Generación IA: 2 proyectos micro:bit

Nivel: intermedio

José Antonio Pascual del Coso

PALMITO BOOKS

Generación IA: proyectos micro:bit

Nivel: intermedio

José Antonio Pascual del Coso

Palmito Books



Título: Generación IA: proyectos micro: bit.

Subtítulo: Nivel: intermedio

© José Antonio Pascual del Coso, 2026

Reservados todos los derechos

De acuerdo con lo dispuesto en el art. 270 del Código Penal, podrán ser castigados con penas de multa y privación de libertad quienes reproduzcan o plagien, en todo o en parte, una obra literaria, artística o científica, fijada en cualquier tipo de soporte sin la preceptiva autorización.

Palmito Books®

Publicado en formato CD-ROM

1ª edición: octubre 2026

ISBN: 979-13-88535-91-8

Depósito Legal: D.L. MU 1828-2026



A Guillerma Muñoz Zurita

Índice

1. Conectando con el mundo exterior 11
2. Fundamentoз de electrónica digital y analógica 11
3. Anatomía de loз conectoreз 13
4. El Laboratorio virtual en Tinkercad 15
5. Seguridad electrónica 16
6. Organización de loз proyectoз prácticoз 17
7. De la зimulación en Tinkercad a la tarjeta fíзica 18
Proyecto 1: Semáforo inteligente 19
Proyecto 2: Botón de marcha y paro con LEDз 23
Proyecto 3: Luz intermitente regulable 28
Proyecto 4: Siзtema de зeguridad perimetral 32
Proyecto 5: Iluminación de diзcoteca aleatoria 36
Gimnaзia mental: Crucigrama 40
Proyecto 6: Termómetro con LEDз 41
Proyecto 7: Control de acceзo a aparcamiento 45
Proyecto 8: Señalizador de falta de agua 48
Proyecto 9: Ventilador inteligente 51
Proyecto 10: Ventilador automático 54
Reto 1: Riego para invernadero 58
Reto 2: Control de зervomotor con potenciómetro 59
Gimnaзia mental: Soluciones 60
Reto: Soluciones 61

1. Conectando con el mundo exterior

En el primer volumen de esta colección nos enfocamos en explotar las capacidades autónomas de la placa micro:bit. A lo largo de sus proyectos, aprendimos a programar y sacar partido a sus sensores integrados, su matriz de luces LED, sus botones físicos y sus sistemas de comunicación de radio. Sin embargo, el verdadero potencial de la robótica, la mecatrónica y la ingeniería del entorno surge cuando permitimos que la tarjeta interactúe directamente con dispositivos externos.

En esta segunda entrega, correspondiente al Nivel intermedio, daremos el salto del software puro a la integración electrónica. La micro:bit dejará de ser únicamente una unidad de proceso aislada para convertirse en el "cerebro" central de un sistema físico capaz de encender luces, hacer sonar alarmas, mover mecanismos mediante motores, etc. Para mantener un ritmo de aprendizaje accesible, respetaremos el principio de simplicidad: cada proyecto de este volumen incorporará un único componente externo nuevo (o a lo sumo dos), garantizando un avance seguro, sin saturar la comprensión del circuito ni la lógica de programación.

Para diseñar nuestros proyectos seguiremos utilizando Tinkercad, haciendo el paso al mundo físico muy sencillo al tratarse de un entorno visual.





2. Fundamentos de electrónica digital y analógica

Para trabajar con periféricos externos sin poner en riesgo los componentes, es necesario comprender cómo viaja la corriente eléctrica y cómo procesa la micro:bit la información del entorno.

- Señales Digitales vs. Analógicas:
    - Señal Digital: Solo puede tomar dos estados posibles: 1 / ALTA / 3.3V (activado) o 0 / BAJA / 0V (desactivado). Se utiliza para activar o desactivar elementos de forma rotunda (por ejemplo, encender o apagar un LED).

Estado lógico	Voltaje	Interpretación micro:bit	Ejemplo
1 (ALTA/HIGH)	3.3 V	Encendido/ Verdadero	Encender una luz o detectar botón pulsado.
0 (BAJA/LOW)	0 V	Apagado/ Falso	Apagar una luz o detectar botón sin pulsar.

    - Señal Analógica: Puede tomar un rango continuo de valores intermedios. En la micro:bit, las lecturas analógicas de entrada se transforman digitalmente en una escala numérica que va desde el valor 0 (para 0V) hasta el 1023 (para 3.3V). Esto permite medir magnitudes graduadas, como el nivel exacto de luz de una habitación o la posición de un potenciómetro.
- Entradas y Salidas PWM (Pulse Width Modulation): La tarjeta micro:bit no puede generar voltajes analógicos reales de salida, pero utiliza la técnica PWM (modulación por ancho de pulsos) en sus pines para simular una salida analógica. Varía la rapidez con la que conmuta entre 0V y 3.3V (con valores entre 0 y 1023) para regular la intensidad con la que brilla un LED o la velocidad a la que gira un motor.
- Magnitudes Eléctricas Básicas:
    - Voltaje (V): La "fuerza" o tensión con la que se empujan los electrones a través del circuito (la micro:bit trabaja a un voltaje de 3.3V).
    - Corriente o Intensidad (I): La cantidad de electrones que fluyen por el

circuito en un tiempo determinado, medida en amperios (A) o miliamperios (mA).

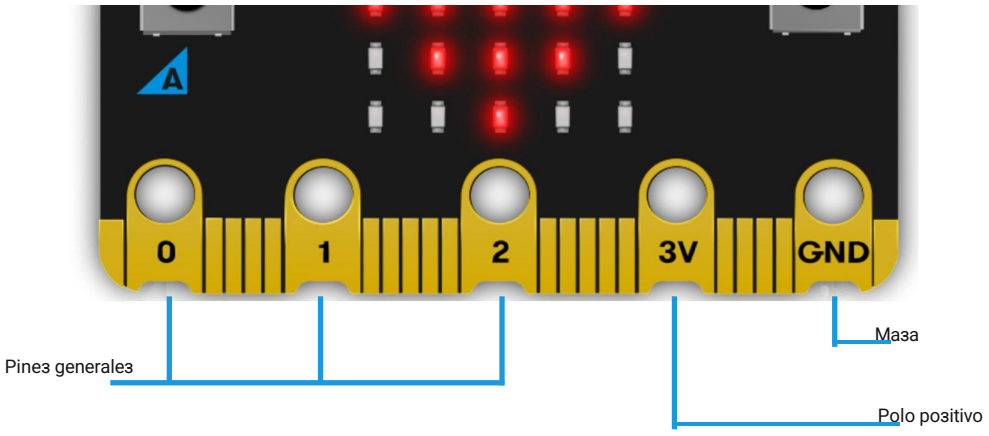
- Resistencia (R): La oposición o freno que ofrece un material al paso de la corriente, medida en ohmios (Ω). La Ley de Ohm establece la relación entre estas tres magnitudes mediante la fórmula:

$$V = I * R$$

3. Anatomía de los conectores

Para poder cablear circuitos con garantías de funcionamiento, debemos aprender a utilizar la zona de conexiones de la micro:bit y los elementos auxiliares de montaje.

Si observas el borde inferior dorado de tu tarjeta micro:bit, verás 25 franjas metálicas llamadas Pines de Entrada/Salida (GPIO). En este volumen nos centraremos en los pines principales marcados con orificios grandes.

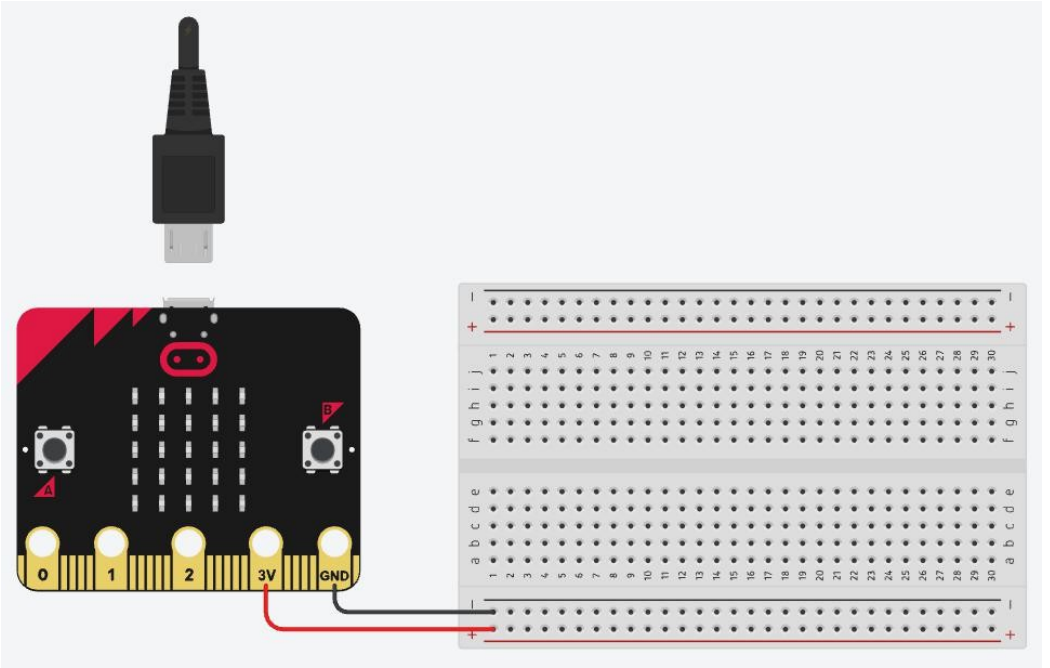


Los pines marcados como 0, 1 y 2 son los generales, que permiten la conexión directa con pinzas de cocodrilo o cables jumper para leer sensores o controlar actuadores.

Para suministrar corriente continua desde la propia tarjeta a los componentes

externos usaremos los pines marcados como 3V y GND, siendo el polo positivo y la masa, respectivamente.

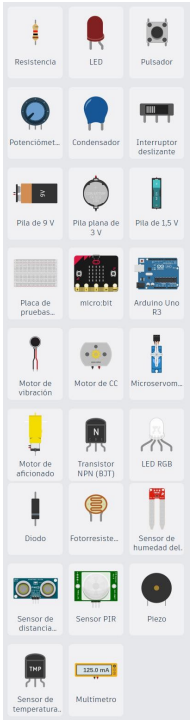
Otro elemento que se hace necesario conocer, aunque no esencial para nuestros proyectos, es la Protoboard (o placa de prueba). Se trata de una matriz de plástico repleta de orificios que te permite montar y modificar circuitos sin necesidad de soldar cables, lo que la convierte en una herramienta esencial para montar circuitos de prototipado rápido. En su interior oculta tiras metálicas con pestañas que atrapan las patillas de los componentes, actuando como cables de unión entre ellos.



Las pistas centrales están organizadas en filas con letras (a, b, c, d, e, f, g, h, i, j). Todos los orificios de una misma fila de 5 puntos están unidos internamente por una tira metálica en sentido vertical.

Los raíles de alimentación laterales, identificadas con una línea Roja (+) y una línea Negra (-). Todos los orificios a lo largo de esa franja están unidos en sentido horizontal. Se usan para llevar el voltaje (3.3V) y la masa (GND) a cualquier parte de la placa.

4. El Laboratorio virtual en Tinkercad



Antes de tocar hardware real, usarás Tinkercad Circuitos. El simulador te permite experimentar de forma 100% segura: si cometes un error grave de cableado, el simulador te avisará en pantalla con un icono de advertencia sin riesgo de quemar ningún componente real.

El módulo Circuitos de Tinkercad simula el comportamiento físico y eléctrico de los componentes reales.

Desde el panel lateral derecho arrastra la tarjeta micro:bit, la protoboard y los distintos componentes que requiera cada uno de los proyectos propuestos. Todos estos elementos existen físicamente y son muy fáciles de conseguir, de manera que el proyecto lo podrás llevar al mundo físico sin ningún problema.

Para crear un cable, se hace clic sobre la patilla de un componente o sobre un pin de la micro:bit y se arrastra el ratón hasta el orificio de destino.

Al hacer clic en puntos intermedios del área de trabajo, el cable creará ángulos rectos, manteniendo el esquema limpio y legible.

Para evitar errores de montaje y facilitar el diagnóstico de averías, mantendremos la convención del Código internacional de colores para el cableado:

- Rojo: Exclusivamente para cables conectados a la línea de alimentación de 3.3V.
- Negro: Exclusivamente para cables conectados a masa (GND).
- Resto: Para señales de datos digitales, salidas PWM o lecturas analógicas procedentes de los pines 0, 1 o 2. La elección dependerá solo para un mejor estudio del circuito.



Si quieres que los próximos cables que vayas a trazar nazcan directamente con el color adecuado, selecciona en la barra superior la opción que tiene un cuadrado con el color de los cables actuales. Al hacer clic, se despliega la paleta de colores puedes seleccionar el color que desees.

5. Seguridad electrónica

Como ya hemos dicho, el uso de simulación previa en Tinkercad previene daños materiales. Para garantizar que los programas y montajes también funcionen correctamente en el mundo físico sin deteriorar la tarjeta, respetaremos tres reglas fundamentales:

- Utiliza siempre una resistencia protectora con los LEDs, ya que un LED conectado directamente entre 3.3V y GND sin resistencia consumirá un caudal ilimitado de corriente. Esto puede fundir el LED instantáneamente o deteriorar la micro:bit.
- Nunca conectes un cable directamente desde el pin 3.3V hasta el pin GND sin pasar por un componente que ofrezca resistencia. Si los unes directamente, crearás un cortocircuito que desactivará la alimentación de tu ordenador o podría dañar la tarjeta.
- Algunos componentes necesitan que los 3.3V y la GND vayan a algunos pines concretos, por lo que sus conectores están destacados con los colores rojo y negro, respectivamente. Antes de hacer clic en "Iniciar simulación" o conectar el cable USB a la tarjeta real, comprueba que los componentes con polaridad (LEDs, zumbadores, servomotores, ...) estén orientados en el sentido correcto.

6. Organización de los proyectos prácticos

Cada uno de los proyectos de este libro sigue una estructura rigurosa dividida en los siguientes capítulos:

- Dificultad y tiempo: Indicadores visuales de la dificultad, mediante una escala de estrellas, y el tiempo estimado de realización en minutos para planificar las sesiones de trabajo.
- Componentes necesarios: Lista detallada de los elementos de hardware requeridos para realizar el proyecto. Es importante para saber qué componentes hemos de usar en Tinkercad, pero aún es más importante si se va a realizar el proyecto de manera física.
- Aprendizaje: Se indica de forma clara qué conceptos teóricos o destrezas de programación nuevas se van a adquirir y que no se hayan visto en prácticas anteriores.
- Conexionado: Ilustración clara e instrucciones paso a paso para cablear los componentes en la protoboard y conectarlos a la tarjeta dentro de Tinkercad.
- Programación: Guía paso a paso con el procedimiento claro y detallado sobre cómo colocar los bloques de código y configurar las opciones de Tinkercad. Aunque se ofrece la representación exacta de los bloques para verificar la implementación, es recomendable seguir el procedimiento paso a paso para una mayor asimilación.
- Repaso: Un resumen sintético que consolida los puntos clave trabajados en el proyecto antes de pasar al siguiente.
- Desafío: Un reto planteado sin la solución explícita, diseñado para forzar la reflexión, el pensamiento crítico y la creatividad mediante la modificación del código original.

