Final Project: Smart Biosphere

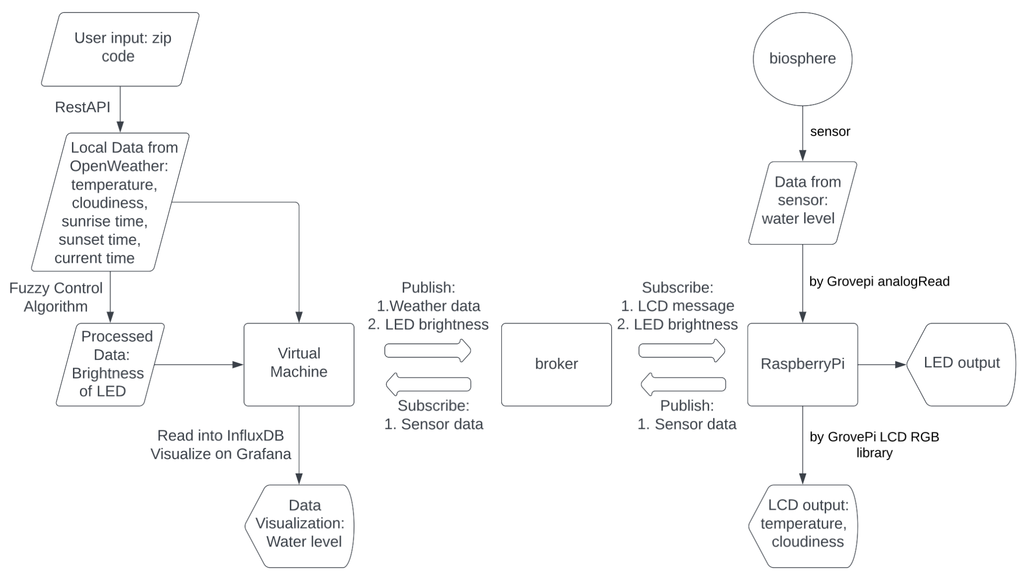
Team members: Christina Wu & Siyang Wang

December 1, 2022

Description

This project seeks to create an environmental intelligence program. It harnesses many techniques we learnt in EE250 like collecting open-source weather data by restAPI, gathering real-time data from sensors, data-processing algorithm, MQTT communication system and data visualization by InfluxDB and Grafana. Incorporating all these techniques, we build a real-time model considering biosphere factors like temperature, cloudiness, water level, sunset and sunrise time, and current time to output the brightness of LED light, temperature and cloudiness on LCD, and water-level visualization on Grafana.

Block Diagram

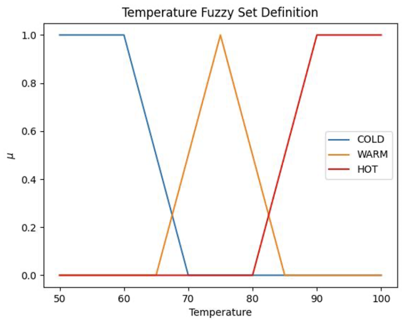


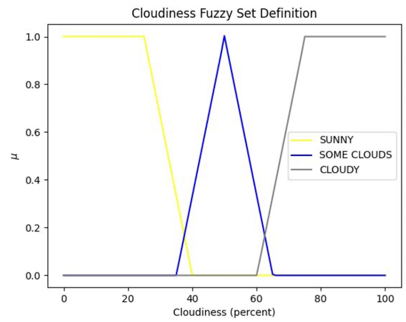
Components and Protocols

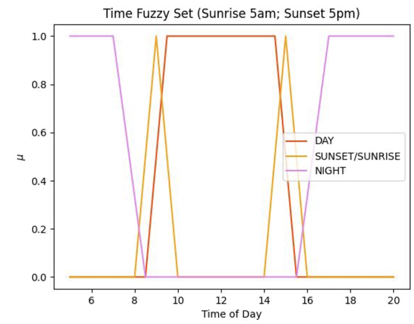
As shown on the block diagram, we have sensor related and VM related components: sensor parts includes biosphere, RPi, publish and subscribe data, LCD and LED; VM related parts includes restAPI, control algorithm, publish and subscribe data, and data visualization. We applied MQTT protocol.

Key processing technique: Fuzzy Control Algorithm

In classical control algorithms, definite numbers are often used to determine the state of the system, which is used in the control logic of the algorithm. However, definite numbers often fail to capture the nuances of realistic states. To encapsulate the nuanced nature of states, we employed the fuzzy control algorithm in our project to accurately model and process the inherent ambiguity of our system. To achieve this, we defined three fuzzy sets to translate the numerical inputs of temperature, time, and cloudiness into fuzzy function μ which indicates the probability of states. The rules of translation are presented in the graphs. Obtaining the inputs in terms of their fuzzy states, we are able to use these data to write our control logic described by the if-statements below.








```
if time==NIGHT or cld==CLD:
    LED=FULL (100%)
if time==SRSS and (temp==COLD or cld==SCLD):
    LED=HIGH (75%)
if time==SRSS and (cld==SNNY or temp==COLD):
    LED=MID (50%)
if time==DAY and (cld==SCLD or temp==WARM):
    LED=LOW (25%)
if time==DAY and cld==SNNY:
    LED=OFF (0%)
```

For each rule above, we use the min value of μ's for AND logic and max value of μ's for OR logic, converting the fuzzy states into the weight w_i of each LED output intensity O_i. In the last step, we calculate the weighted average of LED output intensity in a process called "de-fuzzification," obtaining our calculated LED output with the following formula.

$$I_{\text{out}} = \frac{1}{W}\sum_i w_i O_i \quad W = \sum_i w_i$$

which will then be published to the MQTT broker by the VM.

Limits and Improvements

- The rules we made for Fuzzy Control Algorithm could be more deliberate. We only considered the data grabbed from OpenWeather, while data from sensors are not taken into account. One reason for not taking water-level data is the low accuracy of our water-level sensor. Next time, we should use water-level sensor with higher quality, and incorporate the sensor data into our Fuzzy Control Algorithm.
- The small LED in the GrovePi kit isn't enough to show the variation of brightness vividly, so we bought a larger LED strip. However, the RPi current draw only supports current under 16mA, while the LED strip requires much higher current to operate. This problem could be solved by adding a current amplifier using MOSFET.
- We designed the dashboard of Grafana to be relatively simple, and we only visualized one factor, water level data, on Grafana. The most idealized situation is to visualize all types of data we have on grafana with a better-designed dashboard, which we could do if we have enough time.
- The design of our program is expandable, meaning there is places for adding more sensors like moisture sensor into the control algorithm. In that case, we could use water pump or spray to control the moisture.

Evaluation of Team

We brainstormed the ideas together first, then decided to use the idea of smart biosphere from Siyang. Then, we divided work as following: Christina was responsible for weather data collection, VM related MQTT, and writing the report: Siyang was responsible for sensor data collection, RPi related MQTT, control algorithm and data visualization. We worked together most of the time. We also helped each other on debugging or collecting resources.
We could start earlier next time to design our project better. We should also supervise each other to make the work more effective.