

COMPTE RENDU - SÉANCE 3

TRAVAIL RÉALISÉ LORS DE LA SÉANCE :

Lors de cette troisième séance j'ai repris le code réalisé lors de la séance précédente et essayé de le lier avec le code permettant d'ouvrir la boîte et les codes réalisés par Elisa permettant de créer une zone de texte afin de réinitialiser le code (ce qui sera fait par la suite).

Nous avons décidé de changer de carte et de passer de la esp8266 à la ESP32 puisque l'ESP8266 mettait trop de temps à répondre. Monsieur Peter nous a donc conseillé d'utiliser l'ESP32, successeur de l'ESP8266. La puce est plus performante et possède plus de pin ce qui permettra de connecter directement le keypad, les LED et le servomoteur à la carte.

CODE :

```
#include <WiFi.h>

const char* ssid = "Laure-Anne"; // nom du réseau wifi (ici partage de connexion)
const char* password = "@12Mars2002"; // mot de passe du réseau wifi

int ledVERT = 16;
int ledROUGE = 14;
WiFiServer server(80);

void setup() {
    Serial.begin(9600);
    delay(100);
    pinMode(ledVERT, OUTPUT);
    pinMode(ledROUGE, OUTPUT);
    digitalWrite(ledVERT, LOW);
    digitalWrite(ledROUGE, LOW);

    // Connexion wifi
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);

    WiFi.begin(ssid, password);

    // connection en cours on attend que le module se connecte
    while (WiFi.status() != WL_CONNECTED){
        delay(500);
        Serial.print(".");
    }

    // Wifi connecté
    Serial.println("WiFi connecté");

    // Démarrage du serveur
    server.begin();
    Serial.println("Serveur démarré !");

    // Affichage de l'adresse IP
    Serial.print("Utiliser cette adresse URL pour la connexion :");
    Serial.print("http://");
    Serial.print(WiFi.localIP());
    Serial.println("/");
}

void loop() {
    WiFiClient client;
```

```

// Vérification si le client est connecté.
client = server.available();
if (!client){
    return;}

// Attendre si des données sont envoyées (ici "OUI" ou "NON")
Serial.println("nouvelle demande");
while(!client.available()){
    delay(1);}

String request = client.readStringUntil('\r');
Serial.println(request);
client.flush();

int valeur = LOW;
if (request.indexOf("/LED=ON") != -1) {
    digitalWrite(ledVERT, HIGH); // allumer la led verte
    digitalWrite(ledROUGE, LOW); // éteindre la led rouge
    valeur = HIGH;
}
if (request.indexOf("/LED=OFF") != -1) {
    digitalWrite(ledVERT, LOW); // éteindre la led verte
    digitalWrite(ledROUGE, HIGH); // allumer la led rouge
    valeur = LOW;
}

// Réponse
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println("");
client.println("<!DOCTYPE HTML>");
client.println("<head>");
