

BOTONES Y PANTALLA LED

#R1AS03

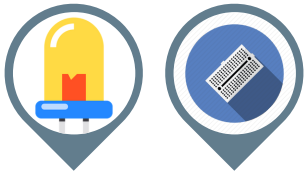


Available on



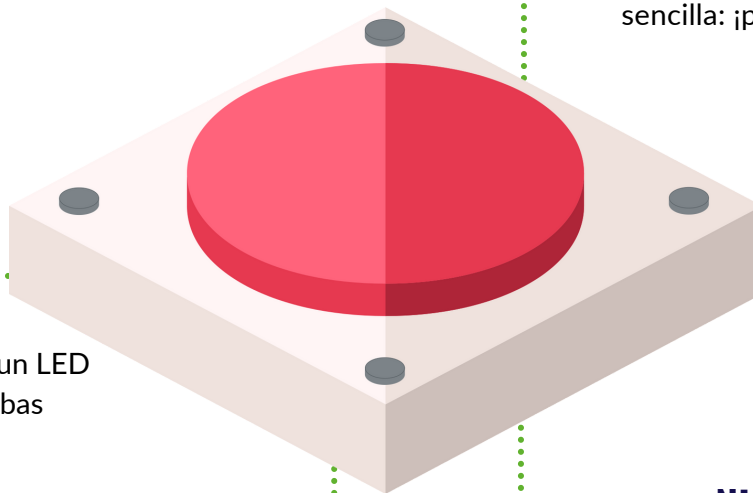
¿Qué es?

Aprenderemos a interactuar con la pizarra utilizando un simple pulsador. Los hay de muchas formas y tamaños, pero todos requieren la interacción más sencilla: ¡pulsarlo!



Requisitos previos

- R1AS01 - Parpadeo de un LED
- R1AS02 - Placa de pruebas



Duración

25 minutos

Material

- 1 Placa de programación "**STM32 IoT Node Board**"
- Cable USB Micro-B
- 2 Pulsadores
- 1 juego de LEDs
- 1 juego de resistencias
- Placa de pruebas (breadboard):
- Cables de puente

Nivel de dificultad

Intermedio

OBJETIVOS DE APRENDIZAJE

- Añadir interactividad
- Reaccionar a un evento en un botón físico
- Utilizar una variable para almacenar el estado actual del programa
- Cablear un circuito sencillo en una placa de pruebas con botones y LEDs
- Utilizar el simulador MakeCode



BOTONES Y PANTALLA LED



Para aprender a utilizar un botón, **juguemos a un juego de preguntas.**

La idea es bastante sencilla: **2 jugadores, un botón y un LED para cada uno.** Cuando el/la animador/a hace una pregunta, el/la jugador/a tiene que pulsar primero su botón para dar la respuesta correcta. Los LEDs indican qué jugador/a pulsa primero el botón y puede hablar.



HAZLO



Cableado de botones y LEDs

Conecta un lado de cada botón al **pin GND** de la placa. Luego conecta el otro lado en el pin **D0** para el/la jugador/a 1, y en el **pin D1** para el/la jugador/a 2. Conecta el ánodo del LED del/la jugador/a 1 en el pin **A0** y el del/la jugador/a 2 en el **pin A1**. Conecta el **cátodo** de cada LED a una resistencia (330 ohmios). A continuación, conecta las patas de las resistencias no conectadas a la patilla **GND**.

i El **LED** tiene una orientación. Para designar la orientación correcta, cada pata tiene un nombre. Así es como se encuentra la diferencia entre ánodo y cátodo: **Ánodo:** Es el "+" del LED. La pata del ánodo es más larga que la del cátodo. **Cátodo:** Es el "-" del LED. La pata del cátodo es más corta que la del ánodo.

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 "**Prográmalo**" de abajo. Si no lo has hecho ya, da un nombre a tu proyecto y haga clic en el "Descargar". Copia el archivo binario en la unidad DIS_L4IOT, espera hasta que la placa termine de parpadear, ¡tu zumbador de pruebas está listo!

Ejecuta, modifica, juega

Tu programa se ejecutará automáticamente cada vez que lo guardes o reinicies tu placa (pulsas el botón etiquetado como RESET). Intenta comprender el ejemplo y empieza a modificarlo cambiando el periodo entre dos sesiones de juego. No dudes en utilizar esta hoja de actividades para hacer que un cuestionario de evaluación dentro del aula sea más atractivo y atractivo para tus alumnos.

