Universidad Iberoamericana Puebla

Departamento de Ciencias e Ingenierías

Laboratorio de Redes Digitales

Alvaro Zaid Gallardo Hernández

Patricia Mercedes Cortés Garcia

Rosa Elena Mar Larios

Date: 19th February 2023 Maestro

Word count: 2,999 Huber Giron Nieto

Reporte de Practica 4: "Wifi ESP8266 Entradas y Salidas Digitales"

Contents

1 Resumen

En esta práctica, se desarrollaran algunos ejercicios para el ESP8266 O ESP32.

2 Introducción

El NodeMCU ESP8266 y el ESP32 son placas de desarrollo de bajo costo que se utilizan para proyectos de Internet de las cosas (IoT) y para crear prototipos de dispositivos conectados a la red. Ambos están diseñados para ser programados utilizando el lenguaje de programación Arduino y se basan en microcontroladores de la empresa Espressif.

El NodeMCU ESP8266 es un microcontrolador de 32 bits que utiliza el procesador ESP8266 de Espressif. Tiene WiFi integrado y soporta una variedad de protocolos de comunicación, como TCP, UDP, HTTP, MQTT y más. También tiene 11 pines de entrada/salida digital, 1 pin analógico y puede ser alimentado a través de un puerto micro USB o una fuente de alimentación externa. Es compatible con una amplia gama de sensores y actuadores, lo que lo hace ideal para proyectos de IoT.

Por otro lado, el ESP32 es una evolución del ESP8266 y también tiene WiFi integrado, pero con algunas mejoras importantes. El ESP32 es un microcontrolador de 32 bits con dos núcleos de procesamiento y ofrece más pines de entrada/salida digitales y analógicos, así como más opciones de conectividad, como Bluetooth y Bluetooth LE. También tiene un mayor rendimiento y es capaz de manejar tareas más complejas. El ESP32 es una excelente opción para proyectos de IoT que requieren una mayor capacidad de procesamiento y conectividad.

En resumen, tanto el NodeMCU ESP8266 como el ESP32 son placas de desarrollo económicas y versátiles que ofrecen capacidades de conectividad inalámbrica y una amplia gama de opciones de entrada/salida para proyectos de IoT. La elección entre ellos depende de las necesidades específicas de cada proyecto y de las capacidades que se requieren.

3 Objetivos

3.1 General

Conocer, identificar y comprobar el funcionamiento de las entradas y salidas digitales en el módulo NodeMCU ESP8266 o ESP32.

3.2 Específicos

Contador de pulsos

1. Utilizando la información e instrucciones de la presentación, configurar el IDE de Arduino para poder programar el NodeMCU directamente desde el entorno de Arduino, instalando las herramientas necesarias.
2. Revisar el pinout y los GPIO disponibles para el uso del NodeMCU seleccionado.
3. Desarrollar un programa de encendido de un LED mediante programación.
4. Desarrollar un programa de apagado de un LED mediante programación.
5. Desarrollar un programa de encendido y apagado de un LED mediante retardos.
6. Desarrollar un sistema de encendido y apagado de un LED mediante el uso de botones.
7. Desarrollar un sistema de encendido y apagado de un LED mediante el uso de dos botones y una condicional OR.
8. Desarrollar un sistema de encendido y apagado de un LED mediante el uso de dos botones y una condicional AND.

4 Materiales

- Node MCU ESP8266 o ESP32.
- LED, resistencia de 220 Ohms.
- 4 Botones y 4 resistencias de 1Khom
- Protoboard y cables

5 Procedimiento

Desarrollar un programa de encendido de un LED mediante programación.

```
void setup() {
  pinMode(16, OUTPUT) // El pin 16 se asigna como salida.
  }

void loop() {
digitalWrite(16, HIGH);} // Se da la instruccion de encender
```

