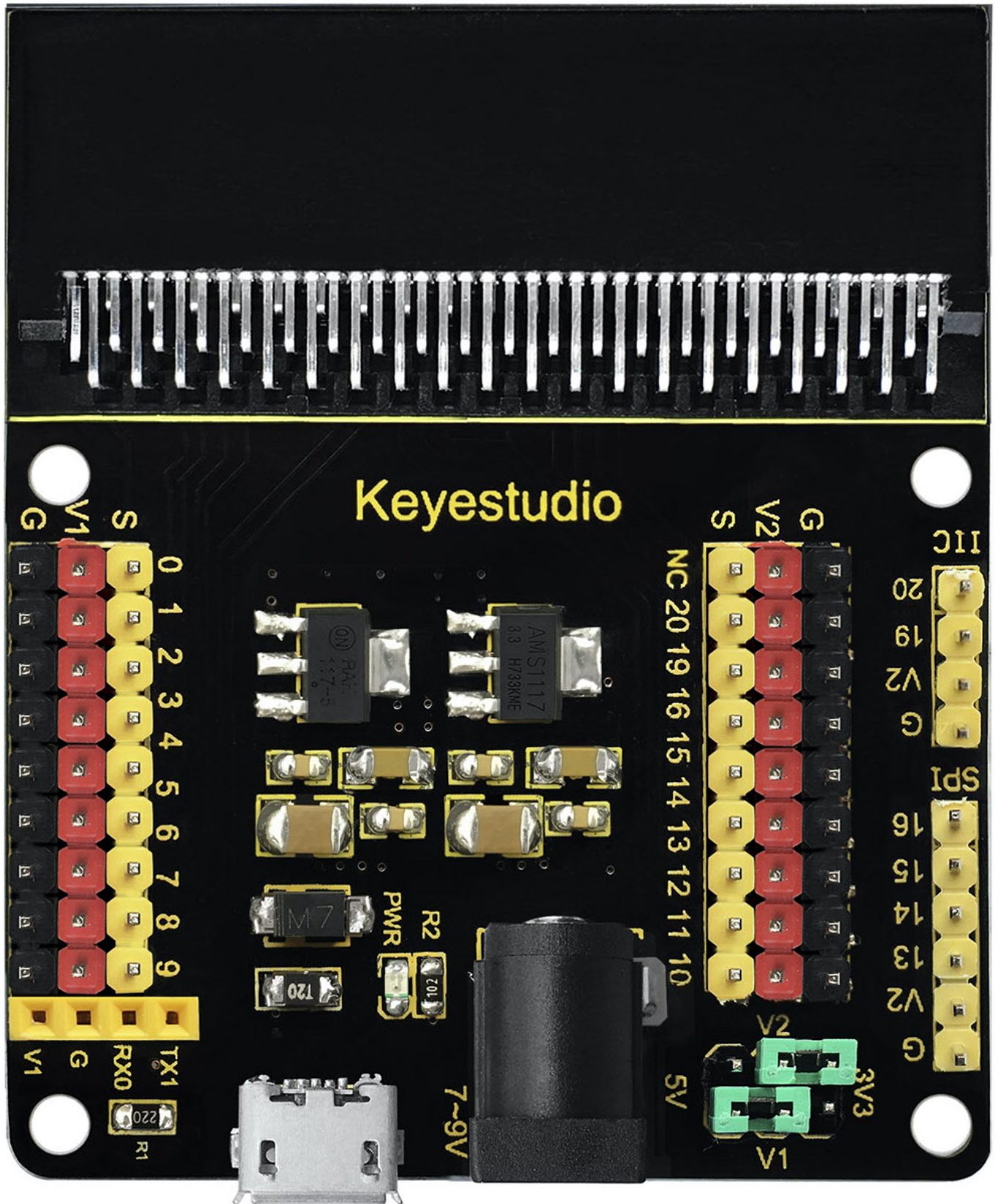


Kit avanzado

This image displays a wide variety of electronic modules from the Keyestudio brand, arranged in a grid-like fashion. The modules include:

- Keyestudio Main Board:** A large black board with numerous pins and components.
- RF Modules:** Two modules with circular antennas, one red and one blue.
- Bluetooth Module:** A small blue module with a USB cable.
- LCD Display:** A green module with a blue liquid crystal display.
- Power Supply:** A black module with a USB cable and a blue DC-DC converter.
- Relay Module:** A blue module with a yellow relay.
- Various Sensors:** Modules for temperature, humidity, light, and other environmental measurements.
- Actuators:** Modules for controlling motors, solenoids, and other mechanical devices.
- Communication Modules:** Modules for RS485, RS422, and other serial communication protocols.
- Other Modules:** Modules for controlling LEDs, relays, and other electronic components.

