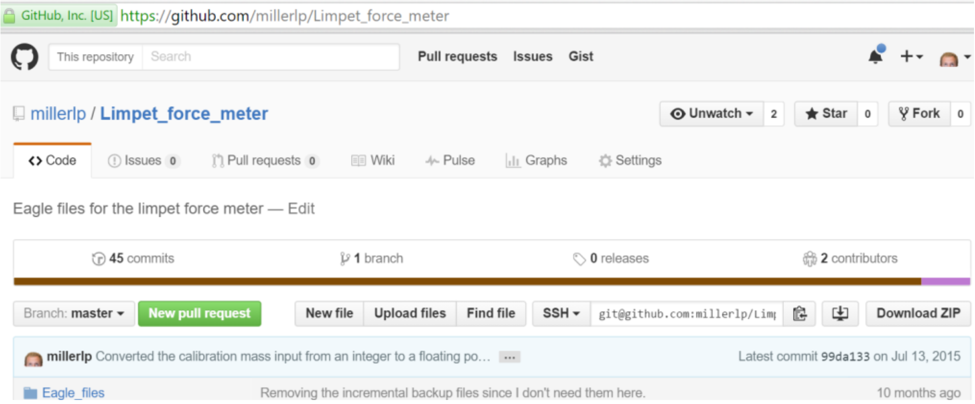
Getting the source files

Limpet_force_meter

The main limpet force meter files are found at:

https://github.com/millerlp/Limpet_force_meter

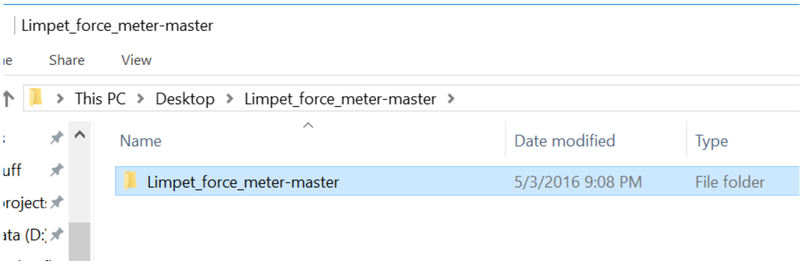


Eagle files for the limpet force meter — Edit

millerlp Converted the calibration mass input from an integer to a floating po… Latest commit 99da133 on Jul 13, 2015

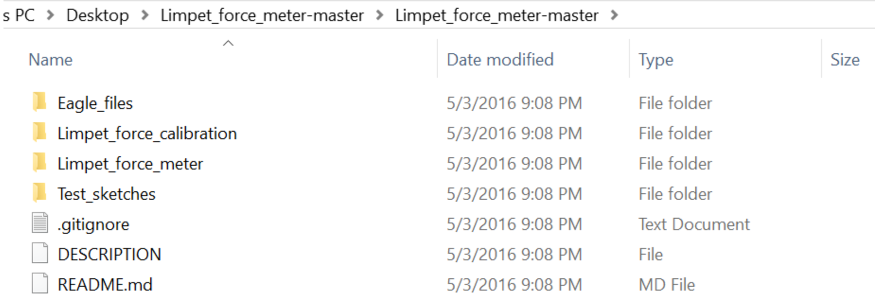
Eagle_files	Removing the incremental backup files since I don't need them here.	10 months ago

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)


Limpet_force_meter-master
Share
View
This PC > Desktop > Limpet_force_meter-master

