

IFN 649

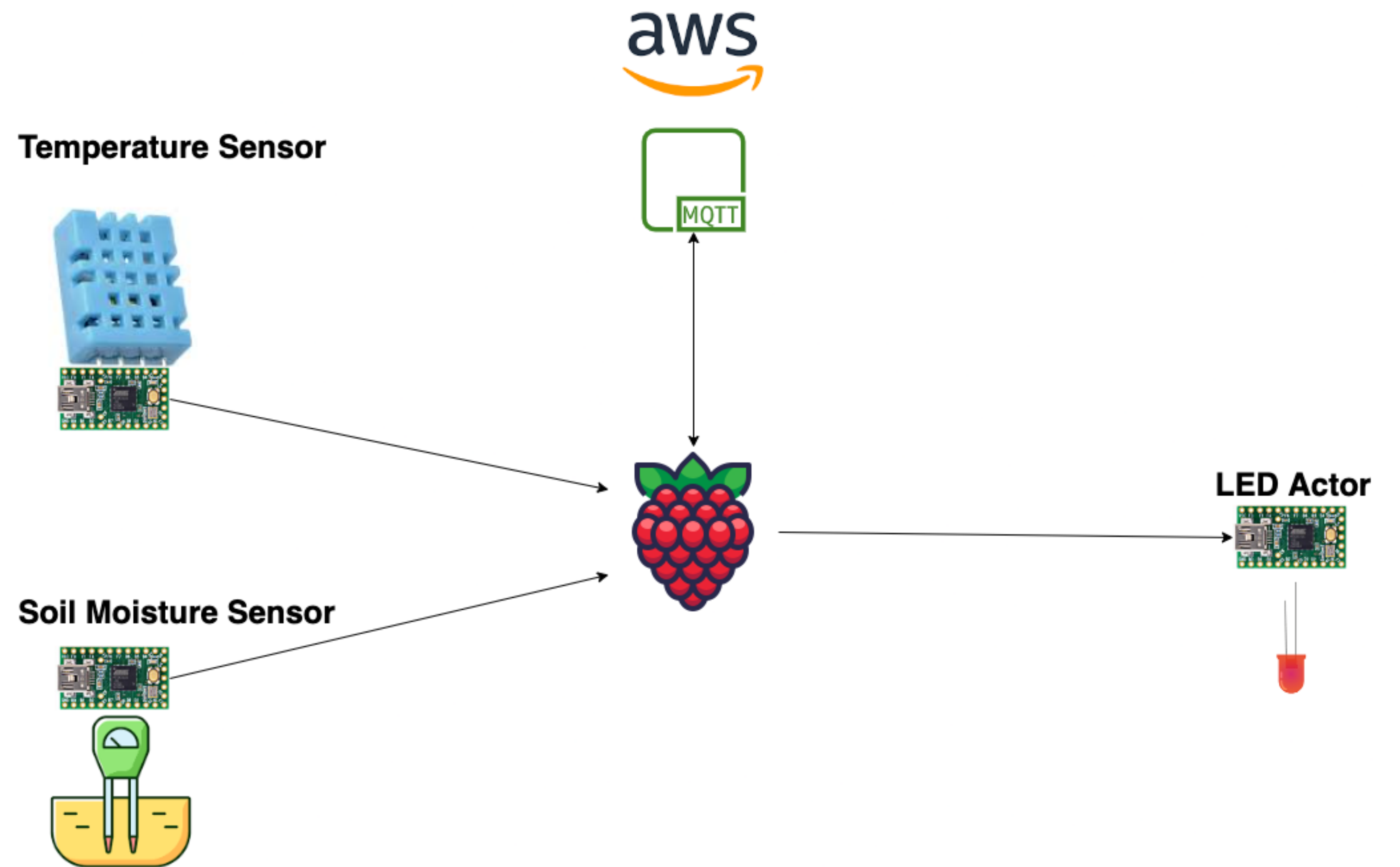
Assessment One: Teensy, Raspberry pi and AWS Demonstration

