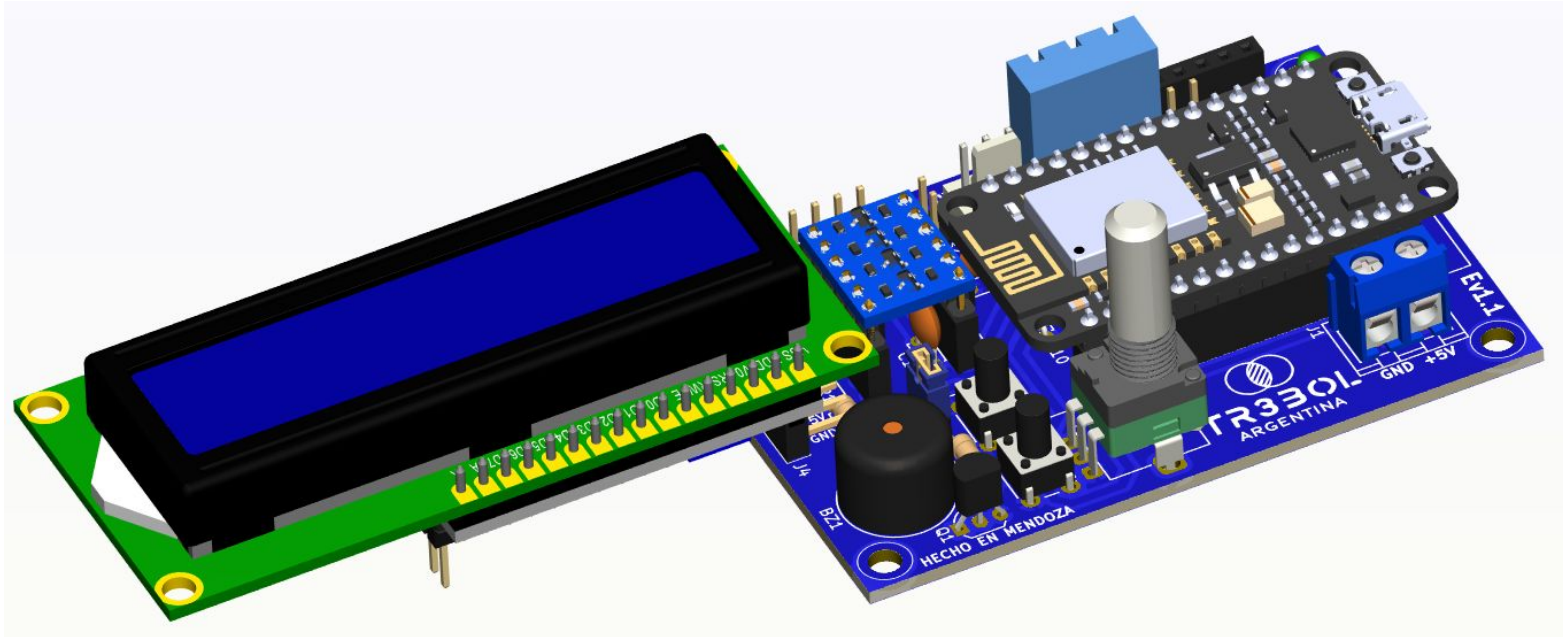


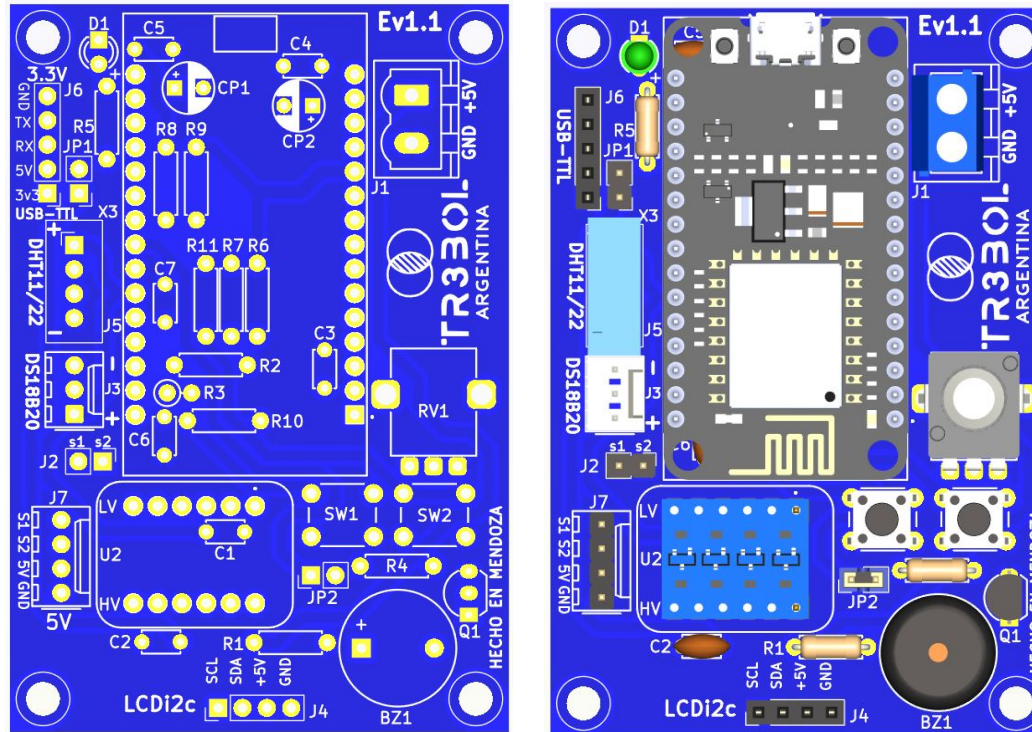
Clase I - Módulo B

Programación Placa Entrenadora TR3BOL



Placa Entrenadora TR3BOL





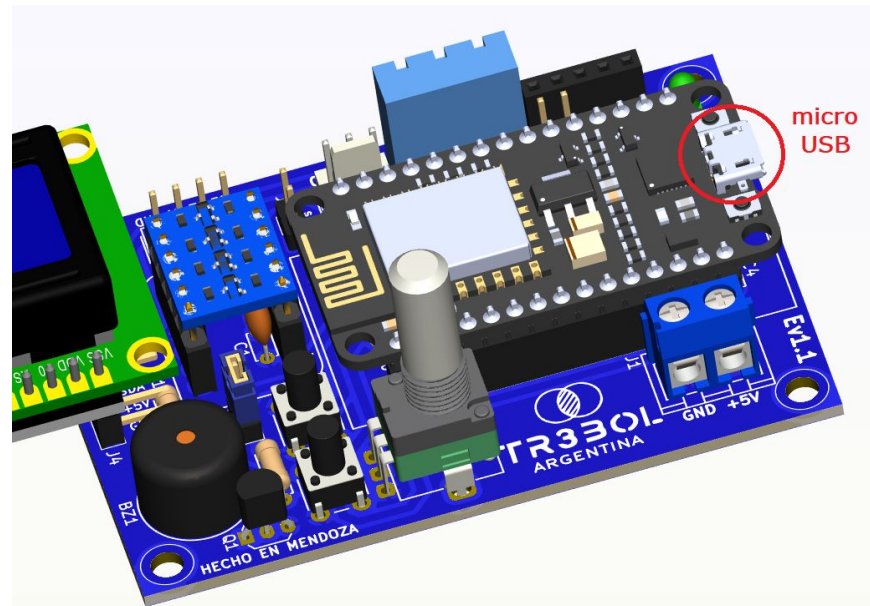
Nos preparamos para la práctica

- Abre el entorno de programación ARDUINO IDE.
- Conecta la placa entrenadora (por ahora sin display ni sonda de T°) a tu computadora.
- Recomendación: presta atención en vivo y luego practica con el video de la clase!

