

Assembly Instructions

GCODE Mini

Parts list

Part	Quantity	Cost
Black Acrylic (laser cut)	2x1 feet	5\$
GCODE Mini PCB	1	
1cm corner bends	8	
5mmx2cm bolts, nuts	16	
3mmx1cm bolts, nuts	12	
3mm washers	10	
Red, Black single-strand wire	20cm each	
HUDZ 5V Buzzer	1	
47E resistor	1	
1k resistor	2	
3.3k resistor	1	
10k resistor	2	

100k resistor	1	
1 uF electrolytic capacitor	1	
3/8" square staggered lead variable resistor: 10k	1	
0.1" side-entry screw terminal block: 2-pin	4	
5mm red LED	1	
BPW34 photodiode	1	
Male pin headers	32	
Arduino Uno	1	
1.5m USB B cable	1	

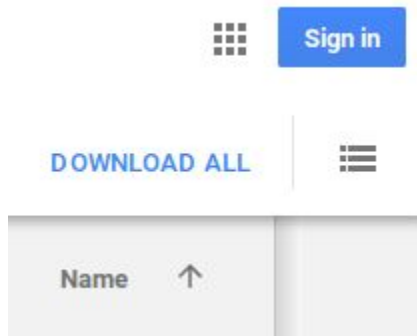
You'll also need two flat-tip screwdrivers (5mm and 2mm long tips), some pliers, a soldering iron, and some solder flux.

Steps

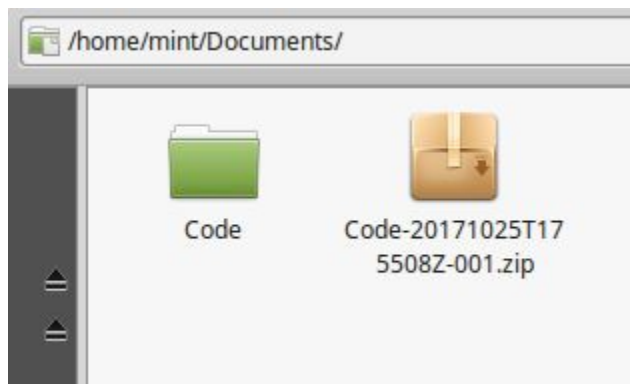
Get the code

Download and unzip this Google Drive folder: <https://drive.google.com/open?id=0B1wo-dUMehCUMnI1dGhmaGpSVzA>

Protip: Hit "**Download All**" to get the whole folder at once



We will call the folder to which you extract this as the “Code” folder.



Arduino code upload

First, we'll have to upload the code to the Arduino, because the PCB will damage the Arduino board¹ if this is not done first.

1. Obtain the Arduino IDE for your OS (Windows/Linux²) from <https://www.arduino.cc/en/Main/Software>. Don't worry, it's not scary like some other IDEs.

Download the Arduino IDE



ARDUINO 1.8.5

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer

Windows ZIP file for non admin install

Windows app [Get](#) 

Mac OS X 10.7 Lion or newer

Linux 32 bits

Linux 64 bits

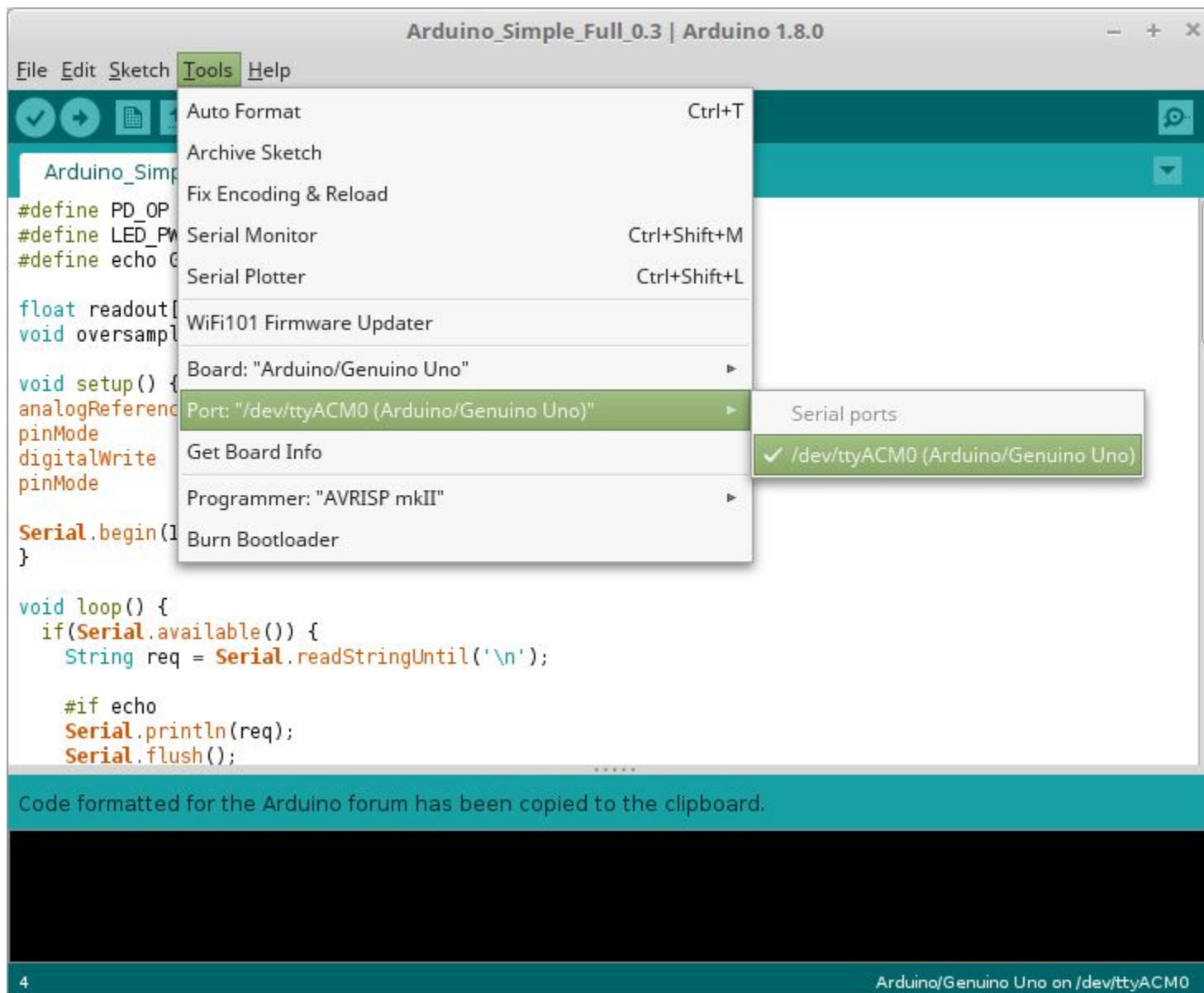
Linux ARM


[Release Notes](#)

