

```
# -*- coding: utf-8 -*-  
"""
```

```
Created on Mon Apr 22 10:08:00 2019
```

```
@author: maxhu  
"""
```

```
#=====IMPORTS, INITIALIZING, AND HOUSE CLEANING =====#  
import uControllersFunctions as F  
import RPi.GPIO as GPIO  
import time  
import numpy as np  
#=====#  
CONREL = 18  
C = 0  
CS = 32  
CLK = 12  
DOUT = 40  
DIN = 38  
FAN = 22  
#=====#  
ref = 3.304  
#=====#  
GPIO.setmode(GPIO.BOARD)  
GPIO.setup(CS, GPIO.OUT)  
GPIO.setup(CLK, GPIO.OUT)  
GPIO.setup(DOUT, GPIO.IN)  
GPIO.setup(DIN, GPIO.OUT)  
GPIO.setup(CONREL,GPIO.OUT)  
GPIO.setup(FAN,GPIO.OUT)  
#=====#  
freq = 1  
T_set = 30  
del_t = 0  
first_del_t = 0  
second_del_t = 0  
third_del_t = 0  
temp_data = []  
time_data = []  
#=====#  
my_pwm_relay = GPIO.PWM(CONREL, freq)  
my_pwm_fan = GPIO.PWM(FAN, freq)  
duty = 0  
duty_fan = 0  
my_pwm_relay.start(duty)  
my_pwm_fan.start(duty_fan)
```

```

#####
def duty_change(error):
    # too cold
    if error > 2:
        duty = 100
        my_pwm_relay.ChangeDutyCycle(duty)
        print('The duty cycle of the heater is: ', duty)
    elif 0 < error < 2:
        duty = 50 * error
        my_pwm_relay.ChangeDutyCycle(duty)
        print('The duty cycle of the heater is: ', duty)

    # too hot
    elif error < 0 and error > -2:
        duty_fan = np.abs(50 * error)
        my_pwm_fan.ChangeDutyCycle(duty_fan)
        print('The duty cycle of the fan is: ', duty_fan)
    elif error < -2:
        duty_fan = 100
        my_pwm_fan.ChangeDutyCycle(duty_fan)
        print('The duty cycle of the fan is: ', duty_fan)

#####DATA PROCESSING BEGINS HERE#####
try:
    current_temp = F.calc_temp(
        ref,F.calc_volts(F.readMCP(C,CS,CLK,DOUT,DIN), 'MCP',ref)
        , 'Thermistor')
    while current_temp < 30:
        current_temp = F.calc_temp(
            ref,F.calc_volts(F.readMCP(C,CS,CLK,DOUT,DIN), 'MCP',ref)
            , 'Thermistor')
        duty = 100
        my_pwm_relay.ChangeDutyCycle(duty)
        print(current_temp)
        time.sleep(3)

    first = True
    start_time = time.time()
    while first == True:
        current_temp = F.calc_temp(
            ref,F.calc_volts(F.readMCP(C,CS,CLK,DOUT,DIN), 'MCP',ref)
            , 'Thermistor')
#####WARMING UP/COOLING DOWN#####
        while current_temp < 30 or current_temp > 30.5:

            current_temp = F.calc_temp(
                ref,F.calc_volts(F.readMCP(C,CS,CLK,DOUT,DIN), 'MCP',ref)

```

```

        , 'Thermistor')

    if current_temp < 29:
        duty = 100
        my_pwm_relay.ChangeDutyCycle(duty)
    if current_temp > 31:
        duty = 0
        duty_fan = 100
        my_pwm_relay.ChangeDutyCycle(duty)
        my_pwm_fan.ChangeDutyCycle(duty_fan)

    print('The heater is at a duty of: ', duty, '\n',
          'The fan is at a duty of: ',
          duty_fan, '\n', 'The current temperature is: ',
          current_temp, '\n')

7 del_t = time.time() - start_time
  time_data.append(del_t)
  temp_data.append(current_temp)

  time.sleep(5)
#=====DATA COLLECTION FOR PROPORTION CONTROL W/ ACTIVE COOLING=====#
  first_start_time = time.time()
  while first_del_t < 1800:

      current_temp = F.calc_temp(
          ref, F.calc_volts(F.readMCP(C, CS, CLK, DOUT, DIN), 'MCP', ref)
          , 'Thermistor')

      T_err = T_set - current_temp
      duty_change(T_err)
      first_del_t = time.time() - first_start_time
      del_t = time.time() - start_time
      time_data.append(del_t)
      temp_data.append(current_temp)
      print('The current temperature is: ', current_temp, '\n')
      time.sleep(1)

  first = False

#=====PROPORTION CONTROL W/ ACTIVE COOLING DONE,=====#
#=====INITIALIZING BANG BANG=====#
  second = True
  start_time = time.time()

  while second == True:

