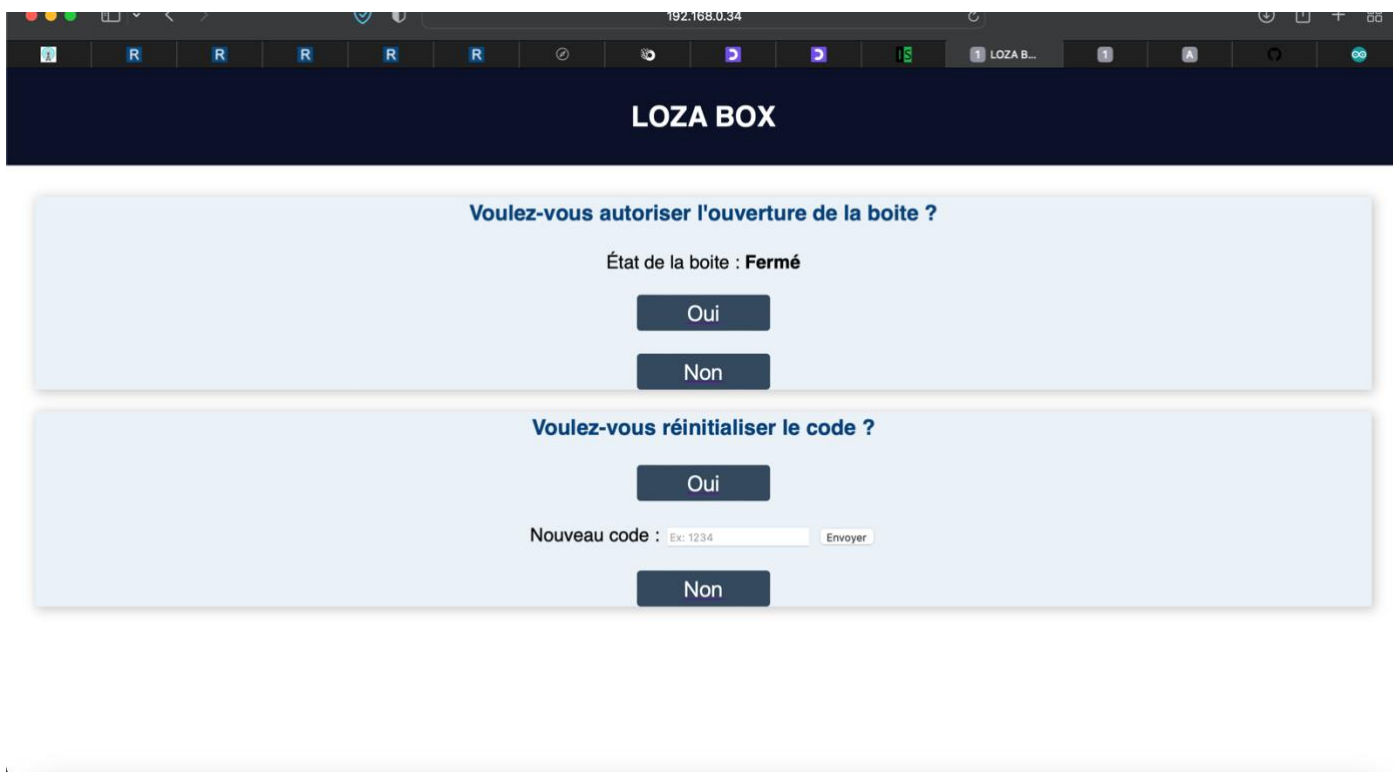


COMPTE RENDU – SÉANCE 4

TRAVAIL RÉALISÉ LORS DE LA SÉANCE

Lors de cette quatrième séance nous avons présenté avec Élisabeth le premier oral du projet. J'ai, durant la séance, également essayé de faire communiquer les deux cartes Arduino : ESP32 et Arduino UNO ensemble afin de pouvoir récupérer les informations de la UNO sur l'ESP32. En effet, j'ai travaillé durant la semaine sur le changement de programme entre l'ESP8266 (ancienne carte) et l'ESP32. Le nouveau programme obtenu permet de connecter la carte au réseau wifi, et de créer un serveur web html (modifiable avec un fichier css) qui permet pour l'instant de contrôler deux Led.



En voulant tout regrouper (keypad, servomoteur et led) sur l'ESP32 j'ai vu que la bibliothèque du servomoteur n'était pas compatible avec la carte Wifi et le keypad ne répondait pas correctement. J'ai donc pendant la séance, cherché à lier les deux cartes via les pins RX/TX.

