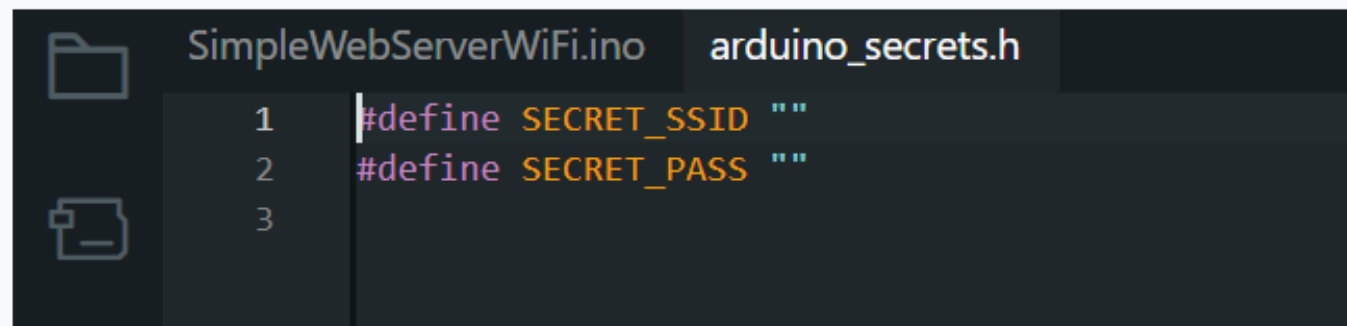


EJEMPLO WIFI DE ARDUINO - encendido y apagado del led 13

Ahora tendremos que completar el nombre de la red wifi y la contraseña de acceso en arduino_secrets.h

NO SE PUEDE USAR WEDU_PROF, en su lugar podemos usar nuestro móvil como hotspot. Completa los campos de la imagen con la información de tu hotspot:



```
SimpleWebServerWiFi.ino  arduino_secrets.h
1  #define SECRET_SSID ""
2  #define SECRET_PASS ""
3
```

Conectamos un led al pin 13.

Cargamos el programa en la Arduino y vemos en el puerto serie la dirección IP a la que tendremos que conectarnos (usa el navegador) para poder encender y apagar el led.

TAREA PROPUESTA: Modifica el programa anterior para controlar un segundo LED en el pin 5.

Mini Servo 360°

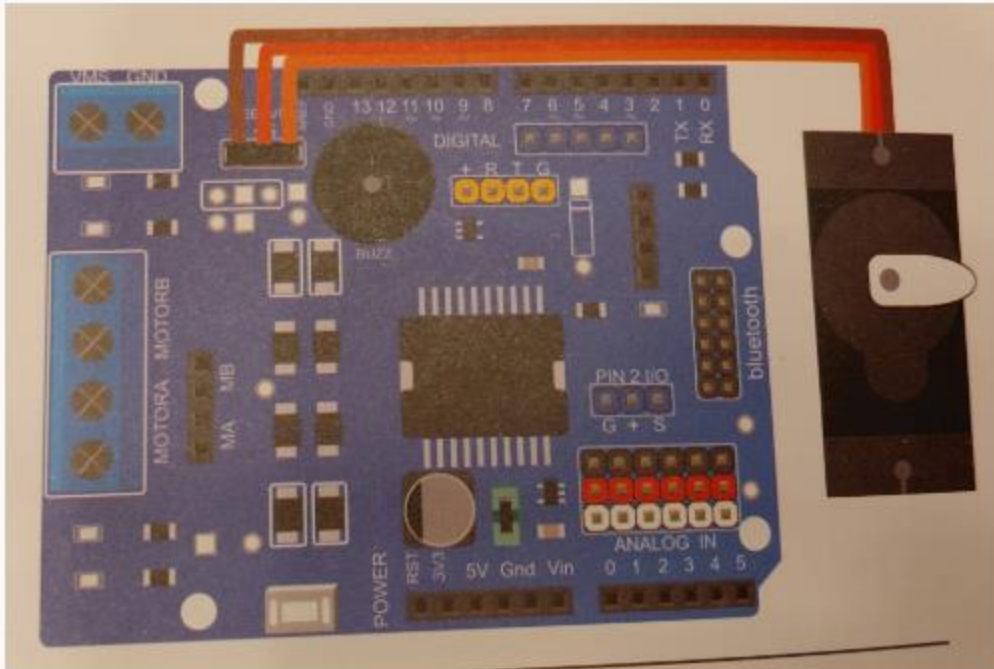
Un mini servo 360° puede girar de manera continua en ambas direcciones.