Name	Date modified	Type
Limpet_force_meter-master	5/3/2016 9:08 PM	File folder

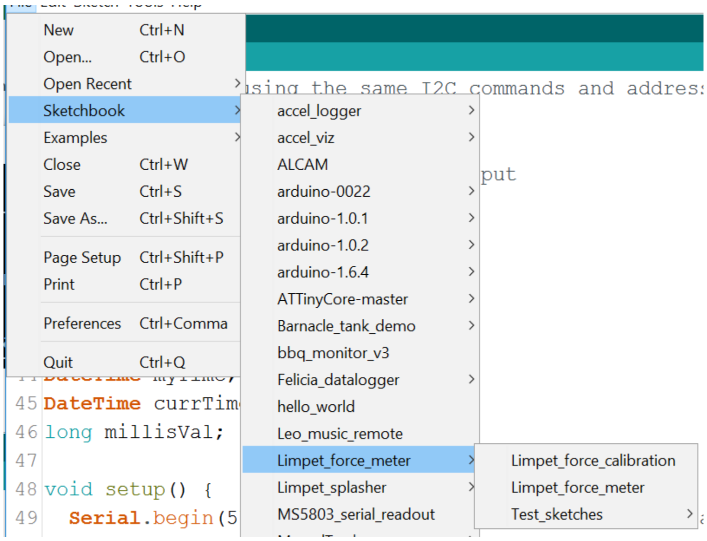
Inside *that* folder, there are several folders that contain the actual programs. Rename the folder 'Limpet_force_meter-master' to just be 'Limpet_force_meter'.


s PC > Desktop > Limpet_force_meter-master > Limpet_force_meter-master

Name	Date modified	Type	Size
Eagle_files	5/3/2016 9:08 PM	File folder	
Limpet_force_calibration	5/3/2016 9:08 PM	File folder	
Limpet_force_meter	5/3/2016 9:08 PM	File folder	
Test_sketches	5/3/2016 9:08 PM	File folder	
.gitignore	5/3/2016 9:08 PM	Text Document	
DESCRIPTION	5/3/2016 9:08 PM	File	
README.md	5/3/2016 9:08 PM	MD File	

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.



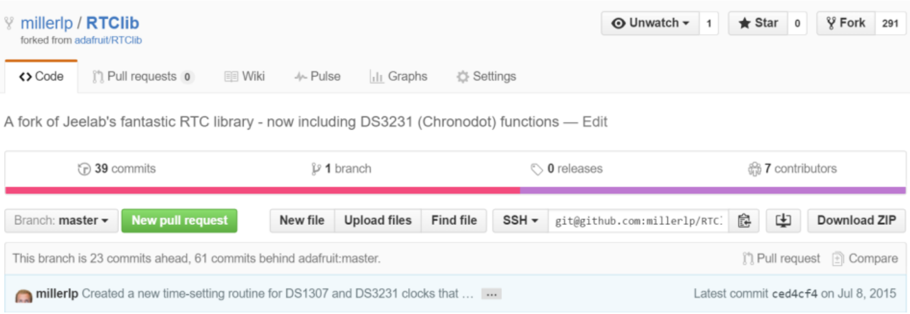
```
using the same I2C commands and addres
put
DateTime currTim
long millisVal;

void setup() {
  Serial.begin(5
```

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.

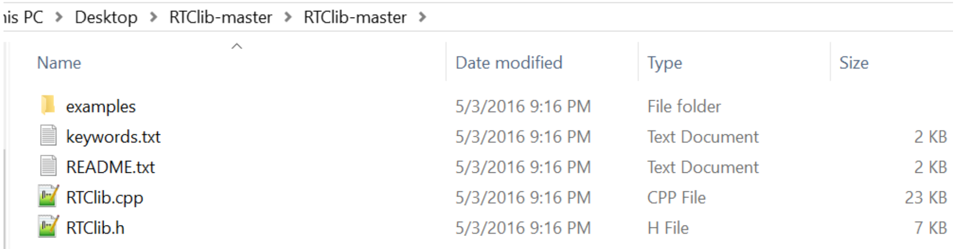


A fork of Jeelab's fantastic RTC library - now including DS3231 (Chronodot) functions — Edit

This branch is 23 commits ahead, 61 commits behind adafruit:master.