Conecta ESP8266

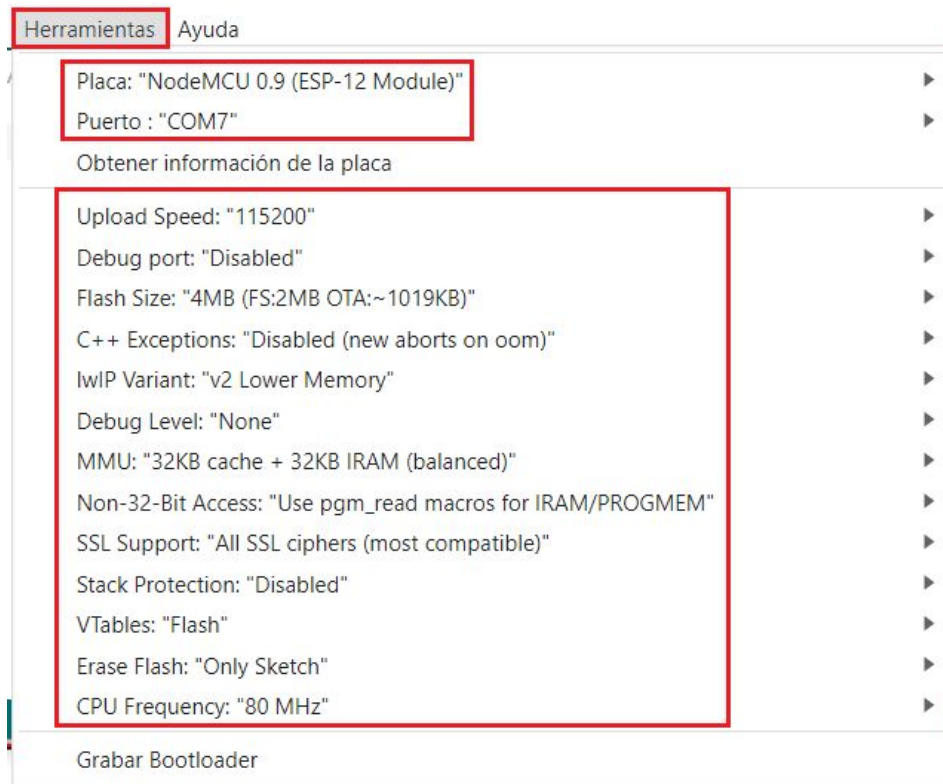
Conecta el **NodeMCU ESP8266** de tu placa de Desarrollo TR3BOL a tu PC, mediante el **cable USB** provisto.

Verifica el **puerto COM** en Administrador de Dispositivos. Si tienes dudas, ve al documento *1) Instalación y Configuración Arduino IDE para ESP8266*.



Configura ESP8266

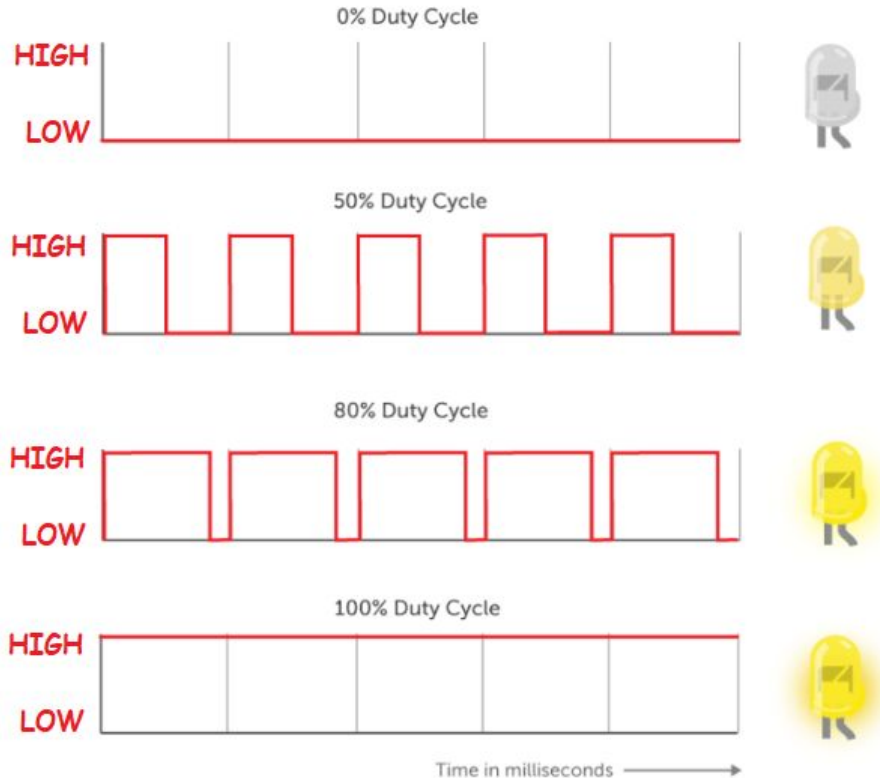
Abre Arduino IDE y configura
tu placa en Herramientas