Enlace: https://youtube.com/shorts/nISYE9hT0qk?feature=share

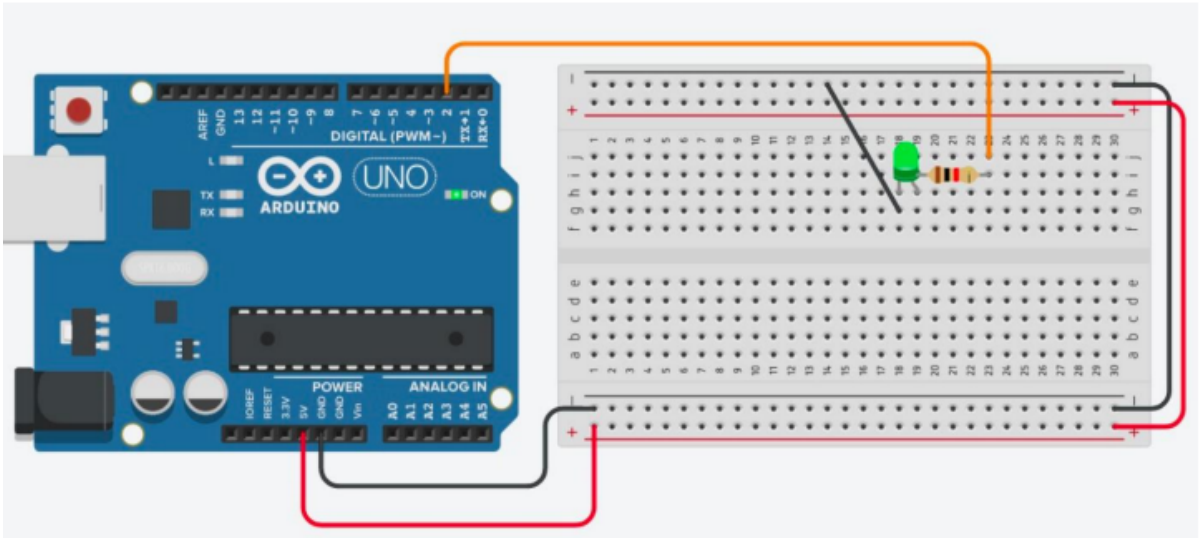

Figure 1: Encendido

Desarrollar un programa de apagado de un LED mediante programación.

Para poder realizar el programa se ocupa el siguiente código:

```
void setup() {
  pinMode(16, OUTPUT); // Se asigna el pin 16 como salida
}
void loop() {
  digitalWrite(16, HIGH); // encendido del led
  delay(1000);
  digitalWrite(16, LOW); // apagado del led
  delay(1000);
  }
```

Enlace: https://youtube.com/shorts/1zgsFm5aCZ0?feature=share

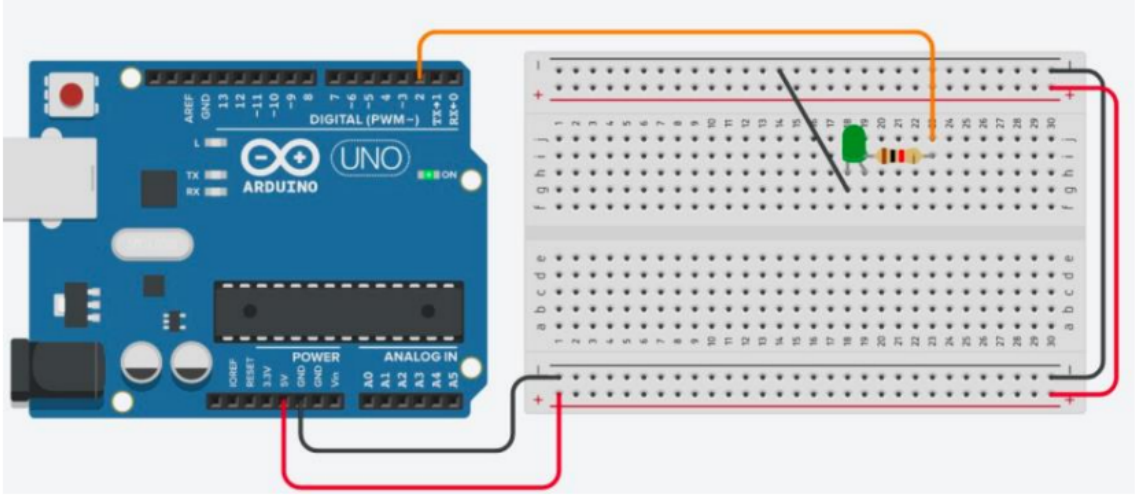

Figure 2: Apagado

Desarrollar un programa de encendido y apagado de un LED mediante retardos.

