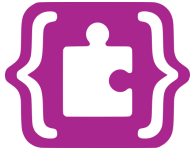


POTENCIÓMETRO

#R1AS05



Disponible en

¿Qué es?

En esta hoja de actividades, aprenderemos sobre el potenciómetro programando la placa para ajustar el brillo de un LED girando un mando.



Requisitos previos

- R1AS01 - Parpadeo de un LED
- R1AS02 - Placa de pruebas
- R1AS04 - Sensor de luz básico



Duración
20 minutos

Material

- 1 placa de programación "**STM32 IoT Node Board**"
- Cable USB Micro-B
- 1 placa de pruebas
- 1 potenciómetro
- 1 juego de LEDs
- 1 juego de resistencias
- Cables de puente

Nivel de dificultad

Intermedio

OBJETIVOS DE APRENDIZAJE

- Cablear los componentes externos a la placa
- Leer una entrada analógica mediante un potenciómetro
- Utilizar una entrada analógica para escribir una salida analógica





Un potenciómetro es **una resistencia de tres terminales** con un contacto deslizante o giratorio que forma un **divisor de tensión** ajustable. Si sólo se utilizan dos terminales, un extremo y el contacto deslizante, actúa como una resistencia variable o reóstato. El instrumento de medida llamado **potenciómetro** es esencialmente un divisor de **tensión** utilizado para medir el **potencial eléctrico** (tensión); el componente es una aplicación del mismo principio, de ahí su nombre.

Los potenciómetros se utilizan habitualmente para **controlar dispositivos eléctricos**, como los controles de volumen de los equipos de audio. Los potenciómetros accionados por un mecanismo pueden utilizarse como **transductores** de posición, por ejemplo, en un **joystick**. Los potenciómetros rara vez se utilizan para controlar directamente una potencia importante (más de un **vatio**), ya que la potencia disipada en el potenciómetro sería comparable a la de la carga controlada.

Fuente: <https://en.wikipedia.org/wiki/Potentiometer>



HAZLO



Cableado del potenciómetro

Conecta la punta izquierda del potenciómetro a **GND**. La punta derecha debe estar conectada a **3,3V**. Conecta la del medio a **A0**.

Cableado del LED

Conecta el **ánodo (+)** del LED en **D9**. Conecta el **cátodo (-)** del LED a una resistencia (330 ohmios). A continuación, conecta el lado no conectado de la resistencia a **GND**.

Conecta la placa al ordenador

Con tu cable USB, conecta la placa a tu ordenador utilizando el conector **micro-USB ST-LINK** (en la esquina derecha de la placa). Si todo va bien deberías ver una nueva unidad en tu ordenador llamada **DIS_L4IOT**. Esta unidad se utiliza para programar la placa simplemente copiando un archivo binario.

Abre MakeCode

Ve al editor de **Let's STEAM MakeCode**. En la página de inicio, crea un nuevo proyecto haciendo clic en el botón "Nuevo proyecto". Dale un nombre a tu proyecto más expresivo que "Sin título" e inicia tu editor.

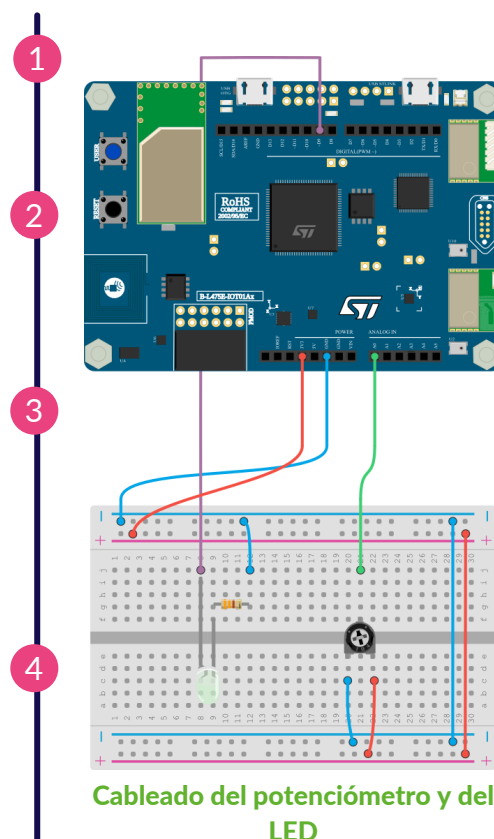
Fuente: makecode.lets-steam.eu

Programa tu placa

Dentro del Editor de Javascript de MakeCode, copia/pega el código disponible en la sección "**Programalo**" de abajo. Si no lo has hecho ya, da un nombre a tu proyecto y haz clic en el botón "**Descargar**". Copia el archivo binario en la unidad **DIS_L4IOT**, espera a que la placa termine de parpadear y ¡tu placa está lista!

