

UNIVERSIDAD NACIONAL 3 DE FEBRERO



“Proyecto ESP32”.

Asignatura: Introducción a la Ingeniería en Computación

Profesor:

- Jorge Fossati

Cursada: 465 - Turno Noche / 2do Cuatrimestre / 2024

Integrantes:

- Rivadera Adrián – 72043 - adrianerivadera@gmail.com

Índice

1. Introducción	3
2. Marco Teórico	3
3. Entornos de Desarrollo y Simulación	3
4. Arquitectura del Sistema	4
5. Descripción del Circuito	4
6. Interacción entre Componentes	5
7. Funcionalidad del Sistema	6
8. Microcontrolador ESP32	6
9. Firmware del Sistema	7
10. Conclusión y Mejoras Futuras	8
11. Referencias	8

1. Introducción

El presente informe tiene como objetivo detallar el desarrollo de un sistema de semáforo basado en la plataforma Arduino utilizando el microcontrolador ESP32, este proyecto fue diseñado con el objetivo de simular un semáforo.

El sistema de semáforo, programado en el entorno **Arduino IDE 2.3.2**, responde a la interacción del usuario mediante un botón que cambia el estado del semáforo a rojo, para que el peatón pueda cruzar. Para su simulación y pruebas, se utilizó el software **Proteus**, permitiendo la verificación visual del funcionamiento del circuito.

- ESP32: Microcontrolador principal encargado de controlar las secuencias de luces del semáforo.
- Arduino IDE: Entorno de desarrollo para la programación del ESP32.
- Proteus: Software de simulación electrónica para visualizar el funcionamiento del semáforo.

2. Marco Teórico

El semáforo tiene un funcionamiento controlado por un botón, es decir, el semaforo permanecera en verde hasta que el usuario toque el boton, esto provocara que el led verde titile 3 veces para que luego pase a amarillo y por ultimo a rojo.

Al simular el semáforo con herramientas como Proteus o Wokwi, se puede verificar el comportamiento del circuito y del código sin necesidad de hardware real.

- **Proteus** permite visualizar cómo interactúan los componentes y validar el flujo de la secuencia del semáforo.
- **Wokwi** proporciona un entorno online que facilita la simulación de proyectos con ESP32 y Arduino, permitiendo al usuario cargar su código y comprobar cómo funcionará en un entorno virtual.

3. Entornos de Desarrollo y Simulación

Proteus es una herramienta de simulación que permite diseñar y simular circuitos electrónicos de manera virtual. En este proyecto, Proteus se utilizó para simular el circuito eléctrico del semáforo, permitiendo probar el comportamiento del hardware sin necesidad de componentes físicos. Esto es especialmente útil para verificar la correcta conexión de los LEDs, la interacción con el botón, y la sincronización de las luces en cada etapa del semáforo (verde, amarillo, rojo).

Wokwi es una plataforma en línea para la simulación de proyectos basados en microcontroladores como ESP32 y Arduino. En este proyecto, Wokwi proporcionó un entorno accesible para un primer acercamiento al sistema de semáforo. Con esta herramienta, se pudo cargar el código, simular el comportamiento del semáforo y verificar su funcionalidad sin necesidad de instalar software adicional ni contar con hardware físico.

Arduino IDE es el entorno de desarrollo utilizado para escribir, compilar y cargar el código en el ESP32, que es el microcontrolador encargado de controlar las luces del semáforo y responder a la interacción del botón. Se desarrolló la secuencia de control del semáforo y la lógica para la interacción con el botón. Luego, el código es verificado y convertido a un formato ejecutable que puede ser entendido por el ESP32. Se genera un archivo .hex en este proceso. Este archivo es una versión binaria del código que puede ser cargada directamente en el ESP32 o utilizada en simulaciones como las de Proteus.

4. Arquitectura del Sistema

El sistema del semáforo está compuesto por varios componentes electrónicos conectados y controlados por el ESP32. El ESP32 es el encargado de gestionar las señales del botón y controlar la secuencia de las luces (verde, amarillo y rojo) para simular el funcionamiento de un semáforo.

