Kurt Trentch

Code:

```python
#!/usr/bin/python
import os
import time
import sqlite3 as mydb
import sys

""" Log Current Time, Temperature in Celsius and Fahrenheit
To an Sqlite3 database """

#reads the temperature from the device "file" and parses it
def readTemp():
    tempfile = open("/sys/bus/w1/devices/28-051691cebcff/w1_slave")
    tempfile_text = tempfile.read()
    currentTime=time.strftime('%x %X %Z')
    tempfile.close()
    tempC=float(tempfile_text.split("\n")[1].split("t=")[1])/1000
    tempF=tempC*9.0/5.0+32.0
    return [currentTime, tempC, tempF]

#get the temperature from the readTemp function and put it in the database
def logTemp():
    con = mydb.connect('/home/pi/temperature.db')
    with con:
        try:
            [t,C,F]=readTemp()
            print "Current temperature is: %s F" %F
            cur = con.cursor()
            #log the time, celsius temperature, and fahrenheit temperature into the database
            cur.execute('insert into TempData values(?,?,?)', (t,C,F))
            print "Temperature logged"
        except:
            #show the exception if something went wrong
            the_type, the_value, the_traceback = sys.exc_info()
            print "Error:"
            print the_type
            print the_value
            print the_traceback

#wait for 20 * 30 seconds = 10 minutes
for i in range(21):
    #don't wait for first reading
    if i != 0:
        time.sleep(30)
    logTemp()
```
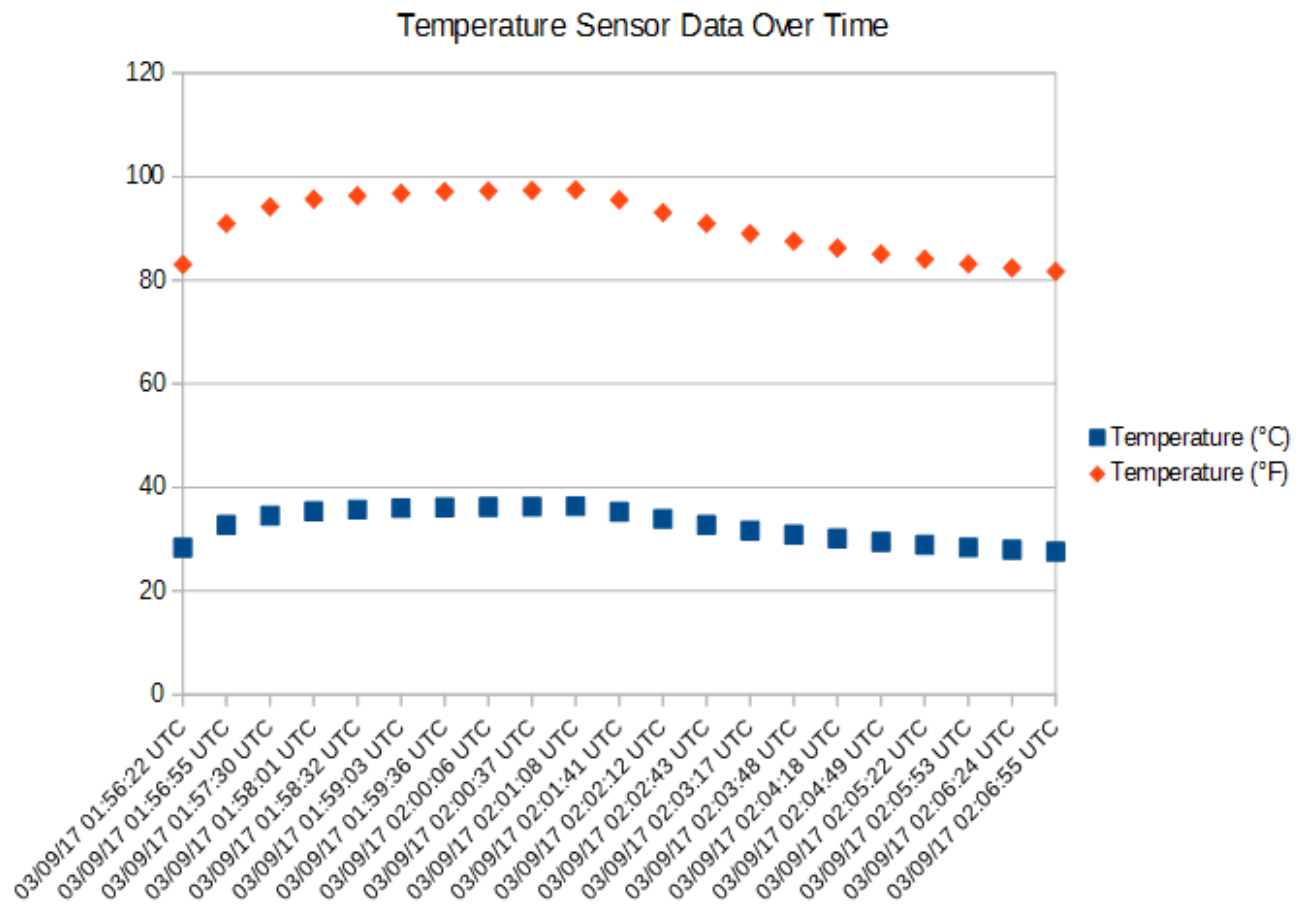
Temperature Sensor Data Over Time

The intervals are not perfectly 30 seconds because updating to the database takes more time than expected, so the extra time added about half a minute of time drift.