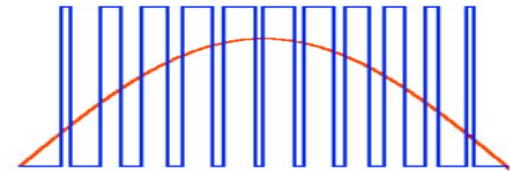
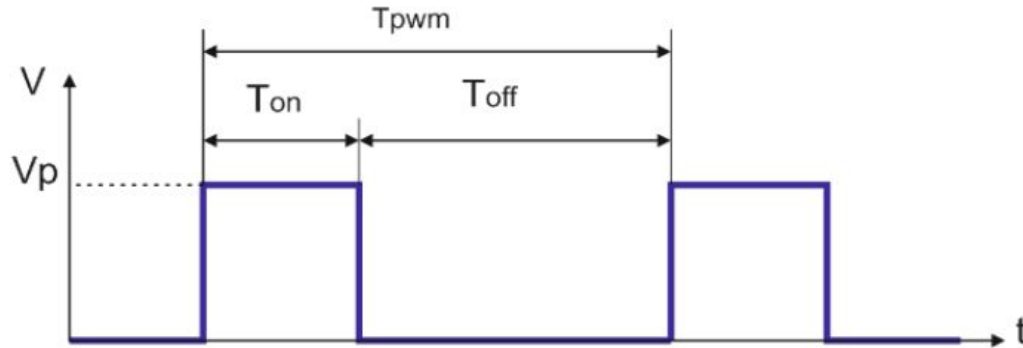
ON/OFF



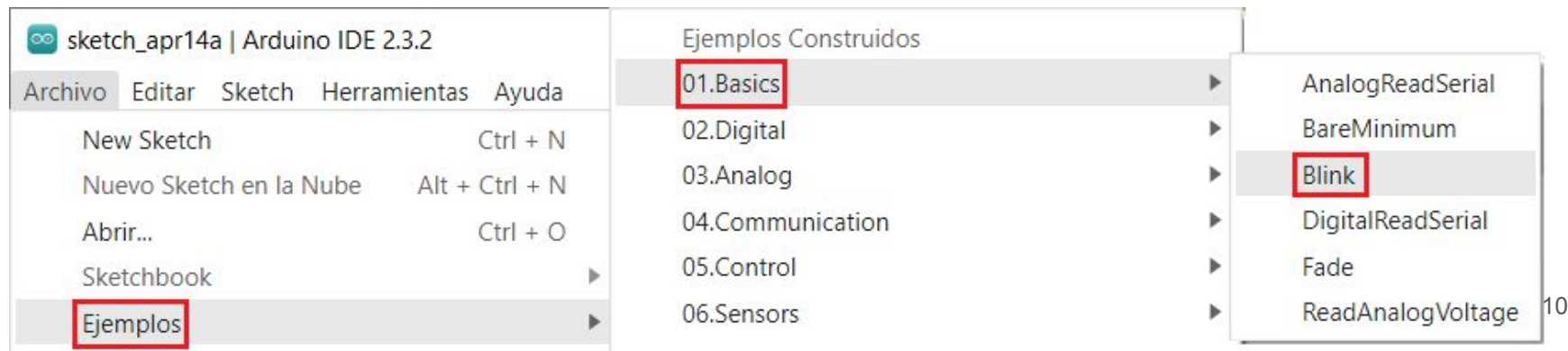
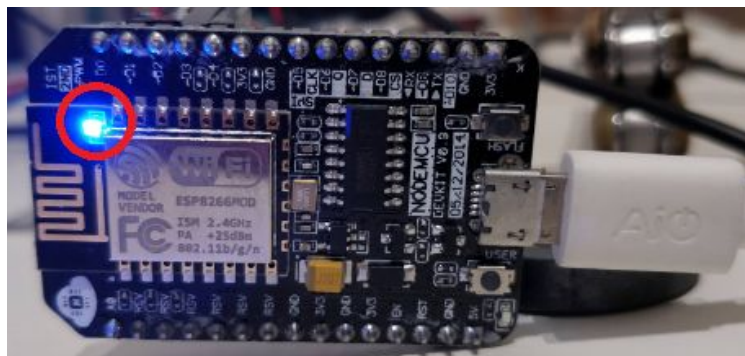
PWM



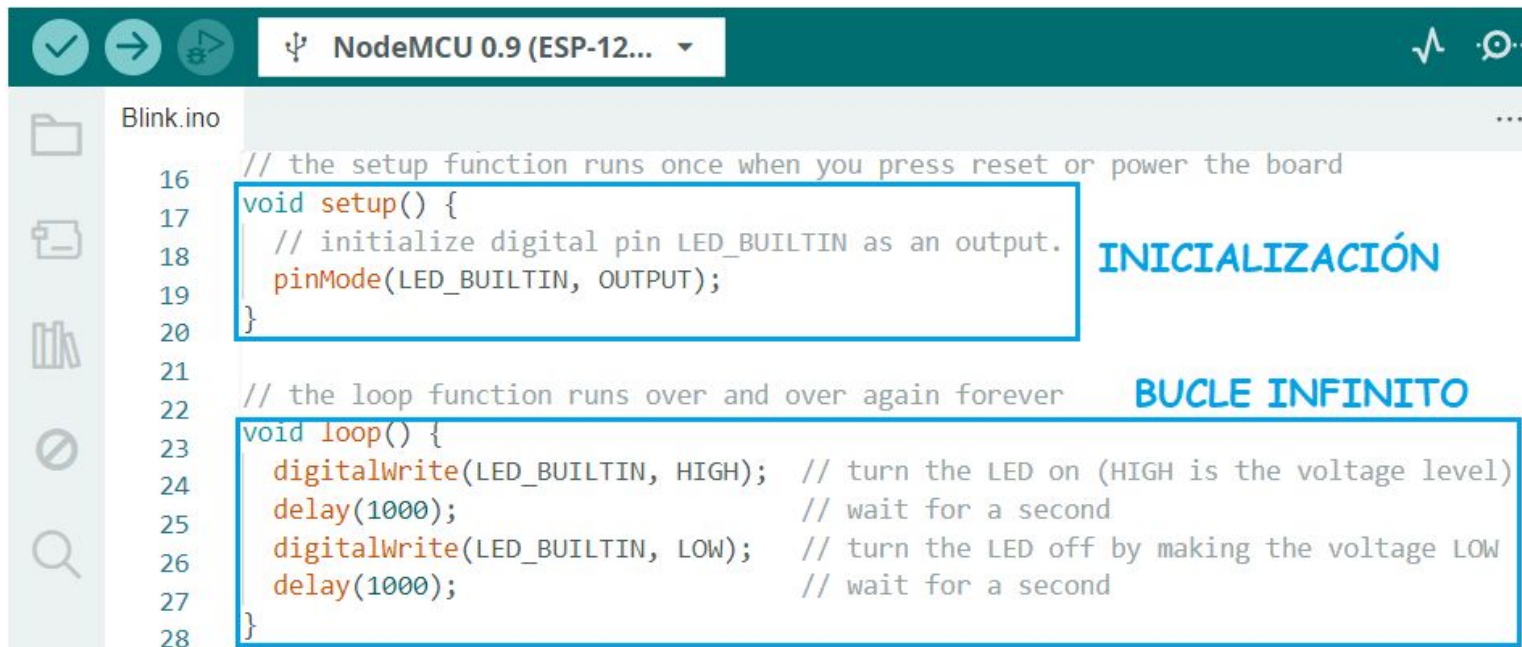
PWM



Ejemplo Blink ESP8266: parpadeo del LED embebido (ON/OFF)



