Zenatix Assignment

Bhanu Nautiyal

January 5, 2025

1 Mosquitto MQTT Broker

I am using "eclipse mosquitto mqtt" broker for this project.

1.1 Why I am using it?

1. Easy to implement especially for prototyping.
2. Lightweight by design.
3. Has inbuilt message storage.

1.2 Why I shouldnt use it ?

1. Has limited scalability.
2. No Web interface.
3. Complex integration with data management softwares like: SQL, redis etc.
4. No in-built clustering.

2 Assumptions

1. Since I am assuming that the broker I am using is just for decoupling the publishers and subscribers and has persistance of the topics only.

2. Thats why I will be storing the alarms data (either ACTIVE or CLEARED) in the Publisher side only.

3. Data coming from sensors will be taken as csv file.

4. Sensors are assumed as PRIMARY and SHUNT.

5. PRIMARY sensors can has 2 types of alarms: 1.) Simple alarm: Triggered when their values when reach a certain threshold for certain time duration. 2.) Conditional alarm: Triggered when the both PRIMARY sensor and SECONDARY sensor data points reach a certain threshold for certain time duration.

6. Alarms can have 2 states: 1.) ACTIVE 2.) CLEARED

7. Alarms which are currently active are only sent to broker.

8. But all alarm data either ACTIVE or CLEARED will be maintained at the database side for logging and monitoring etc.

3 Architecture of Project

3.1 Component Diagram of Alarm System

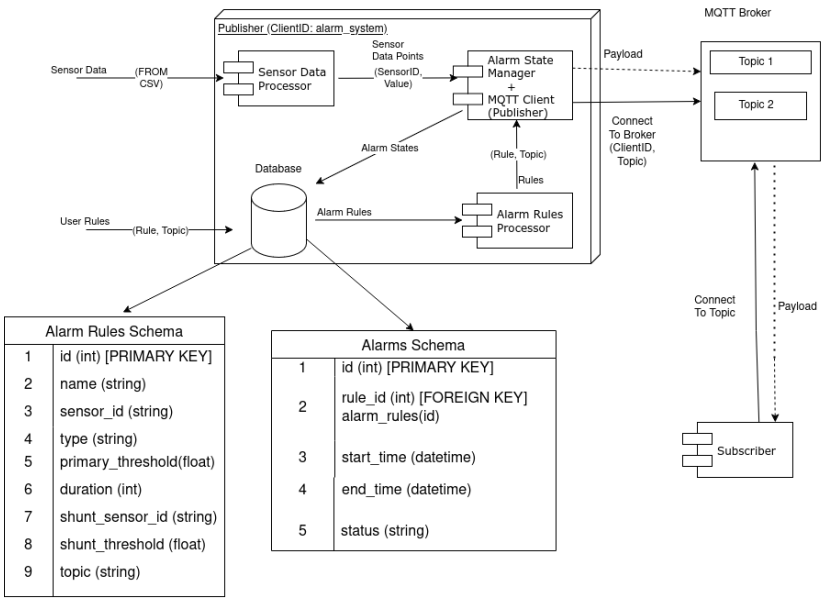

Figure 1: Component Diagram of the Alarm System.

4 Implementation of the Alarm System

4.1 Case 1: Simple Alarm

In this example alarm will get triggered only and only when data points in sensor is above a threshold and for some duration.

Eg: Let say I have sensor1 for Temperature
Its Alarm Rule is as below:

```
Name:             "High Temperature",
SensorID:         "temperature",
Type:             "simple",
PrimaryThreshold: 21.5,
Duration:         2,
Topic:            "alarms/temperature",
```

Its data is as:

Table 1: Temperature Sensor Data Log

Timestamp	Sensor ID	Value
2024-01-02 10:00:00	sensor1	20.00
2024-01-02 10:01:00	sensor1	19.45
2024-01-02 10:02:00	sensor1	22.40
2024-01-02 10:03:00	sensor1	25.40
2024-01-02 10:04:00	sensor1	29.32
2024-01-02 10:05:00	sensor1	28.40
2024-01-02 10:06:00	sensor1	17.20
2024-01-02 10:07:00	sensor1	19.15
2024-01-02 10:08:00	sensor1	18.10
2024-01-02 10:09:00	sensor1	12.10
2024-01-02 10:10:00	sensor1	20.00

Its timing diagram is as below:

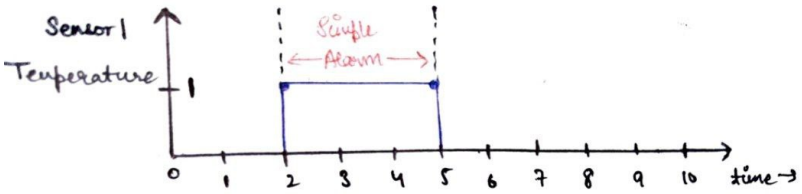

Figure 2: Timing Diagram of the Simple alarm.