CODE

CODE POUR LA CONNEXION WIFI

```
#include <SPIFFS.h>
#include "WiFi.h"
#include <SoftwareSerial.h>
#include "ESPAsyncWebServer.h"
#include "SPIFFS.h"

//liaison entre les deux cartes

String readString;
```

```

#define rxPin 03
#define txPin 01
SoftwareSerial mySerial = SoftwareSerial(rxPin, txPin);

// Replace with your network credentials
const char* ssid = "Laure-Anne";
const char* password = "@12Mars2002";

// Set LED GPIO
const int led_VERTE = 2;
const int led_ROUGE = 13;
// Stores LED state
String ledState;

// Create AsyncWebServer object on port 80
AsyncWebServer server(80);

// Replaces placeholder with LED state value
String processor(const String& var){
  Serial.println(var);
  if(var == "STATE"){
    if(digitalRead(led_VERTE)){
      ledState = "Ouvert";
    }
    else{
      ledState = "Fermé";
    }
    Serial.print(ledState);
    return ledState;
  }
  return String();
}

void setup(){
  // Serial port for debugging purposes
  mySerial.begin(9600);
  Serial.begin(9500);
  #define pin modes for tx, rx:
  pinMode(rxPin, INPUT);
  pinMode(txPin, OUTPUT);

  pinMode(led_VERTE, OUTPUT);
  pinMode(led_ROUGE, OUTPUT);

  // Initialize SPIFFS
  if(!SPIFFS.begin(true)){
    Serial.println("An Error has occurred while mounting SPIFFS");
    return;
  }

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi..");
  }

  // Print ESP32 Local IP Address
  Serial.println(WiFi.localIP());

  // Route for root / web page
  server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send(SPIFFS, "/index.html", String(), false, processor);
  });

  // Route to load style.css file
  server.on("/style.css", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send(SPIFFS, "/style.css", "text/css");
  });

  // Route to set GPIO to HIGH
  server.on("/oui", HTTP_GET, [](AsyncWebServerRequest *request){
    digitalWrite(led_VERTE, HIGH);
    digitalWrite(led_ROUGE, LOW);
    //digitalWrite(led_ORANGE, LOW);

    request->send(SPIFFS, "/index.html", String(), false, processor);
  });

  // Route to set GPIO to LOW
  server.on("/non", HTTP_GET, [](AsyncWebServerRequest *request){
    digitalWrite(led_VERTE, LOW);
    digitalWrite(led_ROUGE, HIGH);
    //digitalWrite(led_ORANGE, LOW);

```

```

    request->send(SPIFFS, "/index.html", String(), false, processor);
  });

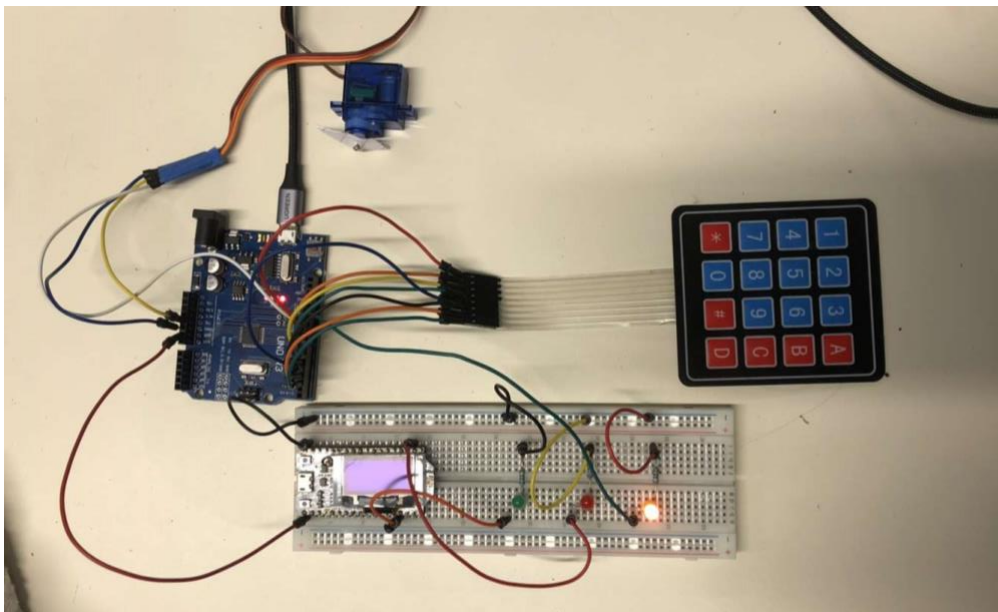
  // Start server
  server.begin();
}

void loop() {
}

```

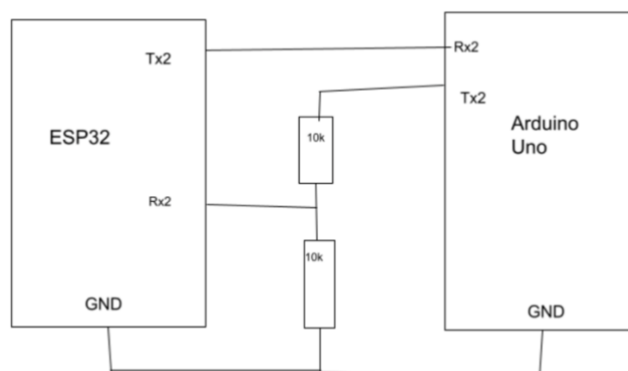
Les lignes en rouges dans le code entraînent des erreurs. En effet la carte ESP32 ne reconnaît pas le "include <SoftwareSerial.h>" ce qui amène l'obtention d'erreurs dans le reste du programme. La bibliothèque espSoftwareSerial pourrait être une solution à ce problème.

MONTAGE



PROBLÈMES RENCONTRÉS

La carte ESP32 fonctionne parfaitement en se connectant au réseau wifi mais rame énormément lorsqu'on la connecte en partage de connexion. De plus, afin de connecter l'arduino UNO à l'ESP32 il faut convertir le 5v de la UNO en 3,3v de l'ESP32. Pour ce faire le site internet d'Arduino ajoute deux résistances au montage :



SITES UTILISÉS :

- <https://www.arduino.cc/reference/en/libraries/espsoftwareserial/>
- <https://create.arduino.cc/projecthub/nidhi17agarwal/uart-communication-between-arduino-uno-and-esp32-1170d5>
- <https://icircuit.net/arduino-interfacing-arduino-uno-esp32/2134>

(liens en commentaire sur le github)