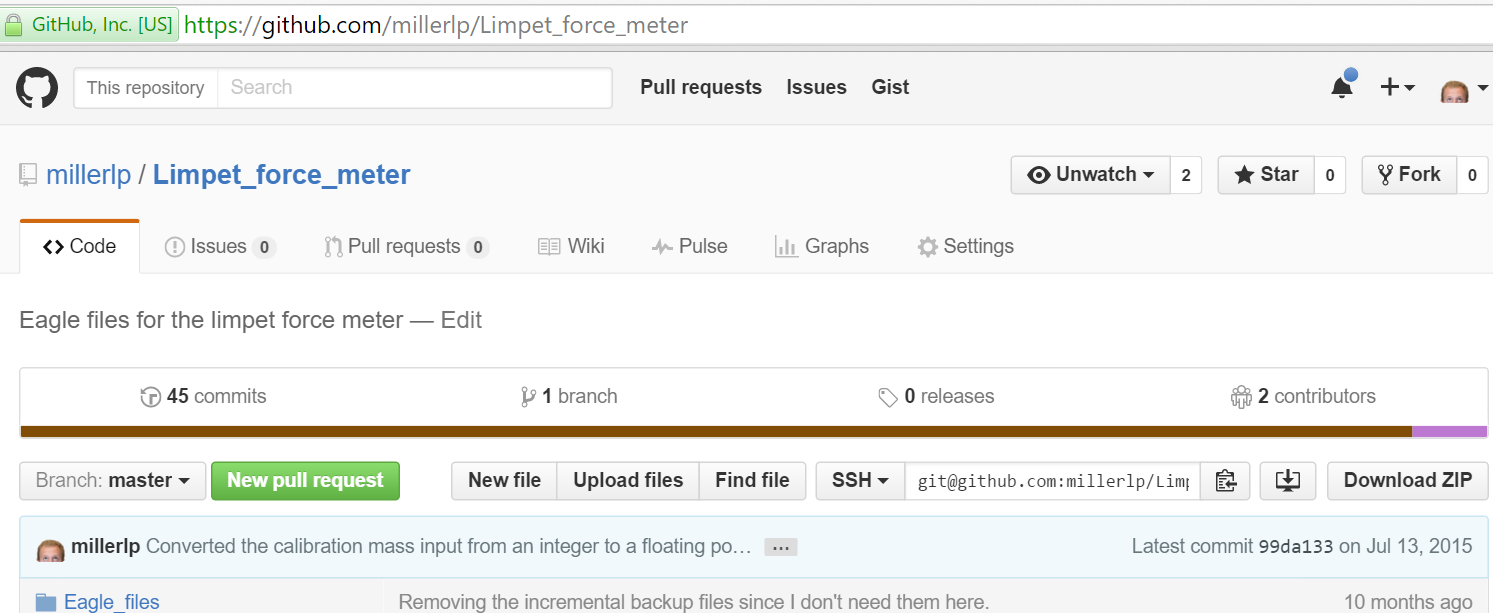
# Getting the source files

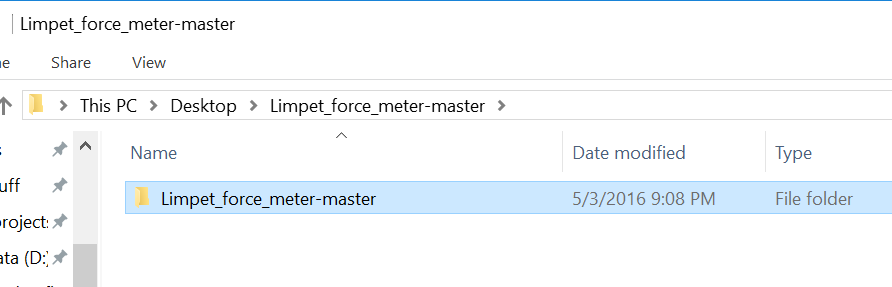
## Limpet\_force\_meter

The main limpet force meter files are found at:

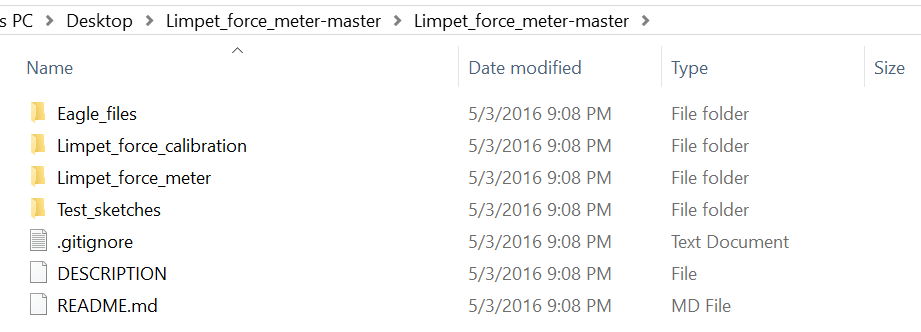
<https://github.com/millerlp/Limpet_force_meter>



To download the files, click the Download ZIP button on the right side of the page. Extract the ZIP file that downloads. Inside the Limpet\_force\_meter-master folder you should find another folder also called Limpet\_force\_meter-master (shown below)

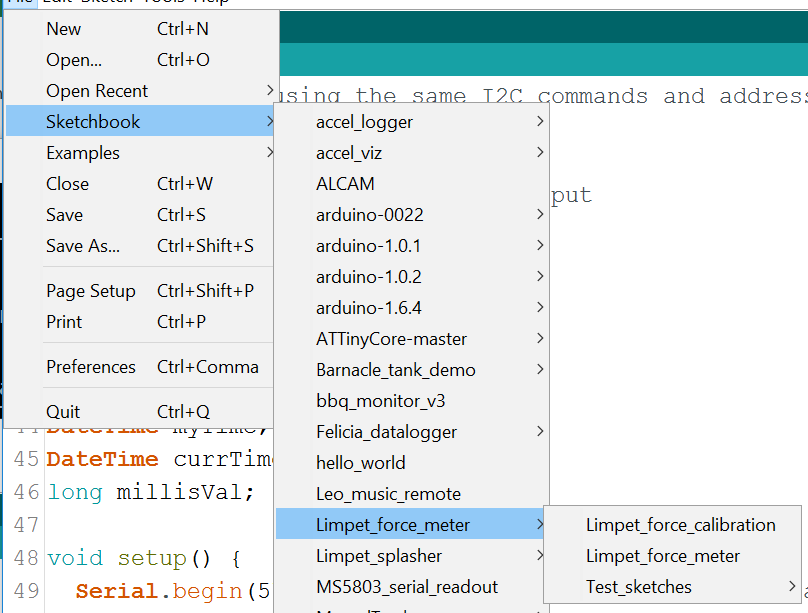


Inside *that* folder, there are several folders that contain the actual programs. Rename the folder ‘Limpet\_force\_meter-master’ to just be ‘Limpet\_force\_meter’.



Take the renamed Limpet\_force\_meter folder, which contains all the subfolders shown above, and put it in your Arduino folder (wherever that is).