Github Repository: https://github.com/krieem/IFN649_AssessmentOne

Abdulkarim Alzahrani
N10560211

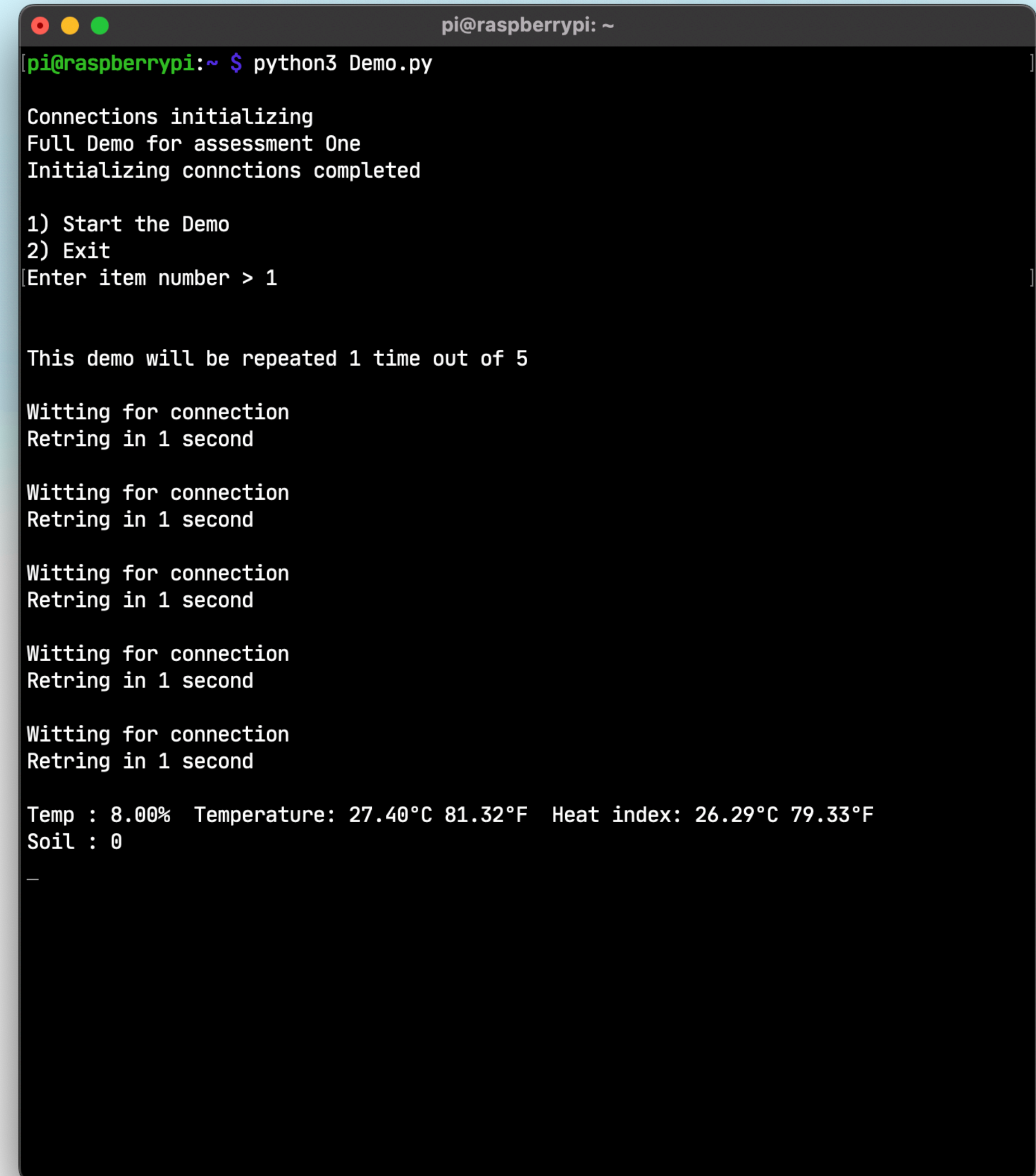
Overview

- MQTT
 - Server: AWS
 - Publisher: Raspberrypi
 - Subscriber: Raspberrypi
- Sensors
 - Temperature: DHT Teensy
 - Soil: Soil Moisture Sensor
- Actors
 - Green LED: Teensy
 - Orange LED: Teensy



Data Collection

1. Raspberrypi will initialise the bluetooth connection to both Teensy sensors.
2. The connection will be established and serial data will be received.
3. The received data will be cleaned and processed.
4. Information will be displayed on the terminal screen

A terminal window titled 'pi@raspberrypi: ~' showing the execution of a Python script 'Demo.py'. The script initializes connections, prompts for a demo item number (1), and displays sensor data: Temperature (27.40°C / 81.32°F), Heat index (26.29°C / 79.33°F), and Soil (0).

```
pi@raspberrypi: ~  
[pi@raspberrypi:~ $ python3 Demo.py  
  
Connections initializing  
Full Demo for assessment One  
Initializing connctions completed  
  
1) Start the Demo  
2) Exit  
Enter item number > 1  
  
This demo will be repeated 1 time out of 5  
  
Witting for connection  
Retring in 1 second  
  
Witting for connection  
Retring in 1 second  
  
Witting for connection  
Retring in 1 second  
  
Witting for connection  
Retring in 1 second  
  
Witting for connection  
Retring in 1 second  
  
Temp : 8.00%  Temperature: 27.40°C 81.32°F  Heat index: 26.29°C 79.33°F  
Soil : 0  
—
```


MQTT

Publisher (Demo.py)



Subscriber (aws.py)

1. While the Raspberrypi receives the data a connection to AWS instance will be established.
2. MQTT connection as a publisher will be created
3. The MQTT publisher will publish the data to three topics.