Para poder realizar el programa se ocupa el siguiente código:

```
1
2  void setup() {
3    pinMode(16, OUTPUT);
4  }
5  void loop() {
6    digitalWrite(16, HIGH); //Encendido
7    delay(3000);
8    digitalWrite(16, LOW); //Apagado
9    delay(1000);
10   }
```

Enlace: https://youtube.com/shorts/aaII-wuXUN8?feature=share

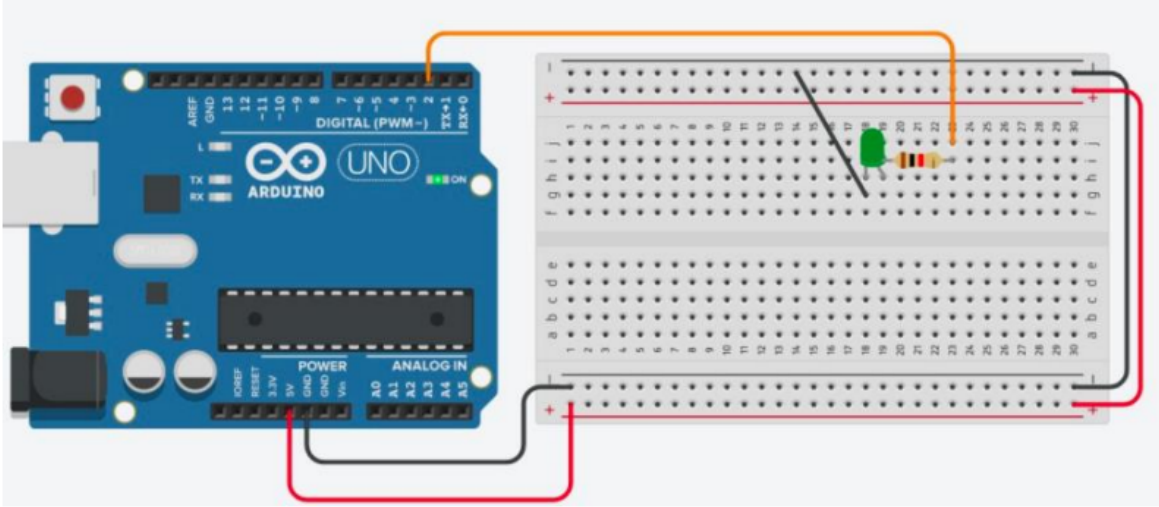

Figure 3: Retardos

Desarrollar un sistema de encendido y apagado de un LED mediante el uso de botones.

Para poder realizar el programa se ocupa el siguiente código:

```
1
2  void setup()
3  {
4    pinMode(16,OUTPUT);
5    pinMode(35,INPUT);
6    pinMode(34,INPUT);
7  }
8  void loop()
9  {
10
11   if  (digitalRead(34)==HIGH)
12   {
13   digitalWrite(16,HIGH);
14   }
15   else {
16     digitalWrite(16,LOW);
17   }
18 }
```

