

## Introduction

This report details the implementation of a sensing application using a Raspberry Pi and a temperature & humidity sensor. The application successfully reads sensor data and transmits it to a computer using the MQTT protocol. The following sections outline the setup, execution, and results of this project.

## System Setup

### Hardware Components:

- Raspberry Pi
- DHT11 Temperature & Humidity Sensor
- Power Adapter
- A computer for data reception

### Software & Libraries:

- Raspberry Pi OS
- Python
- Paho-MQTT Library
- DHT Sensor Library

### Installation Steps:

1. Set up Raspberry Pi: Ensure the OS is updated using `sudo apt update` & `sudo apt upgrade`.
2. **Install Required Libraries:**  
  

```
sudo apt install python3-pip
```

```
pip3 install paho-mqtt seeed-python-dht
```
3. Connect the Sensor: Wire the DHT11 or DHT22 sensor correctly to the Raspberry Pi GPIO pins.
4. Clone the MQTT Setup: Follow the instructions in the MQTT setup document.

## Implementation

### Reading Sensor Data and Transmitting via MQTT

The following Python script was developed to read temperature and humidity values from the DHT11 or DHT22 sensor and publish them to an MQTT broker:

```
name=dht_simpleread_mqtt.py

import time

import seeed_dht

import paho.mqtt.client as mqtt

def main():

    # Define Variables

    MQTT_HOST = "192.168.0.244"

    MQTT_PORT = 1883

    MQTT_KEEPALIVE_INTERVAL = 5

    MQTT_TOPIC = "hello/world"

    MQTT_MSG = ""

    # Define on_connect event Handler

    def on_connect(mosq, obj, rc):

        print ("Connected to MQTT Broker")

    # Define on_publish event Handler

    def on_publish(client, userdata, mid):

        print ("Message Published...")

    # Initiate MQTT Client

    mqttc = mqtt.Client()

    # Register Event Handlers

    mqttc.on_publish = on_publish

    mqttc.on_connect = on_connect

    # Connect with MQTT Broker
```

```

mqttc.connect(MQTT_HOST, MQTT_PORT, MQTT_KEEPLIVE_INTERVAL)
# for DHT11/DHT22
sensor = seeed_dht.DHT("11", 12)
# for DHT10
# sensor = seeed_dht.DHT("10")
while True:
    humi, temp = sensor.read()
    if not humi is None:
        print('DHT{0}, humidity {1:.1f}%, temperature {2:.1f}*'.format(sensor.dht_type, humi,
temp))
        MQTT_MSG = 'DHT{0}, humidity {1:.1f}%, temperature
{2:.1f}*'.format(sensor.dht_type, humi, temp)
        mqttc.publish(MQTT_TOPIC, MQTT_MSG)
    else:
        print('DHT{0}, humidity & temperature: {1}'.format(sensor.dht_type, temp))
    time.sleep(1)
if __name__ == '__main__':
    main()

```

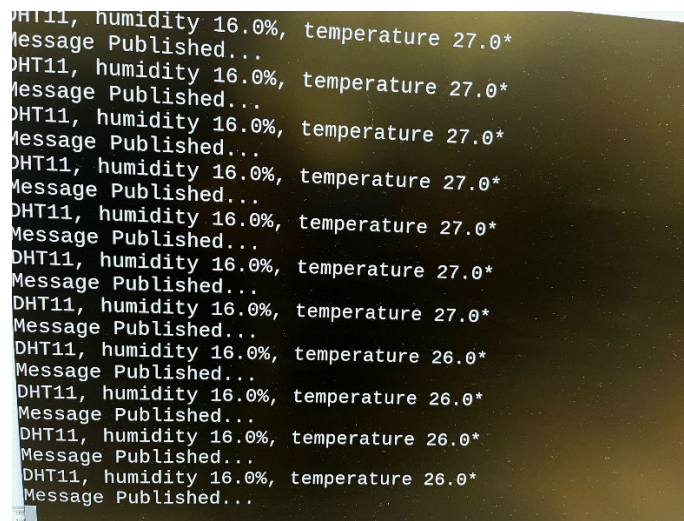
## Testing & Results

### Testing Procedure:

1. Executed the script to read sensor values and confirmed accurate readings.
2. Ran the MQTT publisher script to send data to the broker.
3. Verified data reception on the computer using an MQTT subscriber.

