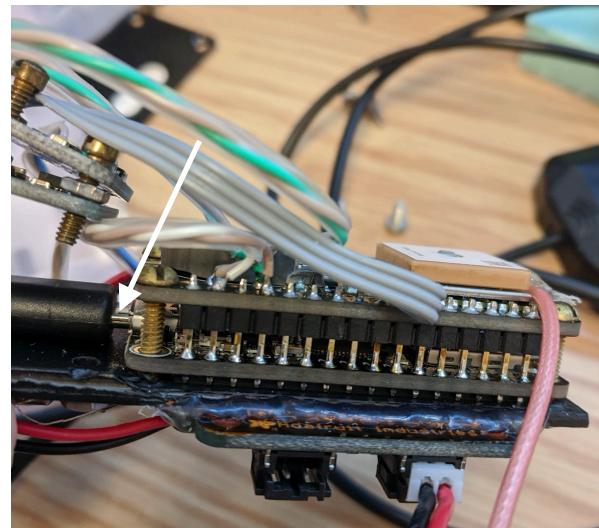
## Setting the Tracker for Data Collection

1. Open the container as described above.
2. Charge the batteries using the micro-usb adapter on the USB LiPoly Charger as shown in the figure to the right. It takes about 12 hours to charge and the “DONE” light will turn green.
3. Remove the humidity sensor and accelerometer to access the USB mini port as shown in the figure below. This will allow communication between the logger and your computer for uploading new images. Connect this to your computer.
4. Remove the micro SD card on the logger and insert it into a card reader. Delete (after analyzing!) the “DATALOG.txt” file on the



micro SD card. The file will be remade when data collection starts. Reinsert the SD card into the logger.

5. Download the Arduino IDE: <https://wwwarduino.cc/en/software/>
6. Install the following drivers using the Arduino IDE - **Adafruit Si7021 Library**, **Adafruit Unified Sensor Driver**, **Adafruit GPS Driver**, **RTClib library**. Click Sketch -> Include Library -> Manage Libraries and search for each driver, then click "Install".
7. This system uses the **Adafruit Feather M0 Adalogger Controller** (not an Arduino) so additional packages must be installed in order to select the correct board. Click Arduino -> Preferences. Under "Additional Boards Manager URLs:" copy and paste the following URL: [https://adafruit.github.io/arduino-board-index/package\\_adafruit\\_index.json](https://adafruit.github.io/arduino-board-index/package_adafruit_index.json) . Click "OK".
8. Click Tools -> Board: -> Boards Manager. Install both the Arduino SAMD and Adafruit SAMD packages by searching for them and clicking "Install". "Select Adafruit Feather M0" under Tools -> Board. Also, select the port that corresponds to the connection from the logger to your computer.
9. Download the following two files for images you will send to the logger: <https://github.com/mrsolt/Mu2e/blob/master/GPSTracker/ShippingMonitor/ShippingMonitor.ino> and <https://github.com/mrsolt/Mu2e/blob/master/GPSTracker/pcf8523/pcf8523.ino>
10. Sometimes, the clock on the logger will be out of sync with the RTC (you can check the output). If this happens, simply upload the "pcf8523" image to the logger (just click upload in the Arduino IDE) to sync the logger clock to the RTC. Note the GPS syncs with satellite clocks and cannot be reset.
11. Set the following line in *ShippingMonitor* file to the desired start logging time: **DateTime wakeupTime(2021, 2, 24, 9, 0, 0); //Time to "wake-up" and start logging**. For instance, this sets the start time for February 24, 2021 at 9am (according to the logger clock!). This will save some battery life and minimize collecting useless data. Note: The data collection doesn't always start and I don't understand why. However, the battery life is sufficient such that you can just start collecting data the night before a shipment to verify it is working.
12. Upload the *ShippingMonitor* image to the logger. You can test this image by setting the start time for a few minutes into the future and see if it begins collecting data when you expect.<sup>1</sup>
13. Once it's charged and the desired time is set, put the container back together and you are all set to go.



<sup>1</sup> Note: Often times the logger can get stuck in a weird state in which the port can no longer be found by your computer. In these cases, there is a reset button on the logger near the USB mini port. This will fix most connection issues.