Ejemplo Blink ESP8266: parpadeo del LED embebido (ON/OFF)



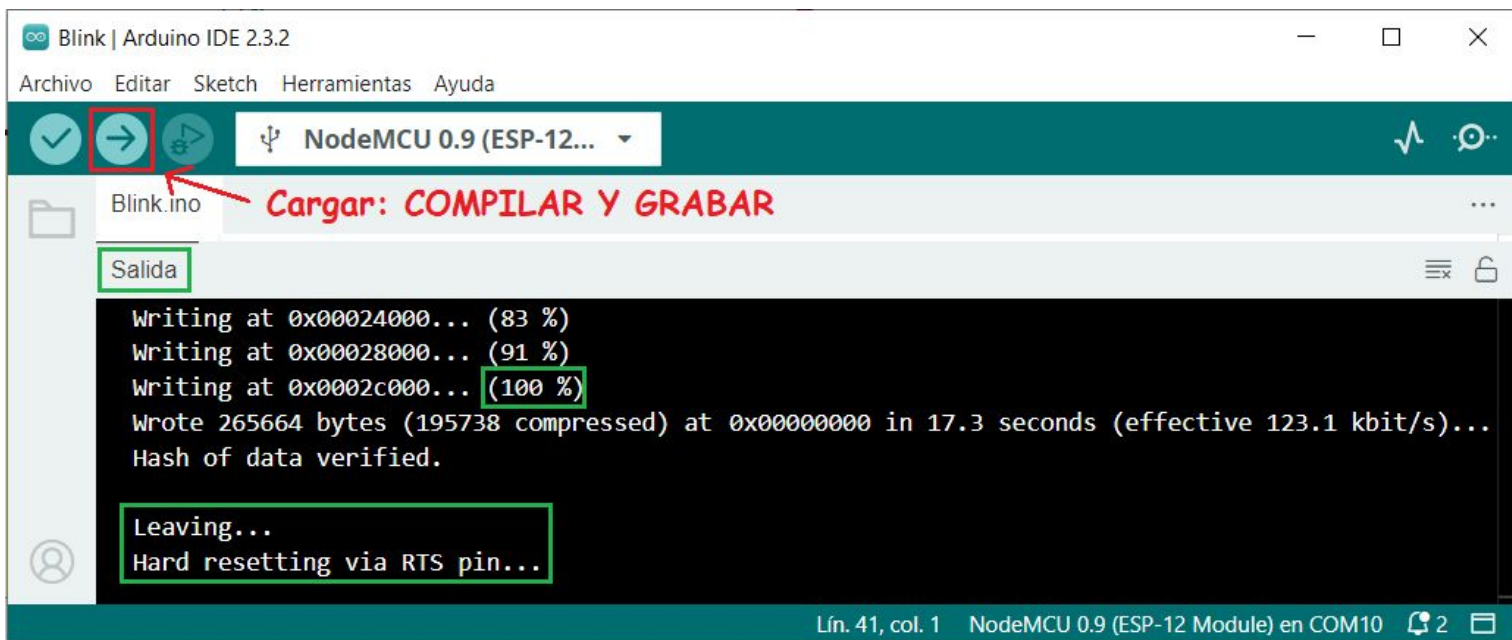
```

16 // the setup function runs once when you press reset or power the board
17 void setup() {
18     // initialize digital pin LED_BUILTIN as an output.
19     pinMode(LED_BUILTIN, OUTPUT);
20 }
21
22 // the loop function runs over and over again forever
23 void loop() {
24     digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
25     delay(1000); // wait for a second
26     digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
27     delay(1000); // wait for a second
28 }
    
```

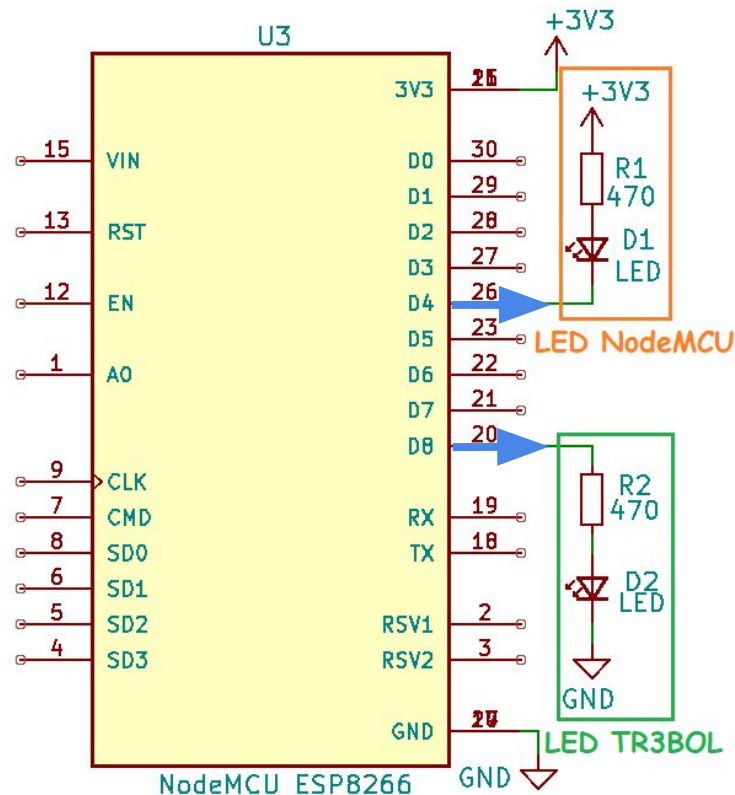
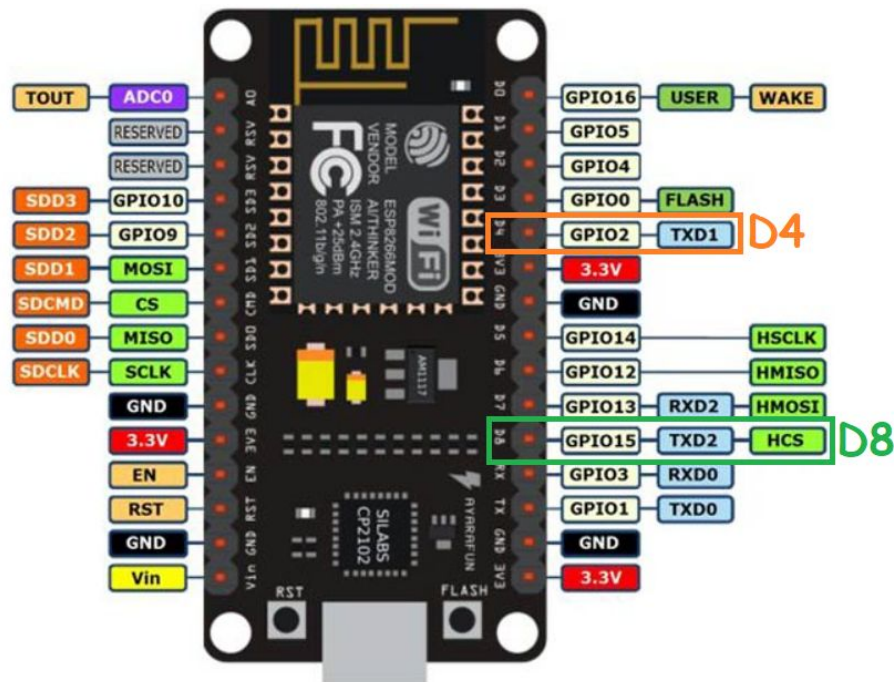
INICIALIZACIÓN