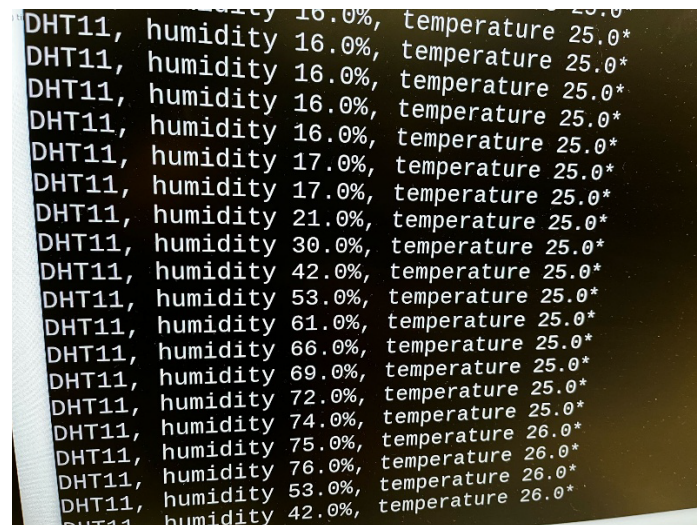
### Screenshots:

Sensor data output on Raspberry Pi terminal.



A screenshot of a Raspberry Pi terminal window displaying the output of a script. The output consists of multiple lines of text, each representing a sensor reading. The format is 'DHT11, humidity 16.0%, temperature 27.0\*' followed by 'Message Published...'. The temperature values are consistently 27.0\* for the first 10 lines and then drop to 26.0\* for the remaining lines. The humidity values are consistently 16.0%.

```
DHT11, humidity 16.0%, temperature 27.0*  
Message Published...  
DHT11, humidity 16.0%, temperature 27.0*  
Message Published...  
DHT11, humidity 16.0%, temperature 27.0*  
Message Published...  
DHT11, humidity 16.0%, temperature 27.0*  
Message Published...  
DHT11, humidity 16.0%, temperature 27.0*  
Message Published...  
DHT11, humidity 16.0%, temperature 27.0*  
Message Published...  
DHT11, humidity 16.0%, temperature 27.0*  
Message Published...  
DHT11, humidity 16.0%, temperature 26.0*  
Message Published...  
DHT11, humidity 16.0%, temperature 26.0*  
Message Published...  
DHT11, humidity 16.0%, temperature 26.0*  
Message Published...  
DHT11, humidity 16.0%, temperature 26.0*  
Message Published...
```