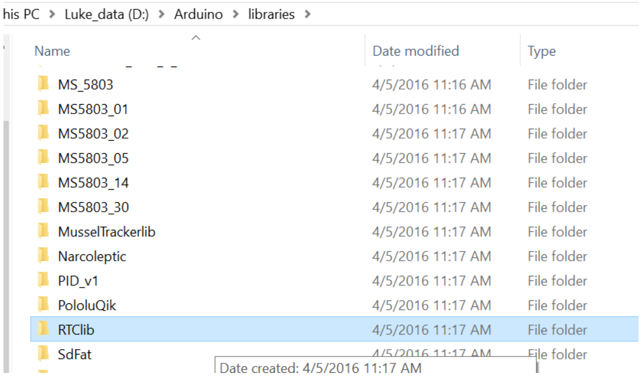
millerlp Created a new time-setting routine for DS1307 and DS3231 clocks that ... Latest commit ced4cf4 on Jul 8, 2015

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.



Name	Date modified	Type	Size
examples	5/3/2016 9:16 PM	File folder	
keywords.txt	5/3/2016 9:16 PM	Text Document	2 KB
README.txt	5/3/2016 9:16 PM	Text Document	2 KB
RTClib.cpp	5/3/2016 9:16 PM	CPP File	23 KB
RTClib.h	5/3/2016 9:16 PM	H File	7 KB

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.



his PC > Luke_data (D:) > Arduino > libraries >

Name	Date modified	Type
MS_5803	4/5/2016 11:16 AM	File folder
MS5803_01	4/5/2016 11:16 AM	File folder
MS5803_02	4/5/2016 11:17 AM	File folder
MS5803_05	4/5/2016 11:17 AM	File folder
MS5803_14	4/5/2016 11:17 AM	File folder
MS5803_30	4/5/2016 11:17 AM	File folder
MusselTrackerlib	4/5/2016 11:17 AM	File folder
Narcoleptic	4/5/2016 11:17 AM	File folder
PID_v1	4/5/2016 11:17 AM	File folder
PololuQik	4/5/2016 11:17 AM	File folder
RTClib	4/5/2016 11:17 AM	File folder
SdFat	4/5/2016 11:17 AM	File folder

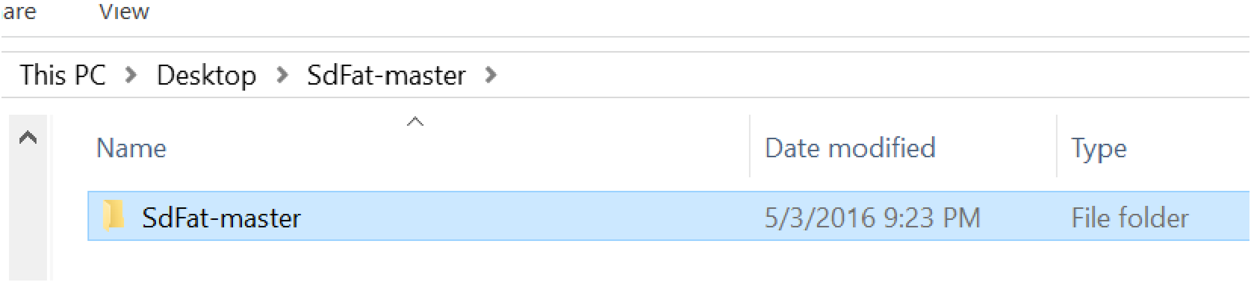
Date created: 4/5/2016 11:17 AM

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'.



This PC > Desktop > SdFat-master >

Name	Date modified	Type
SdFat-master	5/3/2016 9:23 PM	File folder

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:

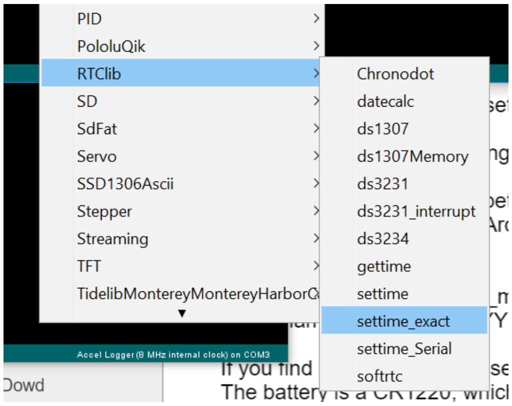
```
GetFileAttributesEx D:\Arduino\libraries\SdFat/SdInfo.h
D:\Arduino\libraries\SdFat/SdSpi.h: The filename, directory name, or volume
label syntax is incorrect.
Error compiling for board Arduino Due (Programming Port).
```

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.

