



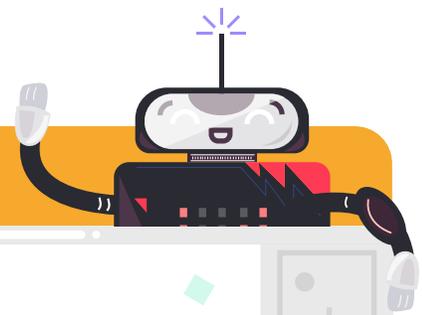
Proyecto Controlador de Luminosidad con micro:bit

Laboratorio de Investigación
e Innovación Tecnológica para
La Educación en Ciencias

LIITEC-ULS

La Serena - 2020





Descripción del proyecto

Controlador de luminosidad con micro:bits

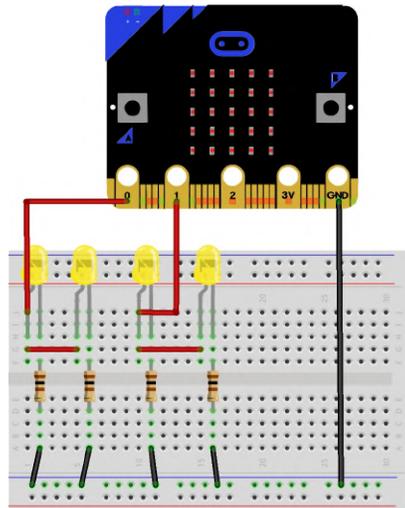


Imagen 1.

Controlador de luminosidad con micro:bits

Datos Personales :

Nombre: _____

Establecimiento : _____

Número de contacto : _____



Descripción del proyecto

Los componentes de la tarjeta micro:bit son los siguientes:

- Pantalla led
- Sensor de temperatura
- Sensor de luminosidad
- Acelerómetro
- Brújula
- Función de radio
- Módulo Bluetooth
- Botones
- Pines

Cada sensor o módulo dentro de la tarjeta cumple una función especial, pero en este caso usaremos el sensor de luminosidad que está incorporado en nuestra tarjeta. Este sensor de luminosidad está conformado por el display de led, que según el funcionamiento de un semiconductor emisor de luz como su nombre lo dice emite luz, pero también a la inversa, si es sometido a luminosidad este devuelve una corriente pequeña que se puede medir y así podemos tener un rango de medición de 0 a 255. En este rango el valor más bajo (0) es considerado como luminosidad nula y su máximo valor (255) es la presencia máxima de luz.

En este proyecto lo que harás será establecer rangos para ir prendiendo secuencialmente diodos leds y así ir iluminando según la luminosidad que haya en el ambiente. Luego te darás cuenta de que puede aplicarse con luminaria de alta eficiencia como las luces de tu casa o luces de exterior.

Además de ver bloques básicos como mostrar número o para siempre también verás bloques lógicos ya sean condicionales, comparadores y booleanos, por ejemplo:

Sentencias condicionales

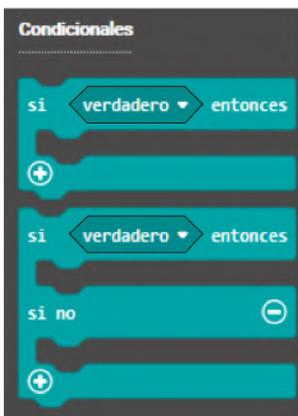


Imagen 2.

Las sentencias condicionales como su nombre lo indica condicionan a ejecutar alguna línea de código que esté en su interior. Si la sentencia de condición se cumple ejecutará el código, si no, saldrá de su ejecución y pasará a la siguiente línea de código

Controlador de luminosidad con micro:bits

Sentencias de comparación



Imagen 3.

Estas sentencias se encargan de comparar los valores que puedas asignarle, retornando un valor "verdadero" o "falso".

