```
import os
     import glob
 3
     import time
 4
     import RPi.GPIO as GPIO
 5
    import sys
 6
    import shutil
 7
    GPIO.setwarnings (False)
 9
    #sets up IO and temp readings
10
     os.system('modprobe w1-gpio')
11
    os.system('modprobe w1-therm')
12
13
    base dir = '/sys/bus/w1/devices/'
14
    #assigns variable to each temperature sensor address
15
     device file1 = glob.glob(base dir + '28-0000074b7f2d')[0] + '/w1 slave'
    device file2 = glob.glob(base dir + '28-0000074b8662')[0] + '/w1 slave'
16
17
    #variable definitions
18
    oldtime = time.time() - 60
19
    oldtime2 = time.time()
20 oldday = time.localtime()
21 otime = time.time()
22 ostate = 0
23 startup = 1
24 global flow 25 flow = 47
26 energy = 0
27 totalen = 0
28 power = 0
29 dayen = 0
30
31 #flashes led on call
32 def flash(pin, state, wait):
33
        boo = not state
34
        GPIO.output (pin, state)
35
         time.sleep(wait)
36
         GPIO.output (pin, boo)
37
38
    #opens temperature sensor when passed address
39
     def read temp raw(d file):
40
         f = open(d file, 'r')
41
         lines = f.readlines()
42
         f.close()
43
         return lines
44
45 #reads and returns the temperature
46 def read temp(d file):
47
         lines = read temp raw(d file)
48
         while lines[0].strip()[-3:] != 'YES':
49
             time.sleep(0.2)
50
             lines = read temp raw1()
51
         equals pos = lines[1].find('t=')
52
         if equals pos != -1:
53
             temp string = lines[1][equals pos+2:]
54
             temp c = float(temp string) / 1000.0
55
             return temp c
56
57
   #updates the vent LED and returns the vent state
58 def IRup():
59
         if GPIO.input(24):
60
             GPIO.output (17, 0)
             state = False
61
62
         else:
63
             GPIO.output (17, 1)
64
             state = True
65
         return state
66
67
     #delays the program until specified time or until the measurement button has been
```

```
pushed or the vent opens
 68
      def wait(itime, oldtime):
 69
          while(((time.time() - oldtime) < itime) and (IRup() == False) and (GPIO.input(23))</pre>
          == True)):
 70
              time.sleep(.1)
 71
              usave()
 72
 73
      #tries to save data to a usb if there is one inserted
 74
      def usave():
 75
          folders = (os.popen("ls -l /media/pi/").readlines())
 76
          i = 0
 77
          flow = 0
 78
          while (i < len(folders)):</pre>
 79
              stringf = folders[i].split(" ")
 80
              if (len(folders[i]) > 15):
                   if (stringf[2] == "pi"):
 81
                       stringf = folders[i].split(":")
 82
 83
                       if (len(stringf) == 1):
 84
                           stringf = stringf[0].split("196")
 85
                       stringf = stringf[1].split(" ")
 86
                       h = len(stringf)
 87
                       q = 1
 88
                       folders[i] = stringf[g]
 89
                       g = g + 1
 90
                       while (q < h):
 91
                           folders[i] = folders[i] + " " + stringf[g]
 92
                           g = g + 1
 93
                       folders[i] = folders[i].rstrip()
 94
                       print (folders[i] + " device has been inserted.\n")
 95
                       GPIO.output (6,1)
 96
                       usbfolders = os.listdir("/media/pi/" + folders[i])
 97
                       j = 0
 98
                       while (j < len(usbfolders)):</pre>
 99
                           if(usbfolders[j] == 'Data'):
100
                               k = 0
101
                               dirfiles = os.listdir("/media/pi/" + folders[i] + '/' +
                               usbfolders[j])
102
                               while (k < len(dirfiles)):</pre>
103
                                    if(dirfiles[k] == '1-AIRSPEED.txt'):
104
                                        if (lps('/media/pi/' + folders[i] +
                                        '/Data/1-AIRSPEED.txt')):
105
                                            shutil.copyfile('/media/pi/' + folders[i] +
                                            '/Data/1-AIRSPEED.txt',
                                            '/home/pi/Desktop/Programs/Data/1-AIRSPEED.txt')
106
                                    k = k + 1
107
                                shutil.rmtree('/media/pi/' + folders[i] + '/Data')
108
                               time.sleep(0.1)
109
                           j = j + 1
110
                       shutil.copytree('/home/pi/Desktop/Programs/Data','/media/pi/' +
                       folders[i] + '/Data')
111
                       time.sleep (0.1)
112
                       os.system('sudo udisks --unmount /dev/sda1')
113
                       GPIO.output(6, 0)
114
                       time.sleep (0.4)
115
                       flash (6,1,0.1)
                       flash(6,0,0.1)
116
117
                       flash(6,1,0.1)
                       print (folders[i] + " device has been removed.\n")
118
119
               i = i + 1
120
          GPIO.output (6, 0)
121
122
      #reads the LPS file drive and assigns a new LPS amount
     def lps(directory):
123
124
          global flow
125
