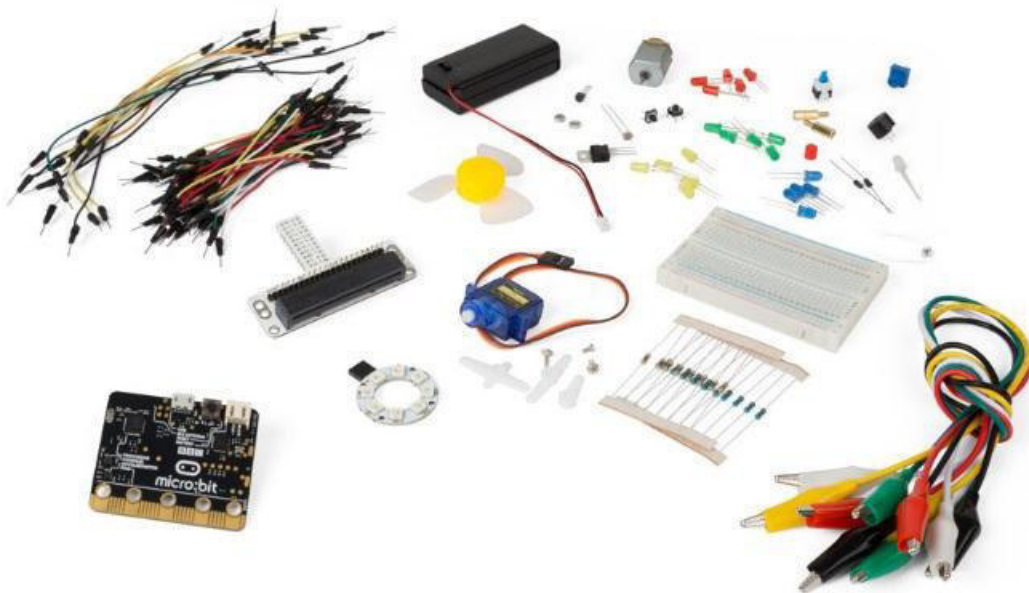


## VMM501

---

### MOTOR SHIELD FOR MICRO:BIT



USER MANUAL	2
HANDLEIDING	27
MODE D'EMPLOI	53
MANUAL DEL USUARIO	79
BEDIENUNGSANLEITUNG	107
INSTRUKCJA OBSŁUGI	136
MANUAL DO UTILIZADOR	162



# USER MANUAL

## 1. Introduction

To all residents of the European Union

### Important environmental information about this product



This symbol on the device or the package indicates that disposal of the device after its lifecycle could harm the environment. Do not dispose of the unit (or batteries) as unsorted municipal waste; it should be taken to a specialized company for recycling. This device should be returned to your distributor or to a local recycling service. Respect the local environmental rules.

**If in doubt, contact your local waste disposal authorities.**

Thank you for choosing Velleman®! Please read the manual thoroughly before bringing this device into service. If the device was damaged in transit, do not install or use it and contact your dealer.

## 2. Safety Instructions



This device can be used by children aged from 8 years and above, and persons with reduced physical, sensory or mental capabilities or lack of experience and knowledge if they have been given supervision or instruction concerning the use of the device in a safe way and understand the hazards involved. Children shall not play with the device. Cleaning and user maintenance shall not be made by children without supervision.



Indoor use only.

Keep away from rain, moisture, splashing and dripping liquids.

## 3. General Guidelines



- Refer to the Velleman® Service and Quality Warranty on the last pages of this manual.
- Familiarise yourself with the functions of the device before actually using it.
- All modifications of the device are forbidden for safety reasons. Damage caused by user modifications to the device is not covered by the warranty.
- Only use the device for its intended purpose. Using the device in an unauthorised way will void the warranty.
- Damage caused by disregard of certain guidelines in this manual is not covered by the warranty and the dealer will not accept responsibility for any ensuing defects or problems.
- Nor Velleman nv nor its dealers can be held responsible for any damage (extraordinary, incidental or indirect) – of any nature (financial, physical...) arising from the possession, use or failure of this product.
- Due to constant product improvements, the actual product appearance might differ from the shown images.
- Product images are for illustrative purposes only.
- Do not switch the device on immediately after it has been exposed to changes in temperature. Protect the device against damage by leaving it switched off until it has reached room temperature.
- Keep this manual for future reference.

## 4. Description

This starter kit is an educational kit based on micro:bit. It comes with basic electronic components, a breadboard, connection wires and a micro:bit.

## 5. Examples

### 5.1 LED

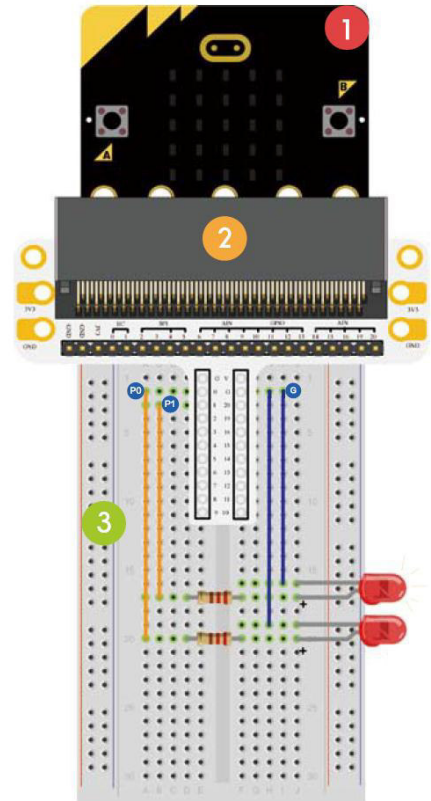
- 1 1x micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1x breadboard
- 4 2x red LED (polarisation: anode (+) = long leg, cathode (-) = short leg)
- 5 2x 100  $\Omega$  resistor (brown/black/brown/gold)



4



5



In this course, we are going to use micro:bit to make two LEDs twinkle alternatively.

Place the necessary components on the breadboard as shown.

The programme is written in code blocks, online in the web browser. Open the website [www.makecode.com](http://www.makecode.com) or <https://www.microsoft.com/en-us/makecode?rtc=1>, click the micro:bit icon and click **Start Project**.

Microsoft Office Windows Surface Xbox Deals Support More

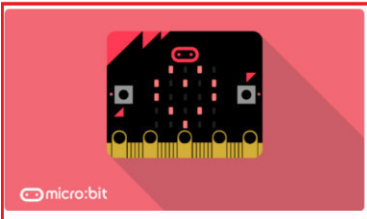
MakeCode About Get started Resources

Search Microsoft.com Sign in

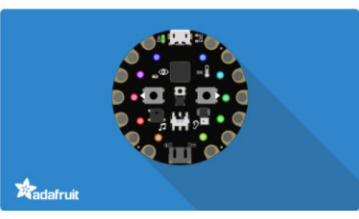
### Hands on computing education

Microsoft MakeCode brings computer science to life for all students with fun projects, immediate results, and both block and text editors for learners at different levels.


*Klik dit icoon aan*



micro:bit  
Start coding with micro:bit >



Circuit Playground Express  
Start coding with Circuit Playground Express >



Minecraft  
Start coding with Minecraft >

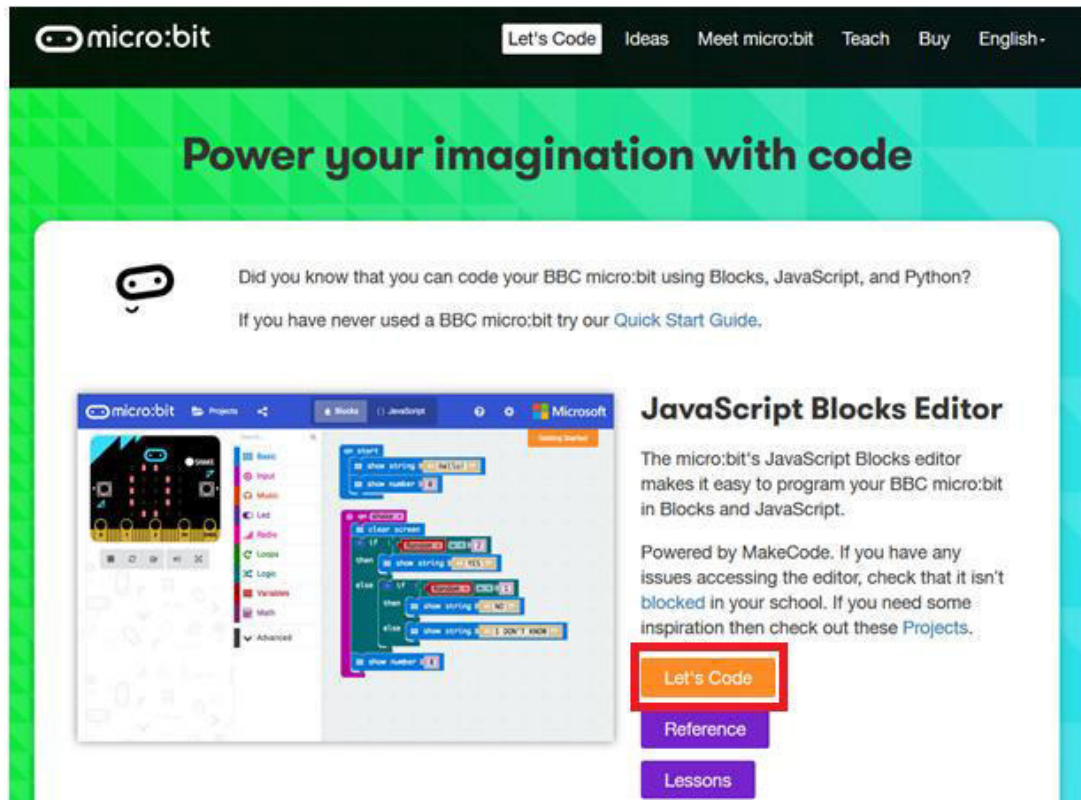
The micro:bit code block opens in a new window. Now, we can start composing the code with code blocks, which we drag and drop from a code drawer to a code editor.

Read below how this works...

## What is MakeCode?

Formerly PXT – Programming eXperience Toolkit Editor to write code for the micro:bit.

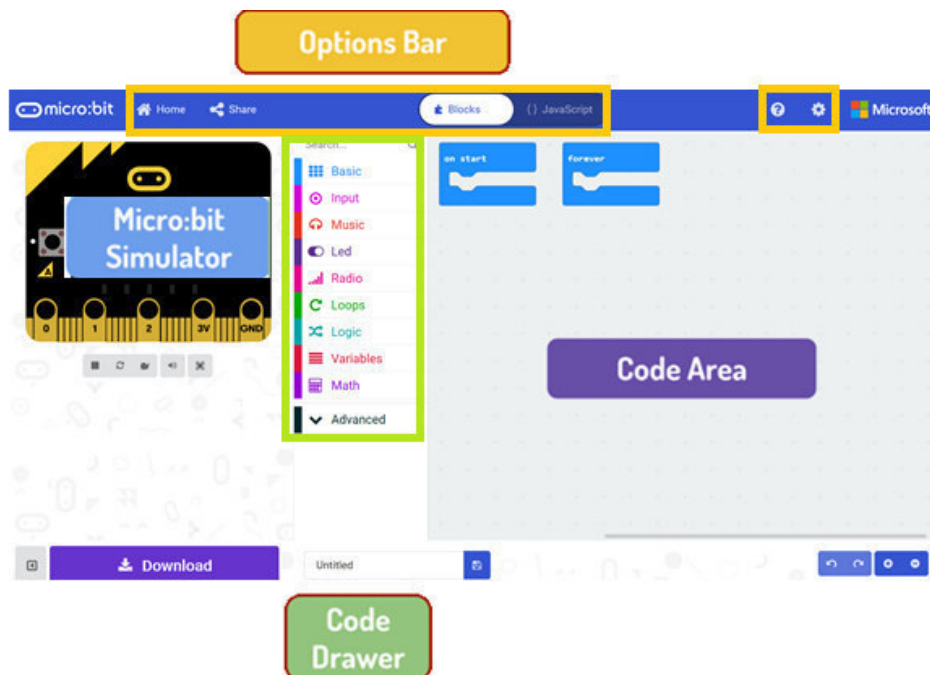
A graphical, beginner-friendly drag-and-drop code editor similar to Scratch. It works online, in the browser.



You can see the editor is made up of different sections. You make your code in the **Code Area**, dragging out blocks from the **Code Drawer**. You can immediately see your code results in the **micro:bit simulator**. At the bottom is where you download and save your project.



Next, we will take a closer look at how to use the **Options Bar** for doing other things.



## Options Bar



Click to create or add a new project.

Import your projects here.

Open up the Blocks interface to write your scripts with the Block editor.

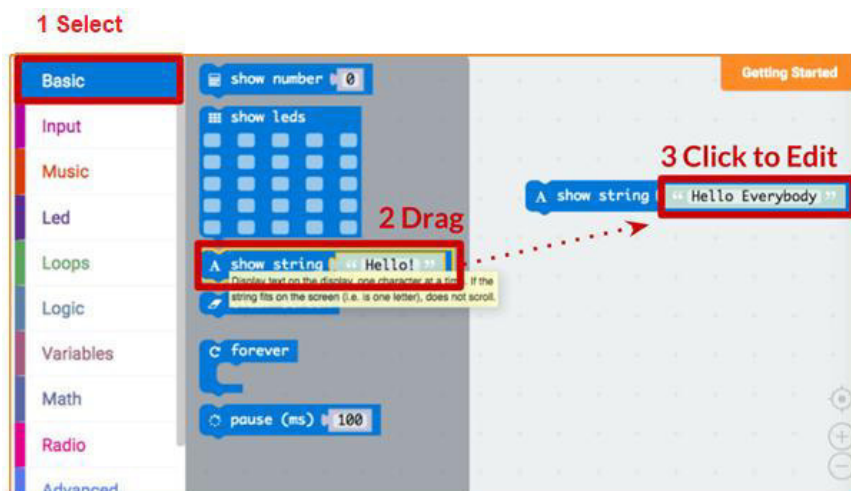
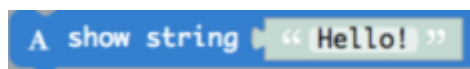
Open up the Javascript interface to type out your script in the Javascript language.

Handy reference when you are unsure about how the various blocks and functions work.

Shortcuts to changing project properties. Rename and delete your projects here. **Reset** deletes all the projects that you have saved, so be very careful. Most of the time, you will just want to stick to **Delete Project**.

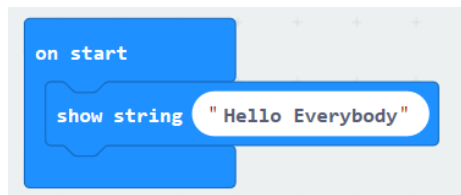
## Using the LED Bar

We start by writing some code! Each time you select from the Code Drawer, it will list all the codes available to you. Drag out the **show string** block and click inside the box to edit.



1	Select
2	Drag

3	Click to edit
---	---------------

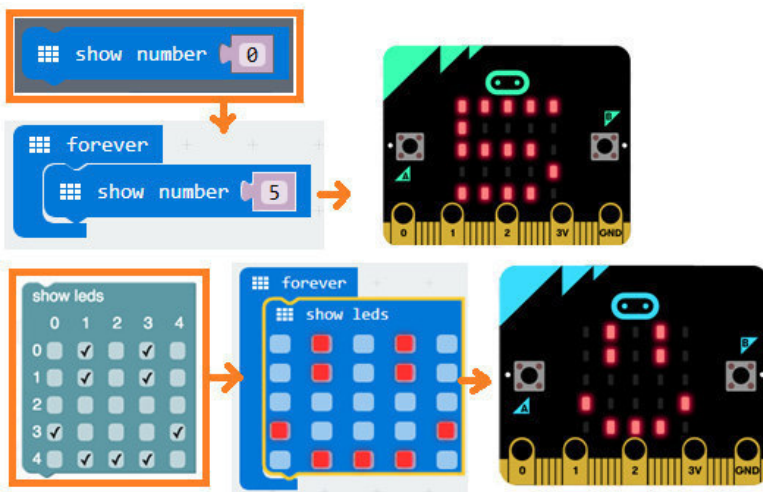


See what happens in the micro:bit simulator!

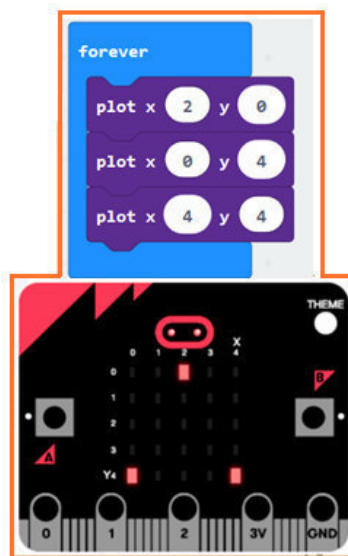
### Say Hello!

The text you have been displaying are called **strings**.

You can show any integer in the LED screen using **show number** or any 5x5 pixel image using **show leds**.



You can also plot one point LED at the time using co-ordinates. Co-ordinate (0,0) is the top left corner.

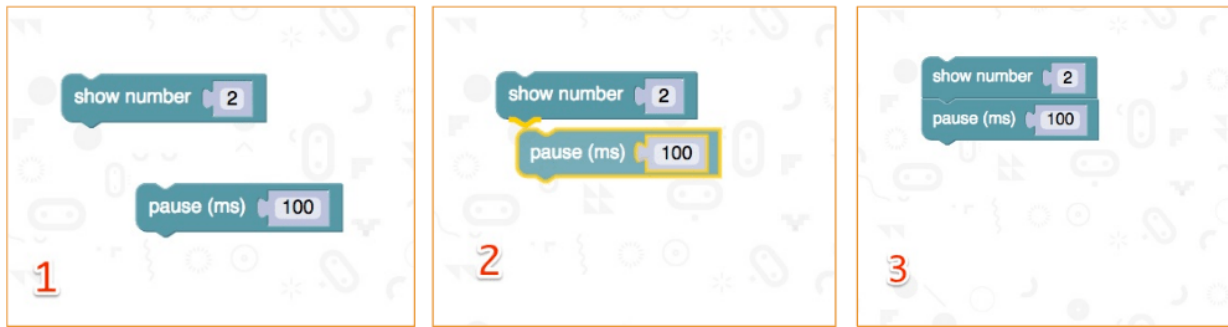


### Joining Blocks

Click and hold the block you would like to join. Drag the block to the target block until a side of the target block is highlighted. Release and the two blocks are now joined!

Clicking on the first block will move the second; clicking on the second will detach it from the first.





### Testing on micro:bit

Connect the micro:bit to your computer using a micro-USB cable.

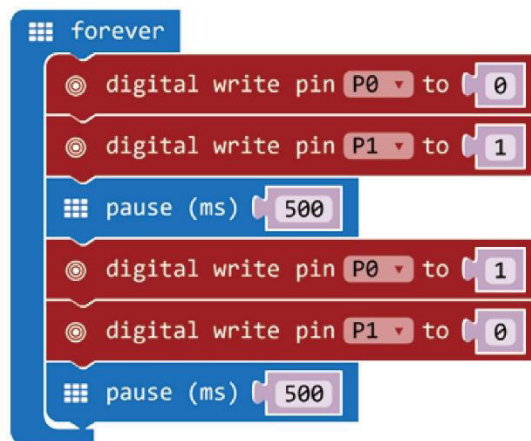
It will go through a default sequence, asking you to press buttons and play a game. Try it out!

Click **Download** to translate your script to a **hex file** and to download it.

If you set Chrome up properly, you will not need this next step. Drag the downloaded hex file to the micro:bit drive, or right-click and **Send To** in Windows®.

You have just programmed your very own device!

Have a look at the code below.

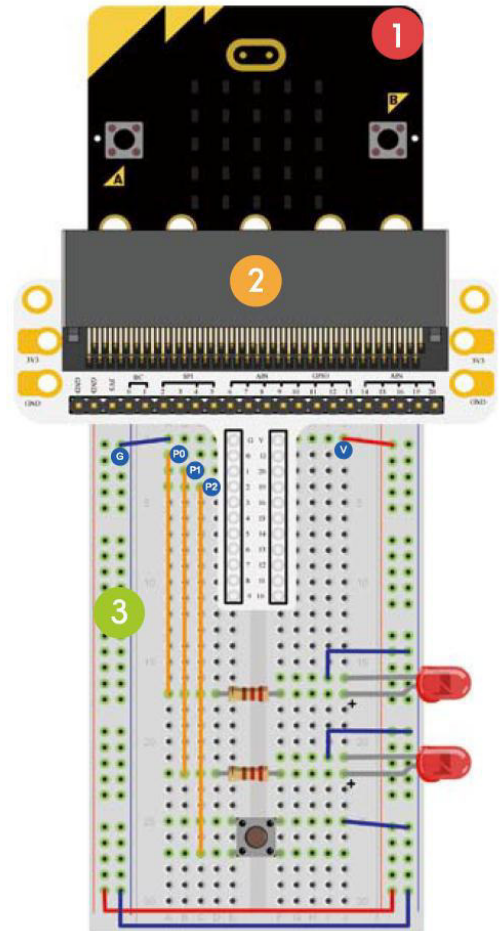
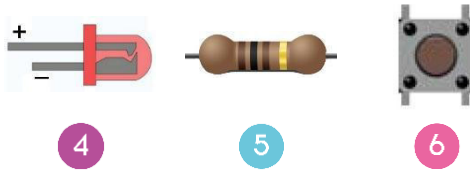


1.	Drag and drop the code blocks to form the code as shown. You can find the <b>digital write pin P0 to 0</b> block in <b>Advanced</b> → <b>Pins</b> in the Code Drawer. You can find the <b>forever</b> and <b>pause (ms) 100</b> in <b>Basic</b> in the Code Drawer.
2.	Set value for P0 to 0. LED0 off = low voltage = 0 V = digital 0. Set value for P1 to 1. LED1 on = high voltage = 5 V = digital 1.
3.	Set the <b>pause</b> to 500 ms.
4.	Set value for P0 to 1. LED0 on = high voltage = 5 V = digital 1. Set value for P1 to 0. LED1 off = low voltage = 0 V = digital 0.
5.	Set the <b>pause</b> to 500 ms.
6.	When complete, we compile the programme and generate a hex. file. Click on the download button and save the hex. file to the Downloads folder → C:\downloads. This hex. file is ready to upload to the micro:bit. Plug the micro:bit into a USB port. The, drag and drop the hex. file onto the micro:bit removable device to upload the programme.

You will see the two LEDs flash alternatively. Now, why not make an RGB traffic light?

## 5.2 Button

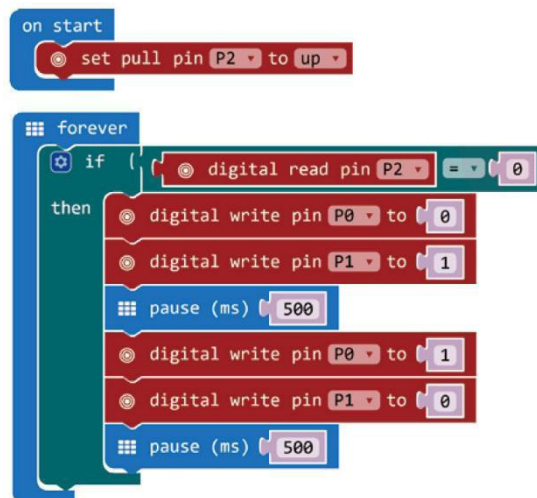
- 1 1x micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1x breadboard
- 4 2x red LED (polarisation: anode (+) = long leg, cathode (-) = short leg)
- 5 2x 100  $\Omega$  resistor (brown/black/brown/gold)
- 6 1x momentary push button



We use a button to control the LED flash. Press the button to make the LED flash in turns. Release to switch off the LED.

Place the necessary components on the breadboard as shown.

Have a look at the code below.



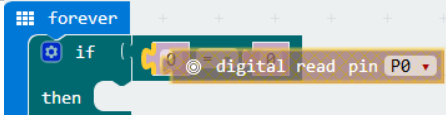
1. Drag and drop the code blocks to form the code as shown. You can find the **digital read pin P0** and **digital write pin P0 to 0** blocks in **Advanced** → **Pins** in the Code Drawer. You can find the **forever** and **pause (ms) 100** in **Basic** in the Code Drawer. You can find the **if then** and **logic equal function** blocks in **Logic** in the Code Drawer. Drag the **logic equal function** block and drop it over the **true** block.



The two blocks will snap together.

Drag the **digital read pin P0** block and drop it over the **0** of the **logic equal function** block as shown.



	
2.	Set P2 to be a pull-up.
3.	Set the <b>digital read pin</b> to P2.
4.	Set the <b>digital write pin P0 to 0</b> (active low (0 V)). Set the <b>digital write pin P1 to 1</b> (active high (5 V)).
5.	Set the <b>pause</b> to 500 ms.
6.	Set the <b>digital write pin P0 to 1</b> (active high (5 V)). Set the <b>digital write pin P1 to 0</b> (active low (0 V)).
7.	Set the <b>pause</b> to 500 ms.
8.	When complete, we compile the programme and generate a hex. file. Click on the download button and save the hex. file to the Downloads folder → C:\downloads. This hex. file is ready to upload to the micro:bit. Plug the micro:bit into a USB port. The, drag and drop the hex. file onto the micro:bit removable device to upload the programme.

Press the button and you will see the LED flash alternatively. Now, how to light the red LED with the button pressed and light the green LED with the button released?

### 5.3 Trimpot

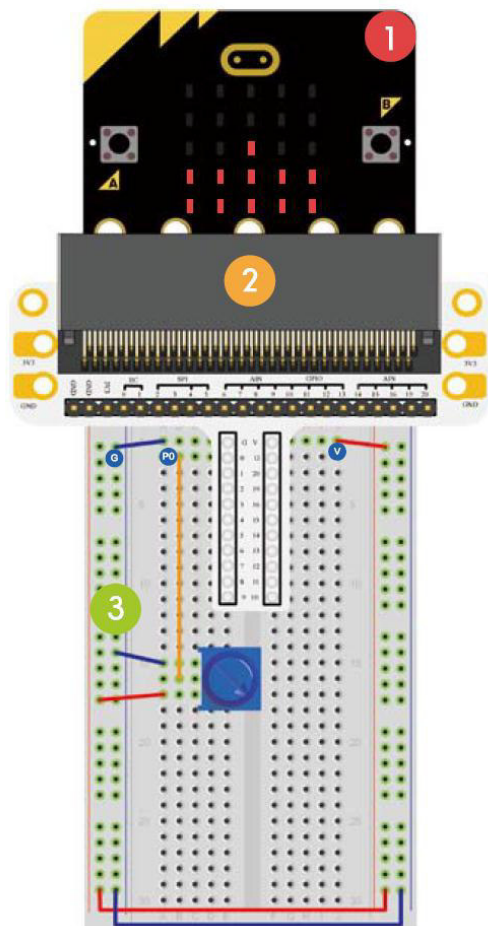
- 1 1x micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1x breadboard
- 4 1x 10 kΩ trimmer



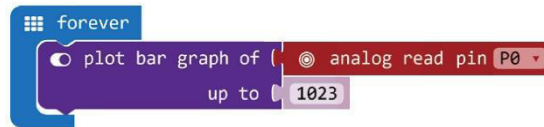
4

We are going to read the output voltage of the trimpot and display it on the micro:bit screen with a bar graph.

Place the necessary components on the breadboard as shown.



Have a look at the code below.



1.	Drag and drop the code blocks to form the code as shown. You can find the <b>plot bar graph of 0 up to 0</b> block in <b>LED</b> in the Code Drawer.
2.	Set <b>analog read pin</b> to P0. You can find this block in <b>Pins</b> in the Code Drawer. Set the value up to 1023.
3.	When complete, we compile the programme and generate a hex. file. Click on the download button and save the hex. file to the Downloads folder → C:\downloads. This hex. file is ready to upload to the micro:bit. Plug the micro:bit into a USB port. The, drag and drop the hex. file onto the micro:bit removable device to upload the programme.

Rotate the trimmer. The voltage will be displayed on the micro:bit screen through a bar graph. When the voltage is 0, the LED screen displays a pixel spot only. When it is 3.3 V, the whole screen will be illuminated. Now, how would you use the trimmer to adjust the LED's brightness?

#### 5.4 Photocell

- 1 1x micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1x breadboard
- 4 1x photocell
- 5 1x 10 kΩ resistor  
(brown/black/black/red/bruin)

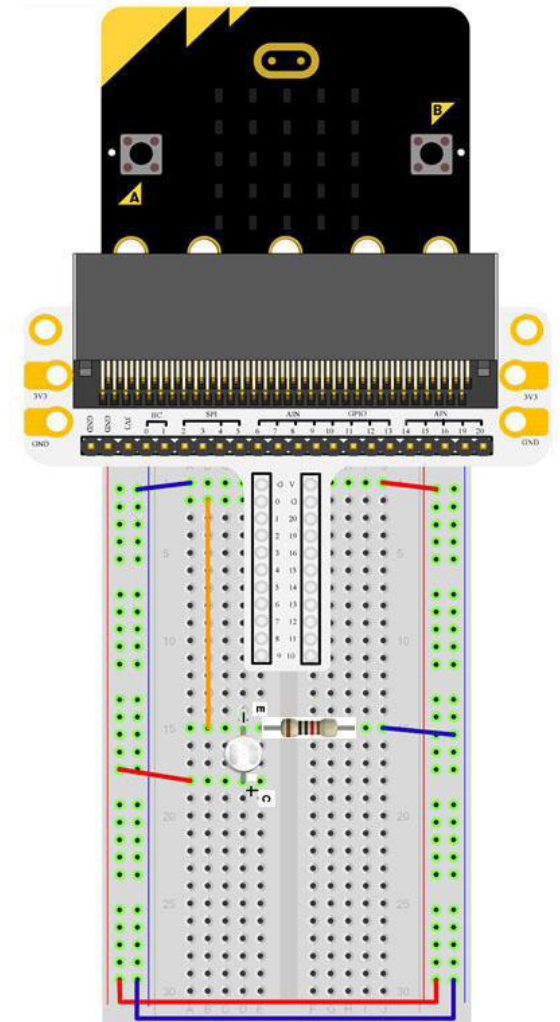


4

Short leg = collector  
Positive anode  
Long leg = emitter  
Negative cathode



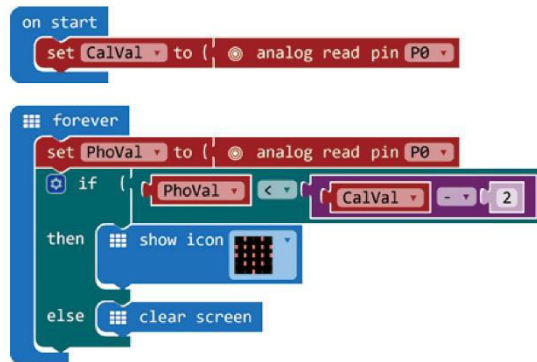
5



We are going to use a photocell to control the brightness of the micro:bit screen.

Place the necessary components on the breadboard as shown.

Have a look at the code below.

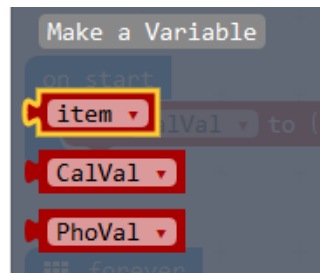


1. First, we make two variables. Go to **Variables** in the Code Drawer and click on **Make a Variable**.

New variable name:

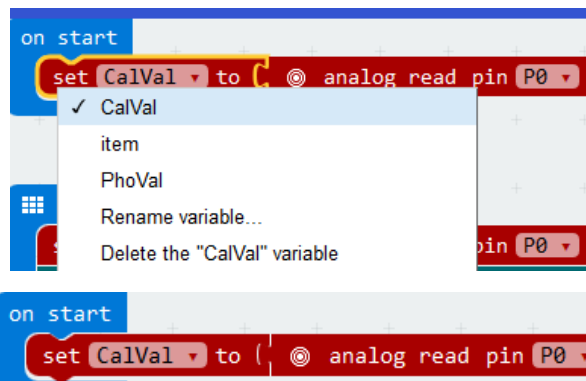
Enter **CalVal** in the window and click **Ok**. Enter **PhoVal** in the window and click **Ok**.

You will see two new variables under **Variables** in the Code Drawer.



We will need these variables later to save the data in a register.

You can find the **logic smaller than function** and the **if then else** blocks in **Logic** in the Code Drawer. You can find the **mathematical function minus** block in **Math** in the Code Drawer. You can find the **show icon** block in **Basic** in the Code Drawer. You can find the **clear screen** block in **Basic** → **More** in the Code Drawer. You can find the **set item to 0** block in **Variables** in the Code Drawer. Click on the arrow and select **CalVal** or **PhoVal**.



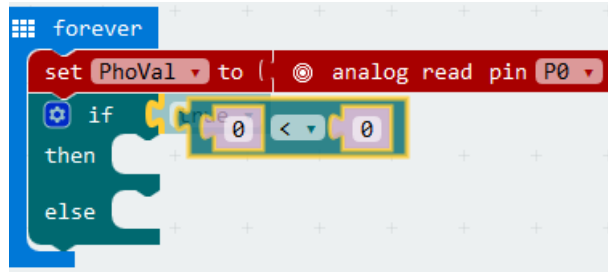
2. Select the **CalVal** variable and set the **analog read pin** to P0.



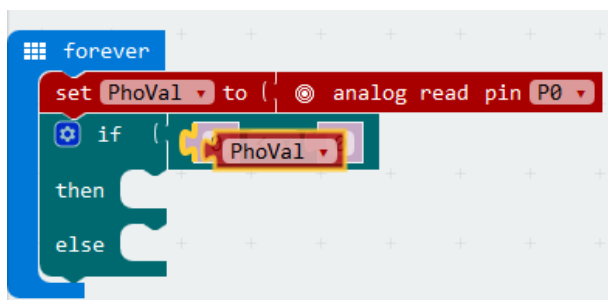
3. In the **forever** block, select the **PhoVal** variable and set the **analog read pin** to P0.



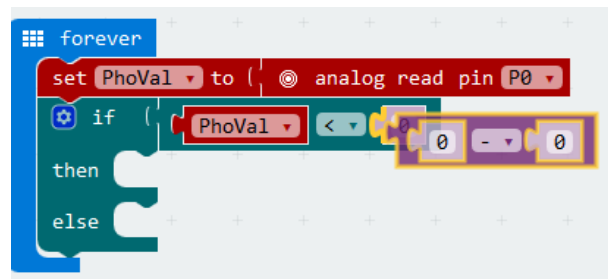
4. Drag the **logic smaller than function** next to the **if** block and drop it over the **true** block.



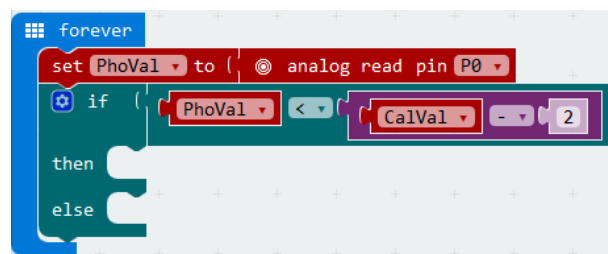
Next, we drag and drop the **PhoVal** variable (from **Variables** in the Code Drawer) and drop it over the first 0 of the **logic smaller than function**.



Drag the **mathematical function minus** block and drop it over the second 0 of the **logic smaller than function**.



Next, we drag and drop the **CalVal** variable (from **Variables** in the Code Drawer) and drop it over the first 0 of the **mathematical function minus** block. Set the second 0 of the **mathematical function minus** block to 2.



5. Drag and drop the **show icon** block next to the **then** block. Drag and drop the **clear screen** block next to the **else** block.

6. When the **PhoVal** is smaller than **CalVal** - 2, the display will show a heart. Else, it will switch off.

- 7.** When complete, we compile the programme and generate a hex. file. Click on the download button and save the hex. file to the Downloads folder → C:\downloads. This hex. file is ready to upload to the micro:bit.  
Plug the micro:bit into a USB port. Then, drag and drop the hex. file onto the micro:bit removable device to upload the programme.

Note: reset the micro:bit to calibrate the reference value according to the current brightness. To run the programme properly, we must start with the light switched on.

When the light is switched on, nothing is displayed. When the light is switched off, the heart is displayed. Now, how can we use a photocell to control an LED?

## 5.5 RGB LED

- 1 1x micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1x breadboard
- 4 1x RGB LED (common cathode)
- 5 3x 10  $\Omega$  resistor (brown/black/brown/gold)

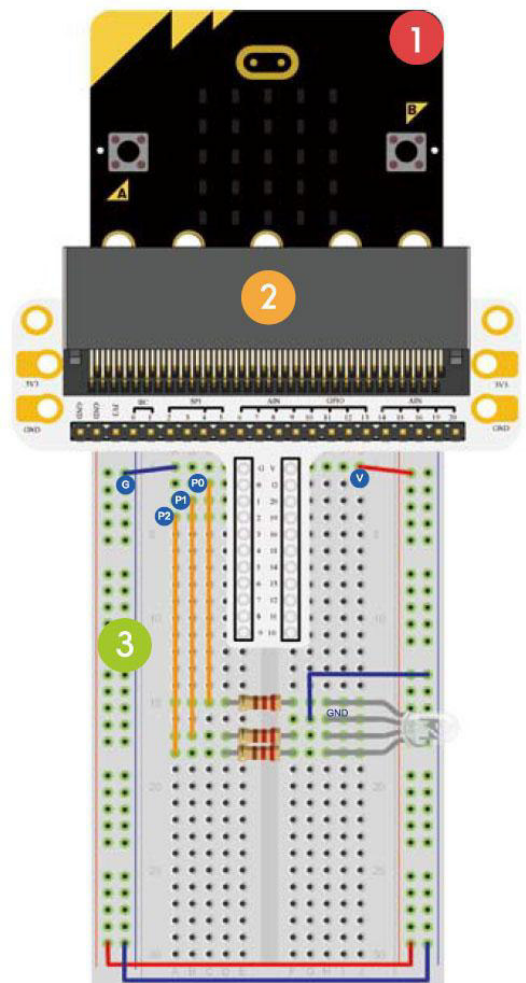


4

1. Red (anode +)
2. Ground (cathode -) – longest leg
3. Green (anode +)
4. Blue (anode +)



5



We are going to make an RGB LED gradually shift its light among red, green and blue.

Place the necessary components on the breadboard as shown.

Have a look at the code below.

```

on button A pressed
  digital write pin P0 to 1
  digital write pin P1 to 0
  digital write pin P2 to 0

on button B pressed
  digital write pin P0 to 0
  digital write pin P1 to 1
  digital write pin P2 to 0

on button A+B pressed
  digital write pin P0 to 0
  digital write pin P1 to 0
  digital write pin P2 to 1
  
```

- Drag and drop the code blocks to form the code as shown.

You can find the **on button A pressed** block in **Input** in the Code Drawer. You can find the **digital write pin P0 to 0** block in **Pins** in the Code Drawer.

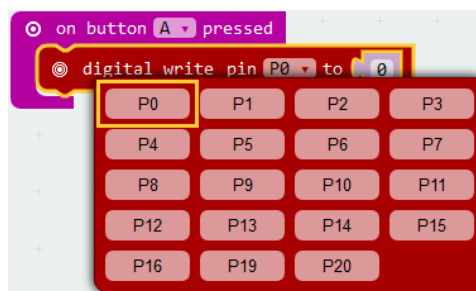
Select the A option in the **on button A pressed** block.

Drag 3 **digital write pin P0 to 0** blocks and insert them in the **on button A pressed** block.

Set pin P0 in the first **digital write pin P0 to 0** block, and set value 0 to 1 (red LED on).

Set pin P1 in the second **digital write pin P0 to 0** block, and set value to 0 (green LED off).

Set pin P2 in the third **digital write pin P0 to 0** block, and set value to 0 (blue LED off).



```

on button A pressed
  digital write pin P0 to 1
  digital write pin P1 to 0
  digital write pin P2 to 0
  
```

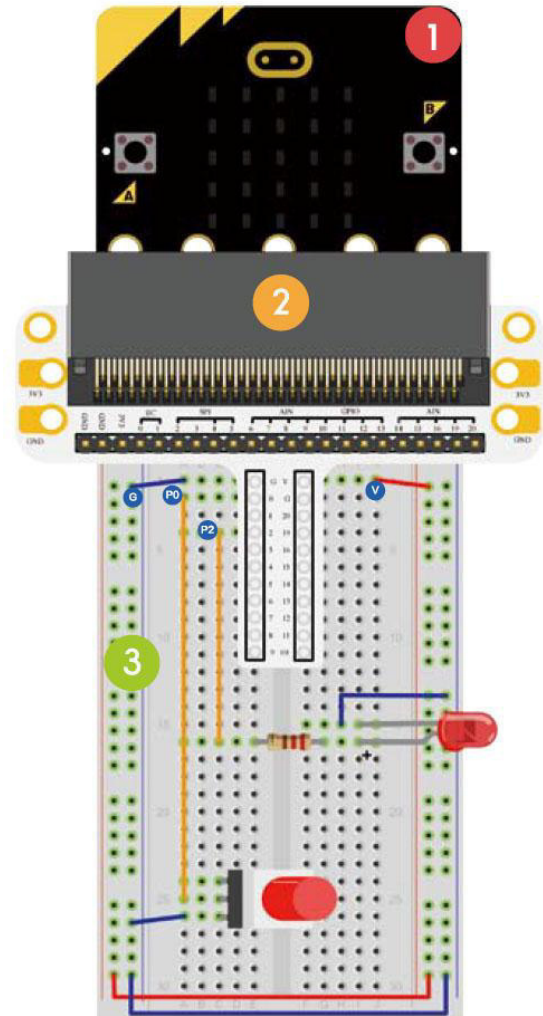
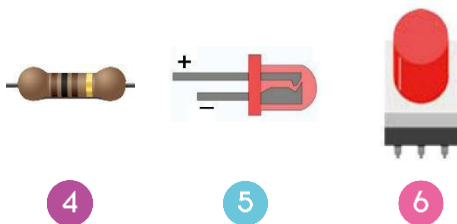


2.	Similarly, compile the two <b>on button B/A+B pressed</b> blocks to form the complete code.
3.	When complete, we compile the programme and generate a hex. file. Click on the download button and save the hex. file to the Downloads folder → C:\downloads. This hex. file is ready to upload to the micro:bit. Plug the micro:bit into a USB port. The, drag and drop the hex. file onto the micro:bit removable device to upload the programme.

Press button A to light the red LED, press button B to light the green LED, press buttons A and B simultaneously to light the blue LED. Now, how would you realize a soft gradient RGB light?

## 5.6 Self-Locking Switch

- 1 1x micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1x breadboard
- 4 1x 100  $\Omega$  resistor (brown/black/brown/gold)
- 5 1x red LED (polarisation: anode (+) = long leg, cathode (-) = short leg)
- 6 1x self-locking or bi-stable switch



We are going to read out the ambient temperature (data) of the analogue temperature sensor and display the data to the micro:bit.

Place the necessary components on the breadboard as shown.

Have a look at the code below.

```

on start
  set pin P0 to emit edge events
  set pull pin P0 to up

on event from MICROBIT_ID_IO_P0 with value MICROBIT_PIN_EVT_FALL
  digital write pin P2 to 1

on event from MICROBIT_ID_IO_P0 with value MICROBIT_PIN_EVT_RISE
  digital write pin P2 to 0

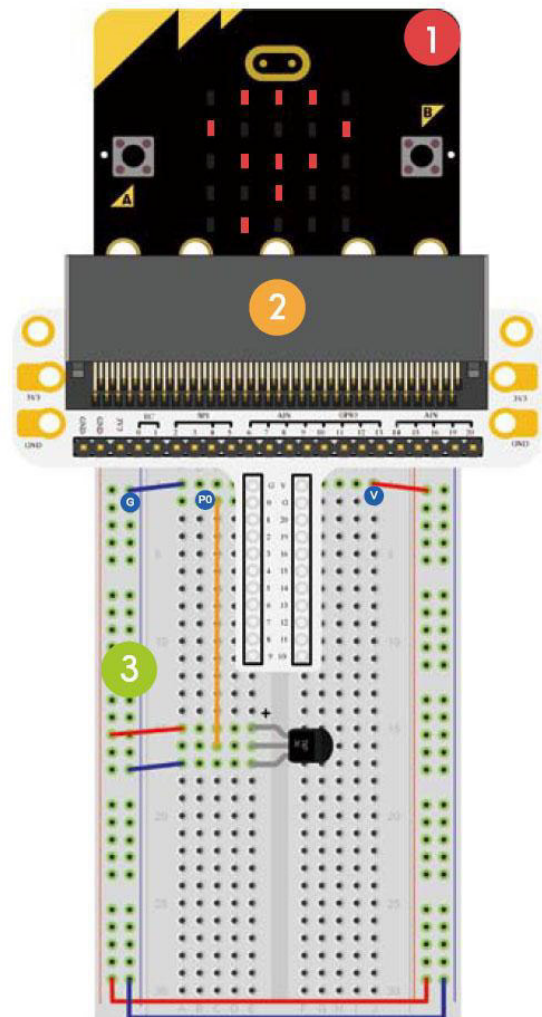
```

1.	Drag and drop the code blocks to form the code as shown. You can find the <b>set pin P0 to emit edge events</b> and <b>set pull P0 to up</b> blocks in <b>Pins</b> → <b>More</b> in the Code Drawer. You can find the <b>on event from MICROBIT...EVT_FALL/RISE</b> blocks in <b>Control</b> in the Code Drawer.
2.	Set the events type on <b>edge</b> . Set the pull pin to <b>P0</b> and to <b>up</b> .
3.	Now, make a fall event block. Drag and drop the <b>on event from MICROBIT...EVT°FALL</b> block below the <b>on start</b> block. Click on the first arrow and select the <b>MICROBIT_ID_IO_P0</b> option in the pull-down menu. Click on the second arrow and select the <b>MICROBIT_PIN_EVENT_FALL</b> option in the pull-down menu.  Drag and drop the <b>digital write pin P0 to 0</b> block into the event block. Set pin P0 to P2 and set 0 to 1. You can find the <b>digital write pin P0 to 0</b> block in <b>Pins</b> in the Code Drawer.
4.	Do the same for the second event block. Instead, click on the second arrow and select the <b>MICROBIT_PIN_EVENT_RISE</b> option in the pull-down menu. Also, set pin P0 to P2 of the <b>digital write pin P0 to 0</b> block and set the value to 0.
5.	When complete, we compile the programme and generate a hex. file. Click on the download button and save the hex. file to the Downloads folder → C:\downloads. This hex. file is ready to upload to the micro:bit.  Plug the micro:bit into a USB port. The, drag and drop the hex. file onto the micro:bit removable device to upload the programme.

Press down the self-locking switch and the LED will switch on. Press again to switch off the LED. Now, how would you control the micro:bit display with this self-locking switch?

## 5.7 Temperature Sensor

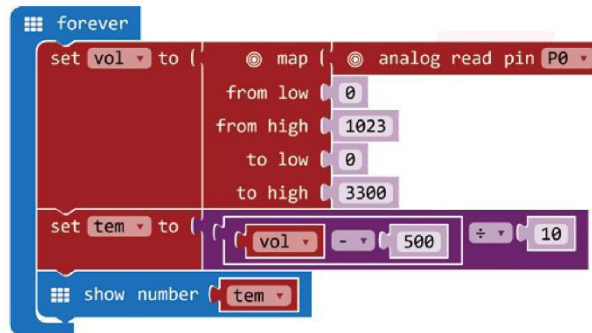
- 1 1x micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1x breadboard
- 4 1x TMP36 temperature sensor



We are going to read out the ambient temperature (data) of the analogue temperature sensor and display the data to the micro:bit.

Place the necessary components on the breadboard as shown.

Have a look at the code below.



1. Drag and drop the code blocks to form the code as shown.  
You can find the **set item to** block in **Variables** in the Code Drawer. You can find the **map/from low/from high/to low/to high** and **analog read pin** blocks in **Pins** in the Code Drawer. You can find the **mathematical function minus** and **divide** in **Math** in the Code Drawer. You can find the **show number** block in **Basic** in the Code Drawer.

2. First, we make two variables. Go to **Variables** in the Code Drawer and click on **Make a Variable**.

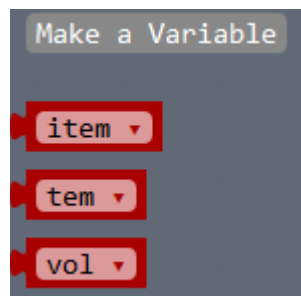
New variable name:

Ok ✓

Cancel ✕

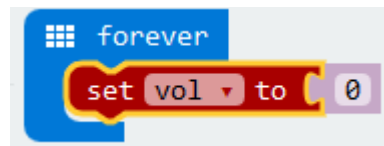
Enter **vol** in the window and click **Ok**. Enter **tem** in the window and click **Ok**.

You will see two new variables under **Variables** in the Code Drawer.

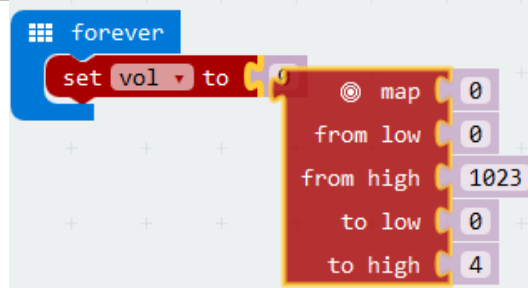


We will need these variables later to save the data in a register.

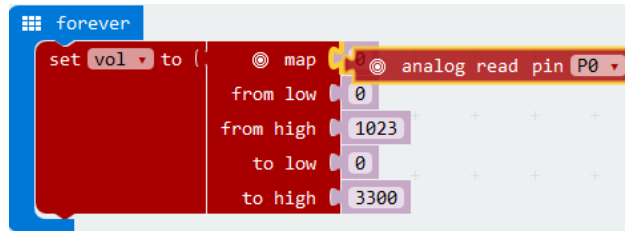
Drag and drop the **set item to** block in the **forever** block and select the **vol** option via the arrow.



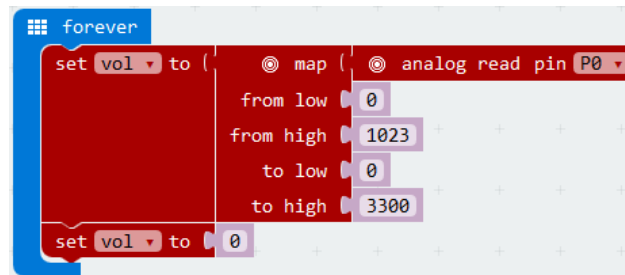
Next, we drag and drop the **map/from low/from high/to low/to high** block over the 0 next to the **set item to** block.



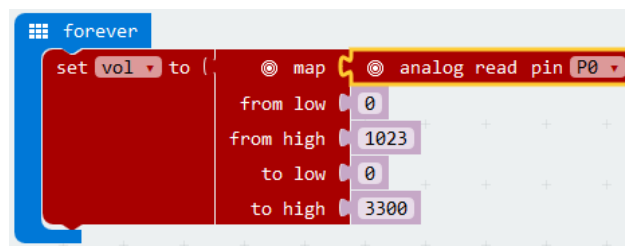
Drag and drop the **analog read pin** block next to the **map/from low/from high/to low/to high** block and set the **to high** value 4 to 3300.



Now, we drag and drop a **set item to** block under the first block. Change the variable **item** to **vol** via the arrow.



The measured voltage in mV via **analog read pin P0** is an 8-bit value of 0-1023 (0-3.3 V) and is mapped from a low value (0 or 0 V) to a high value (3300 mV or 3.3 V). The measured voltage (mV) is saved in the **vol** variable.