Ejemplo

3 es menor (<) que 5 nos dará un resultado verdadero

8 es mayor (>) que 35 nos dará un resultado falso

5 es menor (<) que 5 nos dará un resultado falso debido a que son iguales

Sentencias booleanas



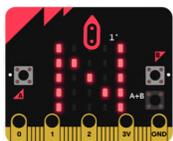
Imagen 4.

Tal como matemáticamente existen operaciones como adición y sustracción, multiplicación y división, en el mundo de la programación existen 3 operaciones lógicas que retornarán valores verdaderos o falsos.

Expresión	Significado
a y b	Al comparar los valores, si queremos obtener un resultado verdadero, ambas sentencias o variables deben ser verdaderas, si una de estas es falsa el resultado también lo será.
a o b	Al comparar los valores, si queremos obtener un resultado verdadero, solo necesitamos que una de las sentencias o variables sea verdadera.
NO a	Es una negación del valor evaluado, por lo tanto, para obtener un resultado verdadero se necesita un valor falso.

Paso 1

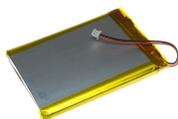
Reúne los siguientes materiales:



Placa micro:bit



Cable USB a micro-USB



Batería de 3,7 V.



4 Leds



4 resistencias de 100 Ω ohm



1 Protoboard



Cables



Conectores

Paso 2

Conecta la tarjeta micro:bit a tu computador mediante el cable USB.

Paso 3

Ejecuta el programa MakeCode descargable en link:

<https://www.microsoft.com/es-cl/p/makecode-for-micro-bit/9pjc7sv48lcx?activetab=pivot:overviewtab>

o ejecuta en tu navegador el editor de código online:

<https://makecode.microbit.org/>

Paso 4

Ahora construirás un Controlador de Luminosidad con micro:bit, ten en cuenta los siguientes consejos :



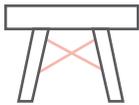
Duración de actividad:

45 minutos aprox.



Comodidad

Ubicate en un lugar cómodo



Superficie

Trabaja en una superficie plana



Luminosidad

Trabaja en un lugar iluminado



Toma apuntes

Escribe tus registros



Comparte tu experiencia

Tu historia puede servir a otros

Controlador de luminosidad con micro:bits

Comparte tu experiencia en RRSS usando [#liiteculs](#)

Paso 4.1

Desde la sección Básico arrastra el bloque “*para siempre*”, que nos dará el inicio de nuestro programa y lo ejecutará permanentemente.

