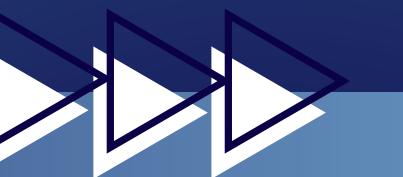




PROGRESS REPORT PRESENTATION

IOT-BASED SMART TUNNEL MONITORING SYSTEM



Student A : Ahmad Syalimin bin Mohd Amin (Student A)
Student B : Muhammad Shafwan bin Anis (Student B)
Student C : Low Ying Chian (Student C)



Background

- Tunnels in urban areas are vulnerable to flooding and hazardous conditions during heavy rainfall.
- Sudden water rise in tunnels can cause accidents, vehicle damage, and traffic disruption.
- Conventional tunnel monitoring systems often rely on manual observation or delayed response.
- Sensor-based automation enables real-time monitoring and faster safety actions.
- This project introduces a Smart Tunnel System using sensors and an automatic gate mechanism.

Problem Statement

- Tunnel conditions can change rapidly due to rainfall and rising water levels.
- Manual monitoring may not respond fast enough during emergency situations.
- Vehicles may still enter tunnels during unsafe conditions, increasing accident risk.
- There is a need for an automated tunnel safety system that can detect danger and respond immediately.

Objective

- To monitor tunnel environmental conditions using multiple sensors.
- To determine SAFE or DANGER status based on predefined thresholds.
- To control a servo motor acting as an automatic tunnel gate.
- To provide visual safety indication using red and green LEDs.