[Source Code](#)

[Checksums \(sha512\)](#)

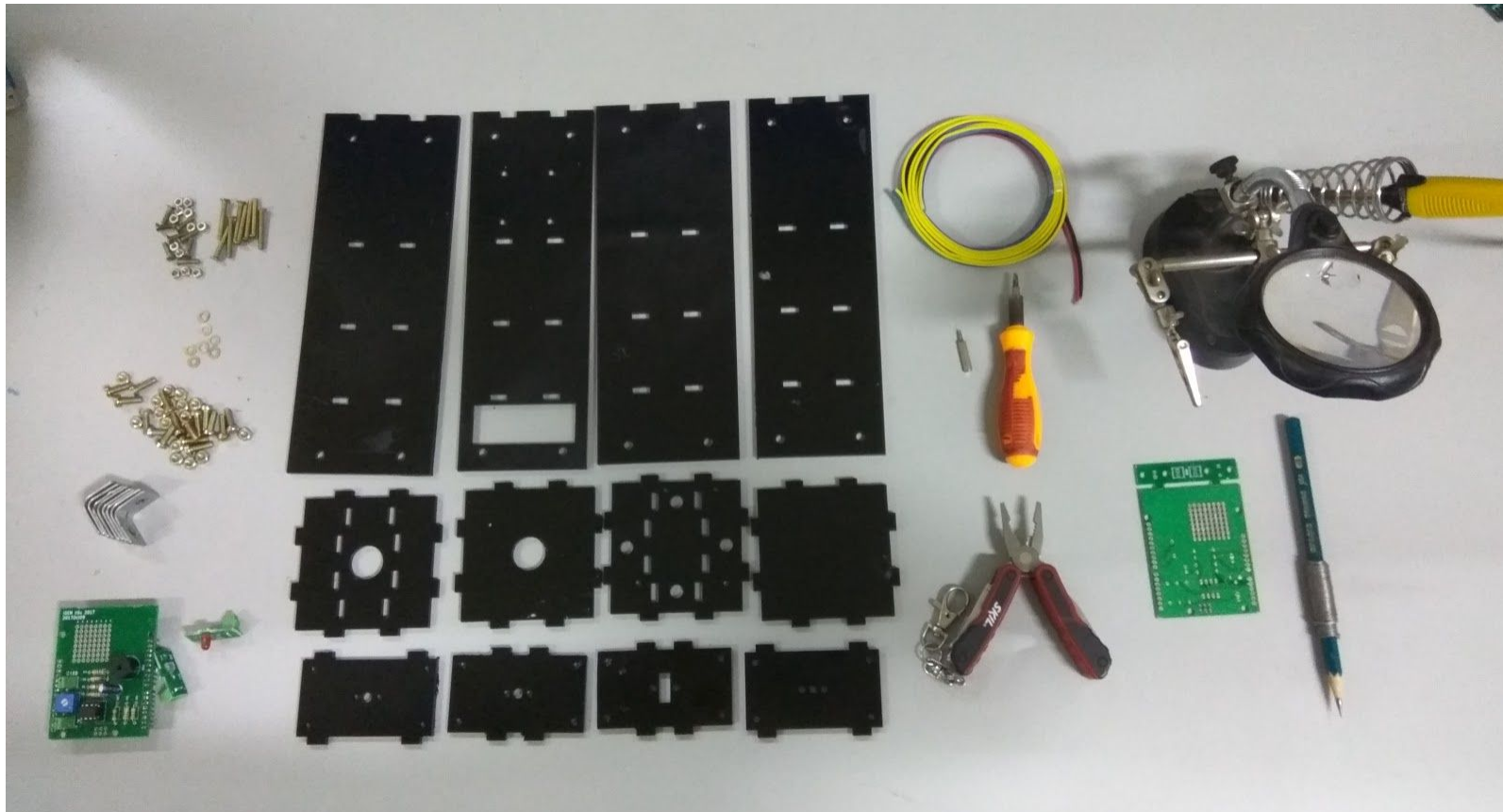
2. Install it. (Mostly involves hitting next)
3. In your Code folder, open the GCODE_Mini_Setup.ino file in the folder of the same name.
4. Double-click it to open it in the Arduino IDE.
5. Connect the Arduino Uno to your PC using the USB cable.
6. Ensure that the device is selected under Tools->Port. If it hasn't, just click on it to select it.



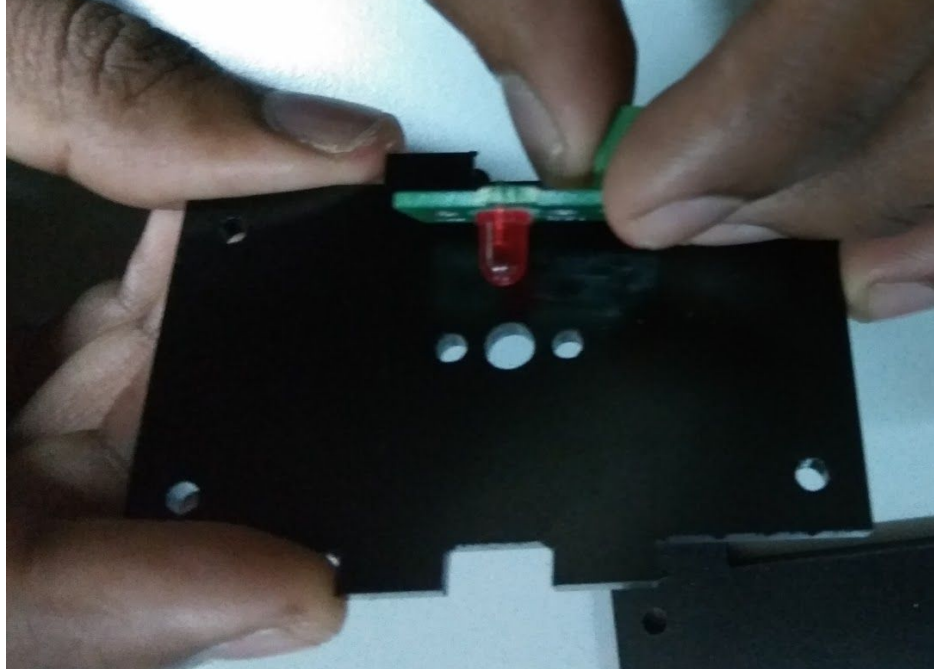
7. Hit the upload button  .
8. When it's done uploading, you can disconnect your Arduino.