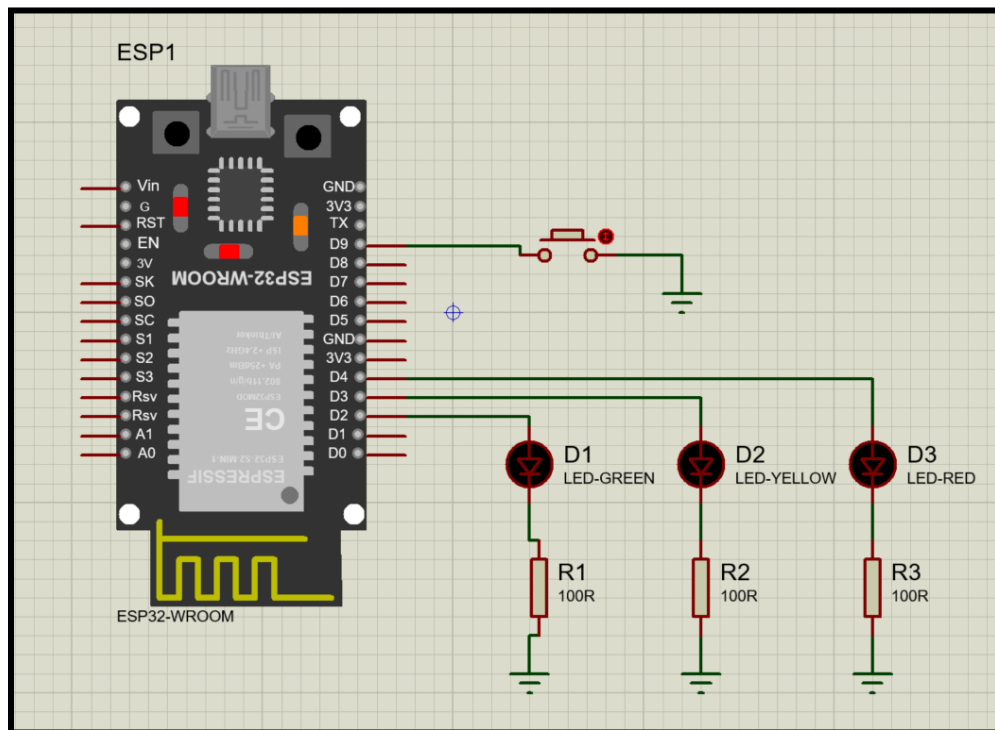


Figura 1: Esquemático del proyecto

5. Descripción del Circuito

Componente	Descripción	Características
ESP32	Microcontrolador con Wi-Fi y Bluetooth integrados	<ul style="list-style-type: none">Microcontrolador con Wi-Fi y Bluetooth integradosPines digitales: 34 pines GPIOConectividad: Wi-Fi 802.11 b/g/n, Bluetooth 4.2

LED Verde	Diodo emisor de luz de color verde	<ul style="list-style-type: none">• Voltaje de funcionamiento: 2V - 3V• Corriente: 20mA (máximo)• Color: Verde
LED Amarillo	Diodo emisor de luz de color amarillo	<ul style="list-style-type: none">• Voltaje de funcionamiento: 2V - 3V• Corriente: 20mA (máximo)• Color: Amarillo
LED Rojo	Diodo emisor de luz de color rojo	<ul style="list-style-type: none">• Voltaje de funcionamiento: 2V - 3V• Corriente: 20mA (máximo)• Color: Rojo
Resistencia (220Ω)	Limitador de corriente para LEDs	<ul style="list-style-type: none">• Valor: 220 ohmios• Tolerancia: ±5%
Botón	Interruptor para iniciar el ciclo del semáforo	<ul style="list-style-type: none">• Voltaje de funcionamiento: 3.3V (compatible con ESP32)• Capacidad de corriente: 50mA (máximo)

6. Interacción entre Componentes

- **Flujo de Señales y Datos**

- **Botón de Activación:** Actúa como un interruptor que envía una señal al ESP32. Cuando el botón se presiona, el ESP32 recibe una señal digital baja (LOW) debido a la configuración del pin en modo pull-up (es decir, el pin está en HIGH por defecto y pasa a LOW cuando se presiona el botón). Esta señal digital determina si el semáforo debe iniciar su ciclo.
- **ESP32:** monitorea continuamente el estado del botón a través de un pin configurado como entrada con resistencia pull-up. Si el botón está presionado (señal LOW), el ESP32 inicia la secuencia del semáforo. Si no está presionado, el semáforo permanece en verde.
- **Control de LEDs:**
 - LED Verde: mantiene el LED verde encendido de manera constante mientras el botón no está presionado. Cuando se presiona el botón, el LED verde titila tres veces antes de cambiar al siguiente estado.
 - LED Amarillo: Después del parpadeo del LED verde, enciende el LED amarillo durante 1 segundos para indicar el cambio de estado.
 - LED Rojo: Finalmente, el ESP32 enciende el LED rojo, que permanece encendido hasta que el botón sea liberado.

