**Project Report: Connecting Sensors to ESP32 for MQTT Data to Grafana via Docker on a Linux Server**

**Author**: Dr. Mudasar Latif Memon  
**Date**: 1st September 2023

**Objective**

The objective of this project is to establish an IoT system utilizing an ESP32 microcontroller to collect temperature, humidity, and distance data from sensors. This data is then transmitted to a Grafana dashboard via MQTT protocol. The project also involves the use of Docker for efficient container management on a Linux server. The data transmission is facilitated through a local hotspot.

**Requirements**

* ESP32 microcontroller
* Temperature and humidity sensor (e.g., DHT22)
* Ultrasonic sensor (e.g., HC-SR04)
* MQTT broker (e.g., Mosquitto)
* Grafana installation
* Docker installed on the Linux server
* Linux operating system
* Local hotspot for data transmission

**Procedure**

**Hardware Setup**

* Connect the temperature and humidity sensor (DHT22) to the ESP32 as per the datasheet instructions.
* Connect the ultrasonic sensor (HC-SR04) to the ESP32, typically requiring two GPIO pins for trigger and echo.
* Ensure proper power and ground connections for both sensors and the ESP32.

**Video Guide**: [Hardware Connectivity Video](https://drive.google.com/drive/u/1/folders/1nxSVwyzRIeo5hr2NL-CwRgtj9hdhhxNL)

**Programming for ESP-32**

pythonCopy code

# ESP32 MicroPython example code (use uPyCraft or Thonny IDE) # ... (existing code)

**MQTT Broker Setup**

* Install Mosquitto MQTT broker on your Linux server.
* Configure Mosquitto to listen on a specific port (e.g., 1883).
* Create MQTT topics for temperature, humidity, and distance data.

**Docker Containerization**

1. **MQTT Broker container**
   * Install and configure Mosquitto MQTT broker inside the container.
2. **Grafana container**
   * Install Grafana inside the container and configure it to connect to the MQTT broker and InfluxDB database.
3. **InfluxDB container**
   * Create a container with InfluxDB to store the data from MQTT.
4. **ESP32 Data Relay container**
   * Run a Python or Node.js script inside this container to receive MQTT data from the ESP32 and push it into the InfluxDB container.

**Docker Compose**

* Create a Docker Compose YAML file to define the four containers (MQTT Broker, Grafana, InfluxDB, ESP32 Data Relay).
* Specify the container configurations, dependencies, and environment variables in the YAML file.

**Container Deployment**

* Deploy the containers on the Linux server using Docker Compose: **docker-compose up -d**.
* Verify that all containers are running without errors: **docker-compose ps**.

**Local Hotspot Setup**

* Create a local hotspot on your Linux server or use an existing one to establish a Wi-Fi connection with the ESP32.
* Ensure the ESP32 is configured to connect to the hotspot.

**Testing and Monitoring**

* Confirm that the ESP32 is publishing data to the MQTT broker via the local hotspot.
* Verify that data is being stored in the InfluxDB database.
* Access the Grafana dashboard through a web browser to visualize the sensor data.

**Insights**

* Docker provides an efficient and scalable solution for managing containers, ensuring smooth deployment and operation of the IoT system.
* Mosquitto MQTT broker facilitates seamless communication between the ESP32 and the Grafana dashboard.
* InfluxDB serves as a reliable database for storing and retrieving sensor data.

**Conclusion**

The project has successfully established an IoT system, integrating sensors with an ESP32 microcontroller. The data collected (temperature, humidity, and distance) is transmitted to a Grafana dashboard via MQTT protocol. Docker containerization enhances the system's scalability and manageability. The utilization of a local hotspot for data transmission ensures remote monitoring and visualization on a Linux server. This comprehensive setup lays a solid foundation for further IoT applications and data-driven insights.

For more details, please visit the complete website: [mlmemon.com/14-data-science](https://www.mlmemon.com/14-data-science).

1. **Setting Up an MQTT-Based IoT Data Pipeline with Docker Containers**

**Bullet Points:**

* **Install and Configure Mosquitto MQTT Broker:**
  + Create a Docker container for the Mosquitto MQTT broker.
  + Configure MQTT broker settings and ports within the container.
* **Install Grafana and Connect to MQTT Broker:**
  + Create a Docker container for Grafana.
  + Configure Grafana to connect to the MQTT broker.
  + Set up Grafana's data source to communicate with InfluxDB.
* **Create a Container for InfluxDB:**
  + Set up a Docker container for InfluxDB to store MQTT data.
  + Define InfluxDB configurations and expose necessary ports.
* **Implement MQTT Data Relay Script:**
  + Create a Python or Node.js script to receive MQTT data from ESP32.
  + Configure the script to push the data into the InfluxDB container.
* **Define Docker Compose YAML File:**
  + Create a Docker Compose YAML file to orchestrate the containers.
  + Specify container configurations, dependencies, and environment variables.
* **Deploy Containers with Docker Compose:**
  + Run the containers using Docker Compose with the **docker-compose up -d** command.
  + Ensure all containers start without errors.
* **Verify Container Status:**
  + Confirm the status of all containers using **docker-compose ps**.