

Obligatorio 1 - Manejo de Pines Digitales: Botones, LEDs integrados y LEDs externos

IIE - Facultad de Ingeniería - Universidad de la República

Tallerine Biónico 2024

El objetivo de este obligatorio es aprender a utilizar la herramienta Arduino IDE para programar la placa Micro:bit. Además se buscará comprender el manejo de los pines de entrada/salida (GPIO¹) mediante el uso de los botones, los LEDs integrados y los LEDs externos.

Para realizar este obligatorio es necesario contar con el software de Arduino IDE previamente instalado. Referirse a la “*Guía para configurar Arduino IDE para programar la placa micro:bit*” disponible en EVA.

1. Pines

El microprocesador nRF51 presente en la Micro:bit posee 48 pines (conexiones), de los cuales 25 son accesibles desde la placa. Sin contar las conexiones disponibles para la alimentación, 3V y GND, la placa cuenta con 19 pines, GPIO, numerados del 0 al 16 y del 19 al 20. Los pines 17 y 18 están conectados directamente a 3V.

Los pines se utilizan para comunicar la placa con el mundo exterior. Internamente, algunos pines están conectados a los 2 botones y a los 25 LEDs integrados a la placa (figura 1).

Hay otras formas de comunicar la placa: USB, Bluetooth, acelerómetro, sensor de temperatura, pero no son objeto de este obligatorio.

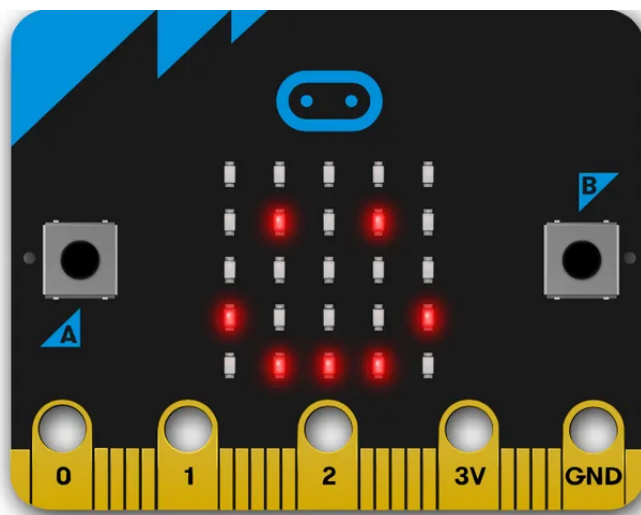


Figura 1: Botones A y B y 25 LEDs integrados a la placa

La figura 2 muestra la correspondencia entre pines y sus distintas funcionalidades.

¹ General Purpose Input Output

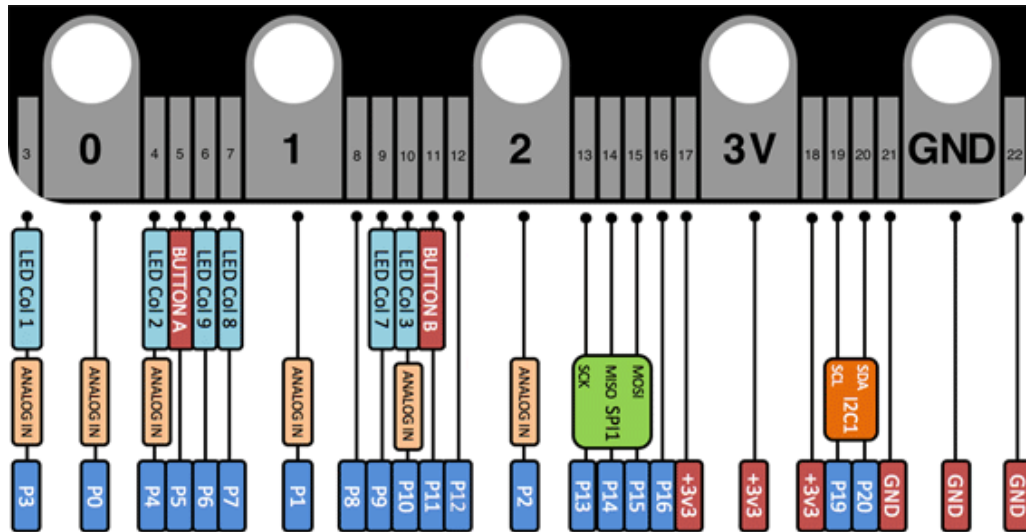


Figura 2: Pines de la Micro:bit v1

Para utilizar los pines de la Micro:bit conectar la placa Kitronik como se observa en la figura 3; los botones y LEDs del lado de los pines P3, P0, ...

