

SMART WASTE MANAGEMENT SYSTEM

Assignment-2

Build a Python code, Assume you get temperature and humidity values and write a condition to continuously detect alarm in case of high temperature.

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)
```