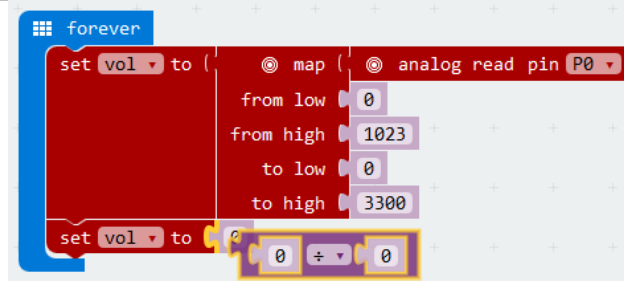


Now, we write a formula to convert the measured voltage into a temperature:

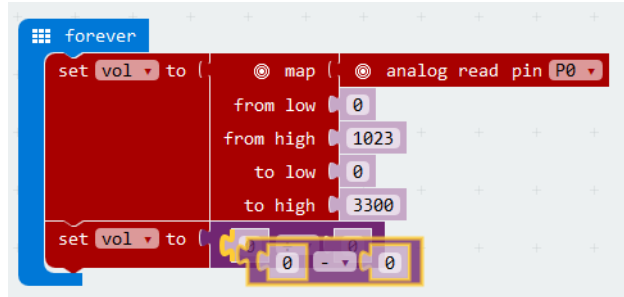
$$\text{Temperature } (^{\circ}\text{C}) = \frac{(\text{Output voltage (mV)} - 500)}{10}$$

In this formula, the output voltage is the **vol** variable (measured through the TMP36 sensor). The result of the formula will be saved in the **tem** variable.

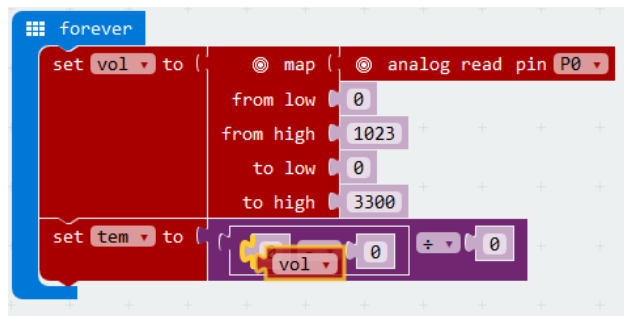
Let's write the formula with blocks. Drag and drop the **mathematical function divide** over the 0 value of the **set vol to** block.



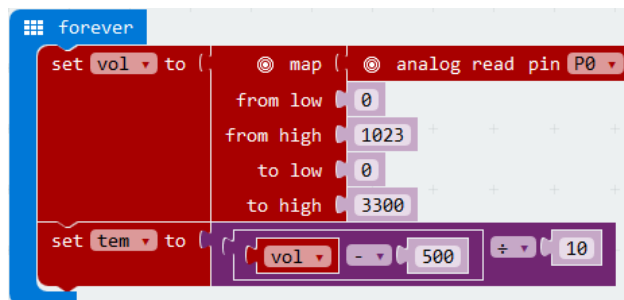
Drag and drop the **mathematical function minus** over the first 0 of the **mathematical function divide**.



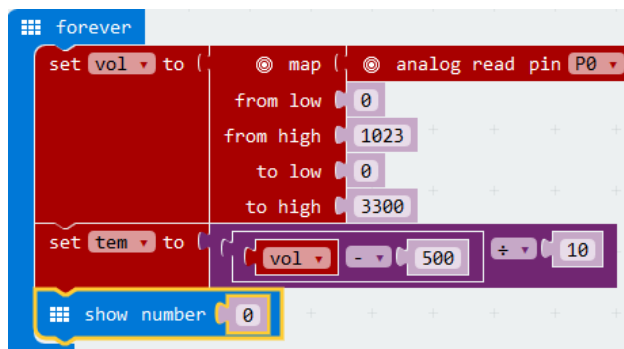
Now, set the second variable from **vol** to **tem**, and drag and drop the **vol** variable over the first 0 of the **mathematical function minus**. The variable **vol** can be found in **Variables** in the Code Drawer.



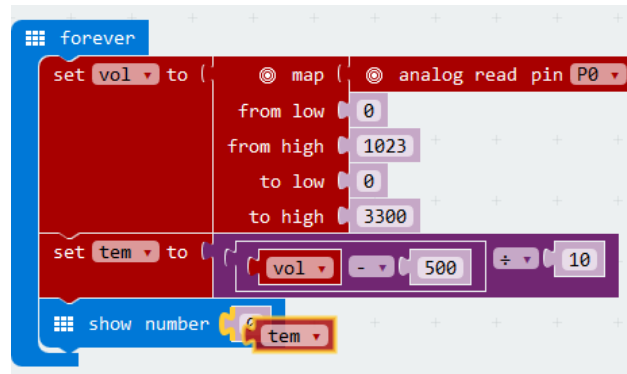
Set the value 0 of the **mathematical function minus** to 500. Also, Set the value of the **mathematical function divide** to 10



Drag and drop the **show number** block under the **set tem to** block.



Finally, drag and drop the **tem** variable over the value 0 of the **show number** block.



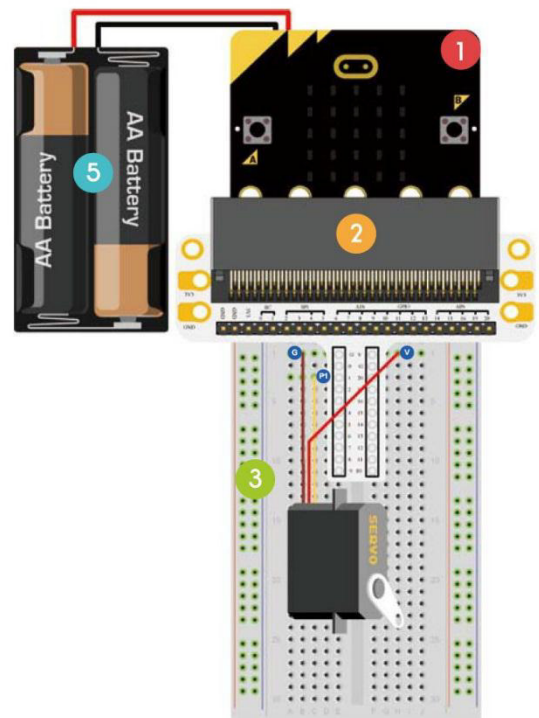
You have written the code!

- 3.** When complete, we compile the programme and generate a hex. file. Click on the download button and save the hex. file to the Downloads folder → C:\downloads. This hex. file is ready to upload to the micro:bit. Plug the micro:bit into a USB port. The, drag and drop the hex. file onto the micro:bit removable device to upload the programme.

You will see the LED beads flash alternatively. Now, how would you display the temperature in degrees Fahrenheit?

## 5.8 Servo

- 1 1x micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1x breadboard
- 4 1x mini servo
- 5 1x battery holder with 2x AA 1.5 V battery



We are going to make a servo rotate continuously within a travel range (0-180°).

Place the necessary components on the breadboard as shown.

Have a look at the code below.



```

forever
  servo write pin P1 to 0
  pause (ms) 2000
  servo write pin P1 to 180
  pause (ms) 2000

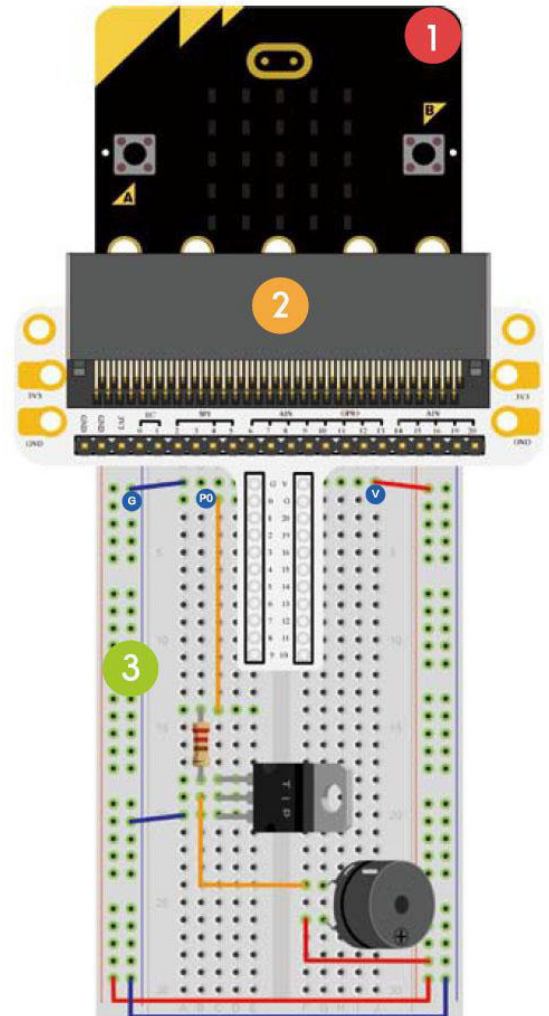
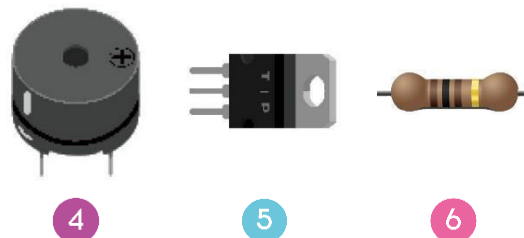
```

1.	Drag and drop the code blocks to form the code as shown. You can find the <b>servo write pin P0 to 180</b> block in <b>Pins</b> in the Code Drawer. You can find the <b>forever</b> and <b>pause (ms) 100</b> in <b>Basic</b> in the Code Drawer.
2.	Set <b>P0</b> to <b>P1</b> and set the value to 0.
3.	Set the <b>pause</b> to 2000 ms.
4.	Set <b>P0</b> to <b>P1</b> . Also, set the value 0 to 180.
5.	Set the <b>pause</b> to 2000 ms.
6.	When complete, we compile the programme and generate a hex. file. Click on the download button and save the hex. file to the Downloads folder → C:\downloads. This hex. file is ready to upload to the micro:bit. Plug the micro:bit into a USB port. The, drag and drop the hex. file onto the micro:bit removable device to upload the programme.

We can see the servo rotating from 0 to 180 degrees. Now, how would you make a dial thermometer with a temperature sensor and servo?

## 5.9 Buzzer

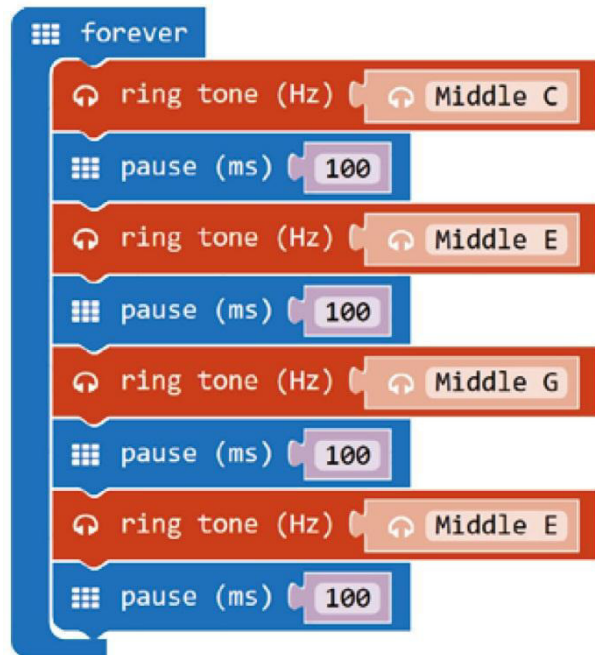
- 1 1x micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1x breadboard
- 4 1x mini speaker
- 5 1x N-channel MOSFET
- 6 1x 100  $\Omega$  resistor (brown/black/brown/gold)

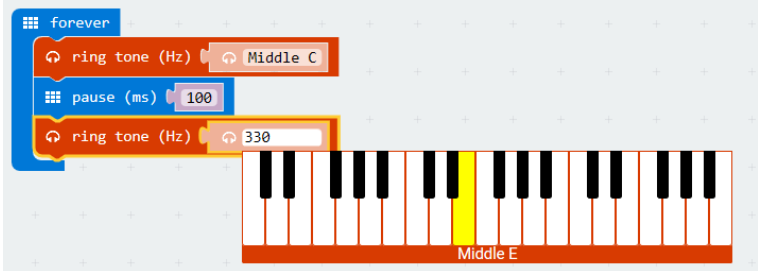


We are going to drive a buzzer.

Place the necessary components on the breadboard as shown.

Have a look at the code below.

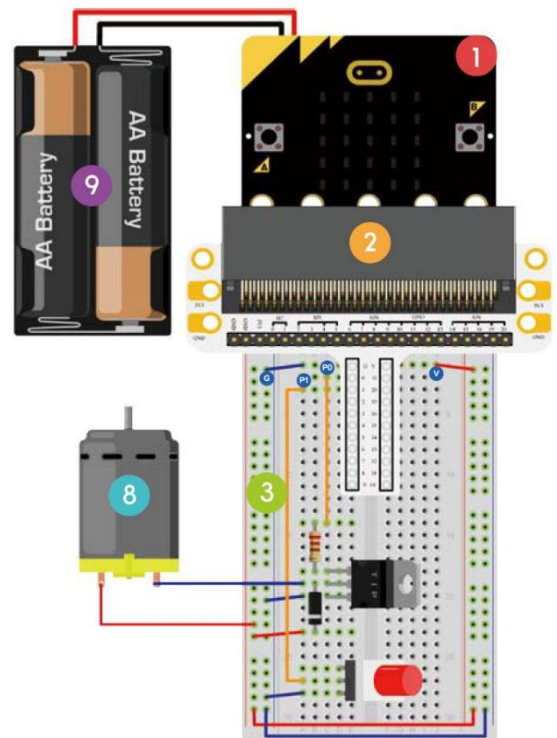
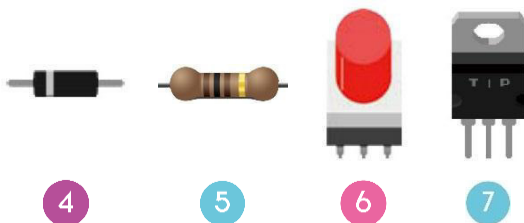


1.	<p>Drag and drop the code blocks to form the code as shown.</p> <p>You can find the <b>forever</b> and <b>pause (ms) 100</b> in <b>Basic</b> in the Code Drawer. You can find the <b>ring tone (Hz)</b> block in <b>Music</b> in the Code Drawer.</p>
2.	<p>Set the first <b>ring tone (Hz)</b> to <b>middle C</b>. To do so, click the text box and select the corresponding piano key.</p>  <p>You will also hear the corresponding tone.</p>
3.	<p>Set the <b>pause</b> to 100 ms.</p>
4.	<p>Do so for the rest of the ring tones and pauses.</p>
5.	<p>When complete, we compile the programme and generate a hex. file. Click on the download button and save the hex. file to the Downloads folder → C:\downloads. This hex. file is ready to upload to the micro:bit.</p> <p>Plug the micro:bit into a USB port. The, drag and drop the hex. file onto the micro:bit removable device to upload the programme.</p>

We can hear a sound from the buzzer. Now, how would you programme your favourite nursery rhyme?

## 5.10 Motor

- 1 1x micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1x breadboard
- 4 1x diode
- 5 1x 100  $\Omega$  resistor (brown/black/brown/gold)
- 6 1x self-locking or bi-stable switch
- 7 1x N-channel MOSFET
- 8 1x mini motor
- 9 1x battery holder with 2x AA 1.5 V battery



We are going to use a switch to control the start and stop of a motor.

Place the necessary components on the breadboard as shown.

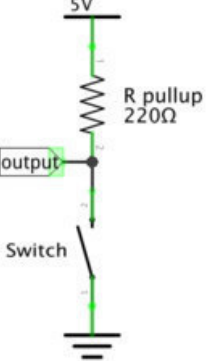
Have a look at the code below.

```

on start
  digital write pin P0 to 0
  set pull pin P1 to up

forever
  if (digital read pin P1 = 0)
  then
    digital write pin P0 to 1
  else
    digital write pin P0 to 0
  
```

1.	Drag and drop the code blocks to form the code as shown.
2.	The <b>on start</b> block runs only once to start the programme.
3.	Set the value of <b>P0</b> to <b>0</b> .
4.	Set the <b>pull pin</b> block to <b>P1</b> and <b>up</b> .

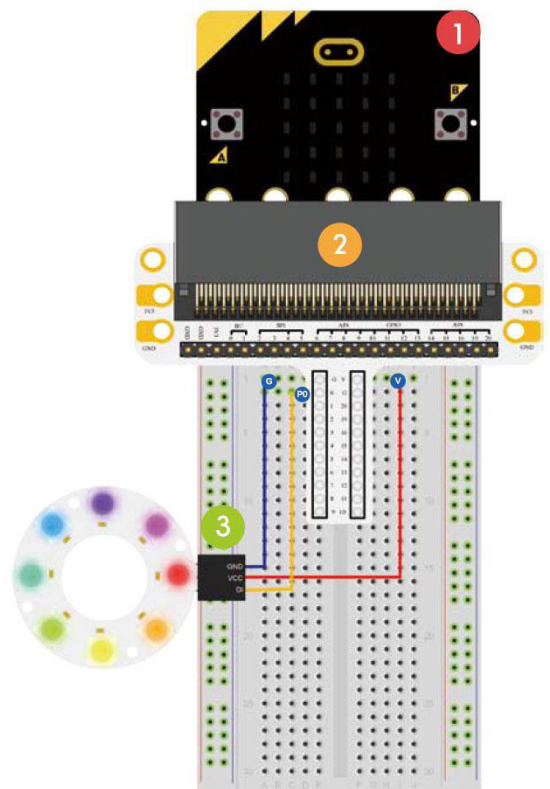
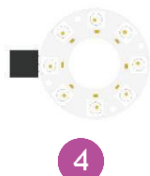
<p><b>5.</b></p>	<p>Within <b>forever</b>, set the <b>digital read pin</b> to <b>P1</b>. Set the <b>mathematical function equal</b> to 0.</p> <p>In reality, this is the scheme. The pull-up resistor must not be added as hardware onto the breadboard. The pull-up function is programmed (see step 3-4) in the code and replaces the need of the resistor on the breadboard.</p> 
<p><b>6.</b></p>	<p>Once the switch is pressed, set high voltage to P0. Set value 0 to 1 (5 V). The motor starts running.</p>
<p><b>7.</b></p>	<p>Once the switch is released, set low voltage to P0. Set value to 0. The motor stops running.</p>
<p><b>8.</b></p>	<p>When complete, we compile the programme and generate a hex. file. Click on the download button and save the hex. file to the Downloads folder → C:\downloads. This hex. file is ready to upload to the micro:bit.</p> <p>Plug the micro:bit into a USB port. The, drag and drop the hex. file onto the micro:bit removable device to upload the programme.</p>

Press the switch to run the motor and release it to stop. Now, how would you use a trimpot to control the motor speed?

Note: Since the micro:bit voltage is 3.3 V only, it may not be enough to support the fan. To make the fan run, you may have to rotate the blade to help it start.

### 5.11 Rainbow LED

- 1 1x micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1x breadboard
- 4 1x RGB LED ring



We are going to drive eight RGB LEDs in a ring and realize a gradual rainbow.

Place the necessary components on the breadboard as shown.

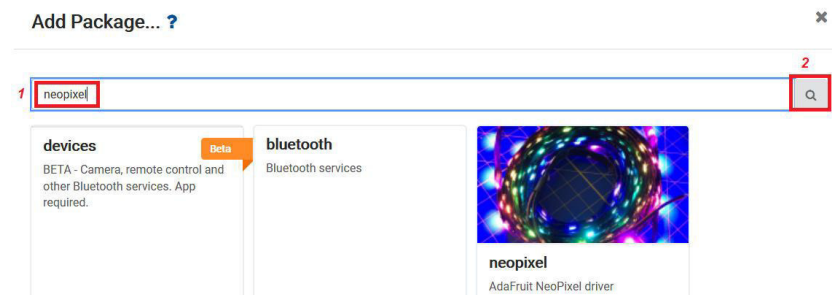
Have a look at the code below.

```

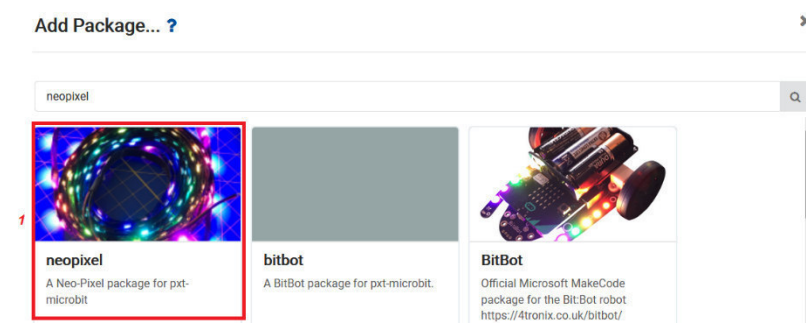
on start
  set strip to NeoPixel at pin P0 with 8 leds as RGB (GRB format)
  strip show rainbow from 1 to 360

forever
  strip show
  strip rotate pixels by 1
  pause (ms) 100
  
```

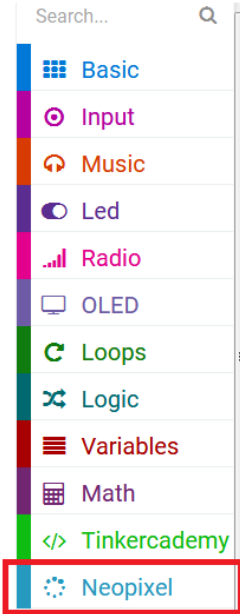

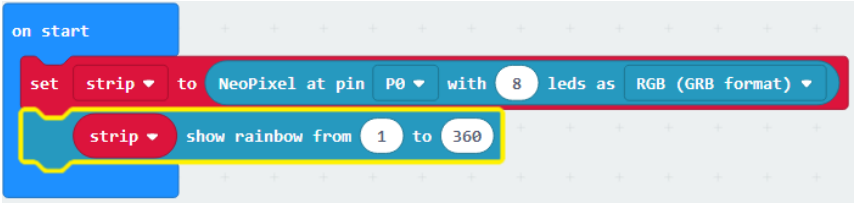
1. Drag and drop the code blocks to form the code as shown.
2. Search and add the NeoPixel library.  
Go to Advanced → Add Package (Extensions) and enter NeoPixel.



Next, select the NeoPixel library.



Now, the library is downloaded and added to the Code Drawer.

	 <p>You can find the <b>NeoPixel at pin P0 with 24 leds, item show rainbow from 1 to 360, item show</b> and <b>items rotate pixels by 1</b> blocks in <b>NeoPixel</b> in the Code Drawer.</p>
<p>3.</p>	<p>Drag and drop the <b>set strip to NeoPixel at pin P0 with 24 leds as RGB (GRB format)</b> block in the <b>on start</b> block. Set pin to <b>P0</b> and the value 24 to 8 (we have 8 LEDs on the LED ring).</p> 
<p>4.</p>	<p>Now, drag and drop the <b>set strip to rainbow from 1 to 360</b> block under the <b>set strip to NeoPixel at pin P0</b> block.</p> 
<p>5.</p>	<p>Complete the code according to the example.</p>
<p>6.</p>	<p>When complete, we compile the programme and generate a hex. file. Click on the download button and save the hex. file to the Downloads folder → C:\downloads. This hex. file is ready to upload to the micro:bit. Plug the micro:bit into a USB port. The, drag and drop the hex. file onto the micro:bit removable device to upload the programme.</p>

We can see rainbow rotating on the LED ring. Now, how would you make the ring blink like an eye?

**Use this device with original accessories only. Velleman nv cannot be held responsible in the event of damage or injury resulting from (incorrect) use of this device. For more info concerning this product and the latest version of this manual, please visit our website [www.velleman.eu](http://www.velleman.eu). The information in this manual is subject to change without prior notice.**

#### © COPYRIGHT NOTICE

**The copyright to this manual is owned by Velleman nv. All worldwide rights reserved.** No part of this manual may be copied, reproduced, translated or reduced to any electronic medium or otherwise without the prior written consent of the copyright holder.



# HANDLEIDING

## 1. Inleiding

**Aan alle ingezetenen van de Europese Unie**

**Belangrijke milieu-informatie betreffende dit product**



Dit symbool op het toestel of de verpakking geeft aan dat, als het na zijn levenscyclus wordt weggeworpen, dit toestel schade kan toebrengen aan het milieu. Gooi dit toestel (en eventuele batterijen) niet bij het gewone huishoudelijke afval; het moet bij een gespecialiseerd bedrijf terechtkomen voor recyclage. U moet dit toestel naar uw verdeler of naar een lokaal recyclagepunt brengen. Respecteer de plaatselijke milieuwetgeving.

**Hebt u vragen, contacteer dan de plaatselijke autoriteiten betreffende de verwijdering.**

Dank u voor uw aankoop®! Lees deze handleiding grondig door voor u het toestel in gebruik neemt. Werd het toestel beschadigd tijdens het transport, installeer het dan niet en raadpleeg uw dealer.

## 2. Veiligheidsvoorschriften



Dit toestel is geschikt voor gebruik door kinderen vanaf 8 jaar, door personen met fysieke, zintuiglijke of verstandelijke beperkingen, of door personen met gebrek aan ervaring en kennis, op voorwaarde dat dit onder toezicht gebeurt van een persoon die verantwoordelijk is voor hun veiligheid of hun aanwijzingen heeft gegeven, hoe zij het toestel moeten gebruiken en zich bewust zijn van de risico's die het gebruik van het toestel met zich meebrengt. Kinderen mogen niet met het toestel spelen. De reiniging en het onderhoud van het toestel mogen niet worden uitgevoerd door kinderen, tenzij ze onder toezicht staan.



Uitsluitend voor gebruik binnenshuis.

Bescherm tegen regen, vochtigheid en opspattende vloeistoffen.

## 3. Algemene richtlijnen



- Raadpleeg de Velleman® service- en kwaliteitsgarantie achteraan deze handleiding.
- Leer eerst de functies van het toestel kennen voor u het gaat gebruiken.
- Om veiligheidsredenen mag u geen wijzigingen aanbrengen. Schade door wijzigingen die de gebruiker heeft aangebracht aan het toestel valt niet onder de garantie.
- Gebruik het toestel enkel waarvoor het gemaakt is. De garantie vervalt automatisch bij ongeoorloofd gebruik.
- De garantie geldt niet voor schade door het negeren van bepaalde richtlijnen in deze handleiding en uw dealer zal de verantwoordelijkheid afwijzen voor defecten of problemen die hier rechtstreeks verband mee houden.
- Noch Velleman nv noch zijn verdelers kunnen aansprakelijk gesteld worden voor schade (buitengewoon, incidenteel of onrechtstreeks) – van welke aard dan ook (financieel, fysisch...) voortvloeiend uit het bezit, gebruik of falen van dit product.
- We streven voortdurend naar verbetering van onze producten. Daarom kan dit product uiterlijk verschillen van de afbeeldingen.
- De afbeeldingen van het product zijn enkel ter illustratie.
- Schakel het toestel niet onmiddellijk in nadat het werd blootgesteld aan temperatuurschommelingen. Om beschadiging te vermijden, moet u wachten tot het toestel de kamertemperatuur heeft bereikt.
- Bewaar deze handleiding voor verdere raadpleging.

## 4. Omschrijving

Deze starterskit is een educatieve kit gebaseerd op micro:bit. Deze kit wordt geleverd met elektronische componenten, een breadboard, aansluitkabels en een micro:bit.

## 5. Voorbeelden

### 5.1 Led

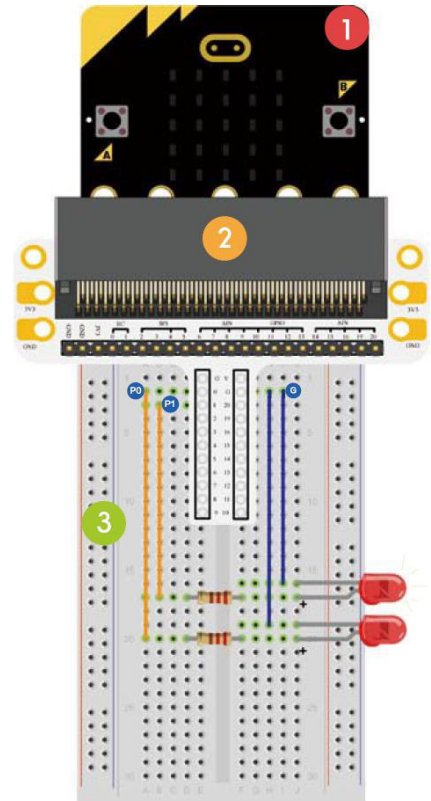
- 1 micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1 x breadboard
- 4 2x rode led (polarisatie: anode (+) = lang pootje, kathode (-) = kort pootje)
- 5 2x 100  $\Omega$  weerstand (bruin/zwart/bruin/goud)



4



5



In deze tutorial gebruiken we de micro:bit om de twee leds afwisselend te laten knipperen.

Plaats de nodige componenten op het breadboard (zie afb.).

Het programma is geschreven in blokken, online in de webbrowser. Ga naar de website [www.makecode.com](http://www.makecode.com) of <https://www.microsoft.com/en-us/makecode?rtc=1> en klik op de knop micro:bit en vervolgens op **Start Project**.

Microsoft Office Windows Surface Xbox Deals Support More

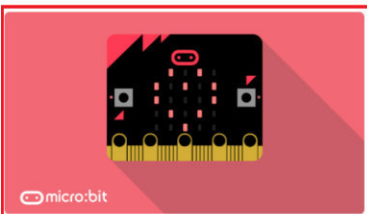
MakeCode About Get started Resources

Search Microsoft.com Sign in

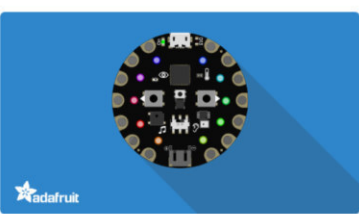
### Hands on computing education

Microsoft MakeCode brings computer science to life for all students with fun projects, immediate results, and both block and text editors for learners at different levels.


*klik dit icoon aan*



micro:bit  
[Start coding with micro:bit >](#)



Circuit Playground Express  
[Start coding with Circuit Playground Express >](#)



Minecraft  
[Start coding with Minecraft >](#)

De micro:bit wordt in een nieuw venster geopend. U kunt nu de code samenstellen met blokken die uit het menu worden gesleept en in de code-editor worden geplaatst.

Lees hieronder hoe het werkt...

## Wat is MakeCode?

Eerdere naam PXT – Programming eXperience Toolkit Editor om codes te schrijven voor de micro:bit.

Een grafische en beginnersvriendelijke drag-and-drop code-editor vergelijkbaar met Scratch, die online werkt in de browser.

micro:bit [Let's Code](#) [Ideas](#) [Meet micro:bit](#) [Teach](#) [Buy](#) [English](#)

## Power your imagination with code

Did you know that you can code your BBC micro:bit using Blocks, JavaScript, and Python?  
If you have never used a BBC micro:bit try our [Quick Start Guide](#).

**JavaScript Blocks Editor**

The micro:bit's JavaScript Blocks editor makes it easy to program your BBC micro:bit in Blocks and JavaScript.

Powered by MakeCode. If you have any issues accessing the editor, check that it isn't blocked in your school. If you need some inspiration then check out these [Projects](#).

[Let's Code](#)

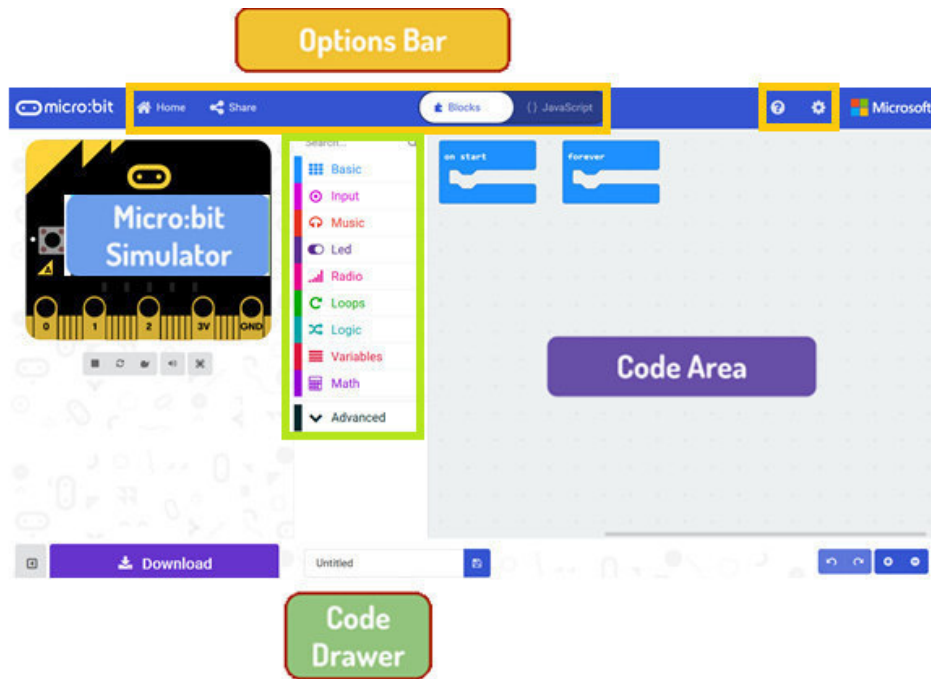
[Reference](#)

[Lessons](#)

U kunt zien dat de editor uit verschillende secties bestaat. Maak uw code in het **codeveld**, door blokken te slepen uit het menu **Code Drawer**. Het resultaat van de code verschijnt onmiddellijk in de **micro:bit simulator**. Met de knop Download (onderaan de pagina) kunt u de code downloaden en naar de micro:bit uploaden.



Vervolgens gaan we dieper in op het gebruik van de **Optiebalk**.



## Optiebalk



Klik om een nieuw project te maken of toe te voegen.

Importeer hier uw projecten.

Klik op Blocks om scripts te schrijven met de Block-editor.

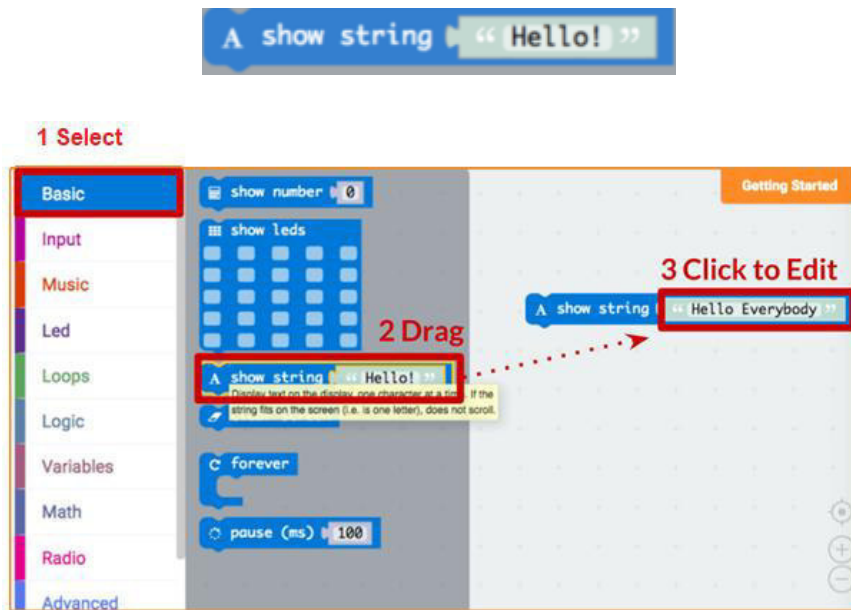
Klik op Javascript om uw script om te zetten in Javascript.

Klik op het vraagteken voor meer informatie over de verschillende blokken en functies.

Klik op het tandwielicoontje om projectinstellingen te bewerken. Via dit menu kunt u uw projecten een andere naam geven en verwijderen. **Reset** verwijdert alle opgeslagen projecten, wees dus voorzichtig. Het is meestal het best om **Project verwijderen** te gebruiken.

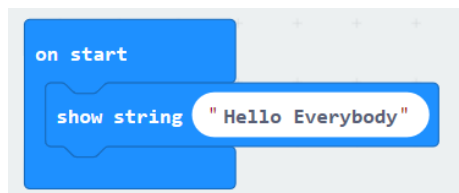
### De ledmatrix gebruiken

We beginnen met het schrijven van de code! Telkens u klikt op een blok uit het menu, wordt een lijst met alle beschikbare codes weergegeven. Sleep het blok **show string** uit het menu en klik in het blok om te bewerken.



<b>1</b>	Selecteren
<b>2</b>	Slepen

<b>3</b>	Klik om te bewerken
----------	---------------------

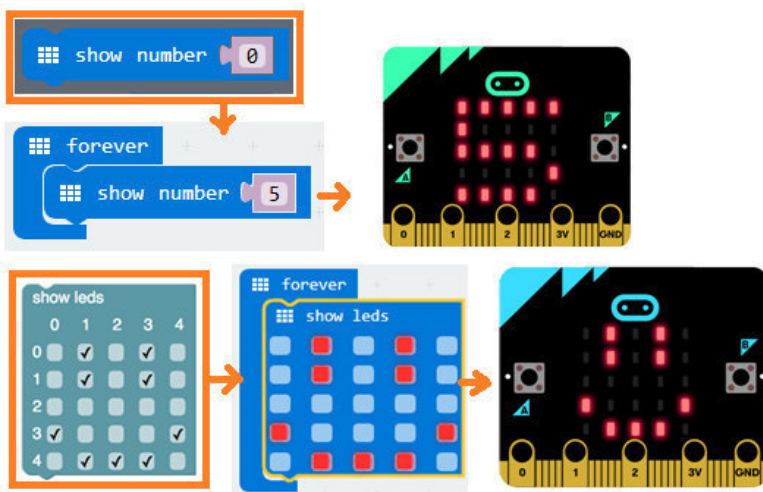


Test de code in de simulator!

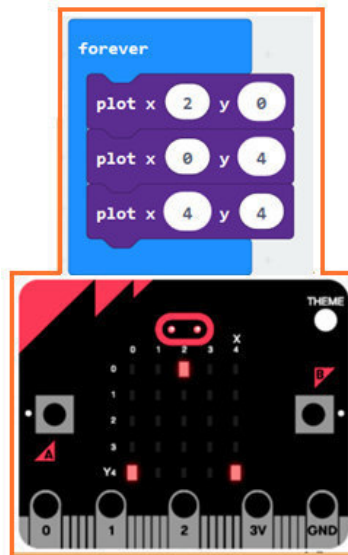
### Say Hello!

De stukjes tekst die verschijnen zijn **strings**.

U kunt een willekeurig getal weergeven met **show number** of een 5x5 pixel afbeelding op het scherm weergeven met **show leds**.



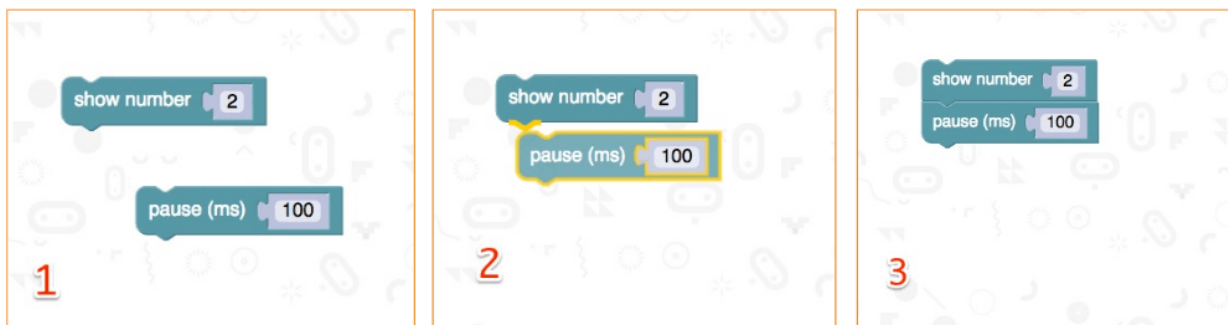
U kunt ook coördinaten gebruiken om een specifieke led te gebruiken. De coördinaten (0.0) bevinden zich in de linkerbovenhoek.



### Blokken samenvoegen

Klik op het blokje dat u wilt samenvoegen. Sleep het blok naar het doelblok totdat de zijkant van het doelblok oplicht. Laat los en de twee blokjes zijn samengevoegd!

Als u op het eerste blok klikt, wordt het tweede verplaatst. Als u op het tweede blokje klikt, wordt deze van het eerste losgekoppeld.



### Op de micro:bit testen

Sluit de micro:bit aan op de computer met een micro-USB-kabel.

Een testroutine verschijnt op de micro:bit waarbij u wordt gevraagd om op knoppen te drukken en een spel te spelen. Probeer het uit!

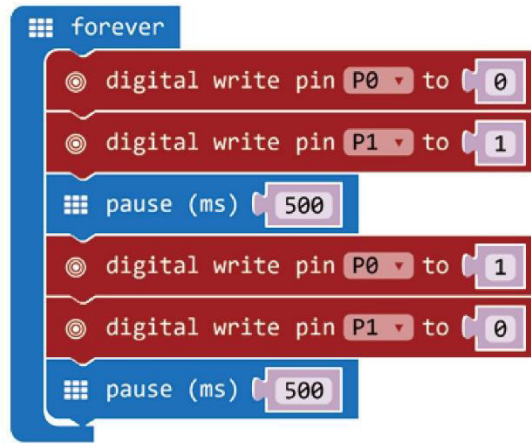
Klik op **Download** om uw script naar een **hex-bestand** om te zetten en te downloaden.

Als Chrome correct is ingesteld, dan kunt u de volgende stap overslaan. Sleep het gedownloade hex.bestand naar de micro:bit drive of klik rechts en kies **Send To** in Windows®.

U hebt uw eigen uw toestel geprogrammeerd!

Bekijk de onderstaande code.



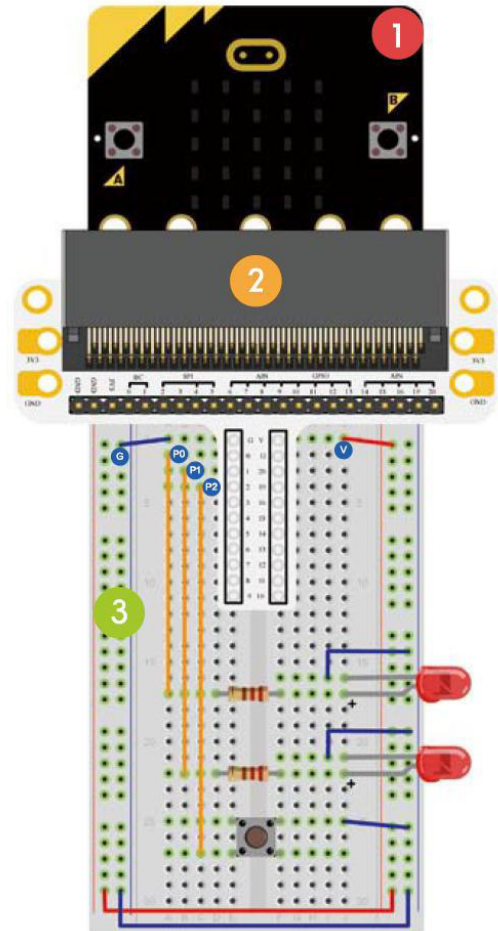
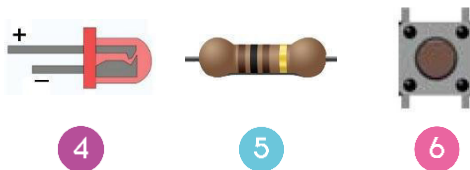


1.	Sleep de blokken om de code te maken (zie afb.). Het blok <b>digital write pin P0 to 0</b> bevindt zich in het menu onder <b>Advanced</b> → <b>Pins</b> . De blokken <b>forever</b> en <b>pause (ms) 100</b> bevinden zich in het menu onder <b>Basic</b> .
2.	Stel de waarde voor P0 in op 0. LED0 off = low voltage = 0 V = digital 0. Stel de waarde voor P1 in op 1. LED1 on = high voltage = 5 V = digital 1.
3.	Stel de <b>pauze</b> in op 500 ms.
4.	Stel de waarde voor P0 in op 1. LED0 on = high voltage = 5 V = digital 1. Stel de waarde voor P1 in op 0. LED1 off = low voltage = 0 V = digital 0.
5.	Stel de <b>pauze</b> in op 500 ms.
6.	Compileer het programma en sla het op als een .hex-bestand. Klik op de knop Download en sla het .hex-bestand op in de map Downloads → C:\downloads. Dit .hex-bestand kan naar de micro:bit geüpload worden. Sluit de micro:bit aan op een USB-poort. Sleep het .hex-bestand naar de micro:bit [removable device] om het programma te uploaden.

De twee leds knipperen afwisselend. Wat dacht u van een RGB-verkeerslicht maken?

## 5.2 Knop

- 1 1x micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1 x breadboard
- 4 2x rode led (polarisatie: anode (+) = lang pootje, kathode (-) = kort pootje)
- 5 2x 100  $\Omega$  weerstand (bruin/zwart/bruin/goud)
- 6 1x drukschakelaar met momentcontact



Een knop wordt gebruikt om de knipperende leds te bedienen. Druk op de knop om de leds afwisselend te laten knipperen. Laat de knop los om de led uit te schakelen.

Plaats de nodige componenten op het breadboard (zie afb.).

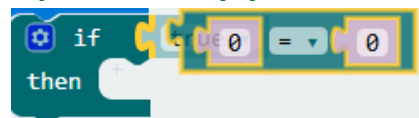
Bekijk de onderstaande code.

```

on start
  set pull pin P2 to up

forever
  if (digital read pin P2 = 0)
  then
    digital write pin P0 to 0
    digital write pin P1 to 1
    pause (ms) 500
    digital write pin P0 to 1
    digital write pin P1 to 0
    pause (ms) 500
  
```

1. Sleep de blokken om de code te maken (zie afb.).  
De blokken **digital read pin P0** en **digital write pin P0 to 0** bevinden zich in het menu onder **Advanced** → **Pins**. De blokken **forever** en **pause (ms) 100** bevinden zich in het menu onder **Basic**. De blokken **if then** en **logic equal function** bevinden zich in het menu onder **Logic**.  
Sleep het blok **logic equal function (=)** over het blok **true**.



De twee blokken schuiven in elkaar.

Sleep het blok **digital read pin P0** over de **0** van het blok **logic equal function (=)** (zie afb.).

	
2.	Stel P2 in als een pull-up weerstand.
3.	Stel <b>digital read pin</b> in op P2.
4.	Stel <b>digital write pin P0 in op 0</b> (laag-actief (0 V)). Stel <b>digital write pin P1 in op 1</b> (hoog-actief (5 V)).
5.	Stel de <b>pauze</b> in op 500 ms.
6.	Stel <b>digital write pin P0 in op 1</b> (hoog-actief (5 V)). Stel <b>digital write pin P1 in op 0</b> (laag-actief (0 V)).
7.	Stel de <b>pauze</b> in op 500 ms.
8.	Compileer het programma en sla het op als een .hex-bestand. Klik op de knop Download en sla het .hex-bestand op in de map Downloads → C:\downloads. Dit .hex-bestand kan naar de micro:bit geüpload worden. Sluit de micro:bit aan op een USB-poort. Sleep het .hex-bestand naar de micro:bit [removable device] om het programma te uploaden.

Druk op de knop en de leds knipperen afwisselend. Hoe zou u de rode led inschakelen door de knop in te drukken en de groene led inschakelen door de knop los te laten?

### 5.3 Trimpotentiometer

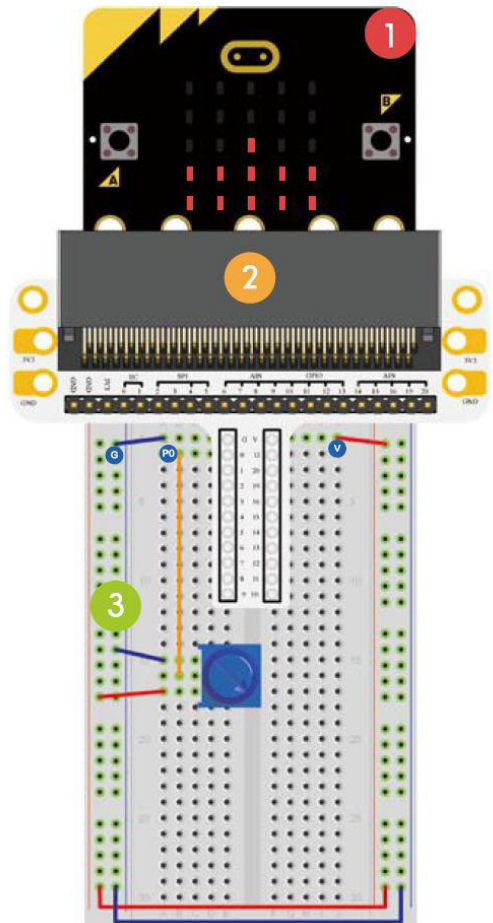
- 1 1x micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1 x breadboard
- 4 1x 10 kΩ trimpotentiometer



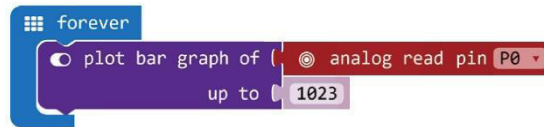
4

In het voorbeeld hieronder lezen we de uitgangsspanning van de trimpotentiometer uit en geven deze weer op het micro:bit-scherm met een staafdiagram.

Plaats de nodige componenten op het breadboard (zie afb.).



Bekijk de onderstaande code.



1.	Sleep de blokken om de code te maken (zie afb.). Het blok <b>plot bar graph of 0 up to 0</b> bevindt zich in het menu onder <b>LED</b> .
2.	Stel <b>analog read pin</b> in op P0. Dit blok bevindt zich in het menu onder <b>Pins</b> . Stel de waarde in op max. 1023.
3.	Compileer het programma en sla het op als een .hex-bestand. Klik op de knop Download en sla het .hex-bestand op in de map Downloads → C:\downloads. Dit .hex-bestand kan naar de micro:bit geüpload worden. Sluit de micro:bit aan op een USB-poort. Sleep het .hex-bestand naar de micro:bit [removable device] om het programma te uploaden.

Draai de trimpotentiometer. De spanning wordt weergegeven op het micro:bit-scherm in de vorm van een staafdiagram. Als de spanning gelijk is aan 0, verschijnt slechts één enkele pixel op het ledscherm. Bij een waarde van 3.3 V licht het volledige scherm op. Hoe zou u de trimpotentiometer gebruiken om de helderheid van de led aan te passen?

#### 5.4 Fotocel

- 1 1x micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1 x breadboard
- 4 1x fotocel
- 5 1x 10 kΩ weerstand  
(bruin/zwart/zwart/rood/bruin)

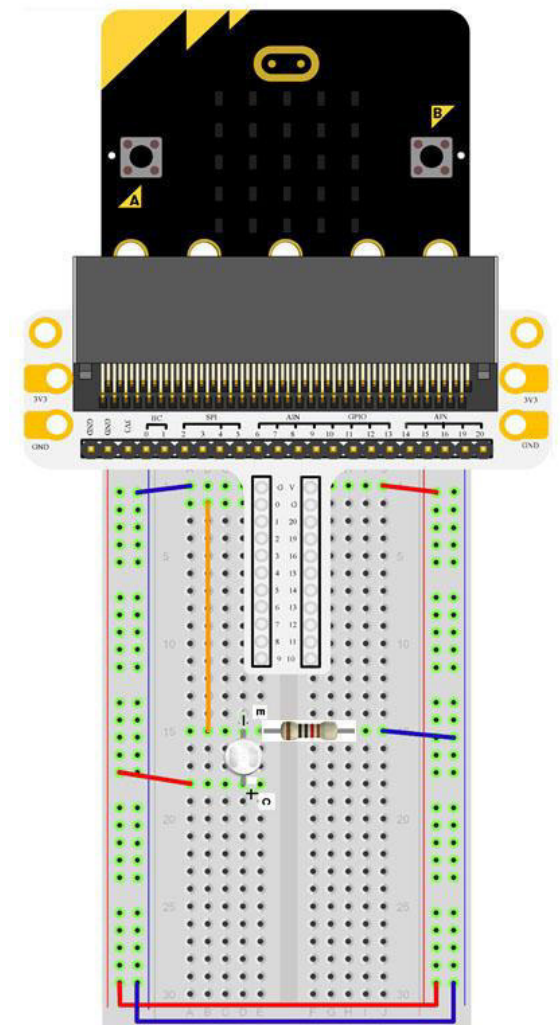


4

Kort pootje = collector  
Positieve anode  
Lang pootje = zender  
Negatieve kathode



5



In het voorbeeld hieronder gebruiken we een fotocel om de helderheid van het micro:bit-scherm te regelen.

