National College of Ireland

Configuration Manual

MSc Research Project
MSc in Cyber Security

Shaikh Sharuk Shaikh Babu
X23225262

School of Computing
National College of Ireland

Supervisor: Dr Raza Ul Mustafa

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Shaikh Sharuk Shaikh Babu		
Student ID:	X23225262		
Programme:	MSc in Cyber Security	**Year:**	2024
Module:	MSc Research Project		
Supervisor:	Dr Raza Ul Mustafa		
Submission Due Date:	12-12-2024		
Project Title:	Configuration Manual		
Word Count:	675	**Page Count:**	5

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Shaikh Sharuk Shaikh Babu

Date: 12/12/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Shaikh Sharuk Shaikh Babu
X23225262

Table of Contents

---

1. Introduction

This guide is a detailed procedure of forming an encrypted smart data home system utilizing AES 128 security and honey words. In the scope of the system, a prototype of the IoT devices such as sensors is simulated through Wokwi hardware emulation and Node-RED for data analysis and control. The results found out that encrypted data is conveyed securely over MQTT and also safeguarded against data interception.

---

2. System Configurations:

Using System:

- **MacBook Pro M1 Chip**
  - 8 GB RAM
  - 256 GB SSD
  - macOS 14.4.1

Virtual Hardware:

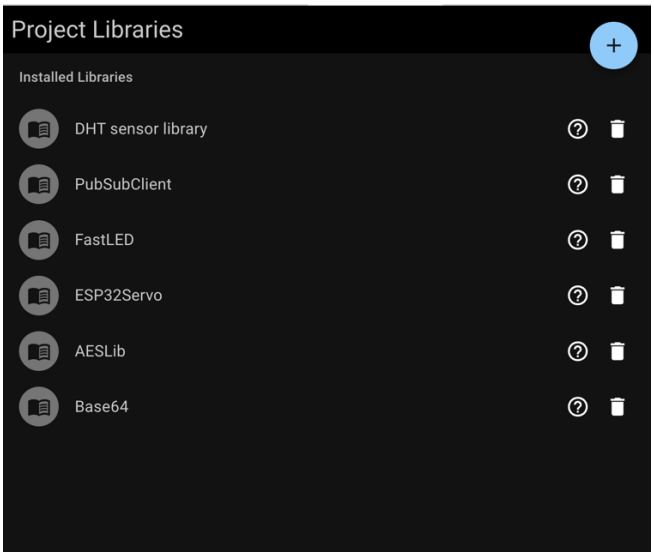
- Virtual ESP32 microcontroller (simulated in Wokwi)
- DHT22 Temperature and Humidity Sensor
- NeoPixel LED Strip (16 LEDs)
- Servo Motor
- Wi-Fi connectivity (Simulated Private Gateway in Wokwi)

Software:

- Wokwi Online Simulator
- Node-RED (running in Docker)
- MQTT Broker (HiveMQ)

Libraries (Installed in Wokwi):

- Adafruit_Sensor
- DHT_U
- WiFi
- PubSubClient
- ESP32Servo
- FastLED
- AESLib
- Base64



---

3. Virtual Hardware Setup (Wokwi)

1. **Create a Wokwi Project**:
   - Go to Wokwi website and create a new project.
   - Add an ESP32 microcontroller to the canvas.
2. **Add Virtual Peripherals**:
   - **DHT22 Sensor**: Connect to GPIO 12.
   - **Servo Motor**: Connect signal pin to GPIO 2.
   - **NeoPixel LED Strip**: Connect data pin to GPIO 4.
   - Use Wokwi's simulation environment to wire components virtually.
3. **Power and Ground Connections**:
   - Ensure all virtual peripherals share a common ground.
4. **Upload Code**:
   - Paste the provided source code into the Wokwi code editor and start the simulation.

---

4. Software Installation

Step 1: Use Wokwi's In-Built Arduino Environment

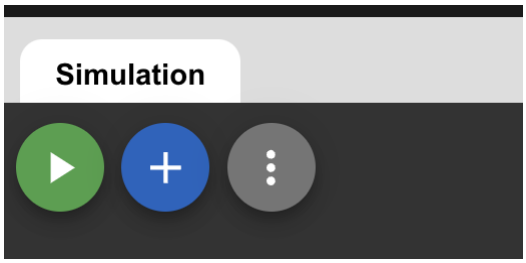
- Wokwi provides a built-in code editor and simulation environment for writing and testing Arduino code.
- No need to download and install a standalone Arduino IDE.
- Access the editor by creating a new project in Wokwi and pasting your code directly into the provided editor window.

