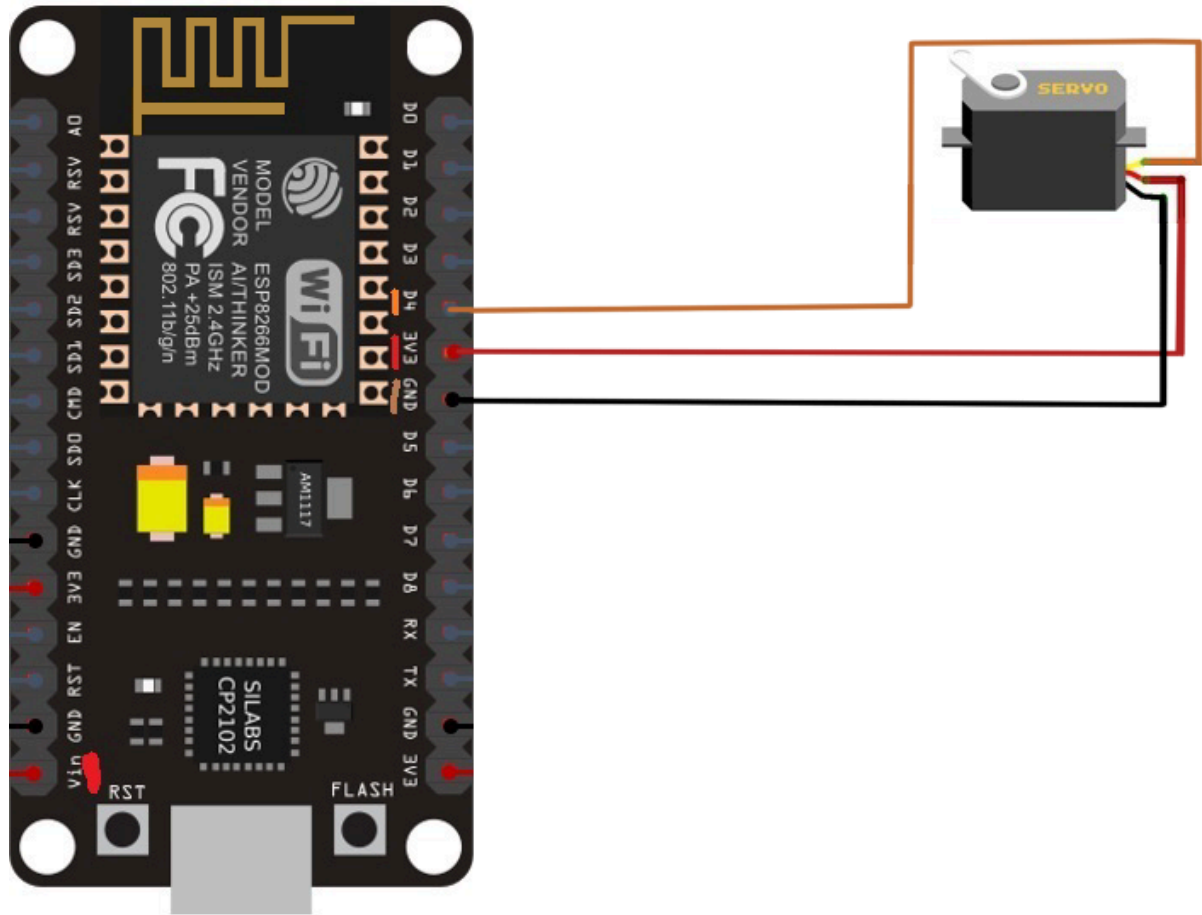
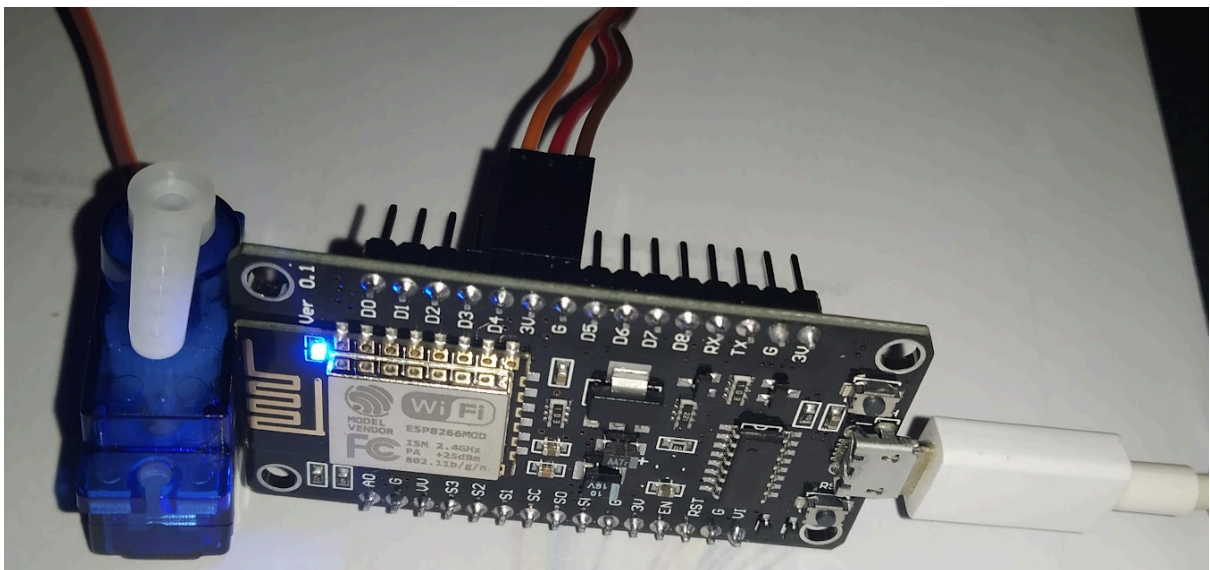