BUCLE INFINITO

Ejemplo Blink ESP8266: parpadeo del LED embebido (ON/OFF)



Ejemplo Blink TR3BOL (ON/OFF)

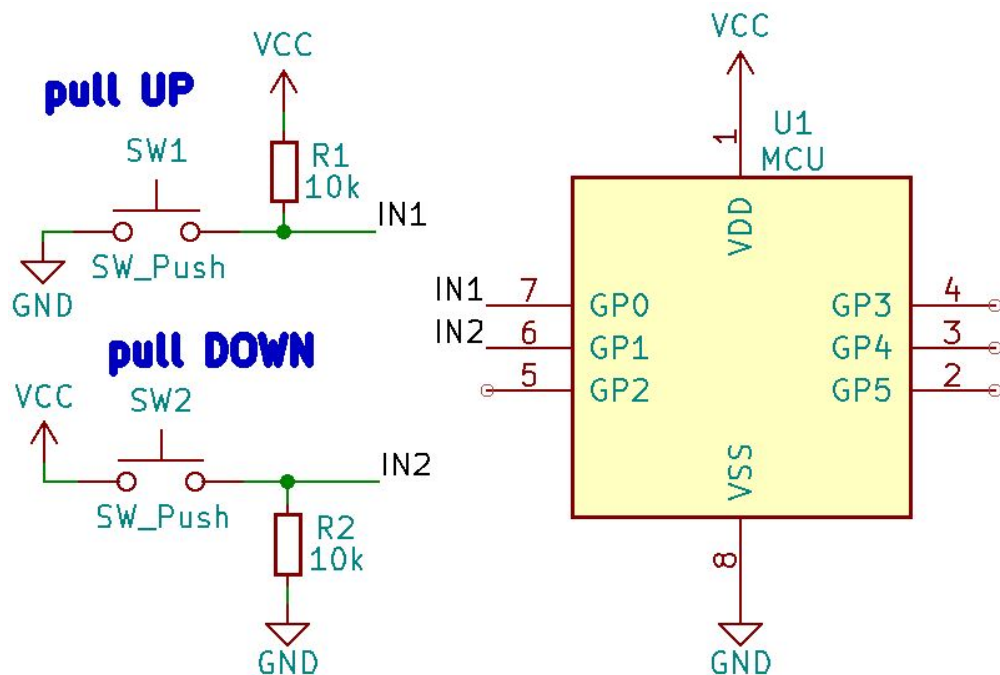


Ejemplo Blink TR3BOL (ON/OFF)

