

# LUMOS

Here comes the programm

...



```
#include <Servo.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#ifndef APSSID
#define APSSID "Lumoss"
#define APPSK "o"
#endif
```

Annonce des librairies que  
l'on va utiliser.

```
Servo myservo;
```

Insertion du servo-moteur

```
int trig = 15;
int echo = 12;
long lecture_echo;
long cm;
int v;
int v1;
int v2;
bool etat = 0;
bool ultrason;
bool site_on = 0;
const char *ssid = APSSID;
const char *password = APPSK;
int pos;|
```

Définition du réseau ainsi  
que son mot de passe.

# Site Web :

```
<script>
function myFunction(){
  var result = document.getElementsByClassName("check")[0].checked ? 'yes' : 'no';
  console.log(result);

  // Get the checkbox
  var checkBox = document.getElementById("togBtn");
  // Get the output text
  var text = document.getElementById("text");

  // If the checkbox is checked, display the output text
  if (checkBox.checked == true){
    text.style.display = "block";
  } else {
    text.style.display = "none";
  }

  if (result == "yes"){
    console.log("on");
    window.open("http://192.168.4.1/On", "_blank");
  }

  if (result == "no"){
    console.log("off");
  }
}
</script>
<p id="text" style="display:none">Vous avez allumé la lumière !</p>

</body>
</html>
)=====;
```

• • •

- Ici on utilise du Javascript pour réaliser une action suite à la pression sur le bouton du site.
- 
- Au clic, cette fonction va afficher l'état "yes" ou "no" en fonction de l'état du bouton. Si le bouton est coché, le programme ouvre la page "On" qui va activer le servo-moteur dans un sens.
- 

A l'inverse, si l'on désactive le bouton, le programme renverra vers la page "Off" qui a pour but d'éteindre les lumières.

Vous l'aurez compris, dans le programme entier, il existe donc 3 pages web :  
La page On, Off et celle de lancement.

# Setup :

```
void setup(){
  pinMode(trig, OUTPUT);
  digitalWrite(trig, LOW);
  pinMode(echo, INPUT);
  pinMode(LED_BUILTIN, OUTPUT); // Initialize the LED_BUILTIN pin as an output
  delay(1000);
  Serial.begin(115200);
  Serial.println();
  Serial.print("Configuring access point...");
  /* You can remove the password parameter if you want the point d'accès
  WiFi.softAP(ssid, password);

  IPAddress myIP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(myIP);
  server.on("/On", SwitchOn);
  server.on("/Off", SwitchOff);
  server.on("/", handleRoot);
  server.begin();
  Serial.println("HTTP server started");
}
```

On initialise le capteur et la  
led de l'ESP 8266

On configure ensuite le  
point d'accès

Ici, on renseigne les différentes pages qui seront  
utilisées. Au lancement de ces pages, on lance  
différentes fonctions

# Loop :



Cette partie va vérifier dans un premier temps la présence de la main afin d'activer le servo-moteur à chaque changement d'état.

La variable "Site\_on" a pour vocation d'éviter les interférences entre les actions du site et les actions physiques.



```
void loop(){

  int angle;

  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  lecture_echo = pulseIn(echo,HIGH);
  cm = lecture_echo /58;
  delay(500);

  v = cm;
  v1 = v;
  v2 = v1;

  Serial.print ("Servo position :");
  Serial.println(pos);

  if (cm > 0 && v < 20 && v1 < 20 && v2 < 20 && pos >= 0) {
    ultrason = 1;
  }

  else {
    ultrason = 0;
  }

  if (ultrason && etat == 0) {
    etat = 1;
    myservo.attach(16);
    site_on = 0;
    delay(1000);
  }

  else if (ultrason && etat == 1) {
    etat = 0;
    myservo.attach(16);

    site_on = 0;
    delay(1000);
  }
}
```

# Loop :

```
if (etat == 0 && !site_on) {  
  
    for (pos = 25; pos <= 145; pos += 1) { // goes from 0 degrees to 180 degrees  
        // in steps of 1 degree  
        myservo.write(pos);           // tell servo to go to position in variable 'pos'  
        delay(15);                   // waits 15ms for the servo to reach the position  
    }  
    myservo.detach();  
  
}  
  
if (etat == 1 && !site_on) {  
  
    for (pos = 145; pos >= 25; pos -= 1) { // goes from 180 degrees to 0 degrees  
        myservo.write(pos);           // tell servo to go to position in variable 'pos'  
        delay(15);                   // waits 15ms for the servo to reach the position  
    }  
  
    myservo.detach();  
  
}  
  
Serial.println(etat);  
server.handleClient();  
}
```

• • •

Ici, si il y a un changement d'état et qu'il n'y a aucune activité sur le site, on effectue les rotations "On" ou "Off" puis on désactive le servo-moteur afin qu'il ne consomme pas d'électricité inutilement.

•  
•  
•  
•  
•  
•  
•  
•

# Voids :

```
ESP8266WebServer server(80);

void handleRoot() {
  server.send(200, "text/html", index_html);
  Serial.println("bravo");
  digitalWrite(LED_BUILTIN, LOW); // Turn the LED on (Note that LOW is the voltage level
  digitalWrite(LED_BUILTIN, HIGH); // Turn the LED off by making the voltage HIGH4
}

void SwitchOn() {
  myservo.attach(16);
  site_on = 1;
  Serial.println("Allumé");
  server.send(200, "text/html", index_html_checked);

  for (pos = 25; pos <= 145; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos); // tell servo to go to position in variable 'pos'
    delay(15); // waits 15ms for the servo to reach the position
  }
  myservo.detach();
}

void SwitchOff() {
  myservo.attach(16);0.
  site_on = 1;
  Serial.println("éteint");
  server.send(200, "text/html", index_html);

  for (pos = 145; pos >= 25; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos); // tell servo to go to position in variable 'pos'
    delay(15); // waits 15ms for the servo to reach the position
  }
  myservo.detach();
}
```

- On retrouve ici les différentes fonctions utilisées par le programme.

- La fonction Handleroot sert tout simplement à envoyer la première page au serveur et à afficher un message, ainsi qu'à allumer la led de l'ESP à chaque connexion.

Les fonctions "switchOn" et "switchOff" effectuent la même action que si l'on passe la main sur le capteur. Elles sont lancées à la pression du bouton.

...

# **Vous voilà prêt !**

Tout fonctionne? Parfait ! Vous en voulez plus?  
Nous continuons le développement de ce beau  
projet. A très vite !

.....