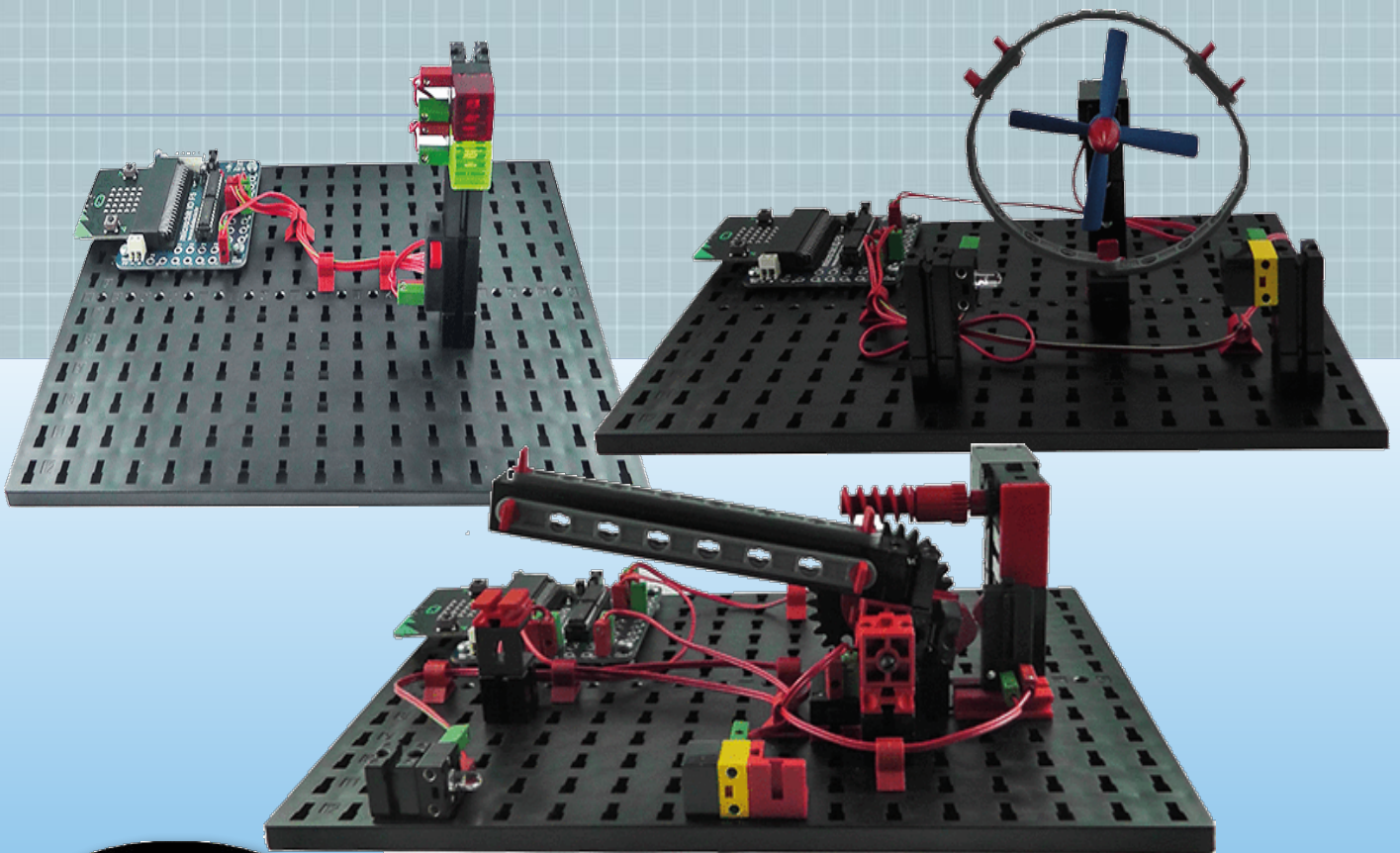
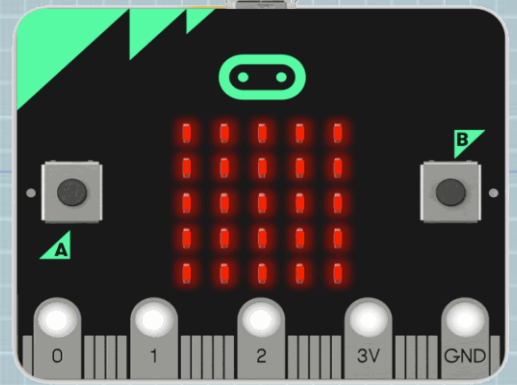


Programación en la escuela primaria

fischertechnik 

micro:bit



3 MODELOS

Contenido

Prólogo	p. 3
Preparación	p. 4
Prueba de funcionamiento	p. 5
Editor	p. 6
Guardar	p. 6
Cargar	p. 7
Bloques/comandos	p. 7
Semáforo	p. 8
Semáforo para peatones con pulsador de solicitud	p. 9
Actuadores/sensores	p. 10
Pulsador	p. 10
LED	p. 11
Semáforo para peatones con luz intermitente	p. 14
Secador de manos	p. 15
Secador de manos con barrera de luz	p. 16
Fototransistor	p. 17
Motor	p. 17
Secador de manos con indicador LED	p. 20
Barrera	p. 21
Barrera de aparcamiento	p. 22
Barrera de aparcamiento con indicador óptico	p. 28
Posibles errores	p. 30

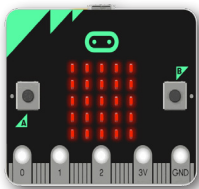
Prólogo



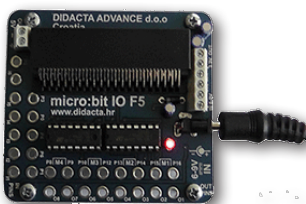
¡Hola!
Permítanme presentarme: me llamo
RoBo y te acompañaré mientras
completas las tareas individuales.

Antes de comenzar con las tareas debo explicarte algunos conceptos básicos.

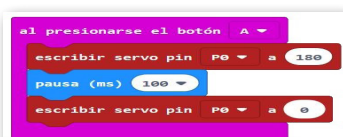
Con los componentes de fischertechnik puedes construir los correspondientes modelos que quieres controlar.



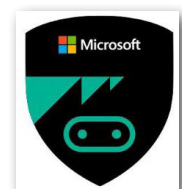
El micro:bit es un pequeño ordenador programable. Este cuenta con una pantalla LED, dos teclas, un sensor de luz, una antena de Bluetooth, así como un sensor de aceleración y una brújula.



El micro:bit se conecta al micro:bit IO F5 mediante una regleta de contactos de cuchilla. Se puede acceder a todas las entradas y salidas a través de conectores hembra. Al mismo tiempo se puede conectar una fuente de energía.



Con unos pocos clics en makecode puedes generar tus propios programas para el microprocesador y poner tus modelos en movimiento.



Que disfrutes realizando las distintas tareas.

Tu RoBo

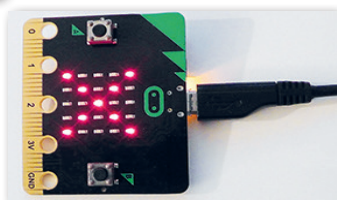
Preparación

Antes de comenzar con la construcción de los modelos y la programación debes realizar algunos preparativos.

Primero, desempaca tu micro:bit. Lo recibirás en una pequeña caja de cartón.