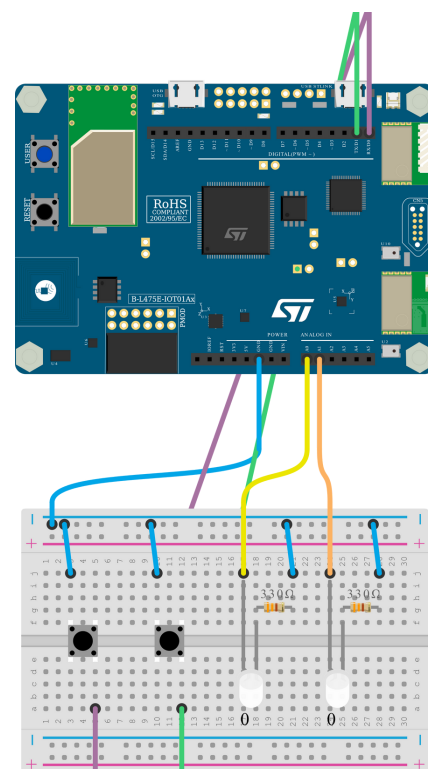
1

2

3

4

5



Cableado de botones y LEDs



PROGRÁMALO

```
//Initialization
let weCanPushIt = true
pins.A0.digitalWrite(false)
pins.A1.digitalWrite(false)
```

Inicialización

Como primer paso, necesitamos declarar una variable llamada **weCanPushIt**, de tipo booleano - una forma de datos con sólo dos valores posibles, normalmente **"true"** y **"false"**. Esta variable nos servirá para saber si podemos pulsar el botón, o si el otro jugador ya lo está haciendo. Las 2 últimas líneas dan la información de que todos los LEDs están apagados.



Una **variable** es una forma de nombrar y almacenar un valor para su uso posterior por el programa, como los datos de un sensor o un valor intermedio utilizado en un cálculo. La variable tiene un nombre y un tipo. El tipo permite especificar qué tipo de datos puede contener la variable.

```
input.buttonD0.onEvent(ButtonEvent.Down, function () {
  if (weCanPushIt) {
    weCanPushIt = false
    pins.A0.digitalWrite(true)
    pause(3000)
    pins.A0.digitalWrite(false)
    weCanPushIt = true
  }
})

input.buttonD1.onEvent(ButtonEvent.Down, function () {
  if (weCanPushIt) {
    weCanPushIt = false
    pins.A1.digitalWrite(true)
    pause(3000)
    pins.A1.digitalWrite(false)
    weCanPushIt = true
  }
})
```

Interacciones

El código principal se refiere a las interacciones de los botones realizadas con las funciones **input.buttonXX.onEvent**.



Una **función** es un bloque de código que ejecuta una tarea específica. Es realmente útil para simplificar el código y hacer un bloque de código más expresivo.

La línea más importante aquí es la condición **if (weCanPushIt) { ... }** que comprueba si los jugadores ya han pulsado su botón o no lo han hecho todavía. Si este es el caso (**weCanPushIt** es igual a **true**), nosotros:

1. Poner **weCanPushIt** a **false**, para no permitir que el oponente pulse su botón.
2. Enciende el LED del jugador para mostrar quién es el ganador
3. Espera 3 segundos (3.000 milisegundos)
4. Apagar el LED del ganador
5. Establece **weCanPushIt** a **true**, para permitir a los jugadores pulsar sus botones.

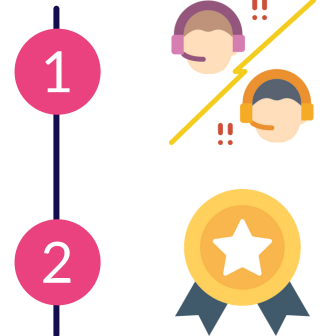


MEJÓRALO



Añade otros botones y LEDs y modifica tu programa para jugar con más jugadores.

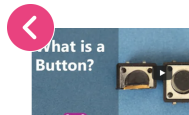
Modifica tu programa para que **el LED del/la ganador/a parpadee** utilizando la hoja de actividades "blink a led".



¿QUIERES IR MÁS ALLÁ?



- **Pulsador** - Conoce los usos del pulsador.
<https://en.wikipedia.org/wiki/Push-button>
- **Detrás del hardware de MakeCode** - Botones en micro:bit - Todo sobre los botones y su uso en MakeCode con Shawn Hymel, creador de contenido técnico.
https://www.youtube.com/watch?v=t_Qujld_38o, <https://shawnhymel.com>
- **Juego de reacción** - Haz un juego de reacción con interruptores físicos reales que puedes golpear tan fuerte como quieras!
<https://microbit.org/projects/make-it-code-it/reaction-game/>
- **Descubre qué es una variable** - Aprende más sobre las variables y ¿Qué es una función en la programación? - Aprende más sobre la función.
<https://www.computerhope.com/jargon/v/variable.htm>, <https://www.makeuseof.com/what-is-a-function-programming/>



Fichas de actividades enlazadas

R1AS04 - Sensor de luz básico