1 / 10



Extensor

Proyectos de iluminación

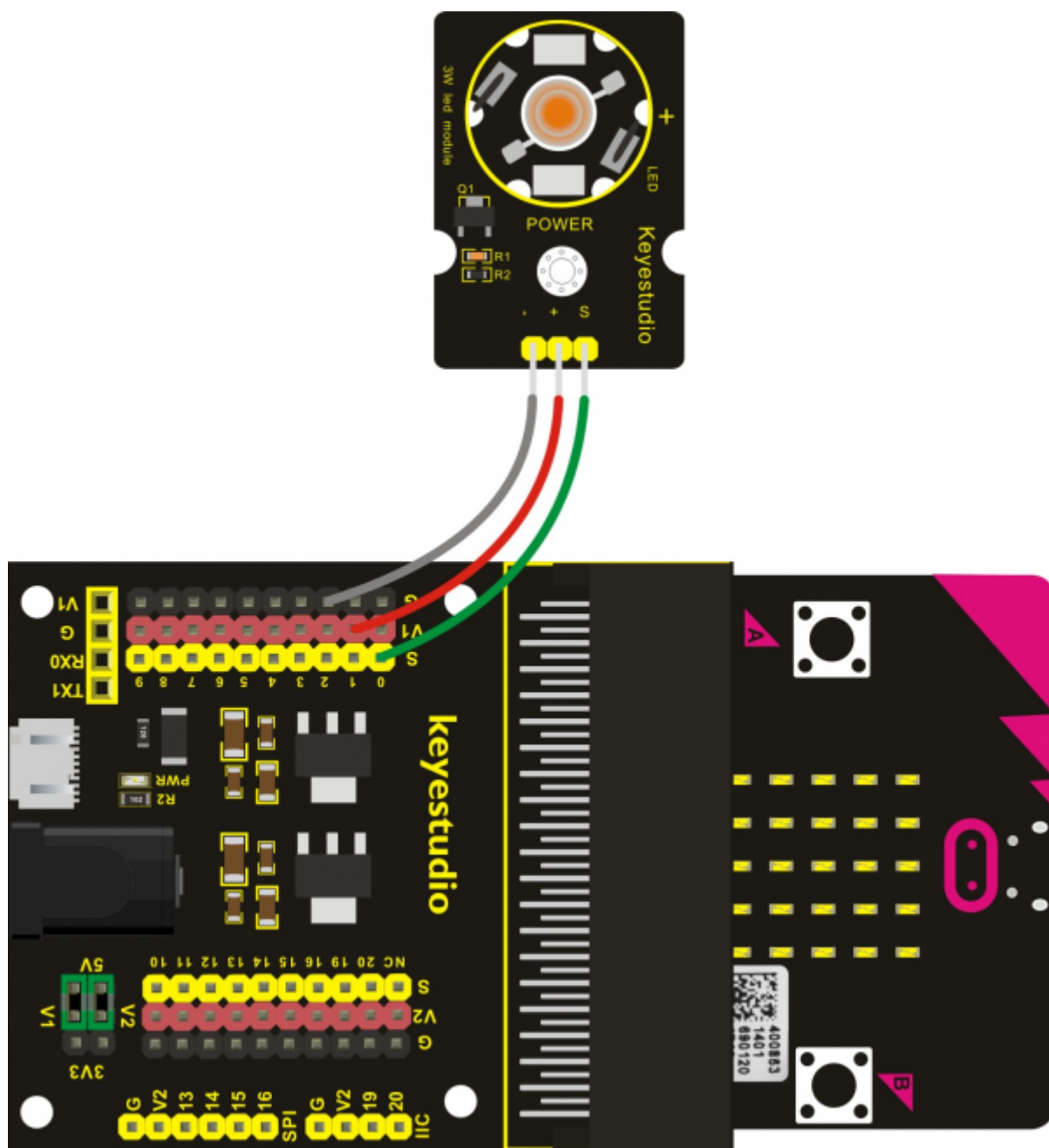
¡¡Cuidado con la alimentación!! Ya empezamos a añadir dispositivos que consumen. Recuerda alimentar el extensor o bien con pilas o con su conector USB. Si llegamos a conectar más dispositivos puedes usar un cargador de móvil USB para alimentarlo, que el puerto USB del PC da sólo 500mA (poco para motores, servos o LEDs potentes)

