```
import os
import glob
import time
import RPi.GPIO as GPIO
import sys
import shutil
GPIO.setwarnings(False)

#sets up IO and temp readings
os.system('modprobe w1-gpio')
os.system('modprobe w1-therm')

base_dir = '/sys/bus/w1/devices/'
#assigns variable to each temperature sensor address
device_file1 = glob.glob(base_dir + '28-0000074b7f2d')[0] + '/w1_slave'
device_file2 = glob.glob(base_dir + '28-0000074b8662')[0] + '/w1_slave'
#variable definitions
oldtime = time.time() - 60
oldtime2 = time.time()
oldday = time.localtime()
otime = time.time()
ostate = 0
startup = 1
global  flow
flow = 47
energy = 0
totalen = 0
power = 0
dayen = 0

#flashes led on call
def flash(pin, state, wait):
    boo = not state
    GPIO.output(pin, state)
    time.sleep(wait)
    GPIO.output(pin, boo)

#opens temperature sensor when passed address
def read_temp_raw(d_file):
    f = open(d_file, 'r')
    lines = f.readlines()
    f.close()
    return lines

#reads and returns the temperature
def read_temp(d_file):
    lines = read_temp_raw(d_file)
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = read_temp_raw1()
    equals_pos = lines[1].find('t=')
    if equals_pos != -1:
        temp_string = lines[1][equals_pos+2:]
        temp_c = float(temp_string) / 1000.0
```

```
        return temp_c

#updates the vent LED and returns the vent state
def IRup():
    if GPIO.input(24):
        GPIO.output(17, 0)
        state = False
    else:
        GPIO.output(17, 1)
        state = True
    return state

#delays the program until specified time or until the measurement button has been pushed or the
vent opens
def wait(itime, oldtime):
    while(((time.time() - oldtime) < itime) and (IRup() == False) and (GPIO.input(23) == True)):
        time.sleep(.1)
        usave()

#tries to save data to a usb if there is one inserted
def usave():
    folders = (os.popen("ls -l /media/pi/").readlines())
    i = 0
    flow = 0
    while (i < len(folders)):
        stringf = folders[i].split(" ")
        if (len(folders[i]) > 15):
            if (stringf[2] == "pi"):
                stringf = folders[i].split(":")
                if (len(stringf) == 1):
                    stringf = stringf[0].split("196")
                stringf = stringf[1].split(" ")
                h = len(stringf)
                g = 1
                folders[i] = stringf[g]
                g = g + 1
                while (g < h):
                    folders[i] = folders[i] + " " + stringf[g]
                    g = g + 1
                folders[i] = folders[i].rstrip()
                print (folders[i] + " device has been inserted.\n")
                GPIO.output(6,1)
                usbfolders = os.listdir("/media/pi/" + folders[i])
                j = 0
                while (j < len(usbfolders)):
                    if(usbfolders[j] == 'Data'):
                        k = 0
                        dirfiles = os.listdir("/media/pi/" + folders[i] + '/' + usbfolders[j])
                        while (k < len(dirfiles)):
                            if(dirfiles[k] == '1-AIRSPEED.txt'):
                                if (lps('/media/pi/' + folders[i] + '/Data/1-AIRSPEED.txt')):
                                    shutil.copyfile('/media/pi/' + folders[i] +
                                    '/Data/1-AIRSPEED.txt',
                                    '/home/pi/Desktop/Programs/Data/1-AIRSPEED.txt')
```

```
                            k = k + 1
                        shutil.rmtree('/media/pi/' + folders[i] + '/Data')
                        time.sleep(0.1)
                    j = j + 1
                shutil.copytree('/home/pi/Desktop/Programs/Data','/media/pi/' + folders[i] +
                '/Data')
                time.sleep(0.1)
                os.system('sudo udisks --unmount /dev/sda1')
                GPIO.output(6, 0)
                time.sleep(0.4)
                flash(6,1,0.1)
                flash(6,0,0.1)
                flash(6,1,0.1)
