

**“Proyecto ESP32”.**

Asignatura: Introducción a la Ingeniería en Computación

Profesor:

* Jorge Fossati. Cintia Capece

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

Integrantes:

* Ignacio Nieto, nieto.ignacio98@gmail.com
* Legajo: 74968

**Índice**

[**1. Introducción 3**](#_heading=h.gjdgxs)

[**2. Marco Teórico 3**](#_heading=h.30j0zll)

[**3. Entornos de Desarrollo y Simulación 3**](#_heading=h.1fob9te)

[**4. Arquitectura del Sistema 3**](#_heading=h.3znysh7)

[**5. Descripción del Circuito 3**](#_heading=h.2et92p0)

[**6. Interacción entre Componentes 3**](#_heading=h.tyjcwt)

[**7. Funcionalidad del Sistema 3**](#_heading=h.3dy6vkm)

[**8. Microcontrolador ESP32 4**](#_heading=h.1t3h5sf)

[**9. Firmware del Sistema 4**](#_heading=h.4d34og8)

[**10. Conclusión y Mejoras Futuras 4**](#_heading=h.2s8eyo1)

[**11. Referencias 4**](#_heading=h.17dp8vu)

### 1. Introducción

El objetivo de este informe es poder aprender a como realizar y utilizar los programas Arduino y Proteus para simular y programar el simulacro de un semáforo mediante el microcontrolador ESP32.

El semáforo, mediante un botón, activa la secuencia de luces en los leds donde en Proteus podemos observar como se realiza la simulación.

### 2. Marco Teórico

La simulación es una herramienta fundamental en este tipo de proyectos, marcando asi varios aspectos: Validación previa al desarrollo físico y posibles costos, el aprendizaje es importante en este tipo de simulaciones y seguridad al no manipular objetos físicos.

Proteus: Es una herramienta de simulación de circuitos electrónicos que permite integrar microcontroladores (como Arduino, PIC, entre otros) con componentes como LEDs, resistencias y sensores.

Wooki: También se utiliza para simulaciones y modelado. Su relevancia depende de su capacidad para integrar el software y el hardware en simulaciones de sistemas complejos.

### 3. Entornos de Desarrollo y Simulación

**Proteus**: En un proyecto de semáforo, podrías programar el microcontrolador y visualizar el comportamiento de los leds que representarían las luces del semáforo. Poder descargar el ESP32 en el proteus, los diversos leds, las resistencias, el botón y el ground para unir la resistencia con los leds y el ESP32.

**Wooki**: Wokwi proporciona un primer acercamiento al sistema de semáforo. Con esta herramienta, se pudo cargar el código para ver la simulación del semáforo sin necesidad de instalar software adicional o hardware.

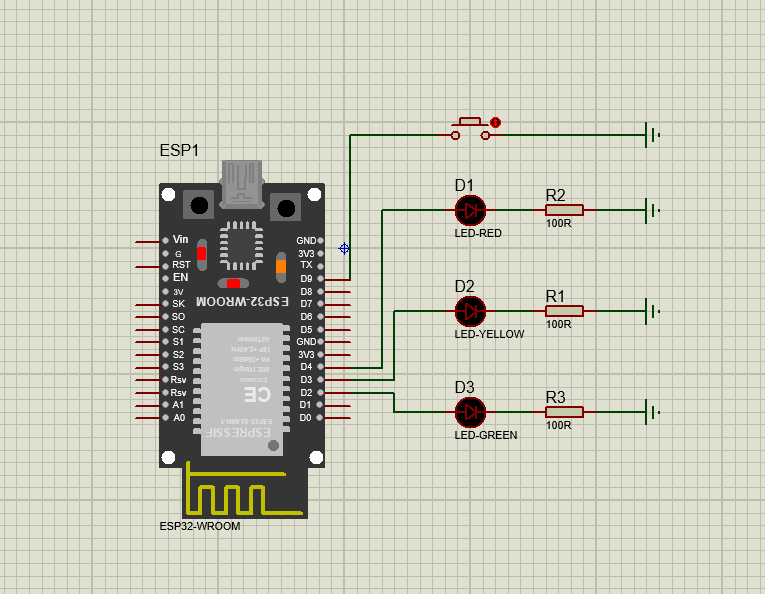
**Arduino IDE**: El Arduino nos permite realizar el código para programar luego en el proteus y así simularlo. El archivo .hex (es una versión binaria del código que puede ser cargada directamente en el ESP32 o utilizada en simulaciones como las de Proteus) nos permite copiarlo y pegarlo en el Proteus para ver la ejecución de la simulación.