          with open (directory, "r") as ff:
126
              fstring = ff.read()
127
              if(len(fstring) > 3):
```

```
128
                  if (fstring[1] == '1'):
129
                      start = 1;
130
                  elif (fstring[2] == '1'):
131
                      start = 2
132
                  elif (fstring[3] == 'l'):
133
                      start = 3
134
                  elif (fstring[4] == 'l'):
135
                      start = 4
                  if ((fstring[start] == 'l') and (fstring[start + 1] == 'p') and
136
                   (fstring[start + 2] == 's')):
137
                      print (fstring)
138
                      k = 0
139
                      lstring = ''
140
                      while (k < start):</pre>
141
                           lstring = lstring + fstring[k]
142
                          k = k + 1
143
                      flow = int(lstring)
144
                      return True
145
              return False
146
147
      #START OF MAIN PROGRAM
148
149
      lps("/home/pi/Desktop/Programs/Data/1-AIRSPEED.txt")
150
                                              #sets LPS
151
      while 1:
                                               #loops indefinitley
152
          GPIO.setmode (GPIO.BCM)
                                               #set up GPIO using BCM numbering
153
          GPIO.setup(5, GPIO.OUT)
154
          GPIO.setup(17, GPIO.OUT)
155
          GPIO.setup(6, GPIO.OUT)
156
          GPIO.setup(24, GPIO.IN)
157
          GPIO.setup(23, GPIO.IN, pull_up_down = GPIO.PUD_UP)
158
          usave()
                                               #tries to copy data to usb
159
          if IRup():
                                               #checks the vent position and waits a diferent
          amount
160
              otime = time.time()
                                               #of time depending on the last time the vent
              was open
161
              wait(1, oldtime)
                                               #if the vent is open now
162
          elif ((time.time() - otime) <= 300):</pre>
163
              wait(10, oldtime)
                                               #if the vent was last open under 300 secs ago
164
          elif ((time.time() - otime) <= 900):</pre>
165
              wait(30, oldtime)
                                               #if the vent was last open under 900 secs ago
166
          else:
167
              wait(60, oldtime)
                                               #else
168
          oldtime = time.time()
                                               #saves the measurement start time
169
          GPIO.output (5, 1)
                                               #turns on the measurement light
170
          temp1 = read temp(device file1)
                                               #reads the first temperature
171
          IRup()
172
          flash(5, 0, 0.1)
                                               #flashes LED and updates vent
173
          IRup()
174
          temp2 = read temp(device file2)
                                               #reads the second temperature
175
          IRup()
176
          flash(5, 0, 0.1)
                                               #flashes LED and updates vent
177
                                               #creates date string
          IRup()
          date2 = '/home/pi/Desktop/Programs/Data/' + str((time.localtime()).tm year) + "-" +
178
          str((time.localtime()).tm mon) + "-" + str((time.localtime()).tm mday) + '.csv'
179
          with open(date2, 'a') as f:
                                               #opens file to save data
180
              if (f.tell() == 0):
                                               #adds headers if it's a new file
181
                  f.write("Day of month, Date, Vent Temperature, Ambient Temperature, Difference
                  in Temperature, Vent Open/Closed, Power (W), Energy (kWh), Total Energy (kWh) \n")
182
              timedata = str(time.ctime()).split(" ")
183
              if (timedata[2] == ''):
                                               #splits up date and time to reformat it
184
                  timedata[2] = timedata[3]
185
                  timedata[3] = timedata[4]
                  timedata[4] = timedata[5] #reformats date string
186
187
              savetime = "\"" + timedata[1] + " " + timedata[2] + ", " + timedata[4] + " " +
              timedata[3] + "\""
188
              tempdata = str((time.localtime()).tm mday) + "," + savetime + ',' +
```

```
temp2), 3)) + ',' + str(IRup())
189
            if IRup():
                                         #calculates power if the vent is open
190
                if(totalen == 0):
                   oldtime2 = time.time() - 2.7
191
                power = 1.125 * flow * (temp1 - temp2)
192
193
                energy = (power * (time.time() - oldtime2)) / 3600 / 1000
                totalen = totalen + energy
194
                tempdata = tempdata + ',' + str(round(power, 3)) + ',' + str(round(totalen,
195
                3)) + ",0"
196
                                         #adds zeroes if the vent is closed
            else:
197
                energy = 0
198
                power = 0
199
                if (not(totalen == 0)):