Hardware & Sensor Integration



STUDENT A

Ahmad Syalimin bin Mohamad Amin

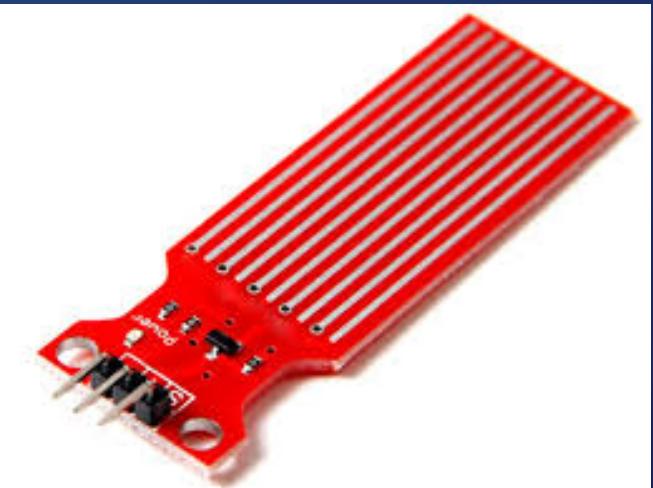
231061451



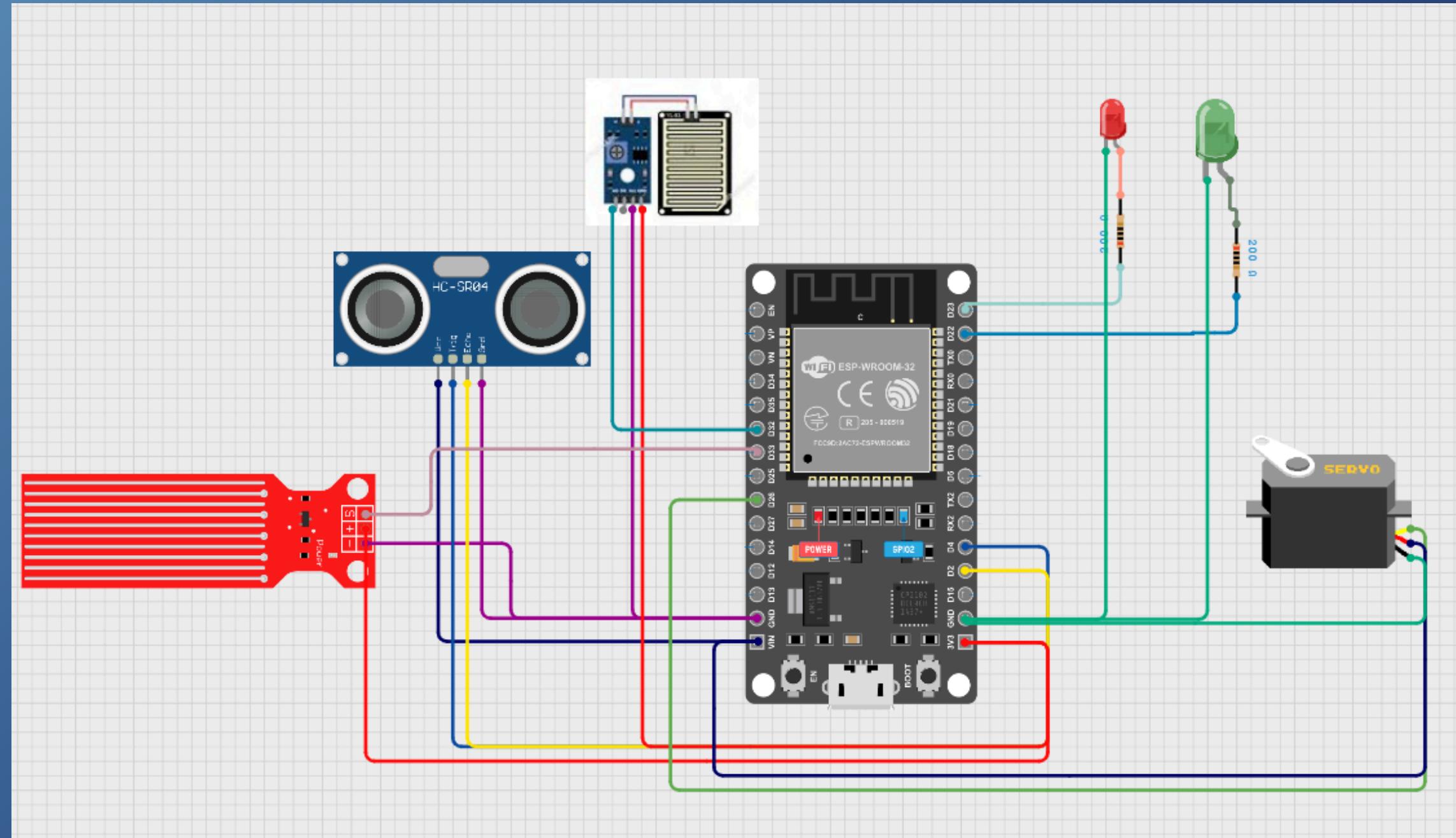
Component



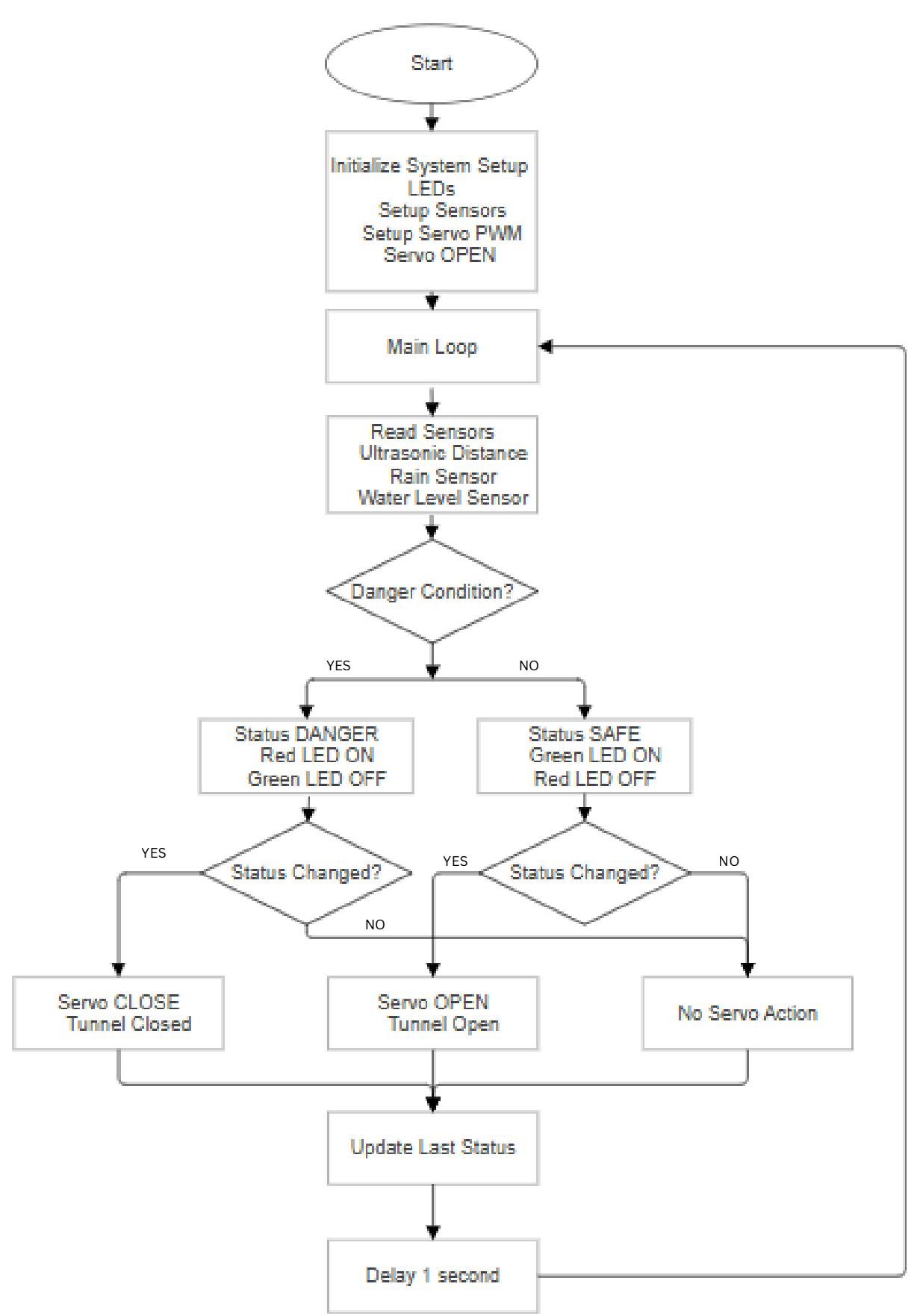
Component	Function
Ultrasonic sensor	Dam Monitor (Early Warning). Measures water level at the dam. If the dam is too full, it alerts the tunnel of flood risk.
Water level sensor	Tunnel Flood Detector. Detects if water is actually flowing inside the tunnel right now.
Rain sensor	Rain Detector. Detects if it is raining heavily outside the tunnel.
Esp 32	Main control. It reads all sensors and decides if the tunnel is Safe or Dangerous.
Red and green LED	Red Led as STOP Signal Green Led as SAFE Signal .
Servo motor	Automatic tunnel gate actuator