Ejecuta, modifica, juega

Tu programa se ejecutará automáticamente cada vez que lo guardes o reinicies tu placa (pulsas el botón etiquetado como RESET). Intenta entender el ejemplo y empieza a modificarlo.





PROGRÁMALO



```
forever(function () {  
  pins.D9.analogWrite(pins.A0.analogRead())  
})
```

¿Cómo funciona?

El código consta de tres elementos:

- un bloque para siempre - **forever**
- un bloque de lectura analógica - **analogRead**
- un bloque escritura analógica - **analogWrite**

El bloque **forever** implementa "un bucle" que sigue ejecutando las instrucciones hasta que la placa se apaga.

El bloque **analogRead** se utiliza para obtener el valor del potenciómetro en el pin A0. Este valor es un número entero entre 0 y 1023. Al girar el mando se modifica el valor.



El potenciómetro actúa como un divisor de tensión ajustable. Cambiando la posición del mando, se modifica la tensión aplicada en A0. Cuanto más lo gires hacia la izquierda, más se acercará la tensión a 0V. Cuanto más lo gires hacia la derecha, más se acercará la tensión a 3,3V.



Un pin de entrada analógica puede ser utilizado para leer un valor entre 0 y 1023. Este valor es proporcional a la tensión aplicada al pin, que debe estar comprendida entre 0V y 3,3V.

El bloque **analogWrite** se utiliza para encender el LED en D9. Usando **analogWrite**, la placa es capaz de limitar el voltaje a un cierto valor para hacer que el LED brille más o menos. El brillo se establece por el valor de **analogRead** en el pin A0: cuanto más alto sea el valor, más brillante será el LED.



Utilizando el pin D9, podemos escribir un valor analógico a través de un pin digital en la placa. El pin D9, al igual que otros pines de la placa, admite la modulación por ancho de pulso o PWM. Esta técnica utiliza patrones de encendido y apagado para simular diferentes voltajes y por lo tanto diferentes señales analógicas. El valor pasado a **analogWrite** debe estar entre 0 y 255. El 0 representa un voltaje de 0V y el 255 un voltaje de 3,3V.

Como verás al utilizar este programa, no utilizarás todo el rango del potenciómetro. Puedes transformar el rango de valor del potenciómetro (0...1023) en el rango del PWM (0...255) con la función **map**.



MEJÓRALO



Al utilizar la función '**map**', intenta utilizar **todo el rango del potenciómetro**. Puedes definir dos variables para ser más expresivo y separar la lectura, la transformación y la escritura en una declaración específica.

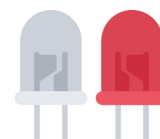
Añade otro LED e invierte el valor del potenciómetro, de modo que el segundo LED se atenúe a medida que se enciende el primero.

Utiliza el potenciómetro para **controlar el tono de un zumbador**. Utiliza un potenciómetro para **controlar la posición de un servo**.

1



2



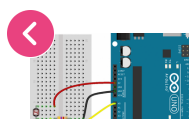
3



GOING FURTHER



- **Pulse Width Modulation** - Tutorial de Arduino sobre el uso de la salida analógica (PWM) para desvanecer un LED. <https://www.arduino.cc/en/Tutorial/Foundations/PWM>
- **Divisores de tensión** - Descubre cómo se comportan los divisores de tensión en el mundo real. <https://learn.sparkfun.com/tutorials/voltage-dividers>
- **Arduino pong game on 24x16 matrix with MAX7219** - Build a small pong console. <https://www.youtube.com/watch?v=dK9F5AJM2XI>
- **Juego con potenciómetro** - Controla el avatar de un juego mediante un potenciómetro. <https://www.hackster.io/matejadjukic03/potentiometer-game-05ee93?f=1#>



Fichas de actividades enlazadas

R1AS11 - Hacer un termómetro muy legible



R1AS15 - Recogida de datos

