

Documentation

Note: The program does compile but nothing will happen as you will need to add the necessary credentials.

Note: The “ArrayList.h” API found from in **Inventory_monitor_ESP32** is copied from:
<https://github.com/wachidsusilo/ArrayList>

It makes it easy to work with arrays as its basically a c++ vector datastructure.

Parts used:

- 1 x TTGO
- 1 x ESP32
- 3 x pushbuttons
- 3 x 10Ω resistors for the buttons
- 1 x LCD

It may seem silly to use an LCD, but I first tried with a TFT hooked directly on TTGO and nothing happened. I then tried on ESP32 and same story. So, I believe it has finished with its purpose.

So, the only way to go was with an LCD.

As stated, the project consists of a LORA board as well as an ESP32 board. The devices communicate with each other through a local MQTT broker.

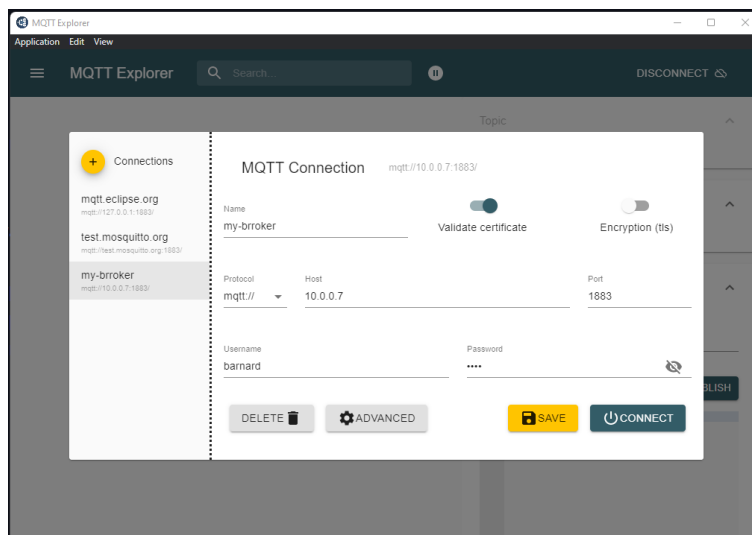
The ESP32 does the “heavy” work, it registers how many drinks are left in the inventory. Each time a customer buys a drink the ESP32 sends a publish message to LORA, there forth LORA sends the data to AWS, witch in turn gets passed down by AWS Amplify to a simple react native app.

```
PS C:\Program Files\mosquitto> net start mosquitto

The Mosquitto Broker service was started successfully.

PS C:\Program Files\mosquitto> .\mosquitto_sub -h "10.0.0.7" -t test -u "barnard" -P "1234"
```

Local broker made on Windows with mosquito and authentication.



Next steps:

- I believe the code has some memory leaks somewhere, so I will have a look at that.
- Obviously the react app need some styling/notifications.
- Configure a Firebase database to store the data needed for the graph.