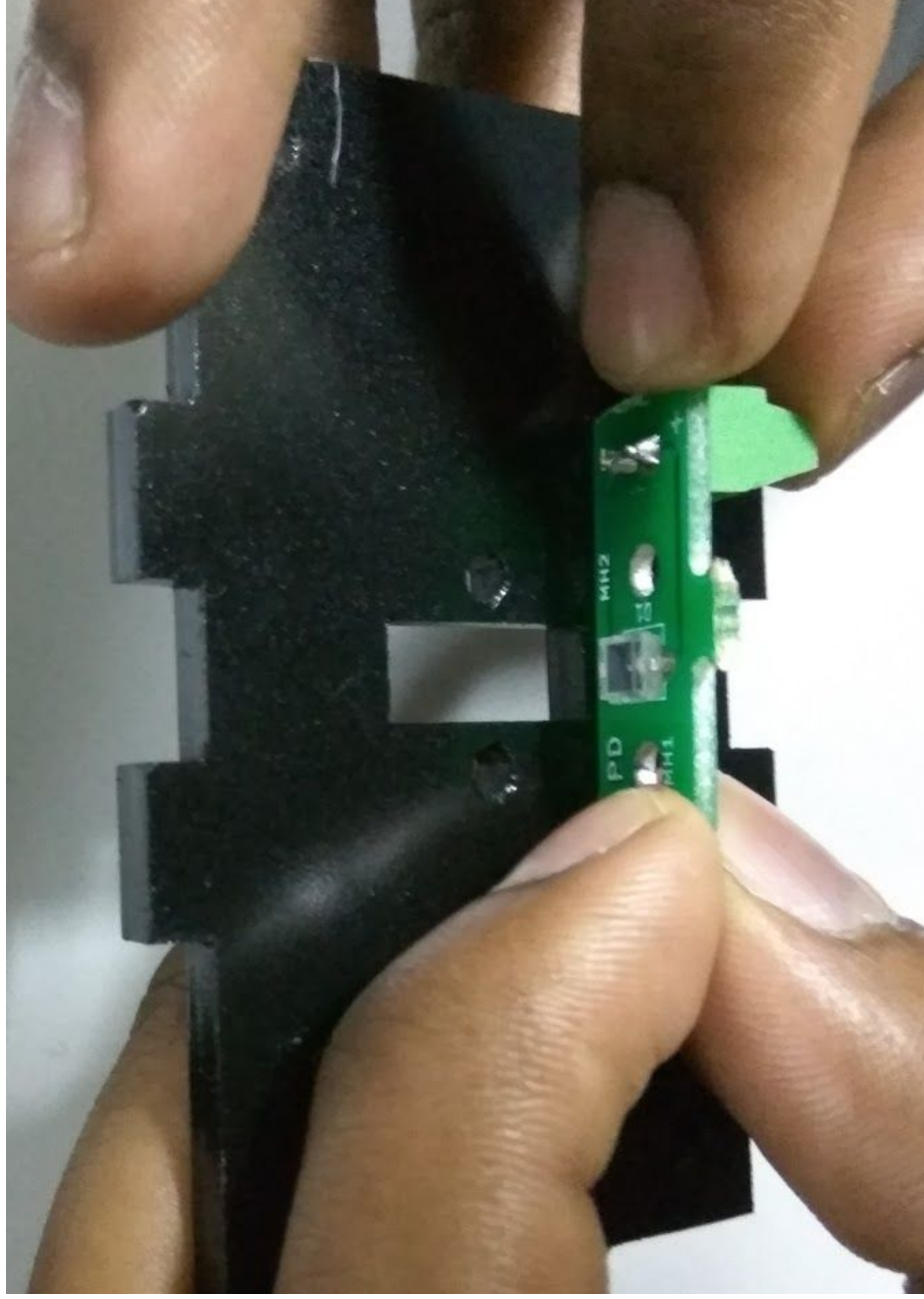
Assembly

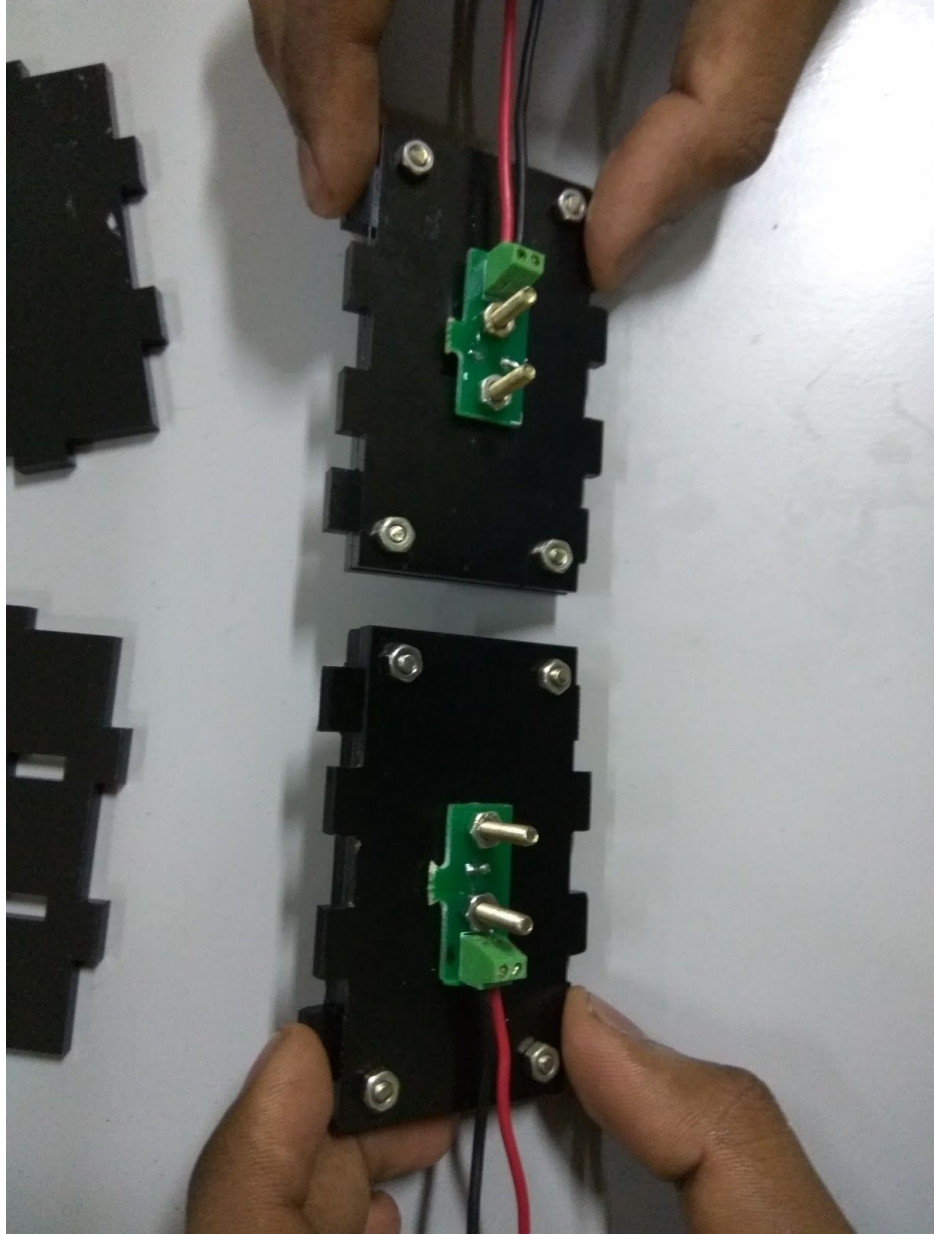
1. Break the PCB and solder the parts. Pay attention to the polarity of the LED and photodiode. It's been marked out in the silk screen (white) and in copper (raised and kinda silvery)
2. Lay your parts out