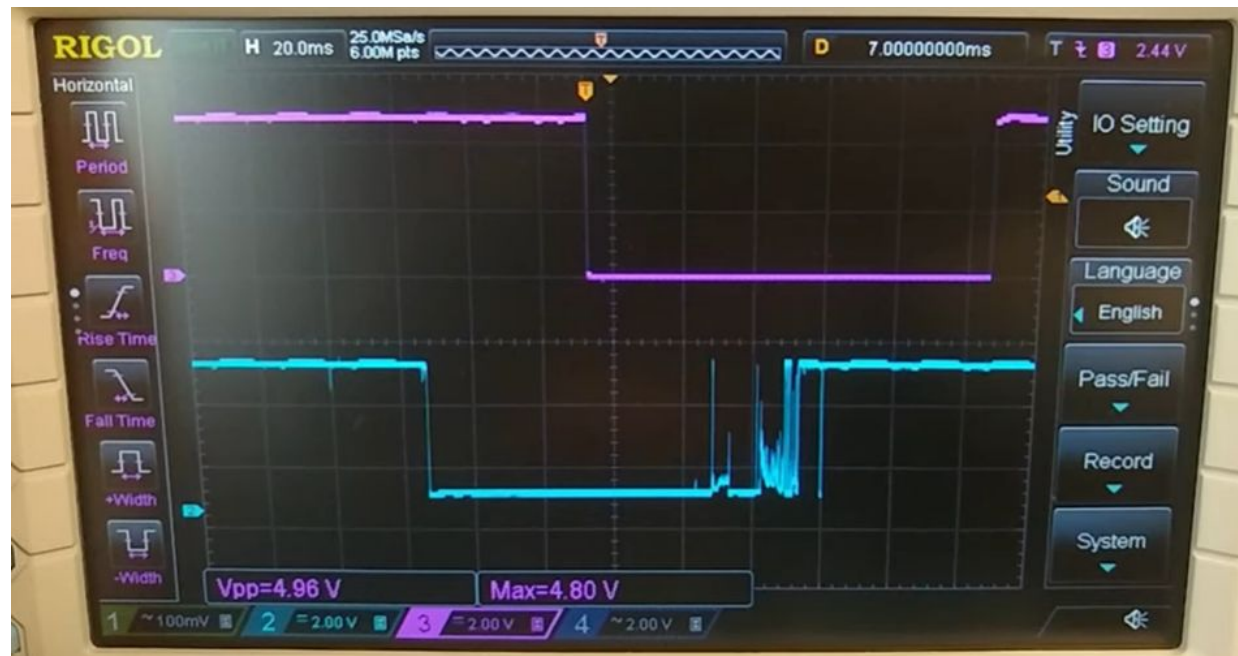
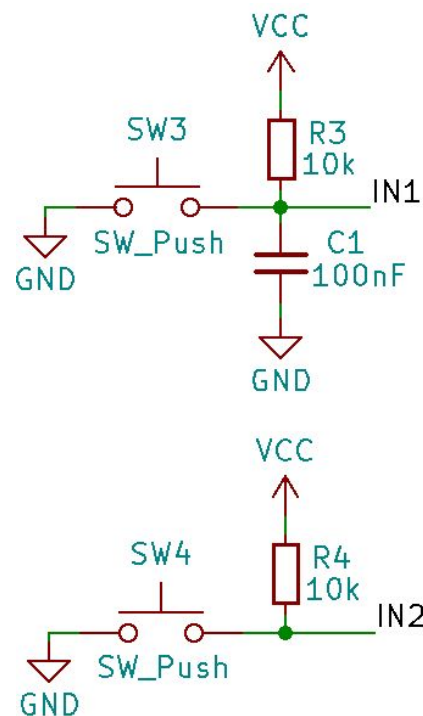
Blink.ino

```
24 #define LED_TR3BOL D8
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27     // initialize digital pin LED_BUILTIN as an output.
28     pinMode(LED_BUILTIN, OUTPUT);
29     pinMode(LED_TR3BOL, OUTPUT);
30 }
31
32 // the loop function runs over and over again forever
33 void loop() {
34     digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
35     digitalWrite(LED_TR3BOL, HIGH); // turn the LED on (HIGH is the voltage level)
36     delay(200); // wait
37     digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
38     digitalWrite(LED_TR3BOL, LOW); // turn the LED on (HIGH is the voltage level)
39     delay(200); // wait
40 }
```

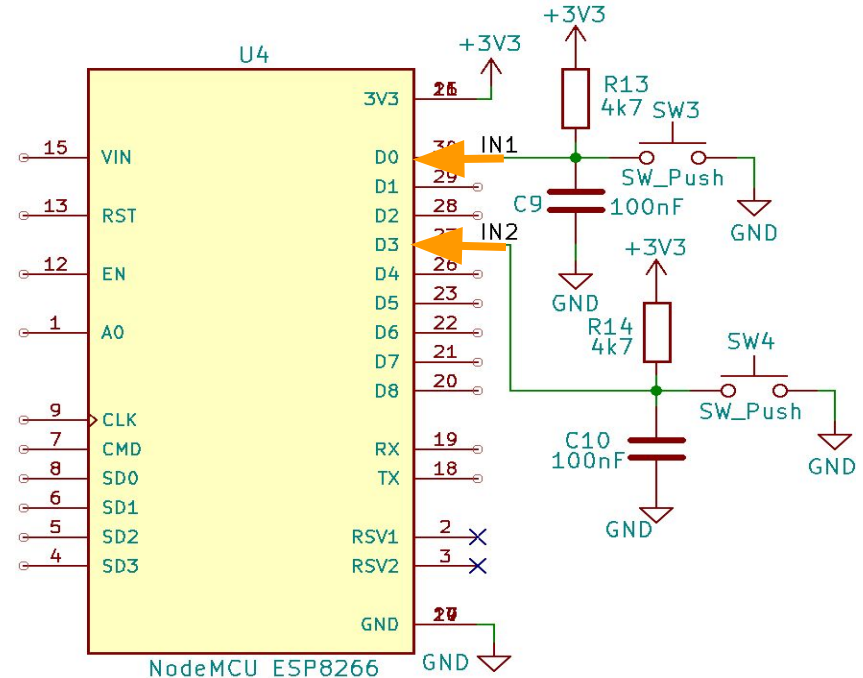
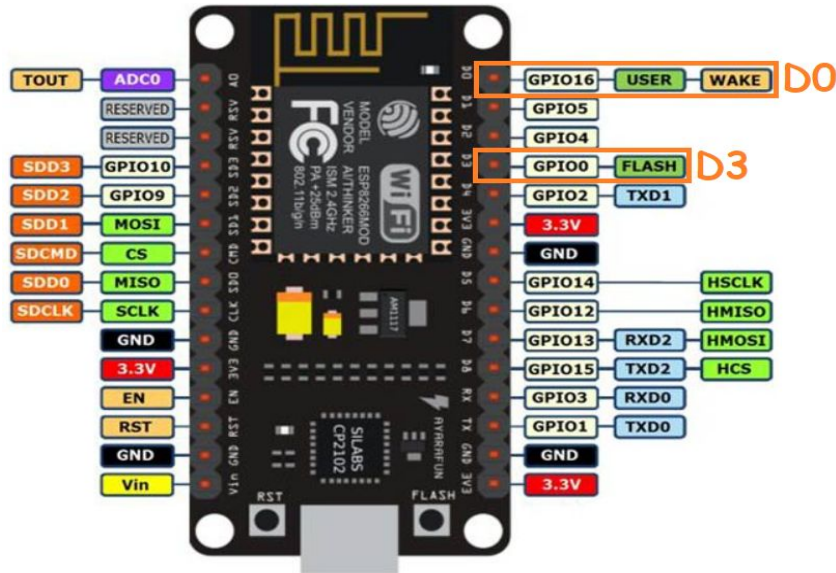
Pulsadores y conexionado pull-Up y pull-Down



Rebote mecánico de pulsadores



Tenemos dos Tact Switch en nuestra placa TR3BOL, en D0 y D3
Conexionado pull Up con filtro pasa-bajo