Paradeo de led de 3W

Vamos a hacer que nuestro LED de 3W comience a parpadear tras pulsar el botón A, y lo detendremos al pulsar el botón B



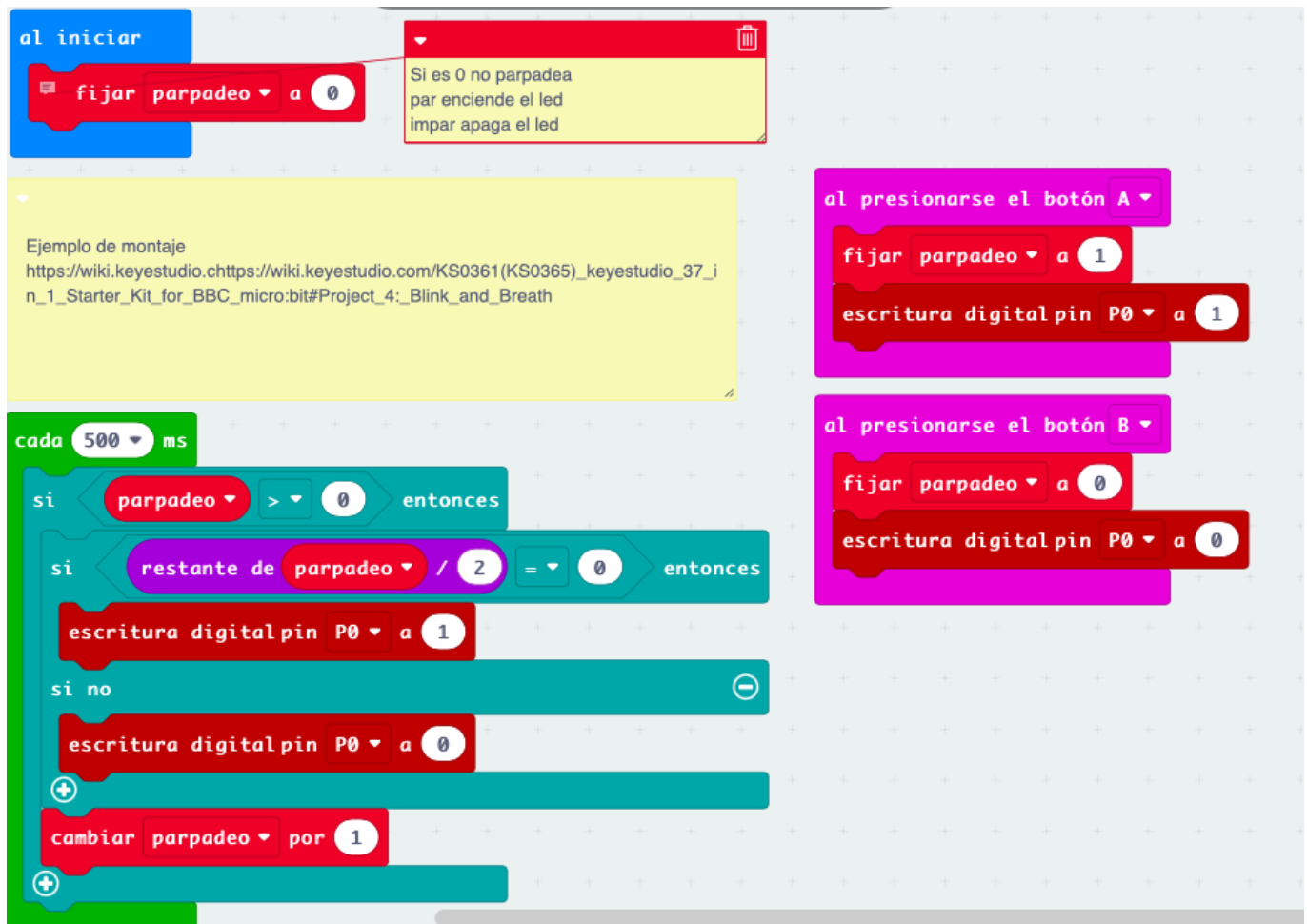
Podemos alimentar el LED con 3.3v o con 5V, lo conectaremos al pin 0



Para encender el led usamos el bloque "escritura digital" con el que activaremos el pin 0 al que está conectada la señal del control del LED. Esta instrucción digital enciende o apaga el led, todo o nada.

