Final Project: Fan Regulator for Heating and Cooling

ECE 4220

Author: Adam Zeller

Instructor: Luis Rivera

May 2015

**Abstract**

The purpose of this project is to help to heat and cool all rooms in a house or building at an even rate. Therefore if there is a room that is significantly cooler that another room, a fan hidden in the duct work of the furnace will be able to turn on and pull more of the warm air into this cooler room while the rest of the rooms are being heated as normal. All of this will be implemented using a Raspberry Pi and its GPIO pins in order to read the current room temperature using a temperature sensor and control the amount of power needed to run the fan at specific speeds. The program will check the temperature and turn the different fans on for the different speeds needed to maintain the house at a constant temperature.

**Introduction**

Imagine a home, not just any home, but the very home that you grew up in. Now think back to all those times that you ended up in the same comfortable spot in this home. Now think as to why you chose that spot as a child. Did it have something to do with how warm it was in the coldest part of the winter, or how cool it was when it was scorching hot outside? Think if everywhere in your home was now this temperature instead of just that favorite room or closet, or nook. Do you think you would have been more inclined to venture to other parts of the house to play or help out? Well, my project is going to help to make that more of a reality. As we all know, large home air conditioners and furnaces turn on and blow air through duct work within the house and it is all controlled by one individual fan at the source. When the air is blown throughout the house, it escapes at random through vents in the duct work and can provide very uneven heating or cooling throughout larger and smaller rooms. With this project, we can hope to control the amount of air flow within a room to help bring it to the correct temperature. By using a temperature sensor, I will be able to regulate how much of a temperature difference there is between the different rooms and adjust the speed of different fans located at air vents in these set rooms. From there, rooms should be able to reach the desired temperature around the same time and continue to heat and cool at the same rate.

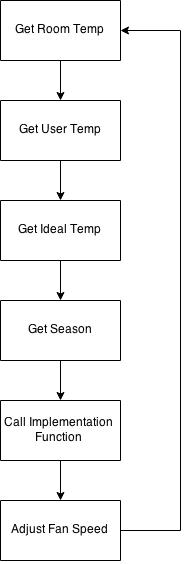
**Problem Statement**

I plan to use a Raspberry Pi to control the sensor that will be monitoring the temperature and controlling the voltage sent to the fans as to determine how fast they are needed to blow air through a vent to cause even air control. After the information is received, the program decides where the different temperatures are in comparison with the base temperature needed. For this project, I will only be using one sensor and a asking the user to input the second room temperature. I will use two fans to demonstrate the different speeds of the fans throughout the system. Through this project, I hope to gain a better understanding of how things are implemented on a Raspberry Pi. I will also have to research how to use the GPIO pins and output voltages on a Raspberry Pi as well as learning Python.

I will be coding the temperature sensor in Python and using different things from this year’s class to implement the project. These items include threads, flags, registers, and kernels. I will be using a kernel to read the temperature sensor at a given time interval and this is written to a file that is accessible to the main program. The main program will then be using different flags and registers to control what will happen.

As far as the software goes, the first thing I will need to learn is how to work with the GPIO pins on the Raspberry Pi. The next thing is that I would need to learn to go through the code generated by the sensor and then be able to set controls as to the different fan speeds needed. This fan speed program would be preprogramed for slowing the fan as the temperature of the room reaches the optimal level which is set by the user in degrees Fahrenheit. The user is also able to set the different season to control how the temperatures affect the fan from the base temperature. In with the season controls, there is also an option to turn off the fans or turn them on.

**Hardware Flowchart**



**Proposed Approach**

After some research, I found that there was a lot of information about GPIO pins on the Raspberry Pi and a lot of people having success with the DS18B20 temperature sensor. I was able to use this sensor with GPIO pin 4 along with a 3.3V power supply and ground. There was already Python code written to use this sensor, so I took some time to investigate how it operates. The sensor automatically writes a file with different temperatures based on the time and then saves this file to be accessible on the Raspberry Pi. The code then reads through this file and breaks it into lines using a newline character. It then takes checks for the letter ‘Y’, which should be 3 characters from the end of one of the lines. Found within that same line is a ‘t=’ which represents the temperature in degrees Celsius. At this point, I modified the code to only send back the degrees Celsius and then I converted it to Fahrenheit to be used with the rest of my code.