Ejemplo lectura Pulsador 1 → activación LED interno

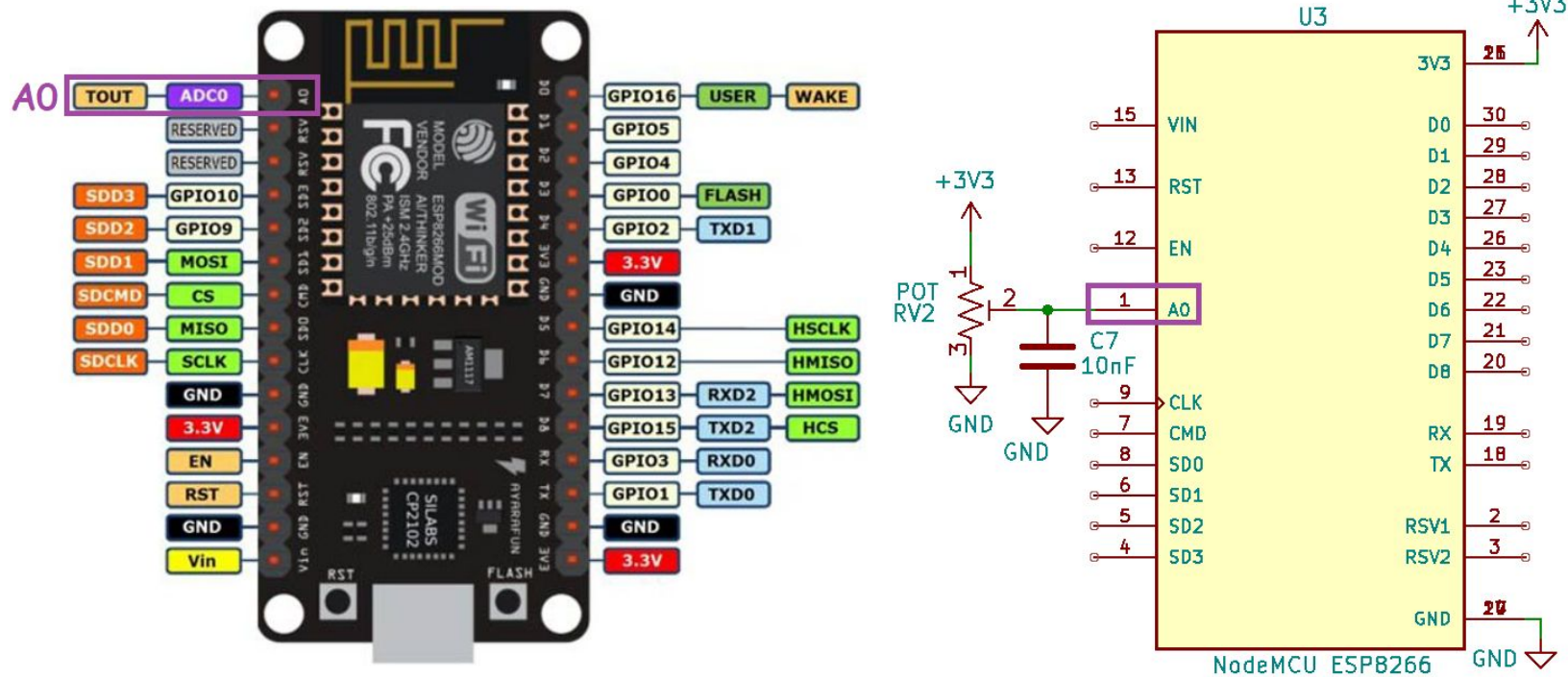
Blink.ino

```
24 #define PULSADOR_1 D0
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27     // initialize digital pin LED_BUILTIN as an output.
28     pinMode(LED_BUILTIN, OUTPUT);
29     pinMode(PULSADOR_1, INPUT);
30 }
31
32 // the loop function runs over and over again forever
33 void loop() {
34     if(digitalRead(PULSADOR_1)==HIGH){
35         digitalWrite(LED_BUILTIN, HIGH);
36     }
37     else{
38         digitalWrite(LED_BUILTIN, LOW);
39     }
40 }
```

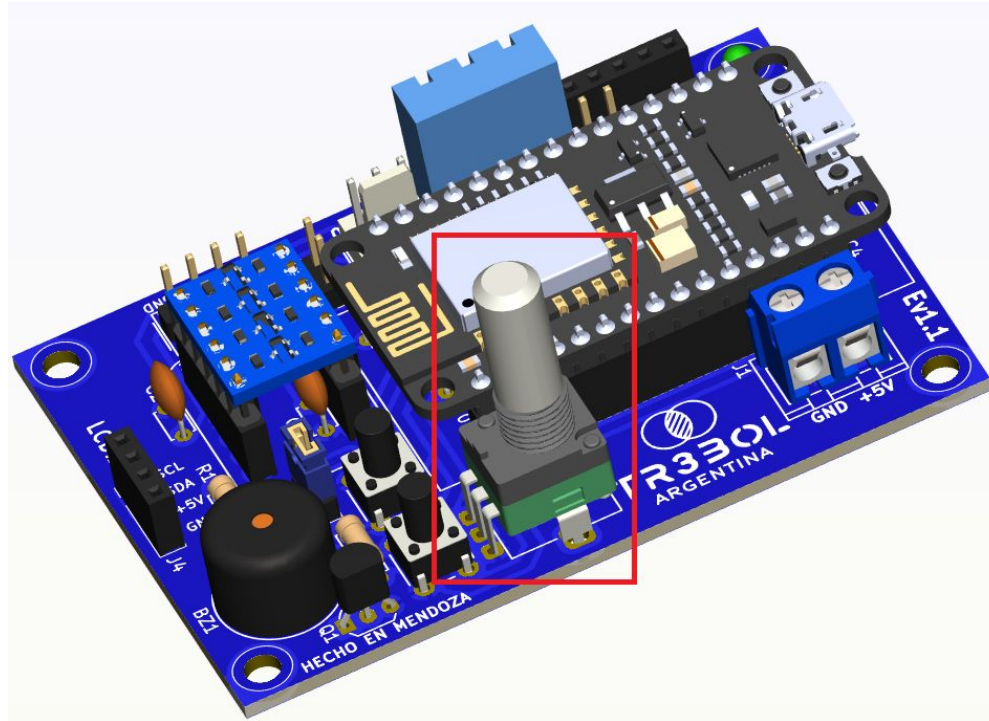
Ejemplo lectura Pulsadores 1 y 2 → activación LEDs interno y externo

```
Blink.ino
24 #define LED_TR3BOL D8
25 #define PULSADOR_1 D0
26 #define PULSADOR_2 D5
27 // the setup function runs once when you press reset or power the board
28 void setup() {
29     // initialize digital pin LED_BUILTIN as an output.
30     pinMode(LED_BUILTIN, OUTPUT);
31     pinMode(LED_TR3BOL, OUTPUT);
32     pinMode(PULSADOR_1, INPUT);
33     pinMode(PULSADOR_2, INPUT);
34 }
35
36 // the loop function runs over and over again forever
37 void loop() {
38     if(digitalRead(PULSADOR_1)==HIGH)    digitalWrite(LED_BUILTIN, HIGH);
39     else                                  digitalWrite(LED_BUILTIN, LOW);
40     if(digitalRead(PULSADOR_2)==LOW)     digitalWrite(LED_TR3BOL, HIGH);
41     else                                  digitalWrite(LED_TR3BOL, LOW);
42 }
```

