**INSTITUTO TECNOLÓGICO AUTÓNOMO DE MÉXICO**



Laboratorio de Principios de Mecatrónica

**Práctica 1. Microcontroladores y Señales Digitales**

* Fuentes Campana Alejandro Antonio
* Apellido Paterno Apellido Materno Nombre (s)

Asignatura: Laboratorio de Principios de Mecatrónica

Docente: M.I. Sergio Hernández Sánchez

Grupo:

Semestre: Otoño 2023

1. **Introducción**

En el siguiente documento usaremos un aparato ESP32 para correr programa simples y familiarizarnos con los pines de propósito general, así como el IDE de andruino.

1. **Objetivos**

* Conocer las plataformas, herramientas y ambientes más comunes para el desarrollo de sistemas mecatrónicos basados en microcontroladores.
* Identificar los elementos básicos de la arquitectura de un microcontrolador digital.
* Identificar los módulos que componen a la tarjeta de desarrollo Arduino y ESP32 y comparar su funcionalidad.

1. **Marco Teórico**

**Un microcontrolador es un circuito integrado, es el componente principal (controla a los demás circuitos integrados) de una computadora. Se encarga de controlar las instrucciones y mandarlas a ejecutar. El esp32 tiene un 32-bit LX6 microprocesador y el Arduino UNO tiene un ATmega328P**

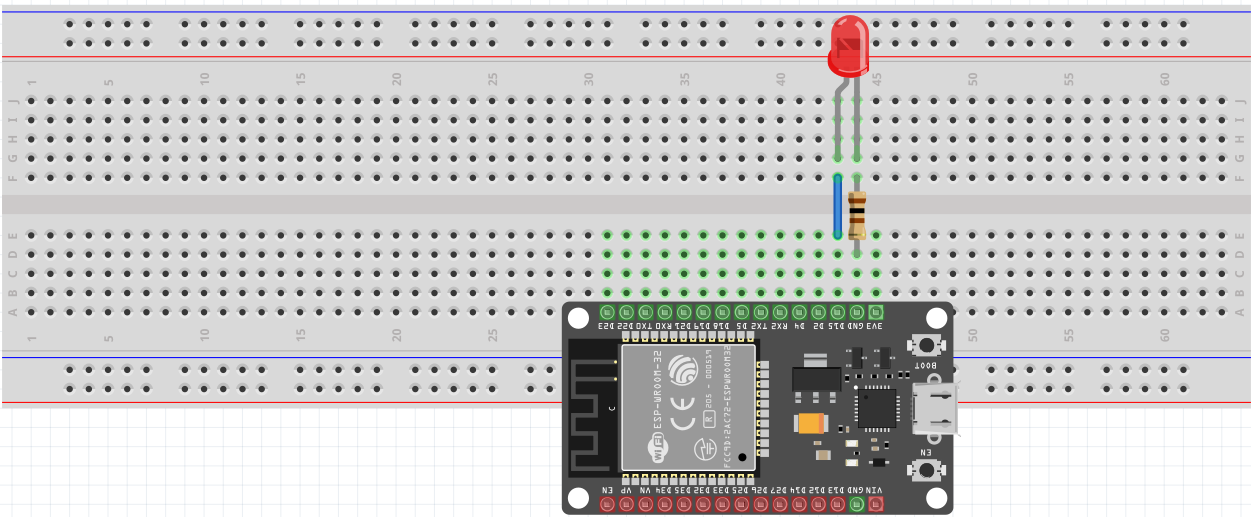
|  |  |  |
| --- | --- | --- |
|  | **ESP32** | **Andruino UNO** |
| **Pines GPIO** | **34** | **28** |
| **Pines IO analog** | **16** | **6** |
| **Pines PWM** | **18** | **8** |
| **V pines** | **0 a 3.3V** | **5V** |
| **V externo** | **5 a 12V** | **7 a 12 V** |
| **Corriente Pines** | **40mA** | **40 mA** |
| **Velocidad reloj** | **80 A 240MHz** | **16MHz** |

**Esp32 se conecta con micro USB o via wifi/bluetooth**

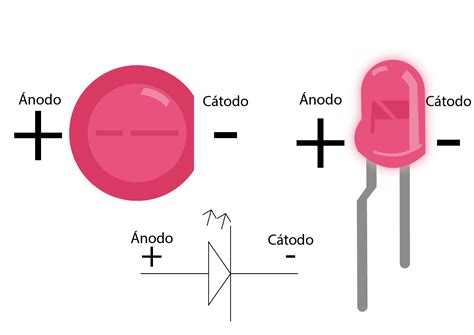
**Andruino UNO se conecta vía USB B, usa los protocolos UART, SPI, 12C para comunicarse.**