Plaats de nodige componenten op het breadboard (zie afb.).

Bekijk de onderstaande code.

```

on start
  set CalVal to (analog read pin P0)

forever
  set PhoVal to (analog read pin P0)
  if (PhoVal < CalVal)
    then show icon
  else clear screen
  
```

1. Eerst maken we twee variabelen. Ga naar **Variables** in het menu en klik op **Make a Variable**.

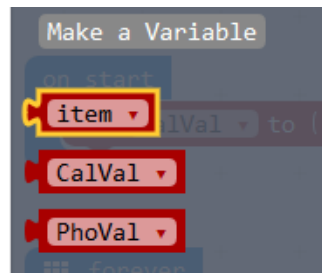
New variable name:

Ok ✓

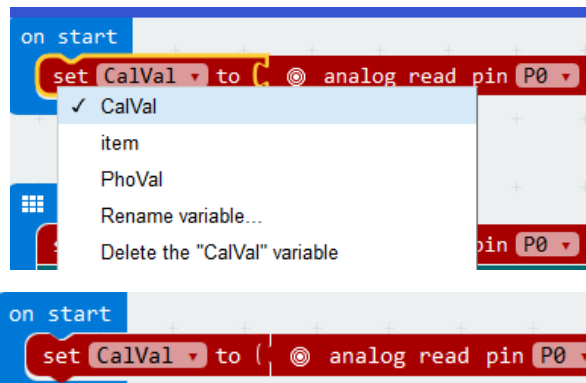
Cancel ✕

Voer **CalVal** in het veld in en klik op **Ok**. Voer **PhoVal** in het veld in en klik op **Ok**.

Twee nieuwe variabelen verschijnen in het menu onder **Variables**.

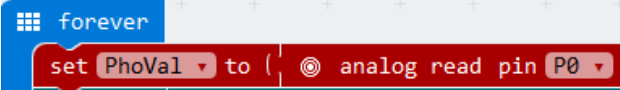
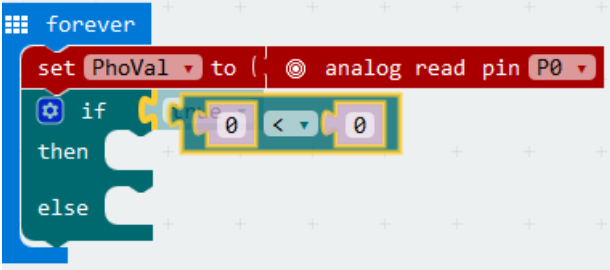
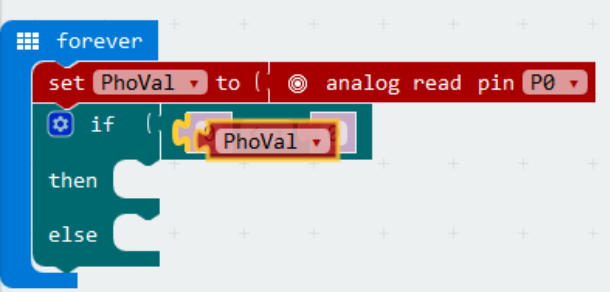
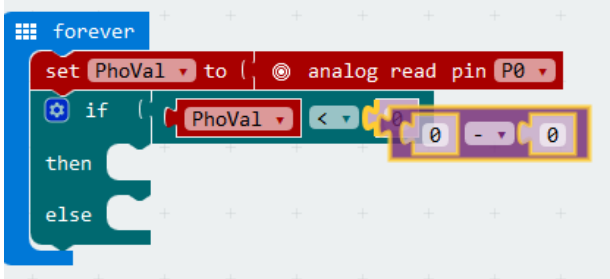
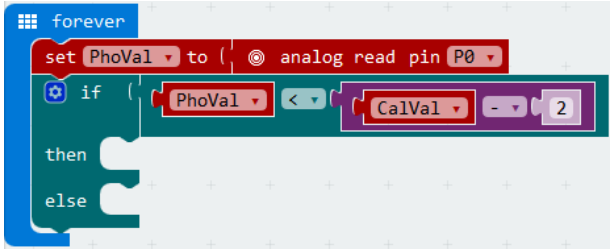


Deze variabelen zijn later nodig om de gegevens in het register op te slaan. De blokken **logic smaller than function** en **if then else** bevinden zich in het menu onder **Logic**. Het blok **mathematical function minus** bevindt zich in het menu onder **Math**. Het blok **show icon** bevindt zich in het menu onder **Basic**. Het blok **clear screen** bevindt zich in het menu onder **Basic** → **More**. Het blok **set item to 0** bevindt zich in het menu onder **Variables**. Klik op het pijltje en selecteer **CalVal** of **PhoVal**.



2. Selecteer de variabele **CalVal** en stel **analog read pin** in op P0.



3.	<p>In het blok <b>forever</b>, selecteer de variabele <b>PhoVal</b> en stel <b>analog read pin</b> in op P0.</p> 
4.	<p>Sleep het blok <b>logic smaller than function (&lt;)</b> naast het blok <b>if</b> en sleep het over het blok <b>true</b>.</p>  <p>Sleep de variabele <b>PhoVal</b> (vanaf het menu onder <b>Variables</b>) over de eerste 0 van het blok <b>logic smaller than function (&lt;)</b>.</p>  <p>Sleep het blok <b>mathematical function minus (-)</b> over de tweede 0 van <b>logic smaller than function (&lt;)</b>.</p>  <p>Sleep de variabele <b>CalVal</b> (vanaf het menu onder <b>Variables</b>) over de eerste 0 van het blok <b>mathematical function minus</b>. Stel de tweede 0 van de <b>mathematical function minus (-)</b> in op 2.</p> 
5.	<p>Sleep het blok <b>show icon</b> naast het blok <b>then</b>. Sleep het blok <b>clear screen</b> naast het blok <b>else</b>.</p>
6.	<p>Als <b>PhoVal</b> kleiner is dan <b>CalVal - 2</b>, verschijnt het hart op het display. Als dit niet het geval is, wordt het uitgeschakeld.</p>



- 7.** Compileer het programma en sla het op als een .hex-bestand. Klik op de knop Download en sla het .hex-bestand op in de map Downloads → C:\downloads. Dit .hex-bestand kan naar de micro:bit geüpload worden. Sluit de micro:bit aan op een USB-poort. Sleep het .hex-bestand naar de micro:bit [removable device] om het programma te uploaden.

Opmerking: Reset de micro:bit om de referentiewaarde te kalibreren volgens het huidige helderheidsniveau. Schakel het licht in om het programma correct te starten.

Als het licht aan is, wordt niets weergegeven. Als het licht uit is, wordt het hart weergegeven. Hoe kunnen we de fotocel gebruiken om een led aan te sturen?

## 5.5 RGB-led

- 1 1x micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1 x breadboard
- 4 1x RGB-led (gemeenschappelijke kathode)
- 5 3x 10  $\Omega$  weerstand (bruin/zwart/bruin/goud)

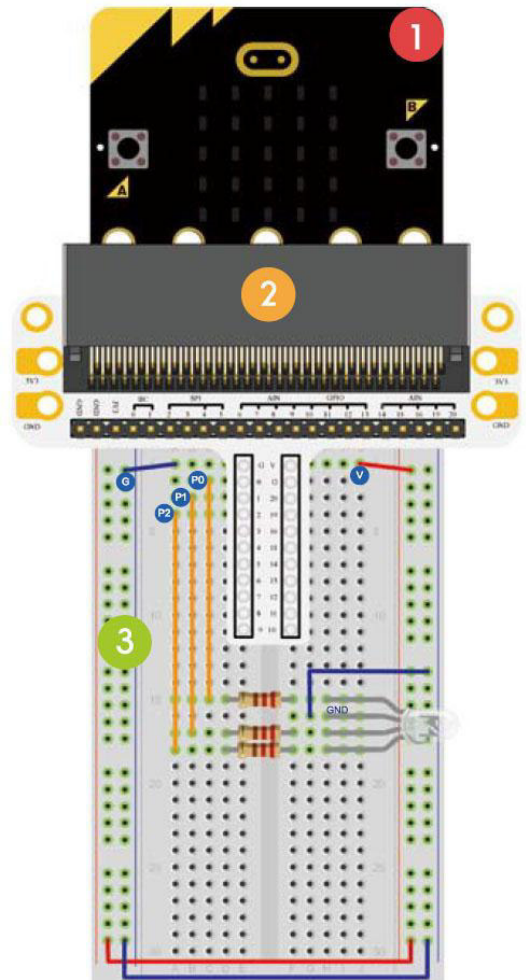


4

1. Rood (anode +)
2. Aarding (kathode -) – langste pootje
3. Groen (anode +)
4. Blauw (anode +)



5



In het voorbeeld hieronder programmeren we een geleidelijke RGB LED-kleurenovergang.

Plaats de nodige componenten op het breadboard (zie afb.).

Bekijk de onderstaande code.

```

on button A pressed
  digital write pin P0 to 1
  digital write pin P1 to 0
  digital write pin P2 to 0

on button B pressed
  digital write pin P0 to 0
  digital write pin P1 to 1
  digital write pin P2 to 0

on button A+B pressed
  digital write pin P0 to 0
  digital write pin P1 to 0
  digital write pin P2 to 1
  
```

- Sleep de blokken om de code te maken (zie afb.).

Het blok **on button A pressed** bevindt zich in het menu onder **Input**. Het blok **digital write pin P0 to 0** bevindt zich in het menu onder **Pins**.

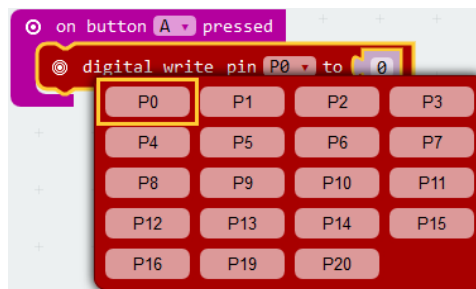
Selecteer optie A in het blok **on button A pressed**.

Sleep 3 blokken **digital write pin P0 to 0** en plaats deze in het blok **on button A pressed**.

Stel pin P0 in het eerste blok **digital write pin P0 in op 0** en verander de waarde 0 naar 1 (rode led aan).

Stel pin P1 in het tweede blok **digital write pin P0 in op 0** en stel de waarde in op 0 (groene led uit).

Stel pin P2 in het derde blok **digital write pin P0 in op 0** en stel de waarde in op 0 (blauwe led uit).



```

on button A pressed
  digital write pin P0 to 1
  digital write pin P1 to 0
  digital write pin P2 to 0
  
```

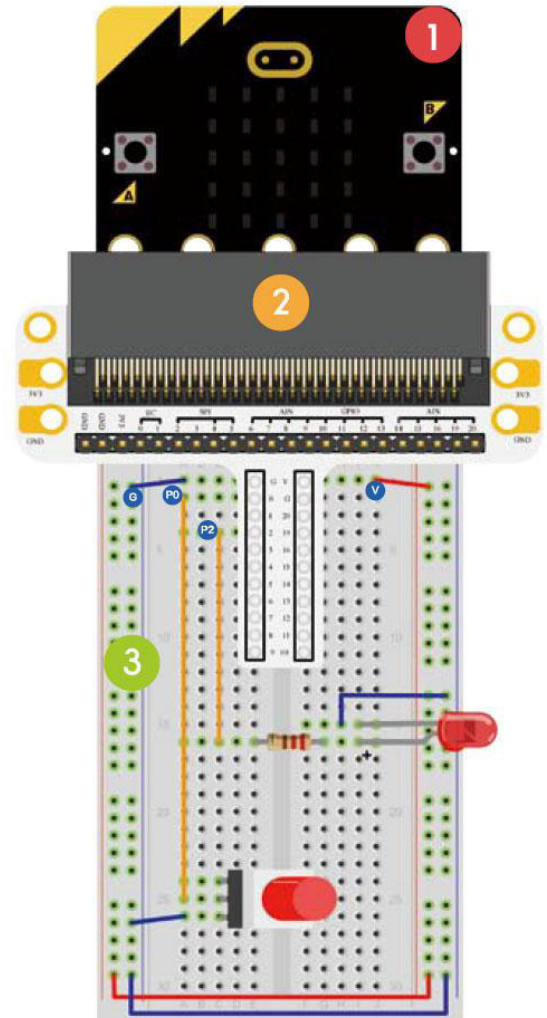
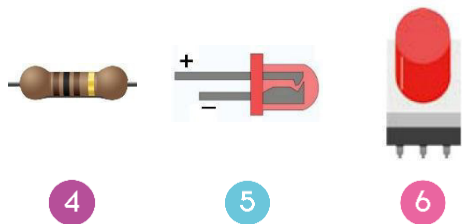
- Op dezelfde manier, compileer de twee blokken **on button B / A + B pressed** om de volledige code te maken.

3. Compileer het programma en sla het op als een .hex-bestand. Klik op de knop Download en sla het .hex-bestand op in de map Downloads → C:\downloads. Dit .hex-bestand kan naar de micro:bit geüpload worden. Sluit de micro:bit aan op een USB-poort. Sleep het .hex-bestand naar de micro:bit [removable device] om het programma te uploaden.

Druk op de knop A om de rode led in te schakelen, druk op de knop B om de groene led in te schakelen, druk de knoppen A en B gelijktijdig in om de blauwe led in te schakelen. Hoe zou u een geleidelijke RGB-kleurovergang creëren?

## 5.6 Zelfvergrendelende schakelaar

- 1 1x micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1 x breadboard
- 4 1x 100 Ω weerstand (bruin/zwart/bruin/goud)
- 5 1x rode led (polarisatie: anode (+) = lang pootje, kathode (-) = kort pootje)
- 6 1x zelfvergrendelende of bistabiele schakelaar



In het voorbeeld hieronder lezen we de omgevingstemperatuur (data) van de analoge temperatuursensor uit en geven de data weer op het micro:bit-scherm.

Plaats de nodige componenten op het breadboard (zie afb.).

Bekijk de onderstaande code.

```

on start
  set pin P0 to emit edge events
  set pull pin P0 to up

on event from MICROBIT_ID_IO_P0 with value MICROBIT_PIN_EVT_FALL
  digital write pin P2 to 1

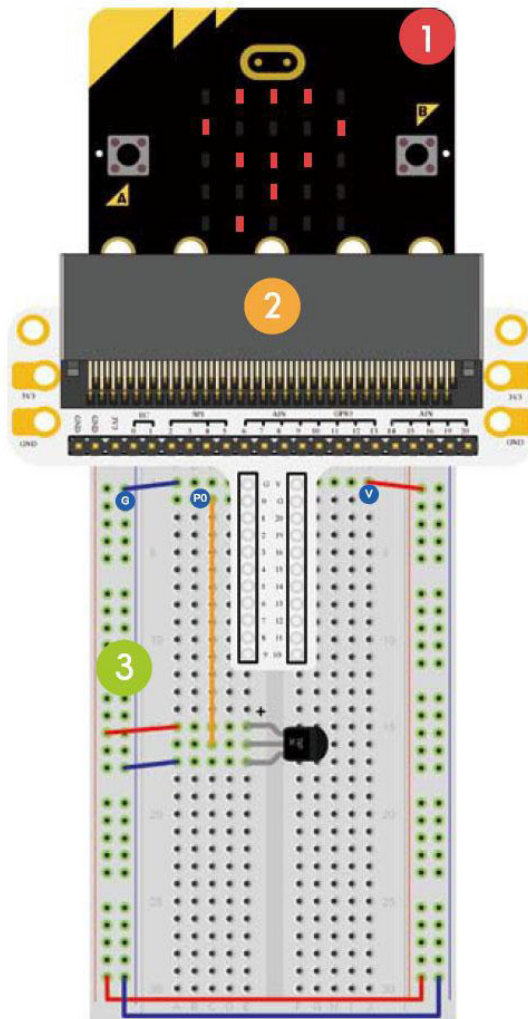
on event from MICROBIT_ID_IO_P0 with value MICROBIT_PIN_EVT_RISE
  digital write pin P2 to 0
  
```

<p><b>1.</b></p>	<p>Sleep de blokken om de code te maken (zie afb.). De blokken <b>set pin P0 to emit edge events</b> en <b>set pull P0 to up</b> bevinden zich in het menu onder <b>Pins</b> → <b>More</b>. De blokken <b>on event from MICROBIT...EVT_FALL/RISE</b> bevinden zich in het menu onder <b>Control</b>.</p>
<p><b>2.</b></p>	<p>Stel het events type in op <b>edge</b>. Stel het blok pull pin in op <b>P0</b> en <b>up</b>.</p>
<p><b>3.</b></p>	<p>Maak een event blok. Sleep het blok <b>on event from MICROBIT...EVT°FALL</b> onder het blok <b>on start</b>. Klik op het eerste pijltje en selecteer de optie <b>MICROBIT_ID_IO_P0</b> in het keuzemenu. Klik op het tweede pijltje en selecteer de optie <b>MICROBIT_PIN_EVENT_FALL</b> in het keuzemenu. Sleep het blok <b>digital write pin P0 to 0</b> in het event blok. Stel pin P0 in op P2 en verander de waarde 0 naar 1. Het blok <b>digital write pin P0 to 0</b> bevindt zich in het menu onder <b>Pins</b>.</p>
<p><b>4.</b></p>	<p>Voer dezelfde stappen uit voor het tweede event blok. Klik vervolgens op de tweede pijl en selecteer de optie <b>MICROBIT_PIN_EVENT_RISE</b> in het keuzemenu. Stel pin P0 in op P2 van het blok <b>digital write pin P0 to 0</b> en stel de waarde in op 0.</p>
<p><b>5.</b></p>	<p>Compileer het programma en sla het op als een .hex-bestand. Klik op de knop Download en sla het .hex-bestand op in de map Downloads → C:\downloads. Dit .hex-bestand kan naar de micro:bit geüpload worden. Sluit de micro:bit aan op een USB-poort. Sleep het .hex-bestand naar de micro:bit [removable device] om het programma te uploaden.</p>

Druk de zelfvergrendelende schakelaar in en de led brandt. Druk nogmaals om de led uit te schakelen. Hoe zou u het micro:bit-scherm aansturen met deze zelfvergrendelende schakelaar?

### 5.7 Temperatuursensor

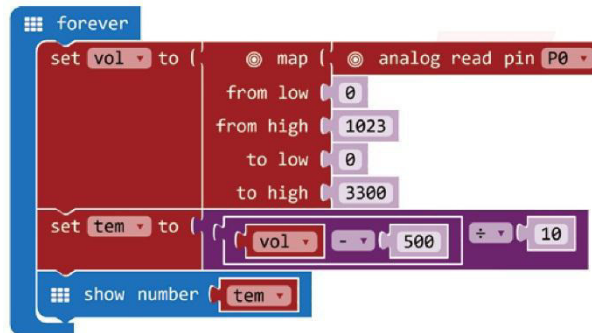
- 1 1x micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1 x breadboard
- 4 1x TMP36 temperatuursensor



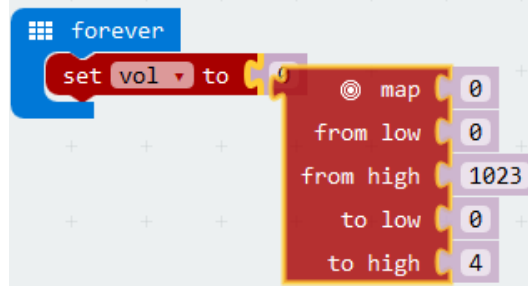
In het voorbeeld hieronder lezen we de omgevingstemperatuur (data) van de analoge temperatuursensor uit en geven de data weer op het micro:bit-scherm.

Plaats de nodige componenten op het breadboard (zie afb.).

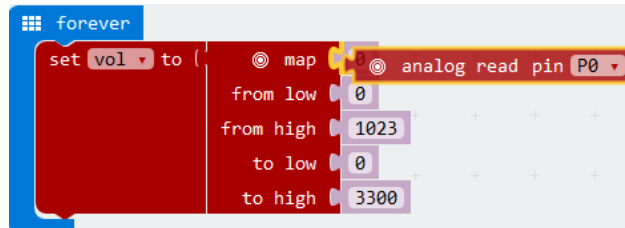
Bekijk de onderstaande code.



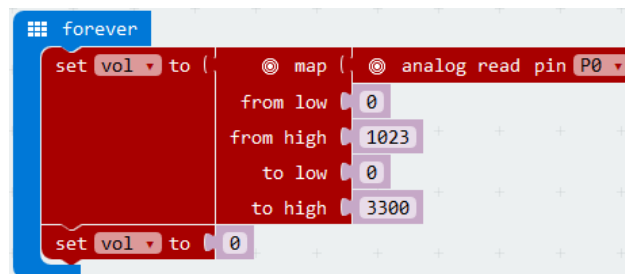
<p><b>1.</b></p>	<p>Sleep de blokken om de code te maken (zie afb.).  Het blok <b>set item to</b> bevindt zich in het menu onder <b>Variables</b>. De blokken <b>map/from low/from high/to low/to high</b> en <b>analog read pin</b> bevinden zich in het menu onder <b>Pins</b>. De blokken <b>mathematical function minus (-)</b> en <b>divide (/)</b> bevinden zich in het menu onder <b>Math</b>. Het blok <b>show number</b> bevindt zich in het menu onder <b>Basic</b>.</p>
<p><b>2.</b></p>	<p>Eerst maken we twee variabelen. Ga naar <b>Variables</b> in het menu en klik op <b>Make a Variable</b>.</p> <div data-bbox="491 846 1230 1048" data-label="Image"> </div> <p>Voer <b>vol</b> in het venster in en klik op <b>Ok</b>. Voer <b>tem</b> in het venster in en klik op <b>Ok</b>.  Twee nieuwe variabelen verschijnen in het menu onder <b>Variables</b>.</p> <div data-bbox="708 1173 1015 1469" data-label="Image"> </div> <p>Deze variabelen zijn later nodig om de gegevens in het register op te slaan. Sleep het blok <b>set item to</b> in het blok <b>forever</b> en selecteer de optie <b>vol</b> met de pijl.</p> <div data-bbox="668 1610 1056 1756" data-label="Image"> </div> <p>Sleep het blok <b>map/from low/from high/to low/to high</b> over de 0 naast het blok <b>set item to</b>.</p>



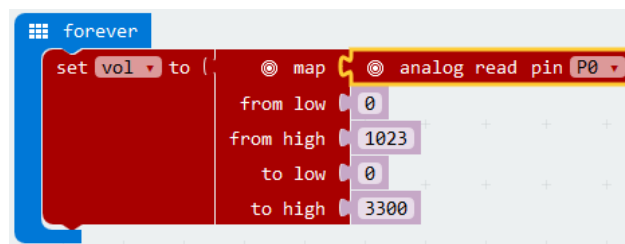
Sleep het blok **analog read pin** naast het blok **map/from low/from high/to low/to high** en stel waarde 4 **to high** in op 3300.



Sleep een blok **set item to** onder het eerste blok. Wijzig de variabele **item** in **vol** met de pijl.



De gemeten spanning in mV met **analog read pin P0** is een 8-bit 0-1023 (0-3.3 V) en wordt weergegeven van laag (0 of 0 V) naar hoog (3300 mV of 3.3 V). De gemeten spanning (mV) wordt opgeslagen in de variabele **vol**.



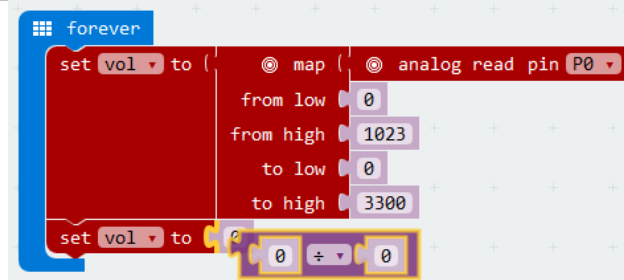
Nu schrijven we een formule om de gemeten spanning om te zetten in temperatuur.

$$Temperature (^{\circ}C) = \frac{(Output\ voltage\ (mV) - 500)}{10}$$

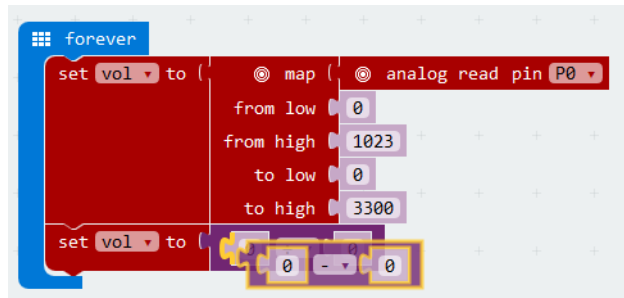
In deze formule is de uitgangsspanning een variabele **vol** (gemeten met de TMP36-sensor). Het resultaat van de formule wordt opgeslagen als variabele **tem**.

Laten we de formule schrijven met blokken. Sleep de **wiskundige functie divide (/)** over de waarde 0 van het blok **set vol to**.

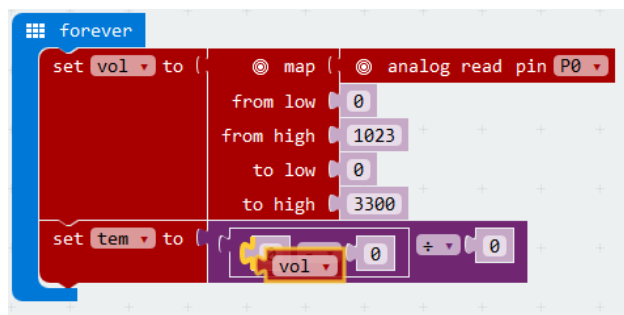




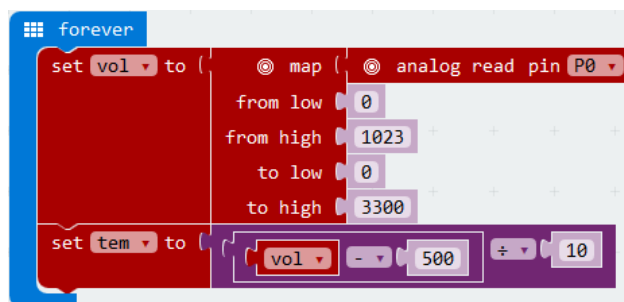
Sleep het blok **mathematical function minus (-)** over de eerst 0 van de **mathematical function divide (/)**.



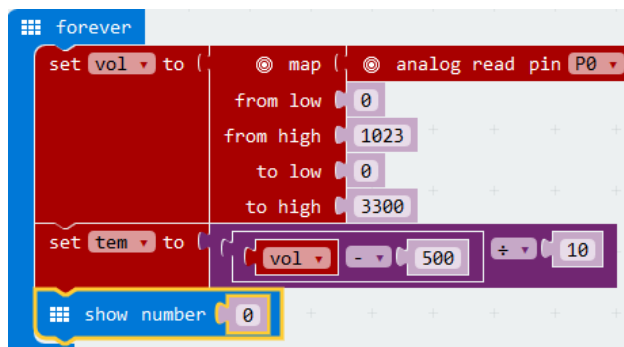
Stel nu de tweede variabele van **vol** in op **tem** en sleep de variabele **vol** over de eerste waarde 0 van de **wiskundige functie minus (-)**. De variabele **vol** bevindt zich in het menu onder **Variables**.



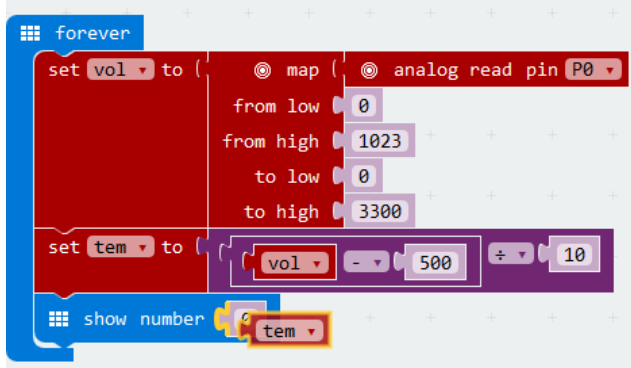
Stel de waarde 0 van de **mathematical function minus (-)** in op 500. Stel de waarde van de **mathematical function device (/)** in op 10.



Sleep het blok **show number** onder het blok **set tem to**.



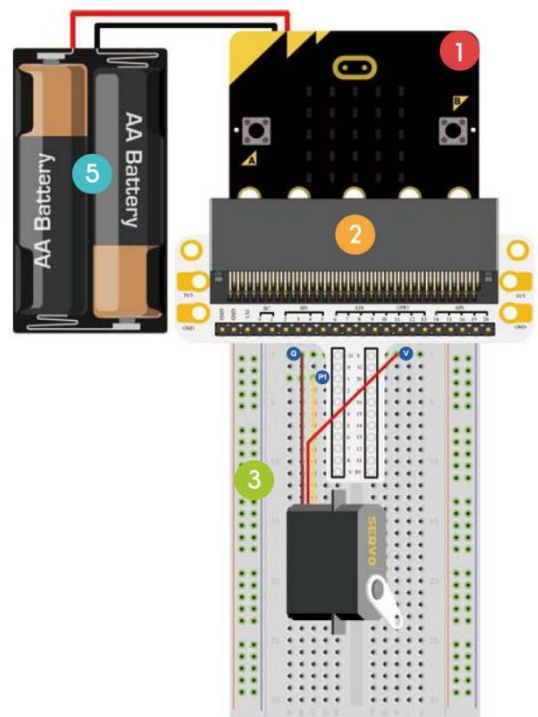
Sleep de variabele **tem** over de waarde 0 van het blok **show number**.

	 <p>U hebt net de code geschreven!</p>
<p><b>3.</b></p>	<p>Compileer het programma en sla het op als een .hex-bestand. Klik op de knop Download en sla het .hex-bestand op in de map Downloads → C:\downloads. Dit .hex-bestand kan naar de micro:bit geüpload worden. Sluit de micro:bit aan op een USB-poort. Sleep het .hex-bestand naar de micro:bit [removable device] om het programma te uploaden.</p>

De twee leds knipperen afwisselend. Hoe zou u de temperatuur in graden Fahrenheit weergeven?

## 5.8 Servo

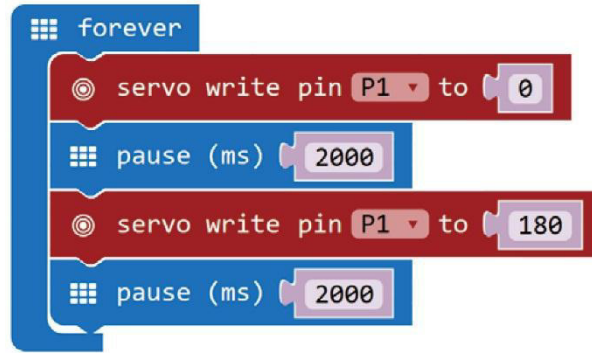
- 1 1x micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1 x breadboard
- 4 1x mini servo
- 5 1x batterijhouder met 2x AA 1.5 V-batterij



In het voorbeeld hieronder laten we de servo continu draaien binnen een bewegingsbereik van 0-180°.

Plaats de nodige componenten op het breadboard (zie afb.).

Bekijk de onderstaande code.

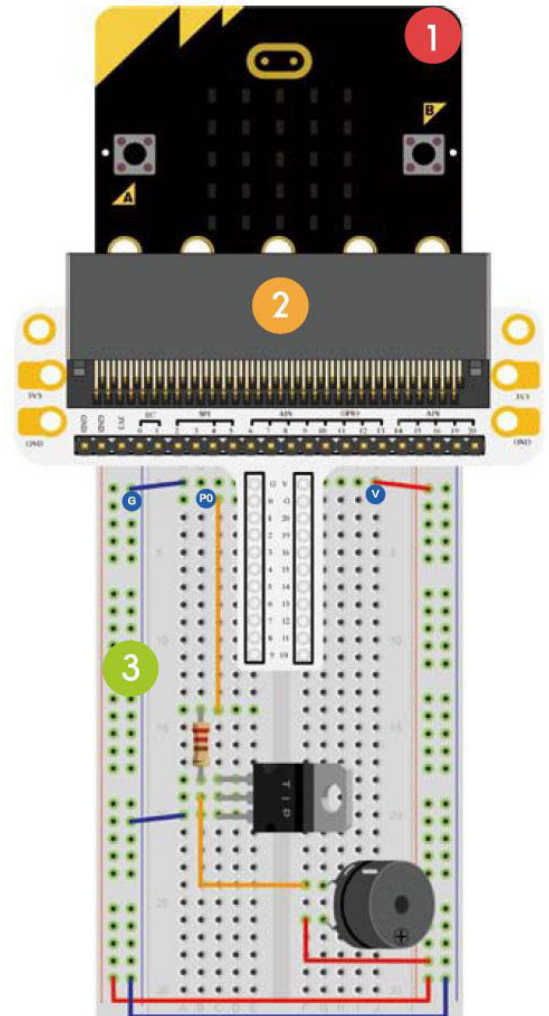
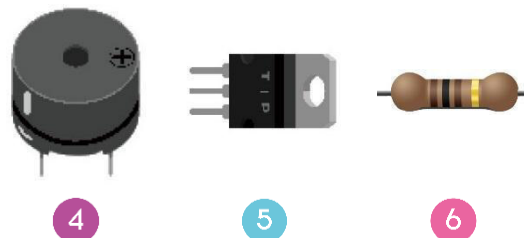


1.	Sleep de blokken om de code te maken (zie afb.). Het blok <b>servo write pin P0 to 180</b> bevindt zich in het menu onder <b>Pins</b> . De blokken <b>forever</b> en <b>pause (ms) 100</b> bevinden zich in het menu onder <b>Basic</b> .
2.	Stel <b>P0</b> in op <b>P1</b> en stel de waarde in op 0.
3.	Stel de <b>pauze</b> in op 2000 ms.
4.	Stel <b>P0</b> in op <b>P1</b> . Stel de waarde 0 in op 180.
5.	Stel de <b>pauze</b> in op 2000 ms.
6.	Compileer het programma en sla het op als een .hex-bestand. Klik op de knop Download en sla het .hex-bestand op in de map Downloads → C:\downloads. Dit .hex-bestand kan naar de micro:bit geüpload worden. Sluit de micro:bit aan op een USB-poort. Sleep het .hex-bestand naar de micro:bit [removable device] om het programma te uploaden.

U kunt zien dat de servo van 0 tot 180 graden draait. Hoe zou u een klokthermometer met een temperatuursensor en een servo maken?

## 5.9 Zoemer

- 1 1x micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1 x breadboard
- 4 1x mini luidspreker
- 5 1x N-channel MOSFET
- 6 1x 100  $\Omega$  weerstand (bruin/zwart/bruin/goud)



In het voorbeeld hieronder sturen we een zoemer aan.

Plaats de nodige componenten op het breadboard (zie afb.).

Bekijk de onderstaande code.

```

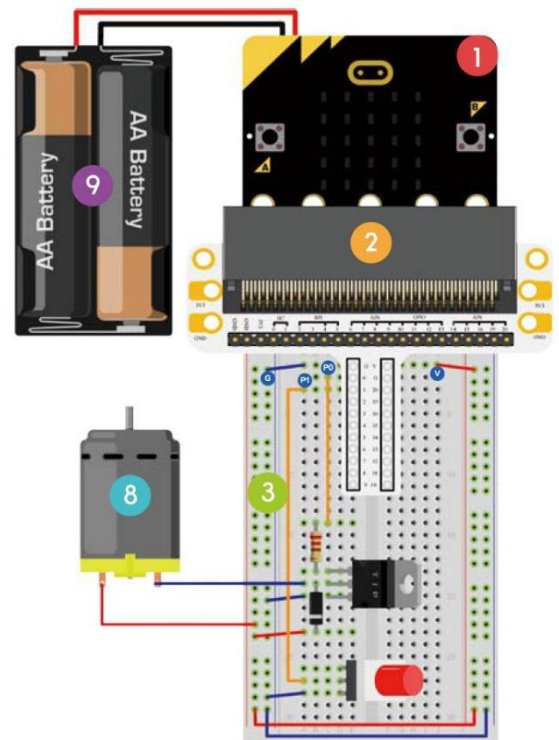
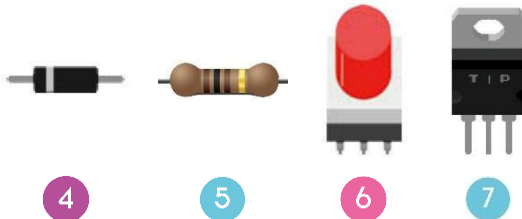
forever
  ring tone (Hz) Middle C
  pause (ms) 100
  ring tone (Hz) Middle E
  pause (ms) 100
  ring tone (Hz) Middle G
  pause (ms) 100
  ring tone (Hz) Middle E
  pause (ms) 100
  
```

1.	<p>Sleep de blokken om de code te maken (zie afb.). De blokken <b>forever</b> en <b>pause (ms) 100</b> bevinden zich in het menu onder <b>Basic</b>. Het blok <b>ring tone (Hz)</b> bevindt zich in het menu onder <b>Music</b>.</p>
2.	<p>Stel het eerste blokje <b>ring tone (Hz)</b> in op <b>Middle C</b>. Klik hiervoor op het tekstveld en selecteer de overeenkomstige pianotoets.</p>  <p>U hoort ook het overeenkomstige geluid.</p>
3.	<p>Stel de <b>pauze</b> in op 100 ms.</p>
4.	<p>Voer de bovenstaande stappen uit voor de resterende belsignalen en pauzes.</p>
5.	<p>Compileer het programma en sla het op als een .hex-bestand. Klik op de knop Download en sla het .hex-bestand op in de map Downloads → C:\downloads. Dit .hex-bestand kan naar de micro:bit geüpload worden. Sluit de micro:bit aan op een USB-poort. Sleep het .hex-bestand naar de micro:bit [removable device] om het programma te uploaden.</p>

Een zoemer weerklinkt. Hoe zou u uw favoriete kinderliedje programmeren?

## 5.10 Motor

- 1 1x micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1 x breadboard
- 4 1x diode
- 5 1x 100  $\Omega$  weerstand (bruin/zwart/bruin/goud)
- 6 1x zelfvergrendelende of bistabiele schakelaar
- 7 1x N-kanaal MOSFET
- 8 1x mini motor
- 9 1x batterijhouder met 2x AA 1.5 V-batterij



In het voorbeeld hieronder gebruiken we een schakelaar om de motor aan te sturen (starten-stoppen).

Plaats de nodige componenten op het breadboard (zie afb.).

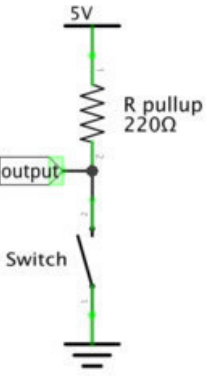
Bekijk de onderstaande code.

```

on start
  digital write pin P0 to 0
  set pull pin P1 to up

forever
  if (digital read pin P1 = 0)
  then
    digital write pin P0 to 1
  else
    digital write pin P0 to 0
  
```

1.	Sleep de blokken om de code te maken (zie afb.).
2.	Het blok <b>on start</b> wordt maar één keer uitgevoerd om het programma te starten.
3.	Stel de waarde voor <b>P0</b> in op <b>0</b> .
4.	Stel het blok <b>pull pin</b> in op <b>P1</b> en <b>up</b> .

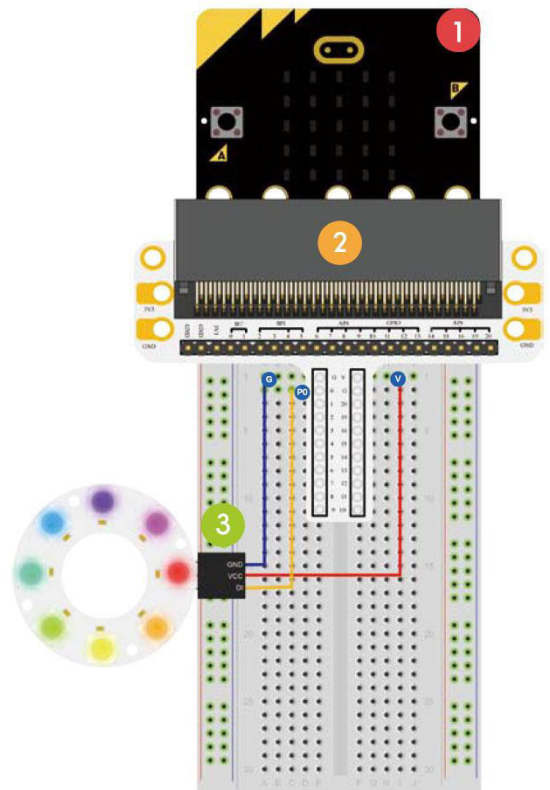
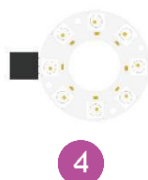
<p><b>5.</b></p>	<p>Stel <b>digital read pin</b> in op <b>P1</b> in het blok <b>forever</b>. Stel de <b>wiskundige functie gelijk aan</b> in op 0.</p> <p>In feite is dit het schema. U moet de pull-up-weerstand niet op het breadboard aansluiten. De pull-up-functie is geprogrammeerd (zie stappen 3-4) in de code en is een weerstand op het breadboard niet nodig.</p> 
<p><b>6.</b></p>	<p>Zodra de schakelaar wordt ingedrukt, stel de hoge spanning in op P0. Stel de waarde voor 0 in op 1 (5 V). De motor begint te werken.</p>
<p><b>7.</b></p>	<p>Zodra de schakelaar wordt losgelaten, stel de lage spanning in op P0. Stel de waarde in op 0. De motor werkt niet meer.</p>
<p><b>8.</b></p>	<p>Compileer het programma en sla het op als een .hex-bestand. Klik op de knop Download en sla het .hex-bestand op in de map Downloads → C:\downloads. Dit .hex-bestand kan naar de micro:bit geüpload worden. Sluit de micro:bit aan op een USB-poort. Sleep het .hex-bestand naar de micro:bit [removable device] om het programma te uploaden.</p>

Druk op de schakelaar om de motor te starten en laat los om te stoppen. Hoe zou u een trimpotentiometer gebruiken om het motortoerental te regelen?

Opmerking: De micro:bit geeft slechts 3.3 V af, wat mogelijk onvoldoende is om de ventilator te gebruiken. Om de ventilator te starten kan het nodig zijn om het mes te draaien.

### 5.11 Rainbow led

- 1 1x micro:bit board
- 2 1x micro:bit breadboard adapter
- 3 1 x breadboard
- 4 1x RGB-ledring



In het voorbeeld hieronder sturen we acht RGB-leds in een ledring aan en creëren een regenboogeffect.

Plaats de nodige componenten op het breadboard (zie afb.).



Bekijk de onderstaande code.

```

on start
  set strip to NeoPixel at pin P0 with 8 leds as RGB (GRB format)
  strip show rainbow from 1 to 360

forever
  strip show
  strip rotate pixels by 1
  pause (ms) 100
  
```

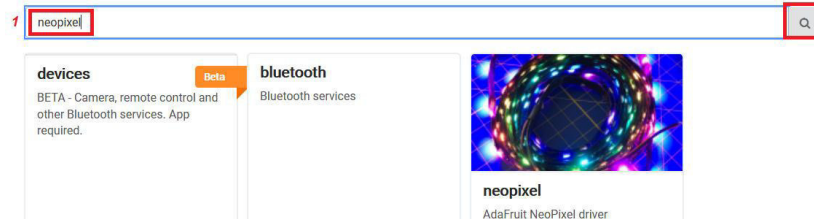
1. Sleep de blokken om de code te maken (zie afb.).

2. Zoek en voeg de NeoPixel-bibliotheek toe.

Ga naar Advanced → Add Package (Extensions) en geef NeoPixel in.

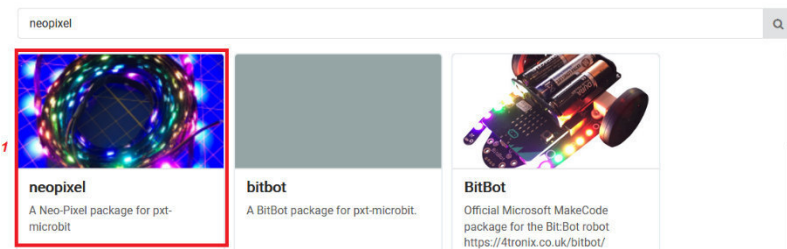


Add Package... ?

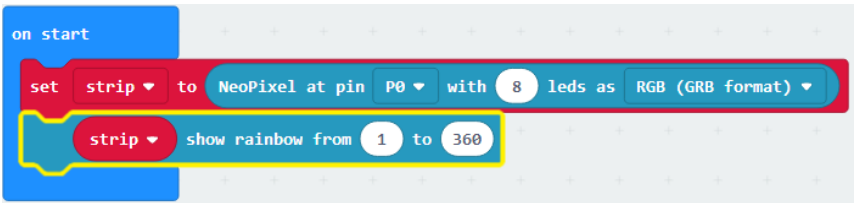


Selecteer de NeoPixel-bibliotheek.

Add Package... ?



De bibliotheek is gedownload en aan het menu toegevoegd.

	 <p>De blokken <b>NeoPixel at pin P0 with 24 leds</b>, <b>item show rainbow from 1 to 360</b>, <b>item show</b> and <b>items rotate pixels by 1</b> bevinden zich in het menu onder <b>NeoPixel</b>.</p>
3.	<p>Sleep het blok <b>set strip to NeoPixel at pin P0 with 24 leds as RGB (GRB format)</b> in het blok <b>on start</b>. Stel de pin in op <b>P0</b> en verander de waarde 24 naar 8 (8 leds op de ledring).</p> 
4.	<p>Sleep het blok <b>set strip to rainbow from 1 to 360</b> onder het blok <b>set strip to NeoPixel at pin P0</b>.</p> 
5.	<p>Vervolledig de code volgens het voorbeeld.</p>
6.	<p>Compileer het programma en sla het op als een .hex-bestand. Klik op de knop Download en sla het .hex-bestand op in de map Downloads → C:\downloads. Dit .hex-bestand kan naar de micro:bit geüpload worden. Sluit de micro:bit aan op een USB-poort. Sleep het .hex-bestand naar de micro:bit [removable device] om het programma te uploaden.</p>

De leds creëren een regenboogeffect op de ledring. Hoe zou u de ring nu als een oog laten knipperen?

**Gebruik dit toestel enkel met originele accessoires. Velleman nv is niet aansprakelijk voor schade of kwetsuren bij (verkeerd) gebruik van dit toestel. Voor meer informatie over dit product en de laatste versie van deze handleiding, zie [www.velleman.eu](http://www.velleman.eu). De informatie in deze handleiding kan te allen tijde worden gewijzigd zonder voorafgaande kennisgeving.**

#### © AUTEURSRECHT

**Velleman nv heeft het auteursrecht voor deze handleiding. Alle wereldwijde rechten voorbehouden.** Het is niet toegestaan om deze handleiding of gedeelten ervan over te nemen, te kopiëren, te vertalen, te bewerken en op te slaan op een elektronisch medium zonder voorafgaande schriftelijke toestemming van de rechthebbende.

# MODE D'EMPLOI

## 1. Introduction

### Aux résidents de l'Union européenne

#### Informations environnementales importantes concernant ce produit



Ce symbole sur l'appareil ou l'emballage indique que l'élimination d'un appareil en fin de vie peut polluer l'environnement. Ne pas jeter un appareil électrique ou électronique (et des piles éventuelles) parmi les déchets municipaux non sujets au tri sélectif ; une déchetterie traitera l'appareil en question. Renvoyer l'appareil à votre fournisseur ou à un service de recyclage local.

Respecter la réglementation locale relative à la protection de l'environnement.

#### En cas de questions, contacter les autorités locales pour élimination.

Nous vous remercions de votre achat ! Lire attentivement le présent mode d'emploi avant la mise en service de l'appareil. Si l'appareil a été endommagé pendant le transport, ne pas l'installer et consulter votre revendeur.

## 2. Consignes de sécurité



Cet appareil peut être utilisé par des enfants âgés de 8 ans et plus et des personnes manquant d'expérience et de connaissances ou dont les capacités physiques, sensorielles ou mentales sont réduites, si elles ont été formées et encadrées quant à l'utilisation de l'appareil d'une manière sûre et connaissent les risques encourus. Ne pas laisser les enfants jouer avec l'appareil. Le nettoyage et l'entretien ne doivent pas être effectués par des enfants sans surveillance.



Utiliser cet appareil uniquement à l'intérieur.

Protéger l'appareil contre la pluie, l'humidité et les éclaboussures.

## 3. Directives générales



- Se référer à la garantie de service et de qualité Velleman® en fin de ce mode d'emploi.
- Se familiariser avec le fonctionnement de l'appareil avant de l'utiliser.
- Toute modification est interdite pour des raisons de sécurité. Les dommages occasionnés par des modifications par le client ne tombent pas sous la garantie.
- N'utiliser l'appareil qu'à sa fonction prévue. Un usage impropre annule d'office la garantie.
- La garantie ne se s'applique pas aux dommages survenus en négligeant certaines directives de ce mode d'emploi et votre revendeur déclinera toute responsabilité pour les problèmes et les défauts qui en résultent.
- Ni Velleman SA ni ses distributeurs ne peuvent être tenus responsables des dommages exceptionnels, imprévus ou indirects, quelles que soient la nature (financière, corporelle, etc.), causés par la possession, l'utilisation ou le dysfonctionnement de ce produit.
- Dans le souci d'une amélioration constante de nos produits, l'apparence du produit peut différer légèrement des images affichées.
- Les images des produits sont à titre indicatif seulement.
- Ne pas brancher immédiatement l'appareil après exposition à des variations de température. Afin d'éviter des dommages, attendre jusqu'à ce que l'appareil ait atteint la température ambiante.
- Garder ce mode d'emploi pour toute référence ultérieure.

## 4. Description

Ce kit de démarrage est un kit éducatif basé sur micro:bit. Le kit comporte des composants de base électroniques, une platine d'expérimentation, des câbles de connexion et un micro:bit.

## 5. Exemples

### 5.1 LED

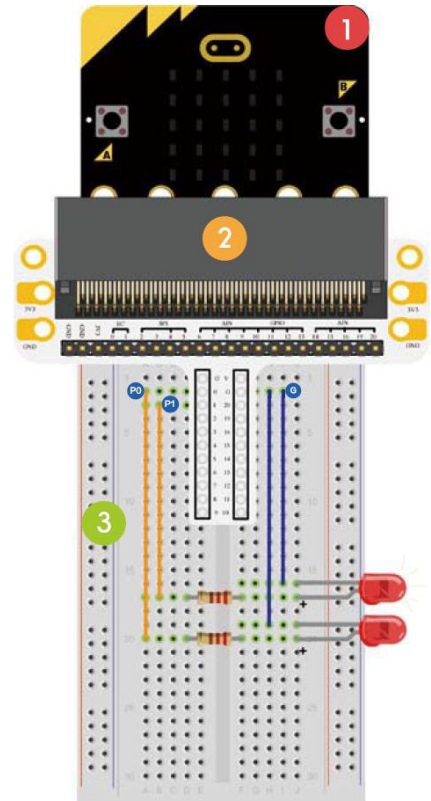
- 1 1x carte micro:bit
- 2 1x adaptateur pour platine d'expérimentation micro:bit
- 3 1x platine d'expérimentation
- 4 2x LED rouge (polarisation : anode (+) = patte longue, cathode (-) = patte courte)
- 5 1x résistance 100  $\Omega$  (marron/noir/marron/or)



4



5



Dans ce tutoriel, nous utiliserons le micro:bit pour faire clignoter deux LED en alternance.