Para esta actividad se debía desarrollar un código que presente una página con dos botones: Giro Horario y Giro Antihorario, de modo que al presionar cada uno de ellos, el servomotor gire un ángulo a elección en el sentido de giro indicado.

diagrama de la implementación:



Fotografía del diagrama implementado:



La solución se desarrollará en base al archivo "testLED_Base.ino" visto en clase.

Las partes implementadas se resaltan en los rectángulos rojos:

- Primero debemos incluir la librería "**Servo.h**", para poder controlar en servomotor.
- Luego asignamos en pin D4 al servo, junto al ancho de pulso mínimo para 0° y el máximo para 180°, en este caso 500 y 2400 respectivamente.
- Como tercer paso (es opcional) se envía el valor 90 al servo para que se posicione a 90° al iniciar el Arduino, esto se utiliza únicamente para demostrar que se reinició el sistema.

```
cervoweb
#include <ESP8266WiFi.h>
#include <Servo.h>

String ssid    = "alfacharly";
String password = "H90zGM364Md";
WiFiServer server(80); //objeto de la clase WiFiServer
int estado = 0;
Servo servoMotor;

void setup() {
    // Inicia Serial
    Serial.begin(115200);
    Serial.println("\n");

    servoMotor.attach(D4, 500, 2400);
    // Conexión WIFI
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("*****");
    Serial.print("Conectado a la red WiFi: ");
    Serial.println(WiFi.SSID());
    Serial.print("IP: ");
    Serial.println(WiFi.localIP());
    Serial.print("macAdress: ");
    Serial.println(WiFi.macAddress());
    Serial.println("*****");

    server.begin(); //begin() levantamos el servidor
    servoMotor.write(90);
}
```

- Dentro del loop establecemos condiciones para guardar en la variable “estado” un valor correspondiente a la información que llega desde la página web; Ésta se utilizará para enviar al servo el valor del ángulo al que debe girar y también se imprimirá el ángulo en el monitor serie (0° y 180° para este caso).
- Debajo se envía al servo motor el valor cargado en la variable “estado”.

```
cervoweb
void loop() {

  WiFiClient client = server.available(); //objeto de la clase WiFiClient
  // available() detecta un cliente nuevo del objeto de la clase WifiServer
  if(!client){
    return;
  }

  Serial.println("Nuevo cliente...");
  while(!client.available()){ //espera a un cliente disponible
    delay(1);
  }

  String peticion = client.readStringUntil('\r'); //lee la peticion del cliente
  Serial.println(peticion);
  client.flush(); //limpia la peticion del cliente

  if(peticion.indexOf("Servo=180") != -1)
  {
    estado=180;
    Serial.print("Angulo: ");
    Serial.println(estado);
  }
  if(peticion.indexOf("Servo=0") != -1)
  {
    estado=0;
    Serial.print("Angulo: ");
    Serial.println(estado);
  }

  servoMotor.write(estado);

  client.println("HTTP/1.1 200 OK");
  client.println("");
  client.println("");
  client.println("");
  client.println("");
}
```

Para finalizar tenemos el código de la página:

- En el primer resaltado se encuentra el título.
- Debajo se declaran las condiciones para mostrar un texto que corresponda a la posición del servomotor.
- Por último se implementan los botones que corresponden a cada giro (Horario/Antihorario) y la información que debe retornar como petición al oprimirlos ("/servo=0" y "/servo=180").

cervoweb

```
//INICIA LA PAGINA

client.println("<!DOCTYPE html><html lang='en'><head><meta charset='UTF-8'>");
client.println("<meta name='viewport' content='width=device-width, initial-scale=1.0'>");
client.println("<title>ServoMotor</title></head>");
client.println("<body style='font-family: Century gothic; width: 800;'><center>");
client.println("<div style='box-shadow: 0px 0px 20px 8px rgba(0,0,0,0.22); padding: 20px; w");
client.println("<h1>Servo 1</h1>");

if(estado==0)
  client.println("<h2>El servo motor está a la derecha</h2>");
else
  client.println("<h2>El servo motor está a la izquierda</h2>");

client.println("<button style='background-color:red; color:white; border-radius: 10px; bor");
client.println("type='button' onClick=location.href='/Servo=0'><h2>GiroHorario</h2>");
client.println("</button> <button style='background-color:blue; color:white; border-radius:");
client.println("type='button' onClick=location.href='/Servo=180'><h2>GiroAntiHorario</h2>");
client.println("</button></div></center></body></html>");

//FIN DE LA PAGINA

delay(20);
Serial.println("Petición finalizada");
Serial.println("");
}
```