1. **Material y equipo utilizado**
   1. 1 ESP32 Dev Kit v1
   2. 1 cable USB A - micro USB
   3. 1 protoboard grande
   4. 2 LED rojo
   5. 2 LED amarillo
   6. 2 LED verde
   7. 1 Push-Button
   8. 6 resistor 100 Ω
   9. 1 resistor 10 k Ω
   10. Jumpers macho - macho
2. **Experimentos y simulaciones**
   1. **Actividad 1 – Blink**

Para esta actividad, se busca encender y apagar un LED, para lo cual, se conectará una resistencia de 100 Ω al ánodo de un LED y el ánodo del LED estará conectado al GPIO 15 de una ESP32 Dev Kit v1. Apóyese de la figura 1 y 2 para conectar lo que se solicita.

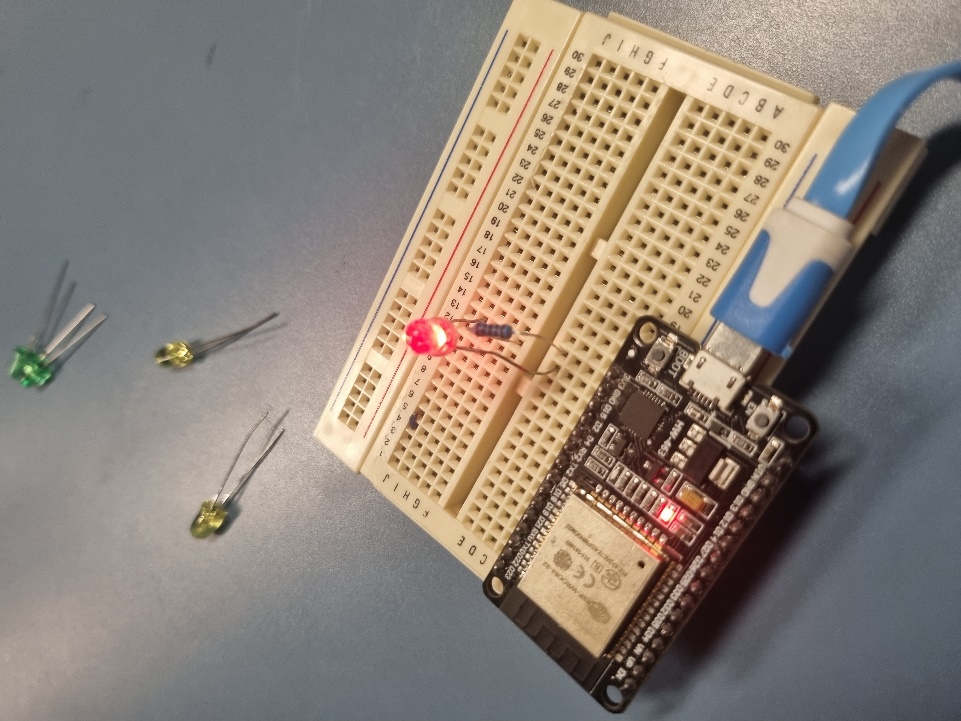


*Figura 1. Circuito para encender y apagar un LED.*



*Figura 2. Identificación del ánodo y cátodo de un LED.*

A continuación, coloque una fotografía del circuito armado en la figura 3 donde se muestre el LED encendido.



*Figura 3. Vista superior del circuito armado con LED.*

Enseguida, en su cuenta de GitHub, deberá crear una carpeta nombrada ***P\_Mecatronica***, y dentro de ella otra carpeta cuyo nombre sea **Practica01** y guarde dentro, el código generado cuyo nombre será ***P1A1-Blink***, y coloque enseguida el enlace de dicho archivo de manera que el profesor pueda verlo y evaluar su código. Este procedimiento descrito deberá hacerse para cada una de las actividades de cada práctica de laboratorio.

*Enlace a su código:*

https://github.com/Alefuecam/LPM/blob/979db7ef4b035fc43e1a533b224f2bc0824608f6/P\_Mecatronica/Practica01/P1A1-Blink