El programa es similar al que ya hemos hecho para hacer sonidos intermitentes: usamos una variable para controlar el estado del parpadeo:

- Si es 0 el parpadeo no está activo
- Si es distinto de 0 el parpadeo está activo:
 - Si es par se enciende el led
 - Si es impar se apaga el led
- Cada vez que apagamos o encendemos el led incrementamos el valor de la variable "parpadeo"



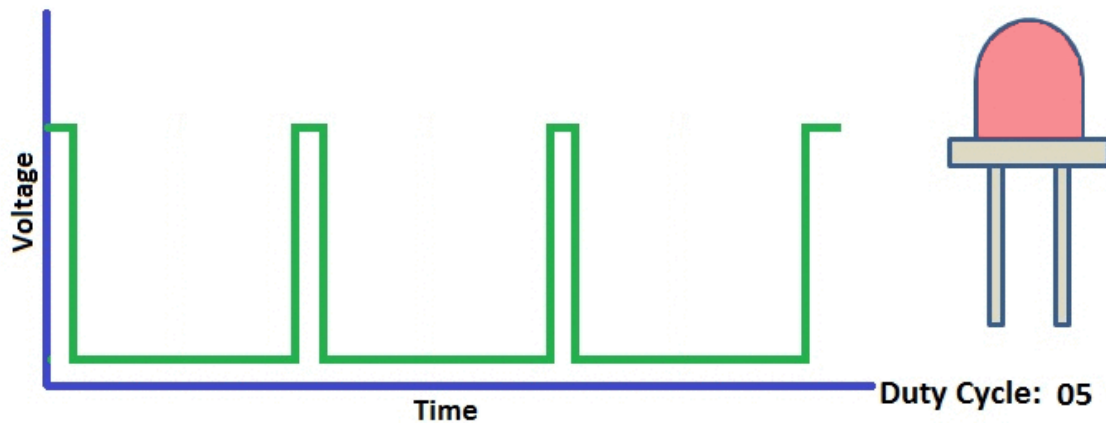
Al pulsar el botón "B" ponemos a 0 la variable "parpadeo" y apagamos el led por si estuviera activo

Parpadeo led 3W

Ejercicio opcional: permitir que se active o desactive el parpadeo remotamente con otra micro:bit usando la radio

Encendido progresivo del LED

Ahora vamos a hacer que el encendido o apagado del led se haga progresivamente, para ello usaremos una funcionalidad que tienen los microcontroladores que se llama PWM: se trata de usar la velocidad de procesamiento y lo lento que funciona nuestro ojo para encender o apagar rápidamente el LED, el efecto visual será que el LED sube o baja de brillo.



Realizamos un ciclo unas 500 veces por segundo y durante ese ciclo mantenemos el led encendido un % del tiempo. Si es el 100% el parecerá iluminado a máximo brillo, si es al 50% estará a medio gas.

Esta técnica también la podemos usar para controlar la velocidad de un motor DC de corriente continua. De hecho es lo que se usa para controlar la velocidad en los patinetes, drones u otros vehículos eléctricos. De ahí el característico zumbido que oímos.

El montaje es el mismo que antes.

Para el programos usaremos el bloque "escritura analógica" donde indicaremos el % de PWM con un número entre 0 y 1023 (para el 100%)

al presionarse el botón A

```

mostrar ícono [LED]
para index de 0 a 1023
  ejecutar
    escritura analógica pin P0 a index
    pausa (ms) 1
borrar la pantalla
        
```

al presionarse el botón B

```

mostrar ícono [LED]
fijar index a 1023
mientras index ≥ 0
  ejecutar
    escritura analógica pin P0 a index
    cambiar index por -1
    pausa (ms) 1
borrar la pantalla
        
```

Mejoras:

- * Usar una variable para guardar el nivel de luz y hacer el apagado o encendido desde ese valor.
- * Permitir que el apagado o encendido se pueda hacer en cualquier momento

Programa encendido progresivo

Otra forma de hacer el apagado progresivo es usando la idea feliz de que si la variable index hacer el bucle 0-1023, $(1023 - \text{index})$ hace justo el contrario...