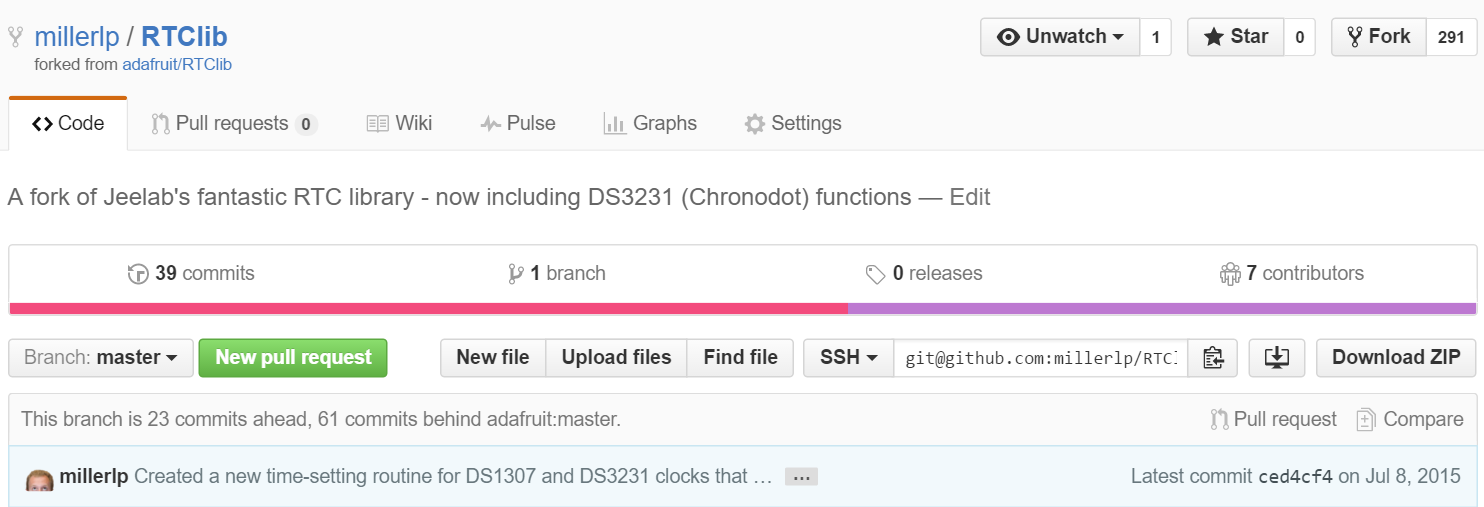
If you close and restart the Arduino program, it should detect the Limpet\_force\_meter folder. Go to File>Sketchbook and look for Limpet\_force\_meter. If the Limpet\_force\_calibration and Limpet\_force\_meter entries are there, the installation worked.



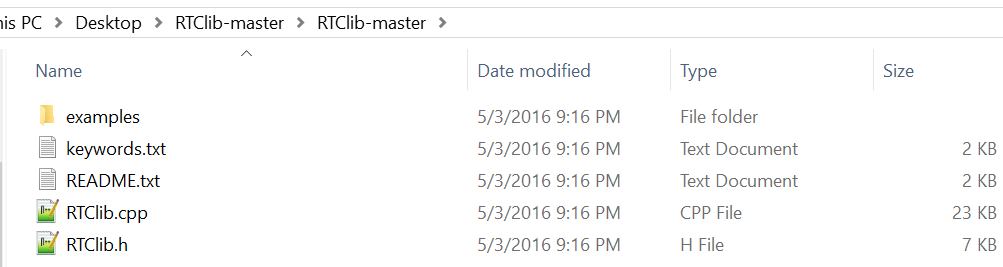
## RTClib

In addition to the Limpet\_force\_meter files, you also need a ‘library’ file for the real time clock (abbreviated RTC). The library file is called RTClib, and can be found at <https://github.com/millerlp/RTClib>