Enlace: https://youtube.com/shorts/Pv2ddGwAshA?feature=share

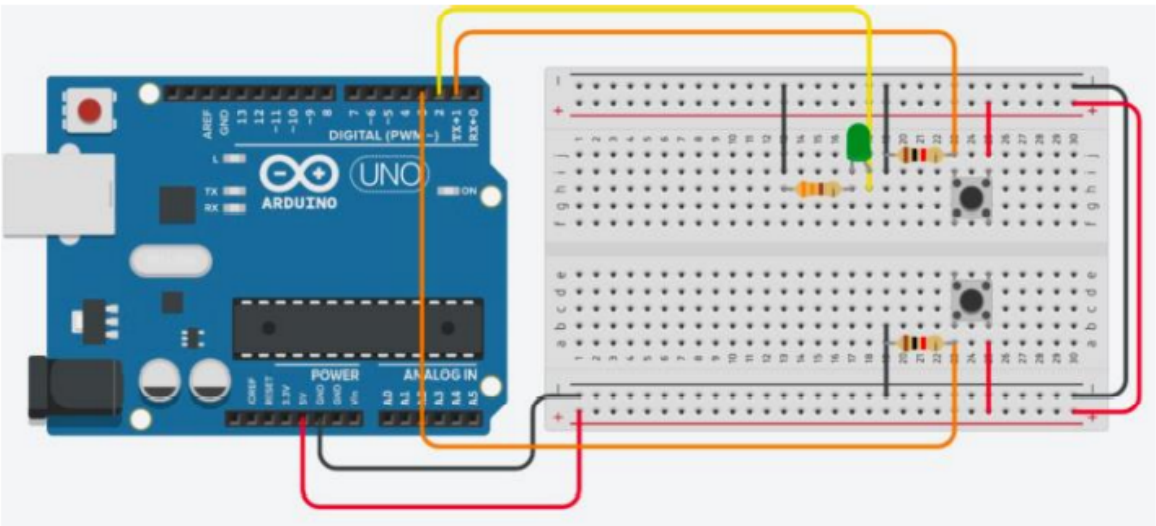

Figure 4: Botones

Desarrollar un sistema de encendido y apagado de un LED mediante el uso de dos botones y una condicional OR.

Para poder realizar el programa se ocupa el siguiente código:

```
void setup()
{
  pinMode(16,OUTPUT); //Salida
  pinMode(35,INPUT);    //Ebntrada
  pinMode(34,INPUT);    //Entrada
}
void loop()
{
  if  (digitalRead(34)==HIGH || digitalRead(35)==HIGH) // OR
  {
  digitalWrite(16,HIGH);
  }
  else {
    digitalWrite(16,LOW);
  }
}
```

Enlace: https: https://youtube.com/shorts/yuZtugWAZnE?feature=share

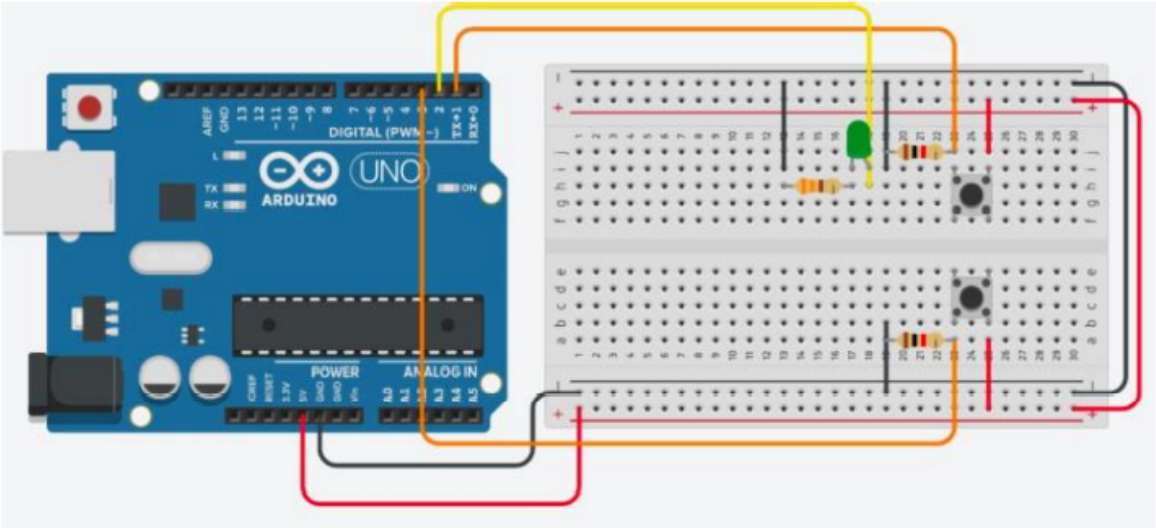

Figure 5: OR

Desarrollar un sistema de encendido y apagado de un LED mediante el uso de dos botones y una condicional AND.