Poser les composants requis sur la platine d'expérimentation (voir ill.).

Le programme est écrit en blocs, en ligne dans le navigateur Web. Accéder au site [www.makecode.com](http://www.makecode.com) ou <https://www.microsoft.com/en-us/makecode?rtc=1> et cliquer sur micro:bit et ensuite sur **Start Project**.

Microsoft MakeCode brings computer science to life for all students with fun projects, immediate results, and both block and text editors for learners at different levels.

**micro:bit**  
Start coding with micro:bit >

**Circuit Playground Express**  
Start coding with Circuit Playground Express >

**Minecraft**  
Start coding with Minecraft >

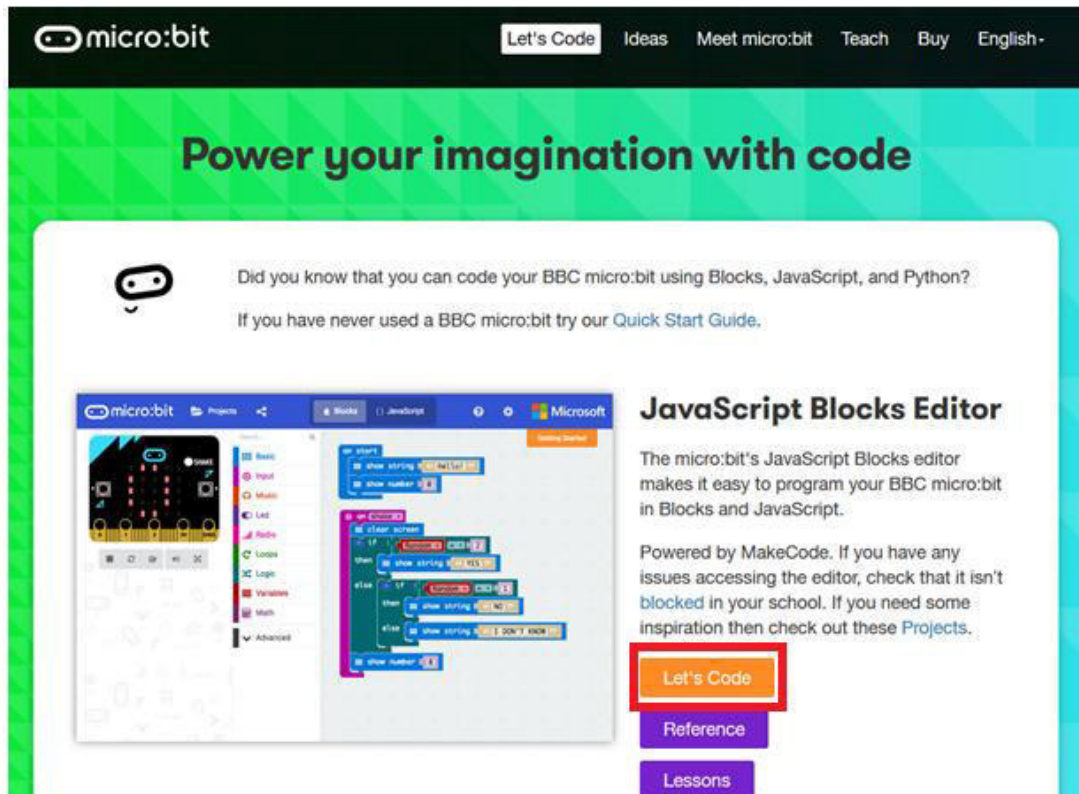
Une nouvelle fenêtre s'ouvre. Vous pouvez commencer à composer le code avec les blocs à glisser-déposer depuis le menu vers l'éditeur de code.

Lire ci-dessous comment cela fonctionne...

## Qu'est-ce que MakeCode ?

(Auparavant PXT – Programming eXperience Toolkit).

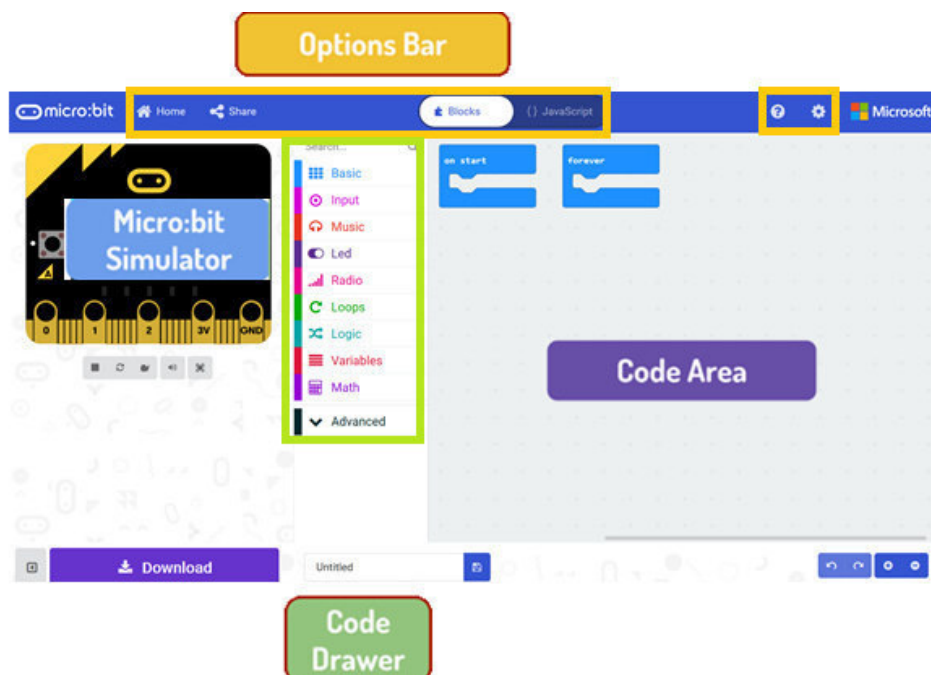
Un éditeur de code graphique par glisser-déposer convivial pour débutants, similaire à Scratch, qui fonctionne en ligne dans le navigateur.



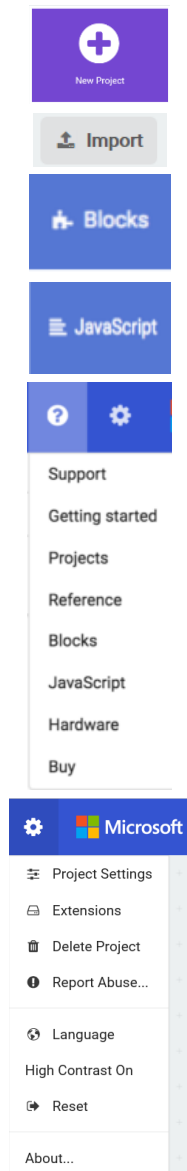
L'éditeur se compose en différent volets : Le code est créé dans la **zone de code**, en faisant glisser des blocs depuis le **menu**. Le résultat du code s'affiche immédiatement dans le **simulateur micro:bit**. En bas se trouve un bouton pour télécharger et sauvegarder le projet.



Regardons tout d'abord comment utiliser la **Options Bar/Barre d'options**.



## Barre d'options



Cliquer pour créer ou ajouter un nouveau projet.

Cliquer sur Import pour importer vos projets.

Cliquer sur Blocs pour écrire votre script avec l'éditeur Block.

Cliquer sur Javascript pour écrire votre script en JavaScript.

Cliquer sur le point d'interrogation pour accéder au menu d'aide.

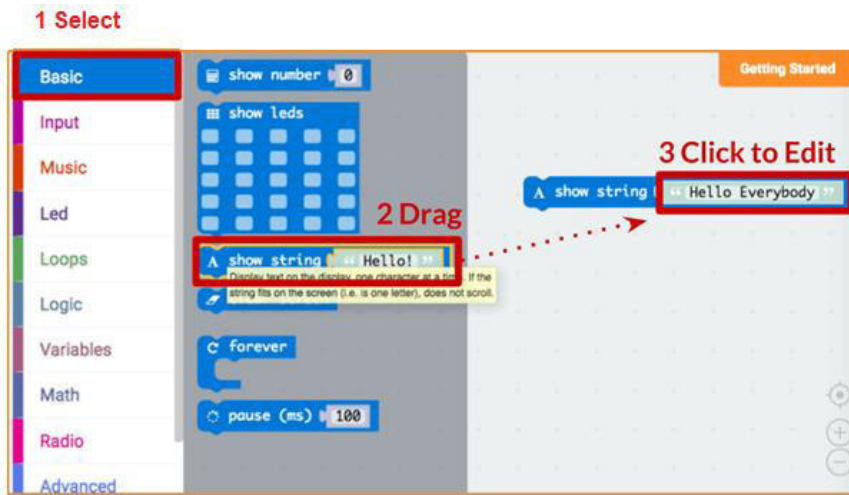
Cliquer sur l'icône d'engrenage pour modifier les paramètres du projet. Depuis ce menu, il est possible de renommer et supprimer vos projets. **Reset/Remise à zéro** supprimera tous les projets enregistrés. Faire attention. Il est généralement préférable d'utiliser **Supprimer le projet**.

## Utiliser la matrice LED

Commençons par écrire le code ! Chaque fois que vous sélectionnez un bloc depuis le menu, une liste avec les blocs disponibles s'affiche. Glisser-déposer le bloc **show string** dans l'espace de travail et cliquer dans le bloc pour modifier.

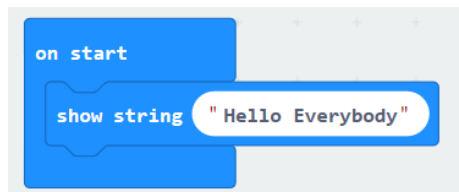






<b>1</b>	Sélectionner
<b>2</b>	Glisser-déposer

<b>3</b>	Cliquer pour modifier
----------	-----------------------

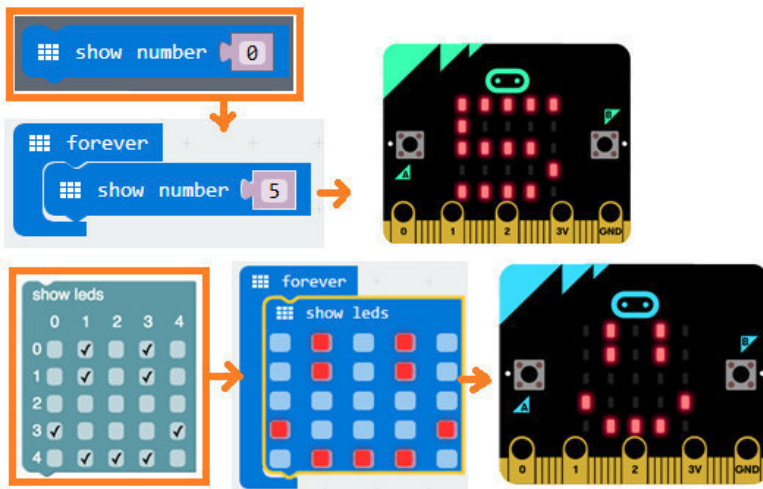


Tester le code dans le simulateur !

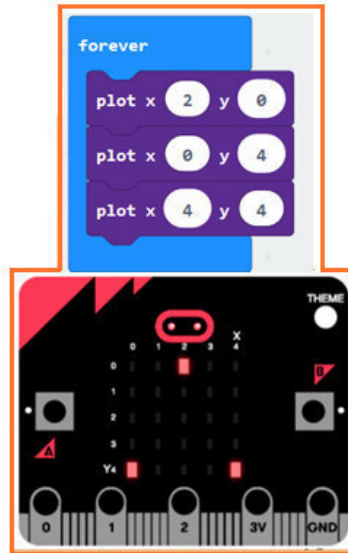
### Say Hello!

Les caractères affichés sont des **chaînes de caractères**.

Vous pouvez afficher du texte et des numéros en utilisant la fonction **show number** ou une image de 5x5 pixels avec **show leds**.



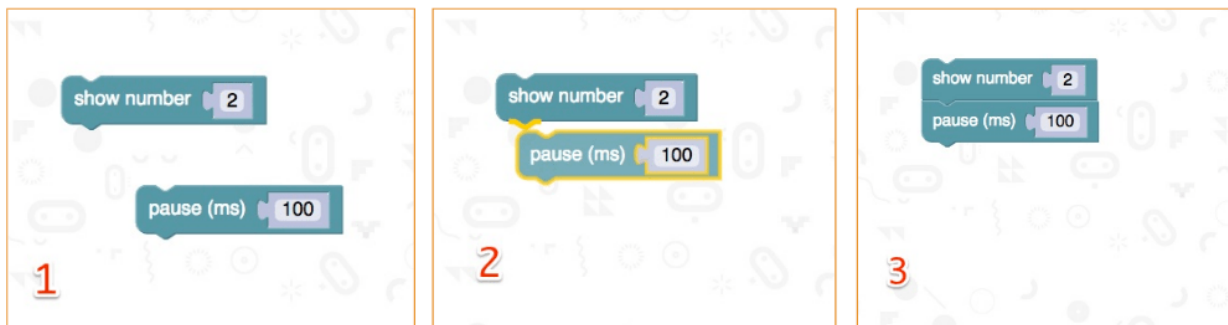
Vous pouvez également allumer et éteindre une seule LED en utilisant des coordonnées. Les coordonnées (0,0) se trouvent en haut à gauche de l'écran.



### Assembler les blocs

Cliquer et maintenir enfoncé le bouton gauche de la souris pour faire glisser le bloc. Faire glisser le bloc vers le bloc cible jusqu'à ce que le bord du bloc s'illumine. Déposer le bloc et les deux blocs s'emboîtent !

Cliquer sur le premier bloc pour déplacer le deuxième bloc. Cliquer sur le deuxième bloc pour le décomposer du premier bloc.



### Tester sur le micro:bit

Connecter le micro:bit à votre PC avec un câble micro-USB.

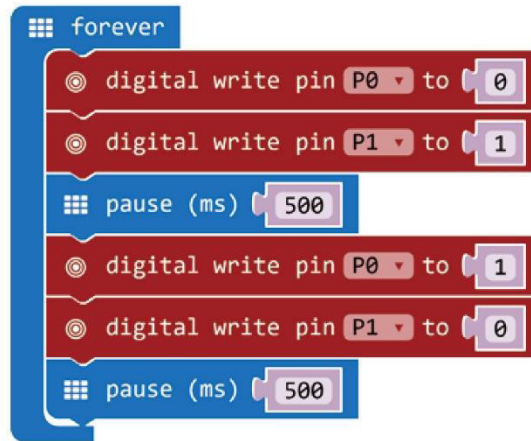
Une routine de test s'affiche sur le micro:bit, vous demandant d'appuyer sur des boutons et de jouer un jeu. Tester le !

Cliquer sur **Download** pour convertir le script en un **fichier .hex** et le télécharger.

Si Chrome est configuré correctement, vous pouvez ignorer l'étape suivante. Glisser-déposer le fichier .hex sur le lecteur micro:bit ou cliquer droit et sélectionner **Send To** dans Windows®.

Vous venez de programmer votre propre appareil !

Regarder le code ci-dessous.

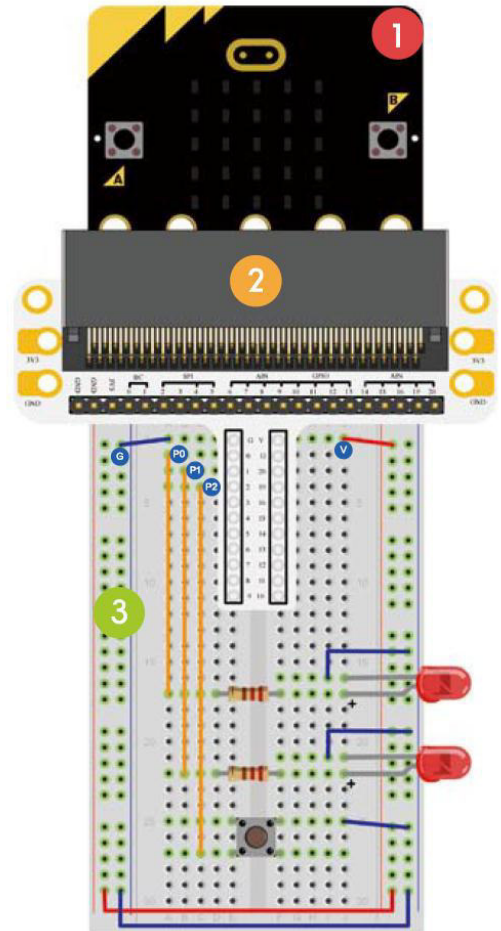
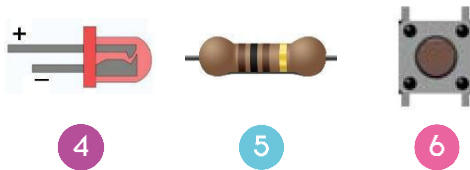


1.	Glisser-déposer les blocs pour former le code comme indiqué. Le bloc <b>digital write pin P0 to 0</b> se trouve dans le menu sous <b>Advanced</b> → <b>Pins</b> . Les blocs <b>forever</b> et <b>pause (ms) 100</b> se trouvent dans le menu sous <b>Basic</b> .
2.	Régler la valeur pour P0 sur 0. LED0 off = low voltage = 0 V = digital 0. Régler la valeur pour P1 sur 1. LED1 on = high voltage = 5 V = digital 1.
3.	Régler la <b>pause</b> sur 500 ms.
4.	Régler la valeur pour P0 sur 1. LED0 on = high voltage = 5 V = digital 1. Régler la valeur pour P1 sur 0. LED1 off = low voltage = 0 V = digital 0.
5.	Régler la <b>pause</b> sur 500 ms.
6.	Une fois terminé, compiler le programme pour générer un fichier .hex. Cliquer sur le bouton Download/Télécharger et sauvegarder le fichier .hex dans le dossier Downloads → C:\downloads. Ce fichier .hex peut être téléversé vers le micro:bit. Connecter le micro:bit à un port USB. Glisser-déposer le fichier .hex sur le micro:bit [Removable device] pour téléverser le programme.

Deux LED clignoteront alternativement. Et maintenant, pourquoi ne pas créer un feu de signalisation RGB ?

## 5.2 Bouton

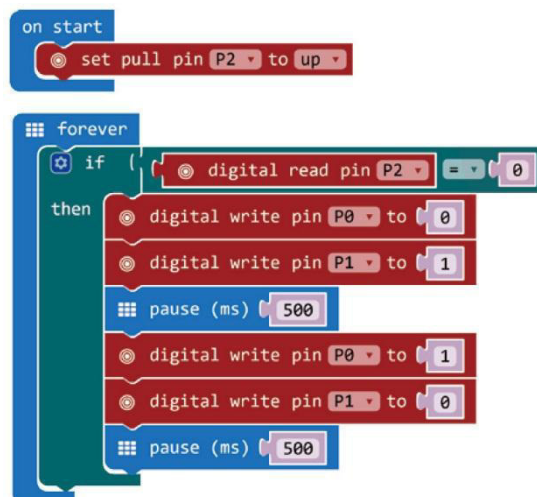
- 1 1x carte micro:bit
- 2 1x adaptateur pour platine d'expérimentation micro:bit
- 3 1x platine d'expérimentation
- 4 2x LED rouge (polarisation : anode (+) = patte longue, cathode (-) = patte courte)
- 5 1x résistance 100 Ω (marron/noir/marron/or)
- 6 1x interrupteur à action momentanée



Nous utilisons un bouton pour contrôler le clignotement de la LED. Appuyer sur le bouton pour faire clignoter la LED en alternance. Relâcher pour éteindre la LED.

Poser les composants requis sur la platine d'expérimentation (voir ill.)

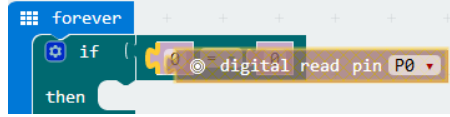
Regarder le code ci-dessous.



1. Glisser-déposer les blocs pour former le code comme indiqué. Les blocs **digital read pin P0** et **digital write pin P0 to 0** se trouvent dans le menu sous **Advanced** → **Pins**. Les blocs **forever** et **pause (ms) 100** se trouvent dans le menu sous **Basic**. Les blocs **if then** et **logic equal function (=)** se trouvent dans le menu sous **Logic**. Faire glisser le bloc **logic equal function** et déposer le sur le bloc **true**.



Les deux blocs s'emboîtent.

	Faire glisser le bloc <b>digital read pin P0</b> et déposer le sur le <b>0</b> du bloc de la fonction <b>logic equal function (=)</b> (voir ill.). 
<b>2.</b>	Configurer P2 comme une résistance pull-up.
<b>3.</b>	Régler <b>digital read pin</b> sur P2.
<b>4.</b>	Régler <b>digital write pin P0 sur 0</b> (actif bas (0 V)). Régler <b>digital write pin P1 sur 1</b> (actif haut (5 V)).
<b>5.</b>	Régler la <b>pause</b> sur 500 ms.
<b>6.</b>	Régler <b>digital write pin P0 sur 1</b> (actif haut (5 V)). Régler <b>digital write pin P1 sur 0</b> (actif bas (0 V)).
<b>7.</b>	Régler la <b>pause</b> sur 500 ms.
<b>8.</b>	Une fois terminé, compiler le programme pour générer un fichier .hex. Cliquer sur le bouton Download/Télécharger et sauvegarder le fichier .hex dans le dossier Downloads → C:\downloads. Ce fichier .hex peut être téléversé vers le micro:bit.  Connecter le micro:bit à un port USB. Glisser-déposer le fichier .hex sur le micro:bit [Removable device] pour téléverser le programme.

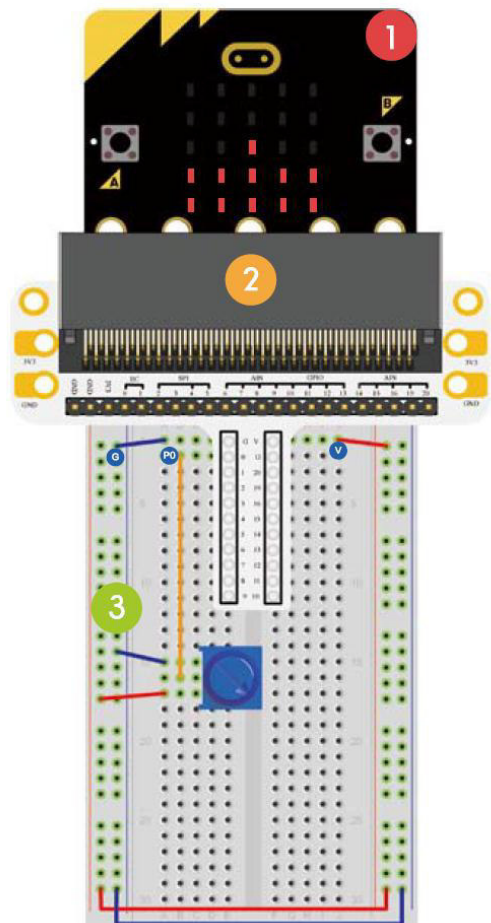
Appuyer sur le bouton pour faire clignoter la LED en alternance. Comment allumer la LED rouge en appuyant sur le bouton et allumer la LED verte en lâchant le bouton ?

### 5.3 Potentiomètre ajustable

- 1 1x carte micro:bit
- 2 1x adaptateur pour platine d'expérimentation micro:bit
- 3 1x platine d'expérimentation
- 4 1x potentiomètre ajustable 10 kΩ



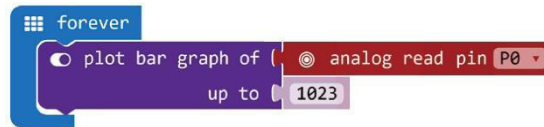
4



Dans l'exemple ci-dessous, nous lirons la tension de sortie du potentiomètre et la visualiserons sur l'écran du micro-bit par un graphique à barres.

Poser les composants requis sur la platine d'expérimentation (voir ill.)

Regarder le code ci-dessous.



1.	Glisser-déposer les blocs pour former le code comme indiqué. Le bloc <b>plot bar graph of 0 up to 0</b> se trouve dans le menu sous <b>LED</b> .
2.	Régler <b>analog read pin</b> sur P0. Ce bloc se trouve dans le menu sous <b>Pins</b> . Régler la valeur sur 1023 max.
3.	Une fois terminé, compiler le programme pour générer un fichier .hex. Cliquez sur le bouton Download/Télécharger et sauvegardez le fichier .hex dans le dossier Downloads → C:\downloads. Ce fichier .hex peut être téléversé vers le micro:bit. Connecter le micro:bit à un port USB. Glisser-déposer le fichier .hex sur le micro:bit [Removable device] pour téléverser le programme.

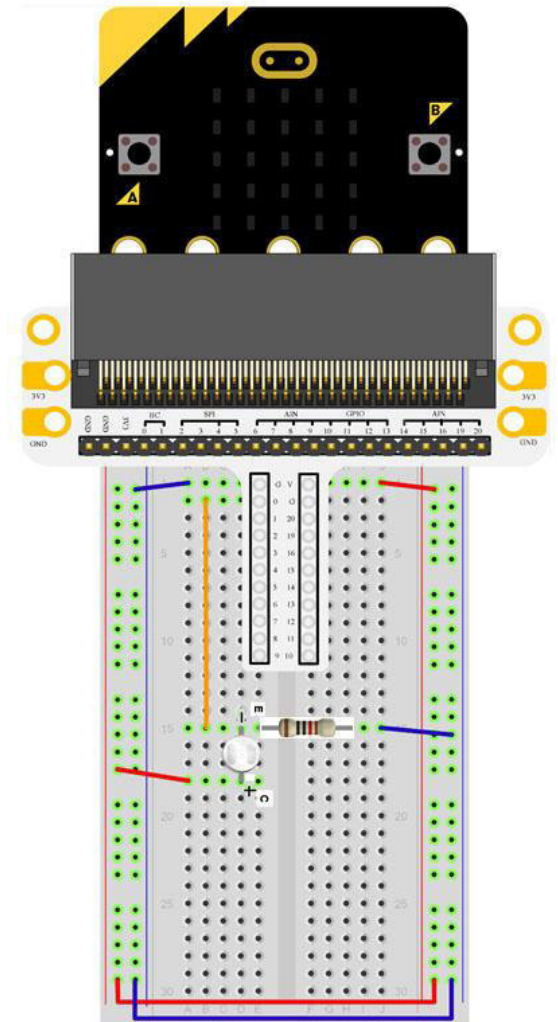
Faire tourner le potentiomètre ajustable. La tension est affichée sur l'écran micro:bit par un graphique à barres. Lorsque la tension est égale à 0, un seul pixel apparaîtra à l'écran LED. Quand la tension est de 3.3 V, tout l'écran s'illuminera. Comment utiliser le potentiomètre ajustable pour régler l'intensité lumineuse de la LED ?

## 5.4 Photocellule

- 1 1x carte micro:bit
- 2 1x adaptateur pour platine d'expérimentation micro:bit
- 3 1x platine d'expérimentation
- 4 1x photocellule
- 5 1x résistance 10 kΩ (marron/noir/marron/rouge/marron)



Patte courte =  
collecteur  
Anode positive  
Patte longue =  
émetteur  
Cathode négative

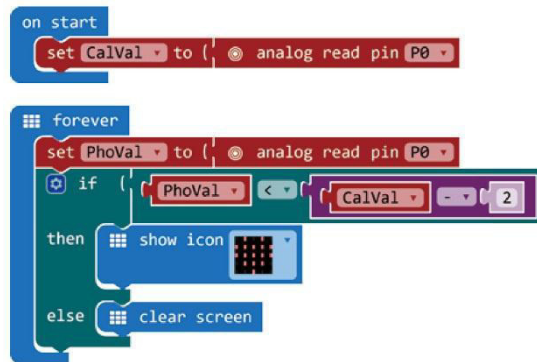


Dans l'exemple ci-dessous, nous utiliserons une photocellule pour contrôler la luminosité de l'écran micro:bit.

Poser les composants requis sur la platine d'expérimentation (voir ill.)



Regarder le code ci-dessous.



1. Tout d'abord, créons deux variables. Aller à **Variables** dans le menu et cliquer sur **Make a Variable**.

New variable name:

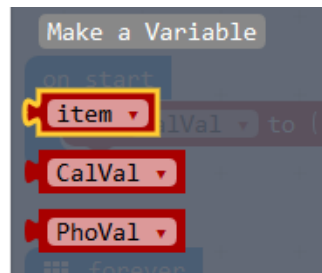
CalVal

Ok ✓

Cancel ✕

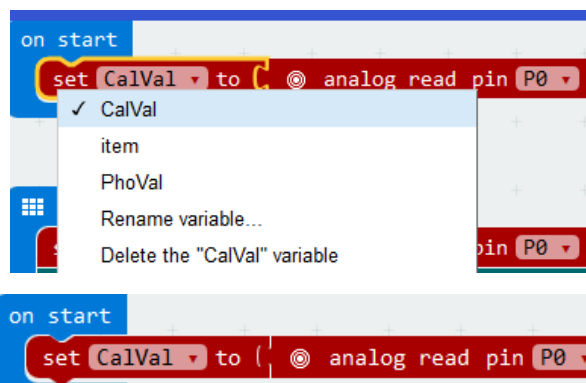
Entrer **CalVal** dans la case et cliquer sur **Ok**. Entrer **PhoVal** dans la case et cliquer sur **Ok**.

Deux nouvelles variables s'affichent dans le menu sous **Variables**.



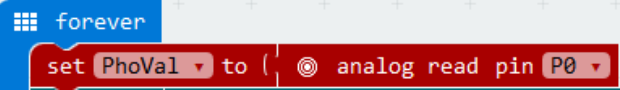
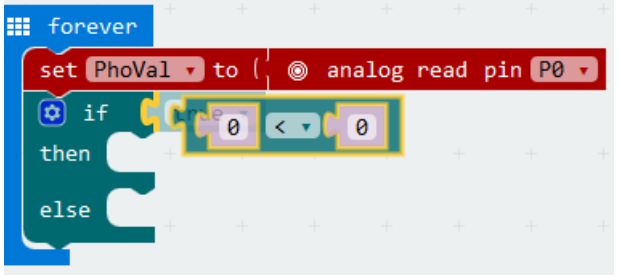
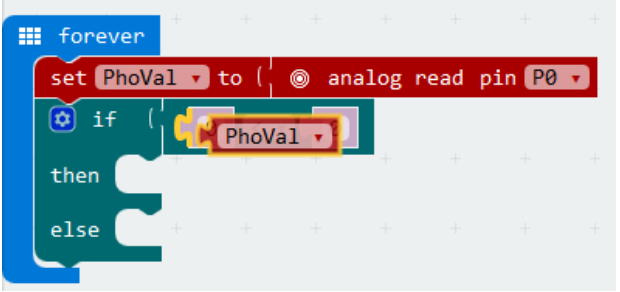
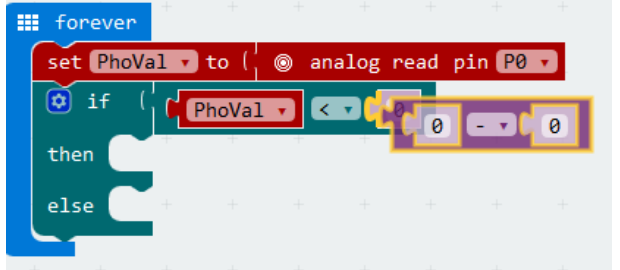
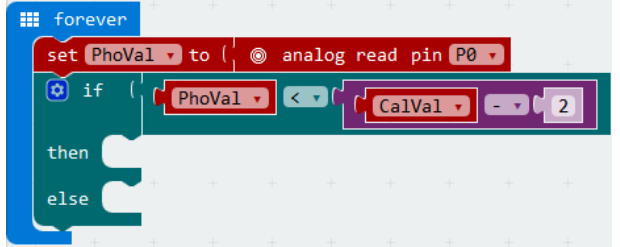
Nous aurons besoin de ces variables plus tard pour sauvegarder les données dans un registre.

Les blocs **logic smaller than function** et **if then else** se trouvent dans le menu sous **Logic**. Le bloc **mathematical function minus (-)** se trouve dans le menu sous **Math**. Le bloc **show icon** se trouve dans le menu sous block in **Basic** in the Code Drawer. Le bloc **clear screen** se trouve dans le menu sous **Basic** → **More**. Le bloc **set item to 0** se trouve dans le menu sous **Variables**. Cliquer sur la flèche et sélectionner **CalVal** or **PhoVal**.



2. Sélectionner la variable **CalVal** et régler **analog read pin** sur P0.



3.	<p>Dans le bloc <b>forever</b>, sélectionner la variable <b>PhoVal</b> et régler le bloc <b>analog read pin</b> sur P0.</p> 
4.	<p>Faire glisser la fonction <b>smaller than (&lt;)</b> à côté du bloc <b>if</b> et déposer le sur le bloc <b>true</b>.</p>  <p>Faire glisser la variable <b>PhoVal</b> (depuis le menu sous <b>Variables</b>) et déposer la sur le première 0 du bloc <b>logic smaller than function (&lt;)</b>.</p>  <p>Faire glisser le bloc <b>mathematical function minus (-)</b> et déposer le sur le deuxième 0 du bloc <b>logic smaller than function (&lt;)</b>.</p>  <p>Faire glisser la variable <b>CalVal</b> (depuis le menu sous <b>Variables</b>) et déposer le sur le premier 0 du bloc <b>mathematical function minus (-)</b>. Régler le deuxième 0 sur le bloc <b>mathematical function minus (-)</b> sur 2.</p> 
5.	<p>Glisser-déposer le bloc <b>show icon</b> à côté du bloc <b>then</b>. Glisser-déposer le bloc <b>clear screen</b> à côté du bloc <b>else</b>.</p>
6.	<p>Lorsque <b>PhoVal</b> est inférieur à <b>CalVal - 2</b>, l'écran affichera un cœur. Sinon, l'écran s'éteindra.</p>

- 7.** Une fois terminé, compiler le programme pour générer un fichier .hex. Cliquer sur le bouton Download/Télécharger et sauvegarder le fichier .hex dans le dossier Downloads → C:\downloads. Ce fichier .hex peut être téléversé vers le micro:bit.
- Connecter le micro:bit à un port USB. Glisser-déposer le fichier .hex sur le micro:bit [Removable device] pour téléverser le programme.

Note : réinitialiser le micro:bit pour calibrer la valeur de référence en fonction de la luminosité actuelle. Pour exécuter le programme correctement, commençons avec la lumière allumée.

Lorsque la lumière est allumée, rien ne s'affiche. Lorsque la lumière est éteinte, le cœur apparaît. Comment utiliser une photodiode pour contrôler une LED ?

## 5.5 LED RGB

- 1 1x carte micro:bit
- 2 1x adaptateur pour platine d'expérimentation micro:bit
- 3 1x platine d'expérimentation
- 4 1x LED RGB (cathode commune)
- 5 1x résistance 100 Ω (marron/noir/marron/or)

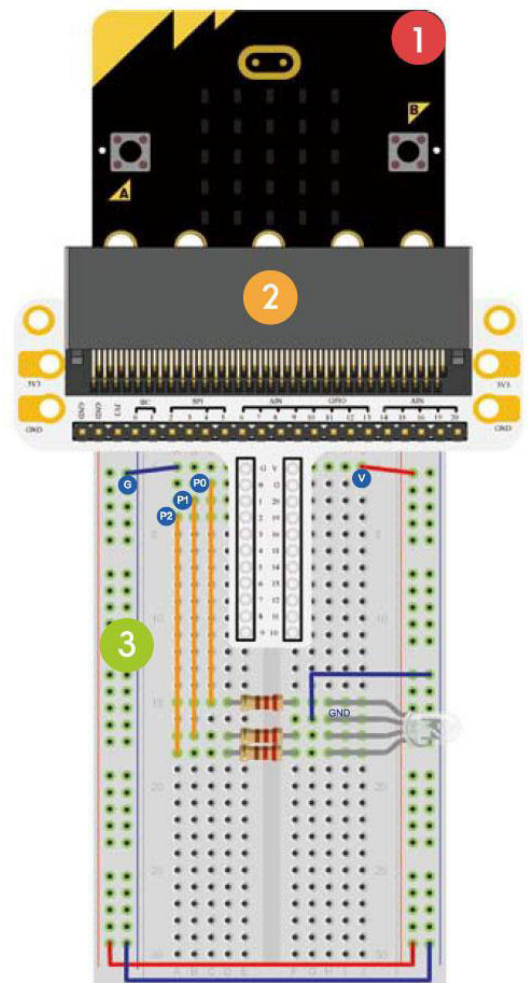


4

1. Rouge (anode +)
2. Terre (cathode -) – la patte la plus longue
3. Vert (anode +)
4. Bleu (anode +)



5



Dans l'exemple ci-dessous, nous programmerons une transition de LED RGB aisée.

Poser les composants requis sur la platine d'expérimentation (voir ill.)

Regarder le code ci-dessous.

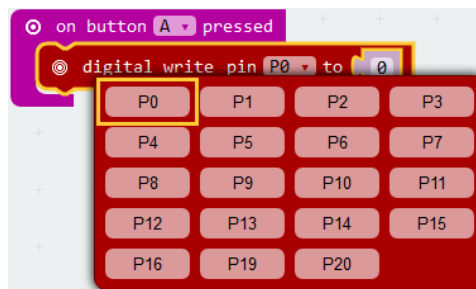
```

on button A pressed
  digital write pin P0 to 1
  digital write pin P1 to 0
  digital write pin P2 to 0

on button B pressed
  digital write pin P0 to 0
  digital write pin P1 to 1
  digital write pin P2 to 0

on button A+B pressed
  digital write pin P0 to 0
  digital write pin P1 to 0
  digital write pin P2 to 1
  
```

- Glisser-déposer les blocs pour former le code comme indiqué.  
 Le bloc **on button A pressed** se trouve dans le menu sous **Input**. Le bloc **digital write pin P0 to 0** se trouve dans le menu sous **Pins**.  
 Sélectionner l'option A dans le bloc **on button A pressed**.  
 Faire glisser 3 blocs **digital write pin P0 to 0** et déposer les dans le bloc **on button A pressed**.  
 Régler la broche P0 dans le premier bloc **digital write pin P0 sur 0** et régler la valeur 0 sur 1 (LED rouge allumée).  
 Régler la broche P1 dans le premier bloc **digital write pin P0 sur 0** et régler la valeur sur 0 (LED verte éteinte).  
 Régler la broche P2 dans le bloc **digital write pin P0 sur 0** et régler la valeur sur 0 (LED bleue éteinte).



```

on button A pressed
  digital write pin P0 to 1
  digital write pin P1 to 0
  digital write pin P2 to 0
  
```

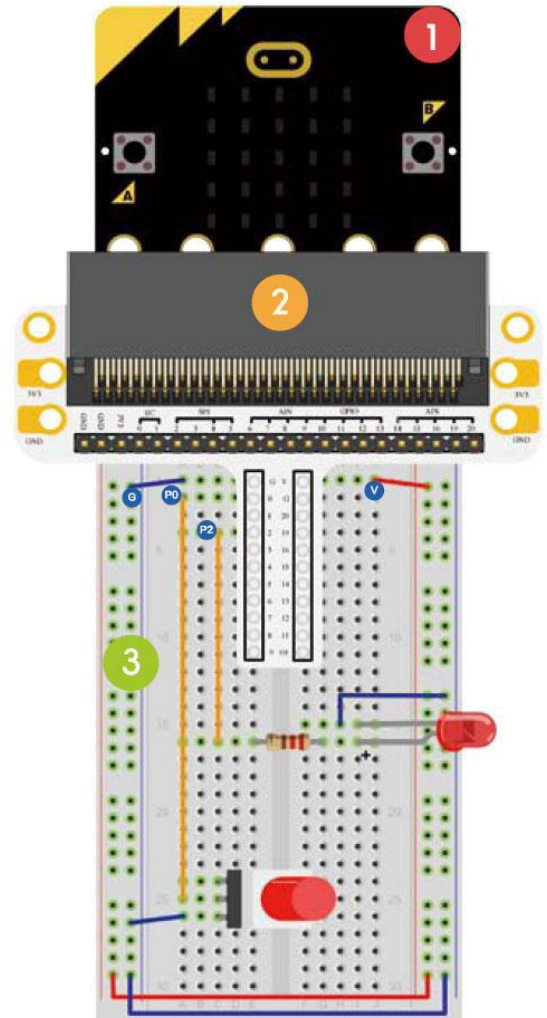
- Compiler les deux blocs **on button B/A+B pressed** pour former le code.

- 3.** Une fois terminé, compiler le programme pour générer un fichier .hex. Cliquer sur le bouton Download/Télécharger et sauvegarder le fichier .hex dans le dossier Downloads → C:\downloads. Ce fichier .hex peut être téléversé vers le micro:bit.
- Connecter le micro:bit à un port USB. Glisser-déposer le fichier .hex sur le micro:bit [Removable device] pour téléverser le programme.

Appuyer sur bouton A pour allumer la LED rouge, appuyer sur bouton B pour allumer la LED verte, appuyer simultanément sur les boutons A et B pour allumer la LED bleue. Comment réaliser une transition RGB aisée ?

## 5.6 Commutateur autobloquant

- 1 1x carte micro:bit
- 2 1x adaptateur pour platine d'expérimentation micro:bit
- 3 1x platine d'expérimentation
- 4 1x résistance 100 Ω (marron/noir/marron/or)
- 5 1x LED rouge (polarisation : anode (+) = patte longue, cathode (-) = patte courte)
- 6 1x commutateur autobloquant ou bistable



Dans l'exemple ci-dessous, nous lirons la température ambiante (données) du capteur de température analogique et affichons les données sur le micro:bit.

Poser les composants requis sur la platine d'expérimentation (voir ill.)

Regarder le code ci-dessous.

```

on start
  set pin P0 to emit edge events
  set pull pin P0 to up

on event from MICROBIT_ID_IO_P0 with value MICROBIT_PIN_EVT_FALL
  digital write pin P2 to 1

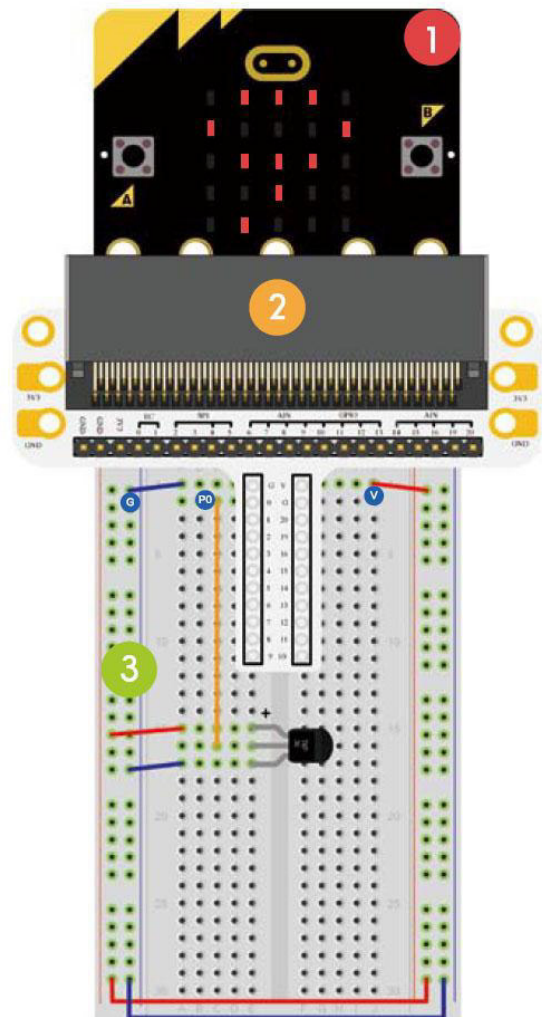
on event from MICROBIT_ID_IO_P0 with value MICROBIT_PIN_EVT_RISE
  digital write pin P2 to 0
  
```

1.	Glisser-déposer les blocs pour former le code comme indiqué. Les blocs <b>set pin P0 to emit edge events</b> et <b>set pull P0 to up</b> se trouvent dans le menu sous <b>Pins</b> → <b>More</b> . Les blocs <b>on event from MICROBIT...EVT_FALL/RISE</b> se trouvent dans le menu sous <b>Control</b> .
2.	Régler le type événement sur <b>edge</b> . Régler le bloc pull pin sur <b>P0</b> et sur <b>up</b> .
3.	Créer un bloc événement. Glisser-déposer le bloc <b>on event from MICROBIT...EVT°FALL</b> sous le bloc <b>on start</b> . Cliquer sur la première flèche et sélectionner l'option <b>MICROBIT_ID_IO_P0</b> dans le menu déroulant. Cliquer sur la deuxième flèche et sélectionner l'option <b>MICROBIT_PIN_EVENT_FALL</b> dans le menu déroulant. Glisser-déposer le bloc <b>digital write pin P0 to 0</b> dans le bloc événement. Régler la broche P0 sur P2 et régler 0 sur 1. Le bloc <b>digital write pin P0 to 0</b> se trouve dans le menu sous <b>Pins</b> .
4.	Procéder de la même manière pour le deuxième bloc événement. Cliquer sur la deuxième flèche et sélectionner l'option <b>MICROBIT_PIN_EVENT_RISE</b> dans le menu déroulant. Régler P0 sur P2 pour le bloc <b>digital write pin P0 to 0</b> et régler la valeur sur 0.
5.	Une fois terminé, compiler le programme pour générer un fichier .hex. Cliquer sur le bouton Download/Télécharger et sauvegarder le fichier .hex dans le dossier Downloads → C:\downloads. Ce fichier .hex peut être téléversé vers le micro:bit. Connecter le micro:bit à un port USB. Glisser-déposer le fichier .hex sur le micro:bit [Removable device] pour téléverser le programme.

Appuyer sur le commutateur autobloquant pour allumer la LED. Appuyer de nouveau pour éteindre la LED. Comment contrôler l'écran micro:bit avec ce commutateur autobloquant ?

## 5.7 Capteur de température

- 1 1x carte micro:bit
- 2 1x adaptateur pour platine d'expérimentation micro:bit
- 3 1x platine d'expérimentation
- 4 1x capteur de température TMP36

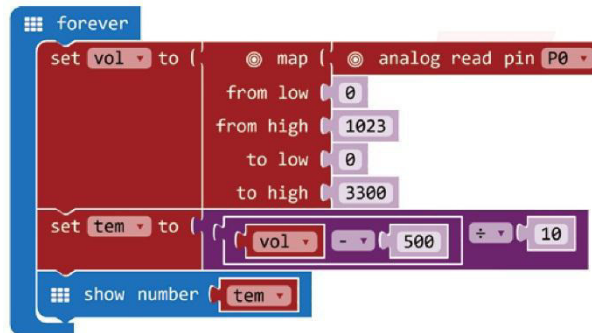


Dans l'exemple ci-dessous, nous lisons la température ambiante (données) du capteur de température analogique et affichons les données sur le micro:bit.

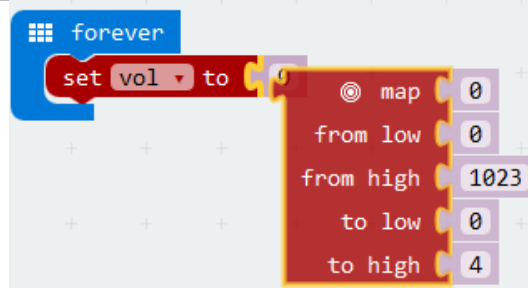
Poser les composants requis sur la platine d'expérimentation (voir ill.)



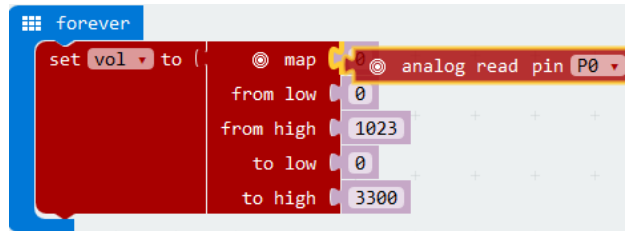
Regarder le code ci-dessous.



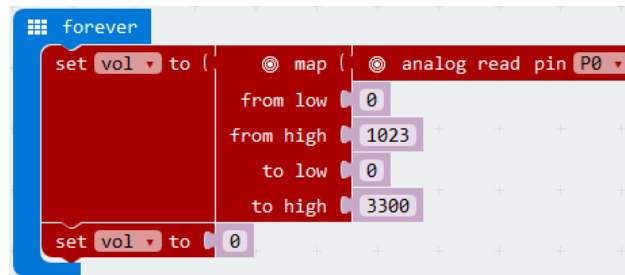
<p>1.</p>	<p>Glisser-déposer les blocs pour former le code comme indiqué. Le bloc <b>set item to</b> se trouve dans le menu sous <b>Variables</b>. Les blocs <b>map/from low/from high/to low/to high</b> et <b>analog read pin</b> se trouvent dans le menu sous <b>Pins</b>. Les blocs <b>mathematical function minus (-)</b> et <b>divide (/)</b> se trouvent dans le menu sous <b>Math</b>. Le bloc <b>show number</b> se trouve dans le menu sous <b>Basic</b>.</p>
<p>2.</p>	<p>Tout d'abord, créons deux variables. Aller à <b>Variables</b> dans le menu et cliquer sur <b>Make a Variable</b>.</p> <div data-bbox="496 846 1230 1048" style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p>New variable name:</p> <input style="width: 400px;" type="text" value="CalVa"/> <div style="display: flex; justify-content: flex-end; gap: 10px; margin-top: 5px;"> <span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">Ok ✓</span> <span style="background-color: #6c757d; color: white; padding: 2px 5px; border-radius: 3px;">Cancel ✕</span> </div> </div> <p>Entrer <b>vol</b> dans la fenêtre et cliquer sur <b>Ok</b>. Entrer <b>tem</b> dans la fenêtre et cliquer sur <b>Ok</b>. Deux nouvelles variables s'affichent dans le menu sous <b>Variables</b>.</p> <div data-bbox="711 1178 1015 1469" style="border: 1px solid #ccc; padding: 5px; margin: 10px 0; background-color: #343a40; color: white;"> <p style="text-align: center; margin: 0;">Make a Variable</p> <div style="margin-top: 10px;"> <div style="background-color: #dc3545; color: white; padding: 2px 5px; margin-bottom: 5px; display: flex; align-items: center; gap: 5px;"> <span style="font-size: 1em;">▶</span> item         </div> <div style="background-color: #dc3545; color: white; padding: 2px 5px; margin-bottom: 5px; display: flex; align-items: center; gap: 5px;"> <span style="font-size: 1em;">▶</span> tem         </div> <div style="background-color: #dc3545; color: white; padding: 2px 5px; display: flex; align-items: center; gap: 5px;"> <span style="font-size: 1em;">▶</span> vol         </div> </div> </div> <p>Nous aurons besoin de ces variables plus tard pour sauvegarder les données dans un registre. Glisser-déposer le bloc <b>set item to</b> dans le bloc <b>forever</b> et sélectionner l'option <b>vol</b> avec la flèche.</p> <div data-bbox="671 1641 1058 1787" style="border: 1px solid #ccc; padding: 5px; margin: 10px 0; background-color: #e9ecef;"> <div style="background-color: #007bff; color: white; padding: 2px 5px; margin-bottom: 5px; display: flex; align-items: center; gap: 5px;"> <span style="font-size: 1em;">▶▶▶</span> forever         </div> <div style="background-color: #dc3545; color: white; padding: 2px 5px; display: flex; align-items: center; gap: 5px;"> <span style="font-size: 1em;">▶▶▶</span> set vol to         </div> </div> <p>Glisser-déposer le bloc <b>map/from low/from high/to low/to high</b> sur le premier 0 à côté du bloc <b>set item to</b>.</p>



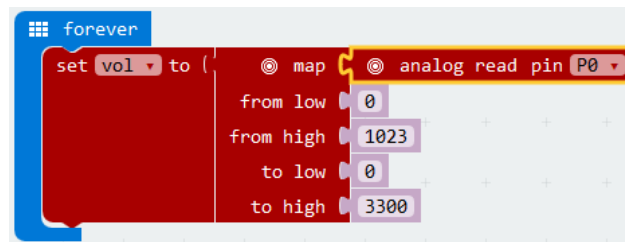
Glisser-déposer le bloc **analog read pin** à côté du bloc **map/from low/from high/to low/to high** et régler la valeur 4 pour **to high** sur 3300.