                print (folders[i] + " device has been removed.\n")
        i = i + 1
    GPIO.output(6, 0)

#reads the LPS file drive and assigns a new LPS amount
def lps(directory):
    global flow
    with open(directory, "r") as ff:
        fstring = ff.read()
        if(len(fstring) > 3):
            if (fstring[1] == 'l'):
                start = 1;
            elif (fstring[2] == 'l'):
                start = 2
            elif (fstring[3] == 'l'):
                start = 3
            elif (fstring[4] == 'l'):
                start = 4
            if ((fstring[start] == 'l') and (fstring[start + 1] == 'p') and (fstring[start + 2]
            == 's')):
                print (fstring)
                k = 0
                lstring = ''
                while (k < start):
                    lstring = lstring + fstring[k]
                    k = k + 1
                flow = int(lstring)
                return True
        return False

#START OF MAIN PROGRAM

lps("/home/pi/Desktop/Programs/Data/1-AIRSPEED.txt")
                                    #sets LPS
while 1:                             #loops indefinitley
    GPIO.setmode(GPIO.BCM)           #set up GPIO using BCM numbering
    GPIO.setup(5, GPIO.OUT)
    GPIO.setup(17, GPIO.OUT)
    GPIO.setup(6, GPIO.OUT)
    GPIO.setup(24, GPIO.IN)
    GPIO.setup(23, GPIO.IN, pull_up_down = GPIO.PUD_UP)
```

```
    usave()                             #tries to copy data to usb
    if IRup():                          #checks the vent position and waits a diferent amount
        otime = time.time()             #of time depending on the last time the vent was open
        wait(1, oldtime)                #if the vent is open now
    elif ((time.time() - otime) <= 300):
        wait(10, oldtime)               #if the vent was last open under 300 secs ago
    elif ((time.time() - otime) <= 900):
        wait(30, oldtime)               #if the vent was last open under 900 secs ago
    else:
        wait(60, oldtime)               #else
    oldtime = time.time()               #saves the measurement start time
    GPIO.output(5, 1)                   #turns on the measurement light
    temp1 = read_temp(device_file1)     #reads the first temperature
    IRup()
    flash(5, 0, 0.1)                    #flashes LED and updates vent
    IRup()
    temp2 = read_temp(device_file2)     #reads the second temperature
    IRup()
    flash(5, 0, 0.1)                    #flashes LED and updates vent
    IRup()                              #creates date string
    date2 = '/home/pi/Desktop/Programs/Data/' + str((time.localtime()).tm_year) + "-" +
    str((time.localtime()).tm_mon) + "-" + str((time.localtime()).tm_mday) + '.csv'
    with open(date2, 'a') as f:         #opens file to save data
        if (f.tell() == 0):             #adds headers if it's a new file
            f.write("Day of month,Date,Vent Temperature,Ambient Temperature,Difference in
            Temperature,Vent Open/Closed,Power(W),Energy(kWh),Total Energy(kWh)\n")
        timedata = str(time.ctime()).split(" ")
        if (timedata[2] == ''):         #splits up date and time to reformat it
            timedata[2] = timedata[3]
            timedata[3] = timedata[4]
            timedata[4] = timedata[5]   #reformats date string
        savetime = "\"" + timedata[1] + " " + timedata[2] + ", " + timedata[4] + " " +
        timedata[3] + "\""
        tempdata = str((time.localtime()).tm_mday) + "," + savetime + ',' + str(round(temp1,
        3)) + ',' + str(round(temp2, 3)) + ',' + str(round(abs(temp1 - temp2), 3)) + ',' +
        str(IRup())
        if IRup():                      #calculates power if the vent is open
            if(totalen == 0):
                oldtime2 = time.time() - 2.7
            power = 1.125 * flow * (temp1 - temp2)
            energy = (power * (time.time() - oldtime2)) / 3600 / 1000
            totalen = totalen + energy
            tempdata = tempdata + ',' + str(round(power, 3)) + ',' + str(round(totalen, 3)) +