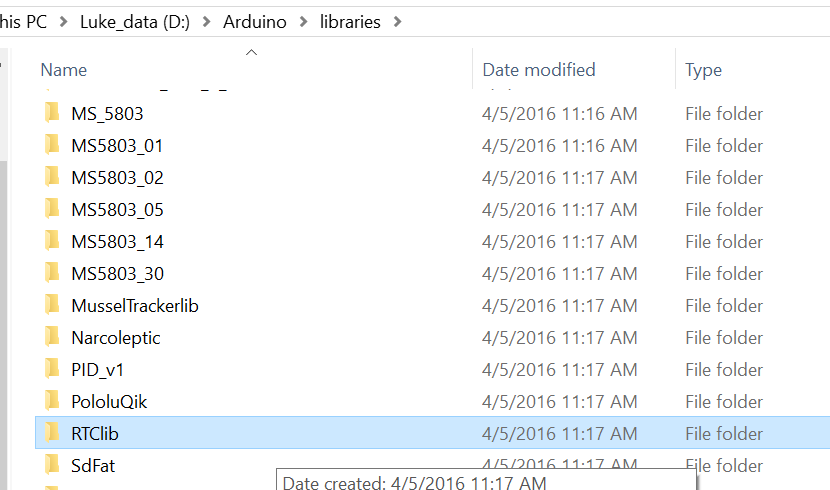
Click on the Download ZIP button on the right side of the page.



That will download a zip file. Extract the zip file, which will create a folder within a folder called RTClib-master, and that folder will contain several files like 'keywords.txt', 'README.txt' etc.



Rename the folder RTClib-master to just RTClib. Take that entire folder and put it in your Arduino>libraries directory, where the old RTClib folder currently sits. Overwrite the old RTClib files if there are any present.

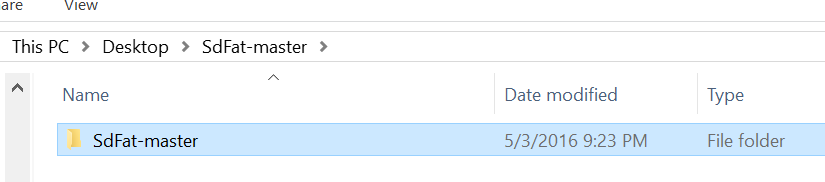


## SD card library

You also need a ‘library’ file to make the micro SD card work. That library file can be found at

<https://github.com/millerlp/SdFat>

Click the Download ZIP button on the right side of the screen. Extract the ZIP file that is downloaded. Once again, you will get a folder called SdFat-master, and inside that will be another folder called SdFat-master. Rename that inner folder to just be ‘SdFat’.



Copy the renamed ‘SdFat’ folder to your Arduino>libraries directory, like you did with the RTClib file folder.

After doing this, close and restart the Arduino software.

**Note: Arduino software v1.6.8 (and 1.6.6, 1.6.7) appear to have introduced a bug where you can only compile or upload a program once for the Due, after which you may get an error similar to the following:**

GetFileAttributesEx D:\Arduino\libraries\SdFat/SdInfo.h D:\Arduino\libraries\SdFat/SdSpi.h: The filename, directory name, or volume label syntax is incorrect.

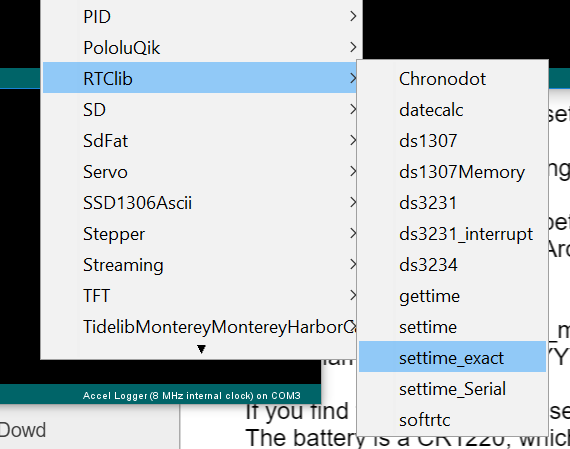
Error compiling for board Arduino Due (Programming Port).