Importante: Si más tarde lo quieres conectar en la regleta de conectores del adaptador de tu modelo, las teclas A y B deben apuntar hacia arriba.



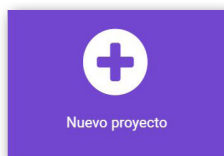
Con un cable USB, conecta el micro:bit a un puerto USB libre en tu ordenador. Del lado inferior del micro:bit se encenderá un diodo emisor de luz amarillo. En la matriz LED del lado superior se verá una X roja.



Abre tu navegador de Internet y abre el editor de programa en

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

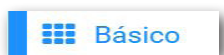
Prueba de funcionamiento



Haz clic en "Nuevo proyecto". Se abrirá la pantalla de trabajo de makecode. En el siguiente capítulo te lo explicaré en detalle. Aquí solo se trata de probar el programa en combinación con el micro:bit.



En la opción "Más..." cambia el lenguaje de programación a "Español".



Con el ratón, haz clic en la selección "Básico". En este bloque aparecen todos los comandos de este área de bloque.

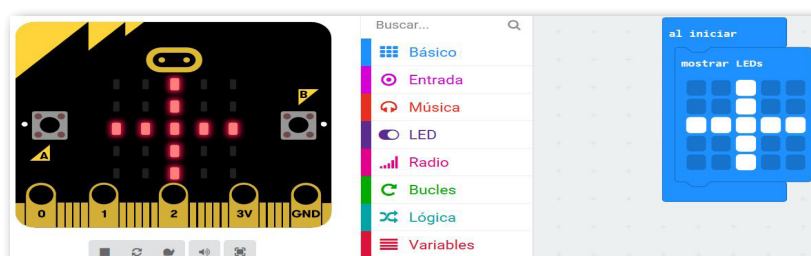


Para nuestra prueba usarás el comando "mostrar LEDs".

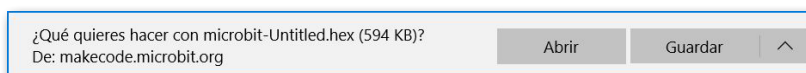
Arrástralo con el ratón hasta el campo vacío del comando "al iniciar".

Cuando haces clic en los LED individuales, se activan y se representan en la simulación.

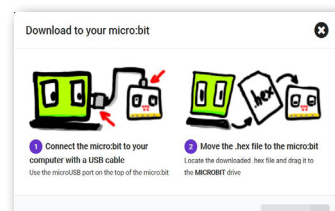
Haz un signo de más (+) con los LED. Tu pantalla de trabajo se debería ver como se ve en la imagen.



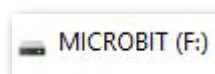
Con la tecla "Descargar" se envía el programa al micro:bit. Si activas la tecla aparecerá el siguiente texto en el borde inferior de la pantalla:



Activa la flecha que se encuentra detrás de "Guardar" y selecciona "Guardar cómo".



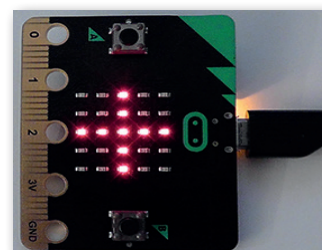
Se abrirá la pantalla de trabajo "Guardar cómo". Selecciona la unidad de USB a la que has conectado el micro:bit. Este se indica visualmente.



Importante: El tipo de archivo debe ser un archivo HEX.



Si haces clic en "Guardar" se apagan los LED del micro:bit y parpadea el LED amarillo del lado inferior. Esto indica que se está transmitiendo el programa. Una vez completado este proceso, se muestra el signo de más (+).



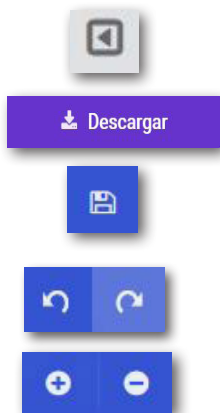
Si todo ha funcionado bien, quisiera explicarte el editor con más detalle.

Editor

La pantalla cuenta con tres áreas: Simulación, comandos de programación y pantalla de programa.



Otras teclas importantes:



Iniciar/Detener el simulador

Descargar el programa al micro:bit

Guardar el proyecto

Deshacer

Acercar/Alejar



Detener el simulador

Reiniciar el simulador

Cámara lenta



Silenciar el audio

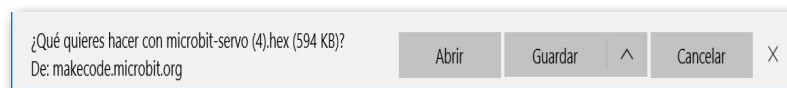


Abrir en pantalla completa

Guardar

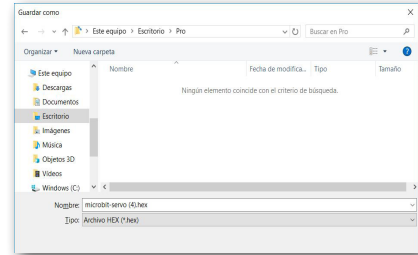


Si has terminado un proyecto debes guardarlo. Para ello debes seleccionar la tecla "Guardar".



Haz clic en la flecha que se encuentra al lado de "Guardar" y luego en "Guardar cómo".

Selecciona tu carpeta o crea una nueva carpeta en la que desees guardar tu proyecto.



Importante: Elige un nombre significativo para tu proyecto.

Si quieres seguir trabajando en un proyecto terminado o en curso, debes cargarlo desde tu dispositivo de memoria.

Para ello debes estar en la página de inicio. Te encontrarás en ella cuando abres makecode.



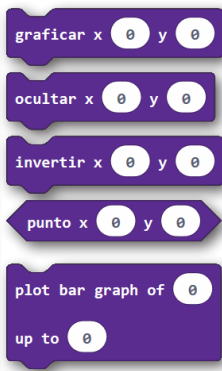
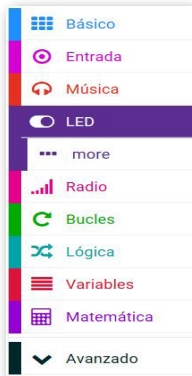
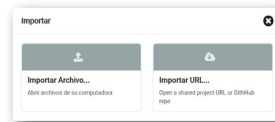
Cargar

Puedes encontrar tus proyectos bajo "Mis proyectos".

También puedes seleccionar la tecla "Importar". Aparece un menú de contexto. Selecciona "Importar archivo".



Aparece otra ventana de contexto. Con la tecla "Examinar" abres tu soporte de datos. Cuando haya encontrado tu proyecto haz clic en "¡Adelante!".



En las instrucciones verás los términos "Bloque" y "Comandos". En makecode, todos los comandos están organizados por bloques según su función. En la imagen de la izquierda puedes ver un ejemplo para el bloque "LED".

Bloques Comandos

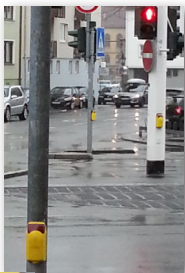
Bueno, ya completamos este punto y ahora te puedes atrever con tu primer modelo.

Semáforo



Seguramente conozcas distintas versiones de semáforos. Casi todos los días te cruzas con semáforos para peatones o semáforos en intersecciones, por lo que no son nada nuevo. Explicado de forma simple, se trata de lámparas que se encienden y apagan en un orden determinado.

