

## Compte rendu – Deuxième séance

### Travail réalisé lors de la deuxième séance :

Lors de la deuxième séance, j'ai travaillé sur la mise en lien via wifi du téléphone avec la carte esp8266. En faisant des recherches, cette carte semblait être la plus utilisée pour la connexion wifi (avec l'esp32), c'est pour cela que nous avons choisi la carte esp8266. J'ai donc réalisé un programme permettant de contrôler à distance l'allumage de plusieurs Led afin de voir dans un premier temps si la carte fonctionnait mais également pour comprendre comment celle-ci marchait.

J'ai également réglé le problème du bouton poussoir rencontré lors de la première séance. Le montage du bouton n'était en effet pas bon et la boucle "for" empêchait le "if" de fonctionner correctement, il fallait donc simplement faire basculer le servomoteur entre deux positions.

### Après rectification du code, le montage avec le bouton marche :

```
#include <Servo.h>

Servo servoM;
const int bouton = 6;
int etat_bouton = 1;
int ancienne_val =1;
int val =1;

void setup() {
  pinMode(bouton, INPUT);
  servoM.attach(9);
  Serial.begin(9600);
}

void loop() {

  val = digitalRead(bouton);

  if ((val == LOW)&&(ancienne_val == HIGH)) {
    etat_bouton = 1 - etat_bouton;
    delay(20);
    Serial.print(etat_bouton);
  }

  ancienne_val=val;

  if (etat_bouton == HIGH) {
```

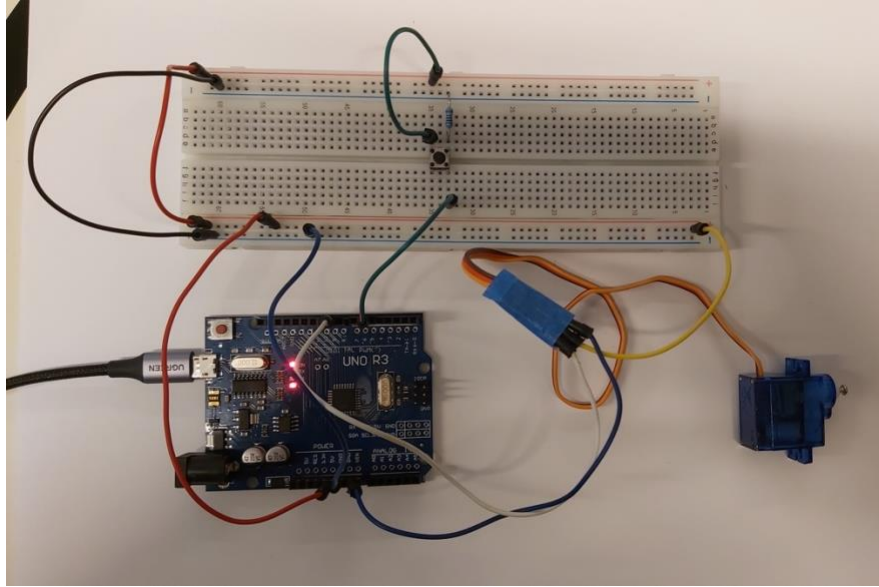
```

servoM.write(90);
}

if(etat_bouton == LOW) {
    servoM.write(-90);
}
}

```

## Montage :



## Code pour la connexion wifi :

```

#include <ESP8266WiFi.h>

const char* ssid = "Laure-Anne"; // nom du réseau wifi (ici partage de connexion)
const char* password = "12345678"; // 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 demarrer !");

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

  // création du fichier html
  client.println("HTTP/1.1 200 OK");
  client.println("Content-Type: text/html");
  client.println("");
  client.println("<!DOCTYPE HTML>");
  client.println("<html>");

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

  if(value == HIGH) {
    client.print("Oui");
  } else {
    client.print("Non");
  }
  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("</html>");

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

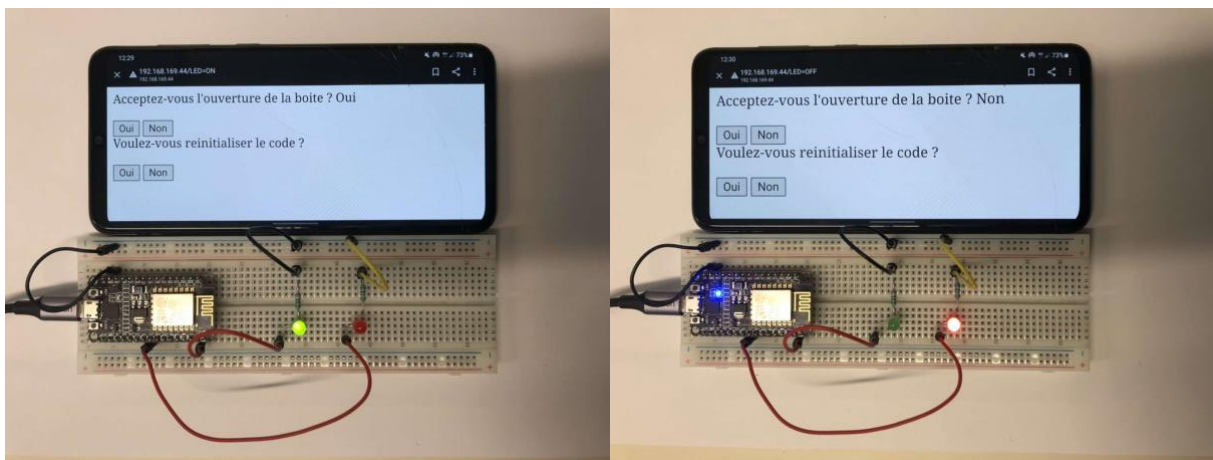
  client.println("<br><br>");
  client.println("<a href=\""/LED=ON\""/><button>Oui </button></a>");
  client.println("<a href=\""/LED=OFF\""/><button>Non </button></a><br />");
  client.println("</html>");

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

```



Ce code permet d'obtenir une adresse IP qui, lorsqu'on la saisit sur un navigateur internet (sur ordinateur, téléphone...), ouvre une page de type Html qu'on pourra par la suite modifier (avec un fichier CSS associé) afin d'obtenir une interface agréable et pratique pour l'utilisateur. Pour l'instant le code permet d'allumer (ou d'éteindre) une Led après le clic sur un bouton.



## Problèmes rencontrés :

Un des problèmes rencontrés fut le montage de la carte esp8266. En effet les pins utilisés dans le code n'ont pas les mêmes noms que ceux sur la carte (par exemple D0 sur la carte correspond au pin 16). La connexion se faisait également avec la carte via partage de connexion ce qui entraînait parfois des déconnexions entre la carte et le réseau wifi.

## Sites utilisés :

- <https://projetsdiy.fr/programmer-esp8266-ide-arduino-librairies-gpio-web-serveur-client/>  
(Lien qui nous a le plus servi)
- <https://projetsdiy.fr/esp8266-client-web-exemples-communication-tcp-ip-esp8266wifi-esp8266httpclient/>
- <https://ouilogique.com/pinouts/>

(les liens sont dans les commentaires du github)

