**IoT Experiment -- 5**

**For “*Exp-5:***

**Temperature dependent auto-cooling system using Raspberry-pi 3” the Python program need to modified/re-written as per the following:**

***Design an auto cooling system using Raspberry-pi 3, DHT22 sensor, relay and a cooling fan such that:***

*(i) If temperature (reported by DHT sensor) > 30 degree Celsius, fan should automatically turn ON. Let the cool air from the fan chill down the DHT sensor to bring its surface temperature down in a steady fashion.*

*(ii) As soon as the temperature (reported by DHT) < 20 degree Celsius, the fan should automatically turn OFF. This will stop cool air supply to DHT sensor and its temperature will start rising. [Assuming a dry hot weather with temp ~ 35 degree Celsius]*

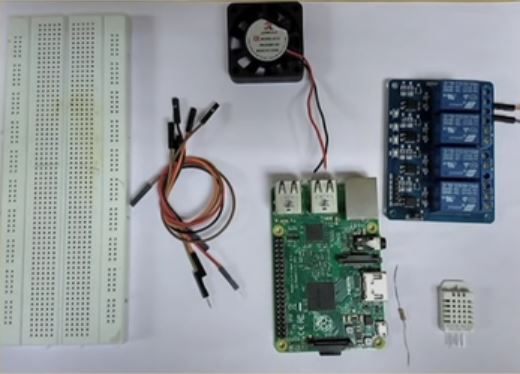
*(iii) Again, as soon as the temperature (reported by DHT sensor) > 30 degree Celsius, step (i) will be repeated followed by step (ii) and so on.*

*This will go on indefinitely until we kill the program (using ctrl+c).*

1. **Components required :**
   * DHT22 Sensor
   * 4.7K ohm resistor
   * Relay
   * Jumper wires
   * Raspberry Pi
   * Fan

1. **Procedure :**

* Connect pin 1 of DHT Sensor to 3.3V pin of raspberry pi
* Connect pin 2 of DHT Sensor to pin 7 of Raspberry pi
* Connect pin 4 of DHT Sensor to the ground pin of the pi
* Connect Vcc pin of relay to the 5V supply pin of pi
* Connect GND pin of relay to ground pin of the pi
* Connect the I/O signal pin of relay to pin 13 of the pi
* Connect relay to the AC appliance (fan)



Code:

**CODE :**

# DHT22 interfaced with Raspberry Pi  
# Displays output as temperature onto the terminal  
  
# importing libraries  
import board  
from time import sleep  
import RPI.GPIO as GPIO  
#importing the Adafruit library ( pip3 install adafruit-circuitpython-dht )  
import adafruit\_dht  
GPIO.setmode(GPIO.BOARD)  
GPIO.setwarnings(False)  
GPIO.setup(13,OUT)  
# create an instance of the sensor type  
dhtDevice = adafruit\_dht.DHT22(board.D7)  
print (‘Getting data from the sensor’)  
  
  
while True:  
 try:  
 # Print the values to the serial port  
 temperature\_c = dhtDevice.temperature  
 print("Temp:{.1f} C".format(temperature\_c))  
 if (temperature\_c > 30):  
 GPIO.output(13, 0) # Relay is active low  
 print(‘Too hot!, Fan is ON’)

else if (temperature\_c < 20):  
 GPIO.output(13, 1)  
 print(‘Too Cold!, Fan is OFF’)  
  
 except RuntimeError as error:  
 # Errors happen fairly often, DHT's are hard to read, just keep going  
 print(error.args[0])  
 time.sleep(2.0)  
 Continue  
 except Exception as error:  
 dhtDevice.exit()  
 raise error  
 time.sleep(2.0)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*