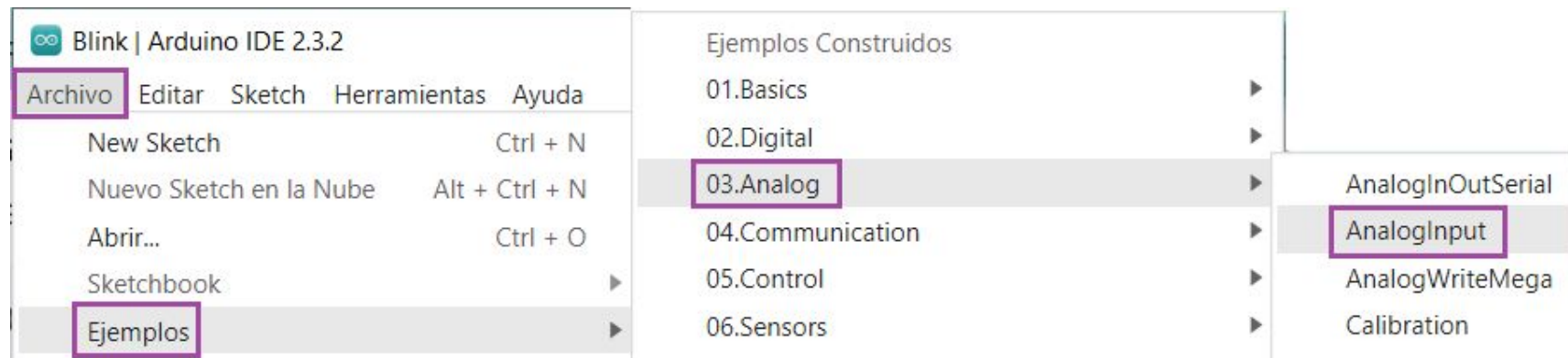

El ESP8266 tiene una sola entrada analógica A0



Ejemplo Arduino IDE AnalogInput



Ejemplo Arduino IDE AnalogInput



Ejemplo Arduino IDE AnalogInput

```
int sensorPin = A0;    // select the input pin for the potentiometer
int sensorValue = 0;  // variable to store the value coming from the sensor
#define ledPin D4      // select the pin for the LED

void setup() {
    pinMode(ledPin, OUTPUT);
}

void loop() {
    // read the value from the sensor:
    sensorValue = analogRead(sensorPin);
    digitalWrite(ledPin, HIGH);
    delay(sensorValue);
    digitalWrite(ledPin, LOW);
    delay(sensorValue);
}
```

Antes de grabar... ¿Cómo se comporta el programa?

WiFi Scan

Blink Arduino 1.8.9

Archivo Editar Programa Herramientas Ayuda

- Nuevo Ctrl+N
- Abrir... Ctrl+O
- Abrir Reciente >
- Proyecto >
- Ejemplos** >
- Cerrar Ctrl+W
- Salvar Ctrl+S
- Guardar Como... Ctrl+Mayús+S
- Configurar Página Ctrl+Mayús+P
- Imprimir Ctrl+P
- Preferencias Ctrl+Coma
- Salir Ctrl+Q

Ejemplos para NodeMCU 0.9 (ESP-12 Module)

- ArduinoOTA
- DNSServer
- EEPROM
- ESP8266
- ESP8266AVRISP
- ESP8266HTTPClient
- ESP8266httpUpdate
- ESP8266HTTPUpdateServer
- ESP8266LLMNR
- ESP8266mDNS
- ESP8266NetBIOS
- ESP8266SdFat
- ESP8266SSDP
- ESP8266WebServer
- ESP8266WiFi**
- ESP8266WiFiMesh

- BearSSL_CertStore
- BearSSL_MaxFragmentLength
- BearSSL_Server
- BearSSL_ServerClientCert
- BearSSL_Sessions
- BearSSL_Validation
- CustomOffer
- HTTPSRequest
- IPv6
- NTPClient
- PagerServer
- RangeExtender-NAPT
- StaticLease
- Udp
- WiFiAccessPoint
- WiFiClient
- WiFiClientBasic
- WiFiEcho
- WiFiEvents
- WiFiManualWebServer
- WiFiMulti
- WiFiScan**
- WiFiShutdown
- WiFiTelnetToSerial

```

30
31 // the loop function
32 void loop() {
33   digitalWrite(LED
34   delay(1000);
35   digitalWrite(LED
36   delay(1000);
  
```

Ejercicios para la clase II

- 1) LED verde parpadea cíclicamente: medio segundo encendido y 3 segundos apagado.
- 2) LED verde parpadea 3 veces rápido: 150ms encendido y 150ms apagado. Luego está 2 segundos apagado. Esto se repite cíclicamente.
- 3) LED verde parpadea como en ejercicio 2. Mientras se pulse SW1, el LED verde se mantiene encendido. Al soltar SW1, vuelve a su ciclo normal.
- 4) Mientras esté pulsado SW1, el LED verde está encendido. Mientras esté pulsado SW2, el LED verde está apagado. Se prioriza SW2 sobre SW1.
- 5) Al pulsar y soltar SW1, el led verde cambia de estado.

Ejercicios para la clase II

- 6) Al pulsar SW1, el led verde permanece encendido aunque se suelte SW1.
Al pulsar SW2, el led verde permanece apagado aunque se suelte SW2.
- 7) Mientras esté pulsado SW1, el LED verde parpadea con PWM: $T_{on}=20\mu s$, $T_{pwm}=100\mu s$. Mientras esté pulsado SW2, el LED verde parpadea con PWM: $T_{on}=80\mu s$, $T_{pwm}=100\mu s$. Se prioriza SW1 sobre SW2.
- 8) LED verde parpadea con PWM: T_{on} es igual a la lectura del potenciómetro RV1 en ms. T_{pwm} es fijo de 1024 ms.
- 9) LED verde parpadea como en ejercicio 6. LED azul parpadea de manera **complementaria** al LED verde. Esto significa, ambos tienen el mismo T_{pwm} pero el T_{on} de uno es igual al T_{off} del otro.