Lo que distinguirá a tu semáforo, es que se encenderá mediante un pulsador que se encuentra en su mástil.



Primero, el técnico enciende el sistema desde un armario eléctrico. Después se enciende la lámpara roja. Cuando el peatón presiona el pulsador de solicitud el sistema conmuta a verde y se puede cruzar la calle.

Si a la vuelta de la escuela o en un paseo con tus padres pasas por un semáforo para peatones, tomate un momento para observarlo.



Hazte las siguientes preguntas:

- ¿El sistema está encendido?
- ¿Cuál de las lámparas brilla?
- ¿El sistema cuenta con un pulsador de solicitud de señal?
- ¿Qué sucede cuando lo presionas?
- ¿Cuánto tiempo dura la fase verde?
- ¿Qué sucede después de la fase verde?



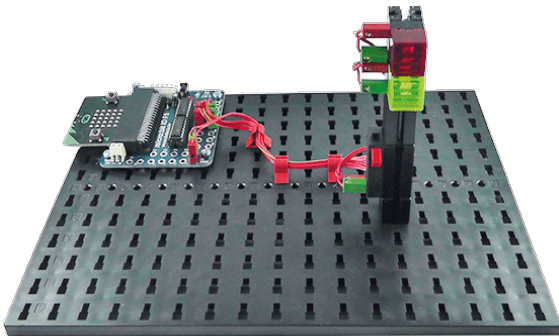
Anótate las respuestas en una hoja de papel y luego utiliza los resultados para los comandos de programación individuales.

Importante: Antes de poner en funcionamiento los modelos reales puede probar tus programas con el simulador de programas.

Nota técnica: Necesitarás una fuente fischertechnik para todos los modelos. Esta no está incluida en el paquete y se debe adquirir de fischertechnik.



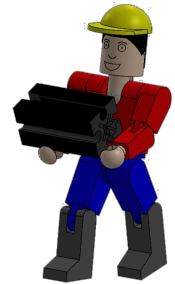
Semáforo para peatones con pulsador de solicitud



Arma tu primer modelo siguiendo las instrucciones.

Cablea el modelo siguiendo el diagrama de cableado.

Para este modelo te daré una tarea y una tarea adicional. Luego implementarás ambas en un programa de control.



Implementarás ambas en un programa de control.

El sistema completo se debe encender mediante un interruptor "Pulsador A" en el micro:bit. El semáforo estará en rojo. Recién cuando se acciona el pulsador de solicitud del mástil, el semáforo cambiará de rojo a verde después de 2 segundos. La fase verde debe durar 5 segundos. Después, el sistema vuelve a rojo y espera a que se vuelva a accionar el pulsador de solicitud.

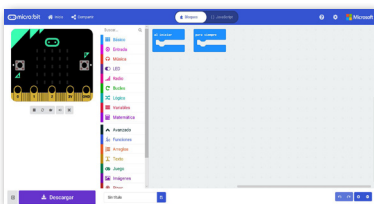
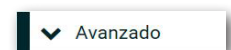
semáforo1.hex



Crearemos el primer proyecto juntos.

A continuación, abre el editor "makecode". Aparece la pantalla que ya conoces.

Para poder trabajar con todos los comandos que están a disposición debes ampliar la barra de comandos con "Avanzado".



Importante: Asegúrate de haber activado el área "Avanzado" de los bloques de makecode con la tecla correspondiente.



Actuadores Sensores

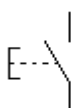
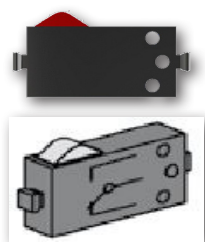


Alto; primero debo explicarte los dos componentes electro-técnicos que se usan en este modelo. Según su función se los llama "actuadores" o "sensores".

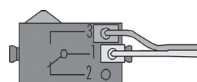


Los actuadores se llaman así porque son activos, hacen algo, como por ejemplo encender un motor o una lámpara. Con los sensores, por ejemplo un pulsador, se pueden controlar actuadores.

Pulsador



Símbolo de circuito



Quisiera empezar con un sensor: el pulsador

Los pulsadores cuentan como sensores de contacto. Si activas el botón rojo, se mueve de manera mecánica un contacto en la carcasa y fluye corriente entre las conexiones 1 y 3. Al mismo tiempo, se interrumpe el contacto entre las conexiones 1 y 2. Esto quiere decir que en tus modelos puedes usar el pulsador como conmutador.



LED



A continuación veremos un actuador, el diodo emisor de luz o LED.

La abreviatura LED proviene del término inglés "light emitting diode". Traducido: "diodo emisor de luz". Si fluye corriente a través del diodo (sentido directo), emite luz.

En el kit de construcción encontrarás dos bloques de construcción LED. Los puedes usar como lámpara normal pero también como generador de señal para una barrera de luz.



Símbolo de circuito

En las conexiones eléctricas debes prestar atención a la polaridad correcta.



Te explicaré los demás componentes en la tarea correspondiente.

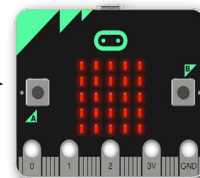
Ahora ya hemos explicado todo lo importante y puedes comenzar con tu primer programa.

Comencemos en orden. Cada programa que escribes siempre comienza con el comando "Inicio".

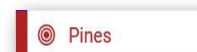
En la tarea se indica que el programa se inicia presionando la tecla A en el micro:bit.



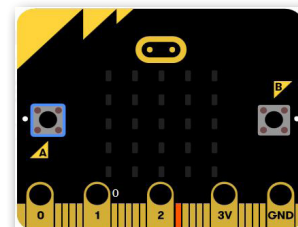
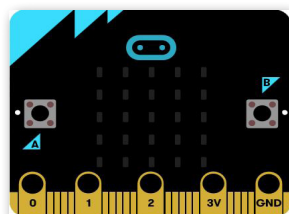
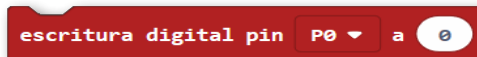
Utiliza el comando "al presionar el botón ..." del bloque "Entrada". Arrastra el comando a tu pantalla de programa.

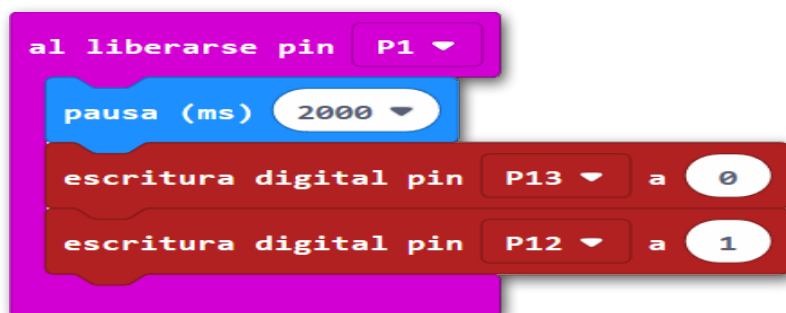
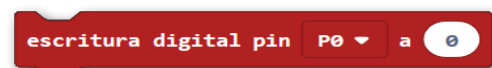
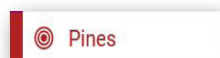
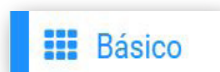
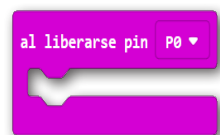
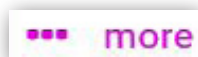


Si se presiona el botón A en el micro:bit se debe encender la lámpara roja (LED en el PIN13). Inserta el comando del bloque "Pines" en el espacio de "al presionar el botón ...". Cambia la entrada de P0 a P13 y el valor de 0 a 1.



