

Raspberry Pi Temperature Logger - Google Interface

Description: I used a temperature logger that interfaced with Xively. While this was a good tool, I was looking for something that would be a little easier to track and give me the ability to download the data to my work computer. This uses a gspread to log the information directly into my google drive account.

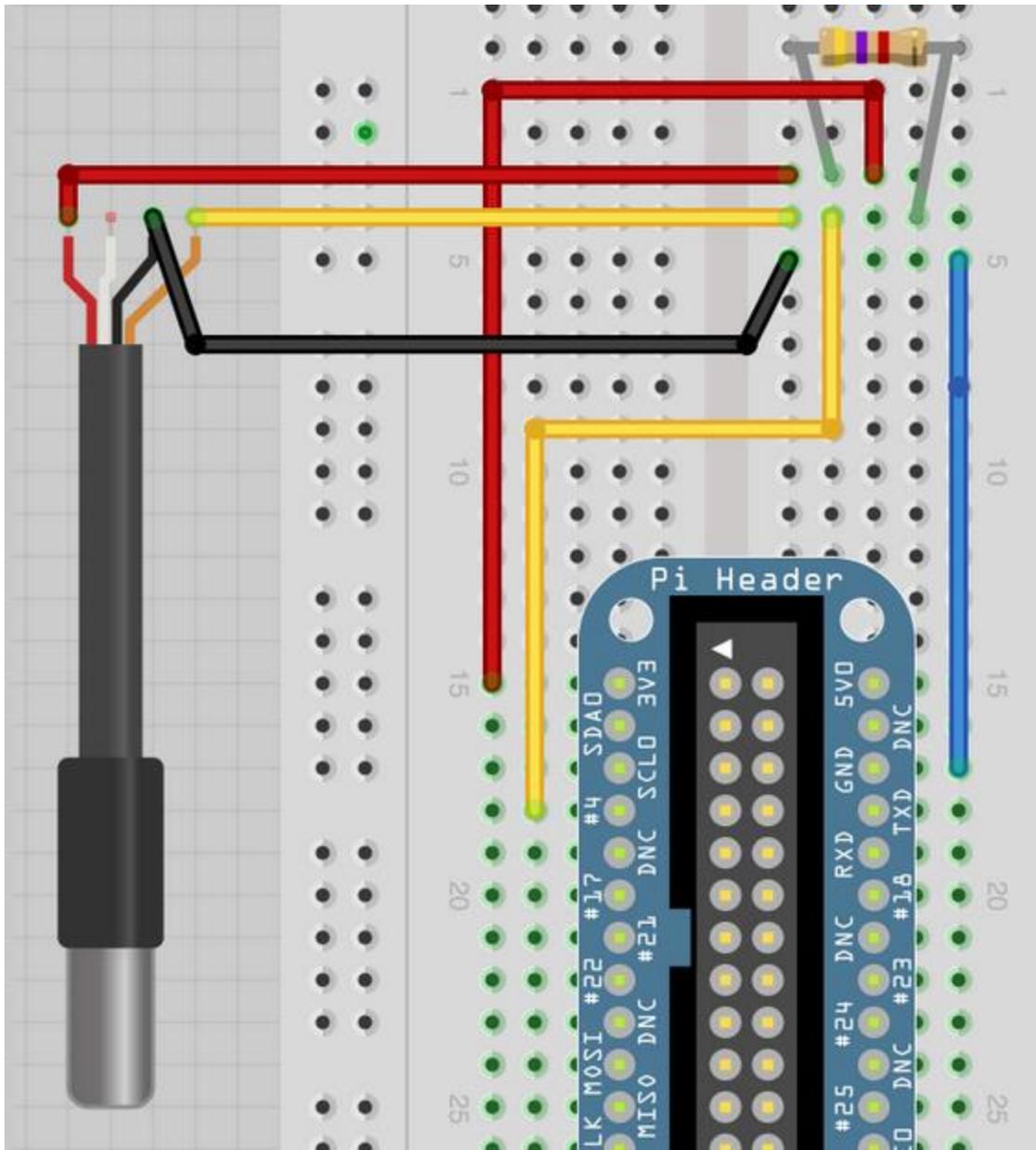
Hardware you will need:

