

## COMPTE RENDU – SÉANCE 6

Lors de cette séance, j'ai pu résoudre un problème rencontré sur la carte depuis quelques semaines, avec l'aide de monsieur Peter. Le programme amenait la carte à répéter des lignes de code en continu au lieu de se connecter au réseau. Il s'agissait en fait d'une carte Heltec Wifi Lora 32 et non d'une Heltec Wifi Lora 32 V2, et le téléversement ne se faisait donc pas sur la bonne carte ce qui entraînait l'erreur précédente. Le problème du partage de connexion est toujours présent mais sera normalement résolu par l'ajout d'une borne wifi au projet qui permettra de connecter la carte au réseau.

J'ai également ajouté quelques lignes au programme de l'ouverture physique de la boîte afin d'inclure la validation à distance du propriétaire dans le code.

## CODE CARTE UNO

```
#include <SoftwareSerial.h>

// variables utilisées pour la communication des cartes
const int tx = 01;
const int rx = 00;
SoftwareSerial MySerial(rx, tx);

String message_recu = "";

void setup() {
  Serial.begin(9600);
  MySerial.begin(4800);
  servo_10.attach(10);
  Serial.println("Entrez votre code :");
}

void loop() {
  lecturechiffre();
  readData();
  Serial.println(message_recu);
}

void lecturechiffre(){
  char transformechiffre = kp4x4.getKey(); //récupère le chiffre du keypad
  if (transformechiffre) {
    tab[i]=transformechiffre;//mets les chiffres à la suite des autres
    i=i+1;
    if(i==CODE){
      Serial.println("****");
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("****");
      delay(2000);
      i=0;

      if((strcmp(tab,codebon)==0)){ //compare le code entré et celui attendu
        Serial.println("Code bon !");
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Code bon");
        digitalWrite(led_ORANGE,HIGH);
        if (message_recu == 1){
          servo_10.write(-90); //ouvre le loquet
          digitalWrite(led_ORANGE,LOW);
          digitalWrite(led_ROUGE,LOW);
          digitalWrite(led_VERT, HIGH);
        }
        if (message_recu == 0){
          digitalWrite(led_ORANGE,LOW);
          digitalWrite(led_ROUGE,HIGH);
          digitalWrite(led_VERT,LOW);
        }
      }
    }
  }
}
```

```

void readData() {
  while (MySerial.available()) {
    char inChar = (char)MySerial.read();
    if (inChar != '\n') {
      message_recu += inChar;
    }
    else{
      message_recu = "";
    }
  }
}

```

Ce code n'est pas celui complet, il montre simplement les lignes permettant de récupérer les informations envoyées par la carte ESP32 et de les ajouter à une variable (message\_recu) sur la UNO afin d'autoriser ou non l'ouverture de la boîte. Lorsque le propriétaire autorise l'ouverture à distance, l'ESP32 envoie un "1" à la UNO, dans le cas contraire un "0".

## CODE CARTE ESP32

```

#include "HardwareSerial.h"

HardwareSerial MySerial(1);

// Replace with your network credentials
const char* ssid = "AndroidCP";
const char* password = "Dzdz0102B";

// variable communication RX/TX
//
//const int tx = 13;
//const int rx = 12;
//SoftwareSerial MySerial(rx, tx);

// variables pour stocker les messages
String message_recu = "";
String etat;

// variables etat leds
int boiteOuverte = 0;

void setup(){
  // Serial port for debugging purposes
  Serial.begin(115200);
  //MySerial.begin(4800);
  MySerial.begin(4800, SERIAL_8N1, 13, 12);

  // Route to set the opening of the box
  server.on("/oui", HTTP_GET, [](AsyncWebServerRequest *request){
    boiteOuverte = 1;
    request->send(SPIFFS, "/index.html", String(), false, processor);
  });

  // Route to set the closing of the box
  server.on("/non", HTTP_GET, [](AsyncWebServerRequest *request){
    boiteOuverte = 0;
    request->send(SPIFFS, "/index.html", String(), false, processor);
  });

  void loop(){
    MySerial.println(boiteOuverte);
  }
}

```

Ce code n'est également pas celui complet et montre seulement les lignes de codes permettant la communication. Ici lorsque la propriétaire autorise l'ouverture la variable "boiteOuverte" prend la valeur 1, à l'inverse 0. La variable est ensuite envoyée sur "MySerial", moniteur série commun aux deux cartes permettant le transfert de données de l'ESP32 vers la UNO.

Monsieur Peter nous a également conseillé de tout rassembler sur l'ESP32 afin d'éviter la communication entre les deux cartes (qui ne marche pas encore) et de n'en garder qu'une. Nous avons décidé d'utiliser deux cartes par peur de manque de place mais il s'avère que tous les appareils pourraient être câblés sur l'ESP32 mais aussi car nous n'avons pas de bibliothèque compatible avec la carte Wifi pour certains

composants comme le servo moteur. Nous essaierons donc pendant les vacances de tout rassembler sur une carte ou à défaut de faire marcher la communication.

Nous avons également prévu, avec Elisa pour qu'il ne reste plus qu'aux dernières séances à régler les derniers problèmes et tout installer dans la boîte.