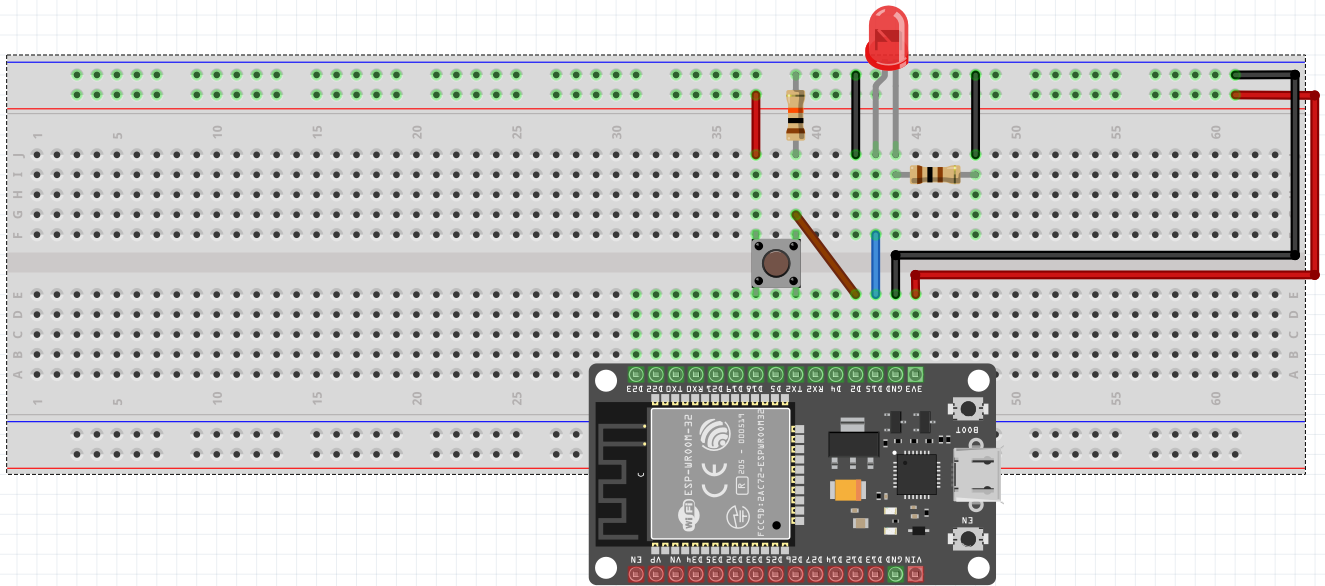
El código deberá encender un LED, dejarlo encendido por 1 segundo y posteriormente apagarlo por 1 segundo. Esto se deberá repetir indefinidamente. Deberá comentar cada línea de código de tal forma que explique su funcionamiento, recuerde que para comentar una línea se pondrá doble diagonal //.

A continuación, en alguna nube que tenga disponible coloque el video de esta primera actividad o si lo prefiere, puede subirlo a YouTube de forma pública o privada de forma que pueda visualizarse el funcionamiento de todo el sistema completo. Esto deberá realizarlo en todas las actividades de todas las prácticas.

*Enlace a su video: https://www.youtube.com/shorts/9B6S06akv6Q*

* 1. **Actividad 2 – Botón y Frecuencia**

Como segunda actividad, cree un nuevo sketch en el IDE de Arduino que deberá guardará como ***P1A2-Boton-Freq*** y guardar en su carpeta de **Practica01** de su repositorio. Con base en el circuito de la actividad 1, se pide que se añada un *push-button* en la configuración *pull-down* utilizando una resistencia de 10 kΩ de tal manera que al ser presionado, envié una señal de un uno lógico o de 5 V a la tarjeta ESP32 y cuando no esté presionado, se envíe un cero lógico o 0 V. Utilice el diagrama de la figura 4 para poder armarlo.



*Figura 4. Circuito para probar el push-button y frecuencia de LED.*

Una vez armado el circuito, genere un código en la IDE de Arduino de manera que cumpla con lo descrito en la tabla 1. Deberá comentar cada línea o cada sección de manera que se describa que está realizando a lo largo de su programa. Añada una fotografía de su circuito armado en la figura 5, el enlace de GitHub y su video de funcionamiento.

*Tabla 1. Lógica que deberá seguir el LED.*

|  |  |
| --- | --- |
| Estado del Push-Button | Frecuencia de encendido del LED |
| LOW | 1 Hz |
| HIGH | 4 Hz |