Monitor serie:

Se puede apreciar que muestra la información de la conexión a la que debemos acceder y al realizar interacciones muestra el dato del ángulo en el que se encuentra el servomotor tras la última petición.

```
.....  
*****  
Conectado a la red WiFi: alfacharly  
IP: 192.168.1.11  
macAdress: C4:5B:BE:56:67:82  
*****  
Nuevo cliente...  
GET / HTTP/1.1  
Petición finalizada  
  
Nuevo cliente...  
GET /favicon.ico HTTP/1.1  
Petición finalizada  
  
Nuevo cliente...  
GET /Servo=180 HTTP/1.1  
Angulo: 180  
Petición finalizada  
  
Nuevo cliente...  
GET /favicon.ico HTTP/1.1  
Petición finalizada  
  
Nuevo cliente...  
GET /Servo=0 HTTP/1.1  
Angulo: 0  
Petición finalizada  
  
Nuevo cliente...  
GET /favicon.ico HTTP/1.1  
Petición finalizada
```

☒ Autoscroll ☐ Mostrar marca temporal Nueva línea 115200 baudio Limpiar salida

Página web:

Al entrar en la página el servomotor se sitúa a la derecha ya que la variable utilizada (estado) se declara "igual a cero".



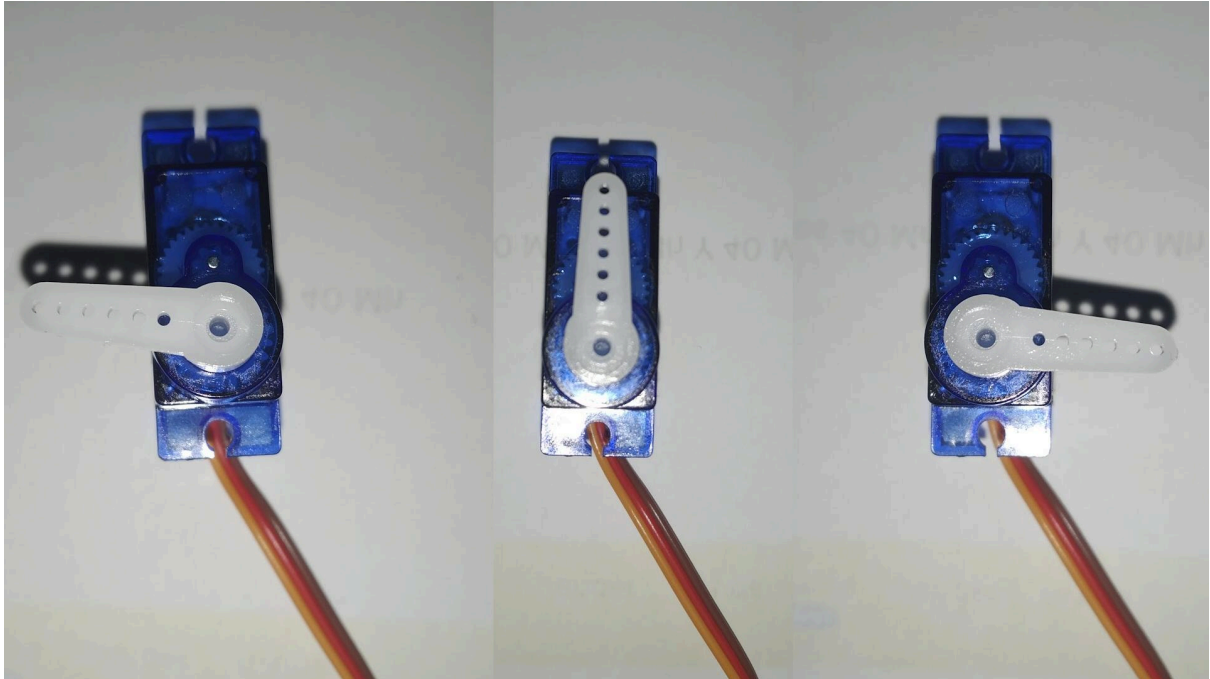
Al oprimir en "GiroAntiHorario" se posiciona a la izquierda.



Al oprimir en “GiroHorario” se posiciona a la derecha.



Demostración fotográfica de los 3 estados que permite la implementación, a 90° cuando inicia, 0° y 180° al oprimir algunos de los botones de la interfaz web.



Video demostrativo:

<https://youtu.be/zd2JTL7NRyM>