The next hardest part came when trying to implement fans using the GPIO pins. Without using a potentiometer, I was at a loss on how to change the voltage leaving the Raspberry Pi. Then I realized my next big problem and that was that in turning on a single GPIO pin and connecting a fan to it, the output voltage could dimly turn the light on the fan on but wasn’t even enough to turn the fan blades. Then after giving it some thought and knowing it was too late to order a digital potentiometer and learn how to use it, I decided to test the voltages of two GPIO pins at the same time. This was enough power to keep the fan going, but not actually start it. Finally after combining three GPIO pins I was able to get it to work properly. I decided this would be the way that I could change speeds on the fan. Using GPIO pins 22, 26, and 27 for the first fan and pins 23, 24, and 25 for the second fan, I could turn all three on for high speed, two on for medium speed, and 1 on to resemble low speed. In reality, the low speed does not turn the fan blades, but does have a very dim lit light shown.

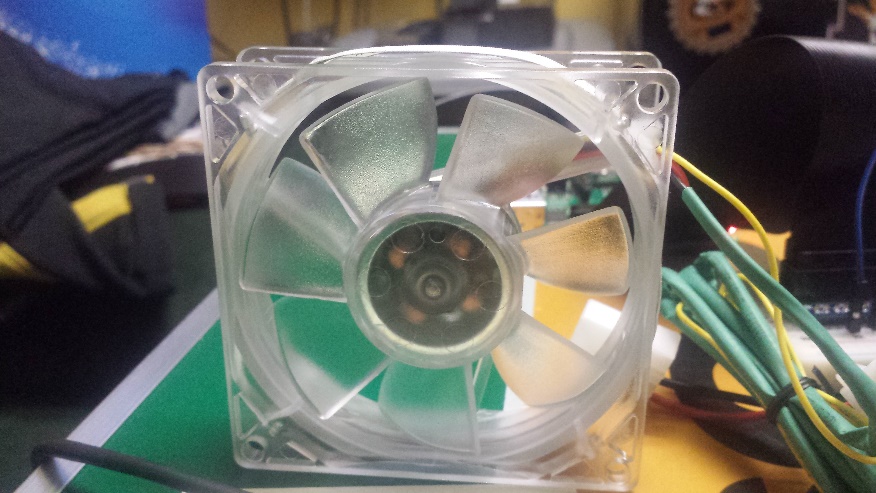


Figure : Low (left), Medium (center), and High (right) views of the fans

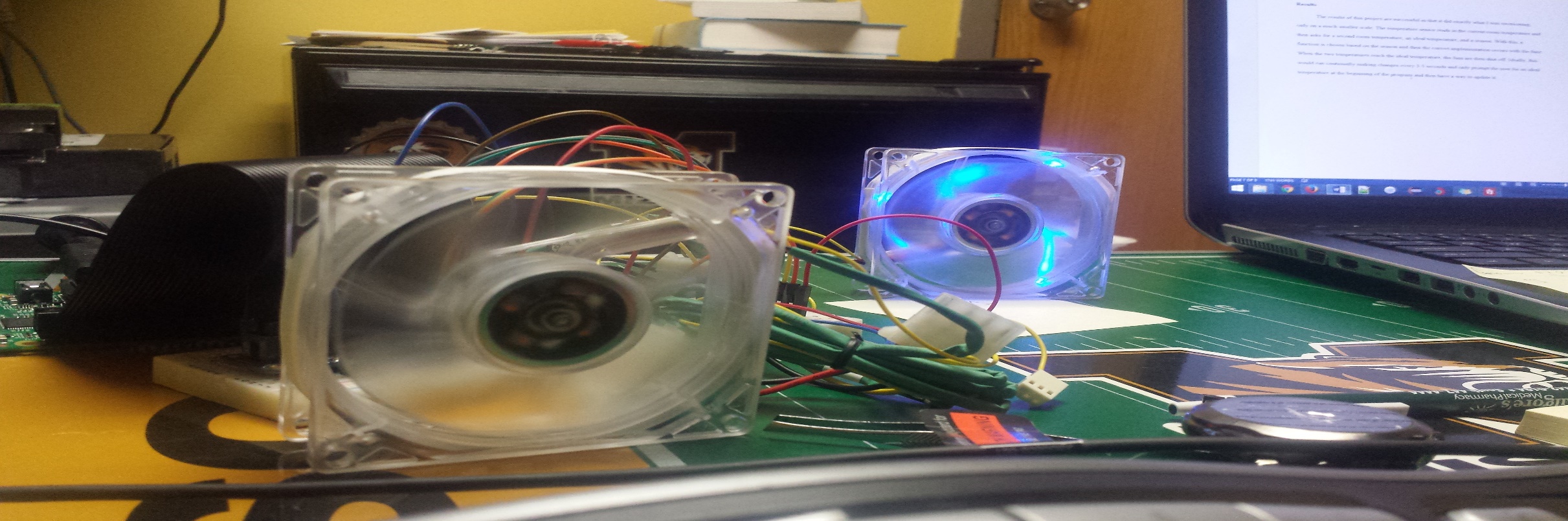


Figure : Fans at Medium (left) and High (right)

After learning of this, I wrote up two separate functions doing very similar tasks of calculating speeds and turning on the correct pins. I decided to implement the fans based off of differences in 3 degrees Fahrenheit, so that if the ideal temperature was more than 9 degrees away from the room temperature, the fans would be on full speed. It also takes into account the difference between the two room temperatures and determines that if one temperature is less ideal than the other, then the less ideal room is at a higher speed. Once the rooms get within 3 degrees Fahrenheit of the ideal temperature, the fans will completely turn off. This is because a typical AC or heating unit can evenly cool and heat this remaining 3 degrees very easily.

The program has a user input section built in so that every three seconds it asks you to input a new second temperature, an ideal room temperature, and a season. The season prompt also allows you to turn the fans on and off and go completely against the implementation. This user section is done this way for testing purposes only and can be changed to only require an input for the ideal temperature and then read a switch for the season.

**Results**

The results of this project are successful in that it did exactly what I was envisioning, only on a much smaller scale. The temperature sensor reads in the current room temperature and then asks for a second room temperature, an ideal temperature, and a season. With this, a function is chosen based on the season and then the correct implementation occurs with the fans. When the two temperatures reach the ideal temperature, the fans are then shut off. Ideally, this would run continually making changes every 3-5 seconds and only prompt the user for an ideal temperature at the beginning of the program and then have a way to update it.