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza baja

*Figura 5. Vista superior del circuito armado de control de frecuencia.*

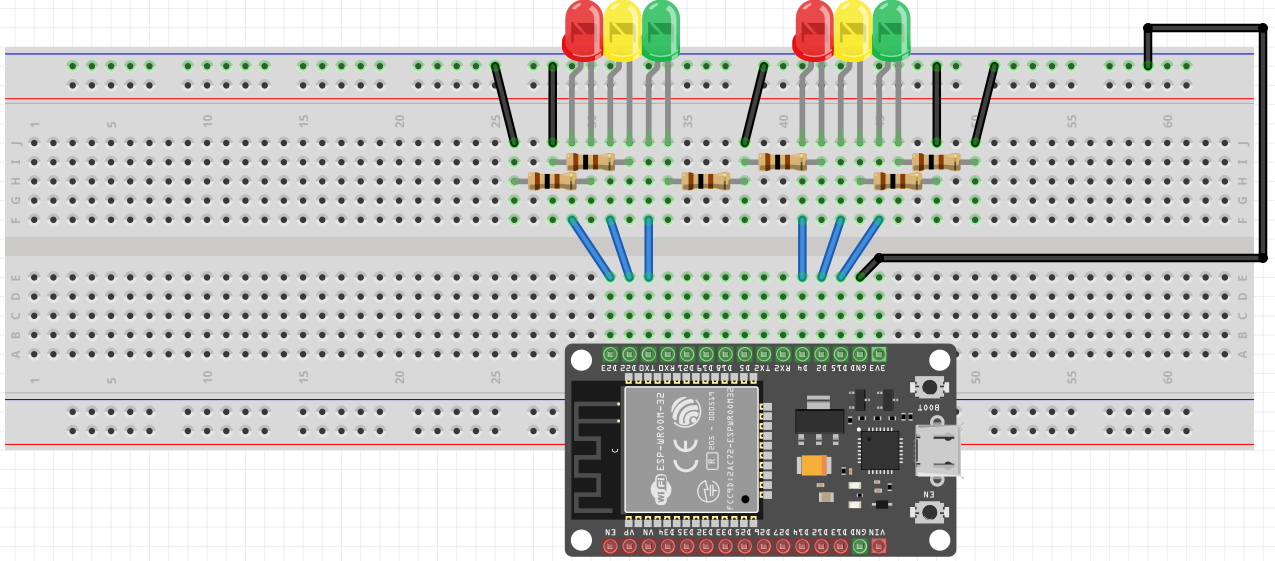
*Enlace a su código:*

*https://github.com/Alefuecam/LPM/blob/d6275e6c5fc5ee2387453347c0ae82877335107f/P\_Mecatronica/Practica01/P1A2-Boton-Freq*

*Enlace a su video: https://www.youtube.com/watch?v=b1-T-LsI8m8*

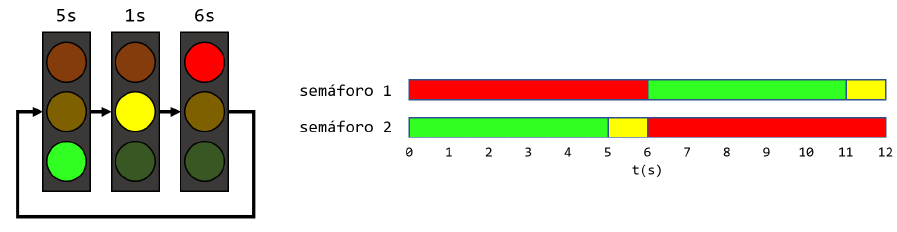
* 1. **Actividad 3 – Semáforo**

Finalmente, en esta última actividad conecte a su circuito 2 LEDs rojos, 2 LEDs amarillos y 2 verdes con sus respectivas resistencias de 100 Ω, de forma que estén acomodados como dos configuraciones de semáforos. Conecte el ánodo del LED a alguno de los pines digitales de su Arduino MEGA, como se muestra en la figura 6.



*Figura 6. Circuito para probar el semáforo.*

A continuación, genere el código necesario guardándolo como ***P1A3-Semaforo***, que deberá seguir la lógica descrita por la figura 7 para lograr el funcionamiento de 2 semáforos. Recuerde comentar su código.



*Figura 7. Lógica de encendido del semáforo.*

Añada una fotografía de su circuito armado en la figura 8, el enlace de GitHub y su video de funcionamiento.

*Figura 8. Vista superior del circuito del semáforo armado.*

*Enlace a su código:*

*https://github.com/Alefuecam/LPM/blob/66460e8ca3bb1d5a9a93d18d6c8e281d5f4a674d/P\_Mecatronica/Practica01/P1A3-Semaforo*

*Enlace a su video: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

1. **Conclusiones**

**Alejandro: Aprendí sobre el manejo de los pines de entrada y salida y como usar los delay para darle nuevas funciones a los elementos conectados. Podríamos juntar la actividad del botón y la del semáforo, si vemos el botón como una señal de que hay peatones podríamos pasar a rojo, si no hay peatones nunca dejar de estar en verde. (esto si la calle no tiene intersección con otra)**

Individuales, demuestra que aprendiste los conocimientos de la práctica utilizándolos en tu comentario final y proponiendo nuevos usos, evita dar opiniones subjetivas (como: “me gustó o estuvo muy bien”), indica el nombre del autor al inicio de su conclusión.

1. **Fuentes consultadas**