Para poder realizar el programa se ocupa el siguiente código:

```
void setup()
{
  pinMode(16,OUTPUT);
  pinMode(35,INPUT);
  pinMode(34,INPUT);
}
void loop()
{
  if  (digitalRead(34)==HIGH && digitalRead(35)==HIGH) //AND
  {
  digitalWrite(16,HIGH);
  }
  else {
    digitalWrite(16,LOW);
  }
}
```

Enlace: https: https://youtube.com/shorts/cgvSH4813Yw?feature=share

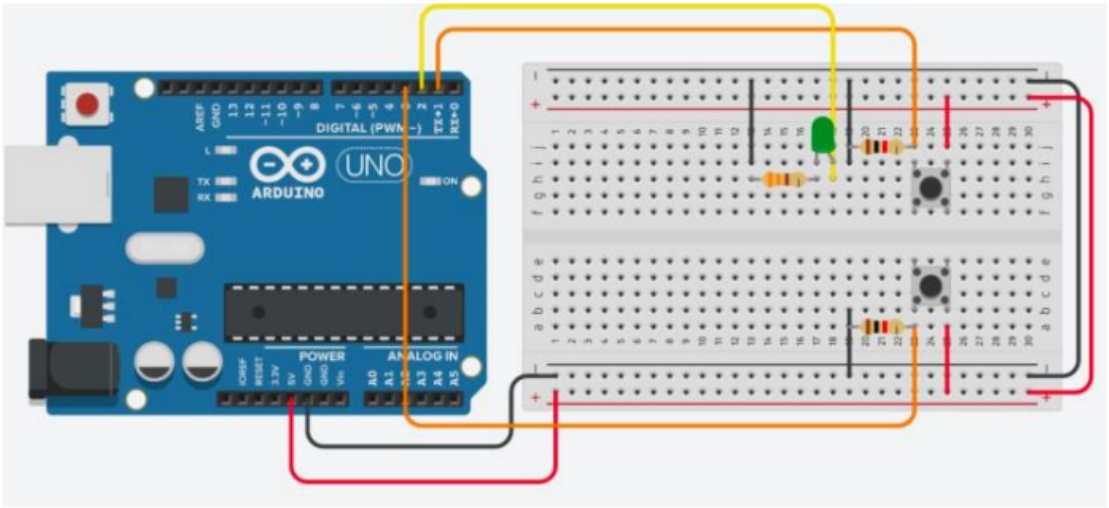

Figure 6: AND

6 Conslusión

En conclusión, en este reporte se presentó la programación del ESP32 y la realización de varios ejercicios con éxito. Se logró programar el ESP32 utilizando el entorno de desarrollo de Arduino y se implementaron diferentes programas para realizar diversas tareas.

Se inició con el ejercicio básico de encender y apagar un LED utilizando el ESP32, lo que permitió demostrar que el ESP32 está funcionando correctamente. Posteriormente, se avanzó en ejercicios más complejos, como la elaboracón de compuertas lógicas.

En cada uno de los ejercicios se realizaron pruebas y se verificó el correcto funcionamiento del programa. Además, se presentaron los resultados obtenidos y se explicó cómo se realizaron las conexiones necesarias para cada ejercicio.

En general, se puede concluir que la programación del ESP32 es relativamente sencilla y que existen muchas posibilidades para utilizarlo en diferentes proyectos. Además, se pudo verificar que el ESP32 es una herramienta muy útil para la realización de tareas de IoT, lo que lo convierte en una herramienta muy valiosa para cualquier proyecto que involucre dispositivos conectados a Internet.

Se lograron todos los objetivos planteados en este reporte y se demostró que el ESP32 es una herramienta muy útil y versátil para la programación de dispositivos de IoT.

7 Referencias

Bertoleti, P. (2019). Proyectos com ESP32 y LoRa. Editora NCB.

Babiuch, M., Foltýnek, P., Smutný, P. (2019, May). Using the ESP32 microcontroller for data processing. In 2019 20th International Carpathian Control Conference (ICCC) (pp. 1-6). IEEE.