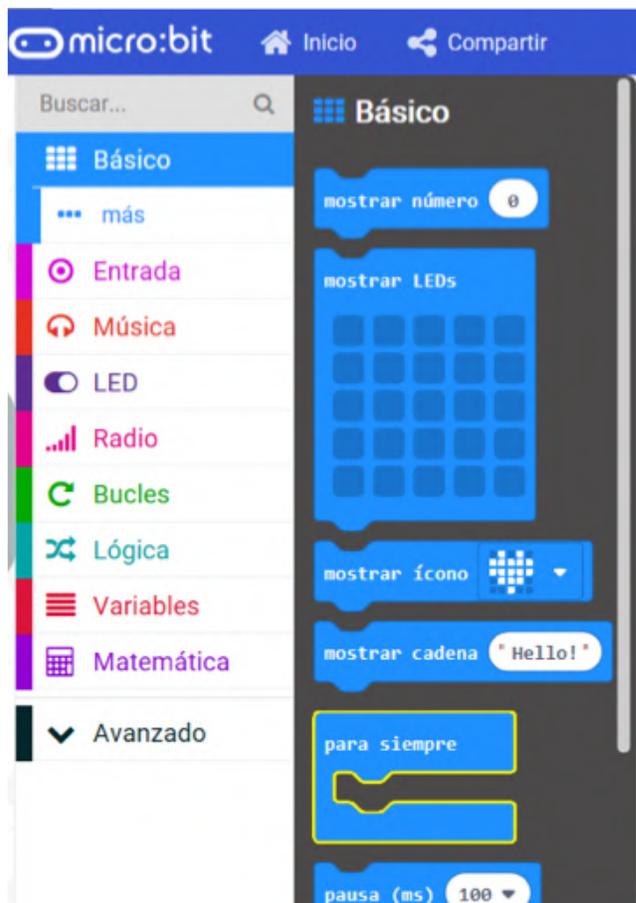


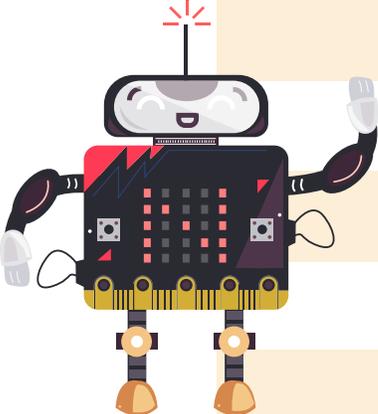
Imagen 5.



Imagen 6.

Paso 4.2

El sensor de luminosidad de tu tarjeta micro:bit mide en un rango entre 0 y 255 en donde 0 es luminosidad nula y 255 es luminosidad alta o máximo de luminosidad. Por lo que dividiremos este rango en 4 rangos para tener 4 estados de luces o luminosidades y un estado final de apagado total (ver tabla).



Nivel de luz (255 Claridad -0 Oscuridad)	Luces encendidas
0-64	4 luces al 100%
65-128	4 luces al 50%
129 - 192	2 luces al 100%
193 - 254	2 luces al 50%
255	4 luces al 0%

Siguiendo estos rangos necesitaríamos condicionar 5 veces por la luminosidad que captura el sensor, por lo que de la sección de Lógica arrastra el bloque llamado **“Si, entonces, si no”** dentro del boque **“para siempre”**, y presionamos 4 veces el signo (+) que tiene debajo el bloque y 1 vez el signo (-) para que quede como la imagen 9. Con esto lograremos el espacio para hacer las 5 condiciones en donde se evaluará en cuál de los rangos queda el valor del sensor.

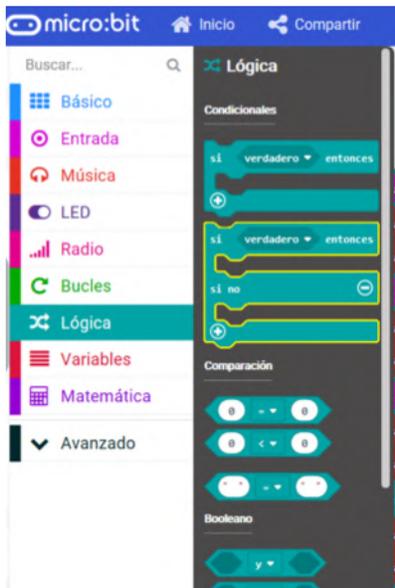


Imagen 7.

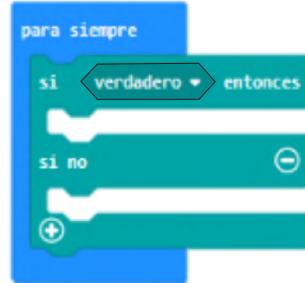


Imagen 8.



Imagen 9.

Paso 4.3

Arrastra el bloque de comparación “*menor que*”, que se puede identificar por el símbolo < ubicado en la sección Lógica. Duplica el bloque para que tengas 5 iguales y arrástralos en cada condición de las preguntas “*Si, entonces, si no*” (Ver imagen 12).