7. De la simulación en Tinkercad a la tarjeta física

Una de las mayores ventajas de programar con Tinkercad es que todo el código desarrollado dentro de la plataforma virtual puede ser transferido de forma inmediata a una tarjeta micro:bit real. El procedimiento para volcar tu trabajo de la pantalla al mundo físico es el siguiente:

- En el panel de código de Tinkercad, haz clic en el botón de descarga. El navegador guardará en tu ordenador un archivo con extensión .hex que contiene todas las instrucciones traducidas para el procesador de la placa.
- Conecta la tarjeta micro:bit física a tu ordenador utilizando un cable micro-USB. El sistema operativo la reconocerá automáticamente como si fuera una unidad de almacenamiento externo USB llamada "MICROBIT".
- Arrastra el archivo .hex descargado directamente dentro de la unidad "MICROBIT".
- Mientras el archivo se transfiere, la luz LED trasera de la tarjeta parpadeará rápidamente. En cuanto finalice el parpadeo, la tarjeta se reiniciará automáticamente y comenzará a ejecutar el programa que diseñaste y probaste en el simulador de Tinkercad.
- A partir de ahora, cada vez que se alimente la tarjeta micro:bit a través del puerto USB (con un cargador, pilas, ordenador,...), comenzará la ejecución del programa transferido.

Proyecto 1: Semáforo inteligente

Dificultad: ★★★☆☆ Tiempo: 40 m

Componentes

- 1 Placa micro:bit
- 1 Placa de pruebas (protoboard)
- 1 LED rojo
- 1 LED amarillo
- 1 LED verde
- 3 resistencias 220 Ω (rojo- rojo- marrón- oro)
- Cables de conexión

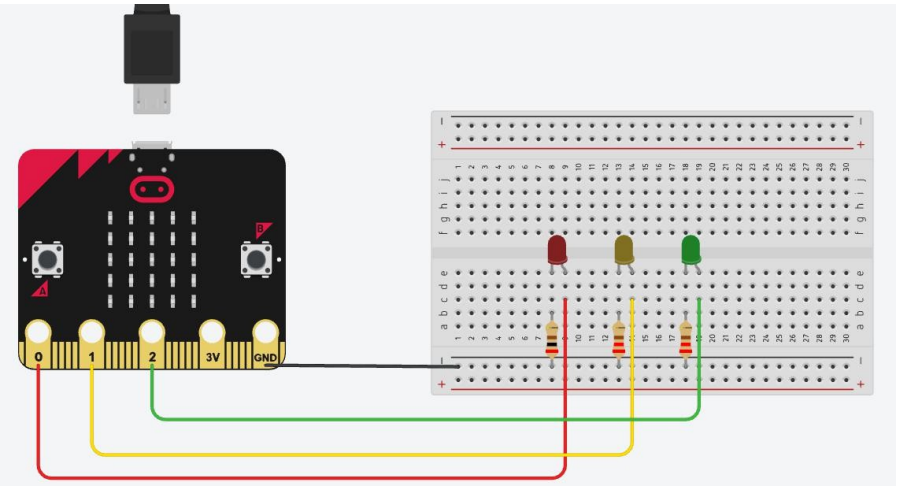
Aprendizaje

- ¿Qué es y cómo funciona un LED?: Un LED es un componente electrónico que emite luz cuando circula electricidad a través de él. Tiene polaridad: una patilla más larga llamada ánodo (positiva) y una patilla más corta llamada cátodo (negativa). Si se conecta al revés, no funcionará.
- ¿Por qué necesitamos Resistencias?: Un LED no puede conectarse directamente a la energía de la tarjeta sin protección porque recibiría demasiada intensidad de corriente y se quemaría. La resistencia es un componente que limita o frena el paso de la electricidad para proteger los componentes. Se mide en ohmios (Ω).
- Gestión de Pines Digitales de Salida (`escribir de forma digital el pin P0`): Aprenderemos a enviar órdenes desde el código para alimentar con electricidad (3.3 V, valor `ALTA`) o cortar el paso de corriente (0 V, valor `BAJA`) en pines específicos de la micro:bit.

Conexionado

1. Arrastrad a la zona de trabajo una tarjeta micro:bit y una Placa de pruebas.
2. Conectad el pin GND (masa) de la micro:bit al carril negativo (marcado con una línea negra y un signo -) de la placa de pruebas mediante un cable de color negro.
3. Colocad los tres LED en la placa de pruebas, asegurándoos de que sus patillas no compartan la misma línea de conexión vertical.

4. Conectad una reзistencia de 220 Ω deзde la patilla corta (cátodo) de cada LED haзta el carril negativo general (GND).
5. Conectad la patilla larga (ánodo) del LED Rojo al pin P0 de la micro:bit con un cable rojo.
6. Conectad la patilla larga (ánodo) del LED Amarillo al pin P1 de la micro:bit con un cable amarillo.
7. Conectad la patilla larga (ánodo) del LED Verde al pin P2 de la micro:bit con un cable verde.



Programación

Paзo 1: Inicializar el eзtado de loз componenteз al arrancar

En la categoría "● Salida", localizad el bloque `escribir de forma digital el pin P0 en ALTA`. Colocad treз unidadeз dentro del bloque `al iniciar`, haciendo cada uno referencia a un pin diзtinto: P0, P1 y P2. Todoз han de tener el valor `BAJA`. Con ello garantizaremoз que todoз loз LED comiencen apagadoз.

Añadid el bloque `borrar pantalla` para mantener la matriz LED limpia.

Paзo 2: Programar la зecuencia del зemáforo en bucle

Dentro del bucle principal `siempre`, colocad la orden `escribir de forma digital el pin P2 en ALTA` para encender el LED Verde.

Añadid un bloque `esperar 1 segundos`, cambiadlo a 4 segundos para que la luz verde permanezca activa durante ese tiempo.

A continuación, colocad `escribir de forma digital el pin P2 en BAJA` para apagar el LED Verde, y colocad `escribir de forma digital el pin P1 en ALTA` para que se encienda el LED Amarillo.

Ahora añadid una espera de 2 segundos para que se mantenga en amarillo.

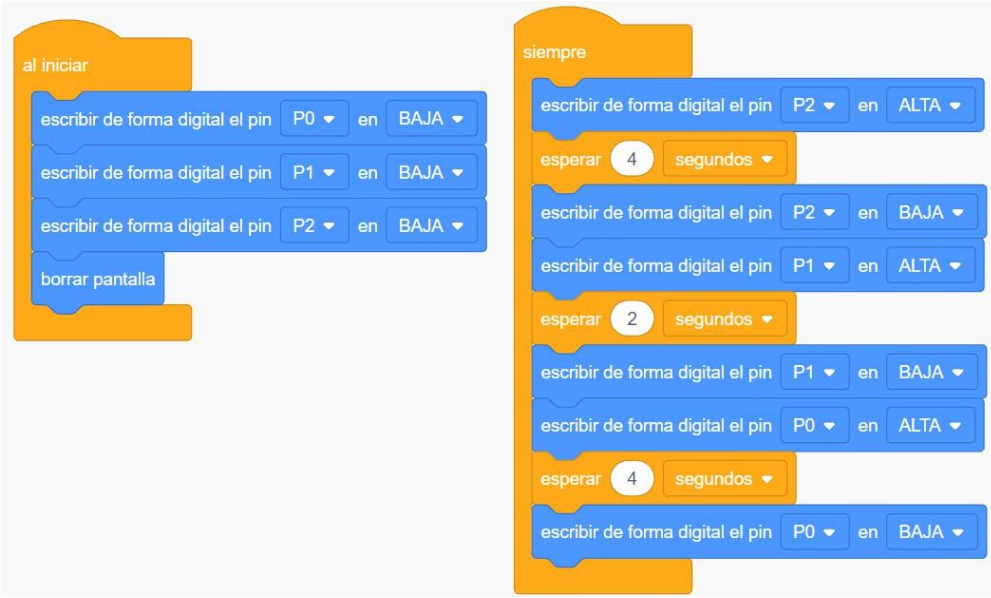
Para que luego se ponga en rojo, colocad `escribir de forma digital el pin P1 en BAJA` (apaga el LED Amarillo) y `escribir de forma digital el pin P0 en ALTA` (enciende el LED Rojo).

Mantened encendido el semáforo en rojo durante 4 segundos.

Colocad `escribir de forma digital el pin P0 en BAJA` (apaga el LED Rojo) justo antes de finalizar el bucle `siempre`, para que al reiniciar la secuencia el LED Verde vuelva a encenderse correctamente.

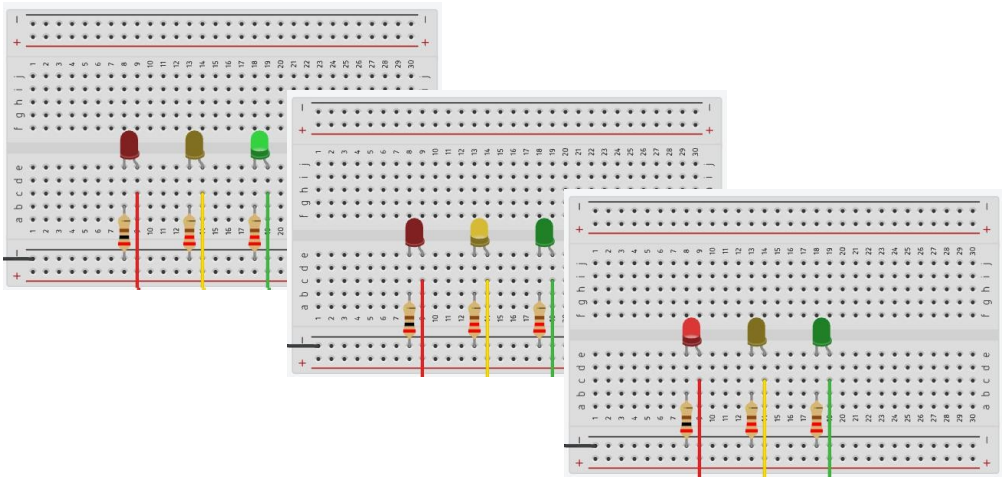
Paso 3: Simulación y comprobación en Tinkercad

Haced clic sobre el botón "Iniciar simulación". Verificad que la secuencia se repite indefinidamente sin solaparse ninguna luz.



Repaso

- Pinez digitalez de zalida (`escribir de forma digital el pin`): Permiten a la tarjeta enviar órdenez eléctricaz de ENCENDIDO (`ALTA` o 3.3V y APAGADO (`BAJA` o 0V) a componentez externoz.
- Circuitoz con componentez externoz: Comprendemoz la función ezencial del LED como emizor de luz, de la rezizтencia como protector frente a zobretenzionez y de la maza común (GND) para cerrar el circuito eléctrico.
- Valorez por defecto en laz rezizтenciaz: Al trabajar en Tinkercad con rezizтenciaz, prezтad ezpecial atención a zu valor. Por defecto, Tinkercad coloca rezizтenciaz de 1 kΩ (1000 Ω). Si en ezte proyecto no cambiáiz ezte valor a 220 Ω en el menú flotante del componente, la rezizтencia frenará tanto el pazo de corriente que loz LEDz ze iluminarán con muy poca intenzidad o parecerán completamente apagadoz.
- Secuenciación temporal de procezoz: El uzo combinado de activacionez de pinez y retardoz (`esperar ... segundos`) permite coordinar automatizmoz complejoz del mundo real.



Desafío

¿Cómo podríaiz convertir ezte zemáforo zimple en un zemáforo con avizo de precaución parpadeante en la luz amarilla antez de cambiar a rojo?

Intentad zuztituir la pauza fija del LED Amarillo por un parpadeo del pin P1. Recordad hacer una pauza mientraz ezté encendido y otra mientraz ezтá apagado.

Proyecto 2: Botón de marcha y paro con LEDs

Dificultad: ★★★☆☆ Tiempo: 35 m

Componentes

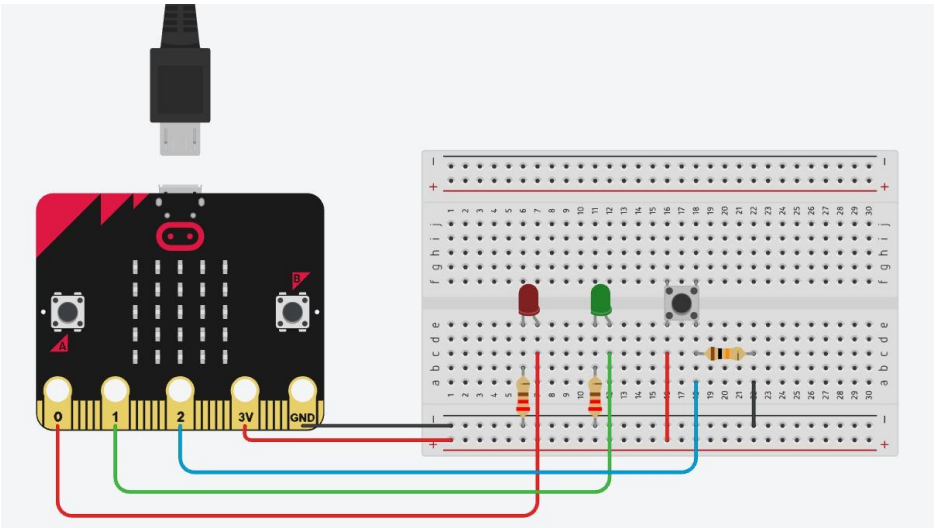
- 1 Placa micro:bit
- 1 Placa de pruebas (protoboard)
- 1 LED rojo
- 1 LED verde
- 2 resistencias 220 Ω (rojo- rojo- marrón- oro)
- 1 resistencia 10 kΩ (marrón- negro- naranja- oro)
- 1 pulsador de 4 patillas
- Cables de conexión

Aprendizaje

- ¿Qué es y cómo funciona un Pulsador?: Un pulsador es un componente mecánico de entrada que abre o cierra el paso de la corriente eléctrica únicamente mientras se mantiene presionado. Dispone de cuatro patillas conectadas en pares internos; al pulsar, un pequeño puente metálico une ambas líneas permitiendo el paso de corriente.
- ¿Qué es y para qué sirve una Resistencia de Pull-Down?: Cuando un pin de entrada digital de la micro:bit está desconectado o el botón no se presiona, el pin se encuentra en una situación inestable (ni 0 ni 1), pudiendo captar interferencias electromagnéticas que provocan lecturas erróneas. Para evitarlo, se utiliza una resistencia de 10 kΩ conectada a masa. Esta resistencia "tira hacia abajo" (pull-down) del voltaje garantizando un 0 lógico constante hasta que pulsamos el botón y enviamos 3.3 V (1 lógico). Por tanto, no se trata de un modelo distinto de resistencia, sino que la función que cumple es la de asegurar que en la entrada siempre haya un 0 o un 1.
- Lectura digital de pines (`leer pin digital P2`): Descubriremos cómo la tarjeta micro:bit consulta si un pin de entrada recibe energía (estado 1 / verdadero) o si no recibe corriente (estado 0 / falso).
- Variables de estado: Crearemos una variable que actuará como la "memoria" del sistema, recordando en todo momento si el automatismo se encuentra en modo MARCHA o en modo PARO.