client.println("<meta charset='utf-8'/>");
client.println("<link rel='stylesheet' href='../ARDUINO.css' type='text/css' />");
client.println("<title> LOZA BOX </title>");
client.println("</head>");
client.println("<html>");
client.println("<body>");
client.println("<header>");
client.println("<h1> BIENVENUE SUR LE SITE OFFICIEL DE LOZA BOX </h1>");
client.println("</header>");

client.print("Acceptez-vous l'ouverture de la boite ? ");

if(valeur == HIGH) {
    client.print("Oui");
} else {
    client.print("Non");
}

client.println("<body>");
client.println("<br><br>");
client.println("<a href='\"/LED=ON\"'><button>Oui </button></a>");
client.println("<a href='\"/LED=OFF\"'><button>Non </button></a><br />");
client.println("<br><br>");
client.println("</body>");
client.println("</html>");

//réinitialiser le code
client.print("Voulez-vous reinitialiser le code ? ");

client.println("<br><br>");
client.println("<a href='\"/CODE=OUI\"'><button>Oui </button></a>");
client.println("</body>");
client.println("</html>");

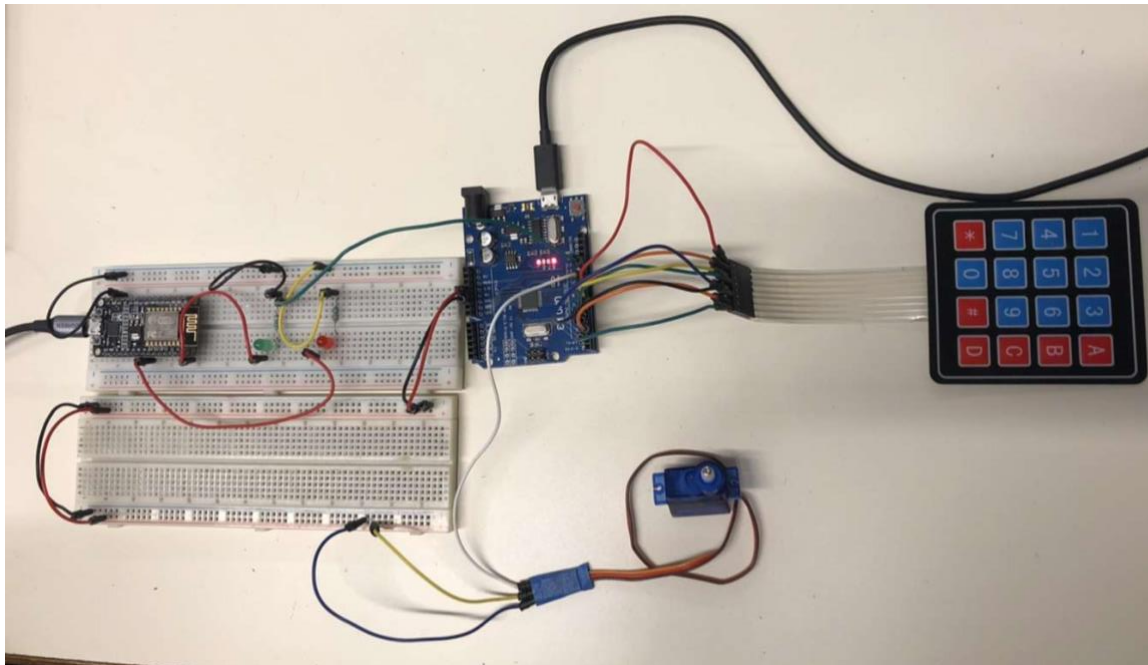
//if (request.indexOf("/CODE=OUI") != -1) {
client.println("<br><br>");
client.println("<form method='post' action='traitement.php'>");
client.println("<p>");
client.println("<label for='codebon'>Veuillez taper un code à 4 chiffres : </label>");
client.println("<input type='text' name='codebon' id='codebon'/>");
client.println("</p>");
client.println("</form>");
client.println("</body>");
client.println("</html>");
//}

delay(1);
Serial.println("Demande finie");
Serial.println("");
}

```

Le code fonctionnait mais mettait trop de temps à répondre avec le montage et le code fait par Elisa pour récupérer l'état d'une Led. Il faudra donc pour la prochaine séance tout regrouper sur un même fichier. La partie permettant de récupérer un nouveau code a également posé problème, notamment au niveau de la syntaxe qui ne fonctionnait pas comme voulu avec Arduino : la zone de texte ne s'affichait pas ou des erreurs apparaissaient. Enfin, le programme n'est pas optimal pour l'ESP32. J'ai donc trouvé plusieurs autres codes fonctionnants sur le même principe mais incluant davantage de bibliothèques qui permettent de créer un serveur web, à partir de l'ESP32, qu'il faudra modifier pour la prochaine séance (en utilisant le code ci-dessus).

MONTAGE :



Ce montage marchait mais sera modifié pour la prochaine séance comme tous les branchements se feront sur l'ESP32.

SITES UTILISÉS :

<https://randomnerdtutorials.com/?s=esp32+web+server>
<https://randomnerdtutorials.com/esp32-pinout-reference-gpios/>

(les liens sont dans les commentaires du github).