            ",0"
        else:                           #adds zeroes if the vent is closed
            energy = 0
            power = 0
            if (not(totalen == 0)):
              tempdata = tempdata + ",0,0," + str(round(totalen, 3))
              dayen = dayen + totalen   #adds the total energy to the end
              totalen = 0
            else:
                tempdata = tempdata + ",0,0,0"
        print (str(round(time.time() - oldtime2, 2)) + " seconds")
```

```
        oldtime2 = time.time()          #saves the last time power was calculated
        if (startup):                   #adds a line to show the start of the program
            tempdata = tempdata + ",Program has started\n"
        else:                           #or adds the line terminator
            tempdata = tempdata + '\n'
        f.write(tempdata)               #saves the data
    if ((time.localtime()).tm_mday != oldday.tm_mday):
        with open('/home/pi/Desktop/Programs/Data/2-DAILYDATA.csv', 'a') as f:
            f.write(str(oldday.tm_year) + '/' + str(oldday.tm_mon) + '/' + str(oldday.tm_mday)
            + ',' + str(round(dayen,6)) + '\n')
        oldday = time.localtime()       #if the day has changed the program will save total kWh
        dayen = 0                       #reset total


    os.system("clear")
    print(savetime.split("\"")[1] + '\n')
    print(str(flow) + "lps")
    print("\nAmbient Temperature")
    print("Degrees C  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32
    33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50")
    print("           |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |")
    temp2a = (str(format(temp2, '.3f'))) + "  -  |"
    temp2 = temp2 - 0.5
    while (temp2 >= 15):
        temp2a = temp2a + "="
        temp2 = temp2 - 0.25
    temp2a = temp2a + "|"
    print(temp2a)
    print("\nVent Temperature")
    print("Degrees C  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32
    33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50")
    print("           |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |")
    temp1a = (str(format(temp1, '.3f'))) + "  -  |"
    temp1 = temp1 - 0.5
    while (temp1 >= 15):
        temp1a = temp1a + "="
        temp1 = temp1 - 0.25
    temp1a = temp1a + "|"
    print(temp1a)

    if (IRup()):
        print("\nPower")
        print("Power kW   0.0     0.1     0.2     0.3     0.4     0.5     0.6     0.7
        0.8     0.9     1.0     1.1     1.2     1.3     1.4     1.5     1.6     1.7")
        print("           |       |       |       |       |       |       |       |
        |       |       |       |       |       |       |       |       |")
        powera = (str(format(power / 1000, '.4f'))) + "  -  |"
        power = power - 25
        while (power >= 0):
            powera = powera + "="
            power = power - 12.5
        powera = powera + "|"
```

```
        print(powera)
        print("\nEnergy")
        print("Energy kWh 0.0     0.1     0.2     0.3     0.4     0.5     0.6     0.7
        0.8     0.9     1.0     1.1     1.2     1.3     1.4     1.5     1.6     1.7")
        print("           |       |       |       |       |       |       |       |
        |       |       |       |       |       |       |       |       |")
        totalena = (str(format(totalen, '.4f'))) + "  -  |"
        totalenb = totalen - 0.025
        while (totalenb >= 0):
            totalena = totalena + "="
            totalenb = totalen - 0.0125
        totalena = totalena + "|"
        print(totalena)

    print("\n\n\n")


    IRup()                              #updates vent
    GPIO.output(5, 0)                   #flashes LED to show end of measurement
    flash(5, 1, 0.1)
    flash(5, 0, 0.1)
    flash(5, 1, 0.1)
    time.sleep(0.5)                      #delay to next read
    IRup()                               #updates vent
    startup = 0


#END OF MAIN PROGRAM
```