- Raspberry Pi (Model B is probably the most appropriate)
- Wi Pi USB Wireless Device (optional)
- Micro USB Power Supply
- [Waterproof Temperature Probe](#) (DS18B20 - \$9.95 from Adafruit)
- Breadboard
- Raspberry Pi Cobbler
- Ribbon wire
- Resistor
- Jumper Cables
- Radio Shack Thingy - Wire connectors (the DS18B20 ends in wire thread, and you really need pins)
- Multimeter

Connecting the temperature probe to the RPi

What you are doing and why: Your RPi will need to interact with the probe. It will need to be plugged in to some power, receive a signal, and report it back to the computer. That takes three wires from the probe, and a breadboard. [The Adafruit tutorial](#) for putting this together is really the best, and I recommend following it. The next couple of steps summarize it.

1. Make sure your RPi is unplugged from power. You don't want to do this while you have power going through the circuit.
2. Connect the ribbon wire to the RPi pins. It should be very obvious where it goes. The part of the ribbon wire that is closest to the SD card is where the #1 pin resides.
3. Plug the Pi Cobbler into the breadboard, and build the circuit as you see it below.



There are a couple important notes:

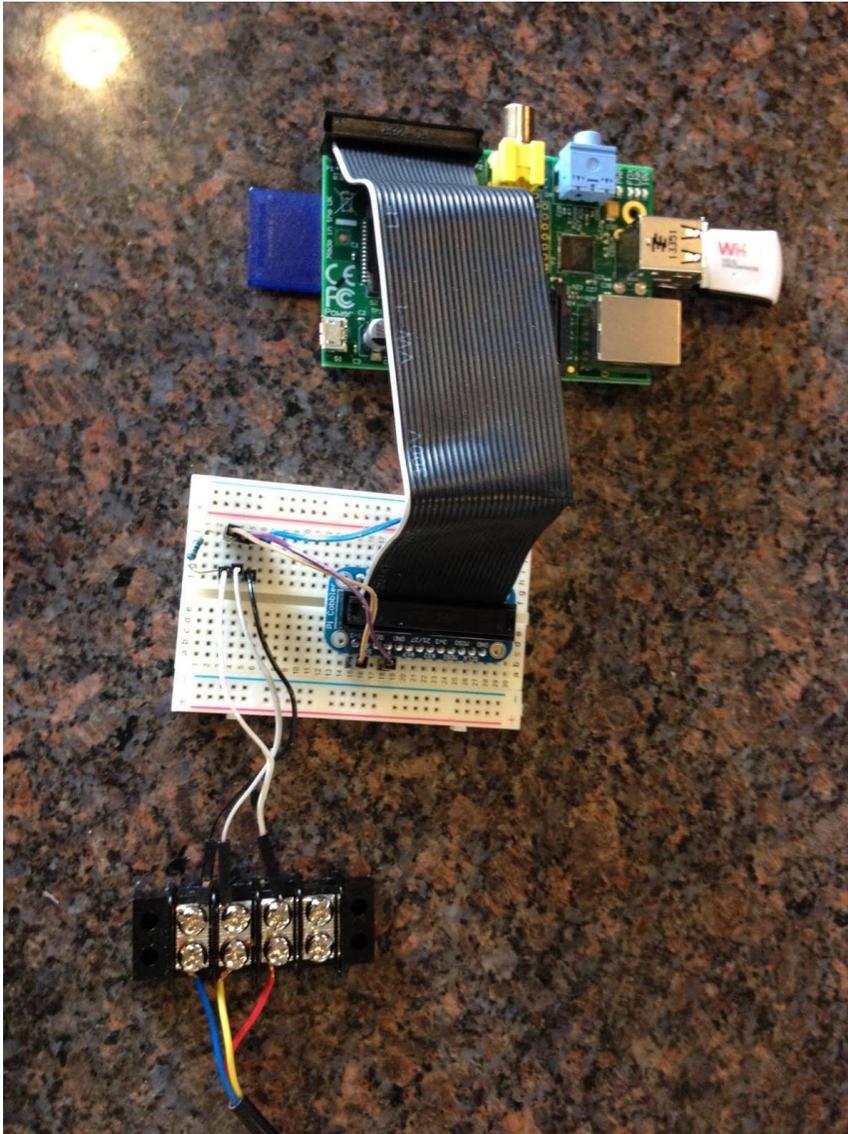
- A. The holes that are in horizontal rows (like 15 or 17 or any other going left to right) are in electrical contact.
- B. The holes that are vertical on the side (where you see the plus and minus strips) are in electrical contact.
- C. The DS18B20 probe has red, yellow, and blue wires, whereas the picture above shows different colors. I have figured out where these should go in the circuit.

The **red probe wire** goes to the **resistor** (red jumper wire).

The **yellow wire** goes **below the resistor** (yellow jumper wire).

The **blue wire** goes below the **resistor to ground** (black jumper wire).

This is what it looks like in real life:



Update the Computer

What you are doing and why: In this step you will look for and apply any updates for your linux operating system. Windows and IOS do this periodically, especially when you are in a hurry, or when your professor isn't paying attention to the classroom computer. The linux system you are will get updated when you ask it to.

1. From the terminal type

```
sudo apt-get update
```

This will produce a lot of activity on your RPi. It will look very impressive.

2. From the terminal type

```
sudo apt-get upgrade
```

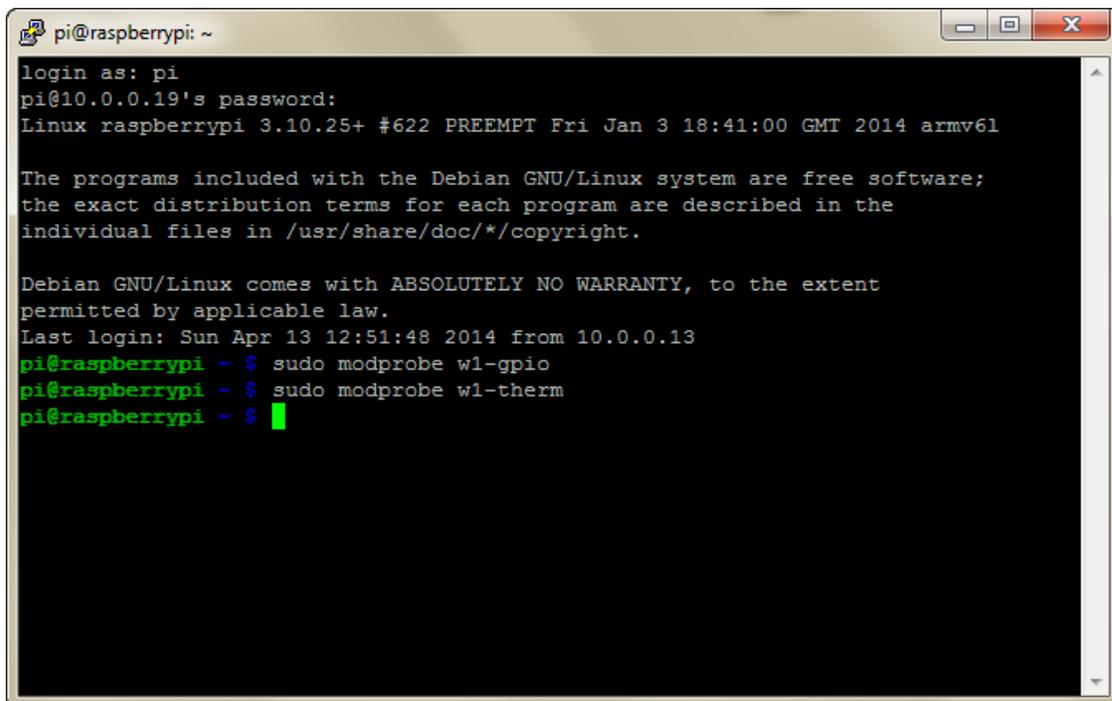
Again, many more lines of activity. It makes you look cool.

Test your Temperature Probe

What you are doing and why: Earlier you spent a lot of time setting up your circuit that will run your temperature probe. You will need to make sure that it is up and running.

1. You need to tell the RPi that it will need to use the probe. That takes two instructions. In the terminal, type:

```
sudo modprobe w1-gpio  
sudo modprobe w1-therm
```



```
pi@raspberrypi: ~  
login as: pi  
pi@10.0.0.19's password:  
Linux raspberrypi 3.10.25+ #622 PREEMPT Fri Jan 3 18:41:00 GMT 2014 armv6l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sun Apr 13 12:51:48 2014 from 10.0.0.13  
pi@raspberrypi ~$ sudo modprobe w1-gpio  
pi@raspberrypi ~$ sudo modprobe w1-therm  
pi@raspberrypi ~$ █
```

Your RPi is now ready to use the probe.

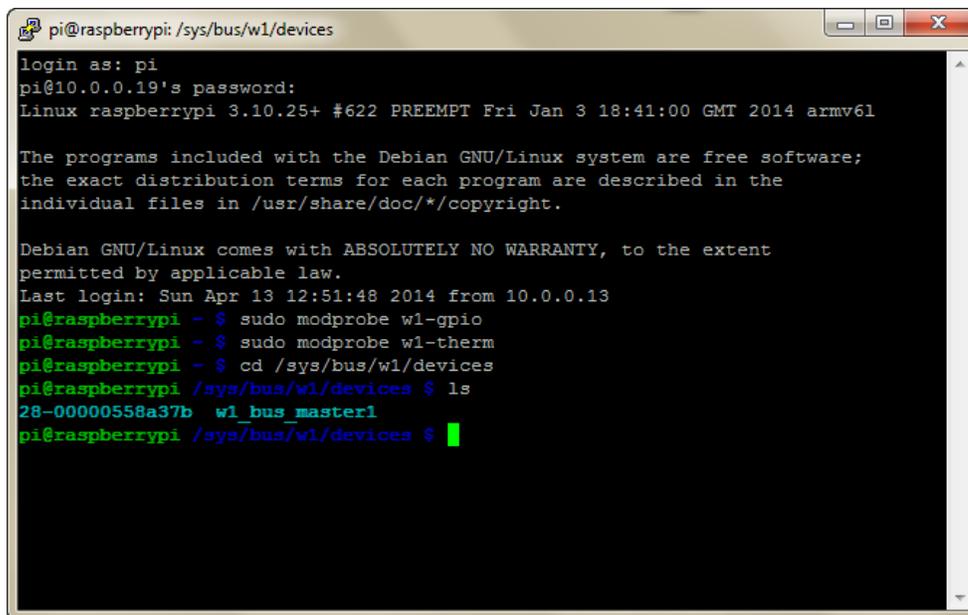
2. Now you will need to check to make sure that it shows up in your devices. If you think of the temperature probe as a USB device on your regular computer, it shows up somewhere. There is no icon in the terminal, and it doesn't happen automatically. This is how you check. In your terminal, type:

```
cd /sys/bus/w1/devices
```

This changes your directory (**cd**) from wherever it is to the **devices** folder that is in the **w1** folder which is in the **bus** folder which is in the **sys** folder.

3. Next, you will want to list (**ls**) the contents of that folder. Type

```
ls
```



```
pi@raspberrypi: /sys/bus/w1/devices
login as: pi
pi@10.0.0.19's password:
Linux raspberrypi 3.10.25+ #622 PREEMPT Fri Jan 3 18:41:00 GMT 2014 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Apr 13 12:51:48 2014 from 10.0.0.13
pi@raspberrypi ~ $ sudo modprobe w1-gpio
pi@raspberrypi ~ $ sudo modprobe w1-therm
pi@raspberrypi ~ $ cd /sys/bus/w1/devices
pi@raspberrypi /sys/bus/w1/devices $ ls
28-00000558a37b  w1_bus_master1
pi@raspberrypi /sys/bus/w1/devices $
```

4. Your new temperature probe will show up as folder. It looks like a long serial number, and it starts with 28-xxxxx. You will want to cd into it. You type the whole serial number there, or you can type:

```
cd 28*
```

5. Now you can give the probe an order. You can ask it to take a temperature. In the terminal type

```
cat w1_slave
```

```
pi@raspberrypi: /sys/bus/w1/devices/28-00000558a37b
login as: pi
pi@10.0.0.19's password:
Linux raspberrypi 3.10.25+ #622 PREEMPT Fri Jan 3 18:41:00 GMT 2014 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Apr 13 12:51:48 2014 from 10.0.0.13
pi@raspberrypi ~$ sudo modprobe w1-gpio
pi@raspberrypi ~$ sudo modprobe w1-therm
pi@raspberrypi ~$ cd /sys/bus/w1/devices
pi@raspberrypi /sys/bus/w1/devices $ ls
28-00000558a37b  w1_bus_master1
pi@raspberrypi /sys/bus/w1/devices $ cd 28*
pi@raspberrypi /sys/bus/w1/devices/28-00000558a37b $ cat w1_slave
5f 01 4b 46 7f ff 01 10 9b : crc=9b YES
5f 01 4b 46 7f ff 01 10 9b t=21937
pi@raspberrypi /sys/bus/w1/devices/28-00000558a37b $ █
```

You should have an output that ends with a number that looks like a temperature. It is the temperature from the probe, in Celsius without the decimal point. If it were the temperature, you'd think you were on the surface of the sun! Since that is probably not the case, you can now rest assured that you are capable of collecting temperatures.

Download and install gspread

What you are doing and why: The python library gspread allows you to output data directly into a google spreadsheet that you have set up. The only things it requires is that you give it the information necessary to access it: the name, your username, and your password. It is your program, so that should be pretty secure.

1. You can clone this directly onto the Raspberry Pi by cloning. You can start this by getting to your home directory. In the command line type:

```
cd ~
```

2. Next you will want to clone gspread from a github site. You can do this by typing the following into your command line:

```
sudo git clone https://github.com/burnash/gspread.git
```

3. Open the directory

```
cd gspread
```

4. From that folder, install the library

```
sudo python setup.py install
```

5. Now the library is available for you to use. All of these instructions are on the gspread website (<https://github.com/burnash/gspread>)

Write the Program

What you are doing and why: Now that you have a functional temperature probe, you want a program that will tell your computer to use it, and send that information to google spreadsheet.

1. While still inside the gspread folder type

```
sudo git clone https://github.com/MDSscience/temperature.git
```

2. Open the program by typing

```
sudo nano temp.py
```

3. In the program there is a line that says

```
gc = gspread.Login('account@account.com', 'password')
```

Change *'account@account.com'* and *'password'* to your login information for Google Drive. Keep the quote marks. **NOTE: If you share your program with anyone, make sure you remove your login information before you do, otherwise they'll have it.**

4. In the program there is a line that says

```
worksheet = gc.open('spreadsheet_name').sheet1
```

change *'spreadsheet_name'* to whatever you will call the spreadsheet.

5. Close out by pressing Control O and the Control X after it verifies the name.

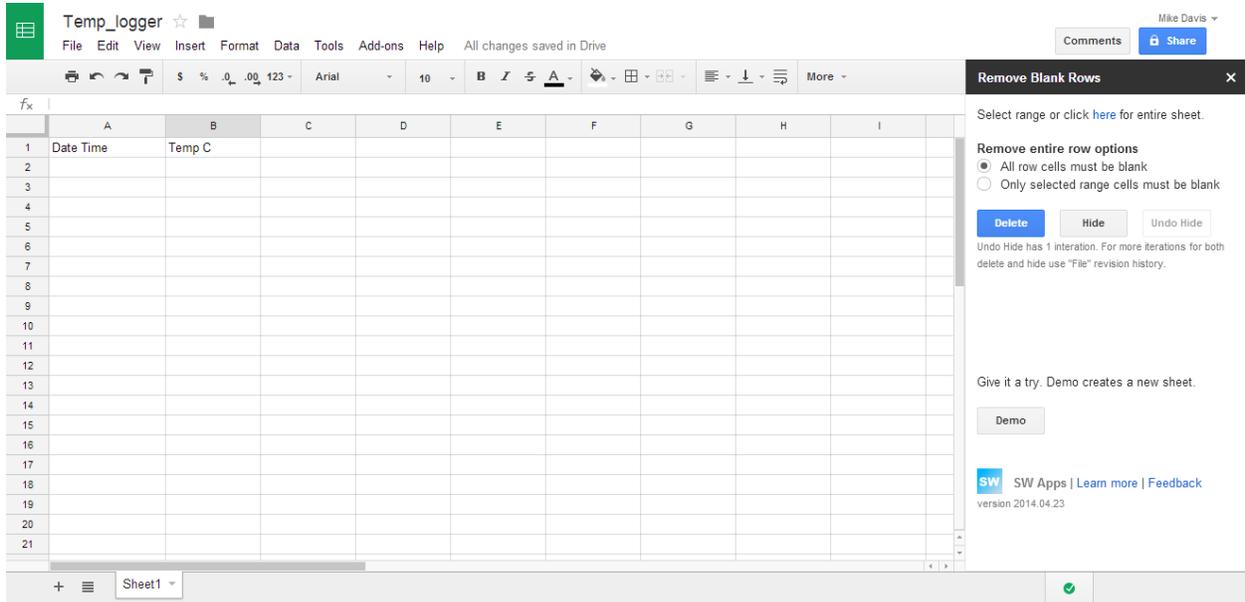
Set Up a Google Spreadsheet

What you are doing and why: The way our program is written, you will need to have an existing spreadsheet.

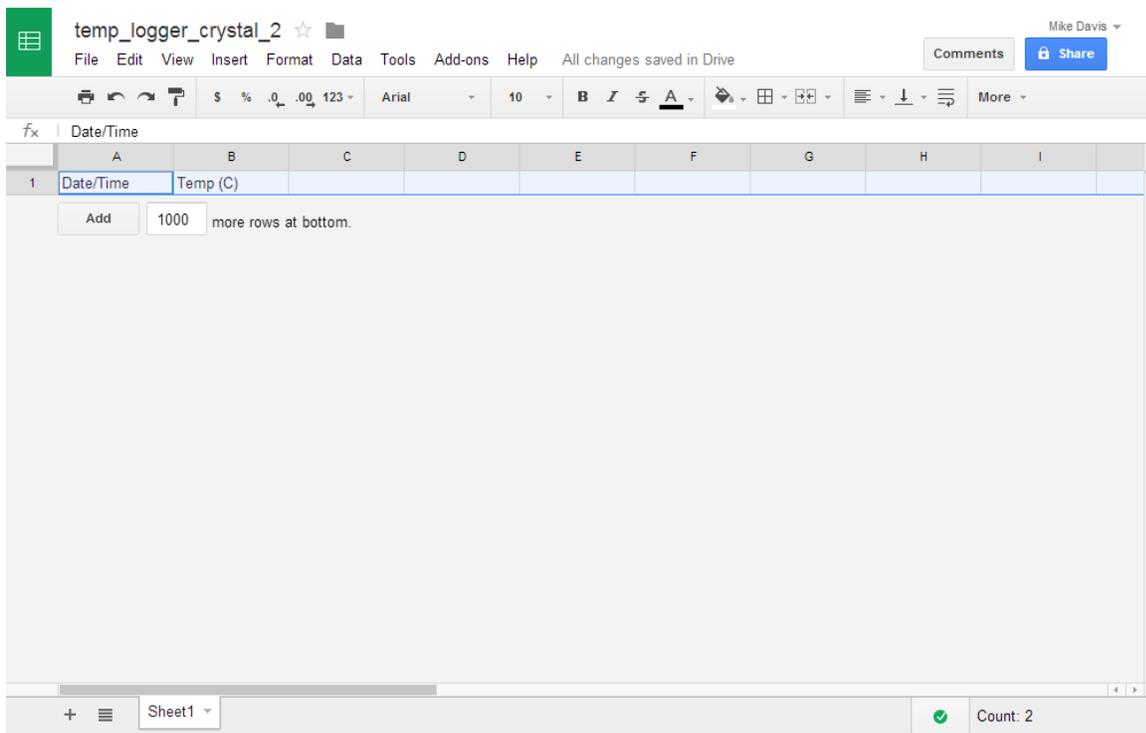
1. You just need to log in to <http://drive.google.com> and create one. Make sure you remember the name of the spread sheet, your username, and password. All of these will be necessary.
2. Label the A1 Cell as Time. Label the B1 Cell as Temperature.
3. You will need install an add-on that deletes empty rows in the spreadsheets you can do