3. Attach the LED and photodiode PCBs here

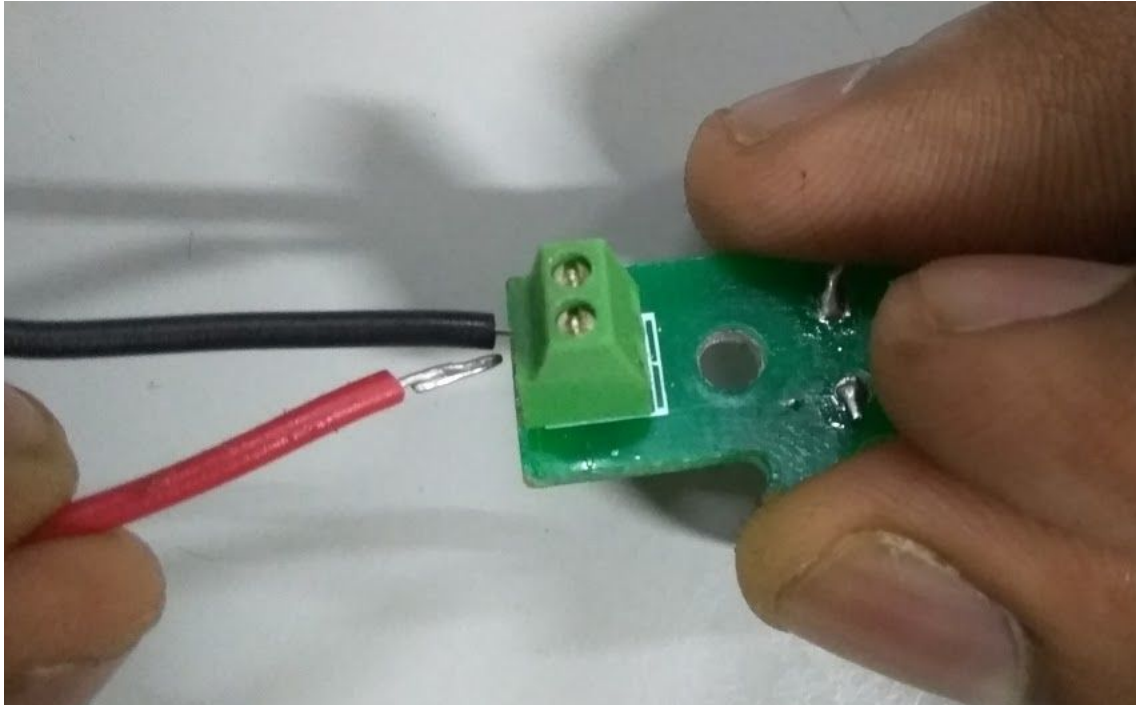




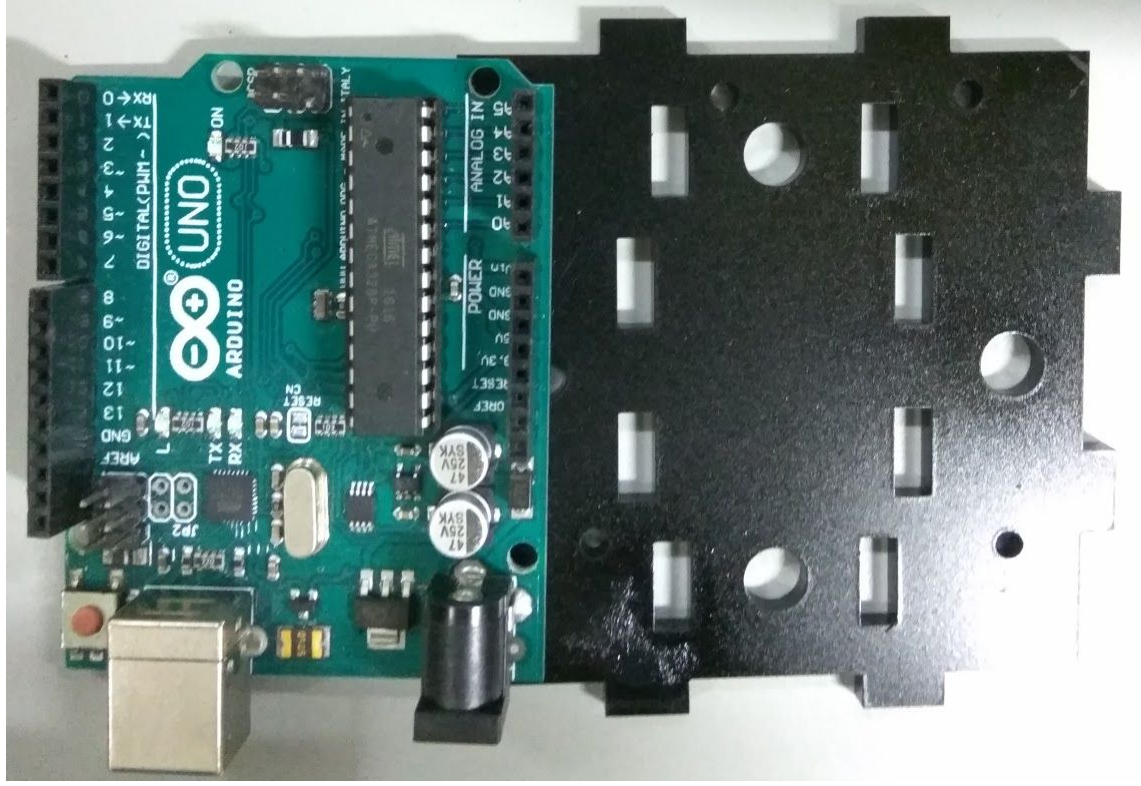


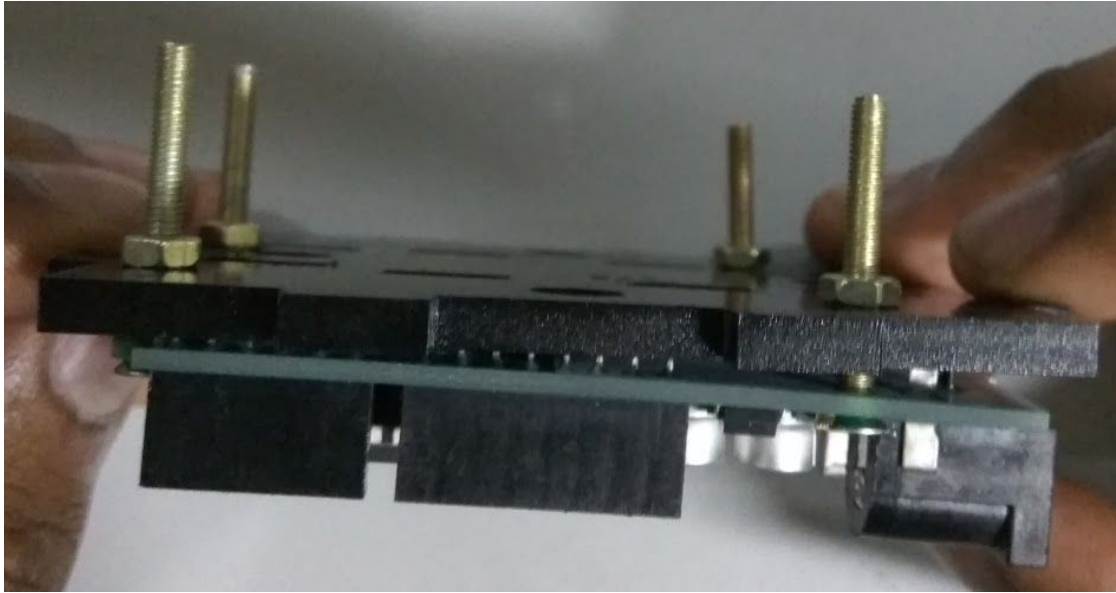
(don't attach the wires yet - we will attach them at a later step)