Step 2: Configure Libraries in Wokwi

- Ensure the required libraries are included in your Wokwi project by adding them to the "Code Libraries" section:
  - Adafruit Unified Sensor
  - DHT sensor library
  - PubSubClient
  - ESP32Servo
  - FastLED
  - AESLib
  - Base64
- Wokwi will automatically handle dependencies and simulate the virtual hardware environment.

Step 3: Start Simulation

- Click on the **"Start Simulation"** button in Wokwi to run your project.



- Monitor the Serial Output within the simulation to check for data transmission and encryption logs.

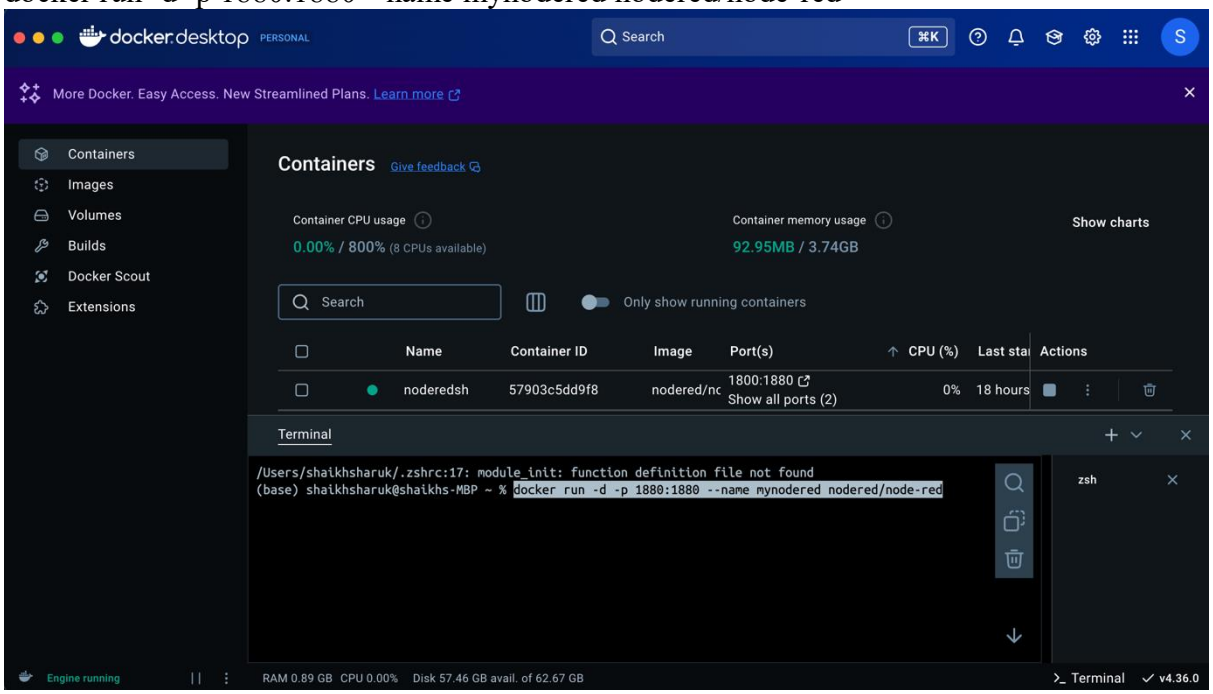
---

5. Node-RED Configuration

Step 1: Install Node-RED

- If using Docker:

```
docker run -d -p 1880:1880 --name mynodered nodered/node-red
```