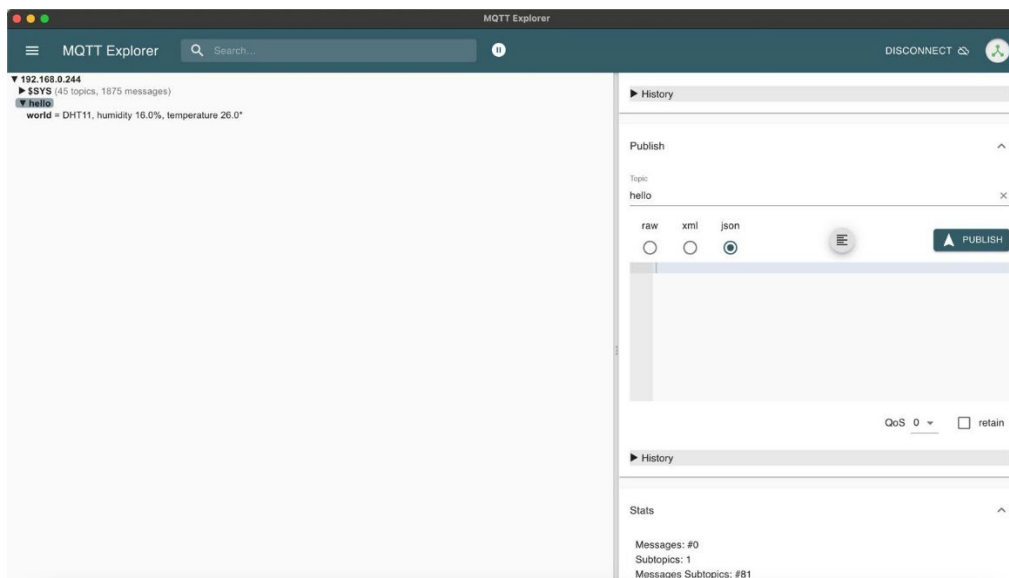
A screenshot of a Raspberry Pi terminal window displaying the output of a script. The output consists of multiple lines of text, each representing a sensor reading. The format is 'DHT11, humidity [value]%, temperature [value]\*'. The humidity values start at 16.0% and increase in increments of 3.0% up to 75.0%, then drop to 53.0% and 42.0%. The temperature values are consistently 25.0\* for the first 15 lines and then drop to 26.0\* for the remaining lines.

```
DHT11, humidity 16.0%, temperature 25.0*  
DHT11, humidity 16.0%, temperature 25.0*  
DHT11, humidity 16.0%, temperature 25.0*  
DHT11, humidity 16.0%, temperature 25.0*  
DHT11, humidity 17.0%, temperature 25.0*  
DHT11, humidity 17.0%, temperature 25.0*  
DHT11, humidity 21.0%, temperature 25.0*  
DHT11, humidity 30.0%, temperature 25.0*  
DHT11, humidity 42.0%, temperature 25.0*  
DHT11, humidity 53.0%, temperature 25.0*  
DHT11, humidity 61.0%, temperature 25.0*  
DHT11, humidity 66.0%, temperature 25.0*  
DHT11, humidity 69.0%, temperature 25.0*  
DHT11, humidity 72.0%, temperature 25.0*  
DHT11, humidity 74.0%, temperature 25.0*  
DHT11, humidity 75.0%, temperature 26.0*  
DHT11, humidity 76.0%, temperature 26.0*  
DHT11, humidity 53.0%, temperature 26.0*  
DHT11, humidity 42.0%, temperature 26.0*
```

```
rpi@raspberrypi:~/Seeed_Python_DHT/examples$ ls
dht_simpleread_mqtt.py dht_simpleread.py publisher.py
rpi@raspberrypi:~/Seeed_Python_DHT/examples$ python d
DHT11, humidity 21.0%, temperature 26.0*
DHT11, humidity 16.0%, temperature 25.0*
DHT11, humidity 16.0%, temperature 25.0*
DHT11, humidity 16.0%, temperature 25.0*
DHT11, humidity 16.0%, temperature 25.0*
DHT11, humidity 16.0%, temperature 25.0*
DHT11, humidity 16.0%, temperature 25.0*
DHT11, humidity 16.0%, temperature 25.0*
DHT11, humidity 16.0%, temperature 25.0*
DHT11, humidity 16.0%, temperature 25.0*
```

```
rpi@raspberrypi:~$ cd ./Seeed_Python_DHT/example
rpi@raspberrypi:~/Seeed_Python_DHT/examples$ ls
dht_simpleread_mqtt.py dht_simpleread.py publis
rpi@raspberrypi:~/Seeed_Python_DHT/examples$ pyt
DHT11, humidity 21.0%, temperature 26.0*
DHT11, humidity 16.0%, temperature 25.0*
DHT11, humidity 16.0%, temperature 25.0*
DHT11, humidity 16.0%, temperature 25.0*
DHT11, humidity 16.0%, temperature 25.0*
DHT11, humidity 16.0%, temperature 25.0*
```

**MQTT subscriber output showing received data.**



**Conclusion:**

The sensing application successfully reads and delivers sensor data using the MQTT protocol. This device displays successful IoT-based communication; further developments might incorporate cloud connectivity and real-time monitoring dashboards.