Glisser-déposer le bloc **set item to** sous le première bloc. Changer la variable **item** to **vol** avec la flèche.



La tension mesurée en mV via **analog read pin P0** est une valeur 8 bits de 0-1023 (0-3.3 V) et est affichée de bas (0 ou 0 V) en haut (3300 mV ou 3.3 V). La tension mesurée (mV) est sauvegardée dans la variable **vol**.

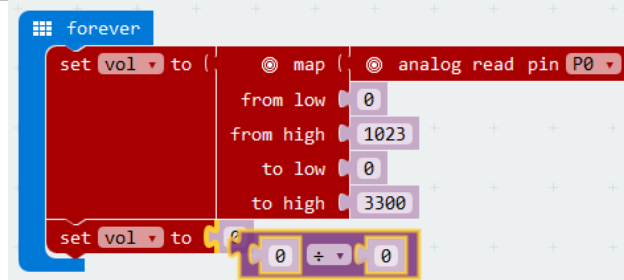


Maintenant, créons une formule pour convertir la tension mesurée en température :

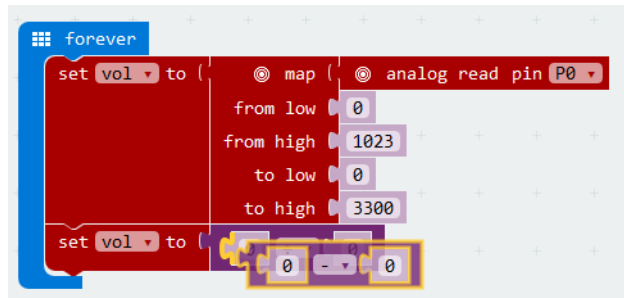
$$Temperature (^{\circ}C) = \frac{(Output\ voltage\ (mV) - 500)}{10}$$

Dans cette formule, la tension de sortie est la variable **vol** (mesurée avec le capteur TMP36). Le résultat de la formule est sauvegardé dans la variable **tem**.

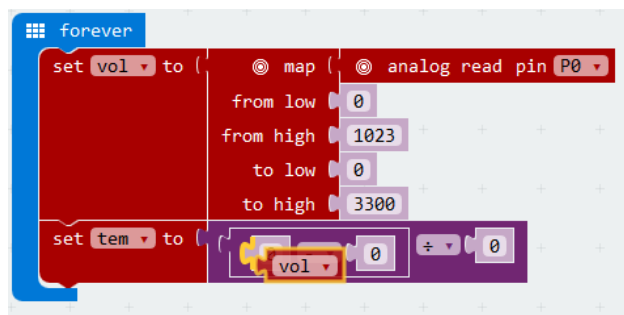
Créons la formule en utilisant les blocs. Glisser-déposer la fonction **divide (/)** sur la valeur 0 du bloc **set vol to**.



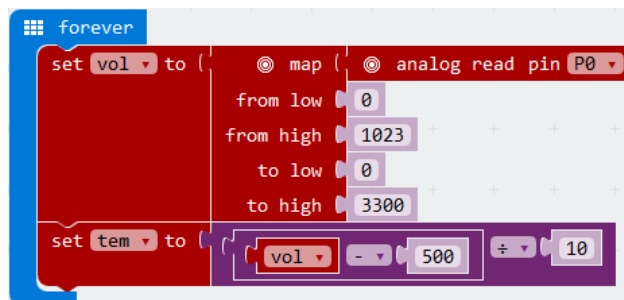
Glisser-déposer **mathematical function minus (-)** sur le premier 0 de la fonction **divide (/)**.



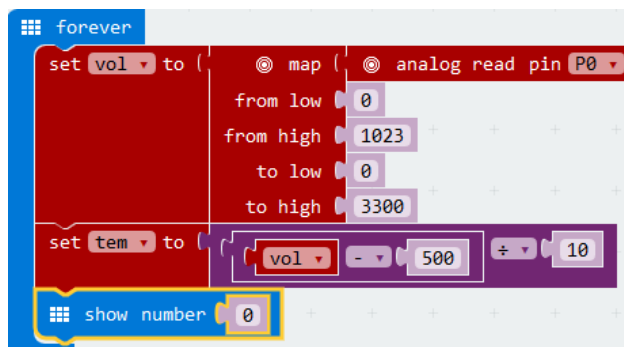
Régler la deuxième variable de **vol** sur **tem** et glisser-déposer la variable **vol** sur le premier 0 du bloc **mathematical function minus (-)**. La variable **vol** se trouve dans le menu sous **Variables**.



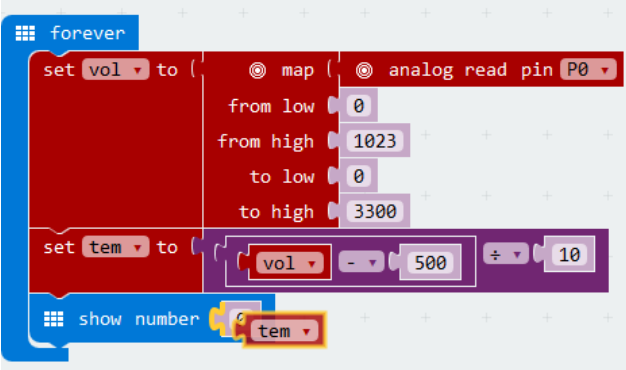
Régler la valeur 0 du bloc **mathematical function minus (-)** sur 500. Régler la valeur 0 du bloc **mathematical function divide (/)** sur 10.



Glisser-déposer le bloc **show number** sous le bloc **set tem to**.



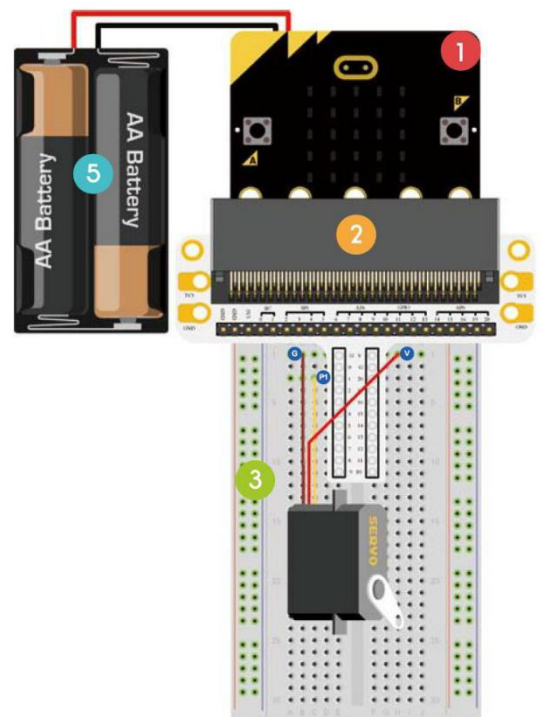
Glisser-déposer la variable **tem** sur la valeur 0 du bloc **show number**.

	 <p>Vous venez de créer le code !</p>
<p><b>3.</b></p>	<p>Une fois terminé, compiler le programme pour générer un fichier .hex. Cliquer sur le bouton Download/Télécharger et sauvegarder le fichier .hex dans le dossier Downloads → C:\downloads. Ce fichier .hex peut être téléversé vers le micro:bit.</p> <p>Connecter le micro:bit à un port USB. Glisser-déposer le fichier .hex sur le micro:bit [Removable device] pour téléverser le programme.</p>

Deux LED clignoteront alternativement. Comment afficher la température en degrés Fahrenheit ?

## 5.8 Servo

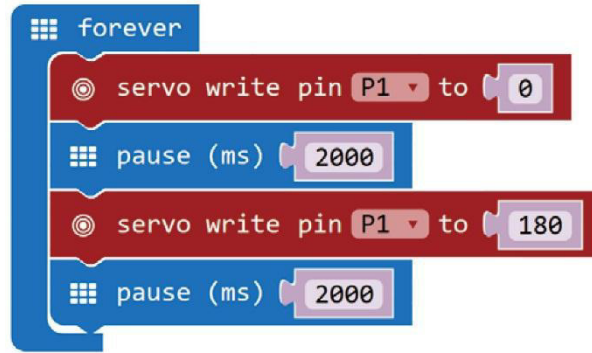
- 1 1x carte micro:bit
- 2 1x adaptateur pour platine d'expérimentation micro:bit
- 3 1x platine d'expérimentation
- 4 1x mini servo
- 5 1x support de pile avec 2x pile AA de 1.5 V



Dans l'exemple ci-dessous, nous ferons tourner un servo en continu dans une plage de 0-180°.

Poser les composants requis sur la platine d'expérimentation (voir ill.)

Regarder le code ci-dessous.

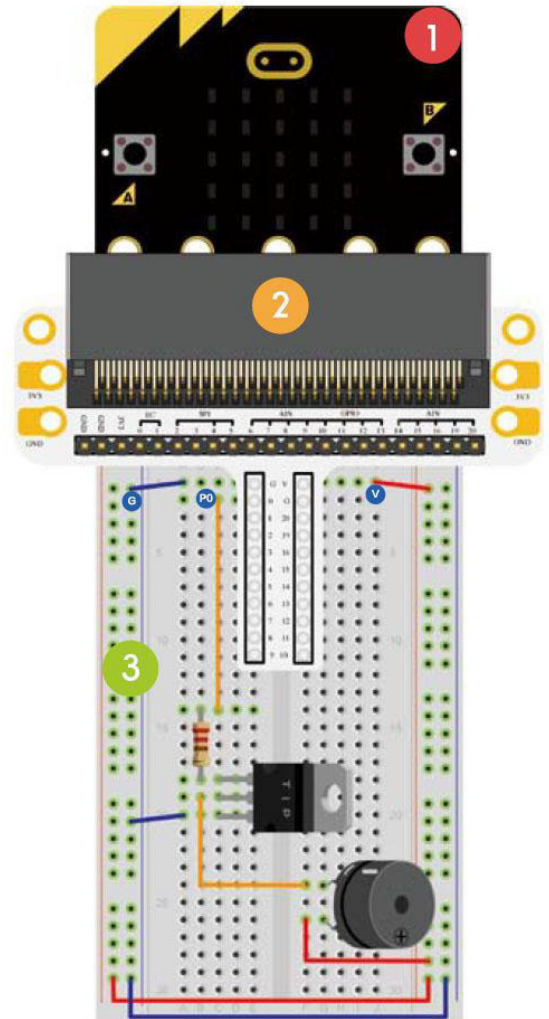
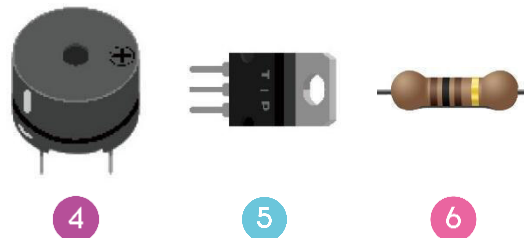


1.	Glisser-déposer les blocs pour former le code comme indiqué. Le bloc <b>servo write pin P0 to 180</b> se trouve dans le menu sous <b>Pins</b> . Les blocs <b>forever</b> et <b>pause (ms) 100</b> se trouvent dans le menu sous <b>Basic</b> .
2.	Régler <b>P0</b> sur <b>P1</b> et régler la valeur sur 0.
3.	Régler la <b>pause</b> sur 2000 ms.
4.	Régler <b>P0</b> sur <b>P1</b> . Régler la valeur 0 sur 180.
5.	Régler la <b>pause</b> sur 2000 ms.
6.	Une fois terminé, compiler le programme pour générer un fichier .hex. Cliquer sur le bouton Download/Télécharger et sauvegarder le fichier .hex dans le dossier Downloads → C:\downloads. Ce fichier .hex peut être téléversé vers le micro:bit. Connecter le micro:bit à un port USB. Glisser-déposer le fichier .hex sur le micro:bit [Removable device] pour téléverser le programme.

Le servo tourne de 0 à 180 degrés. Comment créer un thermomètre à cadran avec capteur de température et servo ?

## 5.9 Ronfleur

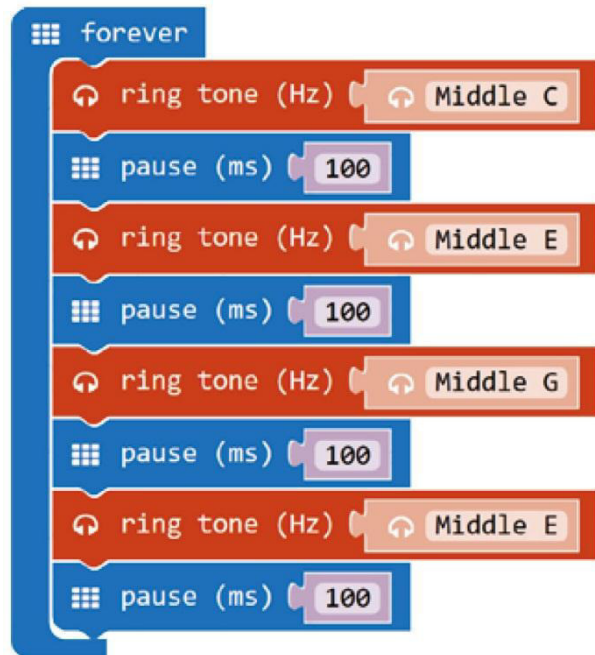
- 1 1x carte micro:bit
- 2 1x adaptateur pour platine d'expérimentation micro:bit
- 3 1x platine d'expérimentation
- 4 1x mini haut-parleur
- 5 1x transistor MOSFET Canal N
- 6 1x résistance 100 Ω (marron/noir/marron/or)

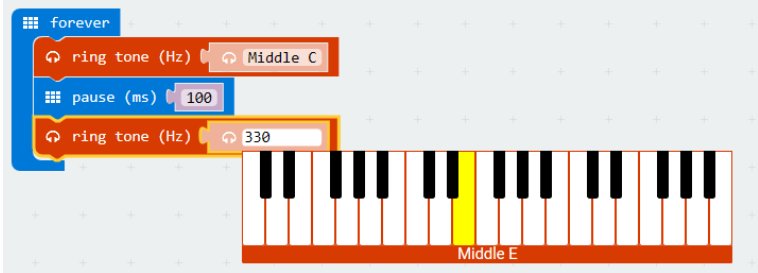


Dans l'exemple ci-dessous, nous contrôlerons un ronfleur.

Poser les composants requis sur la platine d'expérimentation (voir ill.)

Regarder le code ci-dessous.



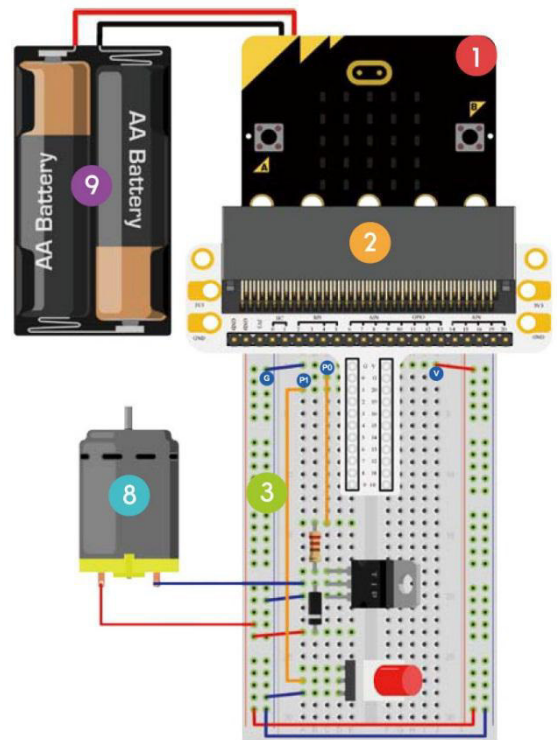
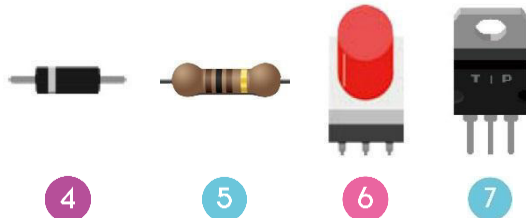
1.	<p>Glisser-déposer les blocs pour former le code comme indiqué. Les blocs <b>forever</b> et <b>pause (ms) 100</b> se trouvent dans le menu sous <b>Basic</b>. Le bloc <b>ring tone (Hz)</b> se trouve dans le menu sous <b>Music</b>.</p>
2.	<p>Régler le premier <b>ring tone (Hz)</b> sur <b>middle C</b>. Pour ce faire, cliquer sur la case de texte et sélectionner la touche de piano correspondante.</p>  <p>Le son correspondant retentit.</p>
3.	<p>Régler la <b>pause</b> sur 100 ms.</p>
4.	<p>Procéder de la même manière pour le reste des tonalités et pauses.</p>
5.	<p>Une fois terminé, compiler le programme pour générer un fichier .hex. Cliquer sur le bouton Download/Télécharger et sauvegarder le fichier .hex dans le dossier Downloads → C:\downloads. Ce fichier .hex peut être téléversé vers le micro:bit.</p> <p>Connecter le micro:bit à un port USB. Glisser-déposer le fichier .hex sur le micro:bit [Removable device] pour téléverser le programme.</p>

Le ronfleur émet un signal sonore. Comment programmer votre comptine préférée ?



## 5.10 Moteur

- 1 1x carte micro:bit
- 2 1x adaptateur pour platine d'expérimentation micro:bit
- 3 1x platine d'expérimentation
- 4 1x diode
- 5 1x résistance 100  $\Omega$  (marron/noir/marron/or)
- 6 1x commutateur autobloquant ou bistable
- 7 1x transistor MOSFET Canal N
- 8 1x mini moteur
- 9 1x support de pile avec 2x pile AA de 1.5 V



Dans l'exemple ci-dessous, nous utiliserons un interrupteur pour contrôler le démarrage et l'arrêt du moteur.

Poser les composants requis sur la platine d'expérimentation (voir ill.)

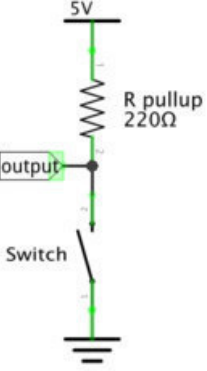
Regarder le code ci-dessous.

```

on start
  digital write pin P0 to 0
  set pull pin P1 to up

forever
  if (digital read pin P1 = 0)
  then
    digital write pin P0 to 1
  else
    digital write pin P0 to 0
  
```

1.	Glisser-déposer les blocs pour former le code comme indiqué.
2.	Le bloc <b>on start</b> est exécuté une seule fois pour lancer le programme.
3.	Régler la valeur pour <b>P0</b> sur <b>0</b> .
4.	Régler le bloc <b>pull pin</b> sur <b>P1</b> et <b>up</b> .

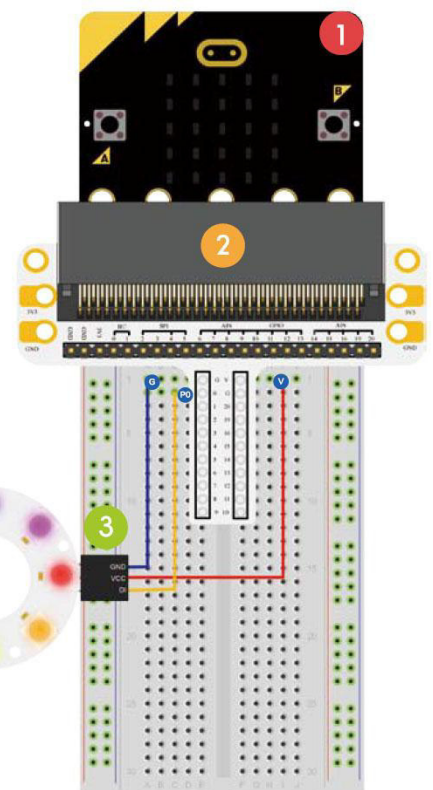
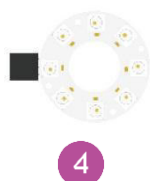
<p><b>5.</b></p>	<p>Dans <b>forever</b>, régler <b>digital read pin</b> sur <b>P1</b>. Régler le bloc <b>mathematical function equal (=)</b> sur 0.</p> <p>En fait, c'est le schéma. La résistance pull-up ne doit pas être connectée à la platine d'expérimentation. La fonction pull-up est programmée (voir étapes 3-4) dans le code et remplace la résistance sur la platine d'expérimentation.</p> 
<p><b>6.</b></p>	<p>Une fois le commutateur enfoncé, régler la haute tension sur P0. Régler la valeur 0 sur 1 (5 V). Le moteur commence à tourner.</p>
<p><b>7.</b></p>	<p>Une fois le commutateur relâché, régler la basse tension sur P0. Régler la valeur sur 0. Le moteur s'arrête.</p>
<p><b>8.</b></p>	<p>Une fois terminé, compiler le programme pour générer un fichier .hex. Cliquer sur le bouton Download/Télécharger et sauvegarder le fichier .hex dans le dossier Downloads → C:\downloads. Ce fichier .hex peut être téléversé vers le micro:bit.</p> <p>Connecter le micro:bit à un port USB. Glisser-déposer le fichier .hex sur le micro:bit [Removable device] pour téléverser le programme.</p>

Appuyer sur le commutateur pour faire tourner le moteur et relâcher pour l'arrêter. Comment utiliser un potentiomètre ajustable pour contrôler la vitesse du moteur?

Note : La tension du micro:bit est limitée à 3.3 V, il se peut qu'elle ne soit pas suffisante pour supporter le ventilateur. Pour faire fonctionner le ventilateur, il peut être nécessaire de tourner la lame pour faciliter le démarrage.

### 5.11 LED arc-en-ciel

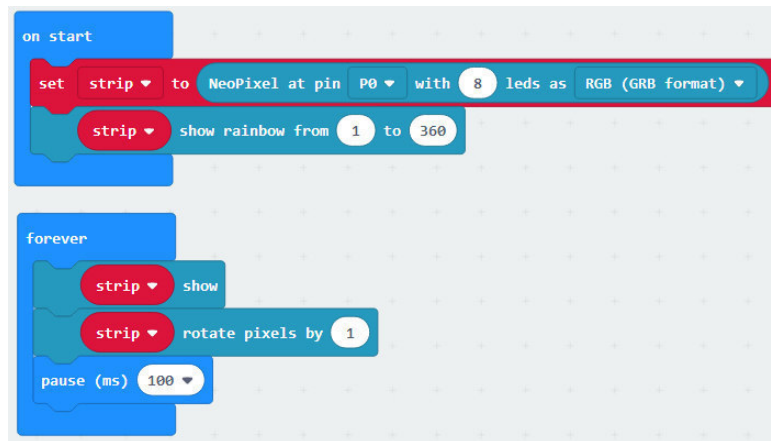
- 1 1x carte micro:bit
- 2 1x adaptateur pour platine d'expérimentation micro:bit
- 3 1x platine d'expérimentation
- 4 1x anneau LED RGB



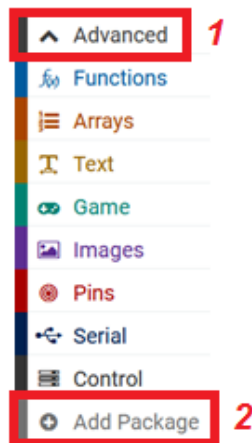
Dans l'exemple ci-dessous, nous contrôlerons huit LED RGB d'un anneau LED pour réaliser un effet arc-en-ciel.

Poser les composants requis sur la platine d'expérimentation (voir ill.)

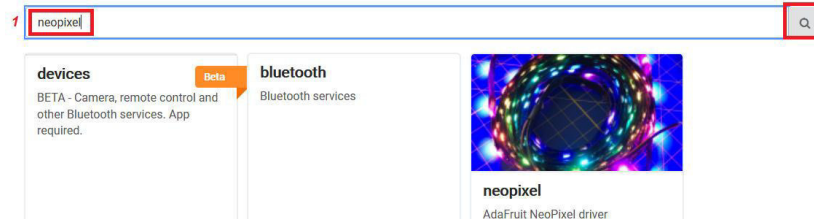
Regarder le code ci-dessous.



1. Glisser-déposer les blocs pour former le code comme indiqué.
2. Chercher et ajouter la bibliothèque NeoPixel.  
Aller à Advanced → Add Package (Extensions) et entrer NeoPixel.

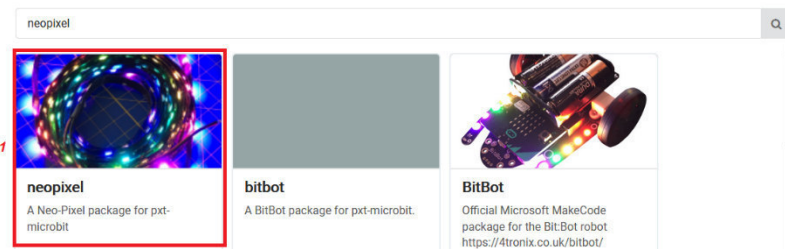


Add Package... ?

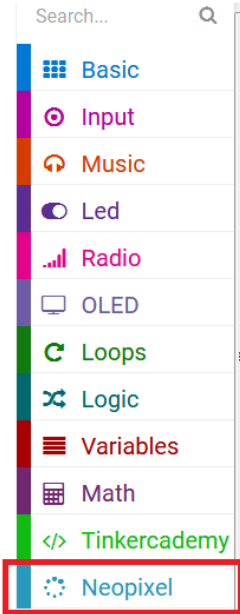

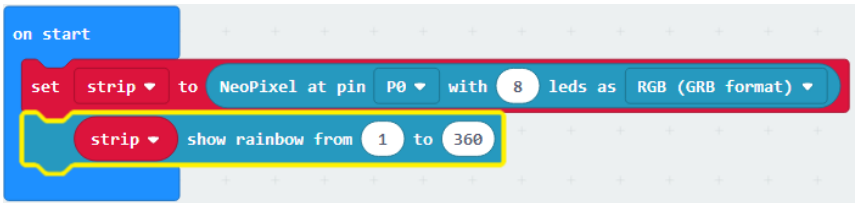


Sélectionner la bibliothèque NeoPixel.

Add Package... ?



La bibliothèque est téléchargée et ajoutée au menu.

	 <p>Les blocs <b>NeoPixel at pin P0 with 24 leds, item show rainbow from 1 to 360, item show</b> et <b>items rotate pixels by 1</b> se trouvent dans le menu sous <b>NeoPixel</b>.</p>
3.	<p>Glisser-déposer le bloc <b>set strip to NeoPixel at pin P0 with 24 leds as RGB (GRB format)</b> dans le bloc <b>on start</b>. Régler la broche sur <b>P0</b> et la valeur 24 sur 8 (l'anneau LED est doté de 8 LED).</p> 
4.	<p>Glisser-déposer le bloc <b>set strip to rainbow from 1 to 360</b> sous le bloc <b>set strip to NeoPixel at pin P0</b>.</p> 
5.	<p>Compléter le code selon l'exemple.</p>
6.	<p>Une fois terminé, compiler le programme pour générer un fichier .hex. Cliquer sur le bouton Download/Télécharger et sauvegarder le fichier .hex dans le dossier Downloads → C:\downloads. Ce fichier .hex peut être téléversé vers le micro:bit.</p> <p>Connecter le micro:bit à un port USB. Glisser-déposer le fichier .hex sur le micro:bit [Removable device] pour téléverser le programme.</p>

Les LED créent un effet arc-en ciel sur l'anneau LED. Comment faire clignoter l'anneau comme un œil ?

**N'employer cet appareil qu'avec des accessoires d'origine. Velleman SA ne peut, dans la mesure conforme au droit applicable être tenue responsable des dommages ou lésions (directs ou indirects) pouvant résulter de l'utilisation de cet appareil. Pour plus d'informations concernant cet article et la dernière version de ce mode d'emploi, consulter notre site [www.velleman.eu](http://www.velleman.eu). Les spécifications et le contenu de ce mode d'emploi peuvent être modifiés sans notification préalable.**

#### © DROITS D'AUTEUR

**Velleman SA est l'ayant droit des droits d'auteur de ce mode d'emploi. Tous droits mondiaux réservés.** Toute reproduction, traduction, copie ou diffusion, intégrale ou partielle, du contenu de ce mode d'emploi par quelque procédé ou sur tout support électronique que ce soit est interdite sans l'accord préalable écrit de l'ayant droit.

# MANUAL DEL USUARIO

## 1. Introducción

### A los ciudadanos de la Unión Europea

#### Importantes informaciones sobre el medio ambiente concerniente a este producto



Este símbolo en este aparato o el embalaje indica que, si tira las muestras inservibles, podrían dañar el medio ambiente. No tire este aparato (ni las pilas, si las hubiera) en la basura doméstica; debe ir a una empresa especializada en reciclaje. Devuelva este aparato a su distribuidor o a la unidad de reciclaje local. Respete las leyes locales en relación con el medio ambiente.

**Si tiene dudas, contacte con las autoridades locales para residuos.**

¡Gracias por elegir Velleman®! Lea atentamente las instrucciones del manual antes de usar el aparato. Si ha sufrido algún daño en el transporte no lo instale y póngase en contacto con su distribuidor.

## 2. Instrucciones de seguridad



Este aparato no es apto para niños menores de 8 años ni para personas con capacidades físicas, sensoriales o mentales reducidas ni para personas con una falta de experiencia y conocimientos del producto, salvo si están bajo la vigilancia de una persona que pueda garantizar la seguridad. Asegúrese de que los niños no jueguen con este dispositivo. Nunca deje que los niños limpien o manipulen el aparato sin supervisión.



Utilice el aparato sólo en interiores.

No exponga este equipo a lluvia, humedad ni a ningún tipo de salpicadura o goteo.

## 3. Normas generales



- Véase la Garantía de servicio y calidad Velleman® al final de este manual del usuario.
- Familiarícese con el funcionamiento del aparato antes de utilizarlo.
- Por razones de seguridad, las modificaciones no autorizadas del aparato están prohibidas. Los daños causados por modificaciones no autorizadas, no están cubiertos por la garantía.
- Utilice sólo el aparato para las aplicaciones descritas en este manual Su uso incorrecto anula la garantía completamente.
- Los daños causados por descuido de las instrucciones de seguridad de este manual invalidarán su garantía y su distribuidor no será responsable de ningún daño u otros problemas resultantes.
- Ni Velleman nv ni sus distribuidores serán responsables de los daños extraordinarios, ocasionales o indirectos, sea cual sea la índole (financiera, física, etc.), causados por la posesión, el uso o el fallo de este producto.
- Debido a las continuas mejoras, el producto podría diferir del de las imágenes.
- Las imágenes son meramente ilustrativas.
- No conecte el aparato si ha estado expuesto a grandes cambios de temperatura. Espere hasta que el aparato llegue a la temperatura ambiente.
- Guarde este manual del usuario para cuando necesite consultarlo.

## 4. Función

Este kit para principiantes es un kit educativo que se basa en el sistema micro:bit. Lleva componentes electrónicos básicos, una placa de pruebas, cables de conexión y un micro:bit.

## 5. Ejemplos

### 5.1 LED

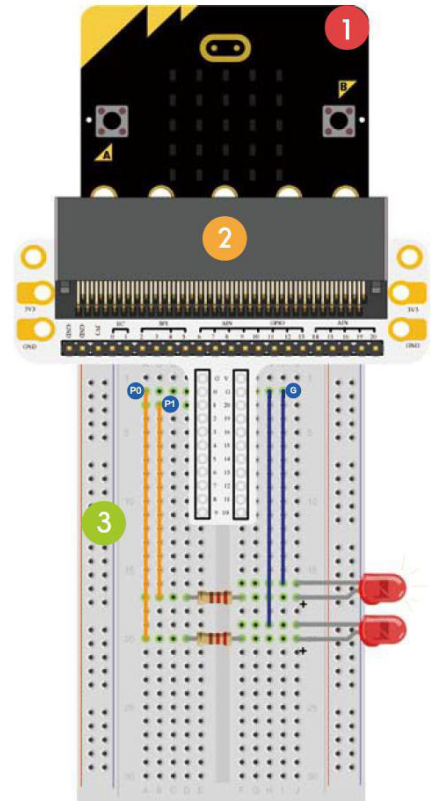
- 1 1x micro:bit
- 2 1x conector micro:bit para placa de pruebas
- 3 1x placa de pruebas
- 4 2x LED rojo (polarización: ánodo (+) = pata larga, cátodo (-) = pata corta)
- 5 2x resistencia 100  $\Omega$  (marrón/negro/marrón/dorado)



4



5



En este manual del usuario, utilizaremos un micro:bit para hacer parpadear dos LEDs alternativamente.

Coloque los componentes necesarios en la placa de pruebas (véase fig.).

El programa lleva bloques de programación y se puede ejecutar en línea. Introduzca en el navegador la dirección [www.makecode.com](http://www.makecode.com) o <https://www.microsoft.com/en-us/makecode?rtc=1> y haga clic en el icono de micro:bit y luego en **Start Project/Nuevo Proyecto**.

Microsoft Office Windows Surface Xbox Deals Support More

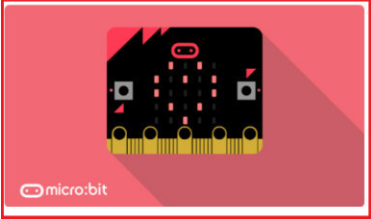
MakeCode About Get started Resources

Search Microsoft.com Sign in

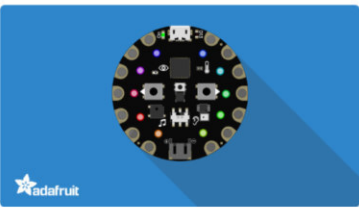
### Hands on computing education

Microsoft MakeCode brings computer science to life for all students with fun projects, immediate results, and both block and text editors for learners at different levels.


*Klik dit icoon aan*



micro:bit  
[Start coding with micro:bit >](#)



Circuit Playground Express  
[Start coding with Circuit Playground Express >](#)



Minecraft  
[Start coding with Minecraft >](#)

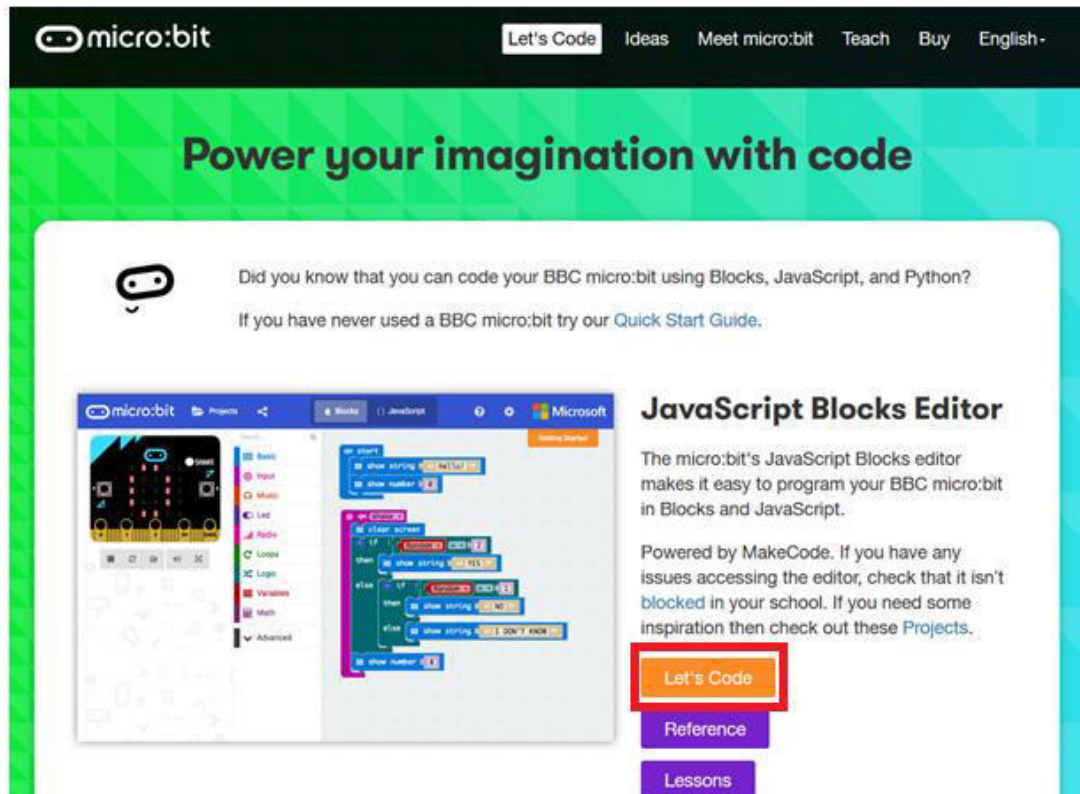
El editor de bloques de micro:bit se abrirá en una nueva ventana. Ahora, puede empezar a crear el código con los bloques arrastrándolos del menú al área de programación.

Veamos a continuación cómo funciona esto.



## ¿Qué es MakeCode?

Anteriormente PXT – Programming eXperience Toolkit Editor para escribir códigos para el micro:bit. Es un sencillo editor gráfico en línea similar a Scratch que posibilita introducirse en el mundo de la programación de forma intuitiva a través de un lenguaje de programación visual basado en bloques. Todo se ejecuta desde la web, es decir desde el navegador.



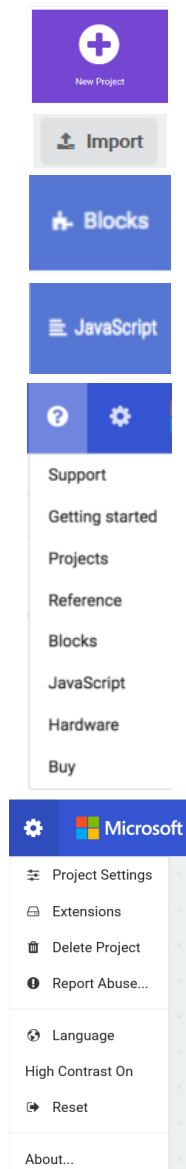
El editor lleva diferentes secciones. Puede crear su código en el **área de programación**, arrastrando los bloques desde el **menú de bloques**. En el **simulador micro:bit** podrá comprobar el funcionamiento del programa. En la parte inferior puede descargar y almacenar el proyecto.



Ahora, explicaremos cómo utilizar la barra de opciones **Options Bar**.



## Barra de opciones



Hacer clic para crear o añadir un nuevo proyecto.

Importar proyectos.

Visualización mediante bloques

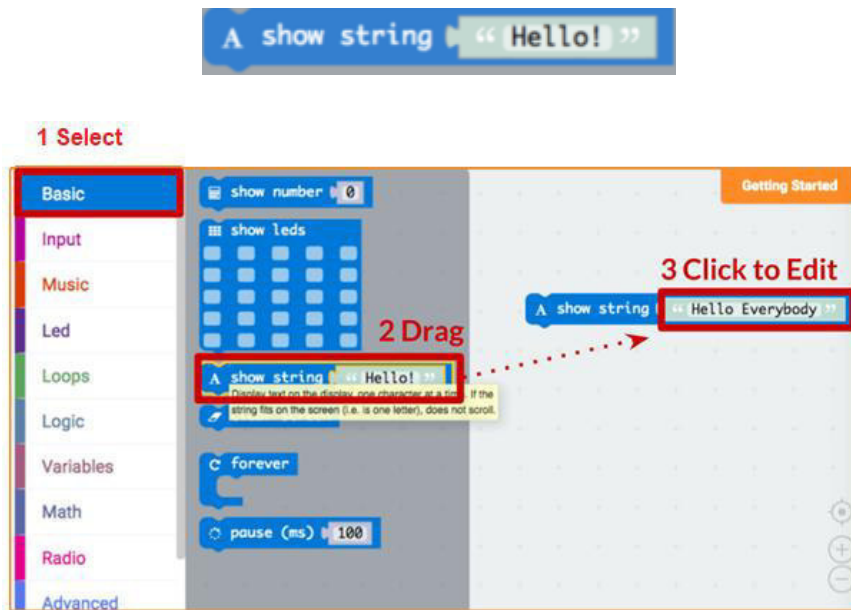
Ver el código escrito en JavaScript

Sección de ayuda

Sección de configuración Aquí es posible cambiar el nombre del proyecto y borrarlo. **Reset/Restablecer**: con esta opción borrará todos los proyectos guardados. ¡Tenga cuidado! La mayoría de las veces, sólo necesitará la opción **Delete Project/Eliminar proyecto**.

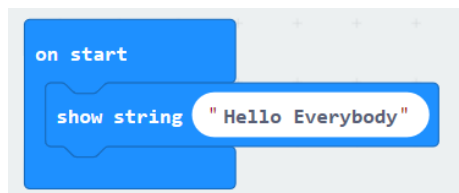
### Barra de LEDs

Vamos a crear un código. En el menú de bloques encontrará todos los códigos disponibles. Arrastre el bloque **show string/mostrar cadena** en el área de programación y haga clic en la caja para editar.



<b>1</b>	Seleccionar
<b>2</b>	Arrastrar

<b>3</b>	Hacer clic para editar
----------	------------------------

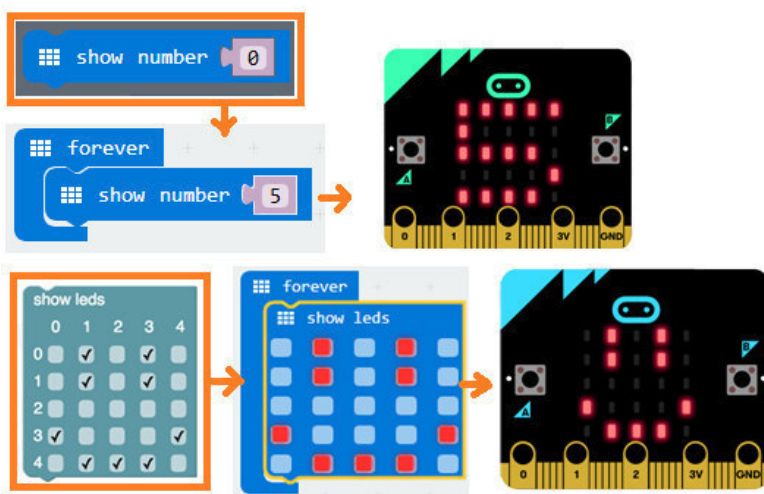


Se puede ver la ejecución del programa en el simulador.

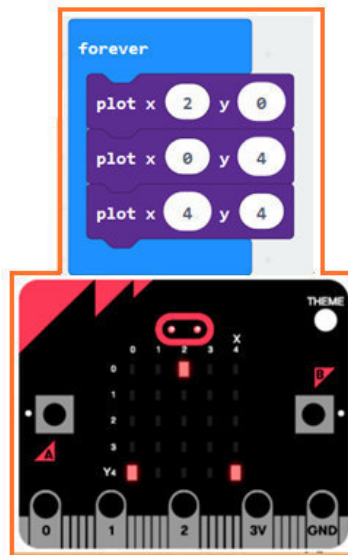
### Say Hello!

El texto visualizado se llama **string/cadena**.

Con la función **show number/mostrar número** se visualizarán números enteros en la pantalla LED. Con la función **show leds/mostrar LEDs** se visualizará una imagen de 5x5 píxeles.



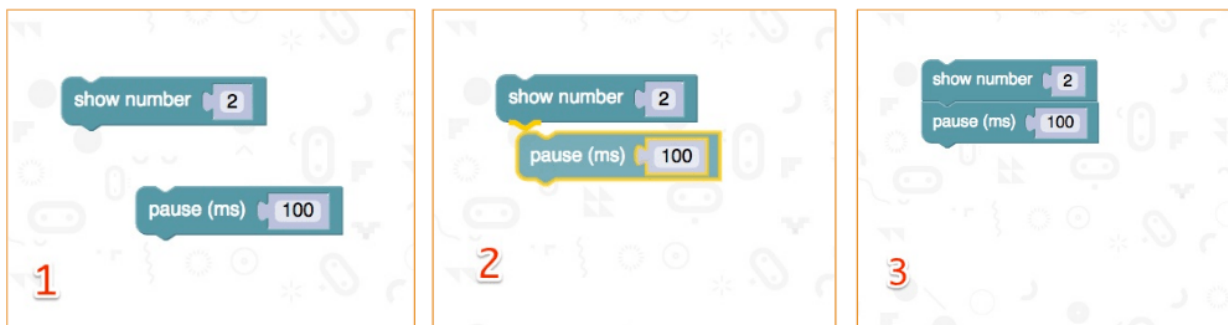
También puede iluminar un punto LED con coordenadas. La Coordenada (0,0) está en la esquina superior izquierda.



### Unir bloques

Haga clic en el bloque que quiere unir y arrástrelo hacia el bloque deseado hasta que un lado de este se ilumine. Al soltar el bloque ambos se unirán.

Al hacer clic en el primer bloque el segundo se moverá. Al hacer clic en el segundo bloque, este se separará del primero.



### micro:bit

Conecte la micro:bit al ordenador con el cable micro-USB.

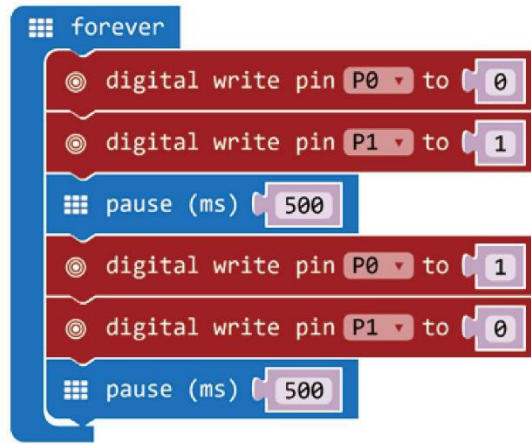
Se ejecutará una secuencia por defecto. Se le pedirá que pulse botones y juegue un juego. ¡Pruébalo!

Haga clic en **Download/Descargar** para convertir su texto en un **fichero hex** y descargarlo.

No necesitará seguir el siguiente paso si Chrome está configurado correctamente. Arrastre el fichero hex descargado en el lector micro:bit o haga clic derecho en **Send To** (Windows®).

Ahora, acaba de programar su aparato.

Crear el código

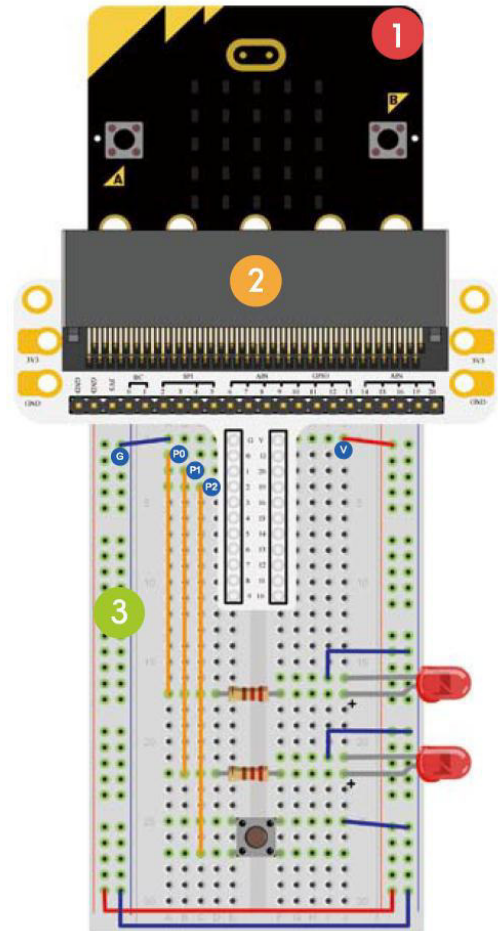
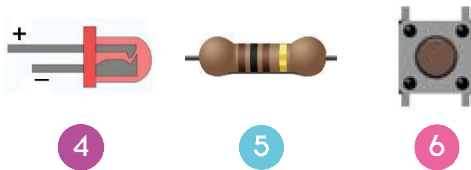


1.	Arrastre y suelte los bloques como se indica en la figura. El bloque <b>digital write pin P0 to 0</b> /escritura digital pin P0 a 0 se sitúa en el menú bajo <b>Advanced/Avanzado</b> → <b>Pins/Pines</b> . El bloque <b>forever/para siempre</b> y el bloque <b>pause/pausa (ms) 100</b> se sitúan en el menú bajo <b>Basic/Básico</b> .
2.	Coloque el valor para P0 en 0. LED0 apagado = baja tensión = 0 V = digital 0. Coloque el valor para P1 en 1. LED1 encendido = alta tensión = 5 V = digital 1.
3.	Coloque el bloque <b>pause/pausa</b> en 500 ms.
4.	Coloque el valor para P0 en 1. LED0 encendido = alta tensión = 5 V = digital 1. Coloque el valor para P1 en 0. LED1 apagado = baja tensión = 0 V = digital 0.
5.	Coloque el bloque <b>pause/pausa</b> en 500 ms.
6.	Luego, compile el programa para generar un fichero .hex. Haga clic en el botón Download/descargar y almacene el fichero .hex file en la carpeta →Downloads (C:\Downloads). Este fichero .hex está listo para su subida a la micro:bit. Conecte micro:bit a un puerto USB. Ahora, arrastre y suelte el fichero .hex en la micro:bit [removable device] para subir el programa.

Los dos LEDs parpadearán alternativamente. Ahora, vamos a crear un semáforo RGB.

## 5.2 Botón

- 1 1x micro:bit
- 2 1x conector micro:bit para placa de pruebas
- 3 1x placa de pruebas
- 4 2x LED rojo (polarización: ánodo (+) = pata larga, cátodo (-) = pata corta)
- 5 2x resistencia 100  $\Omega$  (marrón/negro/marrón/dorado)
- 6 1x interruptor momentáneo



Utilizaremos un interruptor para controlar el parpadeo de los LEDs. Pulse el interruptor para hacer que los LEDs parpadeen alternativamente. Suelte el interruptor para desactivar el LED.

Coloque los componentes necesarios en la placa de pruebas (véase fig.).

Crear el código

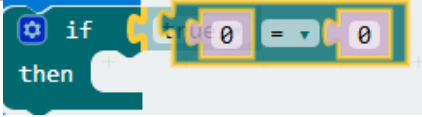
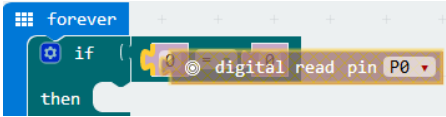
```

on start
  set pull pin P2 to up

forever
  if (digital read pin P2 = 0)
  then
    digital write pin P0 to 0
    digital write pin P1 to 1
    pause (ms) 500
    digital write pin P0 to 1
    digital write pin P1 to 0
    pause (ms) 500
  
```

1. Arrastre y suelte los bloques como se indica en la figura. El bloque **digital read pin P0/lectura digital pin P0** y el bloque **digital write pin P0 to 0/escritura digital pin P0 a 0** se sitúan en el menú bajo **Advanced/Avanzado** → **Pins/Pines**. El bloque **forever/para siempre** y el bloque **pause/pausa (ms) 100** se sitúan en el menú bajo **Basic/Básico**. El bloque **if then/si entonces** y el bloque con el **signo igual** se sitúan en el menú bajo **Logic/Lógico**. Arrastre y suelte el bloque con el **signo igual** sobre el bloque **true/verdadero**.