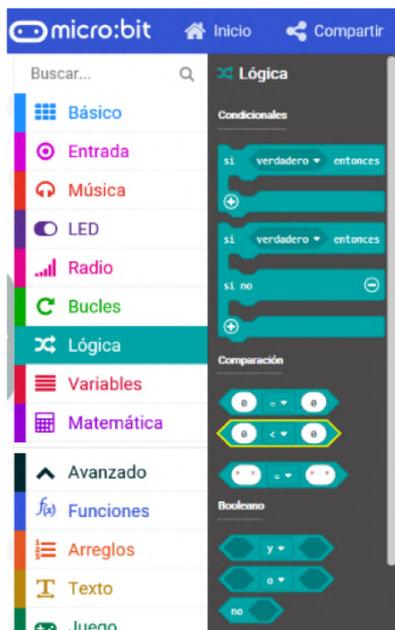


Imagen 10.

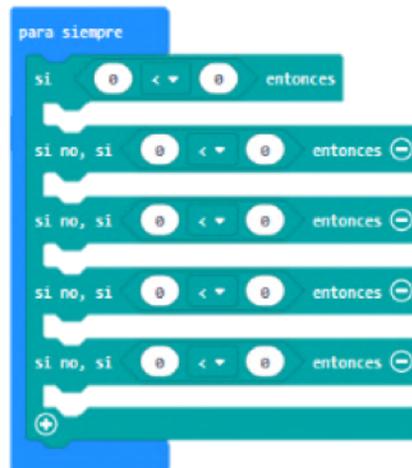


Imagen 11.

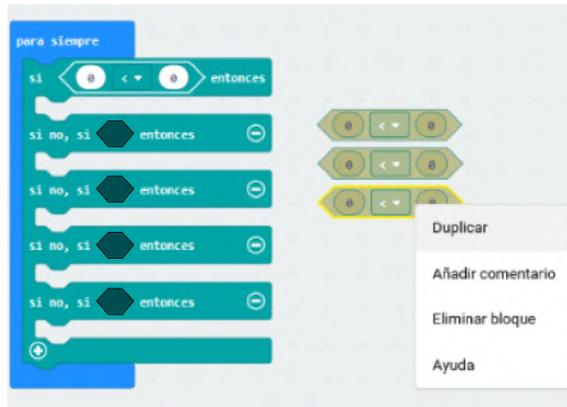


Imagen 12.

Paso 4.4

De la sección Entrada arrastra el bloque *“nivel de luz”* y duplica hasta tener 5 bloques iguales, que debes ubicar en las condiciones de *“Si, entonces, si no”* (ver imagen 14), cuando tengas el *“nivel de luz”* en cada condición tendrás una parte de la condición lista, sólo te faltaría el valor con el cuál lo compararás.

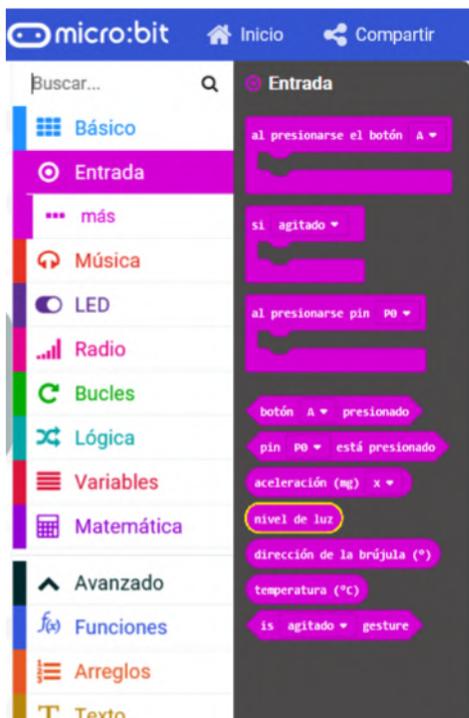


Imagen 13.