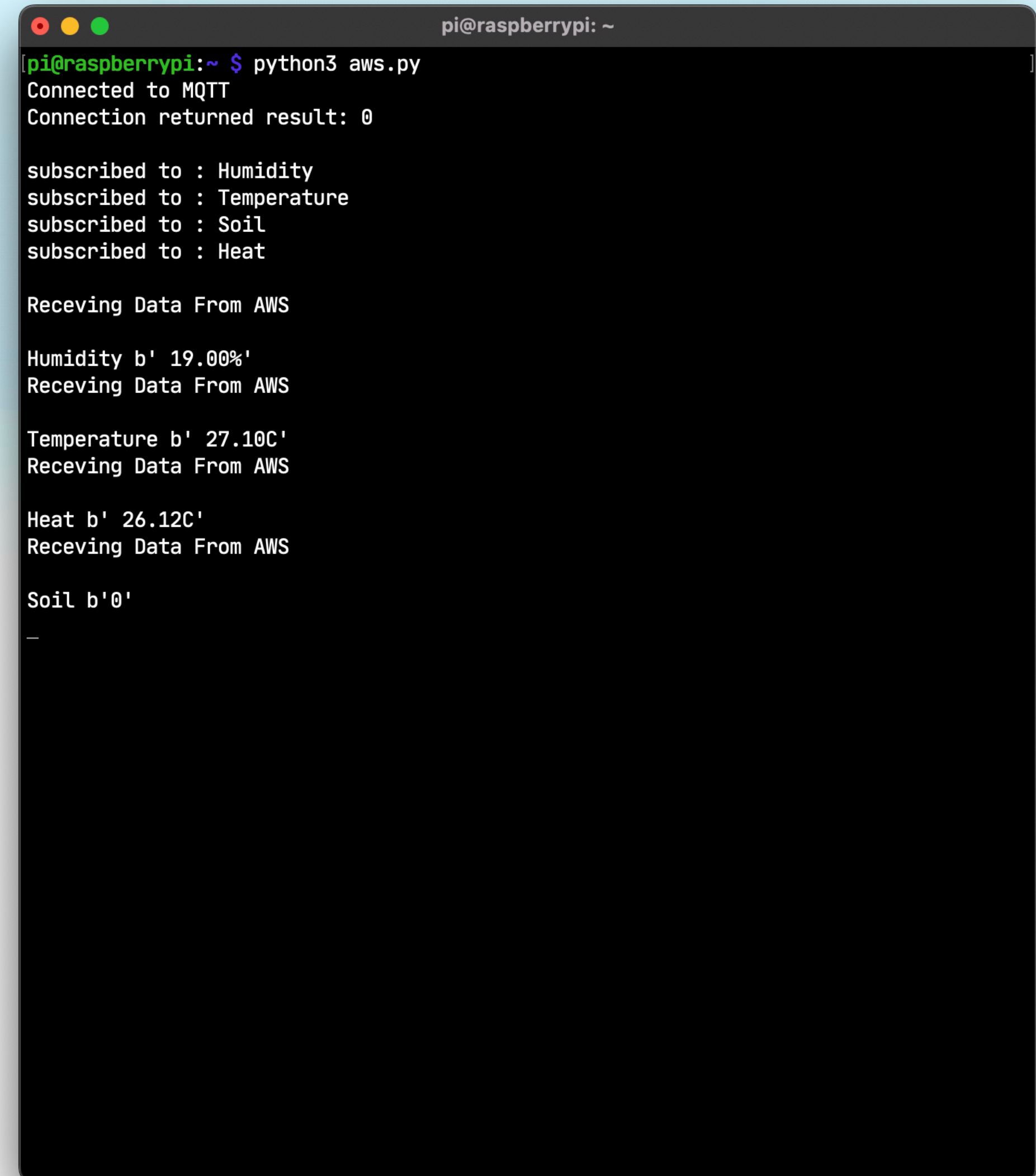
1. MQTT subscriber should be deployed to receive data.
2. The subscriber will subscribe to the topics defined on the publisher.
3. The subscriber will listen to the server and wait for messages.

MQTT Topics

Heat & Temperature & Humidity & Soil

Actor

1. When the subscriber receives the data the information will be displayed on the terminal screen.
2. LED indications on the Teensy actor
 1. Orange LED will be emitted indicating waiting from data.
 2. Green LED will be emitted during the data transmission.

A terminal window titled 'pi@raspberrypi: ~' showing the execution of a Python script 'aws.py'. The script successfully connects to an MQTT broker and subscribes to four topics: Humidity, Temperature, Soil, and Heat. It then receives and displays data from AWS for each topic. The data received is: Humidity b' 19.00%', Temperature b' 27.10C', Heat b' 26.12C', and Soil b'0'.

```
pi@raspberrypi: ~  
[pi@raspberrypi:~ $ python3 aws.py  
Connected to MQTT  
Connection returned result: 0  
  
subscribed to : Humidity  
subscribed to : Temperature  
subscribed to : Soil  
subscribed to : Heat  
  
Receiving Data From AWS  
  
Humidity b' 19.00%'  
Receiving Data From AWS  
  
Temperature b' 27.10C'  
Receiving Data From AWS  
  
Heat b' 26.12C'  
Receiving Data From AWS  
  
Soil b'0'  
_
```