**If this occurs, you can try closing the Arduino program, re-opening it, and then immediately uploading your desired program once. The software may allow that first upload to proceed properly, but fail on the 2nd or later attempts. Closing and reopening Arduino seems to allow the upload to happen again, for the first try.**

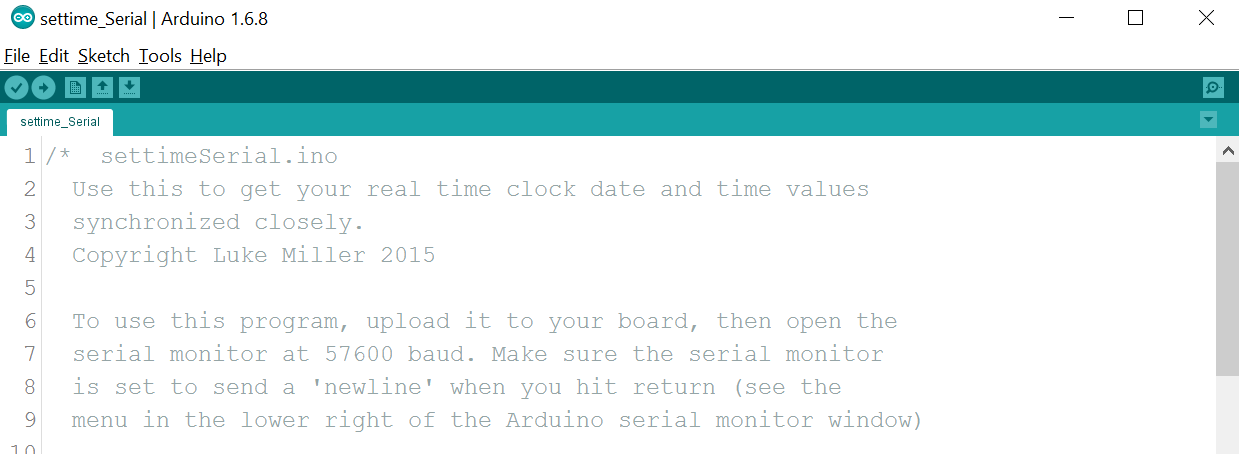
# To reset the clock:

Try opening the arduino software, and plug in the USB cable to the force meter.

In Arduino, go to File>Examples>RTClib>**settime\_Serial**



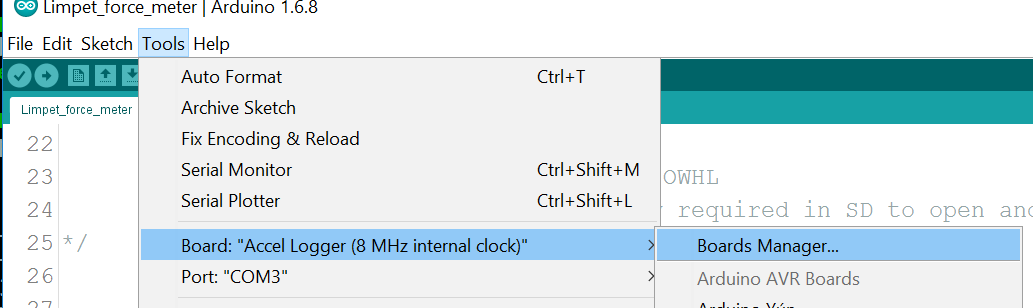
That should open up a new window with the settime\_Serial program.