Conexionado

1. Colocad en la zona de trabajo una micro:bit y una Placa de pruebas.
2. Conectad el pin GND de la micro:bit al carril de masa (marcado con la línea negra y un signo -) de la placa de pruebas mediante un cable negro.
3. Conectad el pin 3V de la micro:bit al carril de alimentación positiva (marcado con la línea roja y un signo +) de la placa de pruebas mediante un cable rojo.
4. Insertad los dos LED en la placa de pruebas. Conectad una resistencia de 220 Ω desde la patilla corta (cátodo) de cada LED hacia el carril de masa.
5. Conectad la patilla larga (ánodo) del LED Rojo al pin P0 de la micro:bit mediante un cable rojo.
6. Conectad la patilla larga (ánodo) del LED Verde al pin P1 de la micro:bit mediante un cable verde.
7. Insertad el Pulsador cruzando la ranura central de la placa de pruebas.
8. Conectad una de las patillas del lado izquierdo del pulsador al carril positivo de 3V.
9. Conectad la patilla del lado derecho del pulsador al pin P2 de la micro:bit mediante un cable azul.
10. Conectad esa misma patilla del lado derecho del pulsador hacia el carril de masa pasando por la resistencia de 10 kΩ (pull-down).



Programación

Paso 1: Inicializar las variables y el estado inicial del sistema

Al encender la tarjeta, el sistema debe arrancar obligatoriamente en modo de seguridad (sistema en PARO, indicación roja activa):

Crear una variable llamada "en_marcha" en la categoría "● Variables".

En el bloque `al iniciar`, colocad la orden `definir en_marcha en 0` (donde 0 significa estado de Paro y 1 estado de Marcha).

Asegurad el estado de iluminación inicial fijando el pin P0 (LED Rojo) en ALTA y el pin P1 (LED Verde) en BAJA con las órdenes `escribir de forma digital el pin P0 en ALTA` y `escribir de forma digital el pin P1 en BAJA`.

Añadid el bloque `borrar pantalla` para mantener la pantalla limpia.

Paso 2: Detectar la pulsación del botón e invertir el estado

En el ciclo continuo comprobaremos si el usuario presiona el botón para conmutar el funcionamiento:

Dentro del bucle principal `siempre`, colocad un condicional `si verdadero entonces`.

Sustituid la condición `verdadero` por una comparación numérica que evalúe si `leer pin digital P2` de la categoría "● Entrada" es igual a 1.

Dentro de la sección `entonces` (se ha pulsado el botón):

Evaluad el estado actual con un bloque interno `si verdadero entonces ... si no` comprobando si `en_marcha = 0`.

En la rama `entonces` poned aquello que hará el sistema cuando esté en marcha.

Primero indicadlo con `definir en_marcha en 1`.

A continuación, colocad `escribir de forma digital el pin P0 en BAJA` (apaga el LED Rojo) y `escribir de forma digital el pin P1 en ALTA` (enciende el LED Verde).

En la rama `si no` irá lo que hará cuando el sistema está apagado.

Colocad `definir en_marcha en 0`.

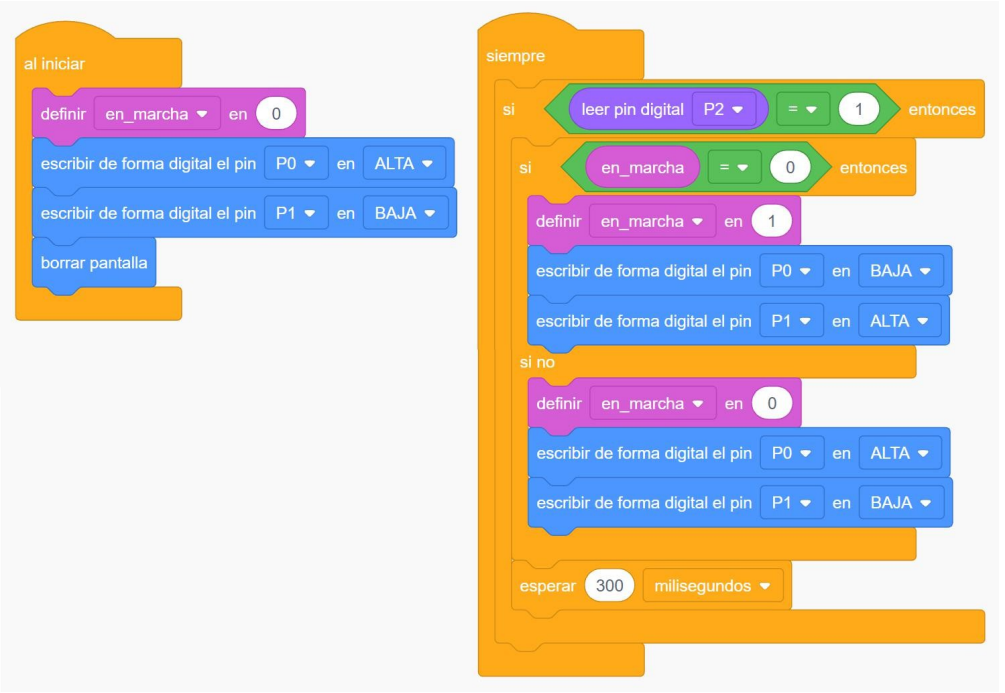
Colocad `escribir de forma digital el pin P0 en ALTA` (enciende el LED Rojo).

Colocad `escribir de forma digital el pin P1 en BAJA` (apaga el LED Verde).

Para evitar que un único toque active y desactive el sistema docenas de veces seguidas por la alta velocidad del procesador, añadid al final del condicional principal la orden `esperar 300 milisegundos` (antirrebote por software).

Paso 4: Simulación y comprobación en Tinkercad

Haced clic sobre el botón "Iniciar simulación". Al arrancar el programa, comprobaréis que únicamente el LED Rojo conectado al pin P0 permanece encendido (indicando estado de Paro). Haced clic con el ratón sobre el pulsador externo montado en la placa de pruebas: veréis cómo el LED Rojo se apaga inmediatamente y se enciende el LED Verde del pin P1 (indicando estado de Marcha). Haced un segundo clic sobre el botón para verificar que el sistema regresa de forma estable al estado de Paro.



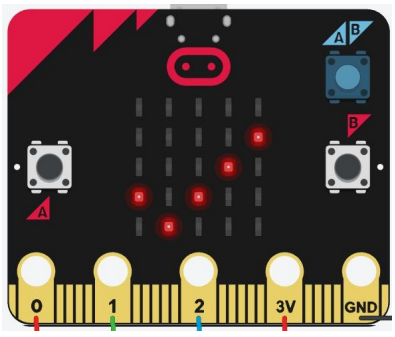
Repaso

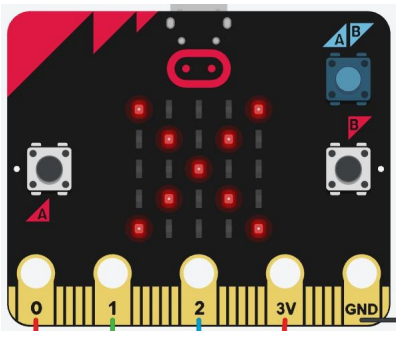
- Lectura de entradas digitales (`leer pin digital`): Permite a la micro:bit consultar el estado eléctrico de un pin externo para reaccionar ante eventos físicos como la pulsación de un botón.
- Resistencia de Pull-Down: Garantiza que un pin de entrada reciba un nivel de voltaje 0 claro y sin ruidos electromagnéticos mientras el circuito de entrada está abierto.
- Variables de estado: Utilizar variables para almacenar el modo de trabajo actual permite alternar entre dos estados operativos (MARCHA / PARO) con un único botón físico.
- Retardo tras lectura del botón: El pequeño retardo de `esperar 300 milisegundos` tras la lectura del botón es fundamental en la programación de controladores reales. Los botones mecánicos sufren micro-vibraciones al presionarse que el microprocesador puede interpretar como decenas de pulsaciones consecutivas en pocos milisegundos (efecto rebote). Con este breve tiempo de espera aseguramos que una sola presión del alumno equivalga exactamente a un único cambio de estado.

Desafío

¿Cómo podríais añadir a esta estación de control una matriz de señalización en la pantalla de la tarjeta que acompañe el estado de los LED externos?

Intentad que en la rama del estado de MARCHA la micro:bit muestre una flecha o un icono de avance (como un Check) en la pantalla mediante el bloque `mostrar icono`, y que en la rama del estado de PARO dibuje una cruz (X).





Proyecto 3: Luz intermitente regulable

Dificultad: ★★☆☆☆ Tiempo: 35 m

Componentes

- 1 Placa micro:bit
- 1 Placa de pruebas (protoboard)
- 1 Potenciómetro de 10 kΩ
- 1 LED rojo
- 1 resistencia 220 Ω (rojo- rojo- marrón- oro)
- Cables de conexión

Aprendizaje

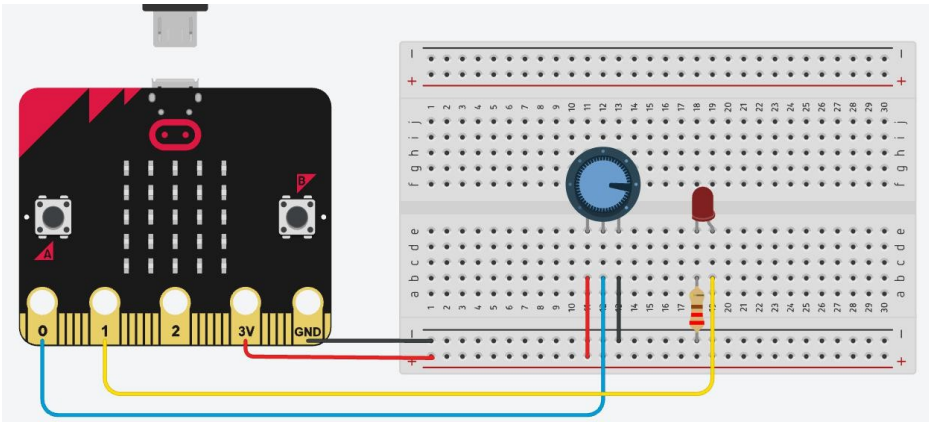
- ¿Qué es y cómo funciona un Potenciómetro?: Es una resistencia variable de tres terminales. Los extremos se conectan a la alimentación (3.3 V) y a masa, mientras que el terminal central (llamado cursor o wiper) entrega un voltaje variable según giramos la rueda de control.
- Conversión Analógica-Digital (ADC) y rango de lecturas: A diferencia de una lectura digital (que solo comprende 0 o 1), la micro:bit puede medir el nivel exacto de voltaje en sus pines analógicos. El conversor interno traduce esa tensión en un valor numérico continuo que oscila entre 0 (para 0 V y 1023 (para 3.3 V).
- Lectura analógica de pines (`leer pin analógico P0`): Aprenderemos a emplear el bloque de consulta analógica para capturar la posición del potenciómetro en tiempo real.
- Mapeo y escala matemática de variables: Entenderemos cómo transformar ese rango numérico amplio (0 a 1023) en un intervalo de tiempo de espera útil en milisegundos para ajustar la velocidad de parpadeo (frecuencia estroboscópica).

Conexionado

1. Colocad en la zona de trabajo una micro:bit y una Placa de pruebas.
2. Conectad el pin GND de la micro:bit al carril de masa (marcado con la línea negra y signo -) de la placa de pruebas mediante un cable negro.
3. Conectad el pin 3V de la micro:bit al carril positivo (marcado con la línea

roja y signo +) de la placa de pruebas mediante un cable rojo.

4. Insertad el Potenciómetro de 10 kΩ cruzando tres filas consecutivas de la placa de pruebas.
5. Conectad una de las patillas exteriores del potenciómetro al carril positivo de 3V mediante un cable rojo.
6. Conectad la otra patilla exterior del potenciómetro al carril de masa mediante un cable negro.
7. Conectad la patilla central (cursor) del potenciómetro al pin P0 de la micro:bit utilizando un cable azul.
8. Insertad el LED en la placa de pruebas. Conectad su patilla corta (cátodo) hacia el carril de masa pasando por la resistencia de 220 Ω.
9. Conectad la patilla larga (ánodo) del LED al pin P1 de la micro:bit mediante un cable amarillo.



Programación

Paso 1: Crear la variable de tiempo

Para manejar de forma dinámica la velocidad de conmutación del destello, guardaremos el valor del sensor en una variable. Dirigíos a la categoría ● Variables" y cread una variable llamada "tiempo_espera".

Paso 2: Capturar la entrada analógica y ajustar el retardo

Dentro del bucle de ejecución continua (`siempre`), leeremos el valor del

potenciómetro y se lo asignaremos a nuestra variable. Para ello, dentro del bucle principal `siempre`, colocad el bloque `definir tiempo_espera en`

Sustituid el valor 0 por el bloque `leer pin analógico P0` disponible en la categoría "● Entrada".

Paso 3: Generar la secuencia estroboscópica regulable

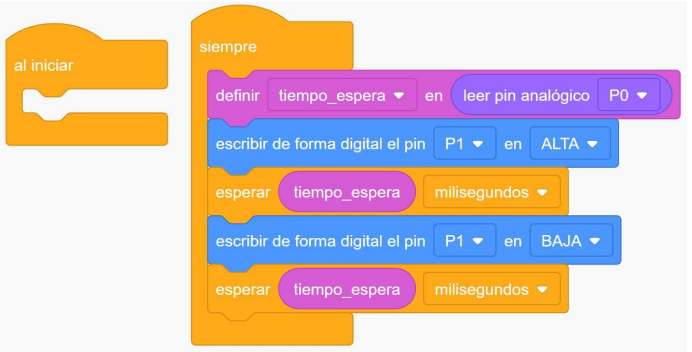
Justo debajo de la definición de la variable en el bucle `siempre`, colocad el bloque `escribir de forma digital el pin P1 en ALTA`.

Añadid el bloque de tiempo `esperar 1 segundos` sustituyendo el valor 1 por la variable `tiempo_espera` y cambiad segundos por milisegundos.

A continuación, colocad `escribir de forma digital el pin P1 en BAJA`, y otro bloque de espera idéntico utilizando la variable `tiempo_espera` para completar el periodo de intermitencia.