### 

### 

### 4. Arquitectura del Sistema

El semáforo está conectado por el ESP32. Este microcontrolador es el encargado de gestionar las señales del botón y controlar la secuencia de las luces.



### 5. Descripción del Circuito

|  |  |
| --- | --- |
| Componentes | Descripción de Componentes |
| ESP32 | Microcontrolador con Wi-Fi y Bluetooth integrados. 34 pines GPIO |
| LEDS | Leds de color rojo, amarillo y verde. Voltaje de 2v-3v. 20mA |
| Resistencia | Limitador de corriente. Sirve para dar más vida útil al led  220 ohmios |
| Botón | Botón para iniciar el ciclo del semáforo. Voltaje 3.3v |

### 6. Interacción entre Componentes

* **Flujo de Señales y Datos**
  + **Botón de Activación:** Es un interruptor que envía una señal HIGH o LOW al ESP32 para encender o apagar.
  + **ESP32:** Monitorea el estado del botón a través de un pin configurado como entrada.
  + LEDs: El ESP32 enciende el LEDS durante 2 segundos.
* **Flujo de Señales en la Simulación**
  + En Proteus, el software simula la respuesta del ESP32 a la entrada del botón y el control de los LEDs para observar la simulación.

### 7. Funcionalidad del Sistema

* **Comportamiento del Semáforo**
  + **Luces:** Son tres luces principales con una secuencia de encendido. El led verde Indica que se puede pasar. Luego hace un parpadeo y cambia al amarrillo. Por último, se prende el rojo indicando que se debe de frenar.

### 8. Microcontrolador ESP32

**ESP32**

El ESP32 es un microcontrolador altamente versátil y potente desarrollado por Espressif Systems. Su estructura de sistema combina múltiples subsistemas y componentes que permiten una amplia gama de aplicaciones. El ESP32 cuenta con componentes como un procesador, memoria, conectividad Wi-Fi, Bluetooth o Ethernet. También cuenta con interfaces de entrada y salida, modos de ahorro de energía, características diseñadas para proteger el dispositivo y las comunicaciones, también incluye varios periféricos que mejoran su capacidad para interactuar con el mundo físico como sensores.

El ESP32 es ideal para aplicaciones IoT, domótica, control de motores, sistemas embebidos industriales, y cualquier dispositivo que requiera conectividad Wi-Fi/Bluetooth con procesamiento eficiente de energía

**Capacidades en Proyectos IoT**

El ESP32 es una excelente opción para proyectos IoT debido a sus capacidades avanzadas gracias a su procesador potente, su versatilidad con gran cantidad de pines y con eficiencia energética

### 9. Firmware del Sistema

**Pseudocódigo**

1. **Inicialización:**

* Configurar los pines de los LEDs y el botón para iniciar

**Loop Principal:**

* Encender el LED verde.
* Esperar 2 segundos.
* El LED verde parpadea 3 veces.
* Cambiar al LED amarillo.
* Esperar 2 segundos.
* Cambiar al LED rojo.
* Esperar 2 segundos.
* Si el botón no está presionado:
* Apagar todos los LEDs.

**Diagramas de Flujo:**

1. **Inicio**

* Configurar pines y comunicación serial.

1. **Botón presionado**

* Encender LED verde.
* Esperar 2 segundos.
* Parpadear LED verde 3 veces.
* Cambiar a LED amarillo.
* Esperar 2 segundos.
* Cambiar a LED rojo.
* Esperar 2 segundos.
* Botón no presionado (se apaga)

### 10. Conclusión y Mejoras Futuras

**Conclusiones**: Para concluir podemos destacar la gran variedad de proyectos que uno puede realizar con el ESP32. El semáforo nos permite entender cómo funcionan todos los componentes del sistema para poder lograr el fin de encenderlo.

### 11. Referencias

1. **Curso ESP32 desde cero**  
   <https://www.youtube.com/watch?v=FwDCSOqxAZ8&list=PLCYCyKTQZlD0aoz8CApZkxmzlTR_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>