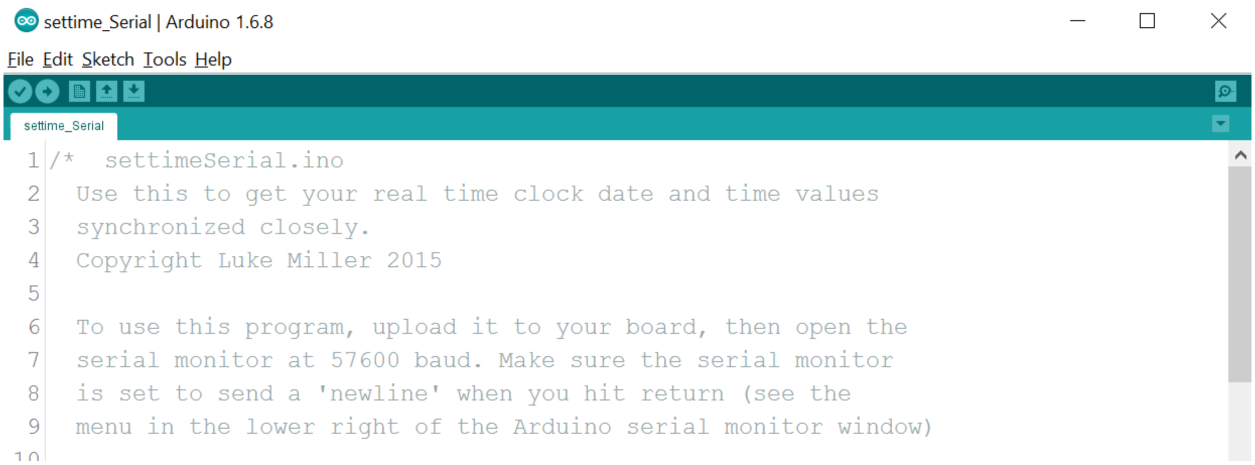
To reset the clock:

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

In Arduino, go to File>Examples>RTClib>**settime_Serial**



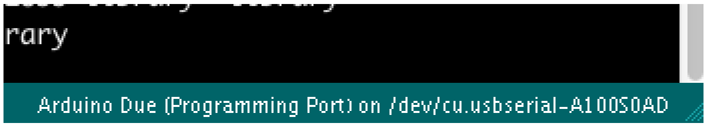
That should open up a new window with the settime_Serial program.



```
/*  settimeSerial.ino
  Use this to get your real time clock date and time values
  synchronized closely.
  Copyright Luke Miller 2015

  To use this program, upload it to your board, then open the
  serial monitor at 57600 baud. Make sure the serial monitor
  is set to send a 'newline' when you hit return (see the
  menu in the lower right of the Arduino serial monitor window)
```

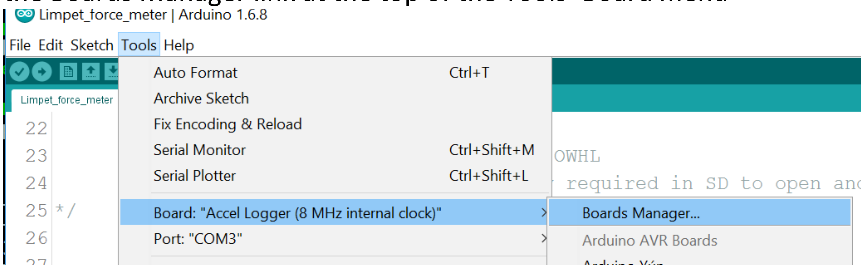
Look at the bottom of the Arduino window, it should say "**Arduino Due (Programming Port)**". If so, good. Otherwise go to Tools>Board and choose **Arduino Due (Programming Port)**. If you find that option, you can skip the next two steps below.