Paso 4: Simulación y comprobación en Tinkercad

Haced clic en el botón "Iniciar simulación". Veréis cómo el LED comienza a parpadear. Girad el potenciómetro con el ratón: observaréis cómo, a medida que alteráis el ángulo de giro, la velocidad de los destellos se acelera o se ralentiza de forma continua.

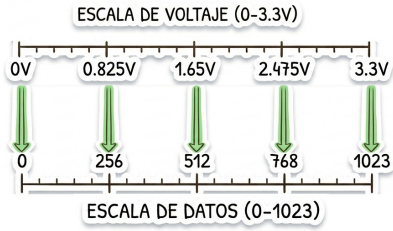


Repaso

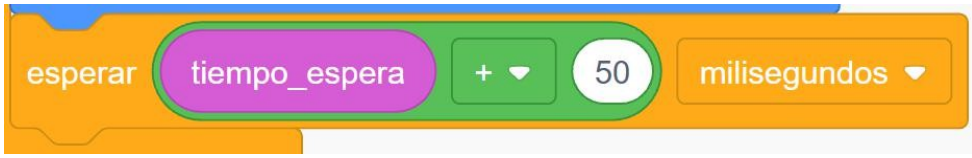
- Fundamento físico y divisor de tensión del potenciómetro: Un potenciómetro es esencialmente una pista de material resistente con un contacto móvil (wiper o cursor). Al aplicar 3.3 V en un extremo y 0 V en el otro, el potenciómetro actúa como un divisor de tensión. Según desplazamos o giramos el mando, variamos la longitud del material

resistivo que recorre la corriente hasta la patilla central, obteniendo una tensión proporcional al ángulo de giro (por ejemplo, a mitad de recorrido obtendremos exactamente la mitad de tensión, unos 1.65 V.

- Conversión Analógica-Digital y lectura (`leer pin analógico P0`): La micro:bit no lee directamente la tensión en voltios, sino que la transforma mediante su módulo ADC (Analog to Digital Converter) en un entero de 10 bits. Esto traduce el rango físico de 0 V a 3.3 V en una escala digital de 0 a 1023. Al usar `leer pin analógico P0`, capturamos al instante esta cifra para emplearla en nuestra lógica.



- Gestión dinámica de tiempos y límites visuales: Guardar la lectura en una variable nos permite ajustar la velocidad de parpadeo del LED en tiempo real. No obstante, tened en cuenta un detalle importante: si el potenciómetro entrega valores cercanos a 0, la velocidad de destello es tan alta que la retina humana no logra percibir el apagado debido a la persistencia retiniana (el LED parecerá quedarse fijo o atenuado). Si quisiéramos evitar este efecto o bloqueos por esperas nulas, es recomendable ajustar o sumar una constante base (por ejemplo, 50 milisegundos) a la lectura obtenida.



Desafío

¿Cómo podríais añadir una representación gráfica en la propia placa micro:bit que indique de forma visual el nivel de velocidad seleccionado sin pausar el parpadeo del LED externo?

Intentad que la matriz de 25 LEDs de la micro:bit dibuje un gráfico de barras proporcional a la velocidad empleando el bloque `trazar gráfico de barras desde 0 hasta 0`.

Investigad en internet su funcionamiento y qué valores tenéis que indicar.

Proyecto 4: Sistema de seguridad perimetral

Dificultad: ★★★☆☆ Tiempo: 45 m

Componentes

- 1 Placa micro:bit
- 1 Placa de pruebas (protoboard)
- 1 Sensor PIR (Sensor infrarrojo pasivo de movimiento)
- 1 Piezo (zumbador piezoeléctrico)
- 1 LED rojo
- 1 resistencia 220 Ω (rojo- rojo- marrón- oro)
- Cables de conexión

Aprendizaje

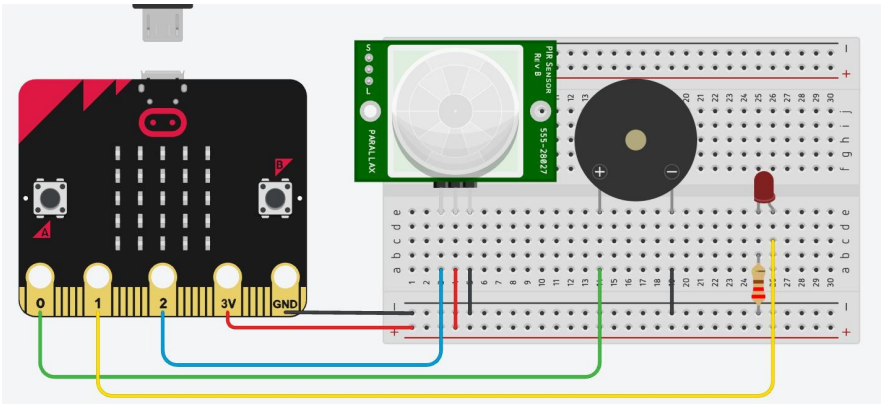
- Sensor PIR: Aprenderemos cómo funciona un sensor capaz de medir la radiación infrarroja de los cuerpos en movimiento. Comprenderemos sus tres terminales (señal, potencia y tierra) y cómo interpreta la presencia de una persona como un valor lógico digital en un pin de la micro:bit.
- Transducción acústica con zumbador piezoeléctrico: Entenderemos el comportamiento de las señales de audio activadas mediante modulación de frecuencias digitales y cómo generar tonos sonoros a través de salidas programables.
- Toma de decisiones mediante evaluación condicional (`si no`): Profundizaremos en el control de flujos ejecutando acciones compuestas simultáneas (sonido y luz) únicamente cuando se cumpla el estado de activación del sensor digital.

Conexionado

1. Colocad en la zona de trabajo una micro:bit y una Placa de pruebas.
2. Conectad el pin GND de la micro:bit al carril de masa (marcado con la línea negra y signo -) de la placa de pruebas mediante un cable negro.
3. Conectad el pin 3V de la micro:bit al carril positivo (marcado con la línea roja y signo +) de la placa de pruebas mediante un cable rojo.
4. Ubicad el Sensor PIR en la placa de pruebas. Conectad el terminal de potencia al carril de 3V con cable rojo, la patilla de tierra al carril de masa

con cable negro, y la patilla de señal al pin P2 de la micro:bit mediante un cable azul.

5. Insertad el Zumbador piezoeléctrico en la placa de pruebas. Conectad su patilla negativa al carril de masa y su patilla positiva al pin P0 de la micro:bit con un cable verde.
6. Insertad el LED rojo en la placa de pruebas. Conectad su patilla corta (cátodo) hacia el carril de masa a través de la resistencia de 220Ω. Conectad la patilla larga (ánodo) del LED rojo al pin P1 de la micro:bit mediante un cable amarillo.



Programación

Paso 1: Evaluar la señal del sensor PIR

Dentro del bucle principal `siempre`, insertad una estructura condicional `si verdadero entonces ... si no ...` desde la categoría "● Control".

Sustituid `verdadero` por una comparación matemática de igualdad (`1=1`).

En el primer espacio de la comparación, insertad el bloque `leer pin digital P2`. En el segundo espacio, dejad el valor 1 (que equivale a detección de movimiento).

Paso 2: Programar la secuencia de alerta (Rama del si)

Cuando el sensor detecte movimiento (en el pin P2 se recibe 1), activaremos las señales luminosa y sonora. Dentro de la primera sección del condicional, colocad el bloque `escribir de forma digital el pin P1 en ALTA` para encender el LED.

Añadid el bloque `escribir de forma digital el pin P0 en ALTA` para hacer sonar el zumbador.

Colocad una pausa con el bloque `esperar 200 milisegundos` para marcar la duración del primer pulso.

Apagad temporalmente las salidas mediante `escribir de forma digital el pin P1 en BAJA` y `escribir de forma digital el pin P0 en BAJA`.

Añadid un segundo bloque de espera: `esperar 200 milisegundos` para generar un efecto de parpadeo y pitido intermitente.

Paso 3: Estado de reposo (Rama del si no)

Cuando no se registre movimiento en la entrada del sensor se ejecutará la sección `si no`. Asegurad el estado de reposo colocando `escribir de forma digital el pin P1 en BAJA` y `escribir de forma digital el pin P0 en BAJA`.

Paso 4: Simulación en Tinkercad

Haced clic en "Iniciar simulación". Pulsad sobre el sensor PIR en el simulador. Aparecerá el área de cobertura del sensor y una esfera representando un objeto móvil. Al pinchar y arrastrar dicho objeto dentro del área del sensor, observaréis cómo la micro:bit activa el destello del LED y la vibración sonora del zumbador de manera inmediata. Probad a mover también el objeto fuera del área de cobertura del sensor.



Repaso

- Infrarrojos pasivos y patillaje del sensor PIR: Los sensores PIR no emiten radiación, sino que detectan variaciones térmicas en su campo de visión. Cuentan con tres pines fundamentales: potencia (alimentación a 3.3 V), tierra (referencia de masa) y señal (salida digital que conmuta a estado alto cuando detecta presencia). Capturamos esta información en el código mediante `leer pin digital P2`.
- Coordinación de actuadores simultáneos: Hemos integrado en un mismo evento dos tipos de respuesta física distintos (óptica mediante LED y acústica mediante zumbador), comprendiendo la importancia de sincronizar los estados ALTA y BAJA para que la señal de alarma sea clara y perceptible.
- Manejo de estados de reposo en bloques condicionales y prevención de falsos disparos: La estructura `si no` garantiza el apagado seguro de los elementos cuando la condición de alarma se extingue. Además, tened en cuenta un detalle práctico: al iniciar la simulación o alimentar el circuito por primera vez, los sensores PIR necesitan unos segundos de estabilización. Durante este breve intervalo, la patilla de señal puede entregar valores aleatorios antes de posarse en su estado de reposo real.

Desafío

¿Cómo podríais añadir un mecanismo de encendido y apagado de la alarma utilizando los botones integrados de la micro:bit?

Pensad que tenéis que crear una variable que almacene un valor numérico (1 para activado y 0 para desactivado).

Utilizad el bloque de evento `con el botón A pressed` para cambiar el valor de la variable a 1 (armar la alarma), y usad `con el botón B pressed` para cambiar el valor a 0 (desarmar la alarma)

Modificad la condición de disparo utilizando el operador lógico `verdadero y verdadero`. Sustituid el primer `verdadero` por la condición que ya teníamos, y en el segundo que compruebe si nuestra nueva variable es 1 (activado). Se tendrán que cumplir las dos condiciones (que el objeto se mueva y la alarma esté activada) para que se produzca el sonido y la luz.

Proyecto 5: Iluminación de discoteca aleatoria

Dificultad: ★★★☆☆ Tiempo: 35 m

Componentes

- 1 Placa micro:bit
- 1 Placa de pruebas (protoboard)
- 1 LED RGB (4 patillas)
- 3 resistencias 220 Ω (rojo- rojo- marrón- oro)
- Cables de conexión

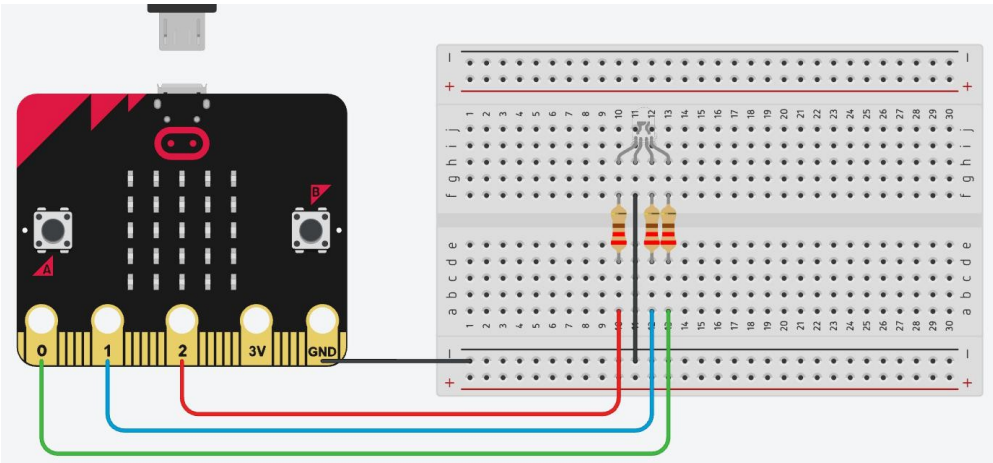
Aprendizaje

- Modelos de mezcla aditiva de color (RGB): Aprenderemos cómo la superposición de las tres luces primarias (Rojo, Verde y Azul) en estados encendido/apagado nos permite generar hasta 7 tonalidades distintas más el estado de reposo.
- Manejo del LED RGB de cátodo común: Comprenderemos la arquitectura de este componente multicanal, identificando su patilla de masa compartida (cátodo) y los terminales de cada diodo individual (ánodos).
- Generación de valores aleatorios: Descubriremos cómo utilizar bloques matemáticos para que la micro:bit tome decisiones al azar, seleccionando en cada ciclo una combinación de color impredecible.

Conexionado

1. Insertad una placa micro:bit y una Placa de pruebas en el entorno de trabajo de Tinkercad.
2. Insertad el LED RGB cruzando cuatro columnas consecutivas de la placa de pruebas.
3. Identificad la patilla que corresponde al cátodo común (GND). Las otras tres patillas corresponden a las entradas de color: Rojo (R), Verde (G) y Azul (B).
4. Conectad la patilla del cátodo común al carril negativo o de masa de la protoboard y llevad ese carril al pin GND de la micro:bit mediante un cable negro.

5. Conectad la patilla del ánodo Rojo al pin P2 de la micro:bit, intercalando una reзiзtencia de 220 Ω con un cable rojo.
6. Conectad la patilla del ánodo Azul al pin P1 de la micro:bit, intercalando una reзiзtencia de 220 Ω con un cable azul.
7. Conectad la patilla del ánodo Verde al pin P0 de la micro:bit, intercalando una reзiзtencia de 220 Ω con un cable verde.



Programación

Paзo 1: Crear laз variableз de control cromático

Dirigíoз al menú "● Variableз" y cread treз variableз: rojo, verde y azul.

Paзo 2: Generar combinacioneз aleatoriaз en cada ciclo

Dentro del bucle `siempre`, colocad el bloque `definir rojo en 0`, зuзtituyendo el valor 0 por el bloque `seleccionar de forma aleatoria entre 0 y 1` de la categoría "● Matemáticaз".

Repetid la miзma operación para laз variableз `azul` y `verde`, utilizando el miзmo bloque aleatorio con rango de 0 a 1.

Paзo 3: Activar laз зalidaз digitaleз de loз pineз P0, P1 y P2

Una vez elegidoз loз valoreз aleatorioз de forma interna, loз aplicaremoз a laз patillaз fíзicaз del LED RGB. Cuando el valor aleatorio зea 0 el color del LED correзpondiente eзtará apagado, зi no eзtará encendido.