                 tempdata = tempdata + ",0,0," + str(round(totalen, 3))
200
201
                 dayen = dayen + totalen #adds the total energy to the end
202
                 totalen = 0
203
                else:
204
                   tempdata = tempdata + ",0,0,0"
205
            print (str(round(time.time() - oldtime2, 2)) + " seconds")
206
            oldtime2 = time.time()
                                        #saves the last time power was calculated
207
            if (startup):
                                         #adds a line to show the start of the program
208
                tempdata = tempdata + ",Program has started\n"
209
                                         #or adds the line terminator
            else:
210
                tempdata = tempdata + '\n'
211
            f.write(tempdata)
                                         #saves the data
212
         if ((time.localtime()).tm mday != oldday.tm mday):
213
            with open('/home/pi/Desktop/Programs/Data/2-DAILYDATA.csv', 'a') as f:
214
                f.write(str(oldday.tm year) + '/' + str(oldday.tm mon) + '/' +
                str(oldday.tm_mday) + ',' + str(round(dayen,6)) + '\n')
215
                                        #if the day has changed the program will save
            oldday = time.localtime()
            total kWh
216
            dayen = 0
                                         #reset total
217
218
219
         os.system("clear")
         print(savetime.split("\"")[1] + '\n')
220
        print(str(flow) + "lps")
221
222
        print("\nAmbient Temperature")
         print("Degrees C 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
223
         31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50")
224
                        print("
            temp2a = (str(format(temp2, '.3f'))) + " - |"
225
226
         temp2 = temp2 - 0.5
227
         while (temp2 >= 15):
            temp2a = temp2a + "="
228
229
            temp2 = temp2 - 0.25
         temp2a = temp2a + "|"
230
231
        print(temp2a)
232
        print("\nVent Temperature")
        print("Degrees C 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
233
         31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50")
         print("
234
                        temp1a = (str(format(temp1, '.3f'))) + " - |"
                                                  235
236
         temp1 = temp1 - 0.5
237
         while (temp1 >= 15):
            temp1a = temp1a + "="
238
239
            temp1 = temp1 - 0.25
         temp1a = temp1a + "|"
240
241
         print(templa)
242
243
         if (IRup()):
244
            print("\nPower")
            print("Power kW 0.0
                                  0.1
                                         0.2
                                                0.3
                                                       0.4
                                                              0.5
245
            0.7
                   0.8
                         0.9
                                1.0
                                        1.1
                                               1.2
                                                       1.3
                                                              1.4
                                                                    1.5
```

str(round(temp1, 3)) + ',' + str(round(temp2, 3)) + ',' + str(round(abs(temp1 - 3))) + ',' + str(round(abs(temp1 - 3)))

```
1.7")
                           . 1
                                 246
            print("
                                                | " )
            powera = (str(format(power / 1000, '.4f'))) + " - |"
247
248
            power = power - 25
249
            while (power >= 0):
250
               powera = powera + "="
251
               power = power - 12.5
252
            powera = powera + "|"
253
            print(powera)
254
            print("\nEnergy")
255
            print("Energy kWh 0.0 0.1 0.2 0.3 0.4 0.5
                                                                    0.6
                        0.9 1.0 1.1
                                              1.2
                                                      1.3
                                                            1.4
                                                                    1.5
            0.7 0.8
            1.7")
256
            print("
                                  totalena = (str(format(totalen, '.4f'))) + " - |"
257
            totalenb = totalen - 0.025
258
259
            while (totalenb >= 0):
               totalena = totalena + "="
260
261
               totalenb = totalen - 0.0125
            totalena = totalena + "|"
262
263
            print(totalena)
264
265
        print("\n\n\n")
266
267
268
        IRup()
                                        #updates vent
269
        GPIO.output (5, 0)
                                        #flashes LED to show end of measurement
270
        flash(5, 1, 0.1)
271
        flash(5, 0, 0.1)
        flash(5, 1, 0.1)
272
273
        time.sleep(0.5)
                                         #delay to next read
274
        IRup()
                                         #updates vent
275
        startup = 0
276
277
    #END OF MAIN PROGRAM
278
279
280
281
282
283
284
```

285