	 <p>Los dos bloques encajarán.</p> <p>Arrastre y suelte el bloque <b>digital read pin P0/lectura digital pin P0</b> sobre el bloque <b>0</b> del bloque con el <b>signo igual</b> (véase fig.).</p> 
<b>2.</b>	Ajuste P2 como una resistencia pull-up.
<b>3.</b>	Coloque <b>digital read pin/lectura digital pin</b> en P2.
<b>4.</b>	Coloque <b>digital write pin P0/escritura digital pin P0 en 0</b> (activo bajo (0 V)). Coloque <b>digital write pin P1/escritura digital pin P1 en 1</b> (activo alto (5 V)).
<b>5.</b>	Coloque el bloque <b>pause/pausa</b> en 500 ms.
<b>6.</b>	Coloque <b>digital write pin P0/escritura digital pin P0 en 1</b> (activo alto (5 V)). Coloque <b>digital write pin P1/escritura digital pin P1 en 0</b> (activo bajo (0 V)).
<b>7.</b>	Coloque el bloque <b>pause/pausa</b> en 500 ms.
<b>8.</b>	Luego, compile el programa para generar un fichero .hex. Haga clic en el botón Download/descargar y almacene el fichero .hex file en la carpeta →Downloads (C:\Downloads). Este fichero .hex está listo para su subida a la micro:bit. Conecte micro:bit a un puerto USB. Ahora, arrastre y suelte el fichero .hex en la micro:bit [removable device] para subir el programa.

Pulse el botón y los LEDs parpadearán alternativamente. Ahora, ¿cómo hacer que el LED rojo se ilumine mientras se pulsa el botón? y ¿cómo hacer que el LED rojo se ilumine cuando el botón no está pulsado?

### 5.3 Potenciómetro de ajuste

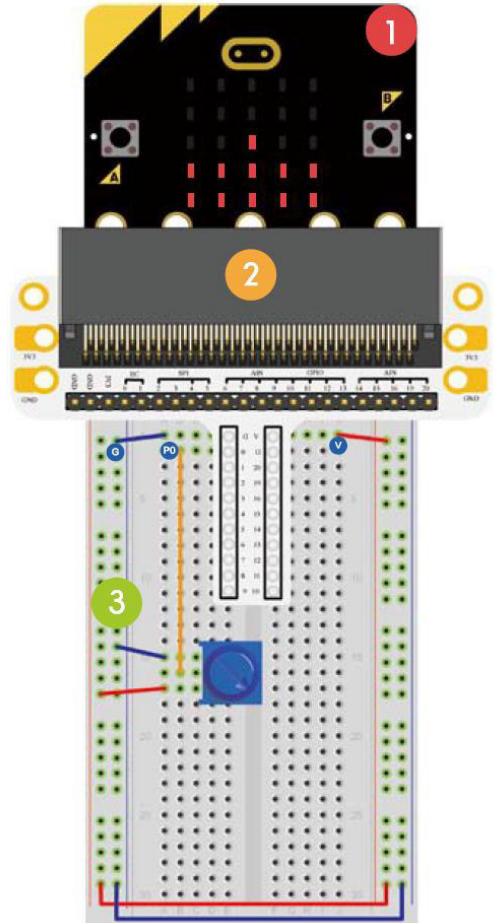
- 1 1x micro:bit
- 2 1x conector micro:bit para placa de pruebas
- 3 1x placa de pruebas
- 4 1x potenciómetro de ajuste 10 k $\Omega$



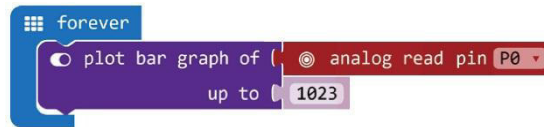
4

Vamos a leer la tensión de salida del potenciómetro de ajuste y la visualizaremos con un diagrama de barras en la pantalla del micro:bit.

Coloque los componentes necesarios en la placa de pruebas (véase fig.).



Crear el código



1.	Arrastre y suelte los bloques como se indica en la figura. El bloque <b>plot bar graph of 0 up to 0</b> se sitúa en el menú bajo <b>LED</b> .
2.	Coloque <b>analog read pin/lectura analógica pin</b> en P0. Este bloque se sitúa en el menú bajo <b>Pins/Pines</b> . Introduzca el valor 1023.
3.	Luego, compile el programa para generar un fichero .hex. Haga clic en el botón Download/descargar y almacene el fichero .hex file en la carpeta →Downloads (C:\Downloads). Este fichero .hex está listo para su subida a la micro:bit. Conecte micro:bit a un puerto USB. Ahora, arrastre y suelte el fichero .hex en la micro:bit [removable device] para subir el programa.

Gire el potenciómetro de ajuste. La tensión se visualizará con un diagrama de barras en la pantalla del micro:bit. Si la tensión es 0, la pantalla LED visualizará sólo un punto pixel. Si la tensión es de 3.3 V, la pantalla entera se iluminará. Ahora, ¿cómo utilizaría el potenciómetro de ajuste para regular el brillo de los LEDs?

#### 5.4 fotocélula

- 1 1x micro:bit
- 2 1x conector micro:bit para placa de pruebas
- 3 1x placa de pruebas
- 4 1x fotocélula
- 5 1x resistencia 10 kΩ (marrón/negro/negro/rojo/marrón)



4

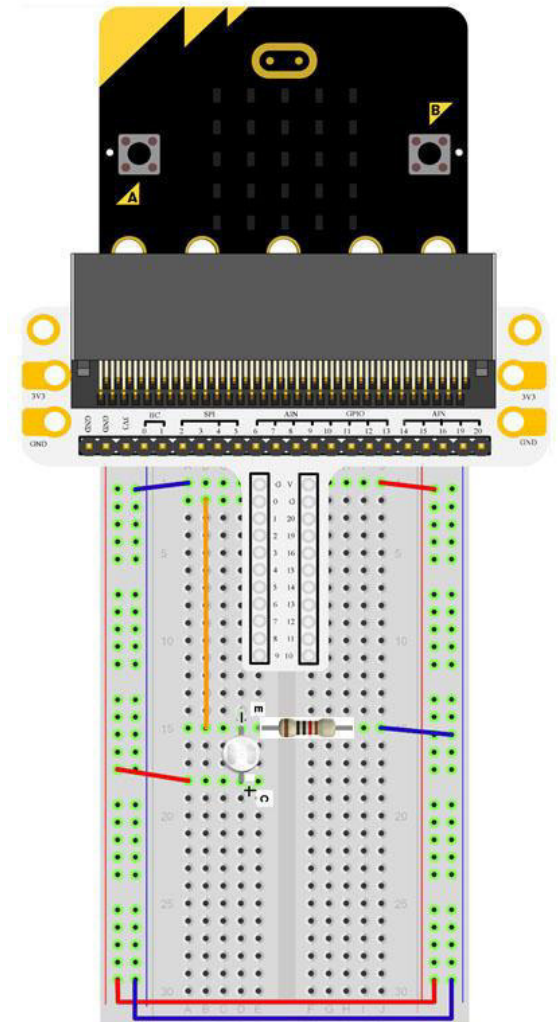
Pata corta = colector  
Ánodo positivo  
Pata larga = emisor  
Cátodo negativo



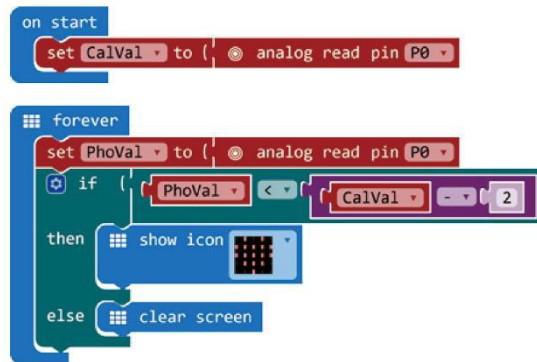
5

Vamos a utilizar una fotocélula para controlar la luminosidad de la pantalla del micro:bit.

Coloque los componentes necesarios en la placa de pruebas (véase fig.).



Crear el código

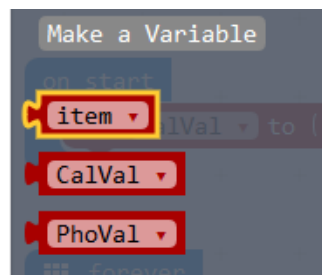


- Primero, vamos a crear dos variables. Seleccione **Variables** en el menú y haga clic en **Make a Variable**.

New variable name:

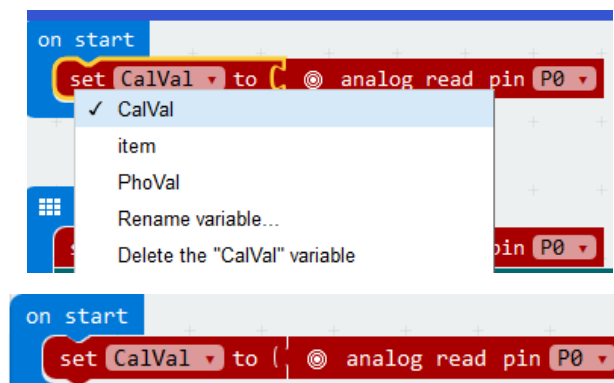
Introduzca **CalVal** en la ventana y haga clic en **Ok**. Introduzca **PhoVal** en la ventana y haga clic en **Ok**.

Se visualizarán dos nuevos variables en el menú bajo **Variables**.



Más adelante, necesitará estos variables para almacenar los datos en el registro.

El bloque con el símbolo para indicar **más pequeño que** y el bloque **if then else/si entonces si nose** sitúan en el menú bajo **Logic/Lógico**. El bloque con el **signo de menos** se sitúa en el menú bajo **Math/Matemática**. El bloque **show icon/mostrar ícono** se sitúa en el menú bajo **Basic/Básico**. El bloque con el **clear screen/borrar la pantalla** se sitúa en el menú bajo **Basic/Básico** → **More**. El bloque con el **set item to 0** se sitúa en el menú bajo **Variables**. Haga clic en la flecha y seleccione **CalVal** o **PhoVal**.



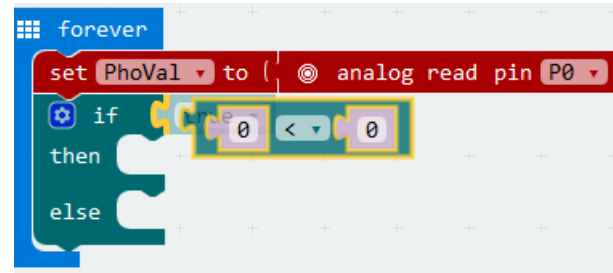
2. Seleccione el variable **CalVal** y coloque **analog read pin/lectura analógica pin** en P0.



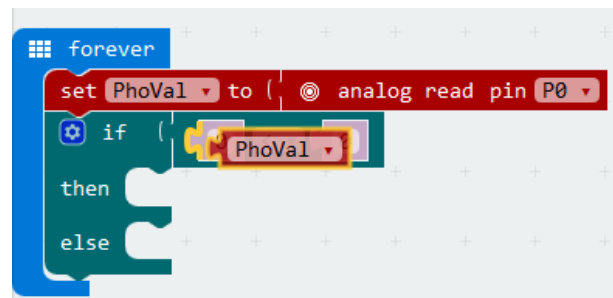
3. En el bloque **forever/para siempre**, seleccione el variable **PhoVal** y coloque **analog read pin/lectura analógica pin** en P0.



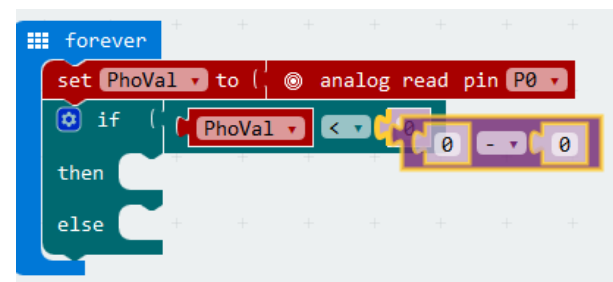
4. Arrastre y suelte el bloque con el símbolo para indicar **más pequeño que** junto al bloque **if/si** y colóquelo sobre el bloque **true/verdadero**.



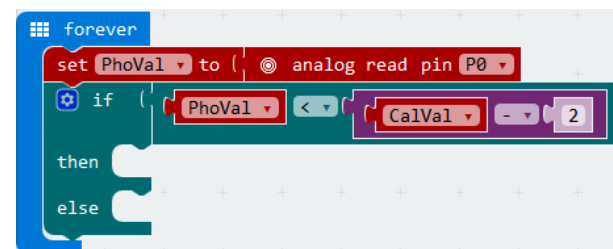
Luego, arrastre y suelte el variable **PhoVal** (bajo **Variables** en el menú) sobre el bloque del primer 0 del bloque con el símbolo para indicar **más pequeño que**.



Arrastre y suelte el bloque con el **signo de menos** sobre el segundo 0 del bloque con el símbolo para indicar **más pequeño que**.



Luego, arrastre y suelte el variable **CalVal** (de **Variables** en el menú) sobre el bloque del primer 0 del bloque con el **signo de menos**. Coloque el segundo 0 del bloque con el **signo de menos** en 2.



5.	Arrastre y suelte el bloque <b>show icon/mostrar ícono</b> junto al bloque <b>then/entonces</b> . Arrastre y suelte el bloque <b>clear screen/borrar la pantalla</b> junto al bloque <b>else/sino</b> .
6.	Si el variable <b>PhoVal</b> es más pequeño que el variable <b>CalVal - 2</b> , la pantalla visualizará un corazón. Si no fuera el caso, se apagará.
7.	Luego, compile el programa para generar un fichero .hex. Haga clic en el botón Download/descargar y almacene el fichero .hex file en la carpeta →Downloads (C:\Downloads). Este fichero .hex está listo para su subida a la micro:bit. Conecte micro:bit a un puerto USB. Ahora, arrastre y suelte el fichero .hex en la micro:bit [removable device] para subir el programa.

Observación: Vuelva a la configuración de fábrica para calibrar el valor de referencia según el brillo actual. Para ejecutar el programa de forma correcta, inícielo con la luz encendida.

Si la luz está encendida, no se visualizará nada en la pantalla. Si la luz está apagada, se visualizará un corazón en la pantalla. Ahora, ¿cómo podemos utilizar una fotocélula para controlar un LED?

## 5.5 LED RGB

- 1 1x micro:bit
- 2 1x conector micro:bit para placa de pruebas
- 3 1x placa de pruebas
- 4 1x LED RGB (cátodo común)
- 5 3x resistencia 10  $\Omega$  (marrón/negro/marrón/dorado)

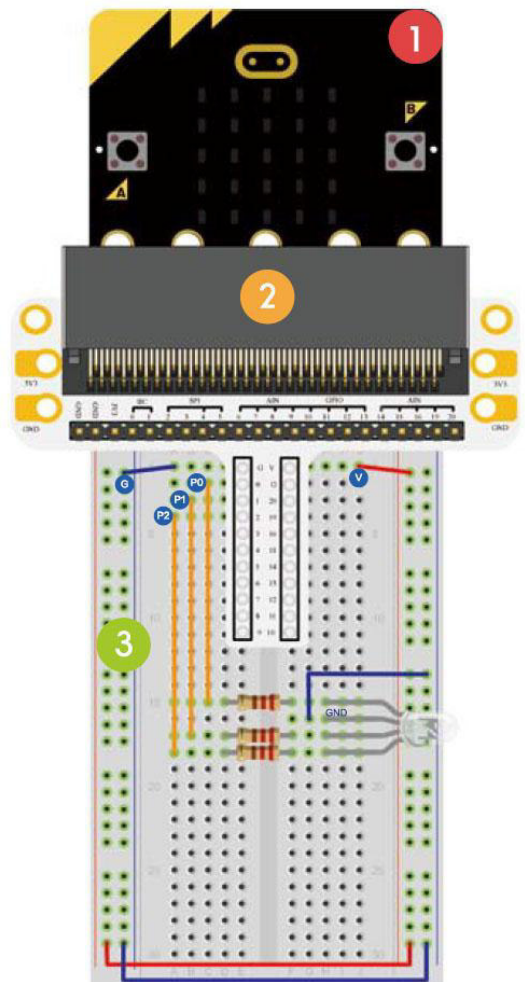


4

1. Rojo (ánodo +)
2. Tierra (cátodo -) – pata larga
3. Verde (ánodo +)
4. Azul (ánodo +)



5

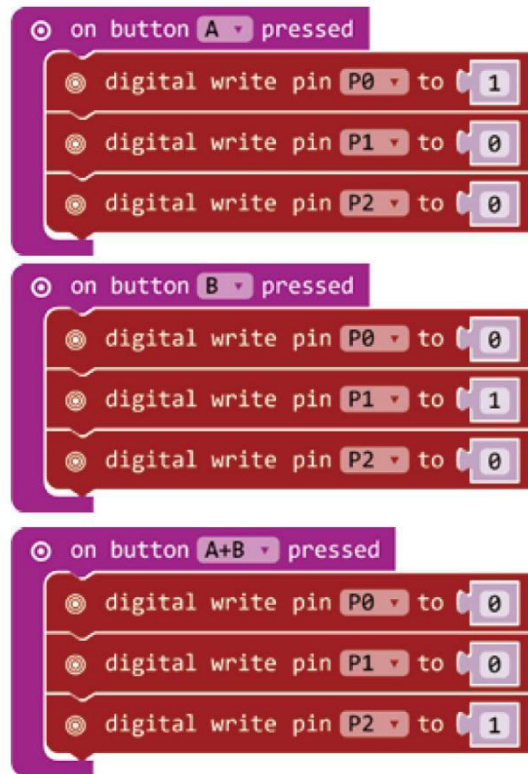


Vamos a hacer que el LED RGB conmute gradualmente entre los colores rojo, verde y azul.

Coloque los componentes necesarios en la placa de pruebas (véase fig.).



Crear el código



- Arrastre y suelte los bloques como se indica en la figura.

El bloque **on button A pressed/al presionarse el botón A** se sitúa en el menú bajo **Input/Entrada**. El bloque **digital write pin P0 to 0/escritura digital pin P0 a 0** se sitúa en el menú bajo **Advanced/Avanzado Pins/Pines**.

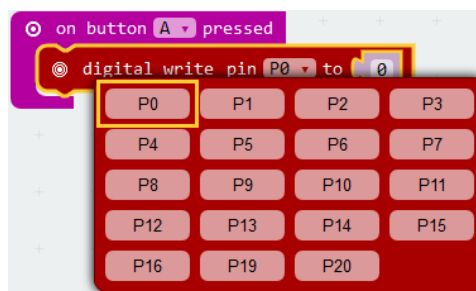
Seleccione la opción A en el bloque **on button A pressed/al presionarse el botón A**.

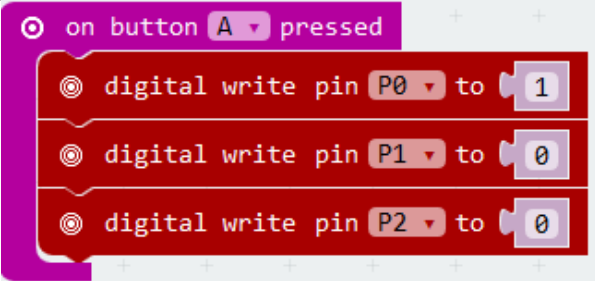
Coloque 3 bloques **digital write pin P0 to 0/escritura digital pin P0 a 0** e introdúzcalos en el bloque **on button A pressed** /al presionarse el botón A.

Seleccione el pin P0 en el primer bloque **digital write pin P0 to 0/escritura digital pin P0 a 0** y coloque el valor en 1 (LED rojo encendido).

Seleccione el pin P1 en el segundo bloque **digital write pin P0 to 0/escritura digital pin P0 a 0** y coloque el valor en 0 (LED verde apagado).

Seleccione el pin P2 en el tercer bloque **digital write pin P0 to 0/escritura digital pin P0 a 0** y coloque el valor en 0 (LED azul apagado).

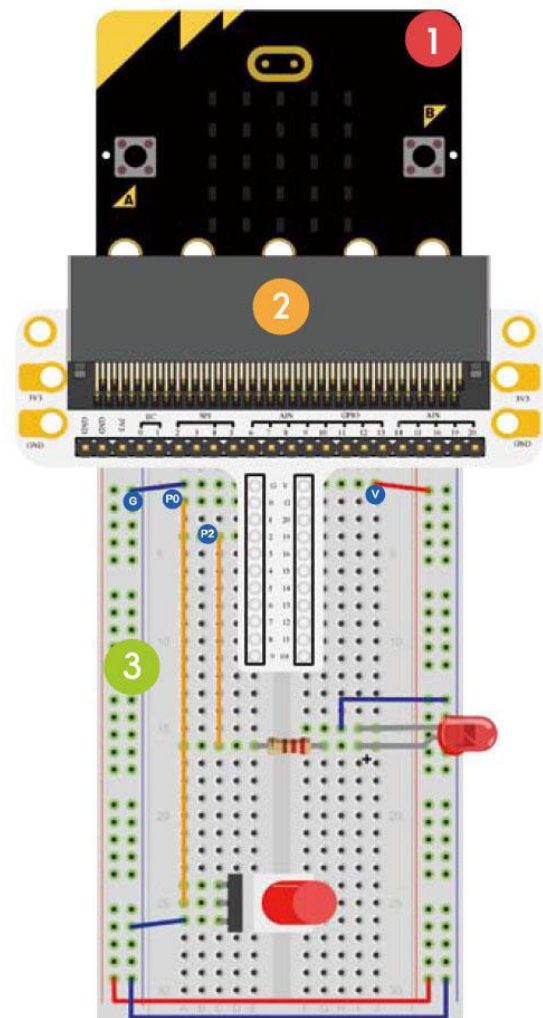
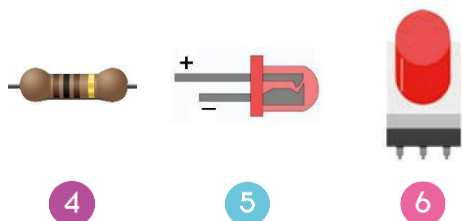


	
2.	De manera similar, compile los dos bloques <b>on button B/A+B pressed/al presionarse el botón B/A+B</b> para crear el código completo.
3.	Luego, compile el programa para generar un fichero .hex. Haga clic en el botón Download/descargar y almacene el fichero .hex file en la carpeta →Downloads (C:\Downloads). Este fichero .hex está listo para su subida a la micro:bit. Conecte micro:bit a un puerto USB. Ahora, arrastre y suelte el fichero .hex en la micro:bit [removable device] para subir el programa.

Pulse el botón A para encender el LED rojo. Pulse botón B para encender el LED verde. Pulse botón A y B simultáneamente para encender el LED azul. Ahora, ¿cómo crearía una luz RGB suave y gradual?

## 5.6 Interruptor de cierre automático

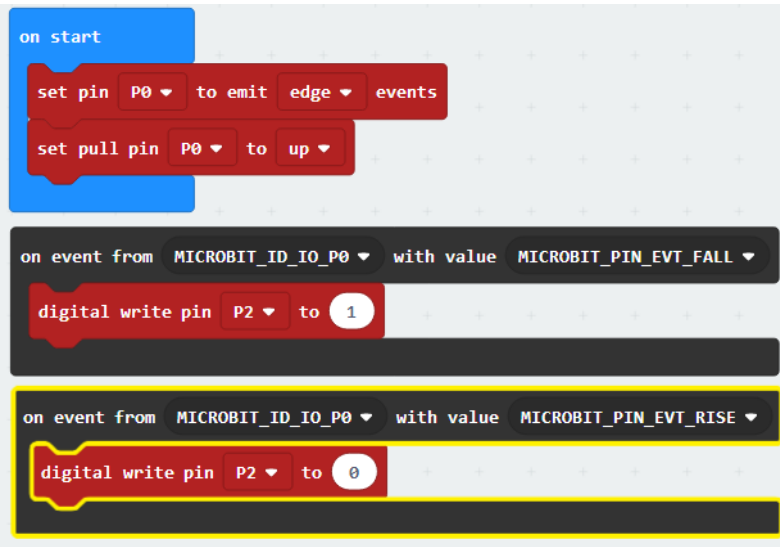
- 1 1x micro:bit
- 2 1x conector micro:bit para placa de pruebas
- 3 1x placa de pruebas
- 4 1x resistencia 100  $\Omega$  (marrón/negro/marrón/dorado)
- 5 1x LED rojo (polarización: ánodo (+) = pata larga, cátodo (-) = pata corta)
- 6 1x interruptor de cierre automático o interruptor biestable



Vamos a leer la temperatura ambiente (datos) del sensor de temperatura analógico y visualizar los datos en el micro:bit.

Coloque los componentes necesarios en la placa de pruebas (véase fig.).

Crear el código

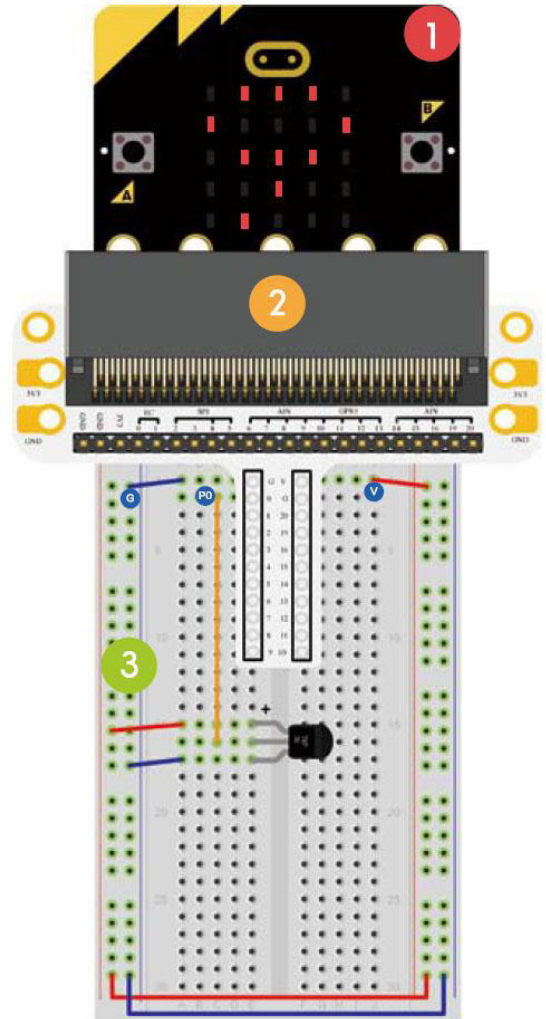
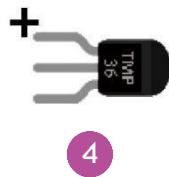


1.	Arrastre y suelte los bloques como se indica en la figura. El bloque <b>set pin P0 to emit edge events/establecer pin P0 para emitir eventos</b> y el bloque <b>set pull pin P0 to up/configurar pull en pin P0 a subir</b> se sitúan en el menú bajo <b>Pins/Pines</b> → <b>More</b> . Los bloques <b>on event from MICROBIT...EVT_FALL/RISE</b> se sitúan en el menú bajo <b>Control</b> .
2.	Seleccione <b>edge/borde</b> en el bloque "set pin P0 to emit edge events/establecer pin P0 para emitir eventos". Seleccione <b>P0</b> y a <b>up/subir</b> en el bloque "set pull pin P0 to up/configurar pull en pin P0 a subir".
3.	Vamos a crear un bloque de evento (flanco descendente). Arrastre y suelte el bloque <b>on event from MICROBIT...EVT°FALL</b> debajo del bloque <b>on start/al iniciar</b> . Haga clic en la primera flecha y seleccione la opción <b>MICROBIT_ID_IO_P0</b> en el menú desplegable. Haga clic en la segunda flecha y seleccione la opción <b>MICROBIT_PIN_EVENT_FALL</b> en el menú desplegable. Arrastre y suelte el bloque <b>digital write pin P0 to 0/escritura digital pin P0 a 0</b> en el bloque "on event from MICROBIT...EVT°FALL". Coloque pin P0 en P2 y coloque 0 en 1. El bloque <b>digital write pin P0 to 0/escritura digital pin P0 a 0</b> se sitúa en el menú bajo Advanced/Avanzado Pins/Pines.
4.	Repita estos pasos para el segundo bloque. Sin embargo, haga clic en la segunda flecha y seleccione la opción <b>MICROBIT_PIN_EVENT_RISE</b> en el menú desplegable. También, coloque pin P0 del bloque <b>digital write pin P0 to 0/escritura digital pin P0 a 0</b> en P2 y coloque el valor en 0.
5.	Luego, compile el programa para generar un fichero .hex. Haga clic en el botón Download/descargar y almacene el fichero .hex file en la carpeta →Downloads (C:\Downloads). Este fichero .hex está listo para su subida a la micro:bit. Conecte micro:bit a un puerto USB. Ahora, arrastre y suelte el fichero .hex en la micro:bit [removable device] para subir el programa.

Pulse el interruptor de cierre automático para encender el LED. Vuelva a pulsar para apagar el LED. Ahora, ¿cómo controlarías la pantalla del micro:bit con el interruptor de cierre automático?

## 5.7 Sensor de temperatura

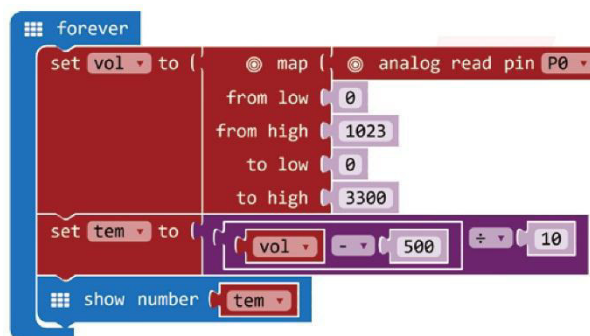
- 1 1x micro:bit
- 2 1x conector micro:bit para placa de pruebas
- 3 1x placa de pruebas
- 4 1x TMP36 sensor de temperatura



Vamos a leer la temperatura ambiente (datos) del sensor de temperatura analógico y visualizar los datos en el micro:bit.

Coloque los componentes necesarios en la placa de pruebas (véase fig.).

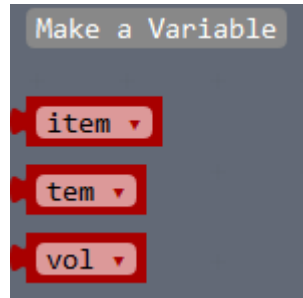
Crear el código



1.	<p>Arrastre y suelte los bloques como se indica en la figura.</p> <p>El bloque <b>set item to/establecer item para</b> se sitúa en el menú bajo <b>Variables</b>. El bloque <b>map/from low/from high/to low/to high/mapear/desde menor/desde mayor/a menor/a mayor</b> y el bloque <b>analog read pin/lectura analógica pin</b> se sitúan en el menú bajo <b>Pins/Pines</b>. El bloque con el <b>signo de menos</b> y el bloque con el <b>signo de división</b> se sitúan en el menú bajo <b>Math/Matemática</b>. El bloque <b>show number/mostrar número</b> se sitúa en el menú bajo <b>Basic/Básico</b>.</p>
2.	<p>Primero, vamos a crear dos variables. Seleccione <b>Variables</b> en el menú y haga clic en <b>Make a Variable</b>.</p> <p>New variable name:</p> <p>CalVa </p> <p>Ok ✓ Cancel ✕</p>

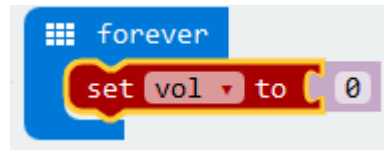
Introduzca **vol** en la ventana y haga clic en **Ok**. Introduzca **tem** en la ventana y haga clic en **Ok**.

Se visualizarán dos nuevos variables en el menú bajo **Variables**.

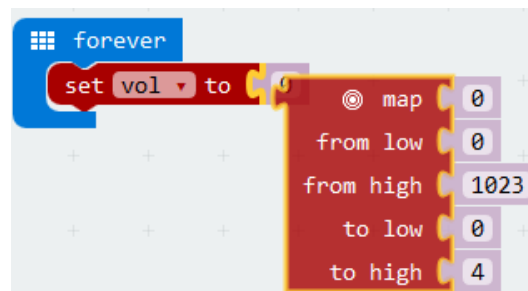


Más adelante, necesitará estos variables para almacenar los datos en el registro.

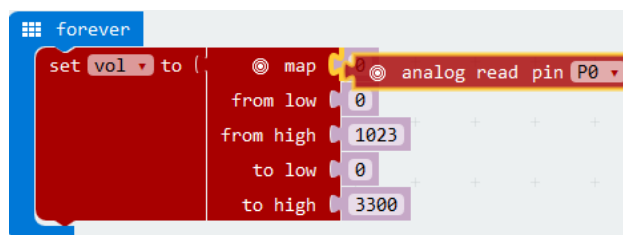
Arrastre y suelte el bloque **/establecer item para** en el bloque **forever/para siempre** y seleccione la opción **vol** con la flecha.



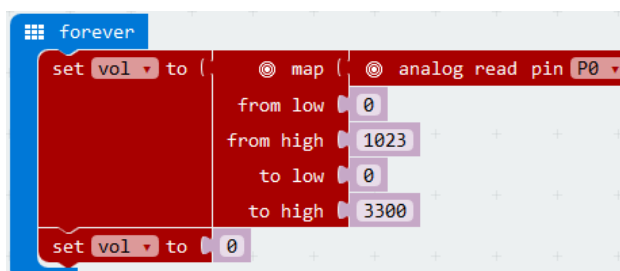
Luego, arrastre y suelte el bloque **map/from low/from high/to low/to high/mapear/desde menor/desde mayor/a menor/a mayor** sobre el 0 junto al bloque **set item to/establecer item para**.



Arrastre y suelte el bloque **analog read pin/lectura analógica pin** junto al bloque **map/from low/from high/to low/to high/mapear/desde menor/desde mayor/a menor/a mayor** y coloque el valor **to high/a mayor** 4 a 3300.

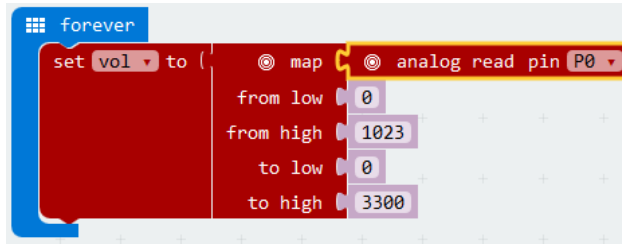


Ahora, arrastre y suelte el bloque **set item to/establecer item para** bajo el primer bloque. Coloque el variable **item** en **vol** con la flecha.



La tensión medida en mV a través de **analog read pin/lectura analógica pin P0** es un valor de 8-bit de 0-1023 (0-3.3 V) y está mapeada del valor

bajo (0 ó 0 V) a uno alto (3300 mV o 3.3 V). La tensión medida (mV) se almacenará en el variable **vol**.

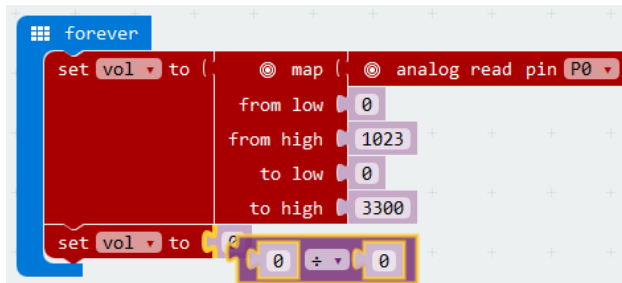


Ahora, vamos a escribir una fórmula para convertir la tensión medida en una temperatura:

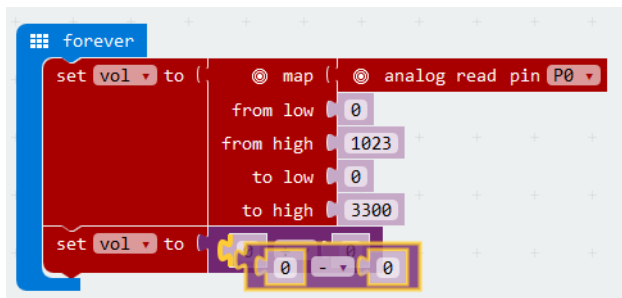
$$\text{Temperature } (^{\circ}\text{C}) = \frac{(\text{Output voltage (mV)} - 500)}{10}$$

En esta fórmula, la salida de tensión es el variable **vol** (se mide con el sensor TMP36). El resultado de la fórmula se almacenará en el variable **tem**.

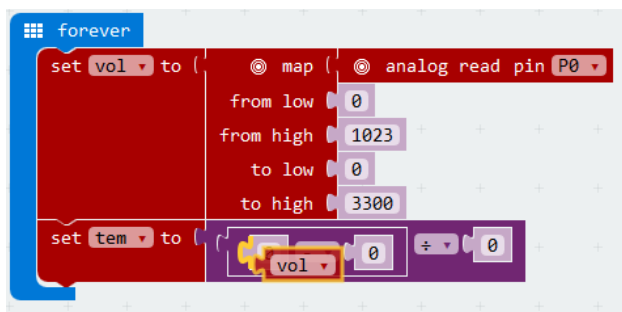
Vamos a escribir el código con bloques Arrastre y suelte el bloque con el **signo de división** sobre el 0 del bloque **set vol to/establecer vol para**.



Arrastre y suelte el bloque con el **signo de menos** sobre el primer 0 del bloque con el **signo de división**.

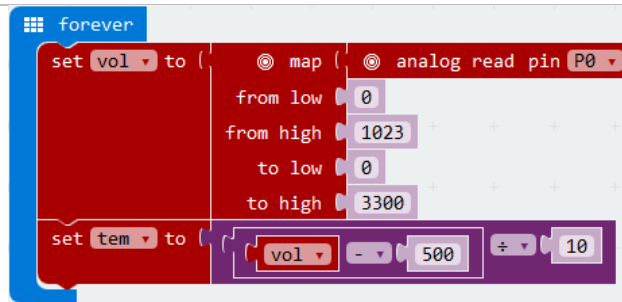


Ahora, cambie el segundo variable **vol** en **tem** arrastre y suelte el variable **vol** sobre el primer 0 del **signo de menos**. El variable **vol** se sitúa en el menú bajo **Variables**.

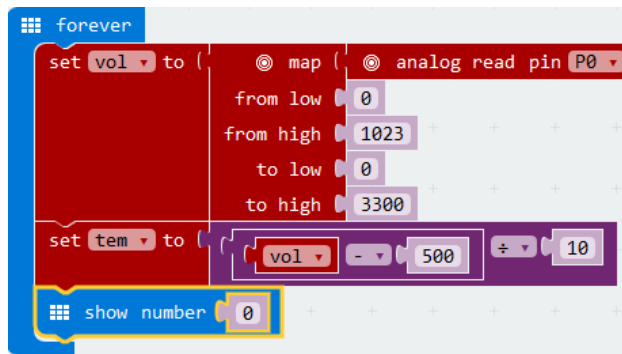


Coloque el 0 del bloque con el **signo de menos** en 500. También, coloque el valor del bloque con el **signo de división** en 10.

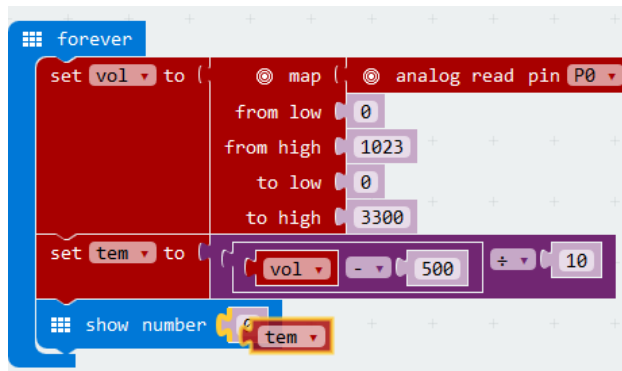




Arrastre y suelte el bloque **show number/mostrar número** bajo el bloque **set tem to/establecer tem para**.



Finalmente, arrastre y suelte el variable **tem** sobre el 0 del bloque **show number/mostrar número**.



¡El código está escrito!

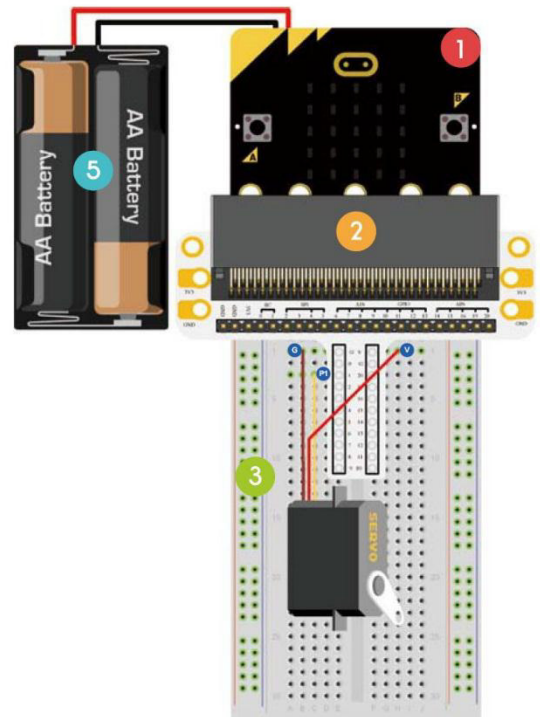
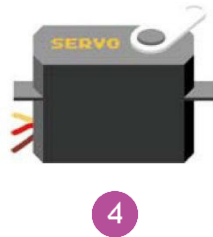
- Luego, compile el programa para generar un fichero .hex. Haga clic en el botón Download/descargar y almacene el fichero .hex file en la carpeta →Downloads (C:\Downloads). Este fichero .hex está listo para su subida a la micro:bit.

Conecte micro:bit a un puerto USB. Ahora, arrastre y suelte el fichero .hex en la micro:bit [removable device] para subir el programa.

Los dos LEDs parpadearán alternativamente. Ahora, ¿cómo visualizaría la temperatura, en Celsius o en Fahrenheit?

## 5.8 Servo

- 1 1x micro:bit
- 2 1x conector micro:bit para placa de pruebas
- 3 1x placa de pruebas
- 4 1x mini servo
- 5 1x portapilas con 2x pila AA de 1.5 V



Vamos a crear un servomotor que gire continuamente dentro de un rango de 0-180°.

Coloque los componentes necesarios en la placa de pruebas (véase fig.).

Crear el código

```

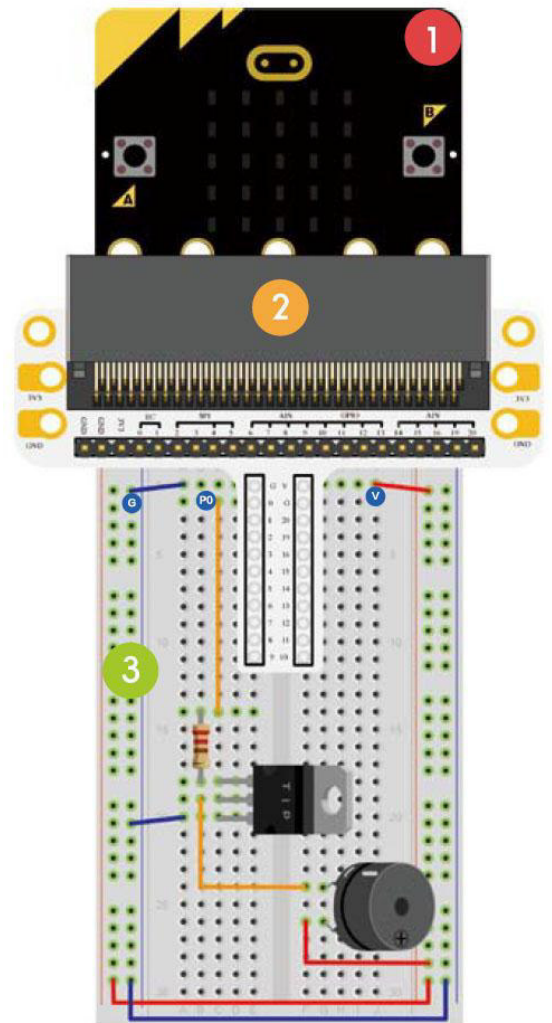
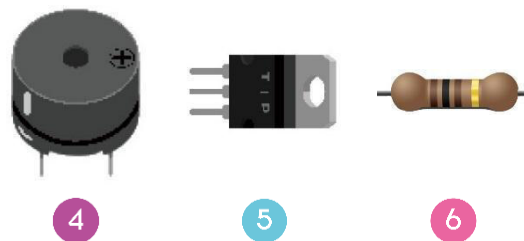
forever
  servo write pin P1 to 0
  pause (ms) 2000
  servo write pin P1 to 180
  pause (ms) 2000
  
```

1.	Arrastre y suelte los bloques como se indica en la figura. El bloque <b>digital write pin P0 to 180</b> se sitúa en el menú bajo <b>Advanced/Avanzado</b> Pins/Pines. El bloque <b>forever/para siempre</b> y el bloque <b>pause/pausa (ms) 100</b> se sitúan en el menú bajo <b>Basic/Básico</b> .
2.	Coloque <b>P0</b> en <b>P1</b> y coloque el valor en 0.
3.	Coloque el bloque <b>pause/pausa</b> en 2000 ms.
4.	Coloque <b>P0</b> en <b>P1</b> . Coloque el valor 0 en 180.
5.	Coloque el bloque <b>pause/pausa</b> en 2000 ms.
6.	Luego, compile el programa para generar un fichero .hex. Haga clic en el botón Download/descargar y almacene el fichero .hex file en la carpeta →Downloads (C:\Downloads). Este fichero .hex está listo para su subida a la micro:bit. Conecte micro:bit a un puerto USB. Ahora, arrastre y suelte el fichero .hex en la micro:bit [removable device] para subir el programa.

El servo girará de 0 a 180°. Ahora, ¿cómo crearía un termómetro analógico con un sensor de temperatura y un servo?

## 5.9 Zumbador

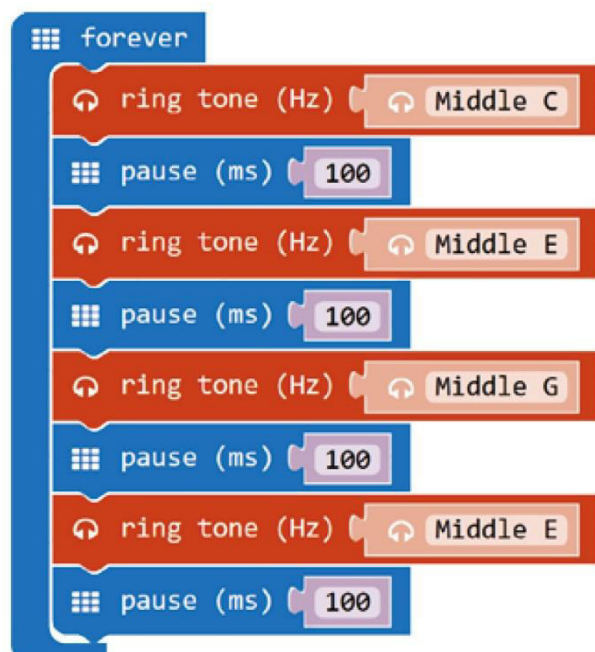
- 1 1x micro:bit
- 2 1x conector micro:bit para placa de pruebas
- 3 1x placa de pruebas
- 4 1x mini altavoz
- 5 1x MOSFET de canal N
- 6 1x resistencia 100  $\Omega$  (marrón/negro/marrón/dorado)



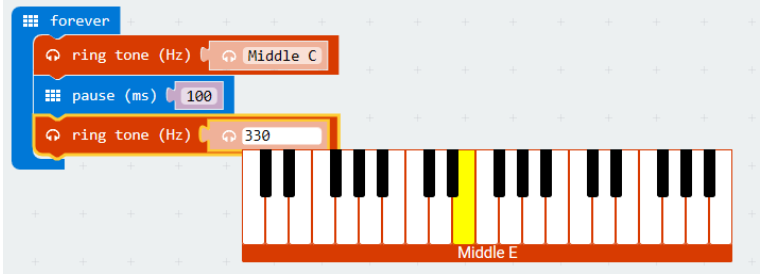
Vamos a controlar un zumbador.

Coloque los componentes necesarios en la placa de pruebas (véase fig.).

Crear el código



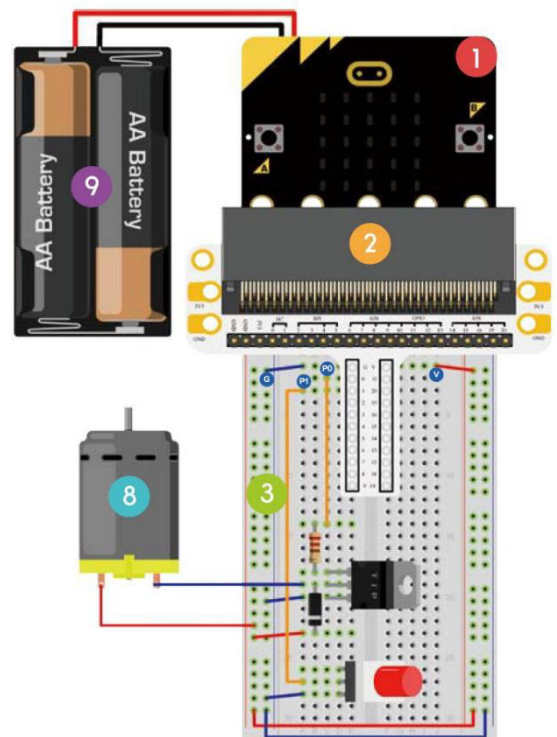
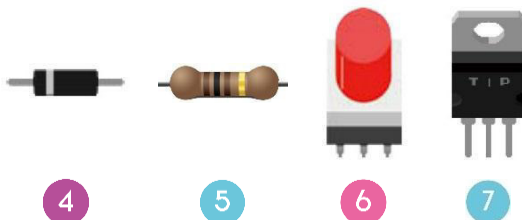
1. Arrastre y suelte los bloques como se indica en la figura. El bloque **forever/para siempre** y el bloque **pause/pausa (ms) 100** se sitúan en el menú bajo **Basic/Básico**. El bloque **ring tone (Hz)/tono de timbre (Hz)** en el menú bajo **Music/Música**.

2.	<p>Coloque el primer <b>ring tone (Hz)/tono de timbre (Hz)</b> en <b>Do medio</b>. Haga clic en la casilla y seleccione la tecla deseada.</p>  <p>Oirá el tono correspondiente.</p>
3.	Coloque el bloque <b>pause/pausa</b> en 100 ms.
4.	Repita estos pasos para los otros tonos y pausas.
5.	<p>Luego, compile el programa para generar un fichero .hex. Haga clic en el botón Download/descargar y almacene el fichero .hex file en la carpeta →Downloads (C:\Downloads). Este fichero .hex está listo para su subida a la micro:bit.</p> <p>Conecte micro:bit a un puerto USB. Ahora, arrastre y suelte el fichero .hex en la micro:bit [removable device] para subir el programa.</p>

El zumbador emitirá una señal acústica. Ahora, ¿cómo programaría su nana favorita?