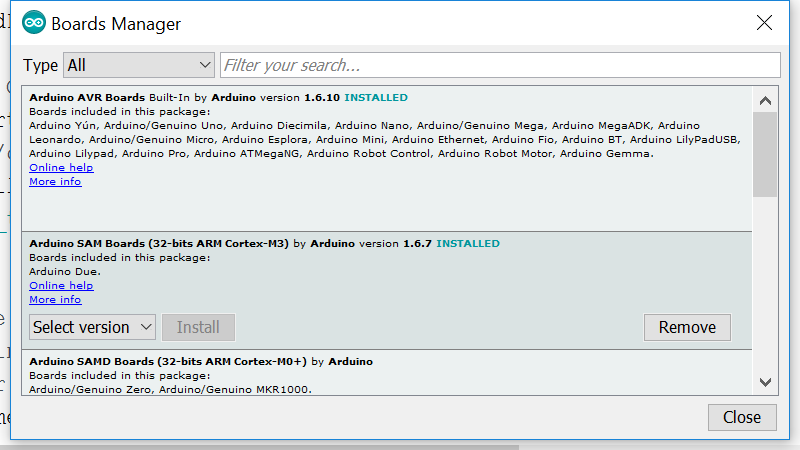
Look at the bottom of the Arduino window, it should say "**Arduino Due (Programming Port)**". If so, good. Otherwise go to Tools>Board and choose **Arduino Due (Programming Port)**. If you find that option, you can skip the next two steps below.



If you can’t even find the **Arduino Due (Programming Port)** option in Tools>Board menu, then click on the Boards Manager link at the top of the Tools>Board menu



That will bring up the Boards Manager window (shown below). Find the entry titled **Arduino SAM Boards (32-bits ARM Cortex-M3),** and click on it. An Install button should appear. Click Install and let it download/install the files.



After doing the above, you should be able to go back to the Tools>Board menu and find **Arduino Due (Programming Port)** at the bottom of the list. Do not choose Arduino Due (Native USB Port), that is the wrong option.

On a Mac, go to Tools>Port and choose the option that looks most similar to **/dev/tty.usbserial,**if that isn't already showing in the bottom right of the Arduino window (see picture above). On a PC, go to Tools>Port and look for the COM ports. It will be named COM + a number, like COM1 (most common) or some other value like COM3.



Figure . Mac version

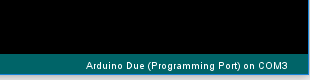


Figure . Windows version

Try pressing the upload button (which looks like a right-arrow, upper left of Arduino window).

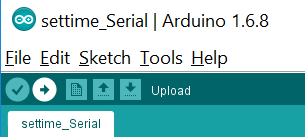


Figure . The upload button is the right-arrow.

If the upload works, you should get a little success message at the bottom of the Arduino window. If you get an error message (could not communicate with Arduino), try hitting Upload again, since the program is flaky some times.

If successful, now click the little icon (a magnifying glass) in the upper right of the Arduino window to open the Serial Monitor window.

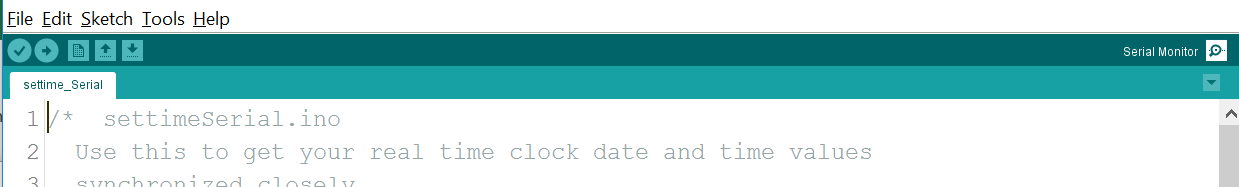


Figure . Serial monitor is opened via the little magnifying glass icon in the upper right.

A new Serial Monitor window should pop up. The menu in the bottom left should be set to 57600 baud.