it from [here](#).

- Once installed, click on Add-Ons and select Delete Empty Rows. You can delete the empty rows. The program looks for the first filled row, and puts data in after it by creating a row.



Before you begin, it should like this:



There are no blank rows in it.

Run the Program

What you are doing and why: Tell the computer to run the program. Its what you've been waiting for.

1. In the command line, type:

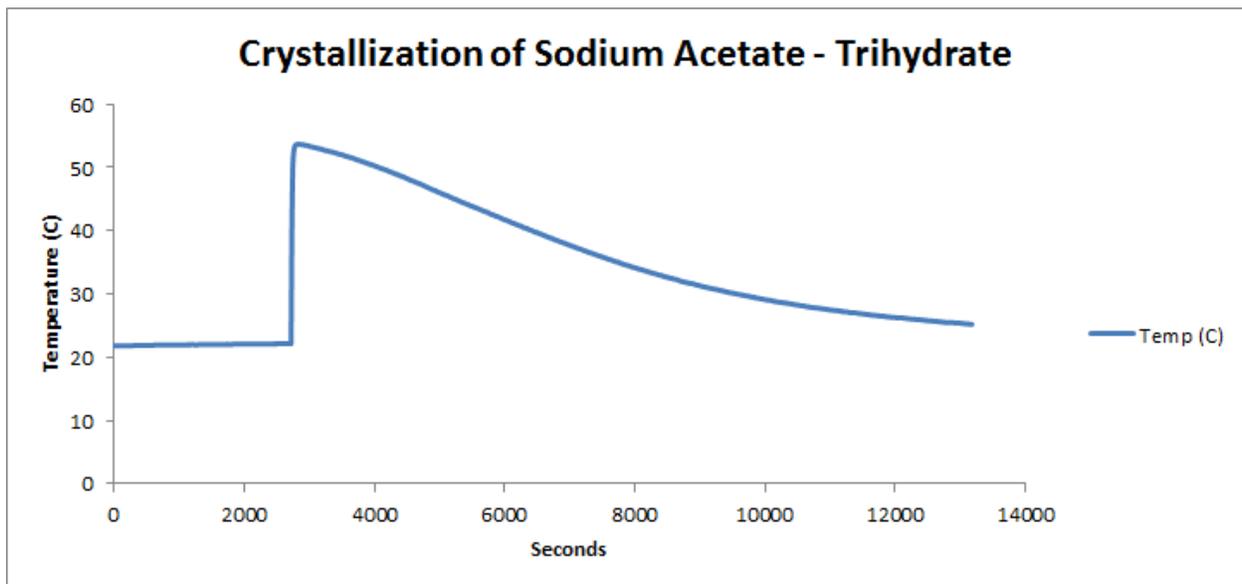
```
sudo python temp.py &
```

You are now done. The '&' tells the computer to keep running this program in the background. That way you can keep the program going and do other things.

NOTE: All of this was done in the gspread folder. When you type the command above you will need to be in the folder that contains temp.py.

Results

This is an example of what you get back (if you a graph being made at the same time).



Resources

<https://learn.adafruit.com/large-pi-based-thermometer-and-clock/software>

<http://www.danielhansen.net/2013/03/raspberry-pi-temperature-logging-using.html>

<http://julianmilligan.co.uk/?p=169>