

Nicholas LaJoie, ECE 331, Project 2

```
// Author: Nicholas LaJoie
// ECE 331 - Project 2
// Date: May 2, 2017

/* i2c Data Grabber - i2c.py */

#!/usr/bin/env python

# Author: Nicholas LaJoie
# ECE 331 - Project 2
# Description: Grab i2c temp data and store in SQLite 3 database
# Sources: dos.python.org/2/library/sqlite3.html
# https://randomnerdtutorials.com/sqlite-database-on-a-raspberry-pi/
# http://stackoverflow.com/questions/6304808/how-to-pass-tuple-as-argument-in-python
# http://stackoverflow.com/questions/209513/convert-hex-string-to-int-in-python
# https://linux.die.net/man/8/i2cget
# http://sharats.me/the-ever-useful-and-neat-subprocess-module.html#running-via-the-shell
# http://www.raspberry-projects.com/pi/programming-in-python/i2c-programming-in-python/using-the-i2c-interface-2
# http://effbot.org/librarybook/datetime.htm

import time                # For date and time calculations
import sqlite3 as sq       # Interact with SQLite3 Database
import subprocess          # Run i2cget

# Initializations
conn = sq.connect('/home/pi/templogger/temp.db') # Connect to the temp database
c = conn.cursor() # Cursor can execute sqlite commands

# Get i2c data
temp = subprocess.check_output(['/usr/sbin/i2cget', '-y', '1', '0x48', '0', 'b'])
cel = int(temp,16) # Convert to decimal, celsius
faren = (cel*(9.0/5.0)) + 32

# Get date and time
curday = time.time()
curday = int(curday / 60)

# Insert data into preexisting table using a tuple
data = ((faren, curday))

c.execute('INSERT INTO templog(temperature, datetime) values(?, ?);', data)

# Remove values more than 24 hours old
yesterday = time.time() - (24*60*60)
yesterday = int(yesterday / 60)

old = (yesterday,)
c.execute('DELETE FROM templog WHERE datetime <= ?;', old)

print "{} {} {}".format(faren, curday, yesterday) # DEBUGGING

# Clean up
conn.commit() # Commit changes to database
conn.close() # Close the connection

/* Data plotter - graphy.py */

#!/usr/bin/python3

# Author: Nicholas LaJoie
# Source: https://matplotlib.org/users/pyplot\_tutorial.html
```

Nicholas LaJoie, ECE 331, Project 2

```
# Numerous Stack Overflow pages
# matplotlib.org

import sqlite3 as sq
import time
import datetime

import numpy as np
import matplotlib.pyplot as plt
import matplotlib.dates as dt

# Plot the SQLite3 data using matplotlib
dpath = '/home/pi/templogger/temp.db'

# Connect to database
conn = sq.connect(dpath) # Connect to the TEST database
c = conn.cursor()       # Cursor can execute sqlite commands

# SQL Commands
query = "SELECT * FROM templog ORDER BY datetime DESC LIMIT 1440"

# Arrays to hold temp and dates
temp = []
dates = []

# Iterate through data in sqlite database
for row in c.execute(query):
    line = str(row).replace(' ','').replace('(','').replace(' ','')
    splitInfo = line.split(',')
    t = int(splitInfo[1]) * 60
    d = datetime.datetime.fromtimestamp(float(t)).strftime('%c') # Converts epoch time to date
    string
    temp.append(splitInfo[0])
    dates.append(d)

# Reverse data order to correct it
temp.reverse()
dates.reverse()

# Set up for plotting
axes = plt.gca()
axes.set_xlim([0, 1439])
axes.set_ylim([20, 90])
datelabels = (dates[0], dates[205], dates[410], dates[615], dates[820], dates[1025], dates[1230], dates[1439])
axes.set_xticklabels(datelabels, rotation = 45, ha = 'right')

# Plot and save figure
plt.figure(1)
plt.tight_layout()
plt.grid()
plt.title('ECE 331 - Project 2 - Temperature Logger')
plt.xlabel('Date/Time')
plt.ylabel('Temperature, Degrees Farenheit')
plt.plot(temp)
fig = plt.gcf()

# Save image so it can be sent to the webpage
fig.savefig('/var/www/html/temp.png', frameon = 'True')

# Clean up
conn.close()
```

Nicholas LaJoie, ECE 331, Project 2

```
/* PHP index page - index.php */

<HTML>
<TITLE>Nicholas LaJoie's Temperature Logger</TITLE>
<BODY>

<h1 style = "text-align:center;">Nicholas LaJoie's TempLogger</h1>
<p style = "text-align:center;">There's room for improvement.</p>

<style type="text/css">
img.center {
    display: block;
    margin-left: auto;
    margin-right: auto;
}
</style>

<?php
echo '<img src = "temp.png" width = "812" height = "612" alt = "PLOT FAILED" class = "center">
';
?>

</BODY>
</HTML>

/* Python data test script - test.py */

#!/usr/bin/env python

# Author: Nicholas LaJoie
# ECE 331 - Project 2
# Description: A test script to generate random temp data for plotting
# Sources: dos.python.org/2/library/sqlite3.html
# https://randomnerdtutorials.com/sqlite-database-on-a-raspberry-pi/
# http://stackoverflow.com/questions/6304808/how-to-pass-tuple-as-argument-in-python
# http://stackoverflow.com/questions/209513/convert-hex-string-to-int-in-python
# https://linux.die.net/man/8/i2cget
# http://sharats.me/the-ever-useful-and-neat-subprocess-module.html#running-via-the-shell
# http://www.raspberry-projects.com/pi/programming-in-python/i2c-programming-in-python/using-the-i2c-interface-2
# http://effbot.org/librarybook/datetime.htm

import time                # For date and time calculations
import random
import sqlite3 as sq       # Interact with SQLite3 Database
import subprocess          # Run i2cget

# Initializations
conn = sq.connect('/home/pi/templogger/test.db') # Connect to the temp database
c = conn.cursor() # Cursor can execute sqlite commands

for i in range (24896188, 24896188 + (24*60)):
    faren = random.randrange(50, 90)

    # Insert data into preexisting table using a tuple
    data = ((faren, i))

    c.execute('INSERT INTO templog(temperature, datetime) values(?, ?);', data)

    # print "{} {} {}".format(faren, curday, yesterday) # DEBUGGING
```

Nicholas LaJoie, ECE 331, Project 2

```
# Clean up
conn.commit() # Commit changes to database
conn.close() # Close the connection
```