```

```

current_temp = F.calc_temp(
    ref,F.calc_volts(F.readMCP(C,CS,CLK,DOUT,DIN),'MCP',ref)
    , 'Thermistor')

while current_temp < 30 or current_temp > 30.5:

    if current_temp < 29:
        duty = 100
        my_pwm_relay.ChangeDutyCycle(duty)

    if current_temp > 31:
        duty = 0
        my_pwm_relay.ChangeDutyCycle(duty)

    del_t = time.time() - start_time
    time_data.append(del_t)
    temp_data.append(current_temp)

    print('The heater is at a duty of: ',duty,'\n',
        'The current temperature is: ',
        current_temp, '\n')

    current_temp = F.calc_temp(
        ref,F.calc_volts(F.readMCP(C,CS,CLK,DOUT,DIN),'MCP',ref)
        , 'Thermistor')
#=====DATA COLLECTION==FOR BANG BANG=====#
    second_start_time = time.time()

    while second_del_t < 1800:

        current_temp = F.calc_temp(
            ref,F.calc_volts(F.readMCP(C,CS,CLK,DOUT,DIN),'MCP',ref)
            , 'Thermistor')

        if current_temp > 30.9:
            duty = 0
            my_pwm_relay.ChangeDutyCycle(duty)

        elif current_temp < 30:
            duty = 100
            my_pwm_relay.ChangeDutyCycle(duty)

        del_t = time.time() - start_time
        second_del_t = time.time() - second_start_time
        temp_data.append(current_temp)
        time_data.append(del_t)

```

```

        print('The current temperature is: ',
              current_temp, '\n')

        time.sleep(1)

        second = False

#=====BANG BANG DONE, INITIALIZING PROPORTIONAL CONTROL=====#
        third = True
        start_time = time.time()
        while third == True:

            current_temp = F.calc_temp(
                ref,F.calc_volts(F.readMCP(C,CS,CLK,DOUT,DIN),'MCP',ref)
                , 'Thermistor')

#=====WARMING UP/COOLING DOWN=====#
            while current_temp < 30 or current_temp > 30.5:

                if current_temp < 29:
                    duty = 100
                    my_pwm_relay.ChangeDutyCycle(duty)

                if current_temp > 31:
                    duty = 0
                    my_pwm_relay.ChangeDutyCycle(duty)

                print('The heater is at a duty of: ',duty,'\n',
                      '\n','The current temperature is: ',
                      current_temp, '\n')

                current_temp = F.calc_temp(
                    ref,F.calc_volts(F.readMCP(C,CS,CLK,DOUT,DIN),'MCP',
                    ref),'Thermistor')

                del_t = time.time() - start_time
                time_data.append(del_t)
                temp_data.append(current_temp)

                time.sleep(5)

#=====DATA COLLECTION FOR PROPORTION CONTROL W/O ACTIVE COOLING=====#
            third_start_time = time.time()
            while third_del_t < 1800:

                current_temp = F.calc_temp(
                    ref,F.calc_volts(F.readMCP(C,CS,CLK,DOUT,DIN),'MCP',
                    ref),'Thermistor')

```

```

    T_err = T_set - current_temp
    duty_change(T_err)
    third_del_t = time.time() - third_start_time
    del_t = time.time() - start_time
    time_data.append(del_t)
    temp_data.append(current_temp)

    time.sleep(1)

    third = False

#=====DATA SAVING=====#
    file = open('TempProcessControl.txt', 'w')
    for n in range(len(time_data)):
        file.write(str(time_data[n]) + ',' + str(temp_data[n]) + '\n')
    file.close()

except KeyboardInterrupt:
    my_pwm_relay.stop()
    my_pwm_fan.stop()
    print('toast got burnt')

finally:
    GPIO.cleanup()

```