Has completado la primera parte del programa. Ya puedes ver el resultado en la simulación. Con el ratón, haz clic en la tecla A. El pin 1 pasa a encendido, se le asigna un 1 lógico y se representa en rojo. Si ya has conectado el modelo a tu ordenador también se debería encender la lámpara roja.





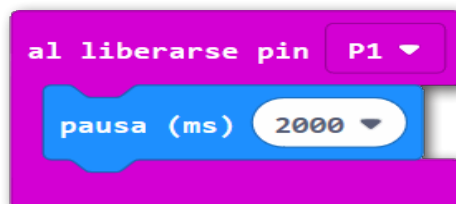
Los bloques "al iniciar" y "para siempre" no se necesitan y por lo tanto se pueden borrar. Haz clic con el botón derecho del ratón sobre el comando y selecciona "Eliminar bloques" en el menú de contexto.

A continuación crearás la parte del programa con el que se incluye el comando "al liberarse pin P1" del bloque "Entrada". Arrastra el comando a tu pantalla de trabajo.

Cambia la entrada "P0" a "P1".

Ahora sigue un comando de pausa. Este lo encontrarás en el bloque "Básico" "pausa (ms) xxxx".

Arrastra el comando hasta el punto de acoplamiento y modifica el tiempo a 2000, que equivale a 2 segundos.

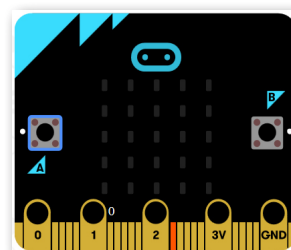


¿Qué sucede a continuación? Si se presiona la tecla de "P1" se debe apagar la lámpara roja y encender la lámpara verde. La lámpara roja está conectada al "pin13" y la lámpara verde, al "pin12". Aquí necesitas el comando "escritura digital pin" de bloque "Pines". Lo debes insertar debajo del comando de pausa.

Como necesitas el comando dos veces, simplemente puedes duplicarlo. Haz clic con el botón derecho del ratón en el comando y selecciona "Duplicar" en el menú de contexto. Acopla el segundo comando debajo de la última

entrada. En el primer comando, cambia "P1" a "P13" y conmuta de "1" a "0". En el segundo comando, cambia de "P1" a "P12" y de "0" a "1".

Puedes volver a probar tu programa parcial. Acciona la tecla virtual A para la simulación. El pin 13 aparece en rojo.



Envía el programa parcial al micro:bit (ver página 5).

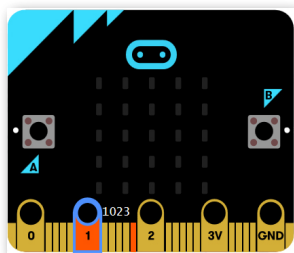
 Descargar

Presiona la tecla A en el micro:bit. Se enciende la lámpara roja.

En la simulación, haz clic en el pin1. Este se muestra como activado.

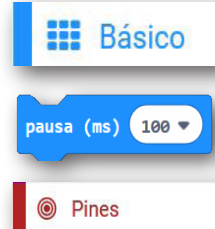
Después de 2 segundos se apaga la lámpara roja y en la simulación se apaga P 13. En su lugar se enciende a lámpara verde o el pin 12.

Si presionas el pulsador para peatones en el mástil del semáforo se apaga la lámpara roja y se enciende la verde.



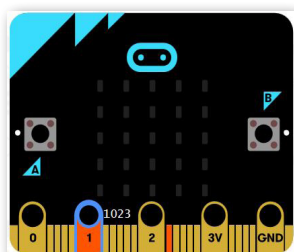
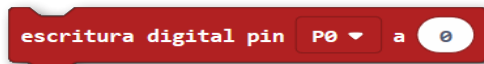
Más adelante, en la tarea se pide que después de 5 segundos la lámpara verde se debe apagar, y la roja se debe volver a encender.

Inserta desde "Básico" el comando "pausa (ms) xxxx" seguido por 2 comandos de "Pines" - "escritura digital pin ...".



Modifica el tiempo de pausa a 5000, que equivale a 5 segundos.

Cambia el pin de "P12" a "P13". Cambia "P13" a "1" y "P12" a "0".



Aquí también, realiza primero la prueba virtual. Después de un tiempo de pausa de 5 segundos se apaga el pin12 y se vuelve a encender el pin13.

"Bueno, ya has dado otro gran paso. Tu programa está listo y lo puedes probar".

Envía el programa completo al micro:bit e inícialo en el modelo. En este caso también, la lámpara verde pasa a rojo después de 5 segundos.



Semáforo para peatones con luz intermitente

Pasemos ahora a la segunda parte de la tarea.

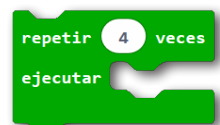
semáforo2.hex



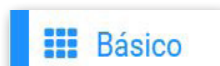
Después de 5 segundos, la lámpara verde debe parpadear tres veces. La frecuencia de parpadeo es de 1 segundo. Después el semáforo vuelve a rojo.



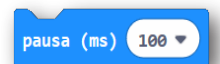
Aquí utilizas el comando "repetir x veces" del bloque "Bucles". Insértalo después de una pausa de 5 segundos.



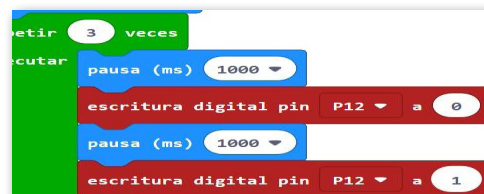
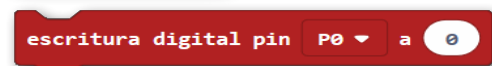
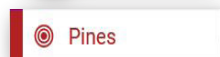
Cambia la repetición de "4" a "3".



Inserta una pausa de 1 segundo. A continuación sigue el comando "escritura digital pin P12 a 0".

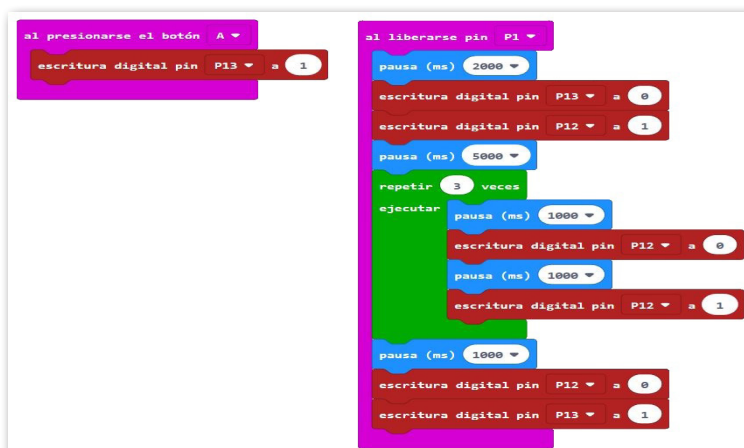


A eso le siguen otro comando de pausa de 1 segundo y el comando para encender el P12.



Después del comando de bucle, inserta otro comando de pausa de 1 segundo.

Con esto hemos programado la segunda parte de las tareas. Guarda el archivo en tu ordenador. Como de costumbre, realiza una prueba virtual y una segunda prueba en tu modelo.



Secador de manos

Seguro que en el baño de tu casa no tienes algo así. Ahí tienes una toalla grande colgada de un gancho. Pero en los baños públicos, en los de tu escuela o el de un restaurante por lo general hay unos ventiladores eléctricos en la pared que te secan las manos con aire caliente.

Un buen invento, sobre todo cuando encuentras uno de esos aparatos modernos en los que no debes presionar ningún botón para encenderlo. Simplemente pones tus manos debajo y arranca.



Ahora puedes seguir las instrucciones para construir un secador de mano con encendido sin contacto y cablearlo según el diagrama eléctrico.

Para esta tarea hazte las preguntas:



- ¿Cómo se enciende el secador de manos?
- ¿Se enciende algún tipo de indicador de control?
- ¿Cuándo se vuelve a apagar el secador de manos?
- ¿Cuánto dura el tiempo de secado?

Anótate las respuestas en una hoja de papel y luego utiliza los resultados para los comandos de programación individuales.

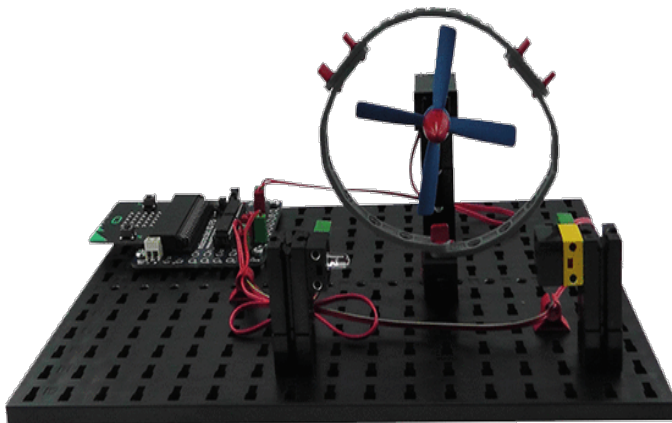


Que disfruten la resolución de esta tarea.

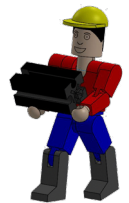
Tu RoBo

Secador de manos con barrera de luz

Arma tu segundo modelo siguiendo las instrucciones.



Cablea el modelo siguiendo el diagrama de cableado.



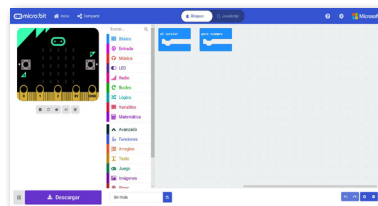
Para este modelo también te daré una tarea y una tarea adicional. Luego implementarás ambas en un programa de control.

ventilador1.hex



Programa el secador de manos de forma que cuando un usuario pone la mano en la barrera de luz se enciende el motor del ventilador. Cuando quita la mano de la barrera de luz, el motor se debe apagar.

Por supuesto que en esta tarea también te ayudaré.



Abre tu editor al igual que en la tarea anterior.

▼ Avanzado

Importante: Asegúrate nuevamente de haber activado el bloque "Avanzados" con la tecla correspondiente.



Antes de comenzar a programar te debo presentar un sensor y un actuador nuevos.



Para la barrera de luz que necesitas para el encendido y apagado del motor del ventilador necesitas el LED, que ya conoces, así como un fototransistor.

Un fototransistor es un interruptor electrónico (sensor) que reacciona a la luz. Seguramente ya te hayas preguntado por qué en los grandes almacenes la puerta de entrada se abre automáticamente, sin que acciones ningún pulsador o interruptor.

Para esto se usa una barrera de luz que consiste en una fuente de luz (emisor) y un sensor (receptor). En el kit de construcción se usa un bloque de construcción LED como sensor y un bloque de construcción de fototransistor como receptor.

El motor es un motor de corriente continua (actuador). Este convierte la energía eléctrica en energía mecánica. Esto genera una rotación del eje del motor.

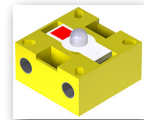
Puedes controlar el motor directamente con el micro:bit.

Bueno, has armado y cableado tu modelo, ahora solo falta programarlo.

Ya has iniciado el editor y comienzas con el bloque "al iniciar". Este ya está asignado en la pantalla, por lo que puedes insertar inmediatamente otro comando.

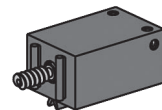
Como podrás ver en las instrucciones de construcción, el motor se conecta en los pines "pin16" y "pin15". Pin16 será el polo negativo y pin15, el polo positivo. Es por ello que al inicio el pin16 se pone en "0". El LED está conectado el pin8. Como debe brillar para siempre, pueden integrar el comando correspondiente en la parte de programa "al iniciar".

Fototransistor



Símbolo de circuito

Motor



Símbolo de circuito

 Básico

al iniciar


 Pines

escritura digital pin P0 a 0

al iniciar

escritura digital pin P16 a 0

escritura digital pin P9 a 1

 Básico



Para el resto del desarrollo del programa necesitas "para siempre". Este comando también ya está en la pantalla del programa.


Como estarás trabajando con un fototransistor debes convertir el valor que este emite de un valor digital a un valor analógico.

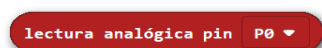
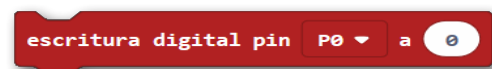
Digital Analógico

Antes de seguir trabajando te debo explicar estos dos términos: "digital" y "analógico". ¿Qué significan?

Los valores digitales solo pueden tener dos estados: 0 o 1, sí o no. Un ejemplo: una lámpara está encendida (1) o está apagada (0).

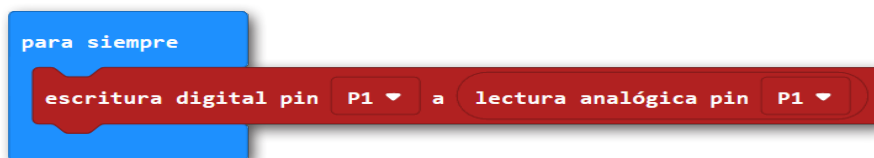
Si necesitas un valor analógico, este puede estar dentro de un rango determinado, por lo que puede tener varios valores. Por ejemplo, si quieres especificar un rango de temperatura determinado: encendido del motor de un ventilador con una temperatura exterior de 20 a 24 grados.

 Pines



Primero, inserta el comando "escritura digital pin". Cambia "P0" a "P1" ya que has conectado el fototransistor a este pin.

Dentro del bloque "Pines" encontrarás el comando "lectura analógica pin ...". Este lo debes poner en el "0" del primer comando.



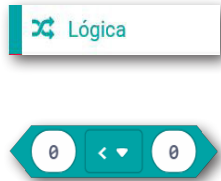
 Lógica



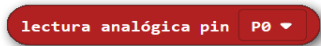
Ahora necesitas un comando que diga "Si un acontecimiento es verdadero, entonces se debe realizar la acción1. Si este no es el caso (si no) se debe realizar una acción2".

Este comando lo encuentras dentro del bloque "Lógica" y allí "si verdadero entonces --- si no". Inserta este comando en tu programa.

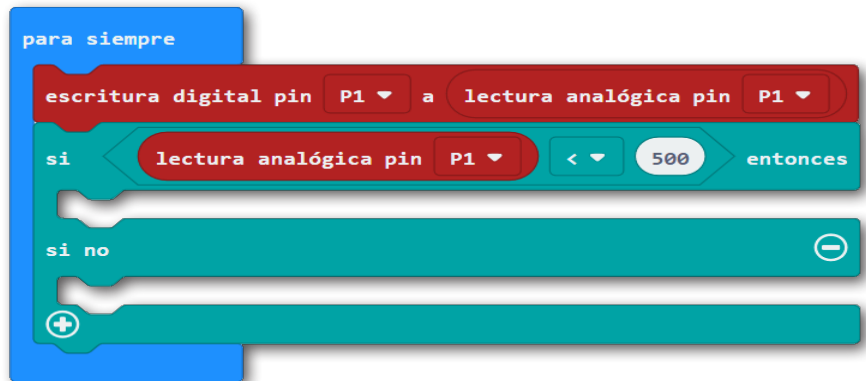
Ya he escrito el programa completo para ustedes y comprobé que la fotorresistencia conmuta con un valor analógico de 500. Este valor se debe consultar en el programa. Encuentras el comando correspondiente en el bloque "Lógica".



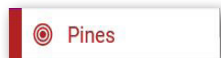
Ahora debes insertar la consulta "si" "lectura analógica pin P1" "< menor" "500" en el programa.



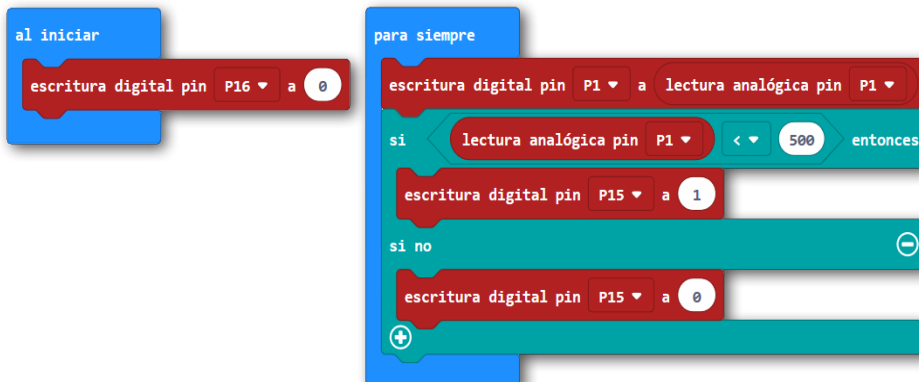
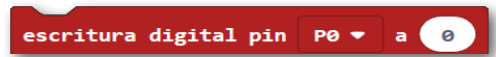
Aquí puedes duplicar el comando e insertarlo en el lugar correcto.



Luego inserta para "entonces" el comando "escritura digital pin P15". Cámbialo a "1". Duplica el comando e insértalo debajo de "si no". Cámbialo a "0".



Con esto se soluciona la tarea 1. Guarda y prueba el programa.



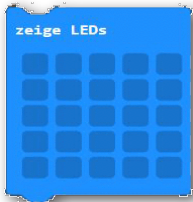
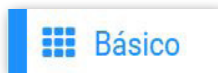
Secador de manos con indicador LED

Ahora una tarea adicional para que resuelvas.

ventilator2.hex



En tu micro:bit se encuentra una matriz LED con 25 LED. Con esta debes indicar el respectivo estado de operación del motor. Cuando el motor corre debe aparecer un "+", si no la matriz estará apagada.



Esto es bastante fácil. En el bloque "Básico" encontrarás el comando "mostrar LEDs". Arrastra este y colócalo antes del comando "si no".

Marca los LED correspondientes haciendo clic con el ratón. Estos aparecerán en blanco.

En "sino" inserta el comando "borrar la pantalla" del bloque "Básico" y "Más...".

Esto también estaría resuelto. Vuelve a guardar el programa y pruébalo con tu modelo.

```

escritura digital pin P15 a 1
mostrar LEDs
si no
  escritura digital pin P15 a 0
  borrar la pantalla
  
```

```

al iniciar
  escritura digital pin P16 a 0
  
```

```

para siempre
  escritura digital pin P0 a lectura analógica pin P0
  si < lectura analógica pin P1 > < 500 > entonces
    escritura digital pin P15 a 1
    mostrar LEDs
  si no
    escritura digital pin P15 a 0
    borrar la pantalla
  
```

Barrera

En muchas ciudades puedes encontrar aparcamientos o grandes estacionamientos que controlan en ingreso y la salida de los vehículos con una barrera. Si, por ejemplo, un aparcamiento está completamente ocupado, esto se indica

mediante un sistema de guía de aparcamiento. En algunas calles hay indicadores electrónicos que te dicen en qué aparcamiento aún quedan lugares libres o cuáles están ocupados.



Hay distintas maneras de activar una barrera.

Por ejemplo, mediante la entrada de un código numérico, una tarjeta con un código, o también con una barrera de luz, como la verás más adelante en el modelo.



Para esta tarea hazte las preguntas:

- ¿Cuándo se abre la barrera?
- ¿Se enciende algún tipo de indicador de control?
- ¿Cuándo se vuelve a cerrar la barrera?
- ¿Cuánto tiempo permanece abierta la barrera?



Anótate las respuestas en una hoja de papel y luego utiliza los resultados para los comandos de programación individuales.



Aquí también les deseo que disfruten la resolución de esta tarea.

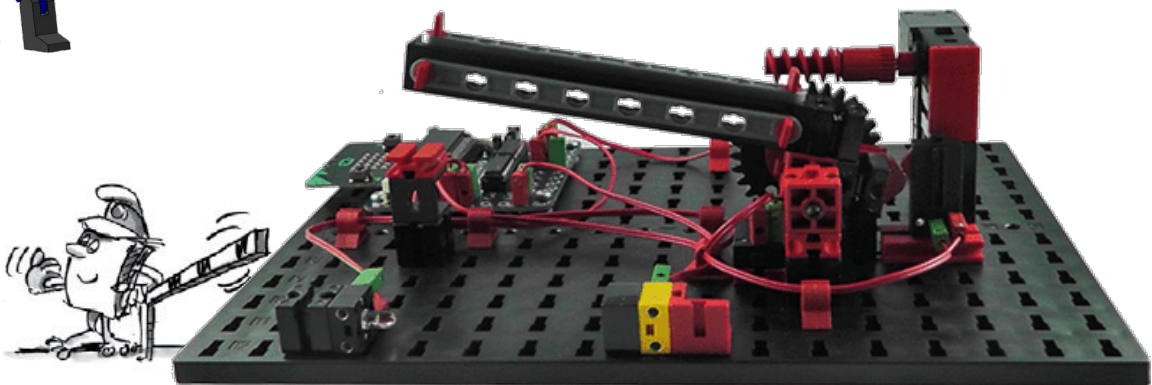
Tu RoBo

Barrera de aparcamiento



Arma tu tercer modelo siguiendo las instrucciones.

Cablea el modelo siguiendo el diagrama de cableado.