## 5.10 Motor

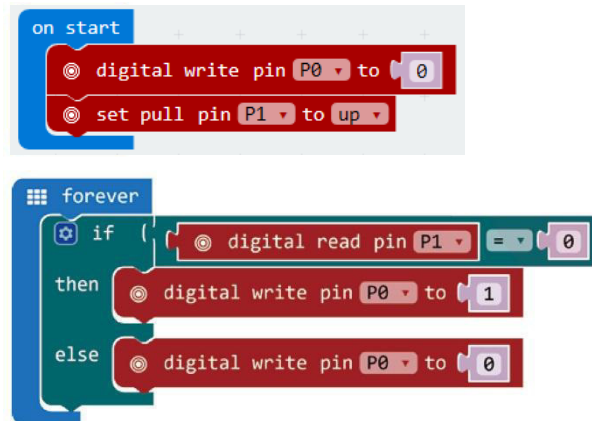
- 1 1x micro:bit
- 2 1x conector micro:bit para placa de pruebas
- 3 1x placa de pruebas
- 4 1x diodo
- 5 1x resistencia 100  $\Omega$  (marrón/negro/marrón/dorado)
- 6 1x interruptor de cierre automático o interruptor biestable
- 7 1x MOSFET de canal N
- 8 1x mini motor
- 9 1x portapilas con 2x pila AA de 1.5 V



Vamos a utilizar un interruptor para controlar el arranque y la parada de un motor.

Coloque los componentes necesarios en la placa de pruebas (véase fig.).

Crear el código



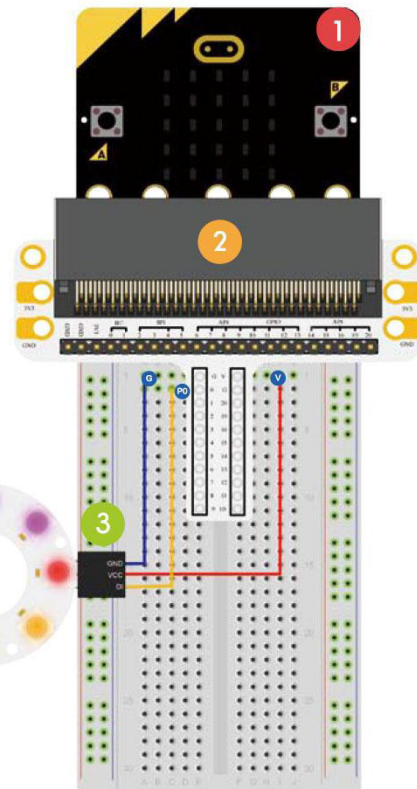
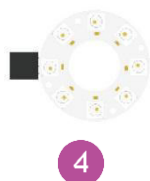
1.	Arrastre y suelte los bloques como se indica en la figura.
2.	El bloque <b>on start/al iniciar</b> se efectuará sólo una vez para iniciar el programa.
3.	Coloque el valor de <b>P0</b> en <b>0</b> .
4.	Coloque el bloque <b>set pull pin P0 to up/configurar pull en pin P0 a subir</b> en <b>P1</b> y <b>a subir</b> .
5.	Coloque el <b>digital read pin/lectura digital pin</b> en <b>P1</b> en el bloque <b>forever/para siempre</b> . Coloque el bloque con el <b>signo igual</b> en 0. En realidad, esto es el esquema. No es necesario fijar una resistencia pull-up a la placa de pruebas. La función pull-up está programada (véase paso 3-4) en el código y reemplaza la necesidad de una resistencia en la placa de pruebas.
6.	Después de haber pulsado el botón, coloque la tensión alta en P0. Coloque el valor 0 en 1 (5 V). El motor arranca.
7.	Después de haber soltado el botón, coloque la tensión baja en P0. Coloque el valor en 0. El motor para.
8.	Luego, compile el programa para generar un fichero .hex. Haga clic en el botón Download/descargar y almacene el fichero .hex file en la carpeta →Downloads (C:\Downloads). Este fichero .hex está listo para su subida a la micro:bit. Conecte micro:bit a un puerto USB. Ahora, arrastre y suelte el fichero .hex en la micro:bit [removable device] para subir el programa.

Pulse el botón para arrancar el motor y suelte el botón para pararlo. Ahora, ¿cómo utilizaría un potenciómetro de ajuste para controlar la velocidad del motor?

Observación: Porque la tensión del micro:bit es de 3.3 V, puede que no sea suficiente para soportar el ventilador. Si fuera necesario, gire la pala para para hacer que el ventilador arranque.

## 5.11 LED arco iris

- 1 1x micro:bit
- 2 1x conector micro:bit para placa de pruebas
- 3 1x placa de pruebas
- 4 1x anillo LED RGB



Vamos a controlar 8 LEDs RGB en un anillo y realizar un arco iris gradual.

Coloque los componentes necesarios en la placa de pruebas (véase fig.).

Crear el código

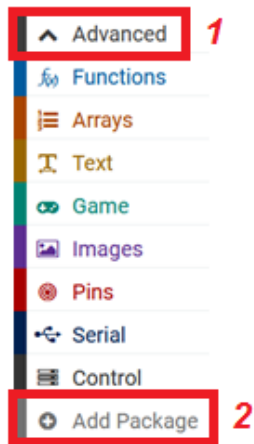
```

on start
  set strip to NeoPixel at pin P0 with 8 leds as RGB (GRB format)
  strip show rainbow from 1 to 360

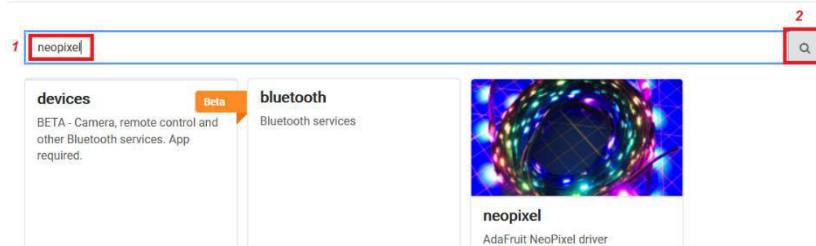
forever
  strip show
  strip rotate pixels by 1
  pause (ms) 100
  
```

1.	Arrastre y suelte los bloques como se indica en la figura.
2.	Busque y añada la biblioteca NeoPixel. Seleccione Advanced/Avanzado → Extensions /Extensiones e introduzca NeoPixel.



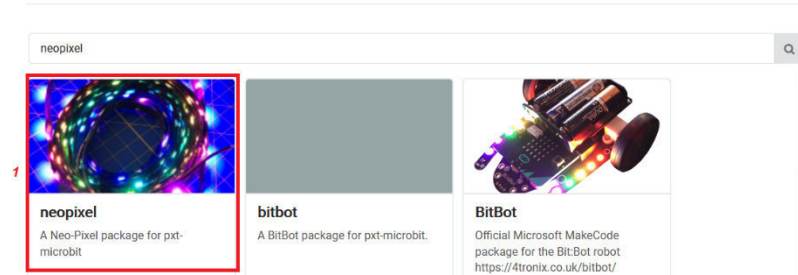


Add Package... ?

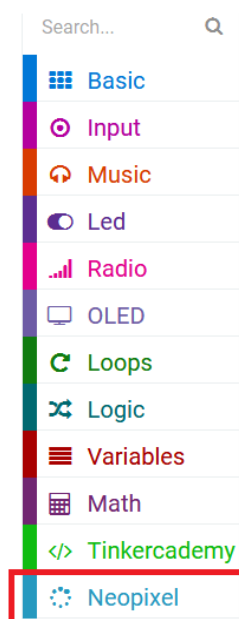


Luego, seleccione la biblioteca NeoPixel.


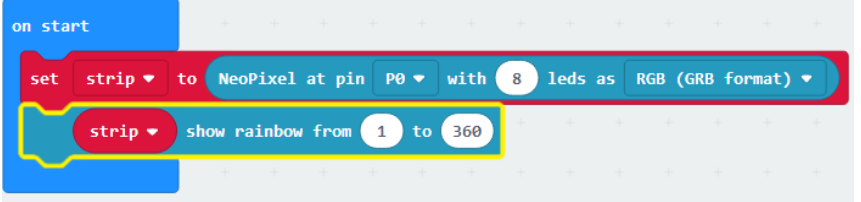
Add Package... ?



La biblioteca está descargada y añadida al menú de bloques.



Los bloques **NeoPixel at pin P0 with 24 leds, item show rainbow from 1 to 360, item show** y **items rotate pixels by 1** se sitúan en el menú de bloques bajo **NeoPixel**.

3.	<p>Arrastre y suelte el bloque <b>set strip to NeoPixel at pin P0 with 24 leds as RGB (GRB format)</b> en el bloque <b>on start/al iniciar</b>. Coloque el pin en <b>P0</b> y el valor 24 en 8 (el anillo LED lleva 8 LEDs).</p> 
4.	<p>Ahora, arrastre y suelte el bloque <b>set strip to rainbow from 1 to 360</b> bajo el bloque <b>set strip to NeoPixel at pin P0</b>.</p> 
5.	<p>Complete el código (véase fig.).</p>
6.	<p>Luego, compile el programa para generar un fichero .hex. Haga clic en el botón Download/descargar y almacene el fichero .hex file en la carpeta →Downloads (C:\Downloads). Este fichero .hex está listo para su subida a la micro:bit.</p> <p>Conecte micro:bit a un puerto USB. Ahora, arrastre y suelte el fichero .hex en la micro:bit [removable device] para subir el programa.</p>

En el anillo LED se visualizará un arco iris que gira. Ahora, ¿cómo crearía un ojo parpadeante con el anillo LED?

**Utilice este aparato sólo con los accesorios originales. Velleman NV no será responsable de daños ni lesiones causados por un uso (indebido) de este aparato. Para más información sobre este producto y la versión más reciente de este manual del usuario, visite nuestra página [www.velleman.eu](http://www.velleman.eu). Se pueden modificar las especificaciones y el contenido de este manual sin previo aviso.**

#### © DERECHOS DE AUTOR

**Velleman NV dispone de los derechos de autor para este manual del usuario. Todos los derechos mundiales reservados.** Está estrictamente prohibido reproducir, traducir, copiar, editar y guardar este manual del usuario o partes de ello sin el consentimiento previo por escrito del propietario del copyright.

# BEDIENUNGSANLEITUNG

## 1. Einführung

**An alle Einwohner der Europäischen Union**

**Wichtige Umweltinformationen über dieses Produkt**



Dieses Symbol auf dem Produkt oder der Verpackung zeigt an, dass die Entsorgung dieses Produktes nach seinem Lebenszyklus der Umwelt Schaden zufügen kann. Entsorgen Sie die Einheit (oder verwendeten Batterien) nicht als unsortierter Hausmüll; die Einheit oder verwendeten Batterien müssen von einer spezialisierten Firma zwecks Recycling entsorgt werden. Diese Einheit muss an den Händler oder ein örtliches Recycling-Unternehmen retourniert werden. Respektieren Sie die örtlichen Umweltvorschriften.

**Falls Zweifel bestehen, wenden Sie sich für Entsorgungsrichtlinien an Ihre örtliche Behörde.**

Vielen Dank, dass Sie sich für Velleman entschieden haben®! Lesen Sie diese Bedienungsanleitung vor Inbetriebnahme sorgfältig durch. Überprüfen Sie, ob Transportschäden vorliegen. Sollte dies der Fall sein, verwenden Sie das Gerät nicht und wenden Sie sich an Ihren Händler.

## 2. Sicherheitshinweise



Dieses Gerät kann von Kindern ab 8 Jahren und darüber sowie von Personen mit verringerten physischen, sensorischen oder mentalen Fähigkeiten oder Mangel an Erfahrung und Wissen benutzt werden, wenn sie beaufsichtigt oder bezüglich des sicheren Gebrauchs des Gerätes unterwiesen wurden und die daraus resultierenden Gefahren verstehen. Kinder sollten nicht mit dem Gerät spielen. Beachten Sie, dass Kinder das Gerät nicht reinigen oder bedienen.



Das Gerät eignet sich nur für die Anwendung im Innenbereich.

Schützen Sie das Gerät vor Regen und Feuchte. Setzen Sie das Gerät keiner Flüssigkeit wie z.B. Tropf- oder Spritzwasser, aus.

## 3. Allgemeine Richtlinien



- Siehe Velleman® Service- und Qualitätsgarantie am Ende dieser Bedienungsanleitung.
- Nehmen Sie das Gerät erst in Betrieb, nachdem Sie sich mit seinen Funktionen vertraut gemacht haben.
- Eigenmächtige Veränderungen sind aus Sicherheitsgründen verboten. Bei Schäden verursacht durch eigenmächtige Änderungen erlischt der Garantieanspruch.
- Verwenden Sie das Gerät nur für Anwendungen beschrieben in dieser Bedienungsanleitung. Bei falscher Anwendung dieses Gerätes erlischt der Garantieanspruch.
- Bei Schäden, die durch Nichtbeachtung der Bedienungsanleitung verursacht werden, erlischt der Garantieanspruch. Für daraus resultierende Folgeschäden übernimmt der Hersteller keine Haftung.
- Weder Velleman nv noch die Händler können für außergewöhnliche, zufällige oder indirekte Schäden irgendwelcher Art (finanziell, physisch, usw.), die durch Besitz, Gebrauch oder Defekt verursacht werden, haftbar gemacht werden.
- Durch ständige Verbesserungen, kann das Produkt von den Abbildungen abweichen.
- Die Abbildungen dienen nur zur Illustration.
- Das Gerät bei Temperaturschwankungen nicht sofort einschalten. Schützen Sie das Gerät vor Beschädigung, indem Sie es ausgeschaltet lassen bis es die Zimmertemperatur erreicht hat.
- Bewahren Sie diese Bedienungsanleitung für künftige Einsichtnahme auf.

## 4. Beschreibung

Dieser Bausatz für Anfänger ist ein Lernbausatz basierend auf micro:bit. Der Bausatz wird mit Basiskomponenten, einer Steckplatine, Anschlusskabel und einem micro:bit geliefert.

## 5. Einige Beispiele

### 5.1 LED

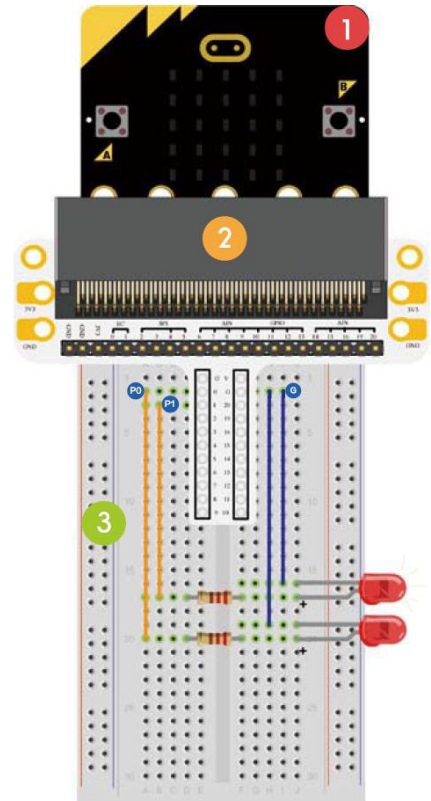
- 1 1x micro:bit
- 2 1x micro:bit-Anschluss für Steckplatine
- 3 1x Steckplatine
- 4 2x rote LED (Polarisation: Anode (+) = langes Bein, Kathode (-) = kurzes Bein)
- 5 2x Widerstand 100 Ω (braun/schwarz/braun/vergoldet)



4



5



In dieser Anleitung, verwenden wir einen micro:bit, um zwei LEDs abwechselnd blinken zu lassen.

Befestigen Sie die benötigten Komponenten an der Steckplatine (siehe Abb.).

Die Programmierumgebung MakeCode ist online verfügbar und bietet eine leicht verständliche Oberfläche, die visuell unterstützte Blockprogrammierung verwendet. Öffnen Sie in einem Browser die folgende URL: [www.makecode.com](http://www.makecode.com) oder <https://www.microsoft.com/en-us/makecode?rtc=1> und klicken Sie auf die Ikone micro:bit icon und danach auf **Start Project/Neues Projekt**.

Microsoft MakeCode brings computer science to life for all students with fun projects, immediate results, and both block and text editors for learners at different levels.

**micro:bit**  
Start coding with micro:bit >

**Circuit Playground Express**  
Start coding with Circuit Playground Express >

**Minecraft**  
Start coding with Minecraft >

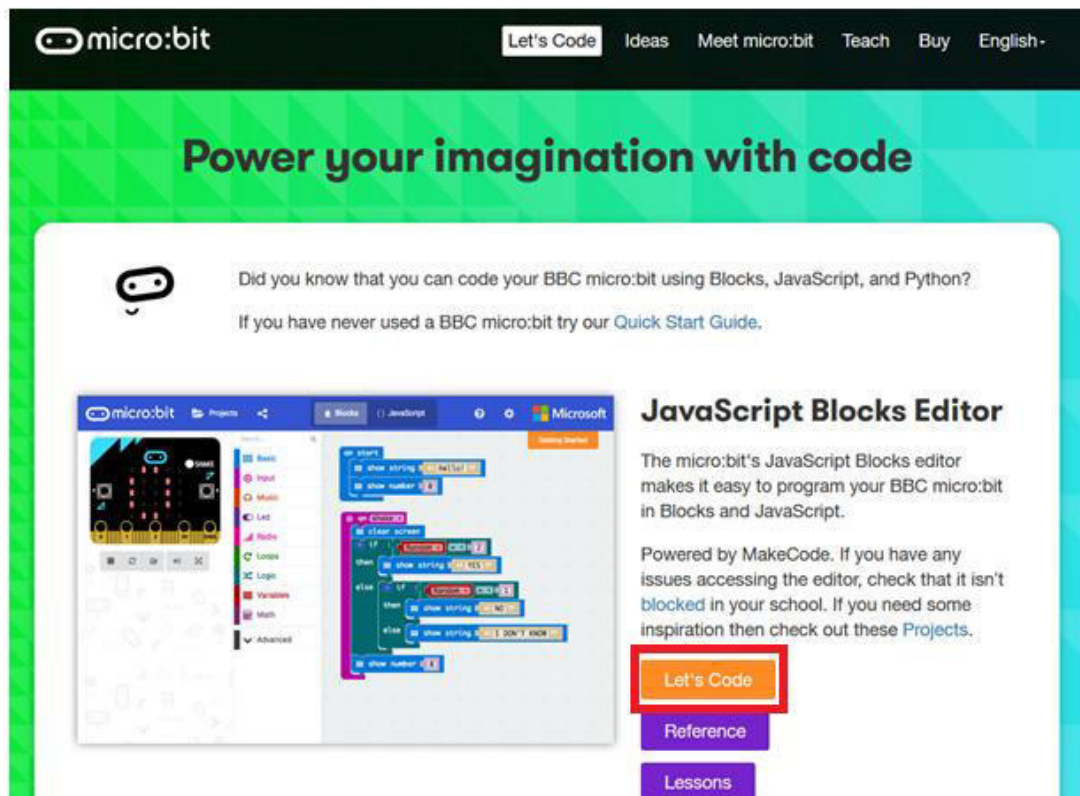
Der micro:bit Block-Editor wird in einem neuen Fenster geöffnet. Sie können nun mit den Blöcken einen Code erstellen, indem Sie diese per Drag and Drop auf die Programmieroberfläche ziehen.

Wie es funktioniert entdecken Sie auf den folgenden Seiten.

## Was ist MakeCode?

Früher PXT – Programming eXperience Toolkit, um einen Code für micro:bit zu schreiben.

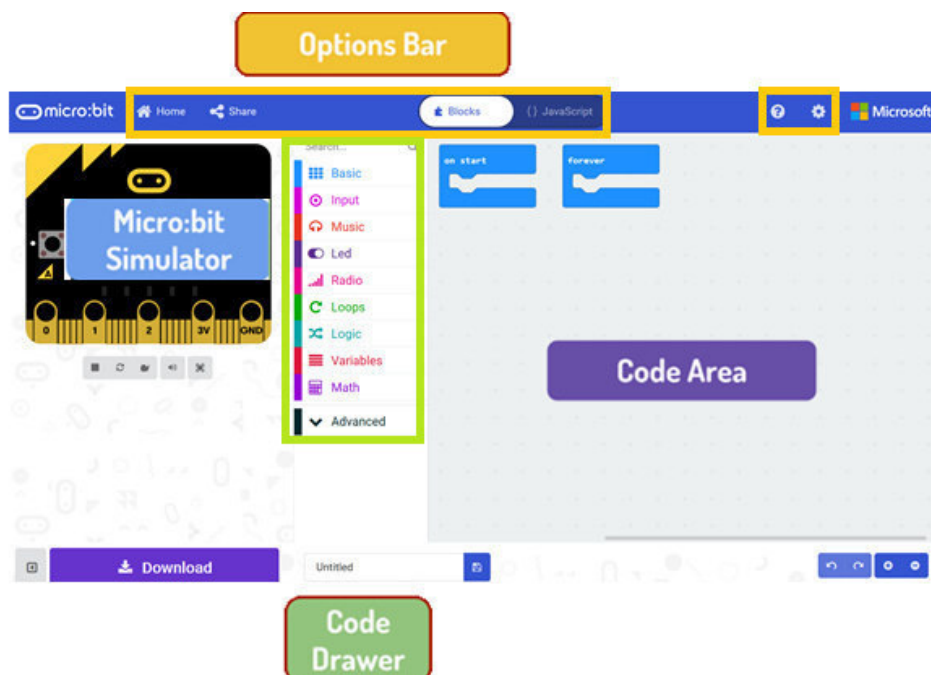
Die Grafische Programmierung ähnelt Scratch. MakeCode funktioniert auf allen Browsern. Es wird nur einen Computer mit Internetverbindung benötigt.



Block-Editor Um einen Code zu kreieren, müssen Sie die Blöcke aus dem **Blockmenü** (in der Mitte) auf die **Programmieroberfläche** (links) ziehen. Links oben gibt es den **micro:bit-Simulator**, wo Sie sofort das Ergebnis sehen können. Unten befindet sich den Button zum Herunterladen und Speichern des Projekts.



Der folgende Screenshot zeigt die **Menüleiste**.



## Menüleiste



Ein neues Projekt erstellen oder hinzufügen

Projekte importieren.

Mit mit vordefinierten Blöcken programmieren.

Mit echten Codezeilen programmieren (JavaScript).

Hilfebereich

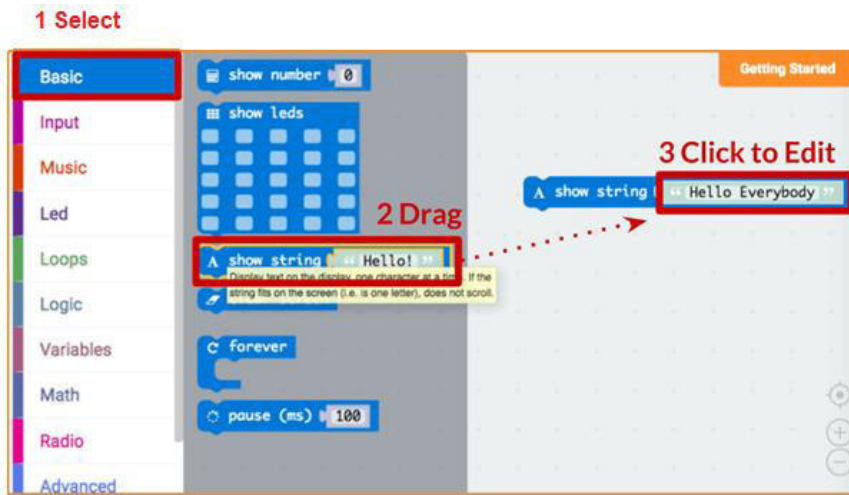
Konfigurationsbereich Projekte einen Namen geben und löschen **Reset**: mit dieser Option löschen Sie alle gespeicherten Projekte. Seien Sie vorsichtig! Sie benötigen meistens nur die Option **Delete Project/Projekt löschen**.

## Die LED-Leiste

Lassen wir einen Code schreiben. Im Block-Menü finden Sie alle zur Verfügung stehenden Blöcke. Verschieben Sie den Block **show string/Zeige Zeichenfolge** in die Programmieroberfläche und klicken Sie das Kästchen an, um den Text zu bearbeiten.

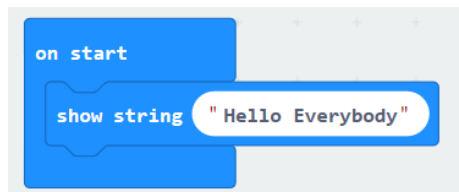






1	Auswählen
2	Verschieben

3	Klicken, um zu bearbeiten
---	---------------------------

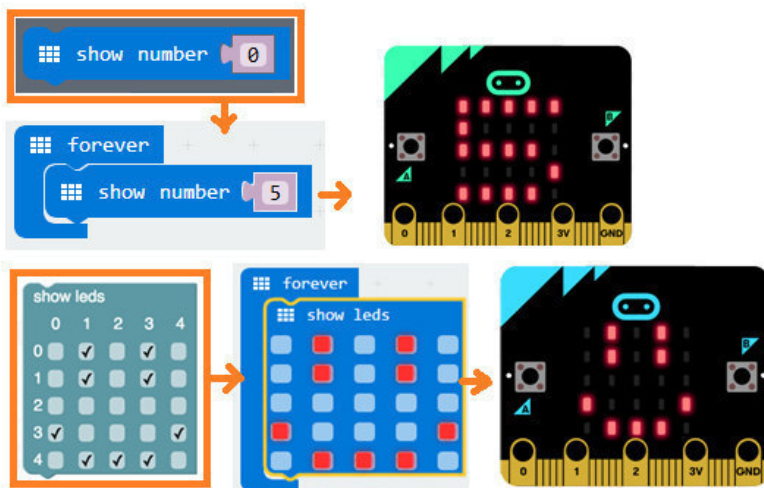


Im micro:bit-Simulator können Sie das Ergebnis sehen.

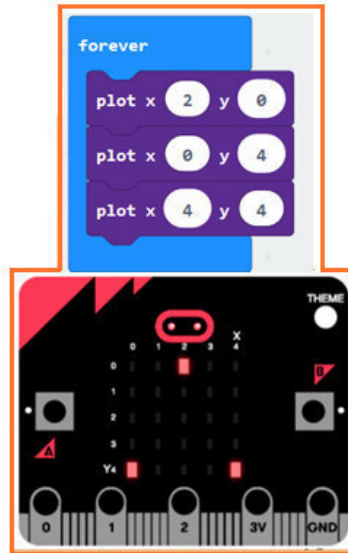
### Say Hello!

Der angezeigte Text nennt man **string/Zeichenfolge**.

Mit der Funktion **show number/zeige Nummer** werden Ganzzahlen im LED-Bildschirm angezeigt. Mit der Funktion **show leds/zeige LEDs** wird ein Bild (5x5 Pixel) angezeigt.



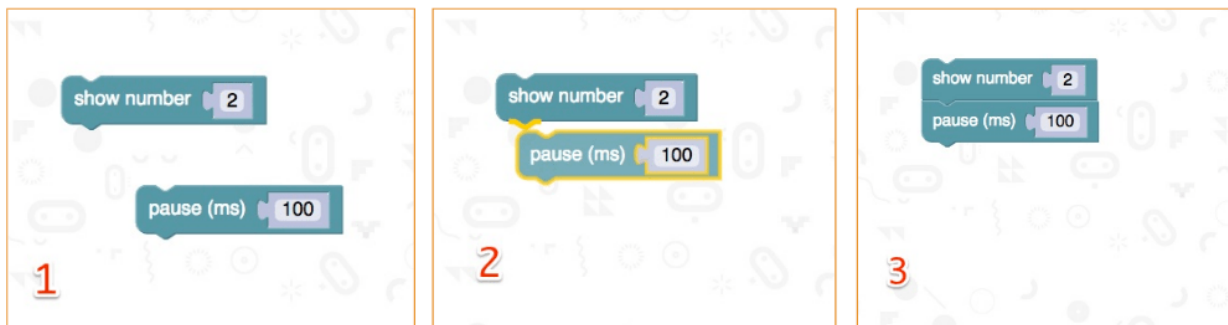
Sie können auch einen LED-Punkt leuchten lassen, indem Sie Koordinaten verwenden. Die Koordinate (0,0) befindet sich links oben.



### Blöcke aneinanderreihen

Klicken Sie auf den gewünschten Block und halten Sie ihn fest. Verschieben Sie den einen Block so bis einer Seite des anderen Blocks gelb unterlegt ist. Lassen Sie nun los. Beide Blöcke rasten ein.

Klicken Sie den ersten Block an, dann verschiebt der zweite Block. Klicken Sie den zweiten Block an, dann löst er sich vom ersten Block.



### micro:bit

Verbinden Sie den micro:bit über das mitgelieferte mit Micro-USB-Kabel Ihrem Rechner.

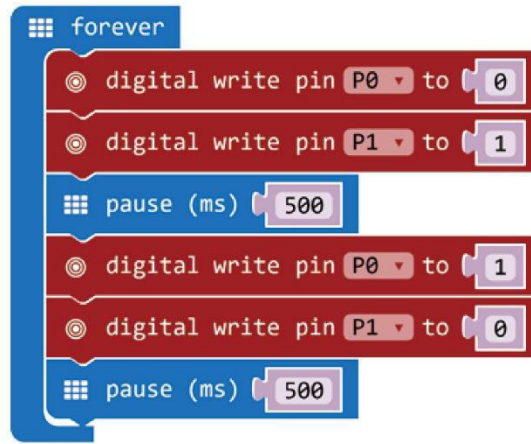
Standardmäßig wird eine Sequenz wiedergegeben. Es wird Ihnen gefragt, Tasten zu drücken und ein Spiel zu spielen. Versuchen Sie!

Klicken Sie auf **Download/Herunterladen**, um das Programm in eine Hex-Datei zu konvertieren und es herunterzuladen.

Sie benötigen den nächsten Schritt nicht, wenn Chrome korrekt konfiguriert ist. Entweder verschieben Sie die heruntergeladene Hex-Datei in das micro:bit-Laufwerk oder rechtsklicken Sie **Send To** in Windows®.

Sie haben das Gerät nun programmiert.

Einen Code erstellen

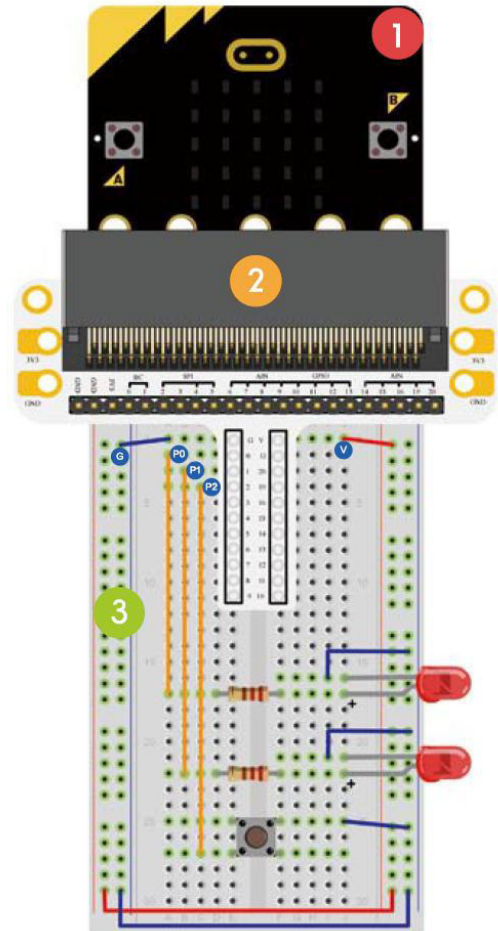
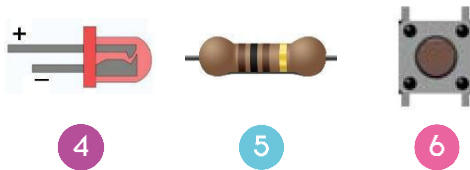


1.	Verschieben Sie die Blöcke per Drag and Drop (siehe Abb.). Den Block <b>digital write pin P0 to 0/schreibe digitalen Wert von Pin P0 auf 0</b> finden Sie im Blockmenü unter <b>Advanced/Fortgeschritten</b> → <b>Pins</b> . Die Blöcke <b>forever/dauerhaft</b> und <b>pause/pausiere (ms) 100</b> finden Sie im Blockmenü unter <b>Basic/Grundlagen</b> .
2.	Stellen Sie den Wert für P0 auf 0. LED0 aus = niedrige Spannung = 0 V = digital 0. Stellen Sie den Wert für P1 auf 1. LED1 ein = hohe Spannung = 5 V = digital 1.
3.	Stellen Sie den Block <b>pause/pausiere</b> auf 500 ms.
4.	Stellen Sie den Wert für P0 auf 1. LED0 ein = hohe Spannung = 5 V = digital 1. Stellen Sie den Wert für P1 auf 0. LED1 aus = niedrige Spannung = 0 V = digital 0.
5.	Stellen Sie den Block <b>pause/pausiere</b> auf 500 ms.
6.	Kompilieren Sie das Programm und speichern Sie es als .Hex-Datei. Klicken Sie danach auf Download/Herunterladen und speichern Sie die .hex-Datei im Ordner →Downloads (C:\downloads). Diese .hex-Datei kann nun auf das micro: bit übertragen werden. Verbinden Sie den micro:bit mit einem USB-Port. Verschieben Sie die .hex-Datei nun per Drag and Drop in micro:bit [removable device], um das Programm hochzuladen.

Die zwei LEDs werden abwechselnd blinken. Wir werden nun eine RGB-Ampel kreieren.

## 5.2 Taste

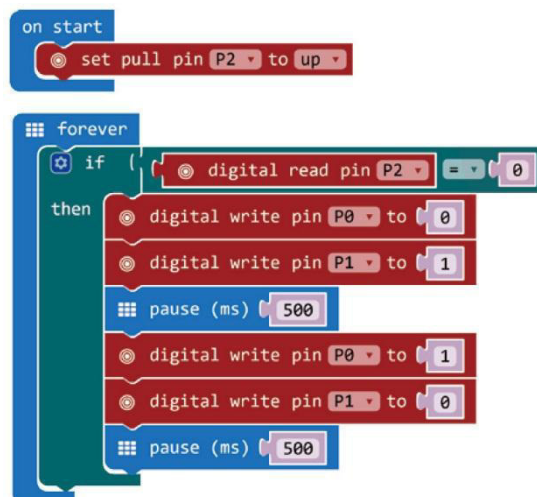
- 1 1x micro:bit
- 2 1x micro:bit-Anschluss für Steckplatine
- 3 1x Steckplatine
- 4 2x rote LED (Polarisation: Anode (+) = langes Bein, Kathode (-) = kurzes Bein)
- 5 2x Widerstand 100  $\Omega$  (braun/schwarz/braun/vergoldet)
- 6 1x momentaner Druckknopf



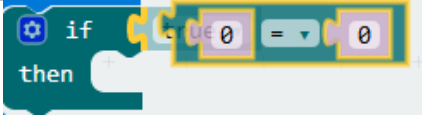
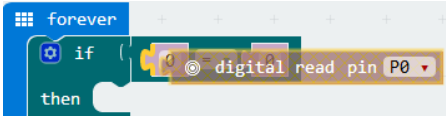
Wir verwenden einen Druckknopf, um das Blinken der LEDs anzusteuern. Drücken Sie den Knopf, um die LEDs abwechselnd blinken zu lassen. Lassen Sie den Knopf los, um die LEDs auszuschalten.

Befestigen Sie die benötigten Komponenten an der Steckplatine (siehe Abb.).

Einen Code erstellen



1. Verschieben Sie die Blöcke per Drag and Drop (siehe Abb.). Den Block **digital read pin P0/digitale Werte von Pin P0** und den Block **digital write pin P0/schreibe digitalen Wert von Pin P0 auf 0** finden Sie im Blockmenü unter **Advanced/Fortgeschritten** → **Pins**. Den Block **forever/dauerhaft** und den Block **pause/pausiere (ms) 100** finden Sie im Blockmenü unter **Basic/Grundlagen**. Den Block **if then/wenn dann** und den Block mit dem **Gleichheitszeichen** finden Sie im Blockmenü unter **Logic/Logik**. Verschieben Sie den Block mit dem **Gleichheitszeichen** über den Block **true/wahr**.

	 <p>Die zwei Blöcke werden einrasten. Verschieben Sie den Block <b>digital read pin P0/digitale Werte von Pin P0</b> über die Ziffer <b>0</b> vom Block mit dem <b>Gleichheitszeichen</b> (siehe Abb.).</p> 
2.	Stellen Sie P2 als Pull-Up-Widerstand ein.
3.	Stellen Sie <b>digital read pin/digitale Werte von Pin</b> auf P2.
4.	Stellen Sie <b>digital write pin P0/schreibe digitalen Wert von Pin P0 auf 0</b> (aktiv niedrig (0 V)). Stellen Sie <b>digital write pin P1/schreibe digitalen Wert von Pin P1 auf 1</b> (aktiv hoch (5 V)).
5.	Stellen Sie den Block <b>pause/pausiere</b> auf 500 ms.
6.	Stellen Sie <b>digital write pin P0/schreibe digitalen Wert von Pin P0 auf 1</b> (aktiv hoch (5 V)). Stellen Sie <b>digital write pin P1/schreibe digitalen Wert von Pin P1 auf 0</b> (aktiv niedrig (0 V)).
7.	Stellen Sie den Block <b>pause/pausiere</b> auf 500 ms.
8.	Kompilieren Sie das Programm und speichern Sie es als .Hex-Datei. Klicken Sie danach auf Download/Herunterladen und speichern Sie die .hex-Datei im Ordner →Downloads (C:\downloads). Diese .hex-Datei kann nun auf das micro:bit übertragen werden. Verbinden Sie den micro:bit mit einem USB-Port. Verschieben Sie die .hex-Datei nun per Drag and Drop in micro:bit [removable device], um das Programm hochzuladen.

Drücken Sie die Taste und die LEDs werden abwechselnd blinken. Wie werden Sie nun dafür sorgen, dass die rote LED leuchtet, wenn die Taste gedrückt ist und, dass die grüne LED leuchtet, wenn die Taste nicht gedrückt ist.

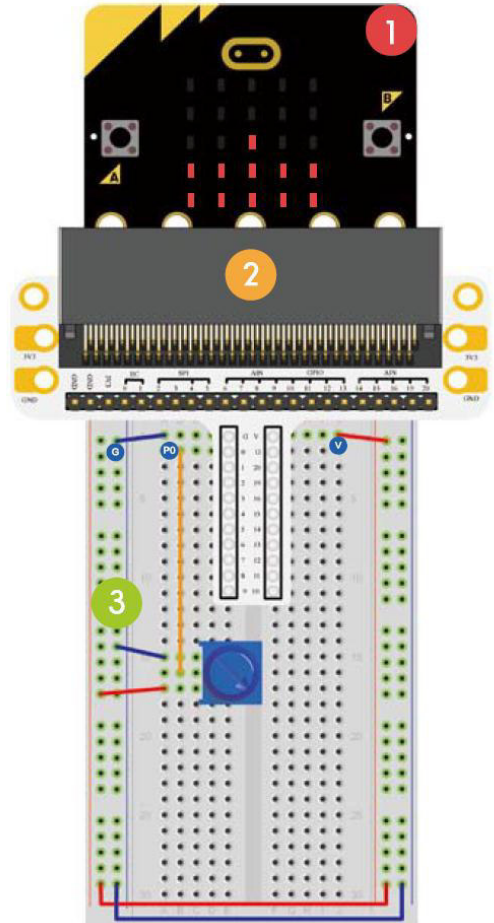
### 5.3 Trimpotentiometer

- 1 1x micro:bit
- 2 1x micro:bit-Anschluss für Steckplatine
- 3 1x Steckplatine
- 4 1x 10 k $\Omega$  Trimpotentiometer



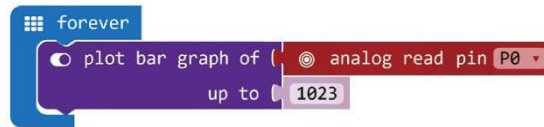
Wir werden die Ausgangsspannung vom Trimpotentiometer lesen und mit einem Säulendiagramm auf dem Bildschirm des micro:bit anzeigen.

Befestigen Sie die benötigten Komponenten an der Steckplatine (siehe Abb.).





Einen Code erstellen



1.	Verschieben Sie die Blöcke per Drag and Drop (siehe Abb.). Den Block <b>plot bar graph of 0 up to 0</b> finden Sie im Blockmenü unter <b>LED</b> .
2.	Stellen Sie <b>analog read pin/analoge Werte von Pin</b> auf P0. Diesen Block finden Sie im Blockmenü unter <b>Pins</b> . Geben Sie den Wert 1023 ein.
3.	Kompilieren Sie das Programm und speichern Sie es als .Hex-Datei. Klicken Sie danach auf Download/Herunterladen und speichern Sie die .hex-Datei im Ordner →Downloads (C:\downloads). Diese .hex-Datei kann nun auf das micro:bit übertragen werden. Verbinden Sie den micro:bit mit einem USB-Port. Verschieben Sie die .hex-Datei nun per Drag and Drop in micro:bit [removable device], um das Programm hochzuladen.

Drehen Sie das Trimpotentiometer. Die Spannung wird mit einem Säulendiagramm auf dem Bildschirm des micro:bit angezeigt. Ist die Spannung 0, dann zeigt das LED-Display nur einen Pixelpunkt an. Ist die Spannung 3.3 V, dann leuchtet das ganze Display. Wie werden Sie nun das Trimpotentiometer verwenden, um die Helligkeit der LEDs einzustellen.

## 5.4 Photozelle

- 1 1x micro:bit
- 2 1x micro:bit-Anschluss für Steckplatine
- 3 1x Steckplatine
- 4 1x Photozelle
- 5 1x 10 kΩ Widerstand (braun/schwarz/schwarz/rot/braun)

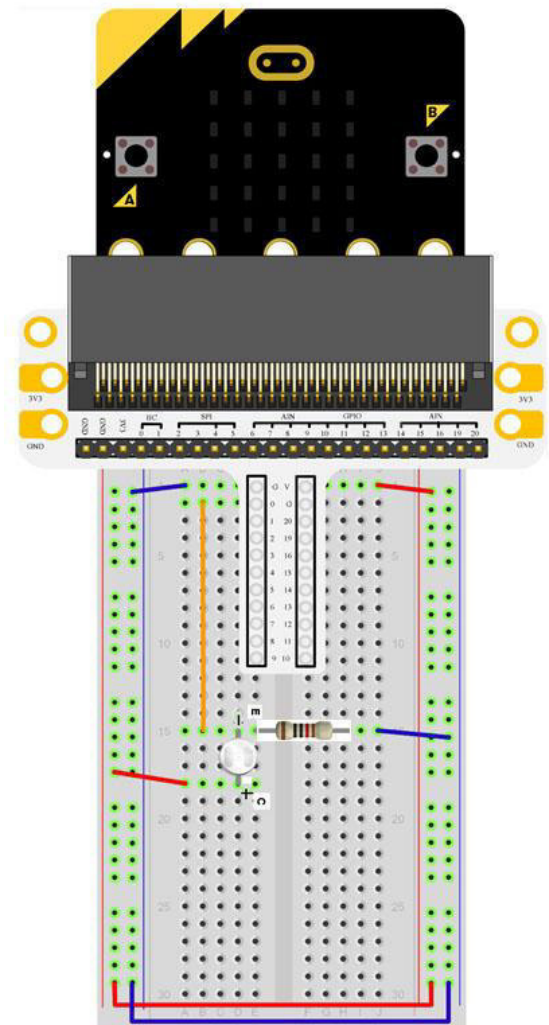


4

Kurzes Bein =  
Kollektor  
Positive Anode  
Langes Bein = Emitter  
Negative Kathode



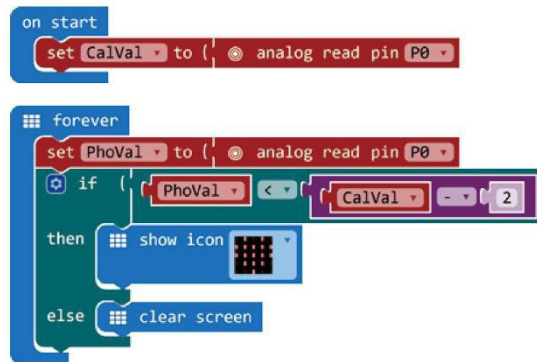
5



Wir werden eine Photozelle verwenden, um die Helligkeit des Displays vom micro:bit anzusteuern.

Befestigen Sie die benötigten Komponenten an der Steckplatine (siehe Abb.).

Einen Code erstellen



1. Wir werden zuerst zwei Variablen kreieren. Wählen Sie **Variables/Variablen** im Blockmenü aus und klicken Sie auf **Make a Variable**.

New variable name:

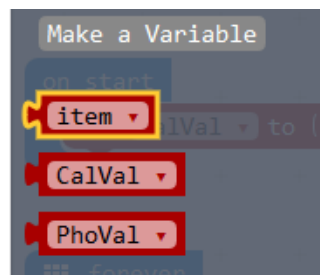
CalVal

Ok ✓

Cancel ✕

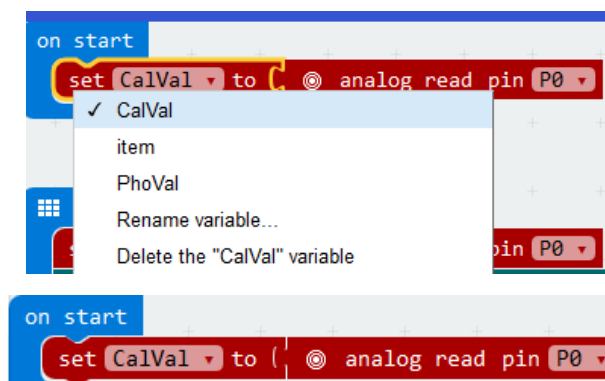
Geben Sie **CalVal** im Fenster ein und klicken Sie auf **Ok**. Geben Sie **PhoVal** im Fenster ein und klicken Sie auf **Ok**.

Es werden zwei neue Variablen im Blockmenü unter **Variables/Variablen** angezeigt.



Sie werden diese Variablen später brauchen, um die Daten in einem Register zu speichern.

Den Block mit dem Symbol für **kleiner als** und den Block **if then else/wenn dann ansonsten** finden Sie im Blockmenü unter **Logic/Logik**. Den Block mit dem **Minuszeichen** finden Sie im Blockmenü unter **Math/Mathematik**. Den Block **show icon/zeige Symbol** finden Sie im Blockmenü unter **Basic/Grundlagen**. Den Block **clear screen/Bildschirminhalt löschen** finden Sie im Blockmenü unter **Basic/Grundlagen** → **More**. Den Block **set item to 0** finden Sie im Blockmenü unter **Variables/Variablen**. Klicken Sie den Pfeil an und wählen Sie **CalVal** oder **PhoVal** aus.



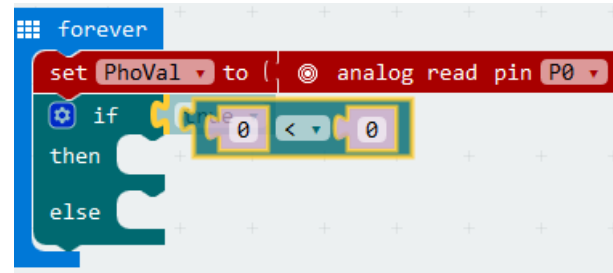
2. Wählen Sie die Variable **CalVal** und stellen Sie **analog read pin/analogue Werte von Pin** auf P0.



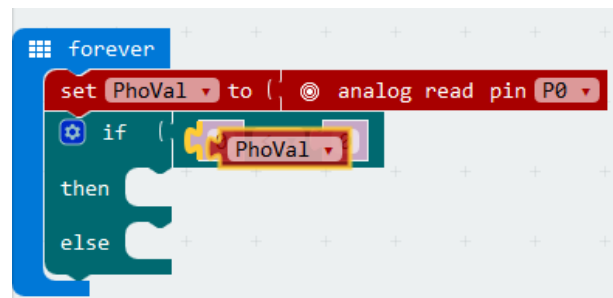
3. Im Block **forever/dauerhaft**, wählen Sie die Variable **PhoVal** und stellen Sie **analog read pin/analogue Werte von Pin** auf P0.



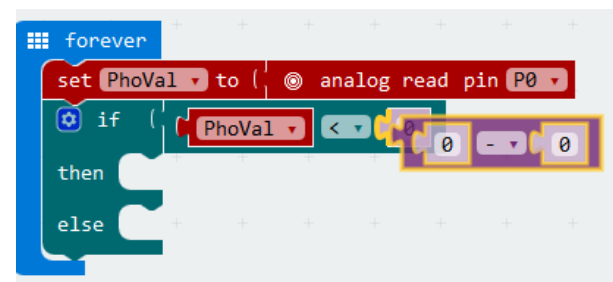
4. Verschieben Sie den Block mit dem Symbol für **kleiner als** per Drag and Drop neben dem Block **if/wenn** und stellen Sie es über den Block **true/wahr**.



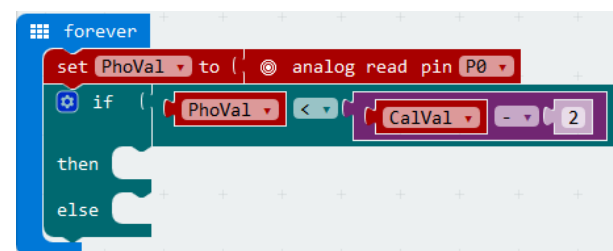
Verschieben Sie nun die Variable **PhoVal** (unter **Variables/Variablen** im Blockmenü) über die erste 0 vom Block mit dem Symbol für **kleiner als**.



Verschieben Sie den Block mit dem **Minuszeichen** per Drag and Drop über die zweite 0 vom Block mit dem Symbol für **kleiner als**.



Verschieben Sie danach die Variable **CalVal** (unter **Variables/Variablen** im Blockmenü) per Drag and Drop über die erste 0 vom Block mit dem **Minuszeichen**. Stellen Sie die zweite 0 vom Block mit dem **Minuszeichen** auf 2.



5.	Verschieben Sie den Block <b>show icon/zeige Symbol</b> per Drag and Drop neben Block <b>then/dann</b> . Verschieben Sie den Block <b>clear screen/Bildschirminhalt löschen</b> per Drag and Drop neben Block <b>else/ansonsten</b> .
6.	Ist die Variable <b>PhoVal</b> kleiner als die Variable <b>CalVal - 2</b> , dann wird ein Herz im Display angezeigt. Ist dies nicht der Fall, dann wird das Display ausgeschaltet.
7.	Kompilieren Sie das Programm und speichern Sie es als .Hex-Datei. Klicken Sie danach auf Download/Herunterladen und speichern Sie die .hex-Datei im Ordner →Downloads (C:\downloads). Diese .hex-Datei kann nun auf das micro: bit übertragen werden.  Verbinden Sie den micro:bit mit einem USB-Port. Verschieben Sie die .hex-Datei nun per Drag and Drop in micro:bit [removable device], um das Programm hochzuladen.

Bemerkung: Setzen Sie den micro:bit auf Werkseinstellungen zurück, um den Referenzwert gemäß der aktuellen Helligkeit zu justieren. Um das Programm korrekt laufen zu lassen, müssen wir mit eingeschaltetem Licht starten.

Ist das Licht eingeschaltet, dann wird sehen Sie nichts im Display. Ist das Licht ausgeschaltet, dann erscheint das Herz im Display. Wie können wir nun eine Photozelle verwenden, um eine LED anzusteuern?