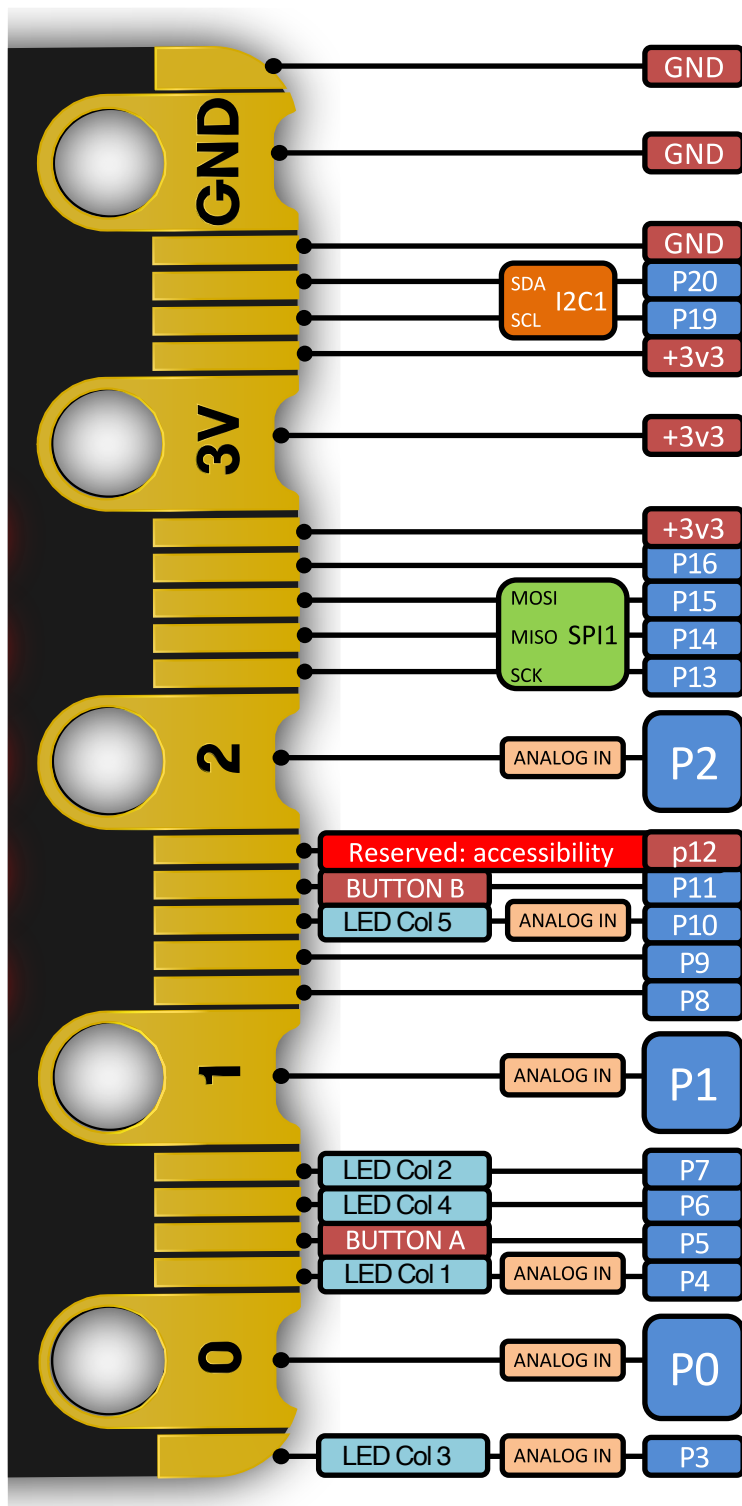


Mejoras:

- Usar una variable para guardar el nivel de luz y hacer el apagado o encendido desde ese valor.
- Permitir que el apagado o encendido se pueda hacer en cualquier momento

Selección de pines

Para seleccionar dónde conectar cada componente vamos a usar el [esquema de pines de la micro:bit](#) para ver las capacidades de cada pin. Más detalles en <https://microbit.pinout.xyz>



Estroboscopio

Si hacemos parpadear el led a cierta velocidad podemos conseguir un efecto que se llama estroboscópico y que consiste en que parece que detenemos el movimiento de las cosas.