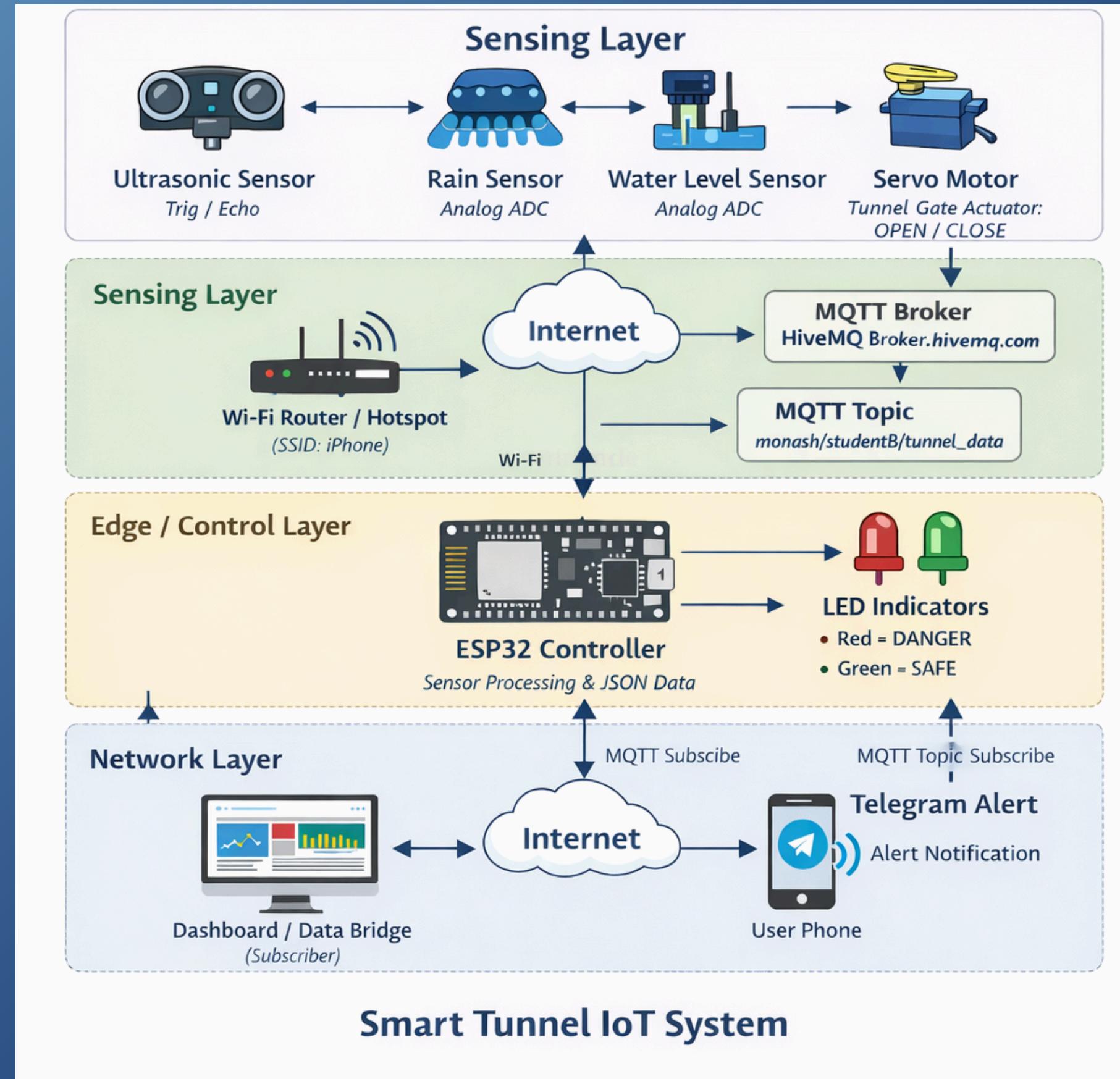
Circuit Diagram



Flow process



Block Diagram



Problems faced

Problem:

Sensor readings fluctuate and are not consistent during operation.

Solution:

Use Vin for the ultrasonic sensor and 3v3 for other sensor

Problem:

Analog sensor values are affected by noise, causing unstable ADC readings.

Solution:

Reduce wire length between sensors and ESP32

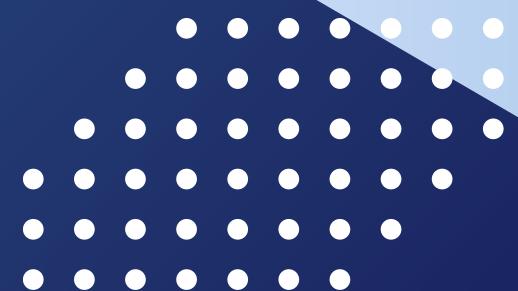
IoT Communication & Dashboard



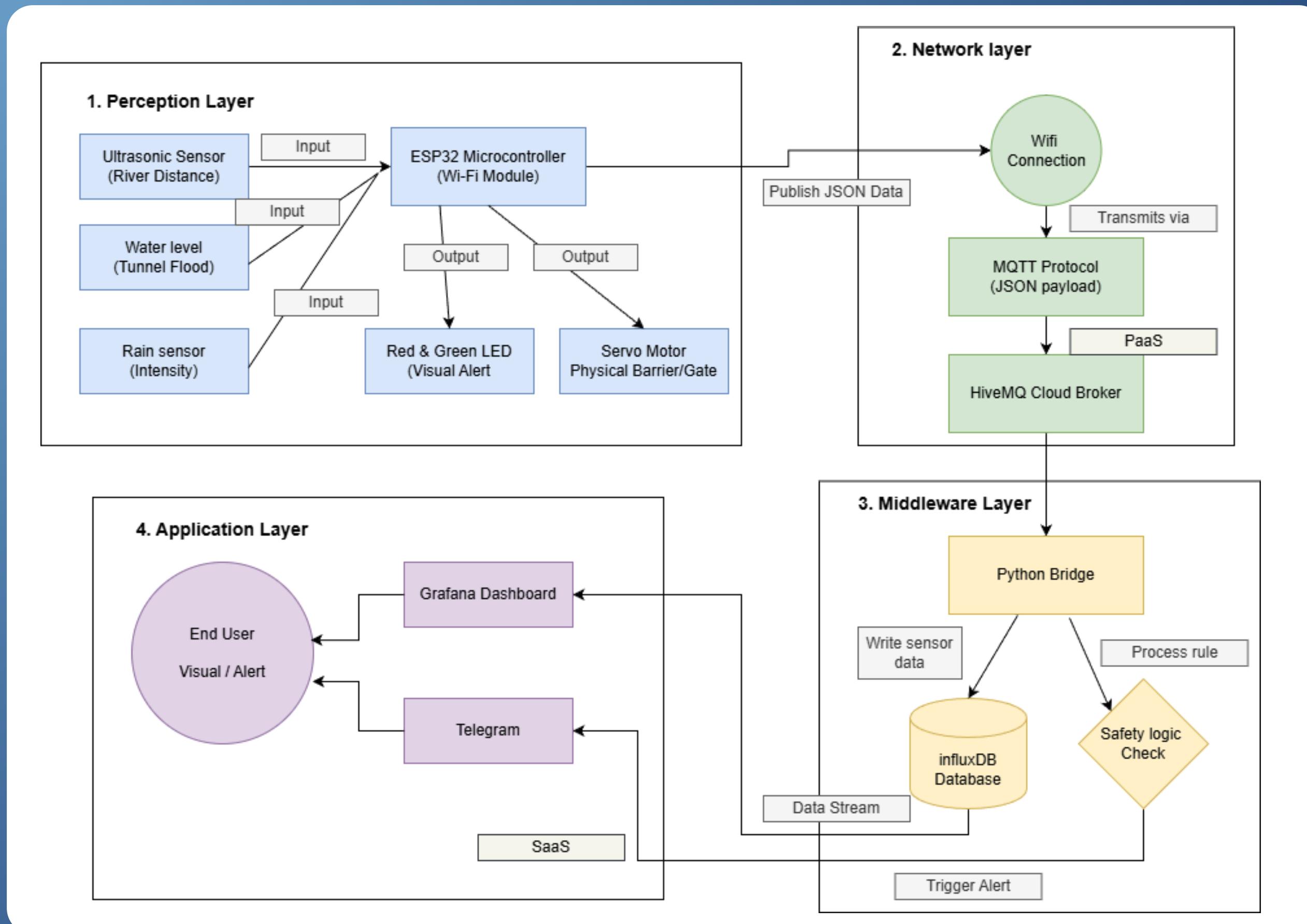
STUDENT B

**Muhammad
Shafwan bin Anis**

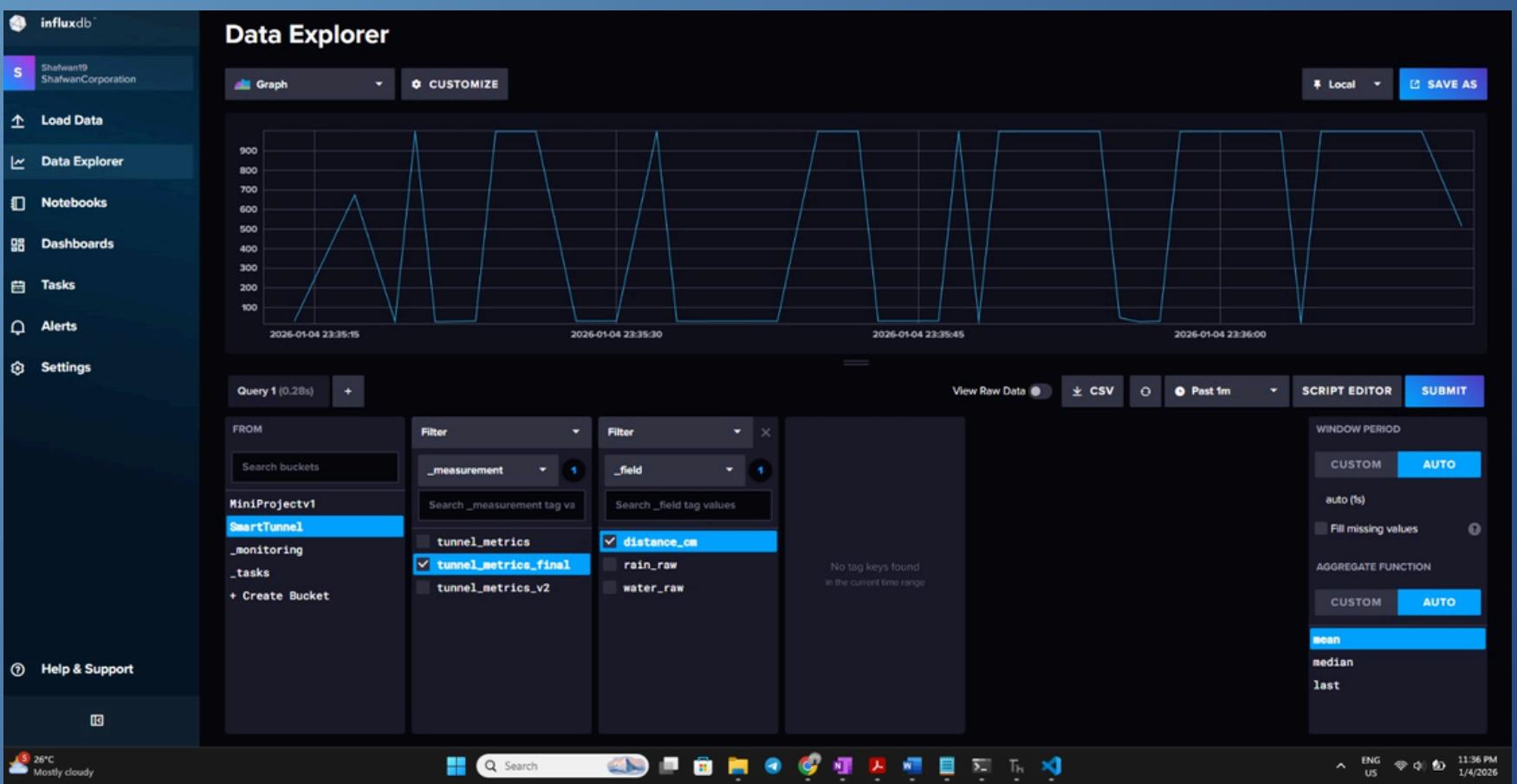
231061512



IOT Architecture



INFLUXDB INTERFACE



GRAFANA INTERFACE