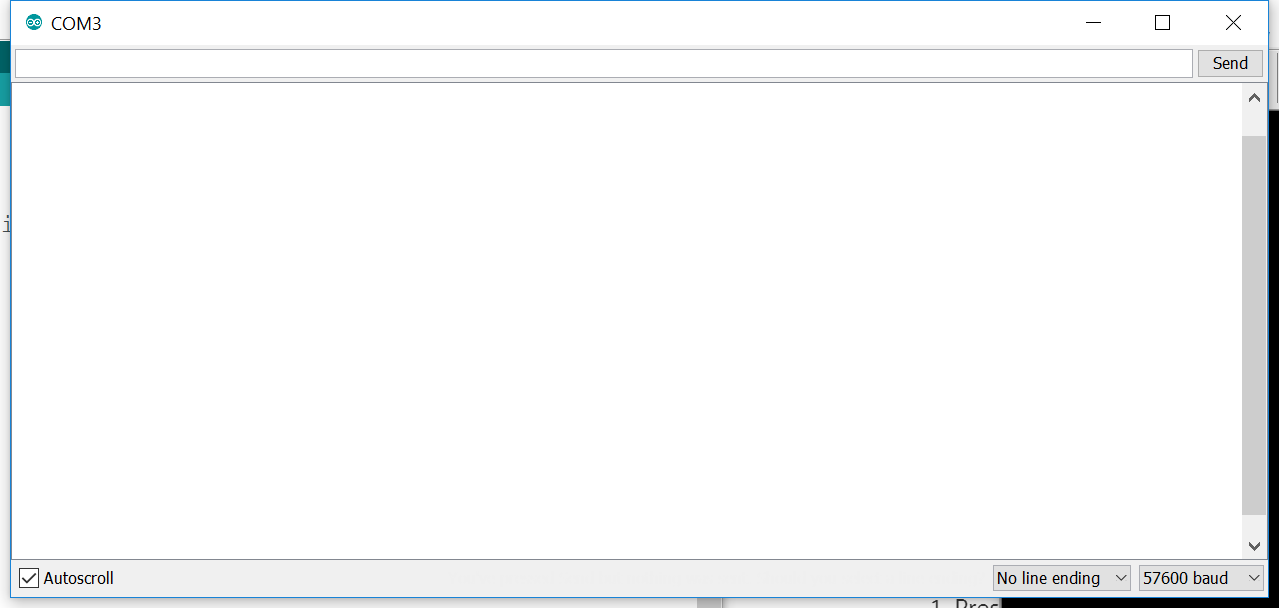




Figure . The serial monitor window.

As the board boots up, it should say:  
 "Enter a new date and time in the following format all on one line"

"YYYY MM DD HH MM SS"

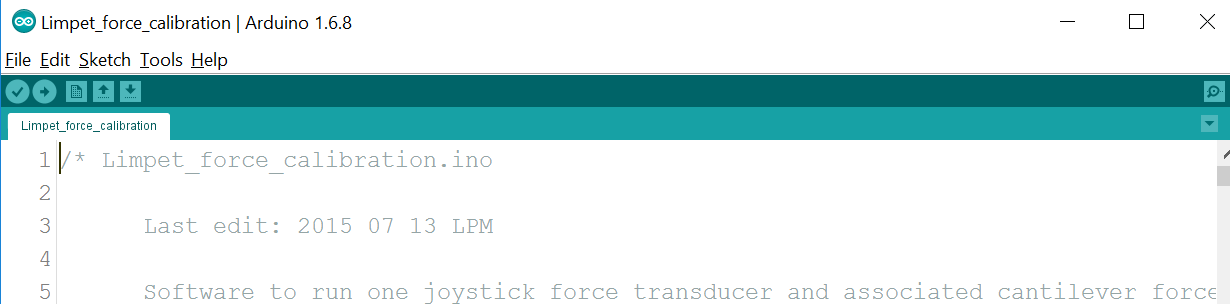
"and hit enter when ready to set time"

In the text field at the top of the window, enter the year, month, day, hour minute second as specified there, with a space between each value. Enter a seconds value that is 10 or 20 seconds ahead of right now (based off your phone or computer time). As that time finally arrives, hit Enter and the new date/time should be programmed into the Arduino. If it is successful, the new date/time should start appearing in the Serial Monitor window. If you mistype something, you can try again by closing the Serial Monitor window and reopening it (this should reboot the arduino board).

Once the date and time are set correctly, you need to put the real force measuring program back on the arduino.

