IBM ASSIGNMENT 2

Team Id:PNT2022TMID26698

Name :GAJALAKSHIMI.K

Reg No:212919205012

Build a. Python. Code,assume U get. Temperature and Humidity values. (generated with random function to variable)and write a condition to continuously detect alarm in case of high temperature

*The temp_humidity.py code uses Raspberry Pi GPIO Pin 26 by default. If you'd like to use a different pin, change the DHT_DATA_PIN variable.

DHT_DATA_PIN = 26

*Before running, we'll need to set our Adafruit IO Key and Adafruit IO Username

Code:

```
# Set to your Adafruit IO key.
# Remember, your key is a secret,
# so make sure not to publish it when you publish this code!
ADAFRUIT_IO_KEY = 'YOUR_AIO_KEY'

# Set to your Adafruit IO username.
ADAFRUIT_IO_USERNAME = 'YOUR_AIO_USERNAME'
```

*The next chunk of code creates an instance of the Adafruit IO REST client, sets up the temperature and humidity feeds, and sets up the DHT22 sensor.

Code:

```
# Create an instance of the REST client.
Aio = Client(ADAFRUIT_IO_USERNAME, ADAFRUIT_IO_KEY)

# Set up Adafruit IO Feeds.
```

Temperature_feed = aio.feeds('temperature')

Humidity_feed = aio.feeds('humidity')

Set up DHT22 Sensor.

Dht22_sensor = Adafruit_DHT.DHT22

* The while True loop, we first try to grab the humidity and sensor readings using Adafruit_DHT.read_retry which retries (up to 15 times) to get a sensor reading.

Code:

Humidity, temperature = Adafruit_DHT.read_retry(sensor, DHT_DATA_PIN)

*If the DHT sensor receives a reading, it'll print out both of the values and send them to the Adafruit IO temperature and humidity feeds.

Code:

Print('Temp={0:0.1f}*C Humidity={1:0.1f}%'.format(temperature, humidity))

Aio.send(temperature.key, temperature)

Aio.send(humidity.key, humidity)

Sometimes you won't get a sensor reading and the results will be null (because Linux can't guarantee the timing of calls to read to the sensor). If that occurs, we'll print to the terminal. Then, we sleep for DHT_READ_TIMEOUT until the next read.

In your terminal, enter the following command to run the code:

Python3 temp_humidity.py

You should now see the temperature and humidity values being sent to Adafruit IO.

Code:

Temp=25.5*C Humidity=59.1%

Temp=25.5*C Humidity=59.1%

Temp=25.5*C Humidity=59.1%

Temp=25.4*C Humidity=59.0%

Check your dashboard on Adafruit IO, and you should see the line chart update with the changes in temperature and humidity.

Program code:


```
# import standard python modules.
Import time


# import adafruit dht library.
Import Adafruit_DHT


# import Adafruit IO REST client.
From Adafruit_IO import Client, Feed


# Delay in-between sensor readings, in seconds.
DHT_READ_TIMEOUT = 5


# Pin connected to DHT22 data pin
DHT_DATA_PIN = 26


# Set to your Adafruit IO key.
# Remember, your key is a secret,
# so make sure not to publish it when you publish this code!
ADAFRUIT_IO_KEY = 'YOUR_AIO_KEY'


# Set to your Adafruit IO username.
# (go to https://accounts.adafruit.com to find your username).
```

```
ADAFRUIT_IO_USERNAME = 'YOUR_AIO_USERNAME'


# Create an instance of the REST client.

Aio = Client(ADAFRUIT_IO_USERNAME, ADAFRUIT_IO_KEY)


# Set up Adafruit IO Feeds.

Temperature_feed = aio.feeds('temperature')

Humidity_feed = aio.feeds('humidity')


# Set up DHT22 Sensor.

Dht22_sensor = Adafruit_DHT.DHT22


While True:

    Humidity, temperature = Adafruit_DHT.read_retry(dht22_sensor, DHT_DATA_PIN)

    If humidity is not None and temperature is not None:

        Print('Temp={0:0.1f}*C Humidity={1:0.1f}%'.format(temperature, humidity))

        # Send humidity and temperature feeds to Adafruit IO

        Temperature = '%.2f'%(temperature)

        Humidity = '%.2f'%(humidity)

        Aio.send(temperature_feed.key, str(temperature))

        Aio.send(humidity_feed.key, str(humidity))

    Else:

        Print('Failed to get DHT22 Reading, trying again in ', DHT_READ_TIMEOUT, 'seconds')

    # Timeout to avoid flooding Adafruit IO

    Time.sleep(DHT_READ_TIMEOUT)
```