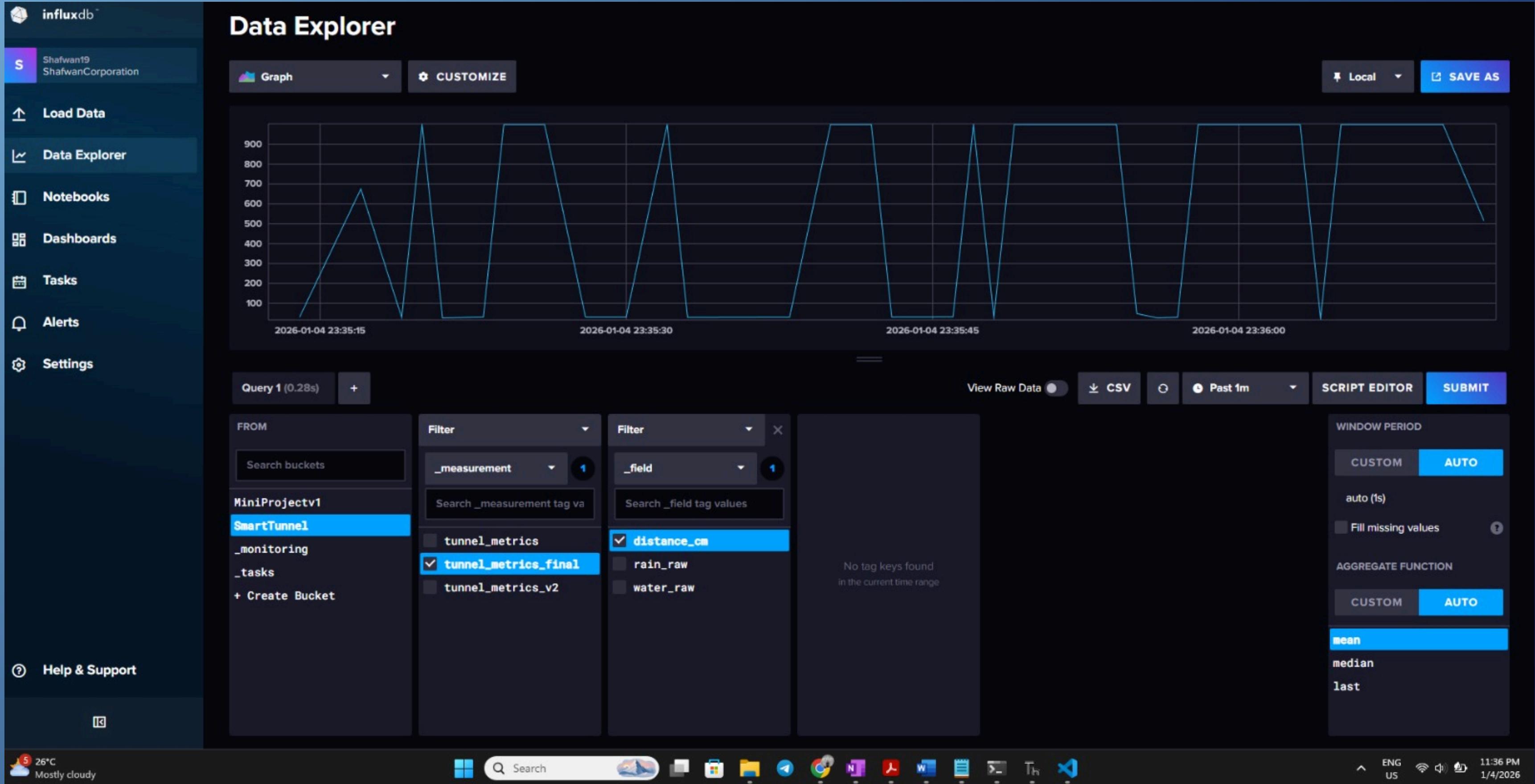




This screenshot shows a Smart Tunnel dashboard interface. The top bar includes navigation links like 'Home', 'Dashboards', 'Smart Tunnel', and 'Smart Tunnel' again, along with search, settings, and save buttons. A status bar at the bottom shows system icons and the date/time.