Alarm Data stored in Sqlite3 database is as:

```
 sqlite3 alarms.db
SQLite version 3.47.2 2024-12-07 20:39:59
Enter ".help" for usage hints.
sqlite> SELECT * FROM alarms;
1|1|2025-01-04 19:39:51.805915935+05:30|2025-01-04 19:39:55.813122838+05:30|CLEARED
2|1|2025-01-04 19:39:51.805915935+05:30|2025-01-04 19:39:55.813122838+05:30|CLEARED
sqlite>
```

4.2 Case 2: Compound Alarm [Primary and Shunt Combination]

In this case alarm will triggered by not only primary condition but also by shunt condition as well.

1. Primary Alarm Rule (Sensor1: Temperature)

```
Name:             "High Temperature with Current",
SensorID:         "temperature",
Type:             "simple",
PrimaryThreshold: 21.5,
Duration:         2,
Topic:            "alarms/temperature",
```

2. Conditional Alarm Rule (Sensor1: Temperature and Sensor2: Current)

```
Name:             "High Temperature with Current",
SensorID:         "temperature",
Type:             "conditional",
PrimaryThreshold: 25.5,
Duration:         3,
ShuntSensorID:    "current",
ShuntThreshold:   0.2,
Topic:            "alarms/conditional",
```

Sensor1 Data (Temperature)

Table 2: Sensor1 Data (Temperature)

Timestamp	Sensor ID	Value
2024-01-02 10:00:00	sensor1	20.00
2024-01-02 10:01:00	sensor1	29.45
2024-01-02 10:02:00	sensor1	22.40
2024-01-02 10:03:00	sensor1	25.40
2024-01-02 10:04:00	sensor1	19.32
2024-01-02 10:05:00	sensor1	18.40
2024-01-02 10:06:00	sensor1	27.20
2024-01-02 10:07:00	sensor1	29.15
2024-01-02 10:08:00	sensor1	28.10
2024-01-02 10:09:00	sensor1	22.10
2024-01-02 10:10:00	sensor1	20.00

Sensor2 Data (Current)

Table 3: Sensor2 Data (Current)

Timestamp	Sensor ID	Value
2024-01-02 10:00:00	sensor2	0.02
2024-01-02 10:01:00	sensor2	0.01
2024-01-02 10:02:00	sensor2	0.03
2024-01-02 10:03:00	sensor2	0.00
2024-01-02 10:04:00	sensor2	0.00
2024-01-02 10:05:00	sensor2	0.07
2024-01-02 10:06:00	sensor2	0.84
2024-01-02 10:07:00	sensor2	0.93
2024-01-02 10:08:00	sensor2	0.92
2024-01-02 10:09:00	sensor2	0.56
2024-01-02 10:10:00	sensor2	0.00

Timing Diagram is as below:

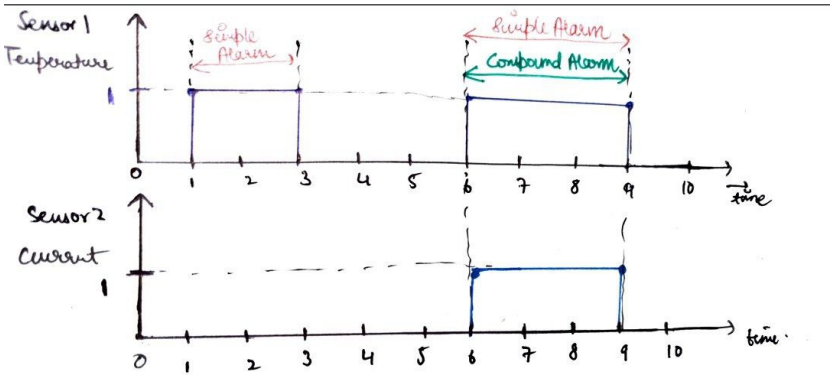

Figure 3: Timing Diagram of the Compound alarm.

Alarm data stored in SQLite3 is as:

```
 sqlite3 alarms.db
SQLite version 3.47.2 2024-12-07 20:39:59
Enter ".help" for usage hints.
sqlite> SELECT * FROM alarms;
```

```
1|1|2025-01-04 20:01:10.483906749+05:30|2025-01-04 20:01:13.487491796+05:30|CLEARED
2|1|2025-01-04 20:01:15.490308617+05:30|2025-01-04 20:01:19.498596802+05:30|CLEARED
3|1|2025-01-04 20:01:15.490308617+05:30|2025-01-04 20:01:19.498596802+05:30|CLEARED
4|2|2025-01-04 20:01:15.490309358+05:30|2025-01-04 20:01:19.500736419+05:30|CLEARED
sqlite>
```