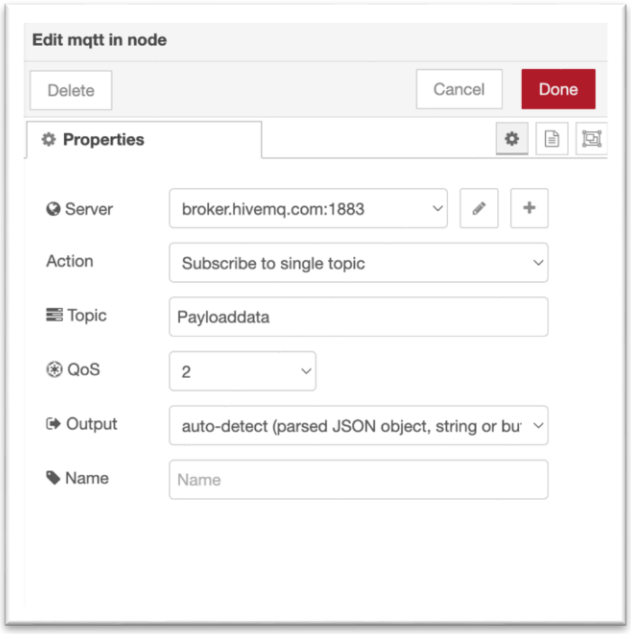


- Alternatively, install Node-RED directly on your system using npm:

```
npm install -g --unsafe-perm node-red
```

Step 2: Configure Node-RED MQTT Input

- Drag an **MQTT In** node into the flow.
- Double-click the node and configure it as follows:
  - Server: broker.hivemq.com
  - Port: 1883
  - Topic: Payload data



Step 3: Visualise without Encrypted Data

- Drag **Chart** and **Gauge** nodes into the flow (named it as Temperature chart and Humidity chart).
- Connect the MQTT In node directly to these nodes.

Important:
The system currently displays encrypted data using Wireshark packet tracer that verifying the payload data is safe and secure with the help of encrypting AES 128 and honey words. due to the limitation of not implementing decryption key in Node-RED was unable to display visualise the payload data. This design is intentional to highlight the importance of encryption for transmission.

---

6. Testing and Validation

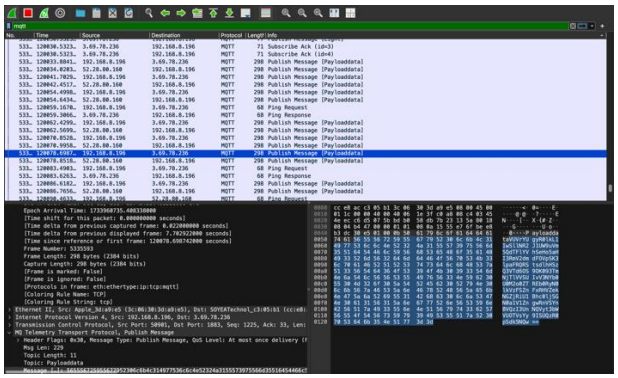
Step 1: Verify Wi-Fi and MQTT Connection

- Check the Serial Monitor for successful Wi-Fi connection and MQTT subscription.

```
WiFi connected!
IP address:
10.13.37.2
MQTT connected
Subscribed to topics
Message received: 44
Data successfully encrypted!
```

Step 2: Validate Encrypted Transmission

- Use Wireshark to capture network traffic and verify that the transmitted data is encrypted.



Step 3: Honeyword Validation

- Confirm that the payload includes both real data and decoy values by analysing the raw encrypted data in the MQTT broker.

533…	120062.4299…	192.168.8.196	3.69.78.236	MQTT	298	Publish Message [Payloaddata]
533…	120062.5699…	52.28.80.160	192.168.8.196	MQTT	298	Publish Message [Payloaddata]
533…	120070.8528…	192.168.8.196	3.69.78.236	MQTT	298	Publish Message [Payloaddata]
533…	120070.9958…	52.28.80.160	192.168.8.196	MQTT	298	Publish Message [Payloaddata]
533…	120078.6987…	192.168.8.196	3.69.78.236	MQTT	298	Publish Message [Payloaddata]
533…	120078.8518…	52.28.80.160	192.168.8.196	MQTT	298	Publish Message [Payloaddata]

---

7. Troubleshooting

Common Issues and Solutions:

1. **Wi-Fi Not Connecting**:
   - Ensure SSID and password are correct.
   - Check Wi-Fi signal strength.
   - Enable Private Gateway
2. **MQTT Not Publishing**:
   - Verify MQTT broker details (Server: broker.hivemq.com and Port: 1883)