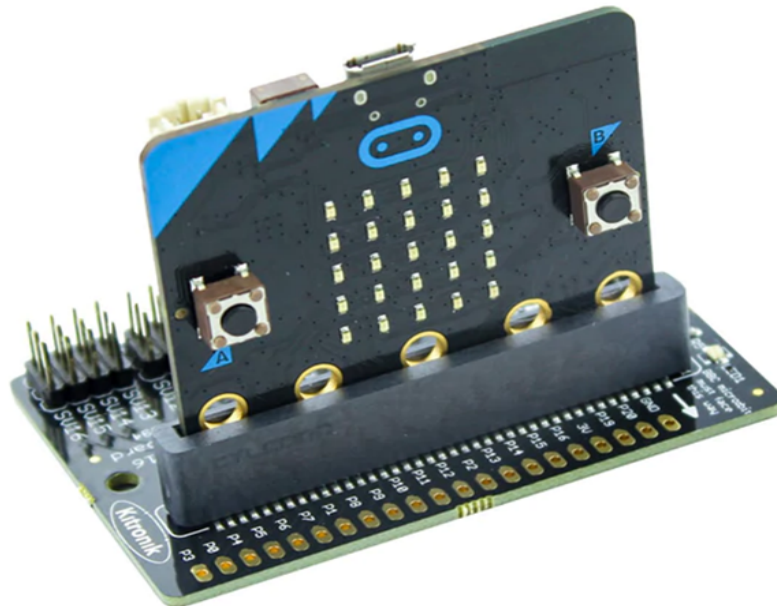


Figura 3: Forma de conectar la placa Kitronik a la Micro:bit v1

2. Botones

Los botones son una forma de proveer información de entrada a la placa. Se dispone de 2 botones integrados, A y B, los cuales se encuentran conectados a los pines 5 y 11 respectivamente.

Si un botón se encuentra presionado, el pin correspondiente tendrá el valor '0' y en caso contrario el valor '1'.

Para utilizar un botón debe configurarse el pin como entrada.

ATENCIÓN: si se usa el botón A no se debe conectar nada al pin P5 de la Micro:bit. Lo mismo aplica para el pin P11 si se utiliza el botón B.

3. LEDs integrados

La placa dispone de 25 LEDs integrados. Cada uno se encuentra conectado a 2 pines según se muestra en el esquema de la figura 4.

Para utilizar un LED deben configurarse ambos pines como salidas.

Para encender el LED se debe poner: $ROW = '1'$ y $COL = '0'$. Cualquiera de las otras 3 combinaciones restantes mantendrá el LED apagado.

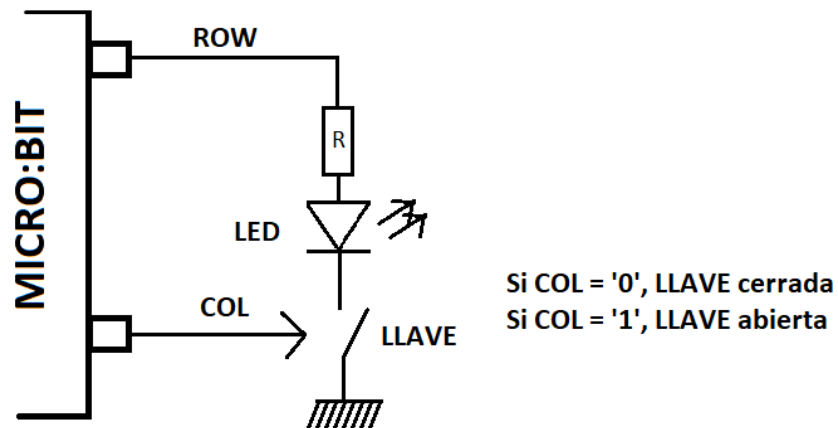


Figura 4: Esquema conexión de un LED.

Para manejar los 25 LEDs de la placa se utilizan 3 señales ROW y 9 señales COL. En el Cuadro 1 se indica a qué pin está conectada cada una de estas señales.

ROW/COL	Pin	ROW/COL	Pin
ROW1	26	COL4	23
ROW2	27	COL5	24
ROW3	28	COL6	25
COL1	3	COL7	9
COL2	4	COL 8	7
COL3	10	COL 9	6

Cuadro 1: Mapeo de las señales ROW_i y COL_j en los pines Micro:bit-LEDs-Arduino

En la figura 5 se muestra una vista de los 25 LEDs y las señales ROW y COL que le corresponden a cada uno, donde Ri/Cj indica ROW_i/COL_j.

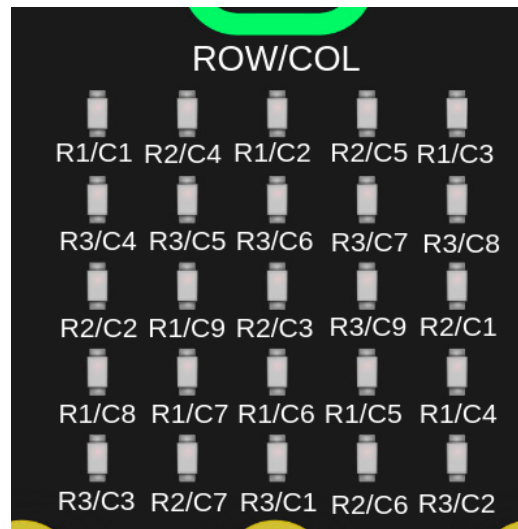


Figura 5: LEDs en Micro:bit

Por ejemplo, para encender el LED de más arriba a la derecha (figura 5) se debe poner ROW1 = '1' y COL3 = '0'. O sea que el pin 26 (ROW1) y el pin 10 (COL3) deben configurarse como salida y escribir un '1' en el pin 26 y un '0' en el pin 10.