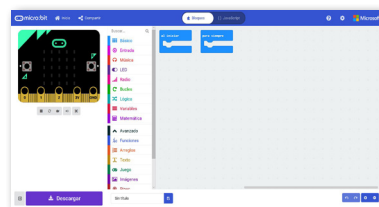
Para este modelo también te daré una tarea, y como ya eres todo un profesional de la programación, tres tareas adicionales que luego implementarás en un programa de control.

barrera1.hex



Cuando un vehículo interrumpe el haz de luz de la barrera de luz se debe abrir la barrera 1 segundo después. Debe quedar abierta 5 segundos y luego se debe volver a cerrar.

Por supuesto que en esta tarea también te ayudaré.



Inicia sesión en Internet e inicia el editor al igual que en las tareas anteriores.

▼ Avanzado



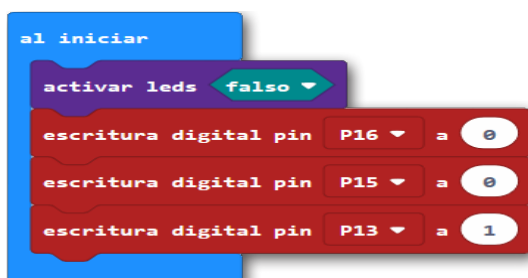
Importante: Asegúrate nuevamente de haber activado el bloque "Avanzado" con la tecla correspondiente.

De nuevo comenzarás con la parte del programa "al iniciar".



Para que el programa luego funcione correctamente, primero debes insertar el comando "activar LED" del bloque "LED" - "Más...".

Al comienzo del programa se deben colocar las conexiones de motor "P16" y "P15" en "0".



En el "pin13" has conectado el LED de tu barrera de luz. Este se debe encender después del inicio del programa. Inserta los comandos y modifica sus propiedades.



La parte principal del programa comienza de nuevo con el comando "para siempre".

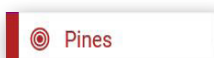
Vuelve a mirar el programa del secador de manos. En este programa consultas la barrera de luz y determinas el valor del fototransistor. Esta secuencia de comandos también la puedes usar en la barrera.

Arrastra los comandos correspondientes a tu programa, al igual que en el caso del secador de manos. El fototransistor también se conecta en el "pin1" en la barrera. Cambia el valor correspondientemente.

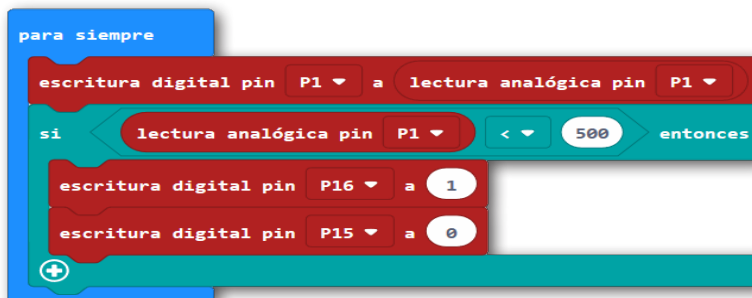
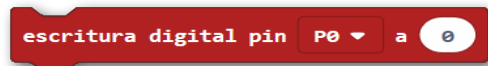


El siguiente comando es la consulta "si ... entonces" del bloque "Lógica". Arrastra el comando de consulta sobre el rombo "verdadero". Cambia el valor que figura después del "<" a 500.

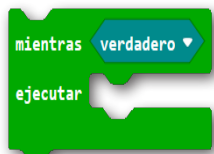




Si el valor de la barrera de luz es menor a 500, se debe abrir. Para esto, inserta para "entonces" 2 veces "escritura digital pin". Cambia el primer comando a "P16" y a "1" y el segundo comando a "P15" y "0".



Si se inicia el programa se abre la barrera o arranca el motor. Este debe funcionar hasta que se cierra el interruptor del "pin3".



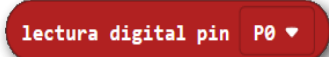
Para esto necesitas el comando "mientras ... ejecutar" del bloque "Bucles". Arrastra este comando debajo del último comando.



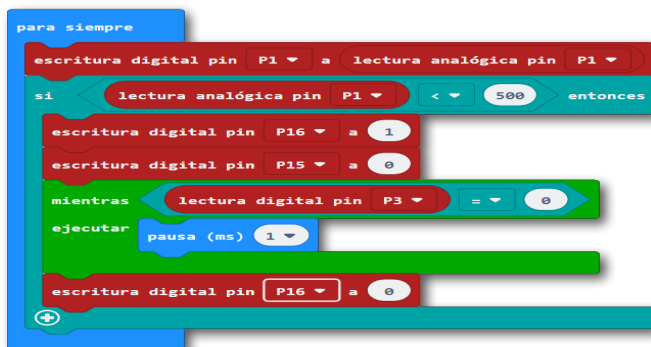
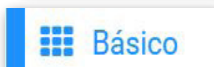
Vuelve a reemplazar el rombo "verdadero" con una consulta del bloque "Lógica".



Como primera variable, inserta el comando "lectura digital ...". Cambia la denominación del pin a "P3".



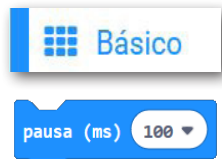
En el lugar vacío del programa, inserta desde el bloque "Básico" el comando "pausa (ms)" con el valor 100.



Prueba el programa parcial. Si interrumpes la barrera de luz se abre la barrera, activa el interruptor y se detiene.

En la tarea dice que cuando se interrumpe la barrera de luz se debe abrir la barrera después de 1 segundo. Inserta este comando en el lugar correcto y cambia el tiempo de espera a 1 segundo.

En la tarea también se pide que la barrera se cierre después de 5 segundos. Para esto, vuelve a incorporar el comando "pausa (ms)". A continuación, el motor debe girar en sentido contrario hasta que se accione el interruptor del "pin0".



```

pausa (ms) 5000
escritura digital pin P16 a 0
escritura digital pin P15 a 1
mientras lectura digital pin P0 = 0
ejecutar pausa (ms) 1
escritura digital pin P15 a 0
    
```

Simply duplicate the necessary commands from the first part of the program.

Change the value of "P16" to "0" and the value of "P15" to "1". Change also the value of the interrupter of "P3" to "P0".

For the barrier to stop when it is closed you must change the last command from "P16" to "P15".

```

para siempre
  escritura digital pin P1 a lectura analógica pin P1
  si lectura analógica pin P1 < 500 entonces
    pausa (ms) 1000
    escritura digital pin P16 a 1
    escritura digital pin P15 a 0
    mientras lectura digital pin P3 = 0
    ejecutar pausa (ms) 1
    escritura digital pin P16 a 0
    pausa (ms) 5000
    escritura digital pin P16 a 0
    escritura digital pin P15 a 1
    mientras lectura digital pin P0 = 0
    ejecutar pausa (ms) 1
    escritura digital pin P15 a 0
    
```



Prueba el programa completo. Si interrumpes la barrera de luz se abre la barrera, activa el interruptor y se detiene. Después de un tiempo de espera determinado se vuelve a cerrar la barrera.

¿Qué sucede si la barrera está abierta al iniciar el programa? En realidad se la debería cerrar manualmente. ¿Pero por qué no podemos hacer eso con un complemento de programa?

barrera2.hex



Si al inicio del programa la barrera está abierta, primero se la debe cerrar.