Con la instrucción `servo360.write()`; se indica al servo en qué sentido y con que velocidad girar:

- `servo360.write(0);` //máxima velocidad en un sentido
- `servo360.write(180);` // máxima velocidad en sentido opuesto al anterior
- **`servo360.write(90);` //punto neutro, parada. Algunos server tienen deriva. Para averiguar el punto neutro usar el programa `calibrador360.ino`**

CONEXIONADO

Vamos a conectar el mini servo 360° a los pines específicos de la placa de expansión (corresponden al pin 9 digital):



IMPORTANTE: la placa de expansión debe estar alimentada con 5 pilas AA (portapilas en caja de kit de componentes)

Mini Servo 360°

Programa Calibrador

```
#include <Servo.h>

Servo miServo;

const int PIN_SERVO = 9;
const int GRADO_MIN = 0;
const int GRADO_MAX = 180;
const int GRADO_STOP = 90; // Punto de partida (neutro teórico)

int valorActual = GRADO_STOP;

void setup() {
  Serial.begin(9600);
  while (!Serial) {}

  miServo.attach(PIN_SERVO);
  miServo.write(valorActual);

  Serial.println("=====");
  Serial.println("  Calibrador de Servo 360° (grados)      ");
  Serial.println("  Arduino UNO R4 WiFi                      ");
  Serial.println("=====");
  Serial.println("Escribe un valor de 0 a 180 y pulsa Enter");
  Serial.println(" + / - → ajuste fino ±1°");
  Serial.println(" > / < → ajuste rápido ±5°");
  Serial.println(" s      → ver valor actual");
  Serial.println(" r      → reset a 90°");
  Serial.println("-----");
  imprimirValor();
}
```

```
void loop() {
  if (Serial.available() > 0) {
    String entrada = Serial.readStringUntil('\n');
    entrada.trim();

    if (entrada.length() == 0) return;

    if (entrada.length() == 1) {
      char cmd = entrada.charAt(0);

      if (cmd == 's' || cmd == 'S') {
        imprimirValor();
        return;
      }
      if (cmd == 'r' || cmd == 'R') {
        valorActual = GRADO_STOP;
        aplicarValor();
        Serial.println(">> Reset a 90°");
        return;
      }
      if (cmd == '+') {
        valorActual = constrain(valorActual + 1, GRADO_MIN, GRADO_MAX);
        aplicarValor();
        return;
      }
      if (cmd == '-') {
        valorActual = constrain(valorActual - 1, GRADO_MIN, GRADO_MAX);
        aplicarValor();
        return;
      }
    }
  }
}
```

Mini Servo 360°

```
if (cmd == '>') {
    valorActual = constrain(valorActual + 5, GRADO_MIN, GRADO_MAX);
    aplicarValor();
    return;
}
if (cmd == '<') {
    valorActual = constrain(valorActual - 5, GRADO_MIN, GRADO_MAX);
    aplicarValor();
    return;
}
}
```

```
// Intenta parsear como número entero
bool esNumero = true;
for (int i = 0; i < entrada.length(); i++) {
    if (!isDigit(entrada.charAt(i))) {
        esNumero = false;
        break;
    }
}
```

```
if (esNumero) {
    int nuevo = entrada.toInt();
    if (nuevo >= GRADO_MIN && nuevo <= GRADO_MAX) {
        valorActual = nuevo;
        aplicarValor();
    } else {
        Serial.print(">> ERROR: Introduce un valor entre ");
        Serial.print(GRADO_MIN);
        Serial.print("° y ");
        Serial.print(GRADO_MAX);
        Serial.println("°");
    }
} else {
    Serial.println(">> Comando no reconocido. Usa un número o: + - > < s r");
}
}

void aplicarValor() {
    miServo.write(valorActual);
    imprimirValor();
}

void imprimirValor() {
    Serial.print(">> Valor actual: ");
    Serial.print(valorActual);
    Serial.print("°");

    if (valorActual < 88) Serial.print(" [girando <-- izquierda]");
    else if (valorActual > 92) Serial.print(" [girando --> derecha]");
    else Serial.print(" [zona de parada ~]");
    Serial.println();
}
```

```
//MOVER SERVO 360° - SERVO DE ROTACIÓN CONTINUA
#include <Servo.h> //los servos necesitan esta librería

Servo servoSAG; //le doy nombre a mi servo

void setup() {
  // put your setup code here, to run once:
  servoSAG.attach(9); //servo 360° conectado a pin 9 (con virgulilla para poder controlar potencia)
}

void loop() {
  // put your main code here, to run repeatedly:
  servoSAG.write(180); //gira en un sentido
  delay(1000);
  servoSAG.write(90); //se detiene. 90° es el neutro teórico, a veces, en vez de 90° tienen una pequeña deriva. Usar calibrado360.ino para averiguar el neutro.
  delay(1000);
  servoSAG.write(0); //gira en sentido opuesto
  delay(1000);
  servoSAG.write(90); //se detiene
  delay(1000);
}
```

Tarea propuesta: controlar por WiFi a través del móvil y una página web el Servo (sería como una persiana) y varias luces LEDs