4. Attach 10 cm of wire to each terminal: red to + and black to -

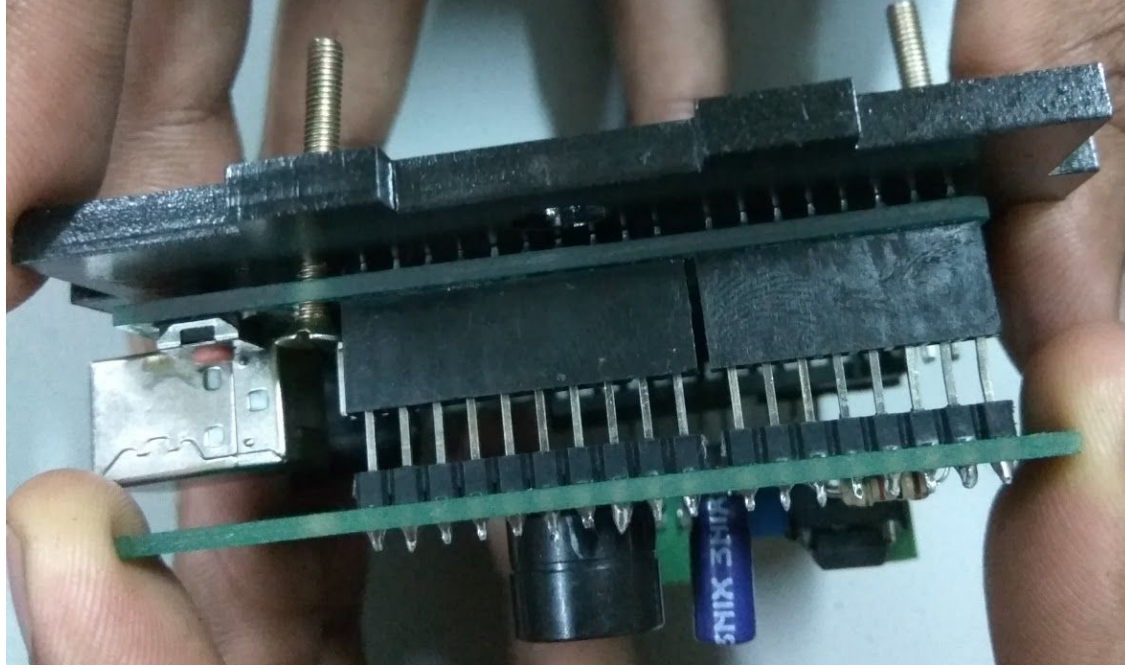


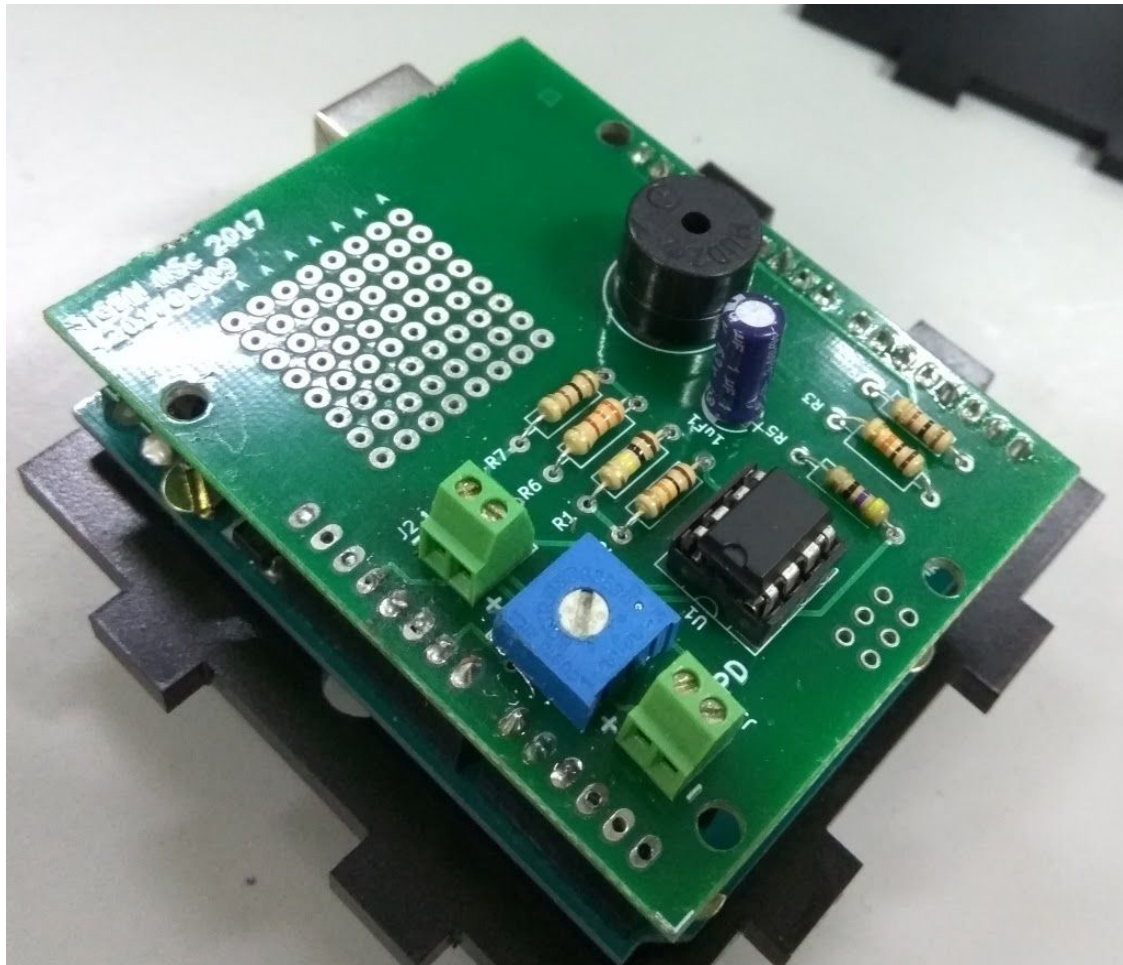
5. Attach the Arduino Uno to this piece using the 3mm nuts and bolts. If it doesn't seem to align, try flipping the black plate around.