Duplica el bloque de programa de la parte de programa "para siempre" en el programa de inicio.

```

escritura digital pin P16 a 0
escritura digital pin P15 a 0
mientras lectura digital pin P0 = 0
ejecutar pausa (ms) 100

```

Ahora puedes probar el programa. Lleva la barrera mecánicamente a una posición

media. Después de descargar el programa, primero se cierra la barrera y después procesa la parte del programa "para siempre".



```

al iniciar
  activar leds falso
  escritura digital pin P13 a 1
  escritura digital pin P16 a 0
  escritura digital pin P15 a 1
  mientras lectura digital pin P0 = 0
  ejecutar pausa (ms) 100
  escritura digital pin P15 a 0

```

El programa completo ahora se ve así.

Guarda el programa en tu ordenador. Usa un nombre nuevo, por ejemplo, barrera2.



Tengo otra idea más. ¿Qué inconveniente habría si instalas un interruptor de servicio (pulsador A), con cuya ayuda se enciende el sistema?



Inserta una parte de programa con la que enciendes el sistema.

barrera3.hex

Los comandos correspondientes los debes insertar en la parte de programa "al iniciar". Inserta antes de la activación LED el comando "mientras ... ejecutar" del bloque "Bucles".



Reemplaza el rombo "verdadero" con el comando "no" del bloque "Lógica".



Para la consulta del pulsador A usas el comando "botón A presionado" del bloque "Entrada".



El lugar para "ejecutar" lo completas con un duplicado de "pausa (ms) 100".



```

al iniciar
mientras no botón A presionado
ejecutar
    pausa (ms) 100
activar leds falso
escritura digital pin P13 a 1
escritura digital pin P16 a 0
escritura digital pin P15 a 1
mientras lectura digital pin P0 = 0
ejecutar
    pausa (ms) 100
escritura digital pin P15 a 0
    
```

Ahora puedes probar el programa. Lleva la barrera a una posición media. Después de haber descargado el programa, espera al pulsador A. Si se acciona, primero se cierra la barrera y después procesa la parte del programa "para siempre".

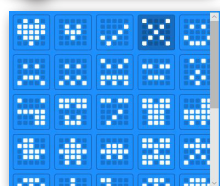
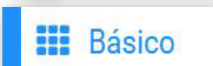
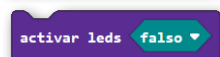


Guarda el programa en tu ordenador. Usa un nombre nuevo, por ejemplo, barrera3.

Barrera de aparcamiento con indicador óptico

Aquí tengo otra ampliación al programa para ti.

barrera4.hex



La posición de la barrera se debe indicar ópticamente en el campo LED. Barrera cerrada: cruz; barrera abierta: gancho.



Primero amplias el "Bloque de inicio" con el comando "activar LED ..." desde el bloque "LED".

Inserta el comando por encima del comando "mientras ... ejecutar" y modifica el valor de "falso" a "verdadero".

A continuación, inserta el comando "mostrar ícono" desde el bloque "Básico". Haz clic en la flecha y selecciona la visualización de una cruz. Esta secuencia de comandos sirve para conmutar la barrera a su posición cerrada al inicio.

```

activar leds verdadero
mostrar ícono
mientras lectura digital pin P0 = 0
    
```

```

escritura digital pin P9 a 0
mostrar ícono
pausa (ms) 5000
escritura digital pin P16 a 0
escritura digital pin P15 a 1
mostrar ícono
    
```

En la parte de programa "para siempre" necesitas el comando "mostrar ícono" en dos lugares. Una vez cuando la barrera está abierta y otra vez más cuando está cerrada. Inserta el comando y modifica el indicador LED.

Básico

```

mostrar ícono
    
```

Prueba el programa con la ampliación. Lleva la barrera a una posición media. Después de haberse descargado el programa, este espera al pulsador A. Si se acciona, aparece la representación gráfica (X) y se cierra la barrera. Después se procesa la parte del programa "para siempre". Si se interrumpe el interruptor de luz, la barrera se abre y la imagen LED cambia por 5 segundos a un gancho. Cuando se vuelve a cerrar la barrera también cambia la imagen del indicador LED.



Guarda el programa en tu ordenador. Usa un nombre nuevo, por ejemplo, barrera4.

```

al iniciar
  mientras no botón A presionado
  ejecutar pausa (ms) 100
  activar leds falso
  escritura digital pin P13 a 1
  escritura digital pin P16 a 0
  escritura digital pin P15 a 1
  activar leds verdadero
  mostrar ícono
  mientras lectura digital pin P0 = 0
  ejecutar pausa (ms) 100
  escritura digital pin P15 a 0

para siempre
  escritura digital pin P1 a lectura analógica pin P1
  si lectura analógica pin P1 < 500 entonces
    escritura digital pin P16 a 1
    escritura digital pin P15 a 0
    mientras lectura digital pin P3 = 0
    ejecutar pausa (ms) 100
    escritura digital pin P16 a 0
    escritura digital pin P15 a 0
    mostrar ícono
    pausa (ms) 5000
    escritura digital pin P16 a 0
    escritura digital pin P15 a 1
    mientras lectura digital pin P0 = 0
    ejecutar mostrar ícono
    pausa (ms) 100
    escritura digital pin P15 a 0
    
```

Si algo no funciona...

... espero que en esta tabla encuentres una solución a tu problema.

Problema	Posible causa	Solución
1. El software makecode no se puede conectar al micro:bit	Cable USB no conectado	Conectar cable USB
2. El pulsador no funciona	Enchufes eléctricos conectados en las conexiones incorrectas del pulsador o del micro:bit	Usar las conexiones 1 y 3 en el pulsador. En el micro:bit, conectar el enchufe en las tomas para I5, I4 o I1.
3. El fototransistor no funciona	Enchufe eléctrico mal conectado	En el fototransistor: Conectar el enchufe rojo del lado del punto rojo y el enchufe verde del lado sin marcar.
	El LED de la barrera de luz no brilla	Conectar el LED a I1 y P1, prestar atención a la polaridad
	La luz del LED pasa por al lado del fototransistor	Mover el LED para que ilumine el fototransistor
4. El motor no gira	Motor no conectado al micro:bit	Conectar el motor al micro:bit tal como se describe en el diagrama de cableado del modelo correspondiente
	Motor conectado a la salida de motor incorrecta del micro:bit	Verificar con el diagrama de cableado a que salida M1 debe estar conectado el motor y conectarlo a esa salida
5. El motor gira en el sentido incorrecto	Los enchufes eléctricos rojo y verde están intercambiados	Intercambiar los enchufes rojo y verde en el motor
		Modificar el sentido de giro del motor en el programa de control pin15/pin16
8. Problema no descrito	No encontrado	Ponte en contacto con fischertechnik, por ejemplo a: www.fischertechnik.de

Para finalizar, una dirección de Internet importante. Si deseas saber más sobre el micro:bit entra a

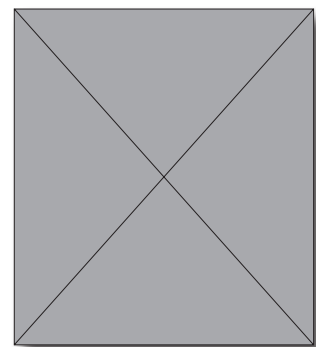
<https://microbit.org>



Bueno, eso es todo. Te deseo mucho éxito con la programación de los modelos ft con makecode.

1era edición 2019

fischertechnik GmbH Tel: (+49) 7443 12 - 4369
Klaus-Fischer-Strasse 1 E-mail: info@fischertechnik.de
72178 Waldachtal



Hermann Weininger,
Docente y maestro
electricista