Podemos usar un potenciómetro, que conectaremos a un pin analógico (P4), para controlar la velocidad del parpadeo



El potenciómetro da una lectura entre 0 y 1023 que será el valor que usaremos para hacer la espera entre parpadeos

Desactivamos la pantalla de leds de la microbit porque el pin P4 se usa para ello y habría conflicto

[Programa estroboscopio](#)

Iluminación automática

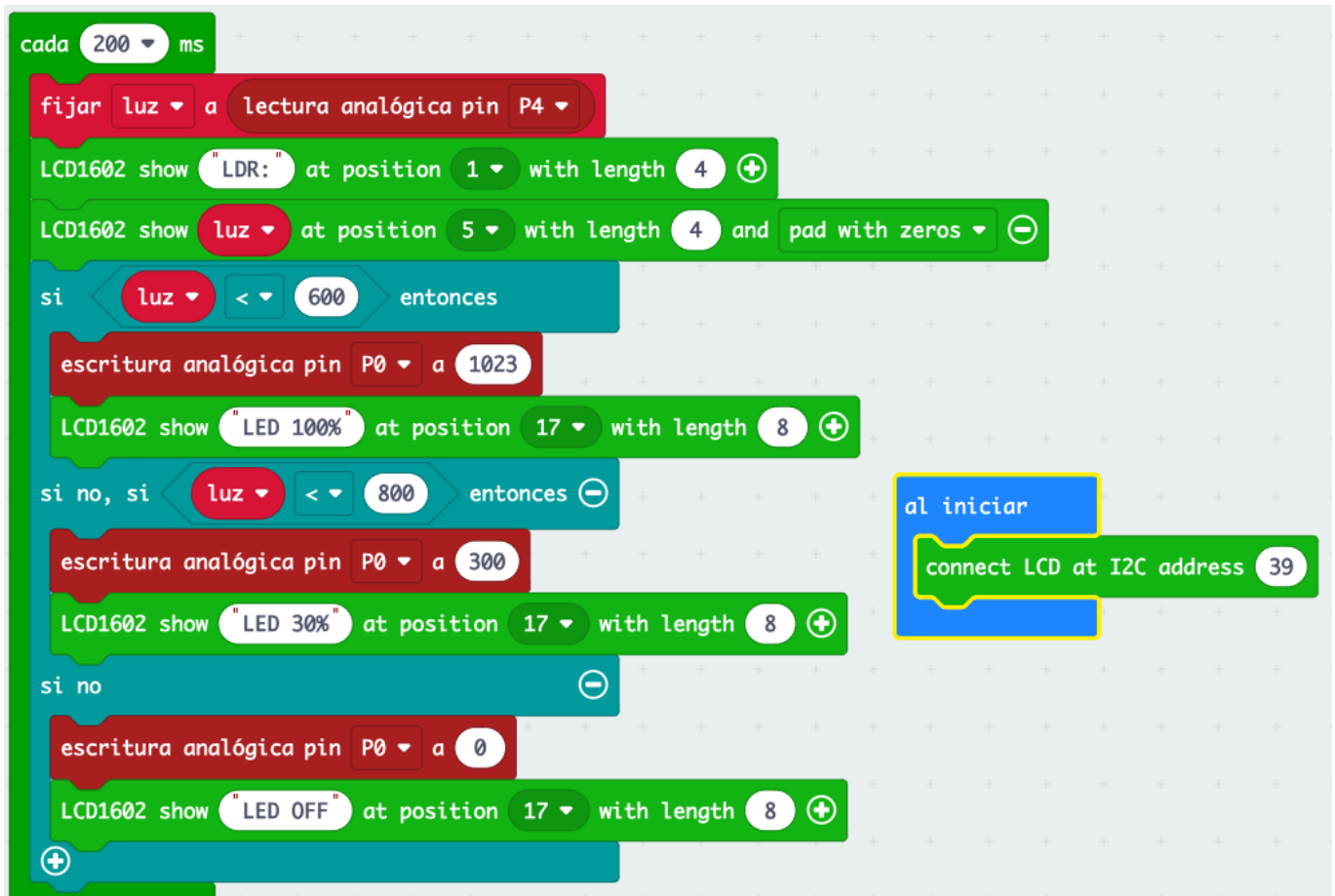
Vamos a añadir un sensor de luz (resistencia LDR) para medir la luz ambiental. En función del valor leído encenderemos más o menos nuestro led.



Para ver los valores que da nuestro sensor LDR, añadiremos la pantalla LCD donde mostraremos los valores.