The main area features several cards:

- Safety Indicator:** Displays the word "SAFE" in large green letters.
- Clock:** Shows the date "2026-01-17" and time "11:04:20 PM".
- Water Level (Ultrasonic):** Shows a value of "17.08 cm" with a change of "+ 0.101%" and a green upward arrow.
- Water Level:** A circular gauge showing "269 unit" with a green segment.
- Rain Sensor:** Shows a yellow sun icon and the word "Dry".
- Distance:** A line chart showing distance over time, with the y-axis ranging from 0.00 cm to 1000.00 cm and the x-axis from 22:56:00 to 23:00:00. A green line represents the "distance_cm" metric.
- Water Level:** A bar chart showing water level over time, with the y-axis ranging from 0.00 unit to 2000.00 unit and the x-axis from 23:00:00 to 23:00:01. A blue bar represents the "water_level" metric.
- Rain Sensor:** A line chart showing rain sensor data over time, with the y-axis ranging from 0 to 8000 and the x-axis from 22:56:00 to 23:00:00. A blue line represents the "rain_rain" metric.



Data Analytics

STUDENT C

Low Ying Chian

231062387



Data Analytics

Analytics Approach

- Descriptive: Monthly water level patterns
- Diagnostic: Rain–water level relationship
- Predictive: Flood probability by month
- Prescriptive: Risk-based operational actions

Key Findings

- Water level peaks during monsoon season (Nov–Dec)
- Strong positive correlation between rainfall and water level
- Flood probability exceeds 70% during peak months



Risk Classification Logic

- Safe: Normal operation
- Alert: Increased monitoring
- Danger: Tunnel closure & pump activation

Impact & SDG 11 Contribution

- Early flood warning
- Improved tunnel safety
- Enhanced urban resilience

Thank You



Backup Slide

Experiment 1 : Calculate latency

```
>> [TIMING] Sensor Recv: 01:18:23.071  
>> [TIMING] Telegram Sent: 01:18:24.544  
>> TOTAL LATENCY: 1.4726 seconds
```

```
>> [TIMING] Sensor Recv: 01:19:03.646  
>> [TIMING] Telegram Sent: 01:19:04.661  
>> TOTAL LATENCY: 1.0146 seconds
```

```
>> [TIMING] Sensor Recv: 01:19:22.690  
>> [TIMING] Telegram Sent: 01:19:23.770  
>> TOTAL LATENCY: 1.0803 seconds
```

```
>> [TIMING] Sensor Recv: 01:19:32.703  
>> [TIMING] Telegram Sent: 01:19:34.680  
>> TOTAL LATENCY: 1.3765 seconds
```

```
>> [TIMING] Sensor Recv: 01:20:03.850  
>> [TIMING] Telegram Sent: 01:20:05.030  
>> TOTAL LATENCY: 1.1803 seconds
```

Average latency : 1.2248 s

Experiment 2 : Calculate latency Variance from distance

Scenario 1: Besides the laptop

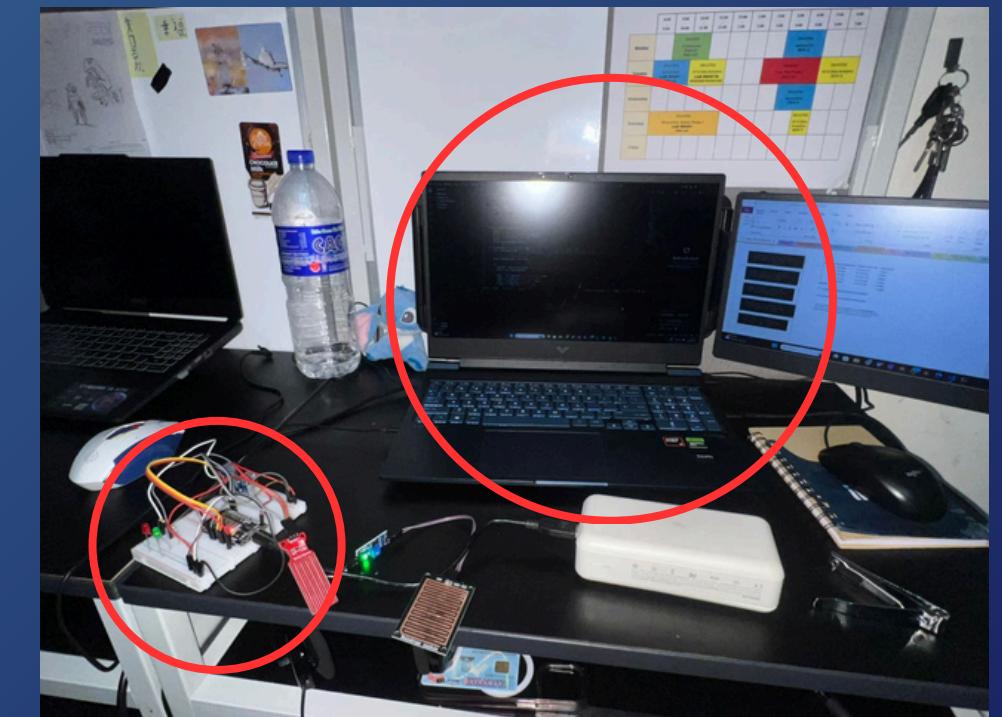
```
>> [TIMING] Sensor Recv: 01:25:40.540  
>> [TIMING] Telegram Sent: 01:25:41.594  
>> TOTAL LATENCY: 1.0535 seconds
```

```
--  
>> [TIMING] Sensor Recv: 01:26:27.244  
>> [TIMING] Telegram Sent: 01:26:28.287  
>> TOTAL LATENCY: 1.0433 seconds
```

```
--  
>> [TIMING] Sensor Recv: 01:26:59.478  
>> [TIMING] Telegram Sent: 01:27:00.637  
>> TOTAL LATENCY: 1.1588 seconds
```

```
--  
>> [TIMING] Sensor Recv: 01:27:33.300  
>> [TIMING] Telegram Sent: 01:27:34.472  
>> TOTAL LATENCY: 1.1715 seconds
```

```
--  
>> [TIMING] Sensor Recv: 01:28:09.212  
[TIMING] Telegram Sent: 01:28:10.889  
>> TOTAL LATENCY: 1.6774 seconds
```