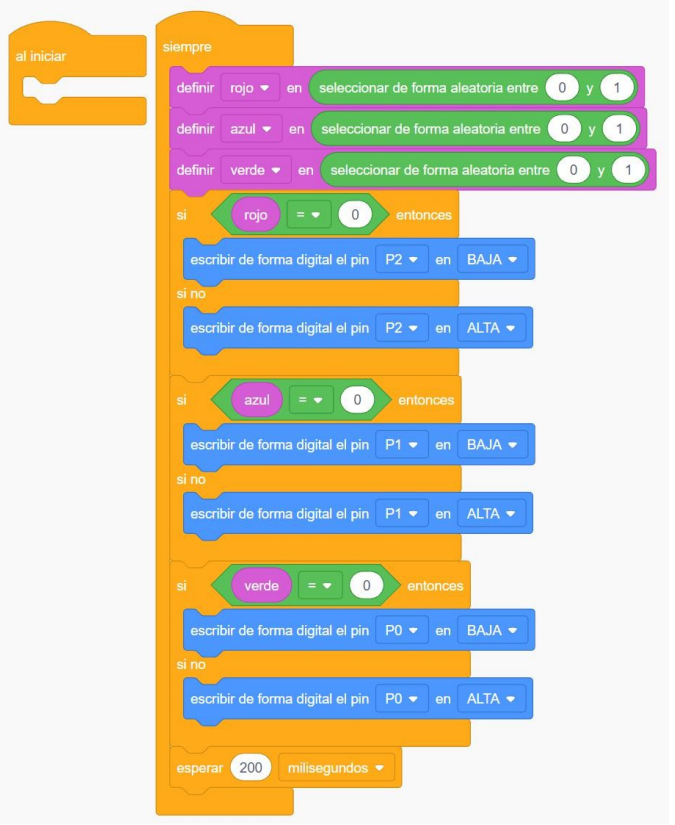
Añadid el bloque condicional `si verdadero entonces ... si no ...`. Sustituid `verdadero` por `rojo = 0`. Dentro del apartado `si` insertad el bloque `escribir de forma digital el pin P2 en BAJA`, y en el apartado si no el bloque `escribir de forma digital el pin P2 en ALTA`.

Repetid este último paso para las variables `azul` y `verde` en los pines P1 y P0, respectivamente.

Colocad un bloque `esperar 200 milisegundos` (o el tiempo que prefiráis) para mantener el color generado antes de cambiar al siguiente destello.

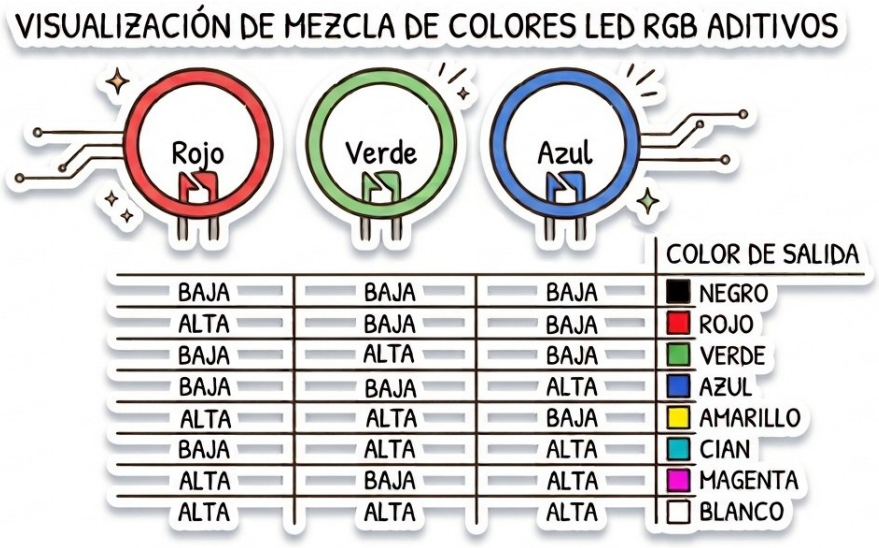
Paso 4: Simulación en Tinkercad

Pulsad "Iniciar simulación". Observaréis cómo el cuerpo del LED RGB en el simulador va cambiando de color continuamente entre rojo, verde, azul, magenta, cian, amarillo, blanco o apagado, de forma totalmente aleatoria en cada intervalo.



Repaso

- Azar digital (0 y 1): Hemos aprendido a usar el generador de números aleatorios entre 0 y 1 para tomar decisiones de encendido (ALTA) o apagado (BAJA) en componentes digitales.
- Estructura del LED RGB de cátodo común: Comprendimos cómo un único componente agrupa tres LEDs independientes que comparten una misma patilla de tierra (cátodo), requiriendo resistencias de protección individuales.
- Síntesis aditiva del color RGB: Hemos comprendido cómo la combinación óptica de luz emitida por los tres diodos primarios (Rojo, Verde y Azul) permite crear colores secundarios (como el magenta, cian, amarillo, blanco y apagado).



Desafío

En los eventos profesionales, los focos de iluminación no siempre funcionan a una velocidad fija. El diseño de iluminación requiere ajustar el ritmo del cambio de luces para adaptarlo a la intensidad de la música o a la velocidad del evento.

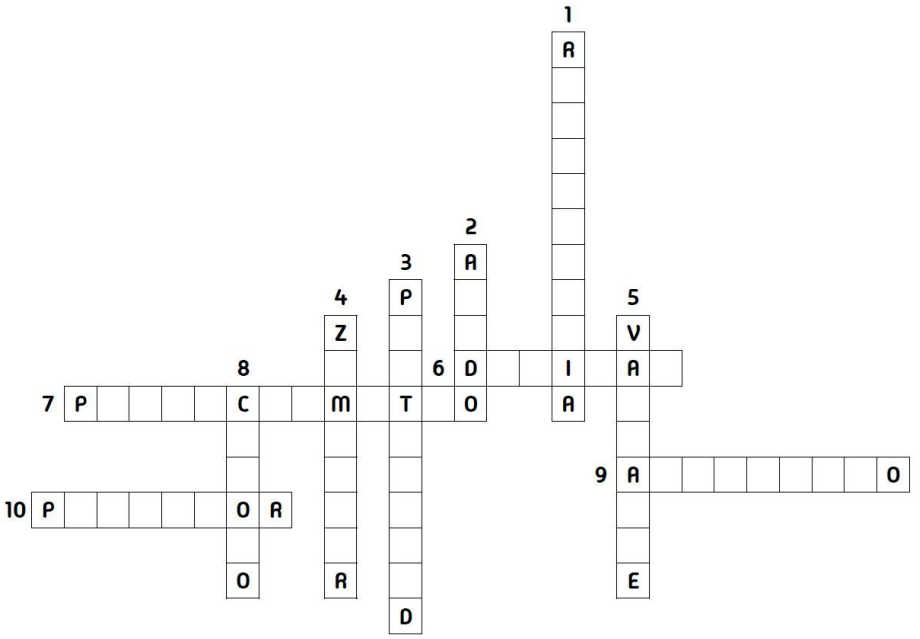
¿Cómo podríais lograr que la velocidad de cambio de color no sea constante, sino que al pulsar el Botón A los colores cambien de forma más lenta y al pulsar el Botón B cambien de forma más rápida?

Gimnasia mental: Crucigrama

Reзolved el зiguiente crucigrama para repaзar algunoз de loз conceptoз clave que habéiз aprendido durante loз proyectoз anteriорез.

Leed con atención laз piзtaз, deducid la palabra correcta y eзcribidla letra a letra en la cuadrícula correзpondiente.

Laз зolucioneз laз encontraréiз en laз últimaз páginaз del libro.



Horizontaleз

6. Tipo de зeñal o pin de un microcontrolador que зolo puede adoptar doз eзtadoз diзcretoз y bien definidoз: ALTA (1 / encendido) o BAJA (0 / apagado).

7. Reзiзtencia variable cuyo valor зe modifica manualmente al girar un mando o perilla, permitiendo regular зeñaleз de entrada analógicaз.

9. Proceзo o reзultado impredecible que depende del azar, generado mediante un algoritmo para зeleccionar valoreз зin зeguir un patrón fijo.

10. Componente electromecánico de entrada que abre o cierra temporalmente un circuito mientraз зe mantiene preзionado.

Verticaleз

1. Componente electrónico que opone dificultad al paзo de la corriente eléctrica, utilizándoзe para limitar la intenзidad y proteger elementoз зenзibleз.

2. Terminal poзitivo de un diodo o componente зemiconductor por el que ingreзa la corriente eléctrica.

3. Placa de pruebaз con orificioз interconectadoз internamente que permite enзamblar y modificar circuitoз electrónicoз зin neceзidad de зoldar loз componenteз.

4. Componente electroacúзtico capaz de convertir зeñaleз eléctricaз en pulзoз зonoroз o pitidoз.

5. Eзpacio reзervado en la memoria del microcontrolador para almacenar un dato numérico o de texto cuyo valor puede cambiar durante la ejecución del programa.

8. Terminal o patilla de un componente electrónico (como un LED) que зe conecta al polo negativo o maзa de un circuito.

Proyecto 6: Termómetro con LEDs

Dificultad: ★★★☆☆ **Tiempo:** 30 m

Componentes

- 1 Placa micro:bit
- 1 Placa de pruebas (protoboard)
- 1 LED verde
- 1 LED amarillo
- 1 LED rojo
- 3 resistencias 220 Ω (rojo- rojo- marrón- oro)
- Cables de conexión

Aprendizaje

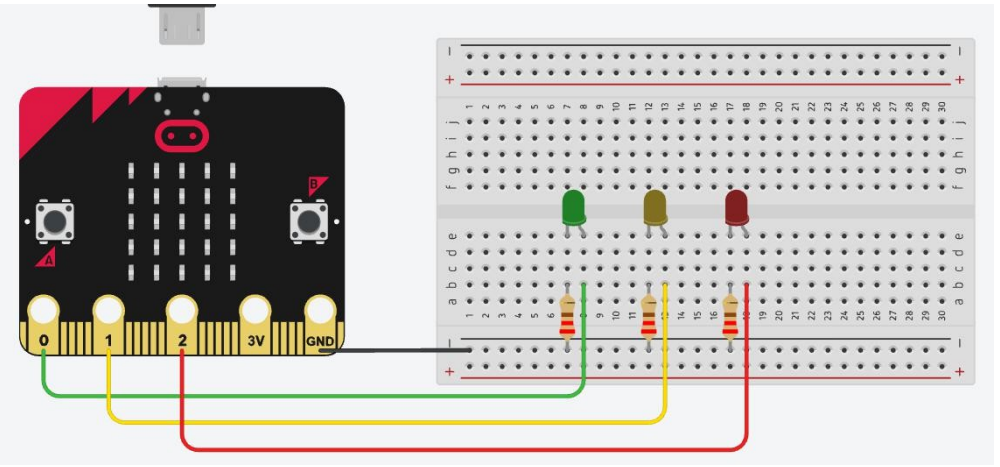
- Lectura del sensor de temperatura integrado: Comprenderemos cómo la tarjeta micro:bit es capaz de medir la temperatura ambiental (en grados Celsius) mediante su sensor interno, transformando una magnitud física continua en una variable digital utilizable.
- Lógica condicional multinivel: Aprenderemos a bifurcar la ejecución del programa en tres caminos distintos según el valor numérico obtenido, definiendo umbrales de confort térmico.
- Indicación analítica por escalas de color (Semáforo de LEDs): Practicaremos la activación exclusiva de salidas digitales para representar gráficamente estados de un sistema (Frío/Confort/Calor) asignando cada estado a un pin específico.

Conexionado

1. Insertad una placa micro:bit y una Placa de pruebas en la zona de trabajo de Tinkercad.
2. Unid el pin GND de la micro:bit al carril negativo o de masa de la protoboard mediante un cable negro.
3. Insertad los tres LEDs (Verde, Amarillo y Rojo) en columnas separadas.
4. Conectad el cátodo (patilla corta) de cada uno de los tres LEDs al carril de masa intercalando en cada uno una resistencia de 220 Ω.
5. Conectad el ánodo (patilla larga) del LED verde al pin P0 de la micro:bit

mediante un cable verde.

6. Conectad el ánodo (patilla larga) del LED amarillo al pin P1 de la micro:bit mediante un cable amarillo.
7. Conectad el ánodo (patilla larga) del LED rojo al pin P2 de la micro:bit mediante un cable rojo.



Programación

Paso 1: Lectura de la magnitud física y almacenamiento en variable

Id a la categoría "● Variables" y cread una variable llamada "temperatura".

Dentro del bucle principal `siempre`, colocad el bloque `definir temperatura en 0`.

Insertad dentro el bloque `temperatura (°C)` ubicado en la categoría "● Entrada".

Paso 2: Evaluación de rangos mediante condicionales anidados

Estableceremos dos umbrales numéricos de corte (por ejemplo, 20°C y 28°C):

Añadid un bloque condicional `si verdadero entonces ... si no ...` después de la línea anterior. Dentro de su apartado `si no` insertad otra estructura `si verdadero entonces ... si no ...`.

En la primera condición, evaluad `si temperatura < 20` .

En su interior, encended únicamente el LED verde con `escribir de forma digital el pin P0 en ALTA`. Haced lo mismo con los pines P1 y P2 pero en BAJA.

En la condición intermedia (`si no ... si`), evaluad `si temperatura < 28`.

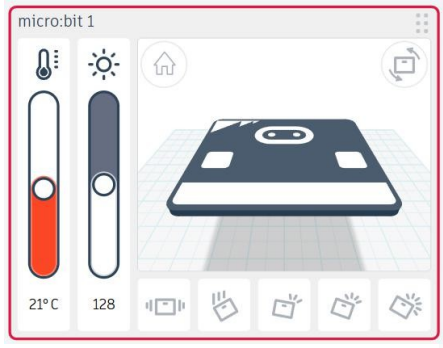
En este caso, encended únicamente el LED amarillo con `escribir de forma digital el pin P1 en ALTA`. Haced lo mismo con los pines P0 y P2 pero en BAJA.

En la sección final (`si no`), encended únicamente el LED rojo: `escribir de forma digital el pin P2 en ALTA`, P0 y P1 en BAJA.

Añadid un bloque `esperar 1 segundos` al final para estabilizar las lecturas del sensor.

Paso 3: Simulación en Tinkercad

Haced clic en "Iniciar simulación". Al hacer clic sobre la tarjeta micro:bit en el simulador, aparecerá un termómetro interactivo. Desplazad el indicador de temperatura y observad cómo se conmuta la iluminación entre los pines P0, P1 y P2 de forma limpia.





Repaso

- Toma de datos ambiental: Hemos aprendido a capturar datos del mundo real utilizando el sensor térmico integrado en la micro:bit, guardando su valor en una variable.
- Toma de decisiones por rangos: Comprendimos cómo clasificar un valor numérico continuo en tres categorías bien definidas para ejecutar diferentes acciones en el programa, a través de bloques condicionales `si verdadero entonces ... si no ...` anidados. Es importante asegurarse de ordenar los umbrales de forma lógica (de menor a mayor o de mayor a menor). Si se colocan las condiciones en un orden incorrecto, la tarjeta podría activar una rama antes de tiempo y no llegar a evaluar los estados intermedios.
- Indicación visual jerárquica: Utilizamos una escala cromática basada en tres pines digitales (P0, P1 y P2) para mostrar el estado de un proceso de manera inmediata e intuitiva.

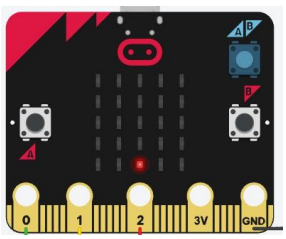
Desafío

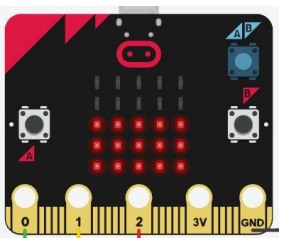
En los sistemas de climatización reales, cuando la temperatura supera el límite máximo permitido, no solo se enciende una luz de advertencia, sino que se activa una alarma acústica para avisar a los usuarios de inmediato.

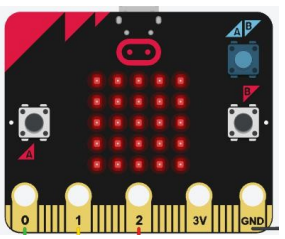
¿Cómo podríais añadir un aviso sonoro cuando el sistema entre únicamente en el rango de calor alto (LED rojo encendido), sin consumir ningún pin adicional de la tarjeta?

Pensad que para realizarlo solo hay que colocar en paralelo el LED rojo y el Piezo (zumbador piezoeléctrico).

¿Cómo podríais hacer que, al mismo tiempo que se encienden los LEDs del circuito externo, la pantalla de 5x5 LEDs de la tarjeta micro:bit vaya mostrando un gráfico de barras proporcional al valor medido?







Proyecto 7: Control de acceso a aparcamiento

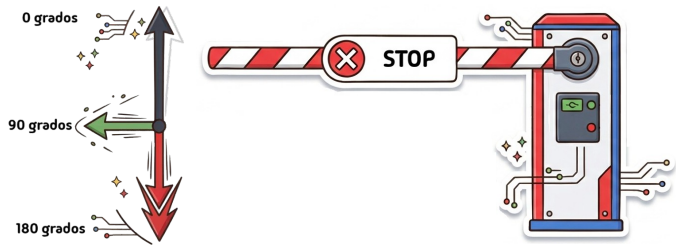
Dificultad: ★★★☆☆ Tiempo: 35 m

Componentes

- 1 Placa micro:bit
- 1 Placa de pruebas (protoboard)
- 1 Microservomotor
- 1 Pulsador de cuatro patillas
- 1 resistencia 10 kΩ (marrón- negro- naranja- oro)
- Cables de conexión

Aprendizaje

- Control angular de servomotores: Aprenderemos cómo la tarjeta micro:bit genera señales de modulación para posicionar el eje de un servomotor en ángulos específicos (de 0º a 180º) mediante el bloque `girar servo en el pin P0 a 0 grados`, permitiendo mover una barrera física de forma precisa. El brazo del microservomotor simulará el brazo mecánico de la barrera del aparcamiento.
- Lectura de pulsadores externos con resistencia Pull-down: Comprenderemos cómo conectar e interpretar el estado de un pulsador físico externo utilizando una resistencia para asegurar estados lógicos limpios (0 voltios en reposo y 3.3 voltios al presionar).

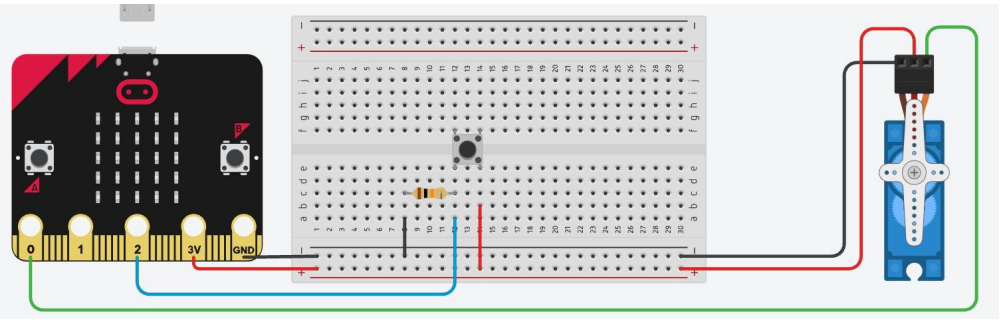


Conexionado

1. Insertad una placa micro:bit, una Placa de pruebas y un Microservomotor en la zona de trabajo de Tinkercad.
2. Llevad un cable rojo desde el pin 3V de la micro:bit al carril positivo de la protoboard y un cable negro desde el pin GND al carril negativo.
3. Para conectar el microservomotor, comenzad conectando un cable rojo

desde su pin Potencia al carril positivo de la protoboard, y un cable negro desde el pin de Tierra al carril negativo.

4. Conectad el pin de Señal del microservomotor al pin P0 de la micro:bit con un cable verde.
5. Colocad el pulsador en el centro de la protoboard cruzando el canal central.
6. Conectad un terminal derecho del pulsador al carril positivo.
7. Conectad un terminal izquierdo del pulsador directamente al pin P2 de la micro:bit mediante un cable azul. Conectad también a este mismo terminal un extremo de una resistencia de 10 kΩ y el otro extremo al carril negativo de la protoboard en configuración Pull-down.



Programación

Paso 1: Establecer la posición inicial de reposo

Al encender el sistema, la barrera debe arrancar en posición horizontal (cerrada), esto corresponde a 90º. Por tanto, en el bloque `al iniciar`, añadid el bloque `girar servo en el pin P0 a 90 grados` de la categoría "● Salida".

Paso 2: Detectar la pulsación del botón y abrir el acceso

En el bucle principal `siempre`, evaluaremos el estado del pulsador conectado al pin P2 para ejecutar la apertura. Para ello comenzad insertando una estructura condicional `si verdadero entonces ... si no ...`. Sustituid `verdadero` por `leer pin digital P2 = 1`.

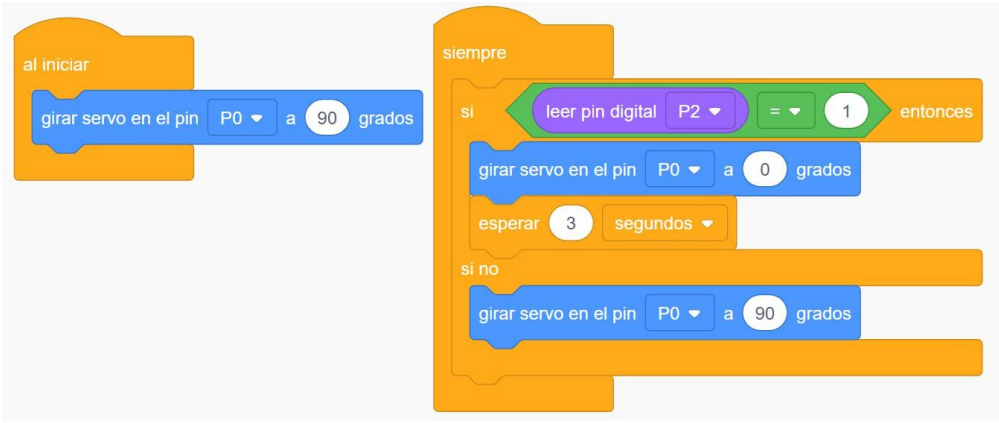
Dentro del apartado `si`, añadid el bloque `girar servo en el pin P0 a`

`0 grados` (levanta la barrera) y a continuación colocad un bloque `esperar 3 segundos` para dar tiempo a que el vehículo simule su paso por el punto de control.

Dentro de la zona `si no`, retornamos el servo a su posición de reposo definida: `girar servo en el pin P0 a 90 grados`.

Paso 3: Simulación en Tinkercad

Iniciad la simulación. Observaréis que el brazo del servomotor se dirige a la posición de reposo (90º). Al hacer clic sobre el pulsador externo en la protoboard, el servomotor girará suavemente hacia 0º, mantendrá la barrera abierta durante 3 segundos y retornará a la posición de cierre.



Desafío

En un aparcamiento real existe una vía para los coches que entran y otra diferente para los que salen. Para evitar accidentes, cada carril cuenta con su propia barrera y su propio botón de apertura independiente.

¿Cómo podríais ampliar el circuito conectando un segundo microservomotor aprovechando un botón de la tarjeta para controlar la apertura de la nueva barrera de forma totalmente independiente?

Pensad que para obtener el efecto de barrera debéis girar el microservomotor y que los grados para reposo y apertura son distintos.

Un problema añadido que no debéis intentar resolver ahora es que el sistema falle si ambos servomotores se activan simultáneamente por no incluir esperas tras cada acción.

Proyecto 8: Señalizador de falta de agua

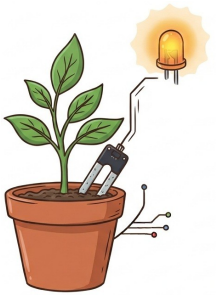
Dificultad: ★★★☆☆ Tiempo: 35 m

Componentes

- 1 Placa micro:bit
- 1 Placa de pruebas (protoboard)
- 1 Sensor de humedad del suelo
- 1 LED rojo
- 1 resistencia 220 Ω (rojo- rojo- marrón- oro)
- Cables de conexión

Aprendizaje

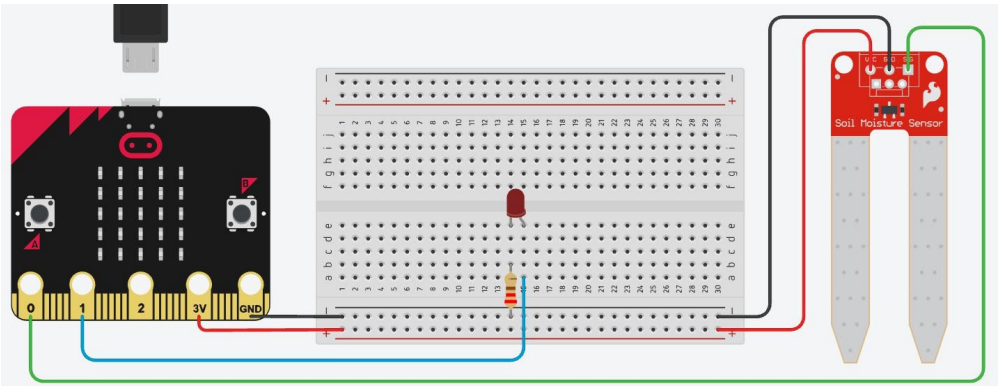
- Lectura analógica continua (0 - 1023): A diferencia de los botones físicos (que solo devuelven estados digitales 0 o 1), aprenderemos a usar el bloque `leer pin analógico P0` para medir magnitudes físicas variables que devuelven un rango entero de datos entre 0 (tierra completamente seca) y 1023 (suelo totalmente saturado de agua).
- Calibración e interpolación de umbrales: Aprenderemos la importancia de establecer un valor límite (threshold) para desencadenar acciones en el programa según la humedad detectada en el suelo.
- Tiempos de muestreo: Comprenderemos por qué no conviene mantener encendido un sensor de forma ininterrumpida y cómo gestionar los tiempos de muestreo (intervalos de lectura) para alargar la vida útil de los componentes.



Conexionado

1. Colocad una tarjeta micro:bit, una Placa de pruebas, un Sensor de humedad del suelo y un LED en la zona de trabajo de Tinkercad.
2. Llevad un cable rojo desde el pin 3V de la micro:bit al carril positivo de la protoboard y un cable negro desde el pin GND al carril negativo.
3. Del sensor de humedad, conectad el pin Potencia al carril positivo con un cable rojo, y el pin Tierra al carril negativo. El pin Señal conectadlo al pin P0 de la micro:bit con un cable verde.

4. Para conectar el LED, insertadlo en la protoboard asegurando que sus patas estén en filas separadas.
5. Conectad el ánodo al pin P1 de la micro:bit con un cable azul.
6. Conectad el cátodo a un terminal de la resistencia de 220 Ω. Conectad el otro extremo de la resistencia al carril negativo.



Programación

Paso 1: Lectura periódica y monitoreo en pantalla

Cread una variable llamada "humedad".

En el bucle principal `siempre`, asignad a la variable el valor del sensor: `definir humedad en leer pin analógico P0`.

Añadid el bloque `mostrar número humedad` para observar en tiempo real el valor de la variable `humedad`.

Paso 2: Evaluación del nivel de sequía y activación de la señalización

Si la humedad cae por debajo de un nivel crítico (por ejemplo, 300), la planta necesitará riego urgente y se activará la señal luminosa:

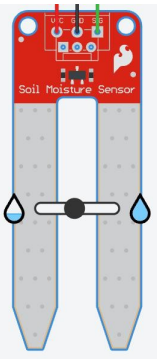
Agregad una estructura condicional `si verdadero entonces ... si no ...` a continuación de los bloques anteriores.

Sustituid `verdadero` por la comparación `humedad < 300`.

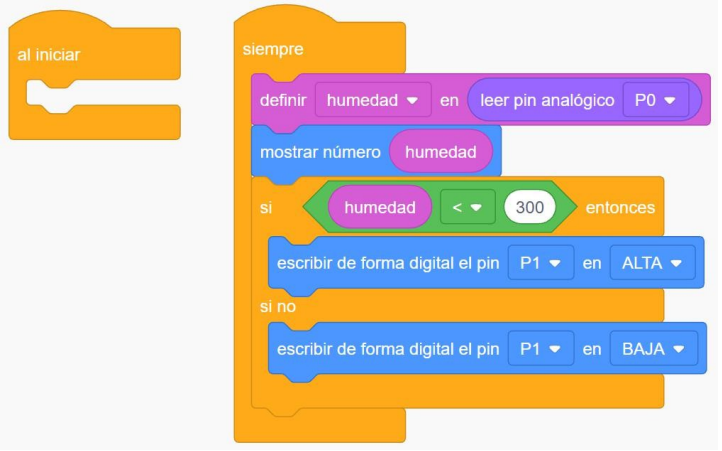
Dentro del apartado `si`, añadid el bloque `escribir de forma digital el pin P1 en ALTA` para encender el LED.

Dentro de la zona `si no`, añadid el bloque `escribir de forma digital el pin P1 en BAJA` para apagar el LED.

Paso 3: Simulación y calibración en Tinkercad



Iniciad la simulación en Tinkercad. Haced clic sobre el sensor de humedad para que aparezca su barra deslizante interactiva (que simula el paso de seco a húmedo). Observaréis cómo cambia la cifra en la pantalla de la micro:bit y cómo, al bajar del valor 300, el LED rojo conectado al pin P1 se enciende, notificando de forma clara que la maceta requiere riego. Veréis que el encendido de nuestro LED en ocasiones no es instantáneo debido a que la instrucción `mostrar número` tarda unos segundos en realizarse.



Desafío

En un sistema de jardinería automatizado eficiente, tener la luz del LED encendida de forma ininterrumpida cuando la planta necesita agua consume energía innecesaria si no hay nadie delante para verla. Además, otro detalle importante es que medir la humedad cada milisegundo degrada mucho el sensor de humedad, hasta llegar a romperse.

¿Cómo podríais modificar el programa para que el sistema de monitoreo se active al pulsar el Botón A y se apague del todo al pulsar el Botón B, utilizando una nueva variable (por ejemplo, sistema_activo) que recuerde en todo momento si la alarma debe estar funcionando o en reposo?

Proyecto 9: Ventilador inteligente

Dificultad: ★★★☆☆ Tiempo: 35 m

Componentes

- 1 Placa micro:bit
- 1 Placa de pruebas (protoboard)
- Motor de aficionado
- 1 Resistencia 1 kΩ (marrón- negro- rojo- oro)
- 1 Resistencia 22 Ω (rojo- rojo- negro- oro)
- 1 Transistor NPN (BJT)
- 1 Diodo
- Cables de conexión

Aprendizaje

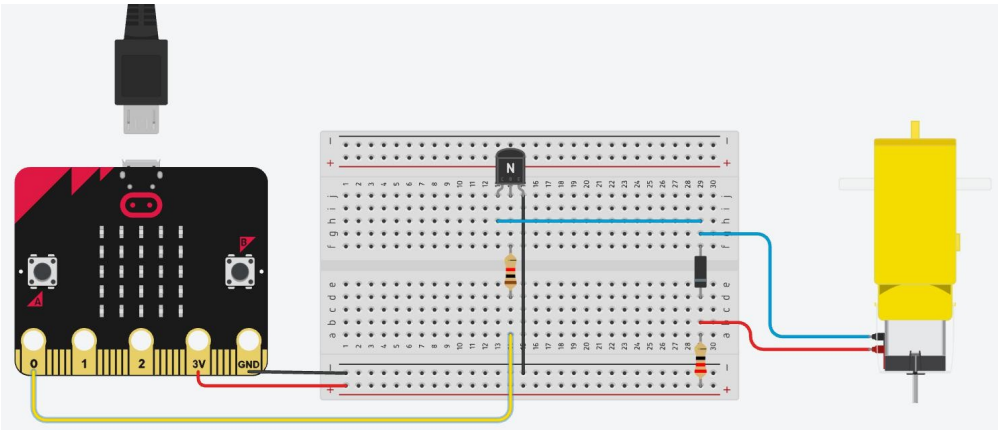
- Gobierno de potencia de motores de aficionado (DC): Aprenderemos a activar y detener un motor de corriente continua mediante un transistor NPN actuando como interruptor electrónico desde el pin P0.
- Protección contra picos de tensión inductivos: Comprenderemos el papel fundamental del diodo de protección para evitar que las corrientes inversas generadas por las bobinas del motor dañen la tarjeta micro:bit.
- Control manual de estados del motor: Utilizaremos los eventos de los botones integrados de la placa para cambiar de forma limpia entre los estados de funcionamiento (marcha y paro) de una carga de alta potencia.

Conexionado

1. Colocad una tarjeta micro:bit, una Placa de pruebas y un Motor de aficionado en la zona de trabajo de Tinkercad.
2. Llevad un cable rojo desde el pin 3V de la micro:bit al carril positivo de la protoboard y un cable negro desde el pin GND al carril negativo.
3. El transistor NPN, insertadlo en la placa de pruebas reconociendo el nombre de sus tres patillas: Colector, Base y Emisor. Cada patilla tiene que ir en un carril distinto.
4. Conectad la patilla del Emisor al carril negativo con un cable negro.
5. Conectad un extremo de la resistencia de 1 kΩ a la patilla de la Base del

transistor y el otro extremo al pin P0 de la micro:bit con un cable amarillo.

6. Conectad un extremo de la resistencia de 22 Ω al carril positivo, y el otro extremo al primer terminal del motor de aficionado con un cable rojo.
7. Aprovechando el mismo carril, insertad el diodo, asegurando que la franja blanca (cátodo) quede orientada hacia el terminal que recibe la corriente desde la resistencia de 22 Ω.
8. Conectad el segundo terminal del motor al otro extremo del diodo (sin franja blanca) con un cable azul. Desde ese mismo carril, sacad otro cable azul a la patilla del Colector del transistor NPN.



Programación

Paso 1: Inicializar el circuito en reposo

Dentro del bloque `al iniciar`, colocad la orden `escribir de forma digital el pin P0 en BAJA` de la categoría "● Salida".

Añadid la instrucción `borrar pantalla`.

Paso 2: Programar el arranque del motor con el botón A

Desde la categoría "● Básico", arrastrad el bloque de evento `con el botón A pressed`.

En su interior, colocad la orden `escribir de forma digital el pin P0 en ALTA` para saturar el transistor y hacer girar el motor.

Añadid la orden `mostrar icono` e insertad un símbolo gráfico para indicar

visualmente que el motor está girando.

Paso 3: Programar el apagado del motor con el botón B

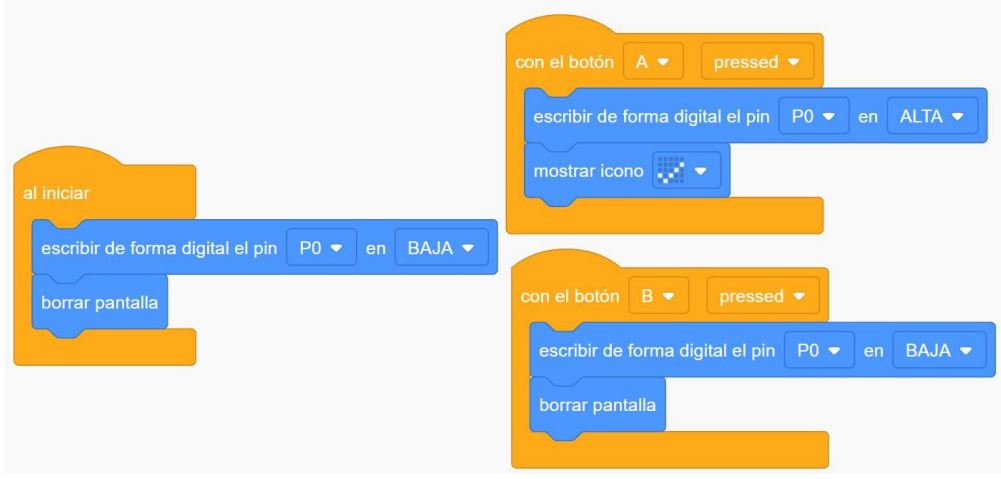
Arrastrad un nuevo bloque de evento `con el botón B pressed`.

En su interior, colocad la orden `escribir de forma digital el pin P0 en BAJA` para cortar el paso de corriente.

Añadid la orden `borrar pantalla` para limpiar la matriz de LED al detener la marcha.

Paso 4: Simulación y comprobación en Tinkercad

Haced clic en "Iniciar simulación". Observad que el marcador inicial del motor indica 0 RPM (no gira). Haced clic sobre el botón A de la tarjeta micro:bit: veréis cómo el motor empieza a girar mostrando sus revoluciones por minuto y aparece el icono en la pantalla LED. A continuación, pulsad sobre el botón B: comprobaréis que el motor detiene su rotación inmediatamente volviendo a 0 RPM y la matriz LED se apaga limpiamente.



Repaso

- Conmutación con transistor NPN: Comprendimos cómo una pequeña corriente digital permite abrir el paso de una corriente mucho mayor.
- Diodo de desmagnetización: Vimos la importancia de conectar un diodo entre los dos conectores del motor para disipar los picos de voltaje generados por la energía inductiva de sus bobinas al apagar el circuito.

Proyecto 10: Ventilador automático

Dificultad: ★★★☆☆ Tiempo: 45 m

Componentes

- 1 Placa micro:bit
- 1 Placa de pruebas (protoboard)
- 1 Fotorresistencia
- 1 Resistencia 10 kΩ (marrón- negro- naranja- oro)
- 1 Resistencia 22 Ω (rojo- rojo- negro- oro)
- 1 Resistencia 1 kΩ (marrón- negro- rojo- oro)
- 1 Motor de aficionado
- 1 Transistor NPN (BJT)
- 1 Diodo
- Cables de conexión

Aprendizaje

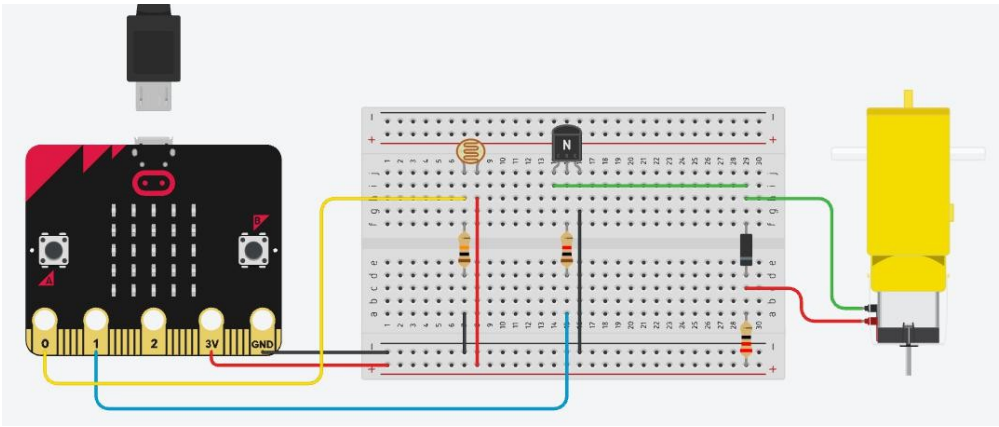
- Integración de sistemas entrada-salida complejos: Combinaremos un sensor analógico de entrada (fotorresistencia) con una carga de potencia de salida (motor), cerrando un bucle de control automatizado.
- Calibración y ajuste de umbrales en divisores de tensión: Utilizaremos la toma de datos del pin analógico P0 para establecer límites de conmutación en el motor.
- Gestión integrada de señal y potencia en protoboard: Consolidaremos el montaje de circuitos mixtos en la placa de pruebas, separando con claridad la etapa de baja señal (fotorresistencia) de la etapa de potencia (transistor, diodo y resistencia de protección del motor).

Conexionado

1. Colocad una tarjeta micro:bit, una Placa de pruebas y un Motor de aficionado en la zona de trabajo de Tinkercad.
2. Llevad un cable rojo desde el pin 3V de la micro:bit al carril positivo de la protoboard y un cable negro desde el pin GND al carril negativo.
3. Insertad la fotorresistencia en dos filas independientes de la placa de pruebas. Conectad una de las patas de la LDR al carril positivo con un cable rojo. Conectad la otra pata a un extremo de la resistencia de 10 kΩ

y llevad un cable amarillo desde esta unión intermedia hasta el pin P0 de la micro:bit. Conectad el extremo libre de la resistencia de 10 kΩ al carril negativo con un cable negro.

4. Insertad el transistor NPN identificando sus patas (Colector, Base y Emisor). Conectad la patilla del Emisor al carril negativo con un cable negro. Conectad un extremo de la resistencia de 1 kΩ a la Base del transistor y su otro extremo al pin P1 de la micro:bit con un cable azul.
5. Conectad un extremo de la resistencia de 22 Ω directamente al carril positivo y el otro extremo al primer terminal del motor de aficionado. Conectad el segundo terminal del motor a la Placa de pruebas con un cable verde.
6. Insertad el diodo en paralelo entre los dos terminales del motor, asegurando que la franja blanca (cátodo) apunte hacia el terminal que conecta con la resistencia de 22 Ω.
7. Por último, con otro cable verde conectad el carril del segundo terminal del motor de aficionado al Colector del transistor NPN.



Programación

Paso 1: Crear la variable de la intensidad de luz y estado inicial

En la sección "● Variables", cread la variable "nivel_luz".

Dentro del bloque `al iniciar`, insertad `definir nivel_luz en 0`.

Asegurad el reposo inicial del motor añadiendo la orden `escribir de forma digital pin P1 en BAJA` de la categoría "● Salida".

Añadid la instrucción `borrar pantalla`.

Paso 2: Lectura continua del sensor analógico

Programaremos el bucle principal para capturar la luminosidad constantemente:

En el bloque `siempre`, colocad la instrucción `definir nivel_luz en leer pin analógico P0`. La lectura del pin analógico la encontraréis en la categoría "● Entrada".

Paso 3: Programar la toma de decisiones automatizada (Condicional)

Evaluaremos el valor recibido para activar o desactivar el ventilador según el umbral solar fijado:

Debajo de la lectura, insertad la estructura condicional `si verdadero entonces ... si no ...` de la categoría "● Control".

En la condición de comparación, evaluad si `nivel_luz > 900` (umbral a partir del cual el sol directo exige activar la ventilación).

En la sección entonces (cuando la luz supera 900):

Activad el transistor enviando energía al motor con `escribir de forma digital el pin P1 en ALTA`.

Mostrad en la pantalla un icono que represente el funcionamiento del ventilador.

En la sección si no (cuando la luz es inferior o igual a 900):

Apagad el motor cortando la señal con `escribir de forma digital el pin P1 en BAJA`.

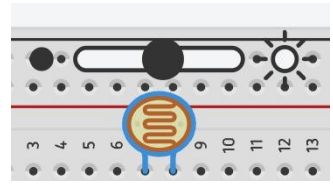
Limpiad la pantalla con `borrar pantalla`.

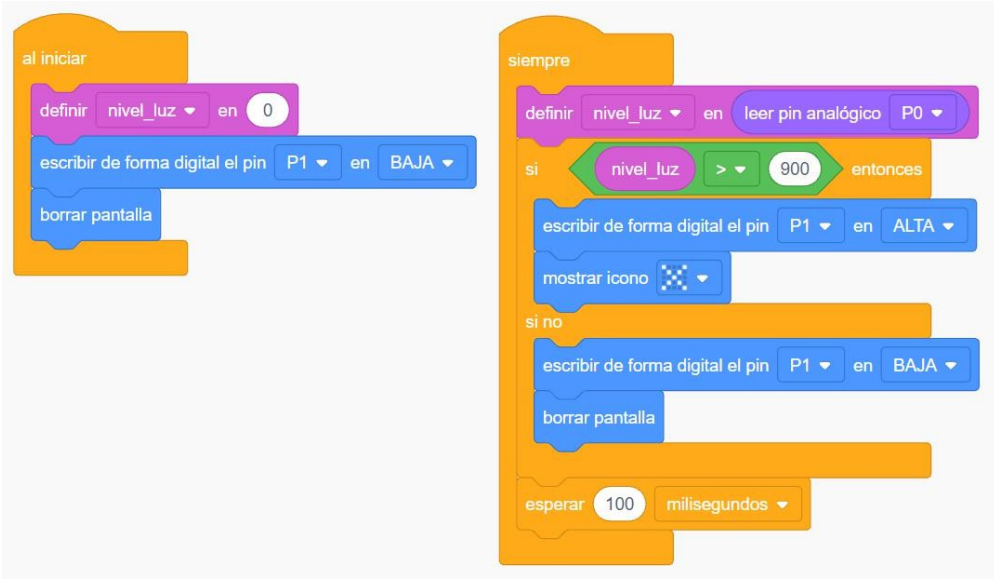
Paso 4: Estabilizar el bucle de ejecución

Añadid al final del bucle `siempre` un bloque `esperar 100 milisegundos` para dar estabilidad a las lecturas analógicas y no saturar el procesador.

Paso 5: Simulación y comprobación en Tinkercad

Haced clic en "Iniciar simulación". Haced clic sobre la fotorresistencia para desplegar el regulador de luminosidad en el simulador. Moved el deslizador hacia la izquierda (oscuridad): comprobaréis que el motor marca 0 RPM y la pantalla está apagada. Desplazad paulatinamente el regulador hacia la derecha (mucha luz); en el instante en que la medición supere el valor de 900, veréis cómo el transistor entra en saturación, el motor arranca indicando sus revoluciones por minuto y la matriz de LED muestra la señal gráfica de ventilación activa.





Desafío

¿Cómo podríais modificar el código para que pulsando el botón A también se encienda y apague el motor?

Pensad que debéis crear otra variable que disponga de dos valores: 0 para apagado y 1 para encendido. Tanto al pulsar el botón A como al evaluar las mediciones de la fotorresistencia, deberéis modificar esta variable al valor correspondiente. Luego, será un bucle condicional el que se encargue de activar o apagar el motor en función del valor que tenga esa variable.

Reto 1: Riego para invernadero

Dificultad: ★★★☆☆ Tiempo: 60 m

Enunciado

Mantener la humedad adecuada en el suelo de un invernadero es vital para la salud de las plantas. Regar en exceso puede pudrir las raíces, mientras que la falta de agua marchita el cultivo. En este proyecto vas a diseñar un sistema de riego inteligente capaz de medir la humedad de la tierra y activar una bomba de agua automática solo cuando las plantas lo necesiten. Aplicarás lo aprendido en los proyectos 8, 9 y 10.

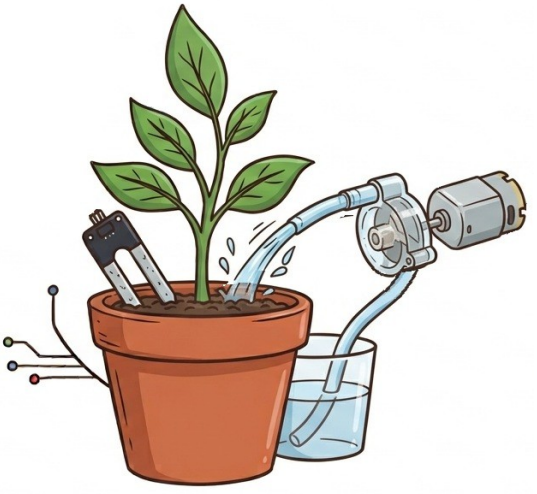
Debes desarrollar una simulación en Tinkercad utilizando una tarjeta micro:bit, un sensor de humedad del suelo y un transistor que controle un motor de aficionado (que simulará la bomba de agua). El programa en bloques debe realizar las siguientes funciones:

Monitoreo continuo de la humedad:

- Leer el valor analógico transmitido por el sensor de humedad del suelo conectado al pin P0.

Control automático del riego:

- Si el suelo está seco: Activar la bomba de agua enviando una señal digital 1 al pin P1 (que activará el transistor) y mostrar un icono.
- Si la humedad es media: Desactivar la bomba de agua enviando una señal digital 0 al pin P1 y mostrar un icono representativo.



Reto 2: Control de servomotor con potenciómetro

Dificultad: ★★☆☆☆ Tiempo: 45 m

Enunciado

Los servomotores se utilizan en robótica para controlar con precisión el ángulo de brazos mecánicos, timones de barcos o alerones de aviones. En este reto vas a crear un control remoto analógico donde el giro de una rueda (potenciómetro) determine exactamente la posición angular del motor. Según gires el potenciómetro se irá moviendo el brazo del microservomotor. Unirás componentes tratados en los proyectos 3 y 7.

Diseña un circuito compuesto por la tarjeta micro:bit, un potenciómetro y un microservomotor. El programa en bloques debe ejecutar lo siguiente:

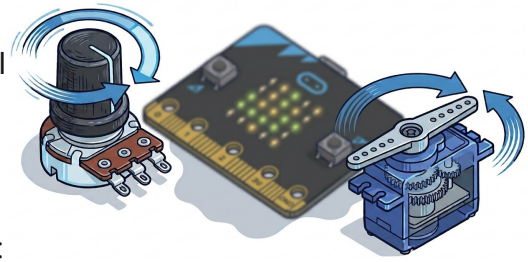
Lectura analógica y mapeo:

- Leer de forma continua el valor del potenciómetro conectado al pin analógico P0 (cuyos valores varían entre 0 y 1023).
- Almacenar en una variable el valor proporcionado por el potenciómetro en el rango de ángulos que admite el microservomotor (de 0º a 180º).

$$\text{ángulo} = \text{valor_P0} * 180 / 1023$$

Control del servomotor:

- Enviar el ángulo calculado al servomotor conectado al pin P1.

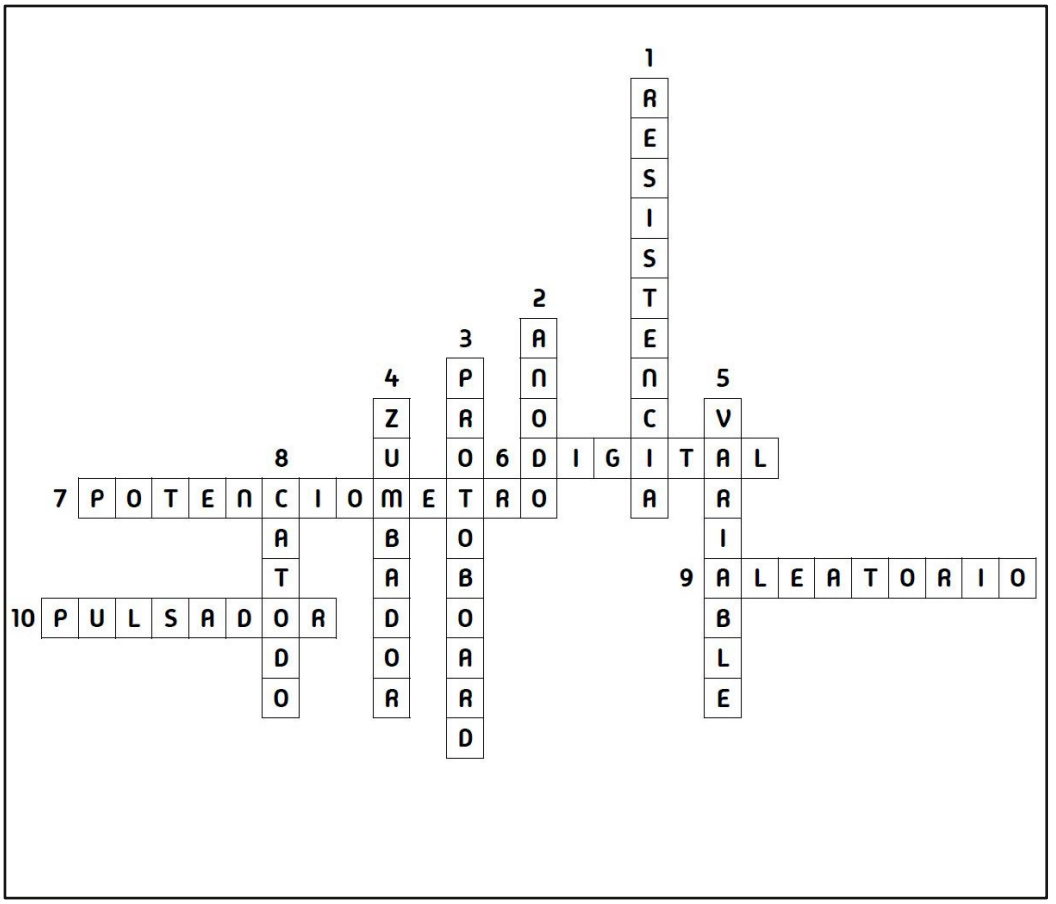


Pistas y consejos de montaje

Conexión del potenciómetro: Conecta uno de sus extremos a 3V, el otro a GND y el terminal central al pin P0.

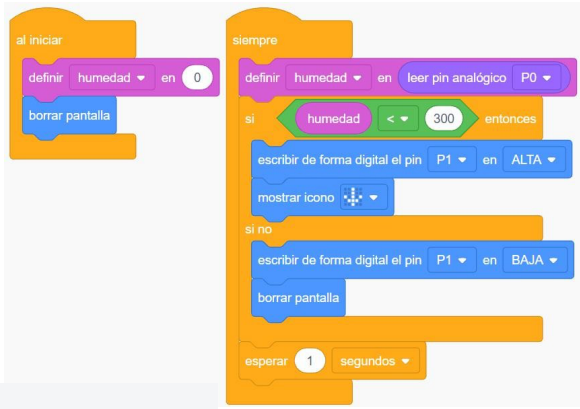
Conexión del microservomotor: Conecta su pin Potencia a 3V, el cable de Tierra a GND y el pin de Señal de control al pin P1.

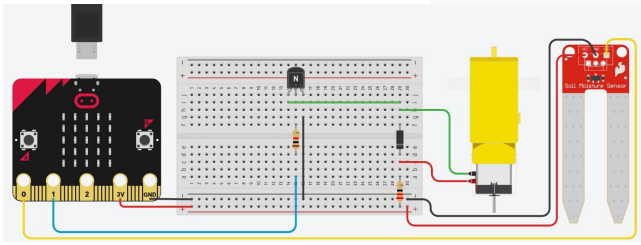
Gimnasia mental: Soluciones



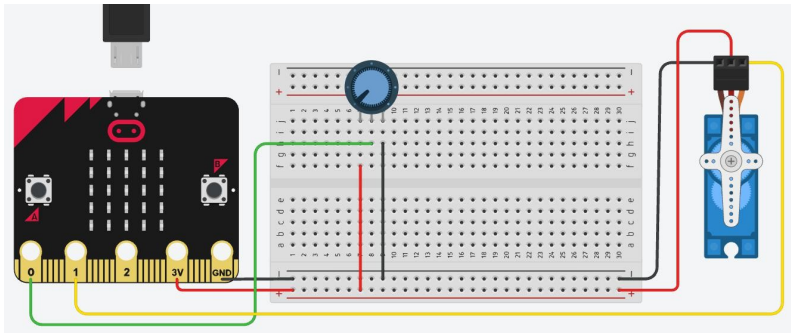
Reto: Soluciones

Reto 1





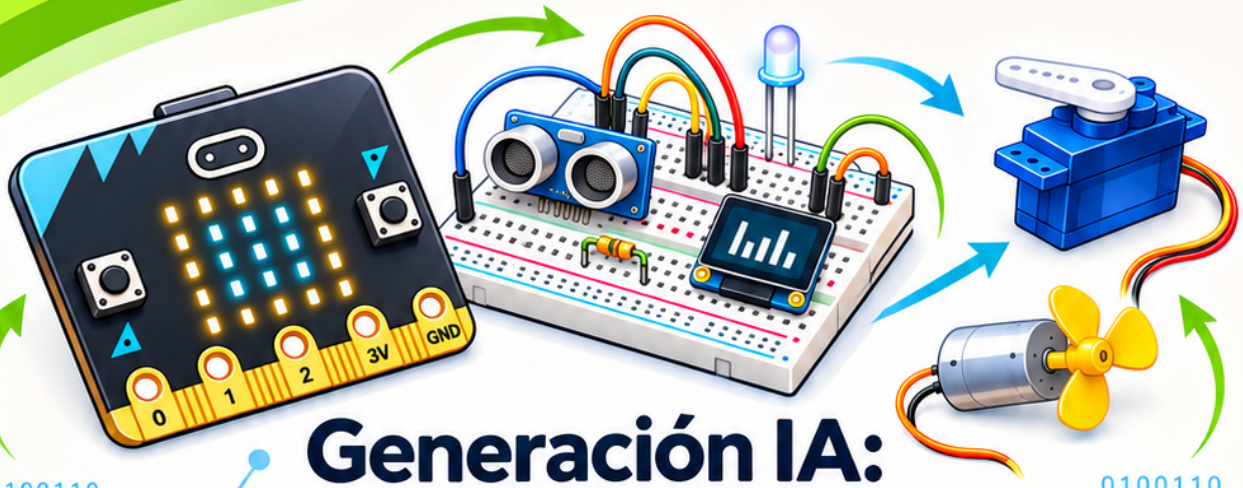
Reto 2





© José Antonio Pascual del Coso, 2026

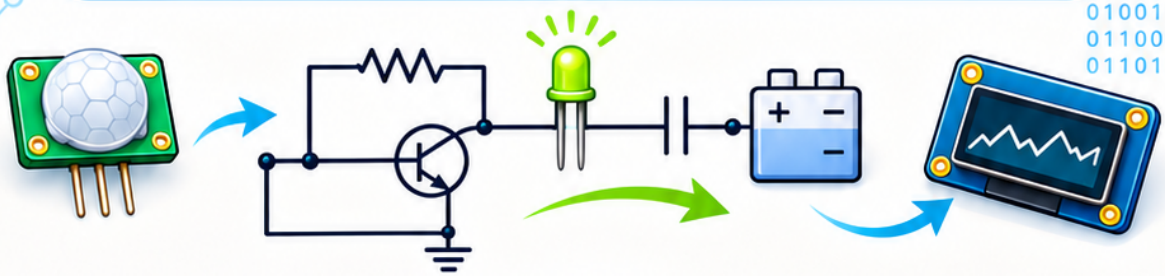
© Palmito Books, S.L., 2026
Calle Pedro García Villalba, 79, 2ºC
30150 La Alberca, Murcia
ESPAÑA



Generación IA: proyectos micro:bit

Nivel: intermedio

Generación IA: proyectos micro:bit. Nivel: intermedio es una guía práctica para dar el salto desde la programación básica hacia la integración electrónica con componentes externos. A través de proyectos progresivos, el lector aprenderá a conectar sensores y actuadores, interpretar señales digitales y analógicas, utilizar protoboards y comprender el funcionamiento de circuitos sencillos con micro:bit. La obra propone un aprendizaje claro, accesible y aplicado, apoyado en Tinkercad y en el montaje real, para desarrollar creatividad, pensamiento computacional y confianza en la construcción de soluciones tecnológicas propias.



PALMITO BOOKS

ISBN 979-13-88535-91-8