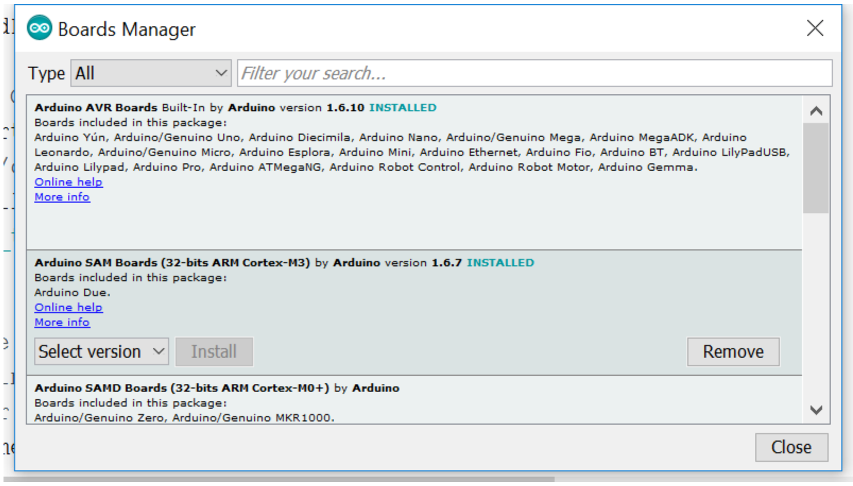
```
rary
```

Arduino Due (Programming Port) on /dev/cu.usbserial-A100S0AD

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


Limpet_force_meter | Arduino 1.6.8
File Edit Sketch Tools Help
Limpet_force_meter
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
Board: "Accel Logger (8 MHz internal clock)"
Port: "COM3"
Boards Manager...
Arduino AVR Boards
OWHL
required in SD to open anc
*/

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.



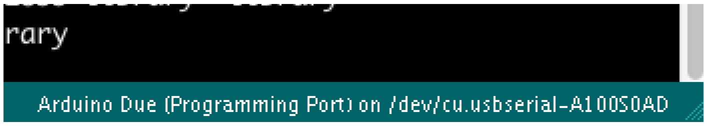
Arduino AVR Boards Built-In by **Arduino** version **1.6.10** INSTALLED
Boards included in this package:
Arduino Yún, Arduino/Genuino Uno, Arduino Diecimila, Arduino Nano, Arduino/Genuino Mega, Arduino MegaADK, Arduino Leonardo, Arduino/Genuino Micro, Arduino Esplora, Arduino Mini, Arduino Ethernet, Arduino Fio, Arduino BT, Arduino LilyPadUSB, Arduino Lilypad, Arduino Pro, Arduino ATMegaNG, Arduino Robot Control, Arduino Robot Motor, Arduino Gemma.
Online help
More info

Arduino SAM Boards (32-bits ARM Cortex-M3) by **Arduino** version **1.6.7** INSTALLED
Boards included in this package:
Arduino Due.
Online help
More info

Arduino SAMD Boards (32-bits ARM Cortex-M0+) by **Arduino**
Boards included in this package:
Arduino/Genuino Zero, Arduino/Genuino MKR1000.

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.

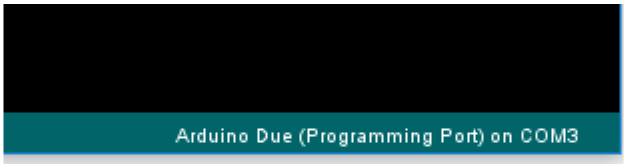
On a Mac, go to Tools>Port and choose the option that looks most similar to **/dev/tty.usbserial,** if that isn't already showing in the bottom right of the Arduino window (see picture above). On a PC, go to Tools>Port and look for the COM ports. It will be named COM + a number, like COM1 (most common)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



```
rary
```

Arduino Due (Programming Port) on /dev/cu.usbserial-A100S0AD

Figure 1. Mac version



Arduino Due (Programming Port) on COM3

Figure 2. Windows version

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

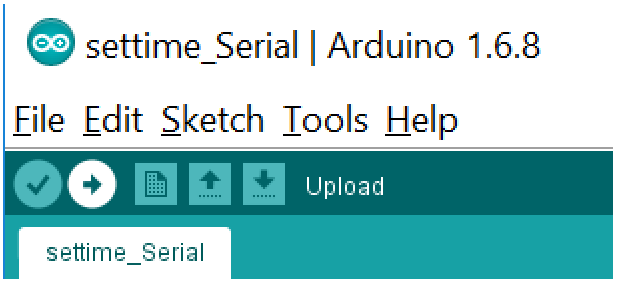
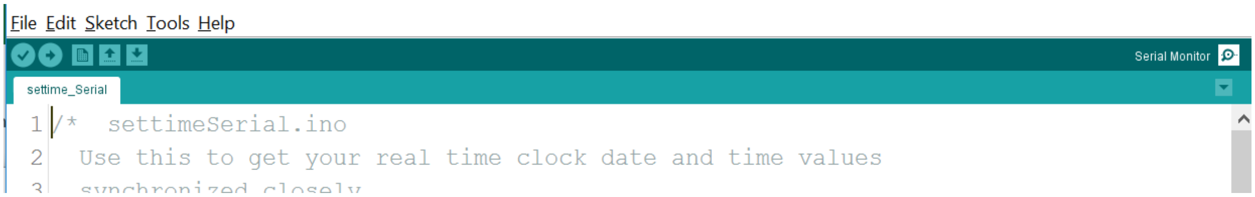


Figure 3. The upload button is the right-arrow.

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.

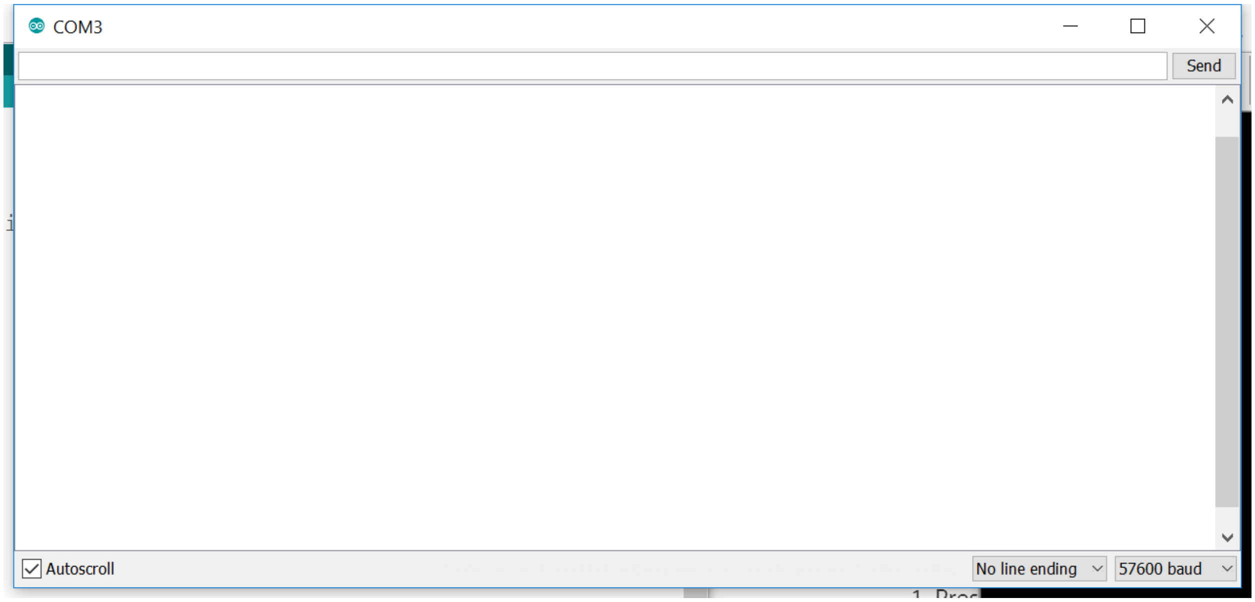
If successful, now click the little icon (a magnifying glass) in the upper right of the Arduino window to open the Serial Monitor window.



```
/*  settimeSerial.ino
 Use this to get your real time clock date and time values
 synchronized closely
```

Figure 4. 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.



☑ Autoscroll   No line ending   57600 baud

Figure 5. 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"

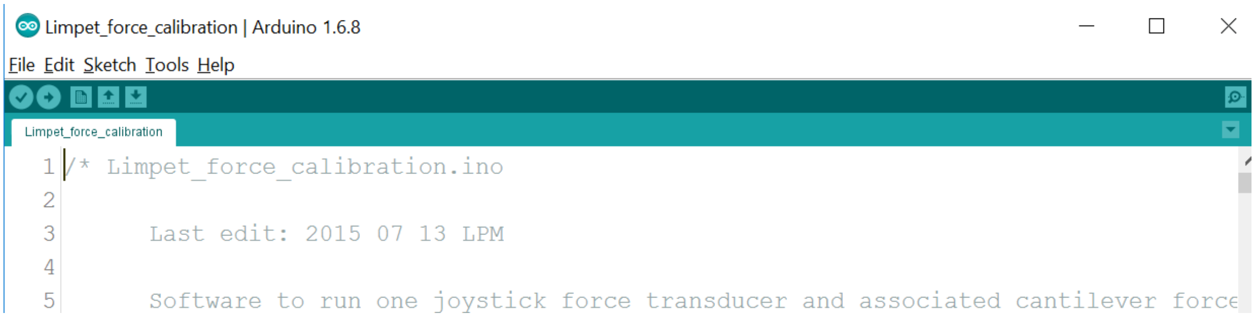
In the text field at the top of the window, enter the year, month, day, hour minute second as specified there, with a space between each value. Enter a seconds value that is 10 or 20 seconds ahead of right now (based off your phone or computer time). As that time finally arrives, hit Enter and the new 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.


Limpet_force_calibration | Arduino 1.6.8
File Edit Sketch Tools Help
Limpet_force_calibration

```
/* Limpet_force_calibration.ino

    Last edit: 2015 07 13 LPM

    Software to run one joystick force transducer and associated cantilever force
```

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.

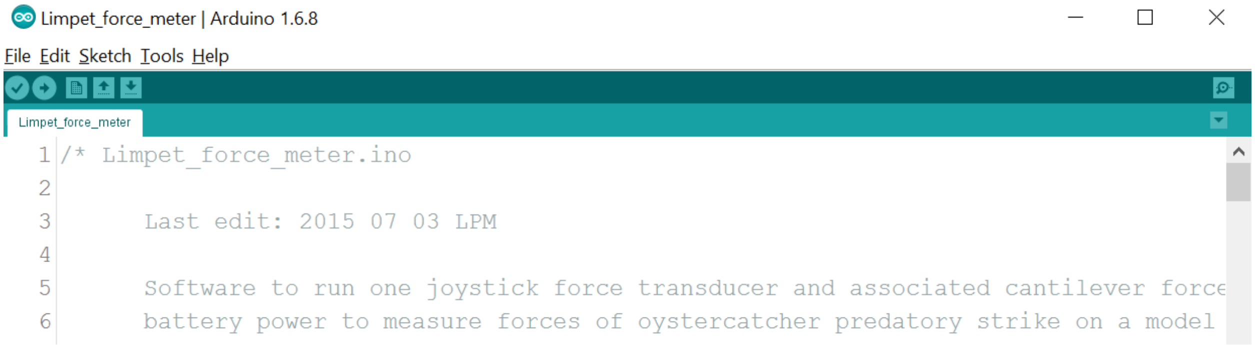
When you quit calibration mode, or reboot the force meter, it will re-enter normal data collection mode by default. The data files will be named in the format YYYYMMDD_HHMM_00.CSV, where YYYY is year, MM is month, DD is day, HH is hour, MM is minute, and 00 will be a numeric counter if there are multiple 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.


Limpet_force_meter | Arduino 1.6.8
File Edit Sketch Tools Help
Limpet_force_meter

```
/* Limpet_force_meter.ino

    Last edit: 2015 07 03 LPM

    Software to run one joystick force transducer and associated cantilever force
    battery power to measure forces of oystercatcher predatory strike on a model
```

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