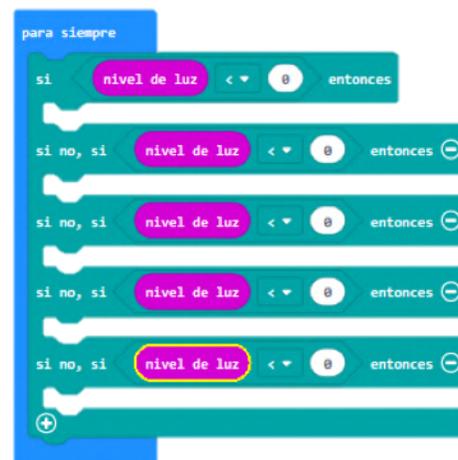


Imagen 14.

Paso 4.5

Modifica los valores de las condiciones que quedan como óvalos blancos con los siguientes valores: 64, 128, 192, 254 y 255; y las condicionantes deben ir con el símbolo que compara el número y pregunta si el valor es menor o igual (\leq).

De esta forma tendremos las siguientes condiciones:



- Nivel de luz \leq 64
- Nivel de luz \leq 128
- Nivel de luz \leq 192
- Nivel de luz \leq 254
- Nivel de luz \leq 255

```

para siempre
si nivel de luz < 0 entonces
si no, si nivel de luz < 0 entonces
si no, si nivel de luz < 0 entonces
si no, si nivel de luz < 0 entonces
si no, si nivel de luz < 0 entonces
    
```

Imagen 15.

```

para siempre
si nivel de luz < 64 entonces
si no, si nivel de luz < 128 entonces
si no, si nivel de luz < 192 entonces
si no, si nivel de luz < 254 entonces
si no, si nivel de luz < 255 entonces
    
```

Imagen 16.

Paso 4.6

Dentro de la sección Avanzado hay otras secciones y aquí se encuentra la sección Pines de donde arrastrarás el bloque “*Escritura digital pin P0 a*” y luego necesitarás 2 de estos bloques por condición por lo que vamos a duplicar hasta tener 10 bloques iguales.

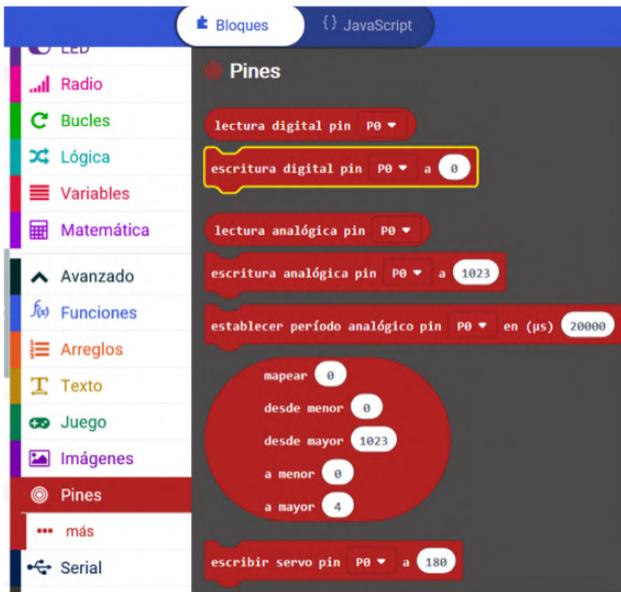


Imagen 17.



Imagen 18.

Paso 4.7

Por último, modifica los valores pines para que queden intercalados entre el P0 Y P1, para luego modificar los valores de los óvalos blancos con los siguientes valores:

	P0	P1
1ra condicional	1023	1023
2da condicional	512	512
3ra condicional	1023	0
4ta condicional	512	0
5ta condicional	0	0

```

para siempre
si nivel de luz ≤ 64 entonces
  escritura analógica pin P0 a 1023
  escritura analógica pin P1 a 1023
si no, si nivel de luz ≤ 128 entonces
  escritura analógica pin P0 a 512
  escritura analógica pin P1 a 512
si no, si nivel de luz ≤ 192 entonces
  escritura analógica pin P0 a 1023
  escritura analógica pin P1 a 0
si no, si nivel de luz ≤ 254 entonces
  escritura analógica pin P0 a 512
  escritura analógica pin P1 a 0
si no, si nivel de luz = 255 entonces
  escritura analógica pin P0 a 0
  escritura analógica pin P1 a 0
  
```

Imagen 19.

Este programa completo lo que hace es: tomar el valor que tiene el sensor de luminosidad y compararlo en cada condición hasta que se cumpla alguna y ejecutará los bloques que hay es su interior. Según los bloques en la primera condición se encenderán las 4 luces al 100%, en la segunda se encenderán las 4 luces, pero al 50%, en la tercera condición solo encenderán 2 luces al 100%, en la cuarta condición encenderán 2 luces al 50% y en la última condición se apagaran todas las luces.

¡Con esto terminaste de programar tu controlador de luminosidad!



Paso 4.8

Arma tu circuito.

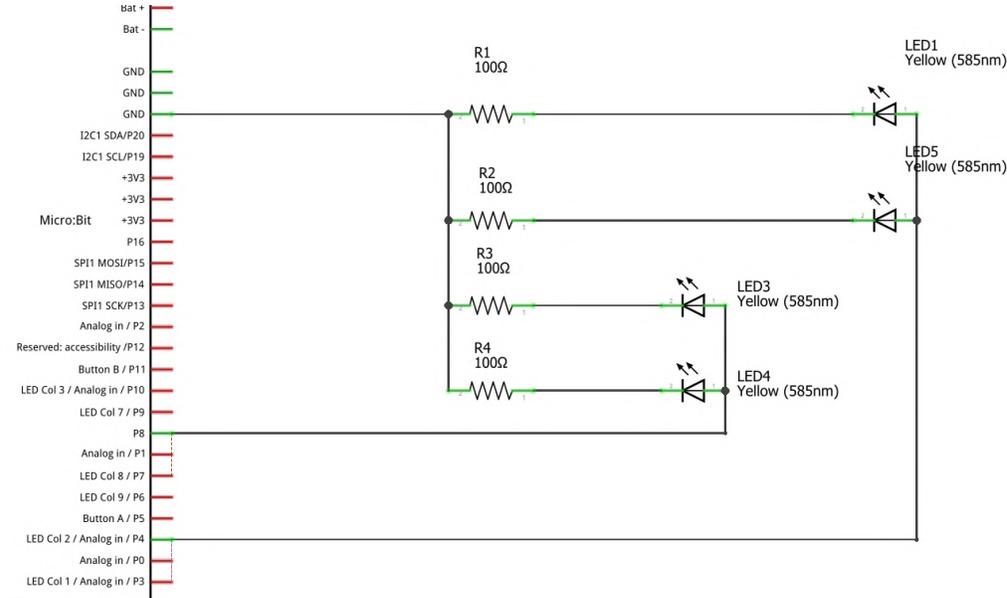


Imagen 20.

Controlador de luminosidad con micro:bits



Este manual ha sido elaborado por LIITEC-ULS (Laboratorio de Investigación e Innovación Tecnológica para la Educación en Ciencias), Universidad de La Serena, Chile.

Equipo de LIITEC-ULS

Producción y creación

Elaboración:

- Tomás Carvajal Rojas

Diseño y diagramación:

- Valeria Pizarro Valdivia

Colaboración y correcciones:

- Ariel Pérez Fernández
- Camilo Ibacache Ibacache
- Mario Zepeda Marambio
- Jimena Núñez Valdés
- Francisco López Cortés

Agradecimientos:

Se agradece la colaboración de los siguientes colegios pertenecientes a la Corporación Gabriel Gonzales Videla

- Liceo Gabriela Mistral
- Liceo Marta Brunnet
- Liceo Jorge Alessandri Rodríguez
- Colegio Gabriel González Videla
- Colegio José Manuel Balmaceda
- Colegio José Miguel Carrera
- Colegio Lambert
- Colegio Arturo Prat Chacón
- Colegio Carlos Condell de la Haza