ATENCIÓN: si se usan los LEDs integrados no debe conectarse nada a los pines P3, P4, P7, P6, P9 y P10 de la Micro:bit.

4. LEDs externos

Un LED, es un dispositivo emisor de luz que se usa como indicador en muchos dispositivos y en iluminación. Los LEDs tienen polaridad, permitiendo solamente el pasaje de corriente en un solo sentido. Por tanto, se debe alimentar el dispositivo de manera correcta para que este pueda emitir luz.

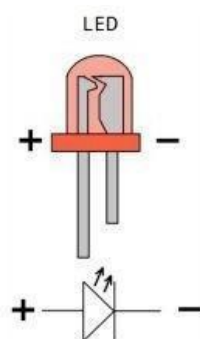


Figura 6: LED

La pata positiva (Ánodo, la más larga) siempre irá conectada al voltaje "más positivo" y la pata negativa (Cátodo, la más corta) irá conectada al voltaje "menos positivo". Si se conecta al revés, solo sucederá que no se podrá encender el LED. Nada se quemará.

Para utilizar un LED siempre se debe conectar una resistencia en serie como se observa en el esquema de la figura 7. La resistencia tiene como objetivo limitar el valor de la corriente y evitar tanto la ruptura del LED como la del microcontrolador que lo alimenta.

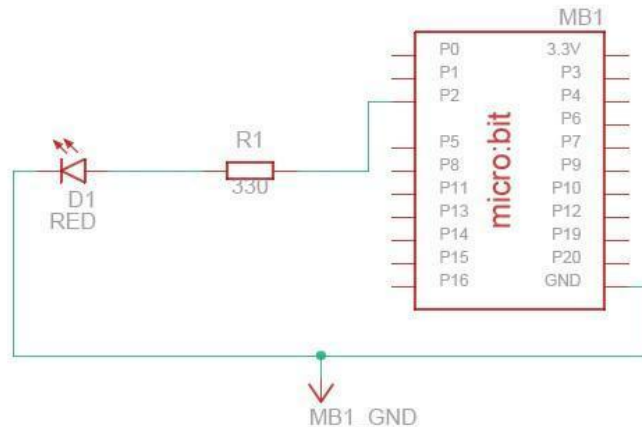


Figura 7: Circuito de conexión del LED

5. Ejercicios

1. Para este ejercicio se debe conectar un LED al pin 0 de la placa Micro:bit. El LED se debe conectar con una resistencia en serie de 330 Ohm (ver figura 7).
 - a. Cargar en la placa el código `parpadeo_v2.ino` y ver que efectivamente el LED se enciende y se apaga.
 - b. Estudiar el código.
 - c. Modificarlo para que el parpadeo sea cada 0.4 segundos.
 - d. Modificar el código para que además parpadee el LED de más abajo a la izquierda de la placa (figura 5). Puede ser útil tomar como referencia el código `parpadeo.ino` disponible en el EVA).

ATENCIÓN: UN ERROR AL CONECTAR LOS LEDS PUEDE QUEMAR LA PLACA

2. Implementar un programa que prenda y apague tres LEDs de forma secuencial. Es decir, se enciende el LED 1 durante medio segundo y luego se apaga. En el momento que se apaga el LED 1, se enciende el LED 2 también durante medio segundo y al apagarse este, se enciende el LED 3 durante medio segundo. Al apagarse el LED 3 comienza nuevamente el ciclo.

Para eso se deberá:

- a. Agregar 3 LEDs y 3 resistencias de 330Ω .
- b. Conectar los tres LEDs externos con su resistencia en serie a 3 pines diferentes de la placa Micro:bit.
- c. Escribir el código acorde a los pines donde hayan conectado los LEDs para poder prender y apagar los LEDs de forma secuencial.

ATENCIÓN: UN ERROR AL CONECTAR LOS LEDS PUEDE QUEMAR LA PLACA.

3. A partir del circuito del ejercicio 1 implementar un programa que haga parpadear el LED:
 - cada 1 segundo si el botón B NO se encuentra oprimido
 - cada 0,4 segundos si el botón B se encuentra oprimido

NOTA:

Para el ejercicio 3 será necesario utilizar la sentencia **if-else**. Se puede acceder a información más detallada de su uso en el siguiente enlace:

<https://www.arduino.cc/reference/en/language/structure/control-structure/if/>