- **Flujo de Señales en la Simulación**

- En la simulación con Proteus, el flujo de señales se replica digitalmente para validar el comportamiento del sistema. El software simula la respuesta del ESP32 a la entrada del botón y el control de los LEDs, permitiendo observar cómo se comporta el circuito en condiciones virtuales.

7. Funcionalidad del Sistema

- **Comportamiento del Semáforo**

- **Cambio de Luces:** Opera con tres luces principales: verde, amarillo y rojo. La secuencia de cambio es la siguiente:
- **Verde:** Indica que el tráfico puede avanzar. El LED verde está encendido de manera constante por defecto. Cuando el botón es presionado, el LED verde parpadea tres veces (encendiéndose y apagándose rápidamente) antes de cambiar al siguiente estado. Cada parpadeo dura 300 ms.
- **Amarillo:** Señala que el semáforo está a punto de cambiar a rojo. El LED amarillo se enciende después del verde, y antes de encenderse, el LED verde parpadea dos veces para advertir del cambio inminente. El LED amarillo permanece encendido durante 2 segundos.
- **Rojo:** Indica que el tráfico debe detenerse. El LED rojo se enciende inmediatamente después del amarillo y permanece encendido mientras el botón esté presionado. Una vez que se suelta el botón, el semáforo regresa al estado verde.

8. Microcontrolador ESP32

Microcontrolador ESP32

El ESP32 es un microcontrolador potente, ideal para una amplia gama de aplicaciones, incluyendo proyectos IoT (Internet of Things). Sus características principales incluyen:

- Procesador: Dual-core Tensilica LX6, con frecuencia de hasta 240 MHz.
- Memoria:
- 520 KB de SRAM.
- Soporte para memoria externa a través de SPI.
- Conectividad:
- WiFi: Compatible con 802.11 b/g/n, con capacidades de red tanto en modo estación como en modo punto de acceso.
- Bluetooth: Soporte para Bluetooth Classic y Bluetooth Low Energy (BLE).
- Entradas/Salidas:
- Amplia cantidad de pines GPIO (hasta 34).
- Interfaces para SPI, I2C, UART, PWM, y ADC.

Capacidades en Proyectos IoT

El ESP32 es una excelente opción para proyectos IoT debido a sus capacidades avanzadas:

- **Procesador Potente:** El procesador dual-core y la alta frecuencia de reloj permiten ejecutar aplicaciones complejas y manejar múltiples tareas simultáneamente.
- **Versatilidad:** Soporta múltiples protocolos de comunicación y tiene una gran cantidad de pines GPIO, lo que facilita la integración con diversos sensores y actuadores.
- **Eficiencia Energética:** Ofrece modos de bajo consumo que son esenciales para dispositivos IoT que deben operar con batería durante largos períodos.

9. Firmware del Sistema

Pseudocódigo

1. Inicialización:

- Configurar los pines de los LEDs como salida.
- Configurar el pin del botón como entrada con pull-up.
- Inicializar la comunicación serial para depuración.

2. Loop Principal:

- a. Leer el estado del botón.
- b. Si el botón está presionado:
 - i. Apagar todos los LEDs.
 - ii. Hacer parpadear el LED verde 3 veces.
 - iii. Cambiar al LED amarillo.
 - iv. Esperar 2 segundos.
 - v. Cambiar al LED rojo.
 - vi. Mantener el LED rojo encendido hasta que el botón se suelte.
- c. Si el botón no está presionado
 - i. Apagar todos los LEDs excepto el LED verde, que se mantendrá encendido de manera constante.

Diagramas de Flujo:



10. Conclusión y Mejoras Futuras

Conclusiones: Este proyecto me pareció muy interesante ya que me permitió profundizar en el funcionamiento de los sistemas controlados por microcontroladores, y aprender de manera práctica cómo realizar pruebas y simulaciones utilizando Arduino IDE y Proteus.

11. Referencias

1. **Curso ESP32 desde cero**
https://www.youtube.com/watch?v=FwDCSOqxAZ8&list=PLCYCyKTQZID0aoz8CApZkxmzITR_SqGHR
2. **Esp32 características y pines**
<https://pasionelectronica.com/esp32-caracteristicas-y-pines/>
3. **Sigma Electronica ESP32-WROOM-32D**
<https://www.sigmaelectronica.net/producto/esp32-wroom-32d/>
4. **Espressif Systems. (2021). ESP32 technical reference manual.**
https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf