

IOT - TP

Nous allons utiliser des cartes ESP32 (pas des arduino mais similaire). Ces cartes peuvent contenir des modules WiFi, Bluetooth ou du Lora ainsi que des capteurs de température ou d'humidité.

PARTIE n°1

Laboratoire n°1

Tips de compréhension pour l'arduino:

- fonction *setup()*: équivalent du *main()* en C;
- fonction *loop()*: boucle en continue jusqu'à la condition d'arrêt

Point 1.1 : Affichage des données sur l'écran OLED

Affichage sur l'écran possible en appelant la fonction *dispData()* dans la fonction *loop*

Appel de la fonction *display()* dans la fonction *loop()*

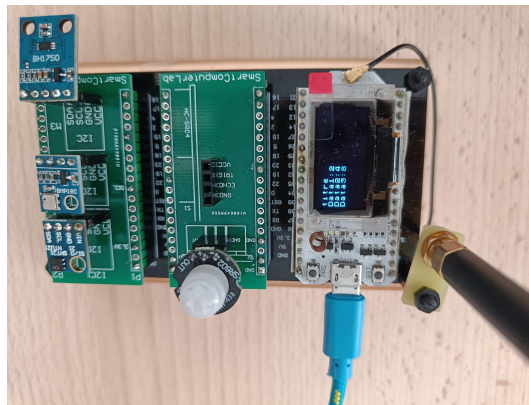


Figure 1 - Affichage sur l'écran OLED de la carte

Point 1.2.1 : Capture et affichage des valeurs

Pour visualiser l'affichage : besoin d'ouvrir le serial monitor

Après un problème capteur, nous avons enfin une température cohérente

```
/dev/ttyUSB0
Humidity(%RH): 56.4   Temperature(C): 23.7   Read Time: 111469
Humidity(%RH): 56.3   Temperature(C): 23.7   Read Time: 111469
Humidity(%RH): 56.2   Temperature(C): 23.7   Read Time: 111469
Humidity(%RH): 56.2   Temperature(C): 23.7   Read Time: 111469
Humidity(%RH): 56.2   Temperature(C): 23.7   Read Time: 111469
Humidity(%RH): 56.2   Temperature(C): 23.8   Read Time: 111469
Humidity(%RH): 56.3   Temperature(C): 23.8   Read Time: 111469
Humidity(%RH): 56.3   Temperature(C): 23.8   Read Time: 111469
Humidity(%RH): 56.2   Temperature(C): 23.8   Read Time: 111469
```

Figure 2 - Affichage de la température sur le serial monitor

Point 1.2.2 : Capture de la température/humidité par HTU21D

Nos capteurs ne fonctionnent pas.

Laboratoire n°2 - Communication en WiFi et serveur

2.2 Mode WiFi – STA, client WEB et serveur ThingSpeak

2.2.1 Envoi des données sur ThingSpeak

Nous avons fait le choix d'utiliser ThingSpeak.com. Ma première étape a été de créer un compte sur le site. La seconde étape a été d'ouvrir un canal sur lequel nous allons envoyer des données. Enfin il nous a fallu setup la partie arduino.

Nous obtenons finalement sur ThingSpeak le graphe suivant:

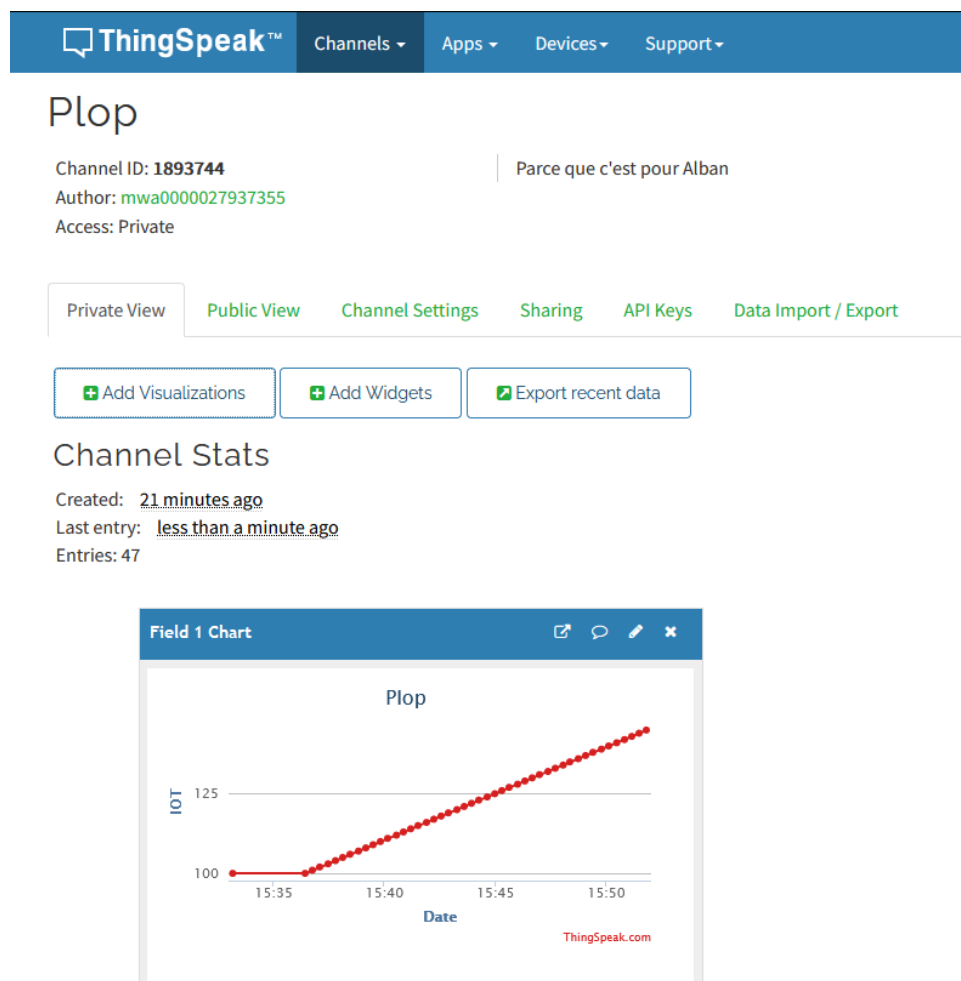


Figure 3 - Capture d'écran du site ThingSpeak

Les points rouges du graphes indiquent la réception de données sur le channel en fonction de l'heure de la réception.

Laboratoire n°3 – MQTT broker et clients

Point 3.2: Protocole MQTT

Nous sommes les receveurs des messages du topic “Lise” que nous avons initié sur “test.mosquitto.org” (autrement dit, nous sommes les clients). Nous avons dû renseigner les identifiants WiFi ainsi que les informations liées au topic. Nous avons dû par ailleurs modifier le Broker :

```
// Reconnect to client
void reconnect() {
  // Loop until we're reconnected

  Serial.println("Connected to the WiFi network");

  client.setServer("test.mosquitto.org",1883);
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Attempt to connect
    if (client.connect(ID)) {
      client.subscribe(TOPIC);
      Serial.println("connected");
      Serial.print("Subscribed to: ");
      Serial.println(TOPIC);
      Serial.println('\n');
    }
  }
}
```

Figure 4 - Code avec la modification du broker

Finalement, nous obtenons la sortie suivante avec tous les messages reçus sur le topic:

