

Once we get the I2C tools installed, we need to make sure we can detect the ADC. We need to scan the I2C bus using

`i2cdetect -r 1` in the command center, which will scan the port and tell us exactly what we have there and some things we may be able to do with it.

When we have confirmed that the Raspberry Pi has detected the ADC, we can begin with our program that will actually chart it's input. Python will be the code language used to do this. We will import a package found online that was initially used to chart humidity and temperature through the I2C port. I've taken the code from this project and made proper changes to change the detection to incoming readings from our ADC.

After installing the necessary files, in Python script, run

```
import Adafruit_DHT
import time
import csv
import sys
csvfile = "volt.csv"
als = True
while als:
    voltage = Adafruit_DHT.read_retry(Adafruit_DHT.DHT11, 4)
    #Change 4 to pin ADC is plugged into
    if voltage is not None:
        voltage = round(voltage, 2)
        print 'Voltage - {0:0.1f}*V.format(voltage)
    else:
        print 'No input'
    timeC = time.strftime("%I")+':' +time.strftime("%M")+':' +time.strftime("%S")
    data = [voltage, timeC]

    with open(csvfile, "a")as output:
        writer = csv.writer(output, delimiter=";", lineterminator = '\n')
        writer.writerow(data)
    time.sleep(0.05) #updates script every half second. can be adjusted as needed
```

#the above code is reading the information from the I2C port and placing it into a csv file. The code below takes the data from the csv file and shows it on a graph

```
import matplotlib.pyplot and plt
import matplotlib.dates as mdates
import matplotlib.animation as animation
from datetime import datetime
```

```

fig = plt.figure()
rect = fig.patch
rect.set_facecolor('#0079E7') #this makes the plot line a light blue and can be changed
def animate(i):
    fvolt = 'volt.csv'
    fv = open(fvolt)
    volt = list()
    timeC = list()
    for line in fv:
        pieces = line.split(',')
        degree = pieces[0]
        timeB = pieces[1]
        timeA = timeB[:8]
        time_string = datetime.strptime(timeA, '%H:%M:%S')
        try:
            volt.append(float(degree))
            timeC.append(time_string)
        except:
            print "dont know"

    ax1 = fig.add_subplot(1,1,1,axisbg='white')
    ax1.xaxis.set_major_formatter(mdates.DateFormatter('%H:%M:%S'))
    ax1.clear()
    ax1.plot(timeC,volt, 'c', linewidth = 3.0)
    plt.title('Voltage')
    plt.xlabel('Time')
ani = animation.FuncAnimation(fig, animate, interval = 50)
plt.show()

```

```

*/

```

```

void setup() {

```

```

}

```

```

void loop() {

```

```

}

```

Note that we likely would have run into an error or two with this code, since as earlier mentioned, it was initially written to detect temperature. However, I don't think we would've had such major issues that the program would be unusable. The code brings in information from the port, not from a specific detector so as long as we could have gotten the ADC hooked up and communicating correctly, this code has a good chance at working just like we want.