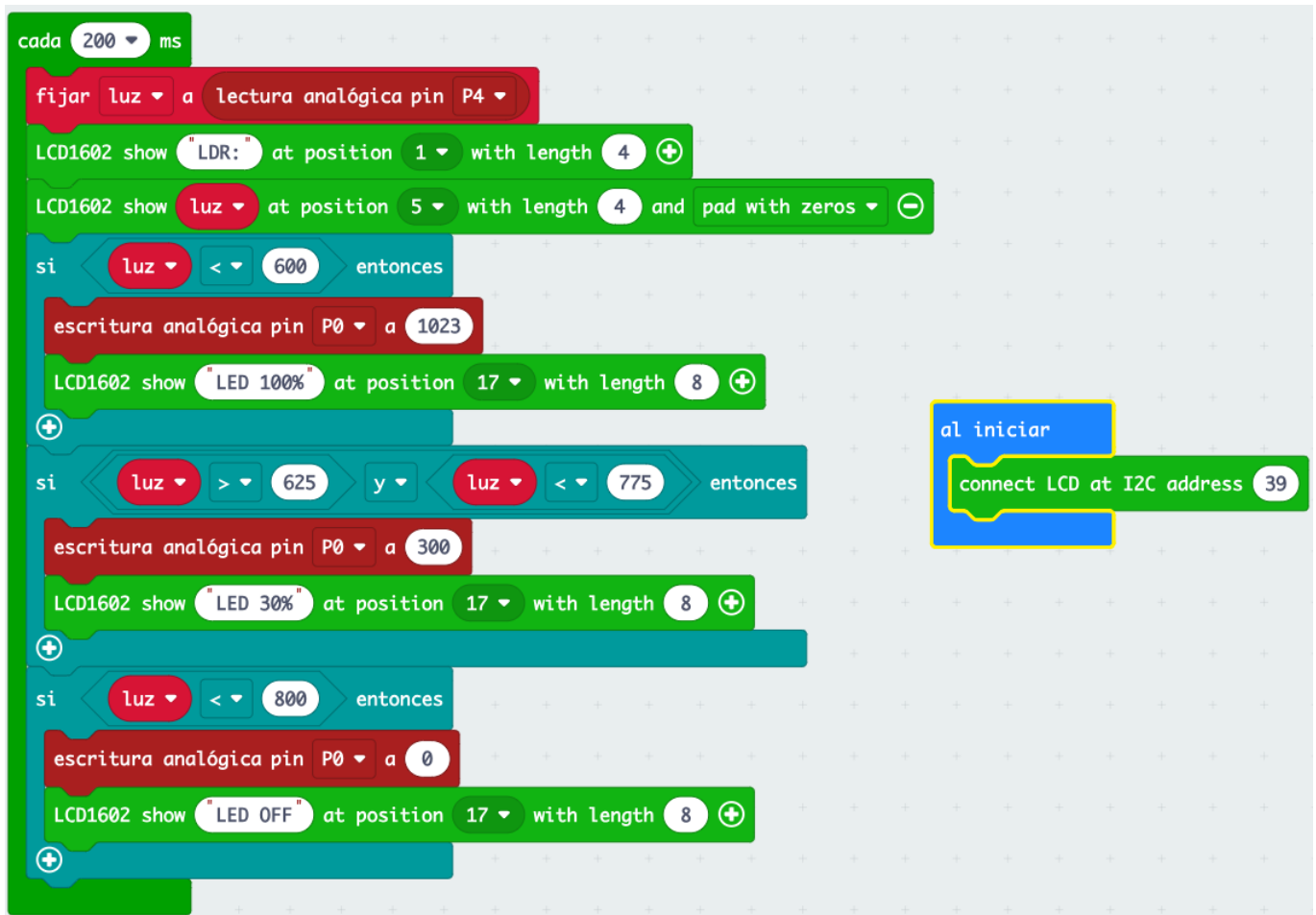
En mi caso tenemos valores entre 450 y 1000. Vamos a hacer 3 tramos:

- Por debajo de 600 donde encenderemos al 100% el LED
- Entre 600 y 800 donde lo encenderemos al 30%
- Por encima de 800 donde lo apagaremos



Programa iluminación automática

Para programar los intervalos también podíamos usar varios bloques if con el rango de valores, de esta forma podemos hacer que los rangos estén separados y no den lugar a problemas de intermitencias no deseadas.



En este caso hemos dejado una separación de 25 entre los estados. Este fenómeno se denomina histéresis

Programa iluminación automática v2