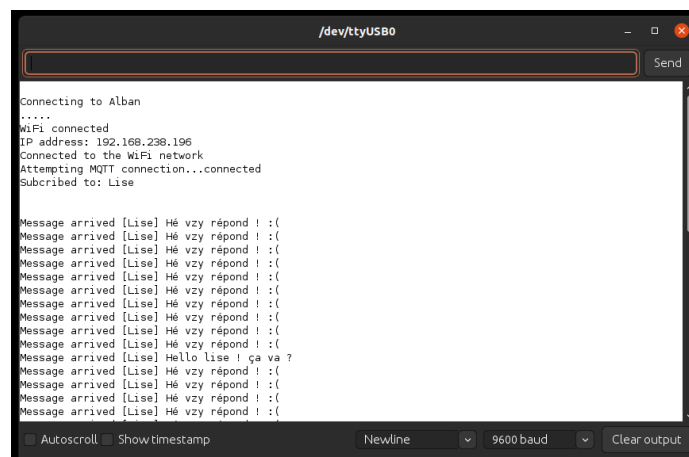


Figure 5 - Console de sortie attestant de la réception des messages sur le topic

En conséquence, du côté de l'expéditeur, ils avaient la fenêtre suivante:

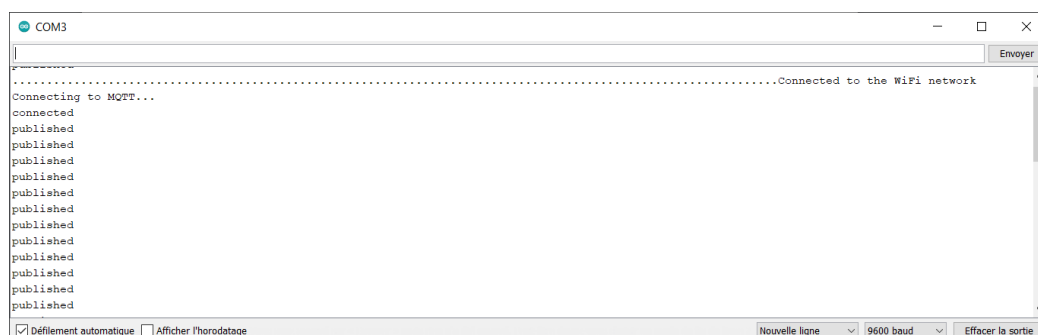


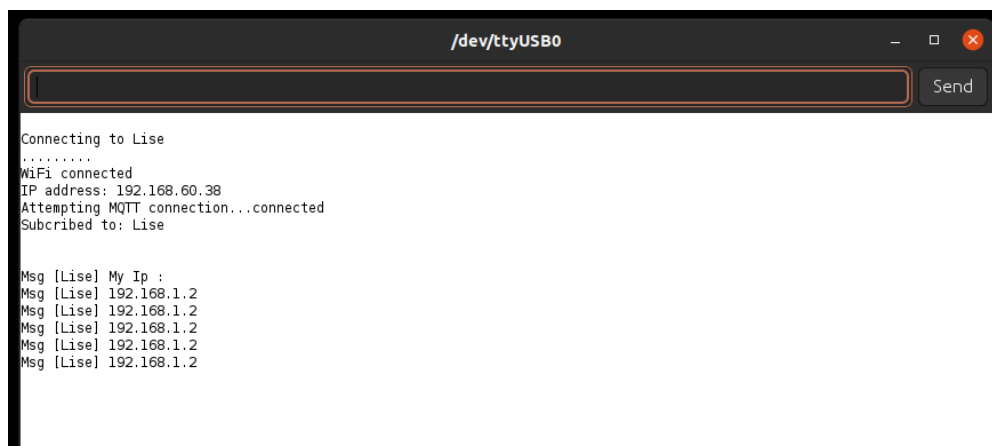
Figure 6 - Console de sortie attestant de la réception des messages sur le topic

PARTIE n°2

Pour cette partie, nous sommes encore des clients et utilisons dans un premier temps MQTT.

Nous reprenons le travail précédent et allons devoir recevoir de la part du serveur l'IP du serveur web. En effet, pour le moment nous sommes connectés sur internet via un point wifi. Pour passer en Lora, dès que nous avons reçu cette adresse IP, nous nous déconnectons (client et serveur) d'internet pour se connecter au point d'accès du serveur.

Voici les premiers messages reçus, envoyés par le serveur. Nous obtenons l'adresse IP de celui-ci.