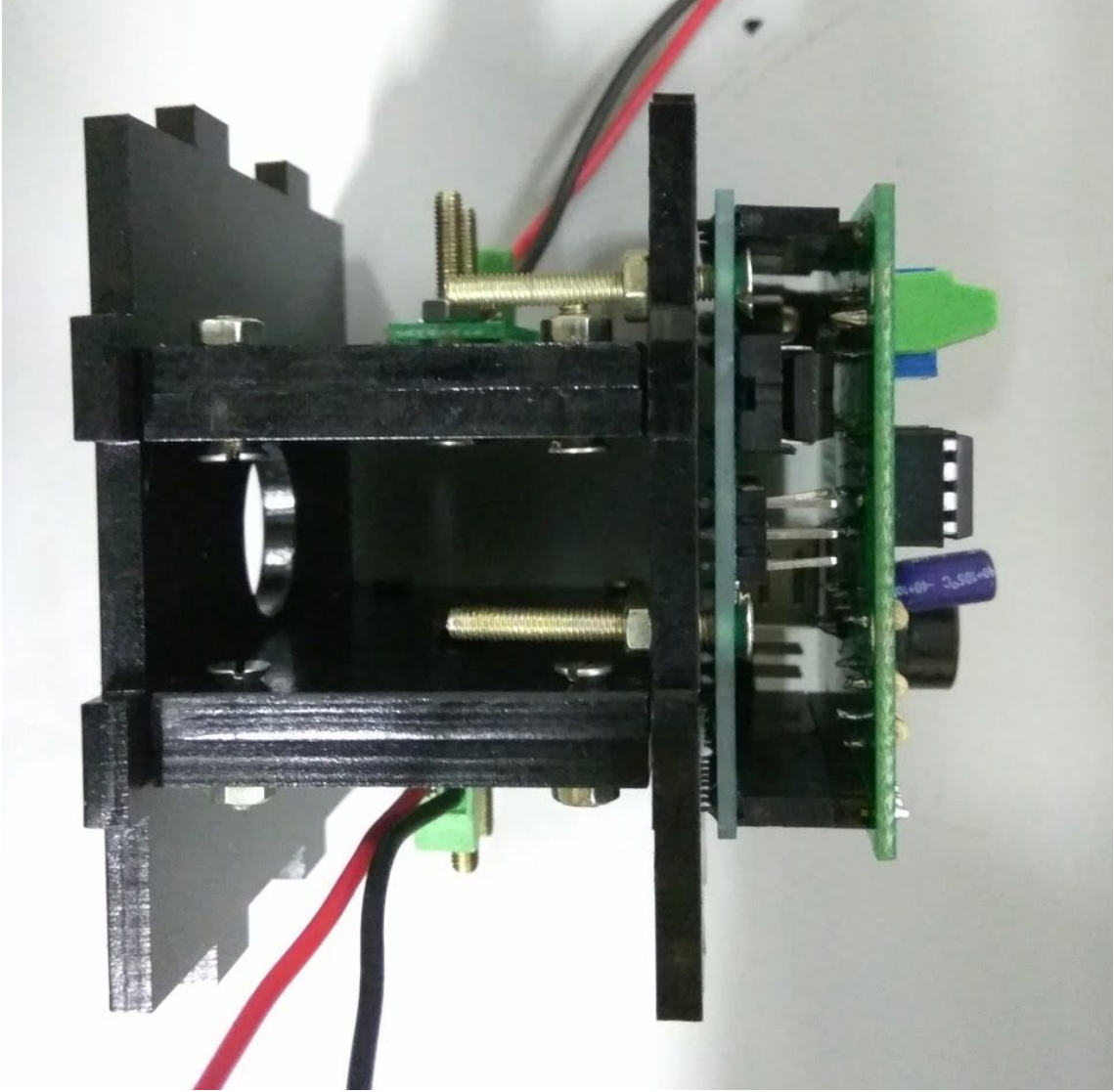


6. Attach the PCB to the Uno

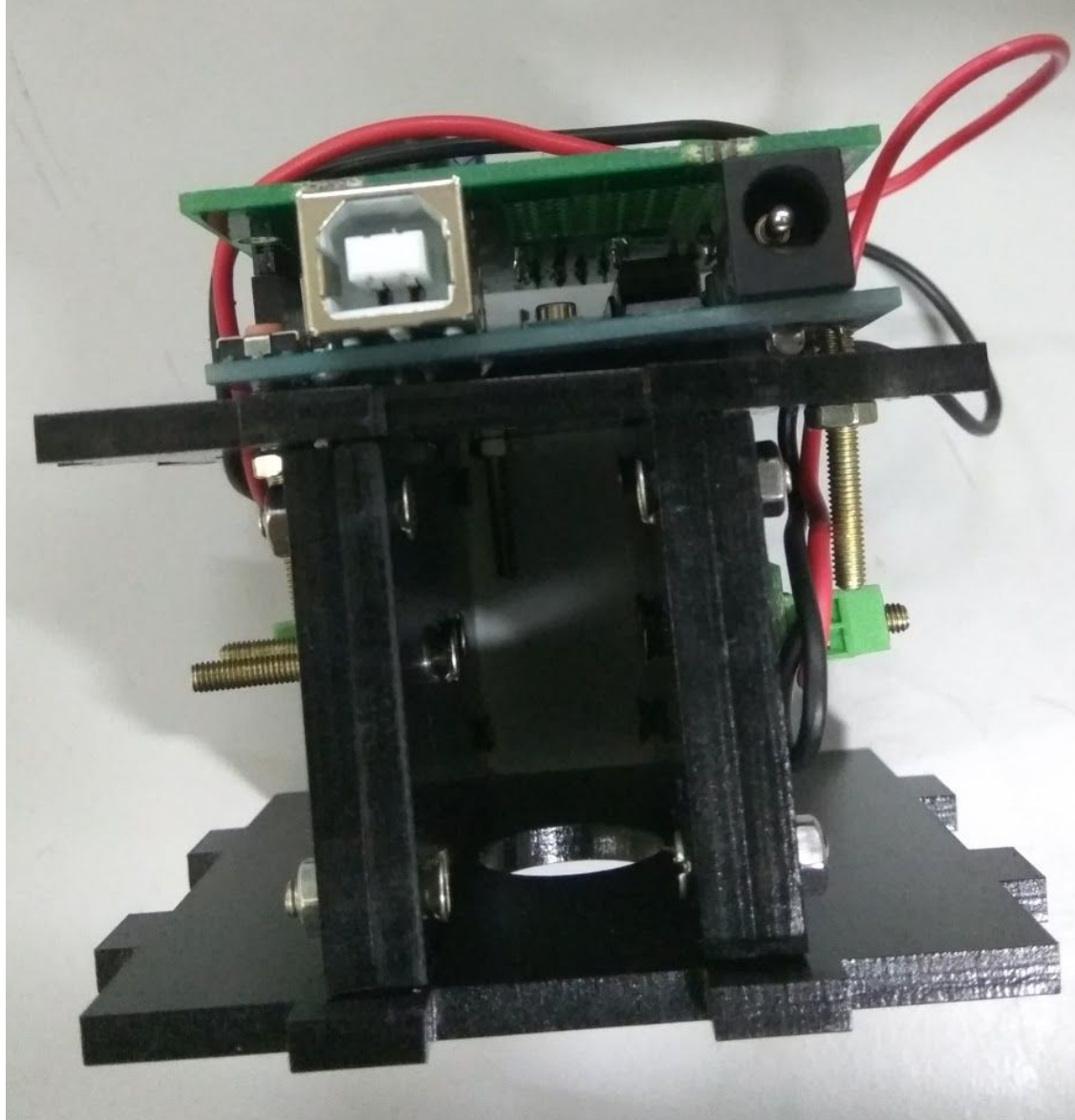




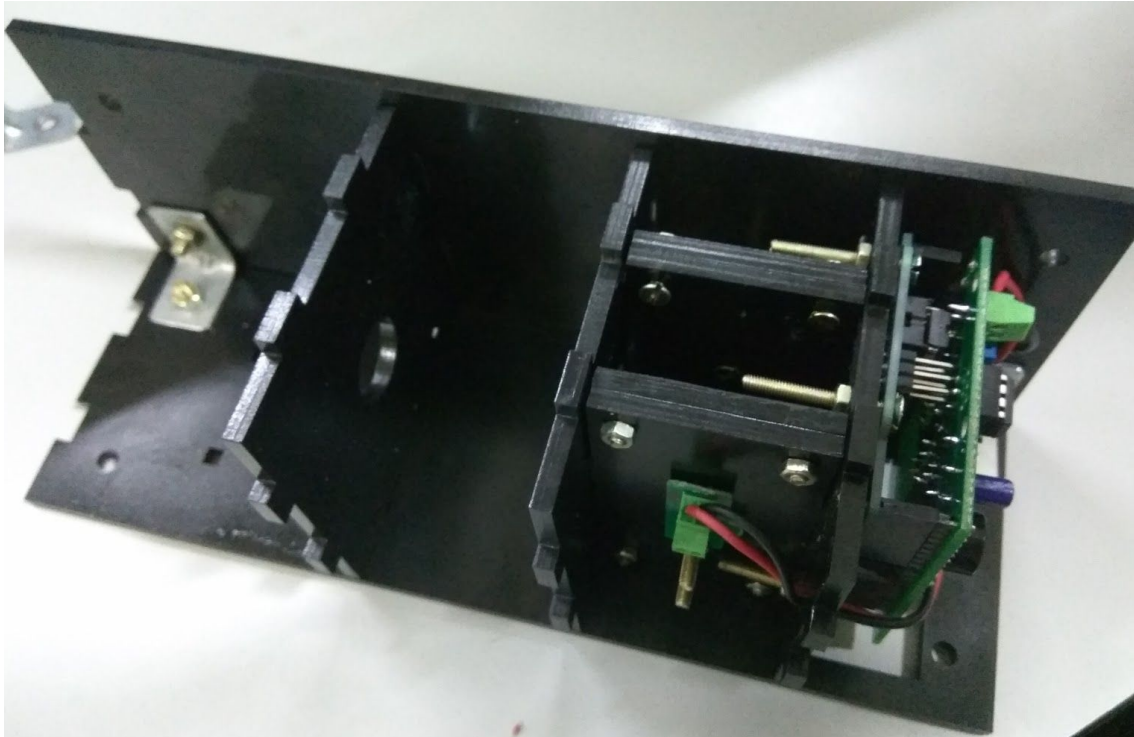
7. Assemble the optical chamber



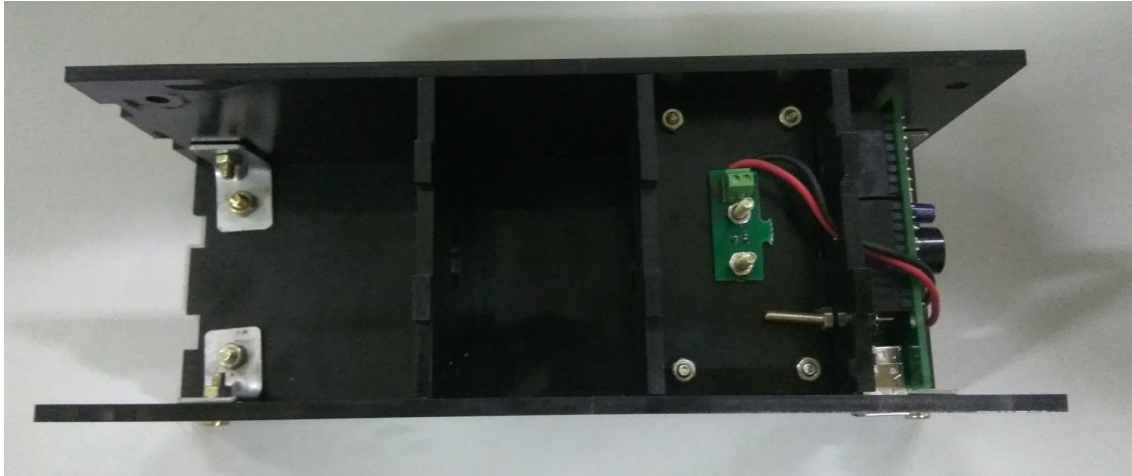
8. Connect the wires from the LED and photodiode to the screw terminals on the PCB marked “LED” and “PD” respectively, and make sure to get the polarity right here too. Red is + and black is -



9. Attach the optical chamber to the faceplate with a hole (in this orientation). Then attach the test-tube holder plate to it. Finally attach one of the wider plates to the right, with L-bends.