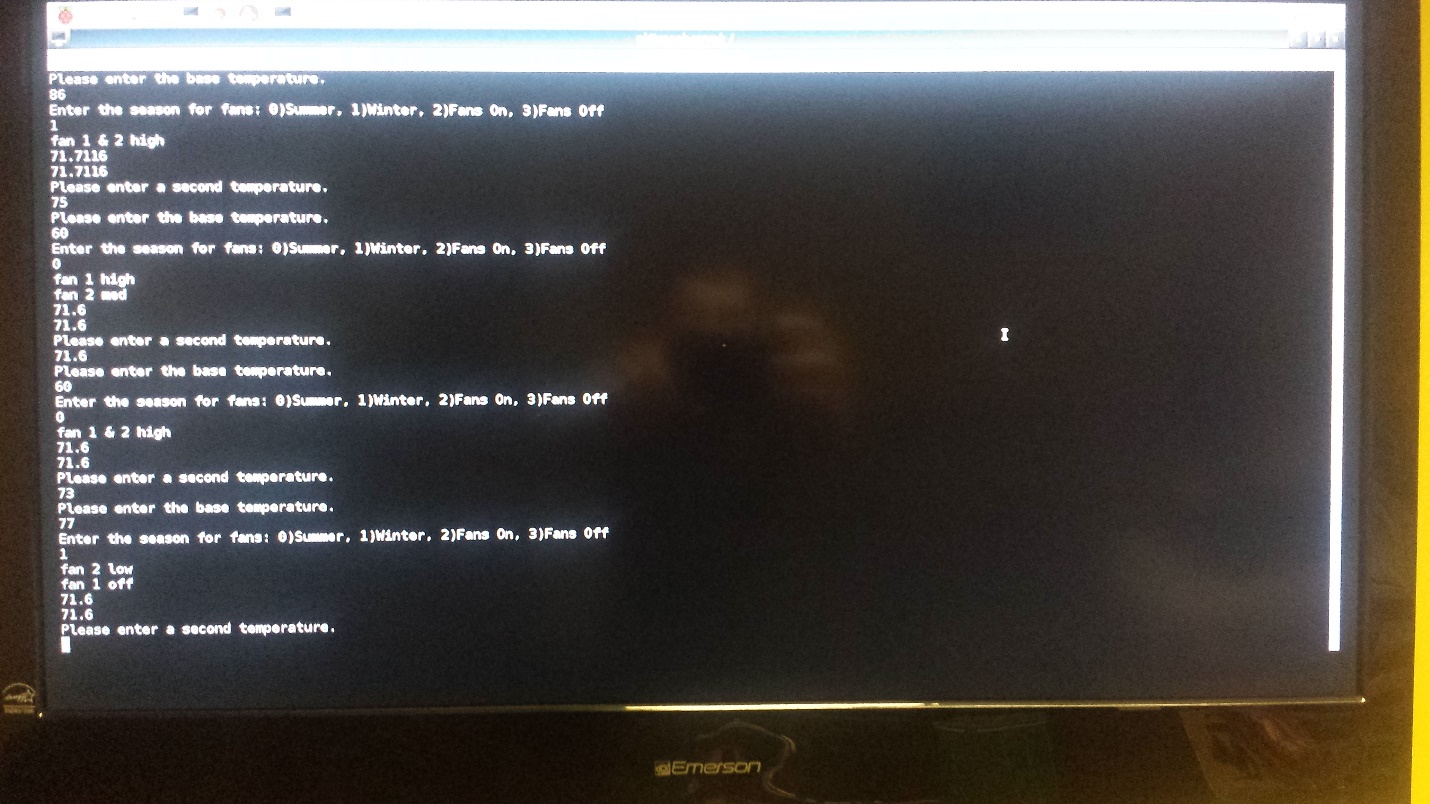


Figure : Screen shot of Program running

**Conclusion**

The tools required to complete this project were mainly learned from lectures and labs from this class. I was able to attach hardware to an embedded system and then design a program on it to complete the task at hand. By using the DS18B20 temperature sensor to get the room temperature, I was able change the fan speeds to simulate a live result of what would happen in an air duct of a home. Over all, this project was a lot of fun and it allowed me to use the Raspberry Pi, which I had been wanting to do for a while only never gotten the chance. I also got to learn how to code in Python which has been on my list of things to do as well. I learned a great deal about the Raspberry Pi and how it uses its GPIO pins for controlling different components attached to them. This will definitely help me in my future classes and with projects both at home and in my future job. Being able to work with embedded systems during this project and throughout this year has truly given me a different look at how electronics can be built and operate and how rewarding it is to see a project through from start to finish to see your goals accomplished.

**Code:**

import RPi.GPIO as GPIO

import os

import glob

import time

os.system('modprobe w1-gpio')

os.system('modprobe w1-therm')

base\_dir = '/sys/bus/w1/devices/'

device\_folder = glob.glob(base\_dir + '28\*')[0]

device\_file = device\_folder + '/w1\_slave'

GPIO.cleanup()

GPIO.setmode(GPIO.BCM)

fan1Pins = [22,26,27]

fan2Pins = [23,24,25]

for a in fan1Pins:

GPIO.setup(a, GPIO.OUT)

GPIO.output(a, False)

for a in fan2Pins:

GPIO.setup(a, GPIO.OUT)

GPIO.output(a, False)

seasonFlag = 0

def read\_temp\_raw():

f = open(device\_file, 'r')

lines = f.readlines()

f.close()

return lines

def read\_temp():

lines = read\_temp\_raw()

while lines[0].strip()[-3:] != 'YES':

time.sleep(0,2)

lines = read\_temp\_raw()

equals\_pos = lines[1].find('t=')

if equals\_pos != -1:

temp\_string = lines[1][equals\_pos+2:]

temp\_c = float(temp\_string) / 1000.0

temp\_f = temp\_c \* 9.0 / 5.0 +32.0

print(temp\_f)

return temp\_c

def fan\_Calculation\_Summer(t1,t2,t3):#t1-sensor, t2-input, t3-base temp

if t1 == t2:

r = t1-t3

d = r/3

if d >= 3:

for a in fan1Pins:

GPIO.output(a, True)

for a in fan2Pins:

GPIO.output(a,True)

print('fan 1 & 2 high')

elif d >= 2:

GPIO.output(22, True)

GPIO.output(26, True)

GPIO.output(27, False)

GPIO.output(23, True)

GPIO.output(24, True)

GPIO.output(25, False)

print('fan 1 & 2 med')

elif d >=1:

GPIO.output(22, True)

GPIO.output(26, False)

GPIO.output(27, False)

GPIO.output(23, True)

GPIO.output(24, False)

GPIO.output(25, False)

print('fan 1 & 2 low')

else:

for a in fan1Pins:

GPIO.output(a, False)

for a in fan2Pins:

GPIO.output(a, False)

print('fan 1 & 2 off')

elif t1<t2:

r = t2-t3

d = r/3

if d >= 3:

for a in fan1Pins:

GPIO.output(a, True)

print('fan 1 high')

if (t1-t3)/3 >=3:

GPIO.output(23, True)

GPIO.output(24, True)

GPIO.output(25, False)

print('fan 2 med')

elif (t1-t3)/3 >=2:

GPIO.output(23, True)

GPIO.output(24, False)

GPIO.output(25, False)

print('fan 2 low')

else:

GPIO.output(23, False)

GPIO.output(24, False)

GPIO.output(25, False)

print('fan 2 off')

elif d >=2:

GPIO.output(22, True)

GPIO.output(27, True)

GPIO.output(26, False)

print('fan 1 med')

if (t1-t3)/3 >=1:

GPIO.output(23, True)

GPIO.output(24, False)

GPIO.output(25, False)

print('fan 2 low')

else:

GPIO.output(23, False)

GPIO.output(24, False)

GPIO.output(25, False)

print('fan 2 off')

elif d >=1:

GPIO.output(22, True)

GPIO.output(27, False)

GPIO.output(26, False)

GPIO.output(23, False)

GPIO.output(24, False)

GPIO.output(25, False)

print('fan 1 low')

print('fan 2 off')

else:

GPIO.output(22, False)

GPIO.output(27, False)

GPIO.output(26, False)

GPIO.output(23, False)

GPIO.output(24, False)

GPIO.output(25, False)

print('fan 1 & 2 off')

elif t1>t2:

r = t1-t3

d = r/3

if d >= 3:

for a in fan2Pins:

GPIO.output(a, True)

print('fan 2 high')

if (t2-t3)/3 >=3:

GPIO.output(22, True)

GPIO.output(27, True)

GPIO.output(26, False)

print('fan 1 med')

elif (t2-t3)/3 >=2:

GPIO.output(22, True)

GPIO.output(27, False)

GPIO.output(26, False)

print('fan 1 low')

else:

GPIO.output(22, False)

GPIO.output(27, False)

GPIO.output(26, False)

print('fan 1 off')

elif d >=2:

GPIO.output(23, True)

GPIO.output(24, True)

GPIO.output(25, False)

print('fan 2 med')

if (t3-t2)/3 >=1:

GPIO.output(22, True)

GPIO.output(27, False)

GPIO.output(26, False)

print('fan 1 low')

else:

GPIO.output(22, False)

GPIO.output(27, False)

GPIO.output(26, False)

print('fan 1 off')

elif d >=1:

GPIO.output(23, True)

GPIO.output(24, False)

GPIO.output(25, False)

GPIO.output(22, False)

GPIO.output(27, False)

GPIO.output(26, False)

print('fan 2 low')

print('fan 1 off')

else:

GPIO.output(23, False)

GPIO.output(24, False)

GPIO.output(25, False)

GPIO.output(22, False)

GPIO.output(27, False)

GPIO.output(26, False)

print('fan 1 & 2 off')

else:

for a in fan1Pins:

GPIO.output(a, False)

for a in fan2Pins:

GPIO.output(a, False)

print('fan 1 & 2 off')

return r

def fan\_Calculation\_Winter(t1,t2,t3):#t1-sensor, t2-input, t3-base temp

if t1 == t2:

r = t3-t1

d = r/3

if d >= 3:

for a in fan1Pins:

GPIO.output(a, True)

for a in fan2Pins:

GPIO.output(a,True)

print('fan 1 & 2 high')

elif d >= 2:

GPIO.output(22, True)

GPIO.output(26, True)

GPIO.output(27, False)

GPIO.output(23, True)

GPIO.output(24, True)

GPIO.output(25, False)

print('fan 1 & 2 med')

elif d >=1:

GPIO.output(22, True)

GPIO.output(26, False)

GPIO.output(27, False)

GPIO.output(23, True)

GPIO.output(24, False)

GPIO.output(25, False)

print('fan 1 & 2 low')

else:

for a in fan1Pins:

GPIO.output(a, False)

for a in fan2Pins:

GPIO.output(a, False)

print('fan 1 & 2 off')

elif t1>t2:

r = t3-t2

d = r/3

if d >= 3:

for a in fan1Pins:

GPIO.output(a, True)

print('fan 1 high')

if (t3-t1)/3 >=3:

GPIO.output(23, True)

GPIO.output(24, True)

GPIO.output(25, False)

print('fan 2 med')

elif (t3-t1)/3 >=2:

GPIO.output(23, True)

GPIO.output(24, False)

GPIO.output(25, False)

print('fan 2 low')

else:

GPIO.output(23, False)

GPIO.output(24, False)

GPIO.output(25, False)

print('fan 2 off')

elif d >=2:

GPIO.output(22, True)

GPIO.output(27, True)

GPIO.output(26, False)

print('fan 1 med')

if (t3-t2)/3 >=1:

GPIO.output(23, True)

GPIO.output(24, False)

GPIO.output(25, False)

print('fan 2 low')

else:

GPIO.output(23, False)

GPIO.output(24, False)

GPIO.output(25, False)

print('fan 2 off')

elif d >=1:

GPIO.output(22, True)

GPIO.output(27, False)

GPIO.output(26, False)

GPIO.output(23, False)

GPIO.output(24, False)

GPIO.output(25, False)

print('fan 1 low')

print('fan 2 off')

else:

GPIO.output(22, False)

GPIO.output(27, False)

GPIO.output(26, False)

GPIO.output(23, False)

GPIO.output(24, False)

GPIO.output(25, False)

print('fan 1 & 2 off')

elif t2>t1:

r = t3-t1

d = r/3

if d >= 3:

for a in fan2Pins:

GPIO.output(a, True)

print('fan 2 high')

if (t3-t2)/3 >=3:

GPIO.output(22, True)

GPIO.output(27, True)

GPIO.output(26, False)

print('fan 1 med')

elif (t3-t2)/3 >=2:

GPIO.output(22, True)

GPIO.output(27, False)

GPIO.output(26, False)

print('fan 1 low')

else:

GPIO.output(22, False)

GPIO.output(27, False)

GPIO.output(26, False)

print('fan 1 off')

elif d >=2:

GPIO.output(23, True)

GPIO.output(24, True)

GPIO.output(25, False)

print('fan 2 med')

if (t3-t2)/3 >=1:

GPIO.output(22, True)

GPIO.output(27, False)

GPIO.output(26, False)

print('fan 1 low')

else:

GPIO.output(22, False)

GPIO.output(27, False)

GPIO.output(26, False)

print('fan 1 off')

elif d >=1:

GPIO.output(23, True)

GPIO.output(24, False)

GPIO.output(25, False)

GPIO.output(22, False)

GPIO.output(27, False)

GPIO.output(26, False)

print('fan 2 low')

print('fan 1 off')

else:

GPIO.output(23, False)

GPIO.output(24, False)

GPIO.output(25, False)

GPIO.output(22, False)

GPIO.output(27, False)

GPIO.output(26, False)

print('fan 1 & 2 off')

else:

for a in fan1Pins:

GPIO.output(a, False)

for a in fan2Pins:

GPIO.output(a, False)

print('fan 1 & 2 off')

return r

def fan\_on():

for a in fan1Pins:

GPIO.output(a,True)

print('fan 1 high')

for a in fan2Pins:

GPIO.output(a,True)

print('fan 2 high')

return a

while True:

c\_temp = read\_temp()

temp\_f1 = c\_temp \* 9.0 / 5.0 + 32.0

print(temp\_f1)

print('Please enter a second temperature.')

temp\_f2 = float(input())

print('Please enter the base temperature.')

base\_t = float(input())

print('Enter the season for fans: 0)Summer, 1)Winter, 2)Fans On, 3)Fans Off')

seasonFlag = int(input())

time.sleep(2)

if seasonFlag == 0:

fan\_Calculation\_Summer(temp\_f1, temp\_f2, base\_t)

elif seasonFlag == 1:

fan\_Calculation\_Winter (temp\_f1, temp\_f2, base\_t)

elif seasonFlag == 2:

fan\_on()

elif seasonFlag ==3:

for a in fan1Pins:

GPIO.output(a, False)

for a in fan2Pins:

GPIO.output(a, False)

print('fan 1 & 2 off')

time.sleep(3)