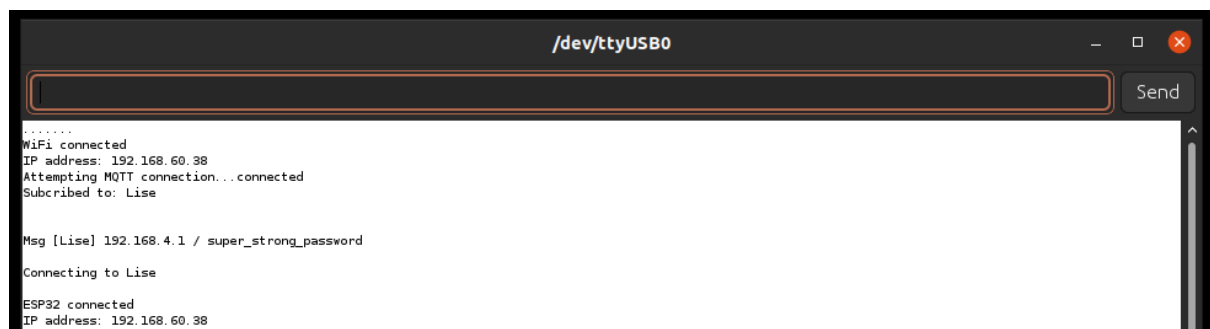


```
/dev/ttyUSB0
Connecting to Lise
.....
WiFi connected
IP address: 192.168.60.38
Attempting MQTT connection...connected
Subscribed to: Lise

Msg [Lise] My Ip :
Msg [Lise] 192.168.1.2
Msg [Lise] 192.168.1.2
Msg [Lise] 192.168.1.2
Msg [Lise] 192.168.1.2
Msg [Lise] 192.168.1.2
Msg [Lise] 192.168.1.2
```

Figure 7 - Console de sortie attestant de la réception d'une adresse IP sur le topic

Nous allons maintenant tenter de nous connecter au point d'accès du serveur après s'être d'abord déconnecté du WiFi.

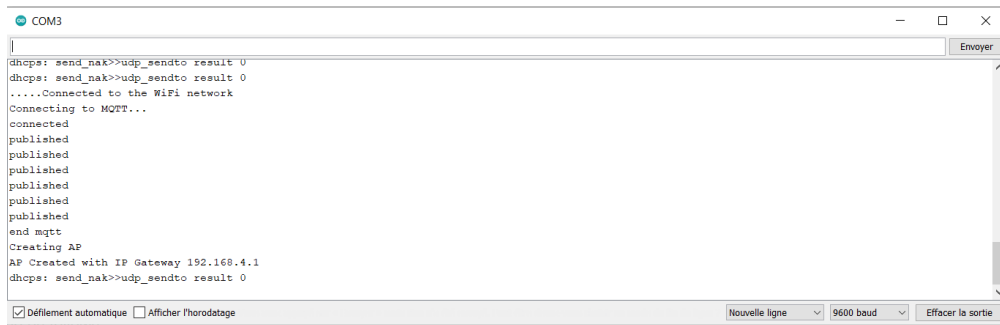


```
/dev/ttyUSB0
.....
WiFi connected
IP address: 192.168.60.38
Attempting MQTT connection...connected
Subscribed to: Lise

Msg [Lise] 192.168.4.1 / super_strong_password

Connecting to Lise
ESP32 connected
IP address: 192.168.60.38
```

Figure 8 - Console de sortie attestant de notre connexion au point d'accès



```
dhcps: send_nak>>udp_sendto result 0
dhcps: send_nak>>udp_sendto result 0
.....Connected to the WiFi network
Connecting to MQTT...
connected
published
published
published
published
published
published
end mqtt
Creating AP
AP Created with IP Gateway 192.168.4.1
dhcps: send_nak>>udp_sendto result 0
```

COM3

Envoyer

☒ Défilement automatique ☐ Afficher l'horodatage

Nouvelle ligne 9600 baud Effacer la sortie

Figure 9 - Console de sortie du serveur attestant de notre connexion au point d'accès

A partir de ce moment, nous pouvons commencer à échanger des messages. Cette action n'a pas pu être capturée pour être ajoutée dans le rapport.