10. Attach the remaining faceplates

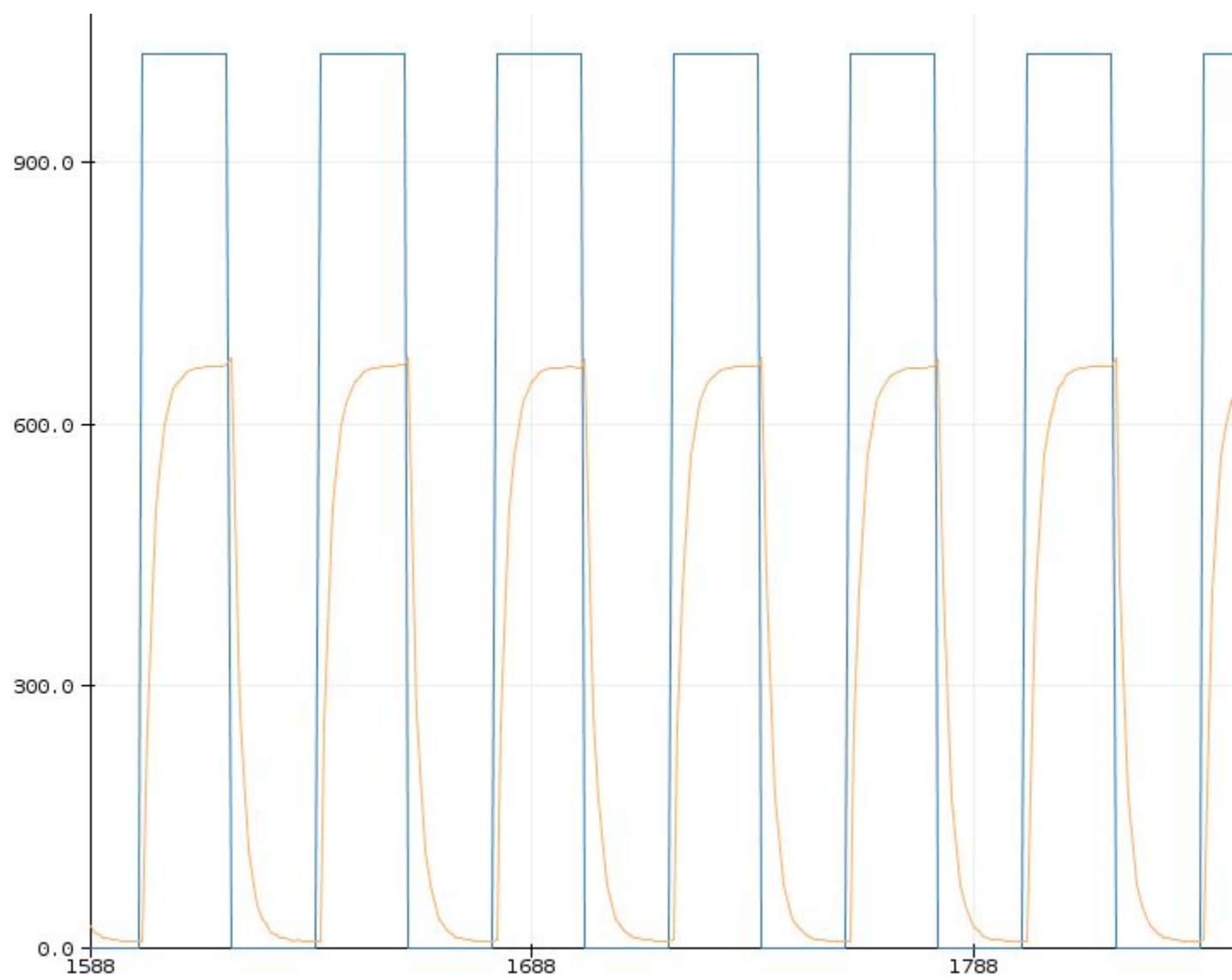


11. Put the lid on, and your GCode Mini is complete!



Gain Adjustment

1. Make sure your GCODe is connected to your PC.
2. Open up the Arduino IDE
3. Open the serial plotter (Ctrl+Shift+L). You may have to set the port again (under Tools->Port)
4. Rotate the variable resistor (the blue cuboid on the PCB) using a screwdriver until your graph looks like this (yellow line should saturate at about 700).
 - If the wiggly line only stays above 900, you probably flipped the polarity on the photodiode.
 - If the wiggly line basically stays at 0, you have probably misconnected the photodiode wires.
 - If it doesn't oscillate with the straight line, you have probably either flipped the polarity on the LED, or haven't connected the wires properly.
4. When you're done with the calibration above, open the GCODe_Mini_User_0.3.ino file in the folder of the same name
5. Hit the upload button 



Software

Get your Pushbullet access token

We use Pushbullet to send notifications to your device.

1. Sign up for an account at <https://www.pushbullet.com/>.
2. We need a Pushbullet access token; so log in and go to <https://www.pushbullet.com/#settings/account>, and scroll down until you see:

Access Tokens

Using an access token grants full access to your account. Don't share this lightly. You need the access token in order to use the API.

Create Access Token

3. Hit "Create Access Token" and then copy the text in the black box somewhere.

Installation on Ubuntu

1. Open the Code folder, right click in a blank corner, and hit "Open in terminal" (Or you could navigate to that location using the terminal)
2. Run:

```
chmod u+x ./install_gcode_mini_ubuntu.sh
sudo -H ./install_gcode_mini_ubuntu.sh
```
3. Connect the mini to the PC
4. Double-click the "GCODe Mini GUI Ubuntu" file
5. Paste the access token you obtained in the previous section and hit next:

Enter your Pushbullet key as described in the documentation.

6. Click on the device name in the device selection screen and hit next. It might take a while in the first run.

Footnotes

1. The internal 3.3v is connected directly to the AREF pin on the PCB. This is to minimise the influence of voltage fluctuations on the USB port - USB isn't supposed to give clean 5V to anyone. We elected not to connect it in a "safe" manner because that could cause variations on the AREF, which will make the readings bad.
2. We haven't tested our software on Mac yet. No reason why the hardware shouldn't work though.