Average latency : 1.2209 s

Experiment 2 : Calculate latency Variance from distance

Scenario 2: ~1 m from the source

```
>> [TIMING] Sensor Recv: 01:31:42.440  
>> [TIMING] Telegram Sent: 01:31:43.772  
>> TOTAL LATENCY: 1.3328 seconds
```

```
>> [TIMING] Sensor Recv: 01:32:18.069  
->> [TIMING] Telegram Sent: 01:32:21.481  
>> TOTAL LATENCY: 3.4117 seconds
```

```
>> [TIMING] Sensor Recv: 01:32:46.008  
>> [TIMING] Telegram Sent: 01:32:47.079  
>> TOTAL LATENCY: 1.0707 seconds
```

```
>> [TIMING] Sensor Recv: 01:33:17.720  
>> [TIMING] Telegram Sent: 01:33:18.811  
>> TOTAL LATENCY: 1.0911 seconds
```

```
>> [TIMING] Sensor Recv: 01:33:55.290  
>> [TIMING] Telegram Sent: 01:33:56.397  
>> TOTAL LATENCY: 1.1071 seconds
```



Average latency : 1.6027 s

Experiment 2 : Calculate latency Variance from distance

Scenario 3 : At the room

```
>> [TIMING] Sensor Recv: 01:42:07.994  
>> [TIMING] Telegram Sent: 01:42:09.080  
>> TOTAL LATENCY: 1.0856 seconds
```

```
>> [TIMING] Sensor Recv: 01:42:35.090  
>> [TIMING] Telegram Sent: 01:42:36.181  
>> TOTAL LATENCY: 1.0911 seconds
```

```
>> [TIMING] Sensor Recv: 01:43:24.179  
>> [TIMING] Telegram Sent: 01:43:25.485  
>> TOTAL LATENCY: 1.3064 seconds
```

```
>> [TIMING] Sensor Recv: 01:43:41.082  
>> [TIMING] Telegram Sent: 01:43:42.436  
>> TOTAL LATENCY: 1.3544 seconds
```

```
>> [TIMING] Sensor Recv: 01:43:57.604  
>> [TIMING] Telegram Sent: 01:43:58.721  
>> TOTAL LATENCY: 1.1167 seconds
```

Average latency : 1.1908 s

Experiment 3 : Calculate WiFi

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

---->
>> [TIMING] Sensor Recv: 01:51:09.438
>> [TIMING] Telegram Sent: 01:51:10.488
>> TOTAL LATENCY: 1.0504 seconds

---->

>> [TIMING] Sensor Recv: 01:58:35.974
>> [TIMING] Telegram Sent: 01:58:36.998
>> TOTAL LATENCY: 1.0240 seconds

---->

Bridge Connected to broker.hivemq.com
Listening to topic: monash/studentB/tunnel_data
```

---> Then try to On & Off
the WiFi

Result :
Python Bridge **can** connect

ESP 32 **Could not** connect

Experiment 3 : Calculate WiFi

Improvement : Add auto connect
`connect_mqtt()`

```
>> [TIMING] Sensor Recv: 02:14:39.948
>> [TIMING] Telegram Sent: 02:14:40.976
>> TOTAL LATENCY: 1.0281 seconds

-----
Bridge Connected to broker.hivemq.com
Listening to topic: monash/studentB/tunnel_data

-----
>> [TIMING] Sensor Recv: 02:17:08.028
>> [TIMING] Telegram Sent: 02:17:09.193
>> TOTAL LATENCY: 1.1652 seconds

-----
>> [TIMING] Sensor Recv: 02:17:39.159
>> [TIMING] Telegram Sent: 02:17:40.317
>> TOTAL LATENCY: 1.1583 seconds
```

Result : Pass