## Loading the data collecting and calibration program

The program **Limpet\_force\_calibration** can serve as both the main data collection program and the calibration program. To load it onto the force meter, go to the File>Sketchbook>Limpet\_force\_meter>**Limpet\_force\_calibration.** This should open a new Arduino window.



Press the Upload button (right-arrow in the upper left of the Arduino window). As usual, the USB cable must be plugged into the force meter’s USB port for this to work.

After the **Limpet\_force\_calibration** file is uploaded, the force meter will start recording data whenever it is powered up.

To enter the calibration mode (leaving the normal data collection mode) open the Serial Monitor by clicking the magnifying glass icon in the upper right of the Arduino window, while the force meter is connected by USB.

This will open the Serial Monitor window. Now hold down Button1 on the force meter circuit board (purple board) for at least 3 seconds. The red and green LEDs will flash 5 times, which means you have entered Calibration mode. The serial monitor will prompt you to choose which axis you want to calibrate. Follow the instructions on the Serial Monitor, choosing an axis to calibrate, and then the mass (in grams) that you will hang on the sensor. If you want to quit, enter ‘q’ when the Serial Monitor asks you which axis to calibrate.

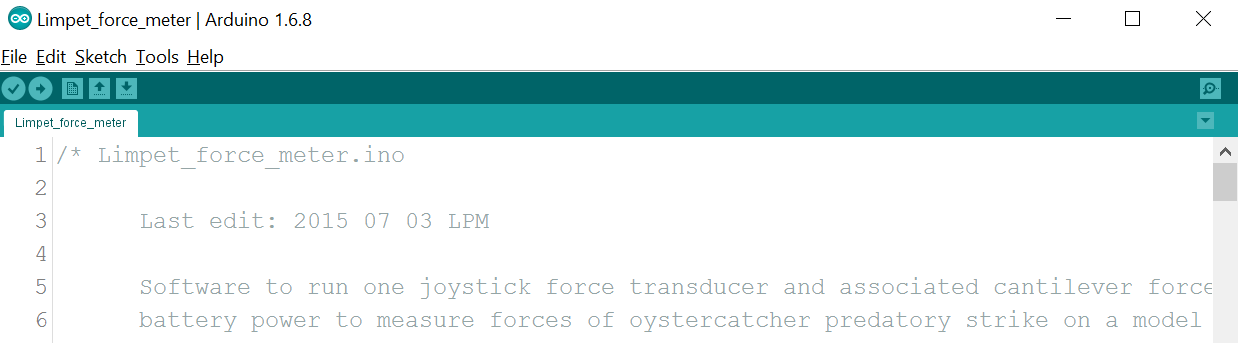
When you quit calibration mode, or reboot the force meter, it will re-enter normal data collection mode by default. The data files will be named in the format YYYYMMDD\_HHMM\_00.CSV, where YYYY is year, MM is month, DD is day, HH is hour, MM is minute, and 00 will be a numeric counter if there are multiple files created in the same minute. The date-time code is based on the starting time of that file, and the time is obtained from the onboard real time clock on the force meter (so the clock must be set before attempting to collect data, see that process described above). Data files are recorded for roughly 5 minutes before the current file is closed and new file is created.

## Reloading the force measuring program:

The program **Limpet\_force\_meter** only records data to the SD card, it does not have a calibration mode. I think in most cases it makes sense to just upload the **Limpet\_force\_calibration** program (above) and use it as the normal data collection program (and calibration program).

If for some reason you want to just use the Limpet\_force\_meter program, go to File>Sketchbook>Limpet\_force\_meter>**Limpet\_force\_meter**  (you'll also see the Limpet\_force\_calibration file there)

That should open up a new Arduino window with the Limpet\_force\_meter program in it.



Hit the Upload button (round right-arrow button in upper left), and this should upload the Limpet\_force\_meter program to the board.