## 5.5 RGB-LED

- 1 1x micro:bit
- 2 1x micro:bit-Anschluss für Steckplatine
- 3 1x Steckplatine
- 4 1x RGB-LED (gemeinsame Kathode)
- 5 3x Widerstand 10 Ω (braun/schwarz/braun/vergoldet)

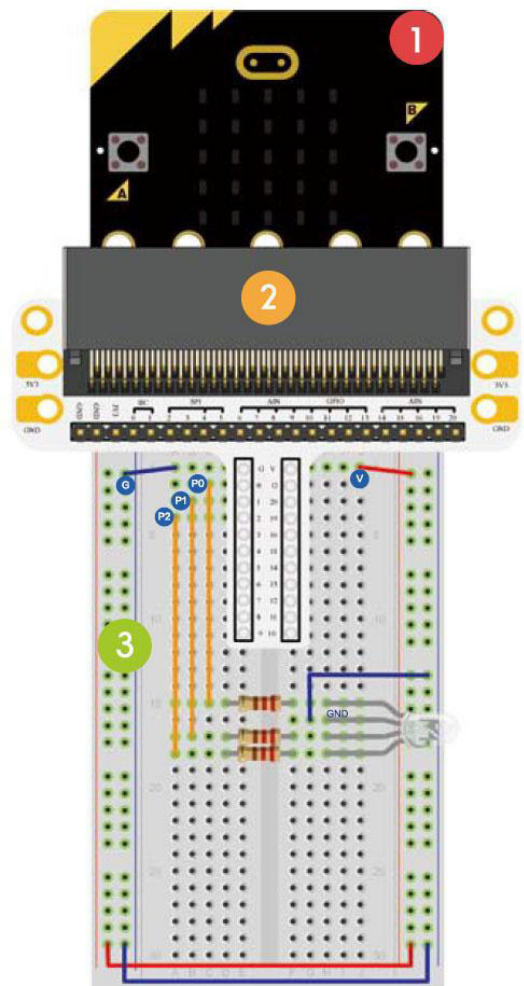


4

1. Rot (Anode +)
2. Masse (Kathode -) – langes Bein
3. Grün (Anode +)
4. Blau (Anode +)



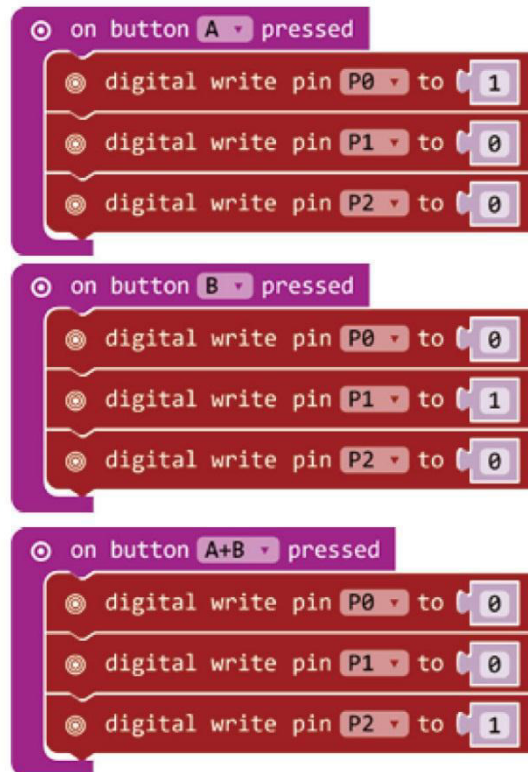
5



Wir werden dafür sorgen, dass die RGB-LED stufenweise zwischen die Farben rot, grün und blau wechselt.

Befestigen Sie die benötigten Komponenten an der Steckplatine (siehe Abb.).

## Einen Code erstellen



- Verschieben Sie die Blöcke per Drag and Drop (siehe Abb.).

Den Block **on button A pressed/wenn Knopf A gedrückt** im Blockmenü unter **Input/Eingabe**. Den Block **digital write pin P0 to 0/schreibe digitalen Wert von Pin P0 auf 0** finden Sie im Blockmenü unter **Advanced/Fortgeschritten Pins**.

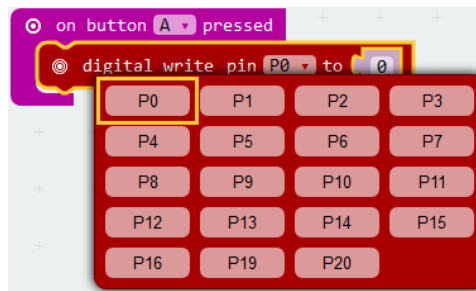
Wählen Sie die Option A für den Block **on button A pressed/wenn Knopf A gedrückt**.

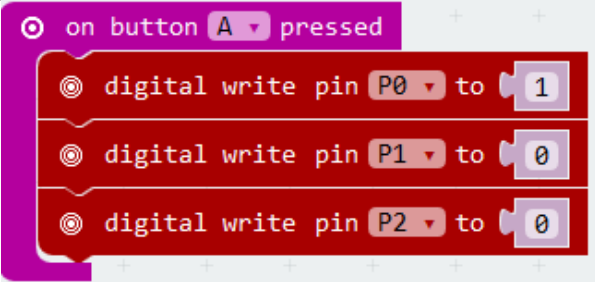
Verschieben Sie drei Blöcke **digital write pin P0 to 0/schreibe digitalen Wert von Pin P0 auf 0** per Drag und Drop in den Block **on button A pressed/wenn Knopf A gedrückt**.

Wählen Sie Pin P0 im ersten Block **digital write pin P0 to 0/schreibe digitalen Wert von Pin P0 auf 0** und stellen Sie ihn auf 1 (rote LED ein).

Wählen Sie Pin P1 im zweiten Block **digital write pin P0 to 0/schreibe digitalen Wert von Pin P0 auf 0** und stellen Sie ihn auf 0 (grüne LED aus).

Wählen Sie Pin P2 im dritten Block **digital write pin P0 to 0/schreibe digitalen Wert von Pin P0 auf 0** und stellen Sie ihn auf 0 (blaue LED aus).

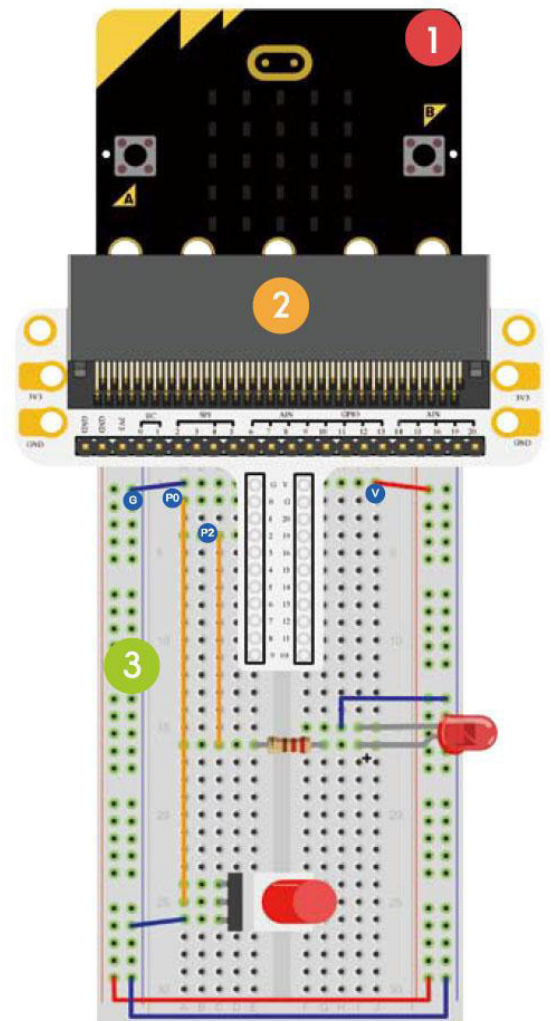
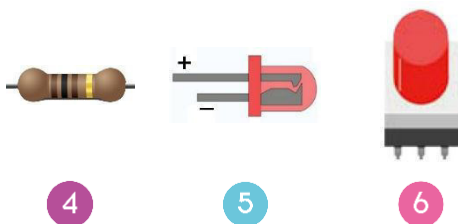


	
2.	Ähnlicherweise, stellen Sie die zwei Blöcke <b>on button B/A+B pressed/wenn Knopf B/A+B gedrückt</b> zusammen, um den ganzen Code zu erstellen.
3.	Kompilieren Sie das Programm und speichern Sie es als .Hex-Datei. Klicken Sie danach auf Download/Herunterladen und speichern Sie die .hex-Datei im Ordner →Downloads (C:\downloads). Diese .hex-Datei kann nun auf das micro:bit übertragen werden. Verbinden Sie den micro:bit mit einem USB-Port. Verschieben Sie die .hex-Datei nun per Drag and Drop in micro:bit [removable device], um das Programm hochzuladen.

Drücken Sie Taste A, um die rote LED einzuschalten. Drücken Sie Taste B, um die grüne LED einzuschalten. Drücken Sie Taste A + B, um die blaue LED einzuschalten. Wie werden Sie nun einen sanften Farbverlauf mit dem RGB-LED kreieren.

## 5.6 Selbstschließender Schalter

- 1 1x micro:bit
- 2 1x micro:bit-Anschluss für Steckplatine
- 3 1x Steckplatine
- 4 1x Widerstand 100  $\Omega$  (braun/schwarz/braun/vergoldet)
- 5 1x rote LED (Polarisation: Anode (+) = langes Bein, Kathode (-) = kurzes Bein)
- 6 1x selbstschließender Schalter oder bistabiler Schalter

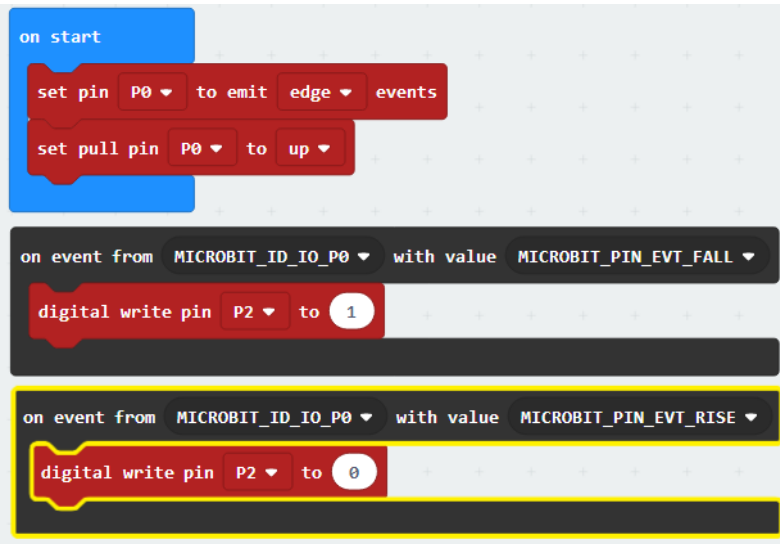


Wir gehen die Umgebungstemperatur (Daten) vom analogen Temperatursensor ablesen und die Daten auf dem micro:bit anzeigen.

Befestigen Sie die benötigten Komponenten an der Steckplatine (siehe Abb.).

Einen Code erstellen





1.	<p>Verschieben Sie die Blöcke per Drag and Drop (siehe Abb.).</p> <p>Den Block <b>set pin P0 to emit edge events/Stelle Pin P0 ein zum Ausgeben von Rand-Ereignissen</b> und den Block <b>set pull P0 to up/setze Anziehungskraft von Pin P0 auf nach oben</b> finden Sie im Blockmenü unter <b>Pins</b> → <b>More</b>. Die Blöcke <b>on event from MICROBIT...EVT_FALL/RISE</b> finden Sie im Blockmenü unter <b>Control/Steuerung</b>.</p>
2.	<p>Wählen Sie <b>edge/Rand</b> im Block "set pin P0 to emit edge events//Stelle Pin P0 ein zum Ausgeben von Rand-Ereignissen". Wählen Sie <b>P0</b> und a <b>up/nach oben</b> im Block "set pull P0 to up/setze Anziehungskraft von Pin P0 auf nach oben".</p>
3.	<p>Wir werden nun einen Block mit einem Ereignis (abfallende Flanke) kreieren. Verschieben Sie den Block <b>on event from MICROBIT...EVT°FALL</b> per Drag and Drop unter Block <b>on start/beim Start</b>. Klicken Sie den ersten Pfeil an und wählen Sie die Option <b>MICROBIT_ID_IO_P0</b> im Pull-Down-Menü. Klicken Sie den zweiten Pfeil an und wählen Sie die Option <b>MICROBIT_PIN_EVENT_FALL</b> im Pull-Down-Menü.</p> <p>Verschieben Sie den Block <b>digital write pin P0 to 0/schreibe digitalen Wert von Pin P0 auf 0</b> per Drag and Drop in den Block "on event from MICROBIT...EVT°FALL". Stellen Sie Pin P0 auf P2 und stellen Sie 0 auf 1. Den Block <b>digital write pin P0 to 0/schreibe digitalen Wert von Pin P0 auf 0</b> finden Sie im Blockmenü unter Advanced/Fortgeschritten Pins.</p>
4.	<p>Wiederholen Sie diese Schritte für den zweiten Block. Klicken Sie den zweiten Pfeil an und wählen Sie hier aber die Option <b>MICROBIT_PIN_EVENT_RISE</b> im Pull-Down-Menü. Stellen Sie Pin P0 im Block <b>digital write pin P0 to 0/schreibe digitalen Wert von Pin P0 auf 0</b> auf P2 und stellen Sie den Wert auf 0.</p>
5.	<p>Kompilieren Sie das Programm und speichern Sie es als .Hex-Datei. Klicken Sie danach auf Download/Herunterladen und speichern Sie die .hex-Datei im Ordner →Downloads (C:\downloads). Diese .hex-Datei kann nun auf das micro: bit übertragen werden.</p> <p>Verbinden Sie den micro:bit mit einem USB-Port. Verschieben Sie die .hex-Datei nun per Drag and Drop in micro:bit [removable device], um das Programm hochzuladen.</p>

Drücken Sie den selbstschließenden Schalter, um die LED einzuschalten. Drücken Sie wieder, um die LED auszuschalten. Wie werden Sie nun das micro:bit-Display mit dem selbstschließenden Schalter ansteuern?



New variable name:

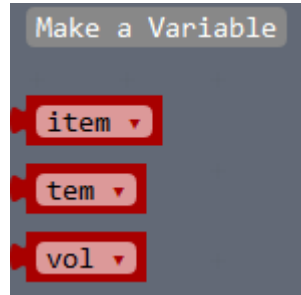
CalVa|

Ok ✓

Cancel ✕

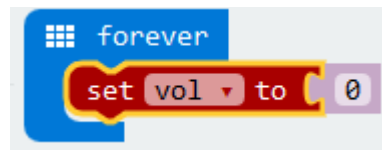
Geben Sie **vol** im Fenster ein und klicken Sie auf **Ok**. Geben Sie **tem** im Fenster ein und klicken Sie auf **Ok**.

Es werden zwei neue Variablen im Blockmenü unter **Variables/Variablen** angezeigt.

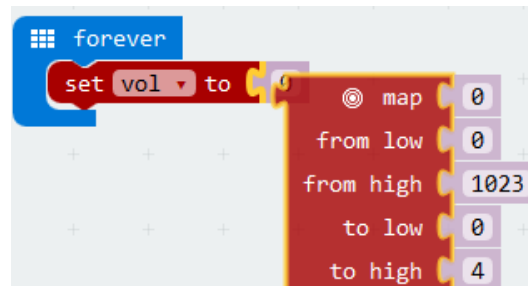


Sie werden diese Variablen später brauchen, um die Daten in einem Register zu speichern.

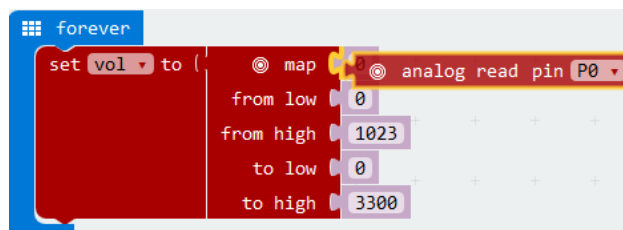
Verschieben Sie den Block **set item to/ändere item auf** per Drag and Drop in den Block **forever/dauerhaft** und wählen Sie die Option **vol** mit dem Pfeil.



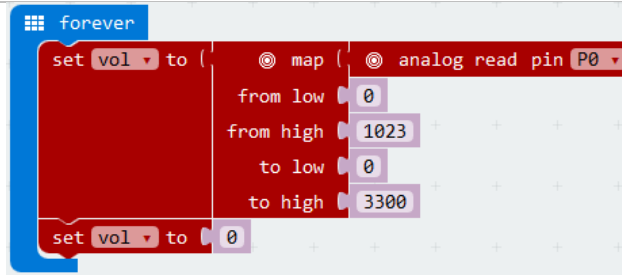
Verschieben Sie nun den Block **map/from low/from high/to low/to high/verteile/von niedrig/von hoch/bis niedrig/bis hoch** per Drag and Drop über die 0 neben Block **set item to/ändere item auf**.



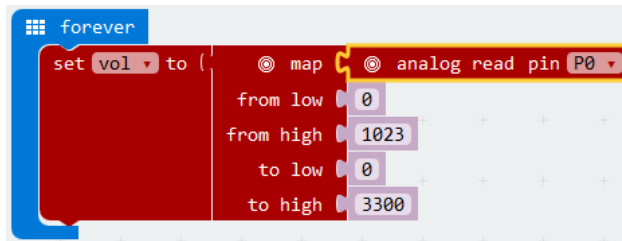
Verschieben Sie nun den Block **analog read pin/analoge Werte von Pin** per Drag and Drop neben Block **map/from low/from high/to low/to high/verteile/von niedrig/von hoch/bis niedrig/bis hoch** und stellen Sie den Wert **to high/bis hoch** 4 auf 3300.



Verschieben Sie nun den Block **set item to/ändere item auf** per Drag and Drop unter den ersten Block. Stellen Sie die Variable **item** mit dem Pfeil auf **vol**



Die gemessene Spannung in mV via **analog read pin/analoge Werte von Pin P0** ist einen 8-bit-Wert von 0-1023 (0-3.3 V) und ist abgebildet von einen niedrigen Wert (0 oder 0 V) bis einen hohen Wert (3300 mV oder 3.3 V). Die gemessene Spannung (mV) wird in Variable **vol** gespeichert.

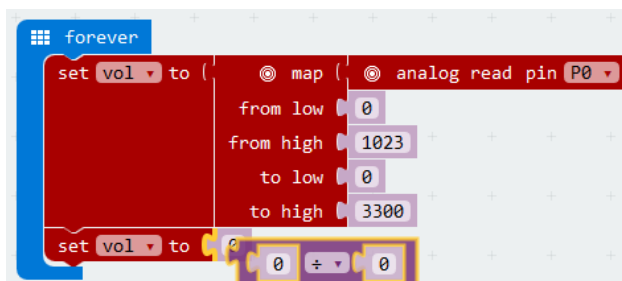


Wir werden nun die Formel schreiben, um die gemessene Spannung in eine Temperatur umzusetzen:

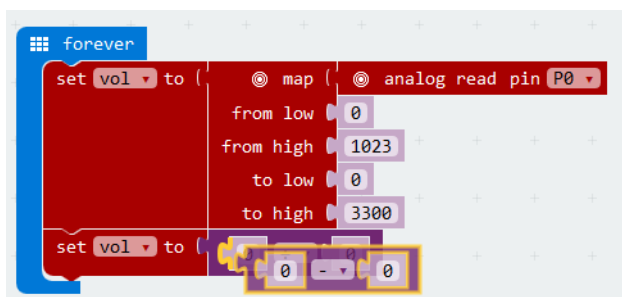
$$\text{Temperature } (^{\circ}\text{C}) = \frac{(\text{Output voltage (mV)} - 500)}{10}$$

In dieser Formel ist die Ausgangsspannung die Variable **vol** (wird mit TMP36-Sensor gemessen). Das Formel-Ergebnis wird in Variable **tem** gespeichert.

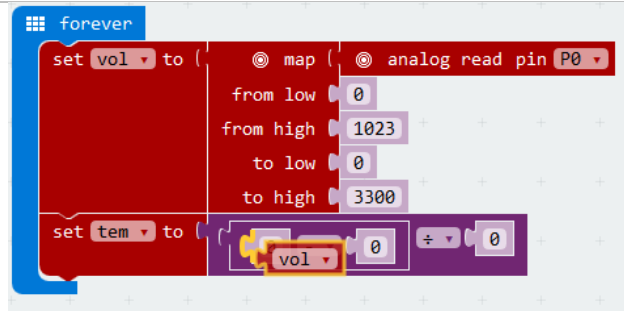
Wir werden nun den Code mit Blöcken schreiben. Verschieben Sie den Block mit dem **Teilungszeichen** per Drag and Drop über die 0 vom Block **set vol to/ändere vol auf**.



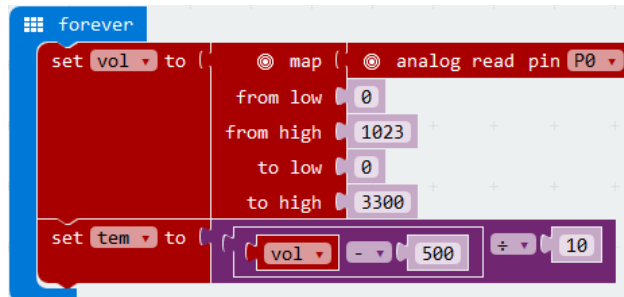
Verschieben Sie den Block mit dem **Minuszeichen** per Drag and Drop über die erste 0 vom Block mit dem **Teilungszeichen**.



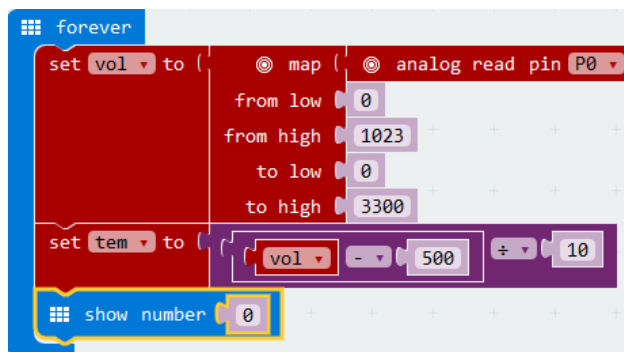
Ändern Sie nun die zweite Variable von **vol** auf **tem**, und verschieben Sie die Variable **vol** per Drag and Drop über die erste 0 vom Block mit dem **Minuszeichen**. Die Variable **vol** finden Sie im Blockmenü unter **Variables/Variablen**.



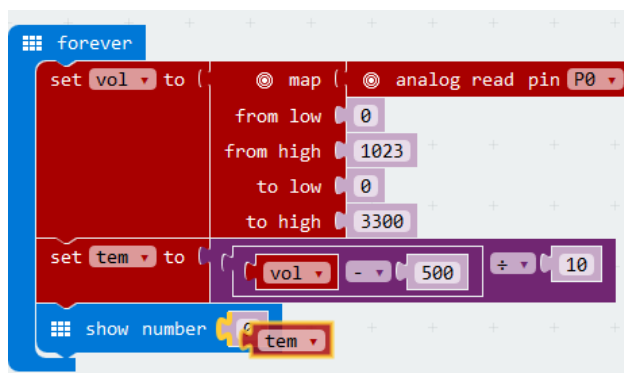
Stellen Sie die 0 vom Block mit dem **Minuszeichen** auf 500. Stellen Sie den Wert vom Block mit dem **Teilungszeichen** auf 10.



Verschieben Sie den Block **show number/Zeige Nummer** per Drag und Drop unter den Block **set tem to/ändere tem auf**.



Verschieben Sie schließlich die Variable **tem** per Drag and Drop über die 0 vom Block **show number/Zeige Nummer**.



Sie haben den Code nun geschrieben!

3. Kompilieren Sie das Programm und speichern Sie es als .Hex-Datei. Klicken Sie danach auf Download/Herunterladen und speichern Sie die .hex-Datei im Ordner →Downloads (C:\downloads). Diese .hex-Datei kann nun auf das micro: bit übertragen werden.

Verbinden Sie den micro:bit mit einem USB-Port. Verschieben Sie die .hex-Datei nun per Drag and Drop in micro:bit [removable device], um das Programm hochzuladen.

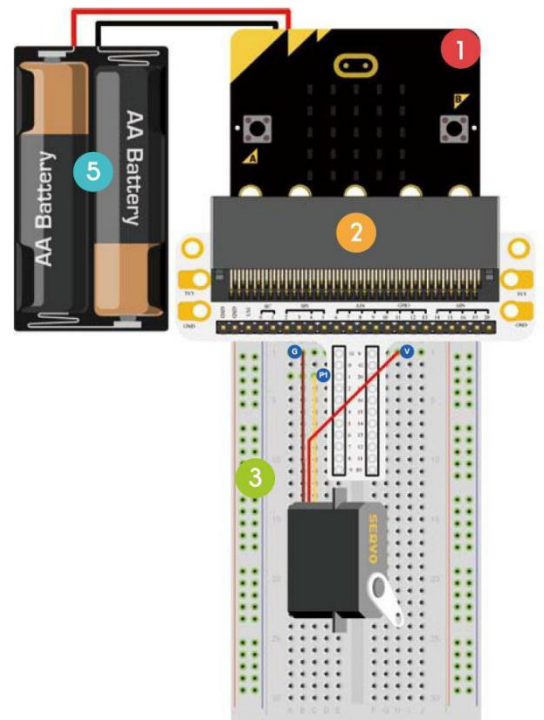
Die zwei LEDs werden abwechselnd blinken. Möchten Sie nun die Temperatur in Celsius oder Fahrenheit anzeigen?

## 5.8 Servo

- 1 1x micro:bit
- 2 1x micro:bit-Anschluss für Steckplatine
- 3 1x Steckplatine
- 4 1x Mini-Servo
- 5 1x Batteriehalter mit 2x AA-Batterie (1.5 V)



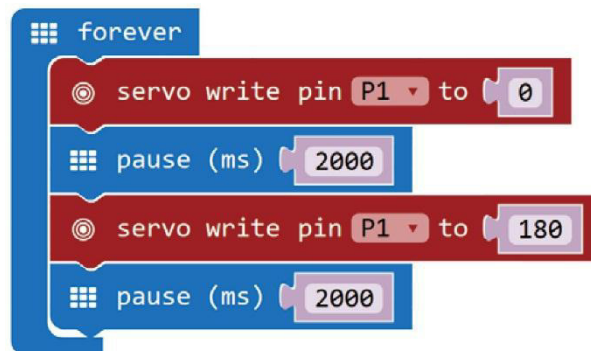
4



Wir werden einen Servomotor kreieren, der ständig innerhalb eines Bereichs von 0-180° dreht.

Befestigen Sie die benötigten Komponenten an der Steckplatine (siehe Abb.).

Einen Code erstellen



1.	Verschieben Sie die Blöcke per Drag and Drop (siehe Abb.). Den Block <b>digital write pin P0 to 180</b> finden Sie im Blockmenü unter <b>Advanced/Fortgeschritten</b> Pins. Den Block <b>forever/dauerhaft</b> und den Block <b>pause/pausiere (ms) 100</b> finden Sie im Blockmenü unter <b>Basic/Grundlagen</b> .
2.	Stellen Sie <b>P0</b> auf <b>P1</b> und stellen Sie den Wert auf 0.
3.	Stellen Sie den Block <b>pause/pausiere</b> auf 2000 ms.
4.	Stellen Sie <b>P0</b> auf <b>P1</b> . Stellen Sie die 0 auf 180.
5.	Stellen Sie den Block <b>pause/pausiere</b> auf 2000 ms.
6.	Kompilieren Sie das Programm und speichern Sie es als .hex-Datei. Klicken Sie danach auf Download/Herunterladen und speichern Sie die .hex-Datei im Ordner →Downloads (C:\downloads). Diese .hex-Datei kann nun auf das micro: bit übertragen werden.

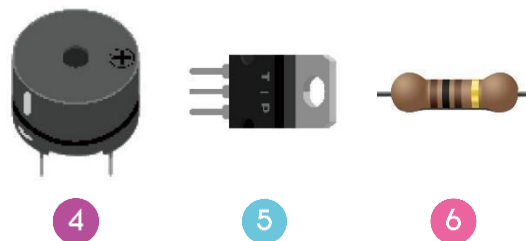


Verbinden Sie den micro:bit mit einem USB-Port. Verschieben Sie die .hex-Datei nun per Drag and Drop in micro:bit [removable device], um das Programm hochzuladen.

Der Servomotor dreht nun von 0 bis 180°. Wie möchten Sie nun einen Zeigerthermometer mit einem Temperatursensor und einen Servomotor kreieren.

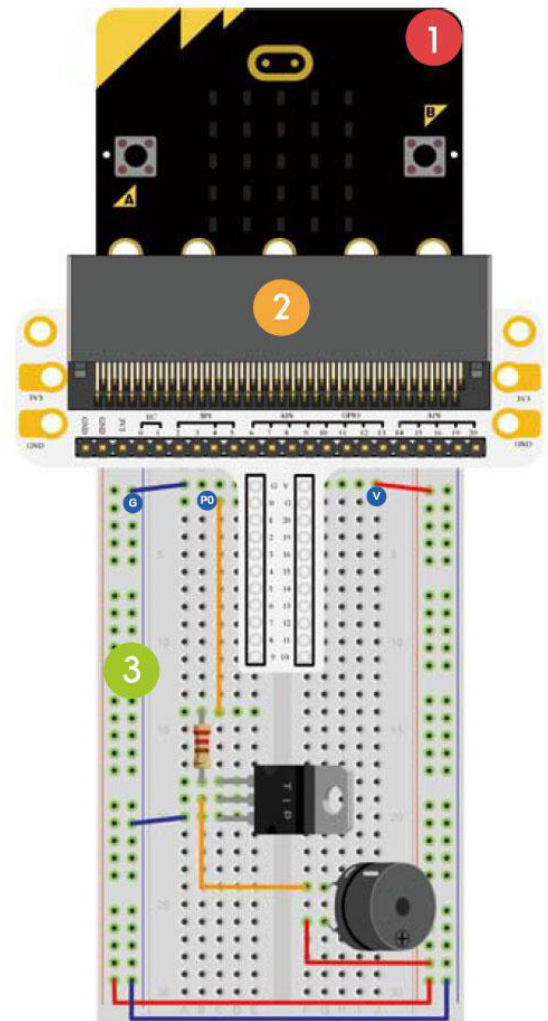
## 5.9 Summer

- 1 1x micro:bit
- 2 1x micro:bit-Anschluss für Steckplatine
- 3 1x Steckplatine
- 4 1x Mini-Lautsprecher
- 5 1x N-Kanal MOSFET
- 6 1x Widerstand 100 Ω (braun/schwarz/braun/vergoldet)

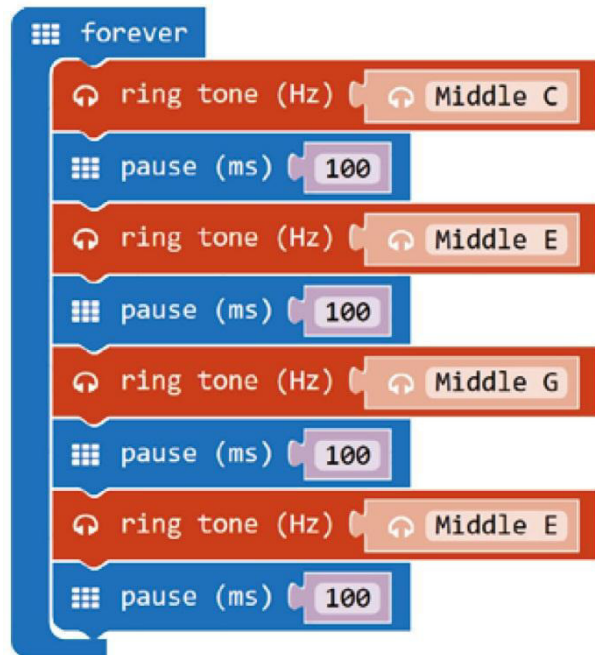


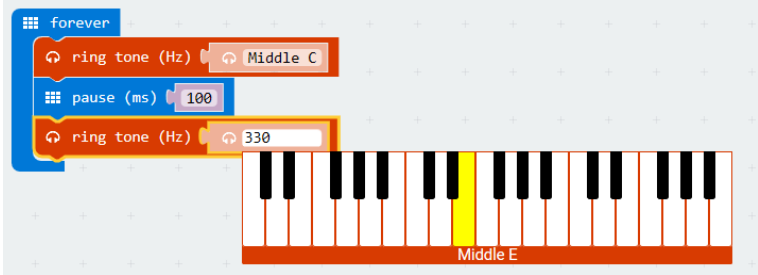
Wir werden einen Summer ansteuern.

Befestigen Sie die benötigten Komponenten an der Steckplatine (siehe Abb.).



Einen Code erstellen

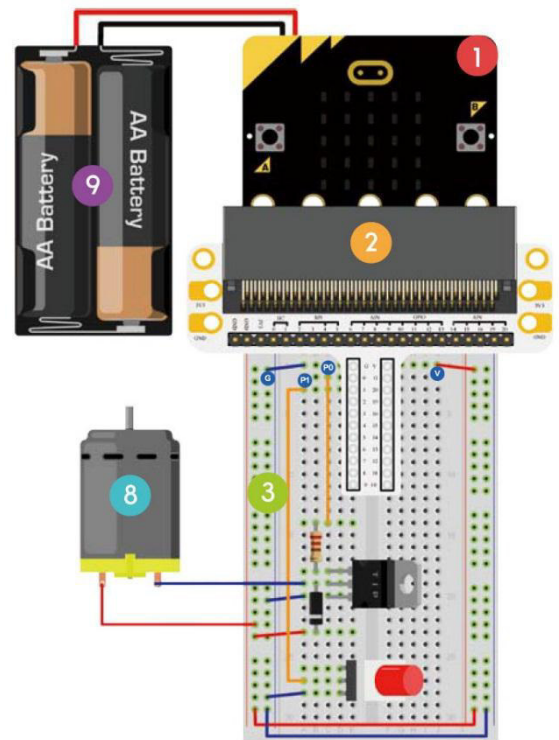
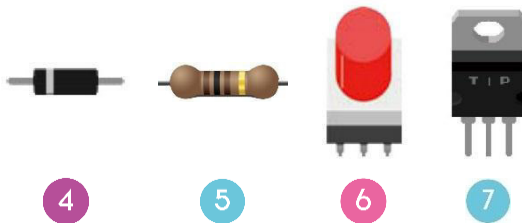


1.	<p>Verschieben Sie die Blöcke per Drag and Drop (siehe Abb.).</p> <p>Den Block <b>forever/dauerhaft</b> und den Block <b>pause/pausiere (ms) 100</b> finden Sie im Blockmenü unter <b>Basic/Grundlagen</b>. Den Block <b>ring tone (Hz)/Klingelton (Hz)</b> finden Sie im Blockmenü unter <b>Music/Musik</b>.</p>
2.	<p>Stellen Sie den ersten <b>ring tone (Hz)/Klingelton (Hz)</b> auf <b>middle C/Mittleres C</b>. Klicken Sie hierzu auf das Textfeld und wählen Sie die gewünschte Klaviertaste.</p>  <p>Sie werden den entsprechenden Ton hören.</p>
3.	<p>Stellen Sie den Block <b>pause/pausiere</b> auf 100 ms.</p>
4.	<p>Wiederholen Sie diese Schritte für die anderen Ringtöne und Pausen.</p>
5.	<p>Kompilieren Sie das Programm und speichern Sie es als .Hex-Datei. Klicken Sie danach auf Download/Herunterladen und speichern Sie die .hex-Datei im Ordner →Downloads (C:\downloads). Diese .hex-Datei kann nun auf das micro: bit übertragen werden.</p> <p>Verbinden Sie den micro:bit mit einem USB-Port. Verschieben Sie die .hex-Datei nun per Drag and Drop in micro:bit [removable device], um das Programm hochzuladen.</p>

Es ertönt ein akustisches Signal. Wie werden Sie nun Ihr Lieblings-Kinderlied programmieren.

## 5.10 Motor

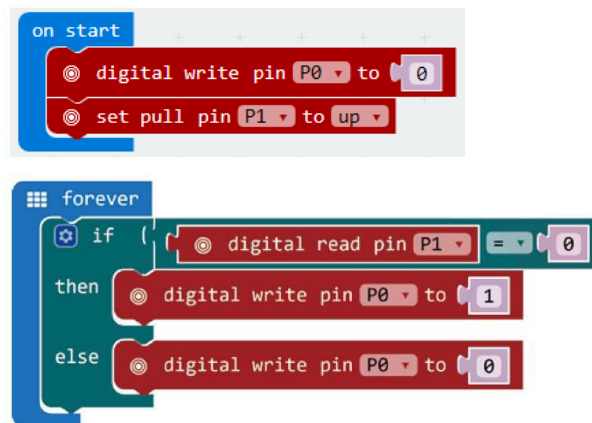
- 1 1x micro:bit
- 2 1x micro:bit-Anschluss für Steckplatine
- 3 1x Steckplatine
- 4 1x Diode
- 5 1x Widerstand 100  $\Omega$   
(braun/schwarz/braun/vergoldet)
- 6 1x selbstschließender Schalter oder bistabiler Schalter
- 7 1x N-Kanal MOSFET
- 8 1x Mini-Motor
- 9 1x Batteriehalter mit 2x AA-Batterie (1.5 V)



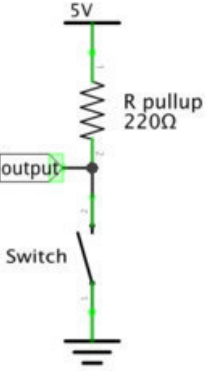
Wir werden einen Schalter verwenden, um den Start und den Stopp eines Motors anzusteuern.

Befestigen Sie die benötigten Komponenten an der Steckplatine (siehe Abb.).

Einen Code erstellen



1.	Verschieben Sie die Blöcke per Drag and Drop (siehe Abb.).
2.	Der Block <b>on start/beim Start</b> wird nur einmal ausgeführt, um das Programm zu starten.
3.	Stellen Sie <b>P0</b> auf <b>0</b> .
4.	Stellen Sie den Block <b>set pull P0 to up/setze Anziehungskraft von Pin P0 auf nach oben</b> auf <b>P1</b> und <b>nach oben</b> .

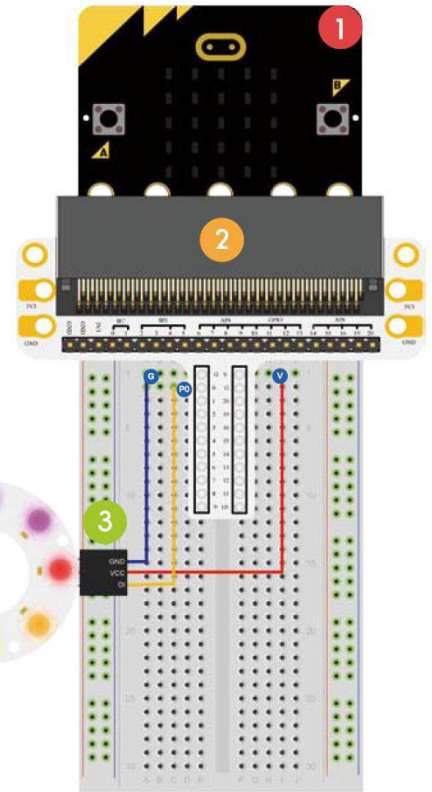
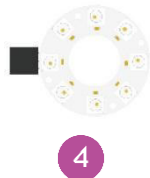
<p><b>5.</b></p>	<p>Verschieben Sie den Block <b>digital read pin/digitale Werte von Pin</b> auf <b>P1</b> per Drag and Drop in Block <b>forever/dauerhaft</b>. Stellen Sie den Block mit dem <b>Gleichheitszeichen</b> auf 0.</p> <p>In Wirklichkeit ist dies ein Schema. Sie brauchen keinen Pull-Up-Widerstand an der Steckplatine zu befestigen. Die Pull-Up-Funktion ist im Code programmiert (siehe Schritt 3-4) und ersetzt die Notwendigkeit, um einen Widerstand an der Steckplatine zu befestigen.</p> 
<p><b>6.</b></p>	<p>Nach dem Drücken des Schalters, stellen Sie die hohe Spannung auf P0. Stellen Sie Wert 0 auf 1 (5 V). Der Motor fängt an zu drehen.</p>
<p><b>7.</b></p>	<p>Nach dem Loslassen des Schalters, stellen Sie die niedrige Spannung auf P0. Stellen Sie den Wert auf 0 Der Motor hört auf zu drehen.</p>
<p><b>8.</b></p>	<p>Kompilieren Sie das Programm und speichern Sie es als .Hex-Datei. Klicken Sie danach auf Download/Herunterladen und speichern Sie die .hex-Datei im Ordner →Downloads (C:\downloads). Diese .hex-Datei kann nun auf das micro: bit übertragen werden.</p> <p>Verbinden Sie den micro:bit mit einem USB-Port. Verschieben Sie die .hex-Datei nun per Drag and Drop in micro:bit [removable device], um das Programm hochzuladen.</p>

Drücken Sie die Taste, um den Motor zu starten und lassen Sie los, um ihn zu stoppen. Wie werden Sie nun ein Trimpotentiometer verwenden, um die Motorgeschwindigkeit anzusteuern?

Bemerkung: Die Spannung vom micro:bit beträgt nur 3.3 V. Dies genügt wahrscheinlich nicht, um den Ventilator zu starten. Um den Ventilator zu starten können, drehen Sie ggf. die Flügel.

## 5.11 Regenbogen-LED

- 1 1x micro:bit
- 2 1x micro:bit-Anschluss für Steckplatine
- 3 1x Steckplatine
- 4 1x RGB-LED-Ring



Wir werden 8 RGB-LEDs in einem Ring ansteuern und eine stufenweise Regenbogen kreieren.

Befestigen Sie die benötigten Komponenten an der Steckplatine (siehe Abb.).

Einen Code erstellen

```

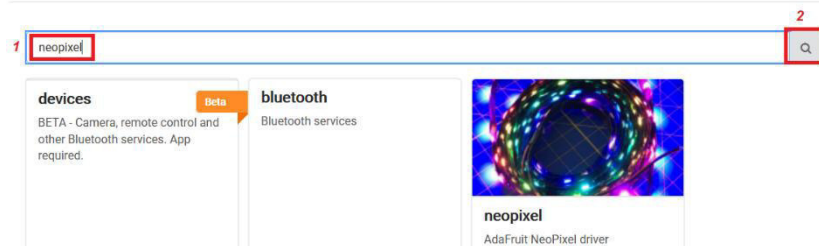
on start
  set strip to NeoPixel at pin P0 with 8 leds as RGB (GRB format)
  strip show rainbow from 1 to 360

forever
  strip show
  strip rotate pixels by 1
  pause (ms) 100
  
```

1.	Verschieben Sie die Blöcke per Drag and Drop (siehe Abb.).
2.	Suchen und fügen Sie die Bibliothek NeoPixel hinzu. Wählen Sie Advanced/Fortgeschritten → Extensions/Erweiterungen und geben Sie NeoPixel ein.

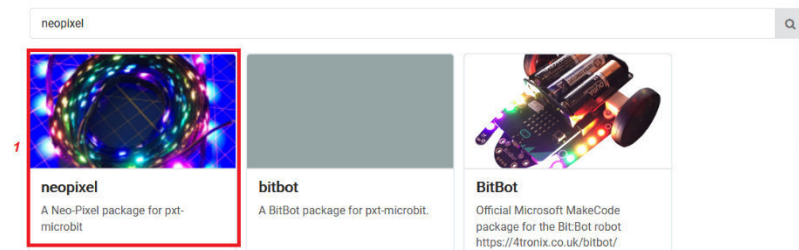


Add Package... ?

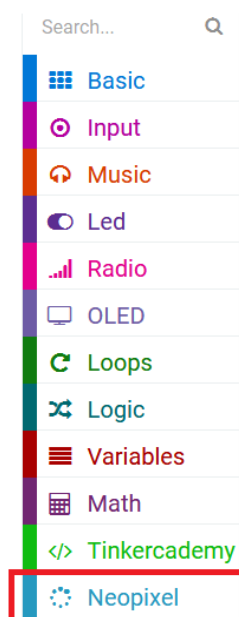


Wählen Sie danach die Bibliothek NeoPixel.

Add Package... ?


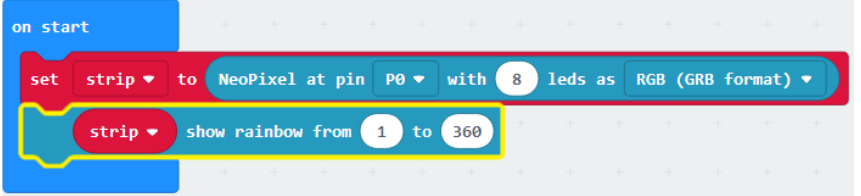


Die Bibliothek wird heruntergeladen und befindet sich nun im Blockmenü.



Die Blöcke **NeoPixel at pin P0 with 24 leds, item show rainbow from 1 to 360, item show** und **items rotate pixels by 1** finden Sie im Blockmenü unter **NeoPixel**.



3.	<p>Verschieben Sie den Block <b>set strip to NeoPixel at pin P0 with 24 leds as RGB (GRB format)</b> per Drag and Drop in Block <b>on start/beim Start</b>. Stellen Sie Pin auf <b>P0</b> und den Wert 24 auf 8 (der LED-Ring enthält 8 LEDs).</p> 
4.	<p>Verschieben Sie nun den Block <b>set strip to rainbow from 1 to 360</b> per Drag and Drop unter Block <b>set strip to NeoPixel at pin P0</b>.</p> 
5.	<p>Vervollständigen Sie den Code (siehe Beispiel).</p>
6.	<p>Kompilieren Sie das Programm und speichern Sie es als .Hex-Datei. Klicken Sie danach auf Download/Herunterladen und speichern Sie die .hex-Datei im Ordner →Downloads (C:\downloads). Diese .hex-Datei kann nun auf das micro: bit übertragen werden.</p> <p>Verbinden Sie den micro:bit mit einem USB-Port. Verschieben Sie die .hex-Datei nun per Drag and Drop in micro:bit [removable device], um das Programm hochzuladen.</p>

Sie werden nun einen drehenden Regenbogen am LED-Ring sehen. Wie werden Sie nun ein blinkendes Auge mit dem LED-Ring kreieren.

**Verwenden Sie dieses Gerät nur mit originellen Zubehörteilen. Velleman NV übernimmt keine Haftung für Schaden oder Verletzungen bei (falscher) Anwendung dieses Gerätes. Mehr Informationen zu diesem Produkt und die neueste Version dieser Bedienungsanleitung finden Sie hier: [www.velleman.eu](http://www.velleman.eu). Alle Änderungen ohne vorherige Ankündigung vorbehalten.**

**© URHEBERRECHT**

**Velleman NV besitzt das Urheberrecht für diese Bedienungsanleitung. Alle weltweiten Rechte vorbehalten.** Ohne vorherige schriftliche Genehmigung des Urhebers ist es nicht gestattet, diese Bedienungsanleitung ganz oder in Teilen zu reproduzieren, zu kopieren, zu übersetzen, zu bearbeiten oder zu speichern.

# INSTRUKCJA OBSŁUGI

## 1. Wstęp

**Przeznaczona dla mieszkańców Unii Europejskiej.**

**Ważne informacje dotyczące środowiska.**



Niniejszy symbol umieszczony na urządzeniu bądź opakowaniu wskazuje, że utylizacja produktu może być szkodliwa dla środowiska. Nie należy wyrzucać urządzenia (lub baterii) do zbiorczego pojemnika na odpady komunalne, należy je przekazać specjalistycznej firmie zajmującej się recyklingiem. Niniejsze urządzenie należy zwrócić dystrybutorowi lub lokalnej firmie świadczącej usługi recyklingu. Przestrzegać lokalnych zasad dotyczących środowiska.

**W razie wątpliwości należy skontaktować się z lokalnym organem odpowiedzialnym za utylizację odpadów.**

Dziękujemy za zakup produktu Velleman®! Prosimy o dokładne zapoznanie się z instrukcją obsługi przed użyciem urządzenia. Nie montować ani nie używać urządzenia, jeśli zostało uszkodzone podczas transportu - należy skontaktować się ze sprzedawcą.

## 2. Wskazówki bezpieczeństwa



Z niniejszego urządzenia mogą korzystać dzieci powyżej 8 roku życia i osoby o ograniczonych zdolnościach fizycznych, zmysłowych bądź umysłowych, jak również osoby nieposiadające doświadczenia lub znajomości urządzenia, jeśli znajdują się one pod nadzorem innych osób lub jeśli zostały pouczone na temat bezpiecznego sposobu użycia urządzenia oraz zdają sobie sprawę ze związanych z nim zagrożeń. Dzieci nie mogą używać urządzenia do zabawy. Prace związane z czyszczeniem i konserwacją nie mogą być wykonywane przez dzieci pozostawione bez nadzoru.



Wyłącznie do użytku wewnątrz pomieszczeń.

Chronić uchwyt przed deszczem, wilgocią, rozpryskami i ściekającymi cieczami.

## 3. Informacje ogólne



- Proszę zapoznać się z informacjami w części Usługi i gwarancja jakości Velleman® na końcu niniejszej instrukcji.
- Przed rozpoczęciem pracy z urządzeniem należy zapoznać się z jego funkcjami.
- Wprowadzanie zmian w urządzeniu jest zabronione ze względów bezpieczeństwa. Uszkodzenia spowodowane zmianami wprowadzonymi przez użytkownika nie podlegają gwarancji.
- Stosować urządzenie wyłącznie zgodnie z przeznaczeniem. Używanie urządzenia w niedozwolony sposób spowoduje unieważnienie gwarancji.
- Gwarancja nie obejmuje uszkodzeń spowodowanych nieprzestrzeganiem niniejszej instrukcji, a sprzedawca nie ponosi odpowiedzialności za wynikłe uszkodzenia lub problemy.
- Firma Velleman ani jej dystrybutorzy nie ponoszą odpowiedzialności za jakiegokolwiek szkody (nadzwyczajne, przypadkowe lub pośrednie) dowolnej natury (finansowe, fizyczne...), wynikające z posiadania, użytkowania lub awarii niniejszego produktu.
- Ze względu na stałe udoskonalanie produktu, rzeczywisty wygląd produktu może różnić się od przedstawionego na zdjęciach.
- Rysunki służą wyłącznie do celów poglądowych.
- Nie włączać urządzenia tuż po tym, jak zostało narażone na zmiany temperatury. Chronić urządzenie przed uszkodzeniem, pozostawiając je wyłączone do momentu osiągnięcia temperatury pokojowej.
- Zachować niniejszą instrukcję na przyszłość.

## 4. Opis

Ten zestaw startowy jest zestawem edukacyjnym na bazie micro:bit. Obejmuje podstawowe elementy elektroniczne, płytkę prototypową, kable łączące i micro:bit.

## 5. Przykłady

### 5.1 LED

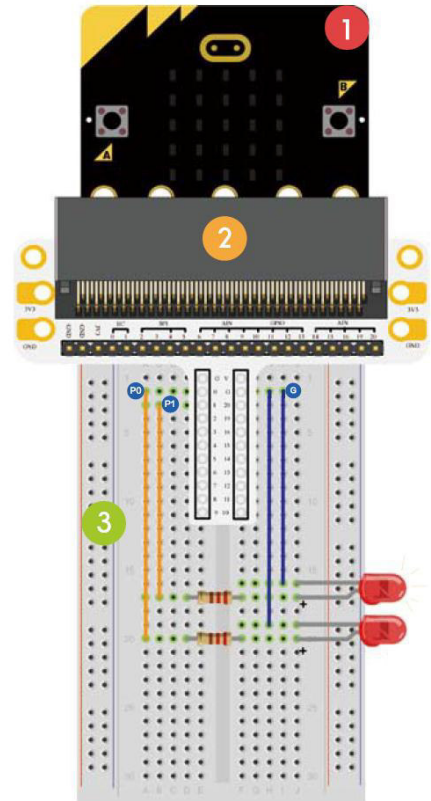
- 1 1x płytkę micro:bit
- 2 1x adapter płytki prototypowej micro:bit
- 3 1x płytkę prototypową
- 4 2x czerwona dioda LED (polaryzacja: anoda (+) = długa nóżka, katoda (-) = krótka nóżka)
- 5 2x rezystor 100  $\Omega$  (brązowy/czarny/brązowy/złoty)



4



5



Podczas tego kursu, wykorzystamy micro:bit, aby dwie diody LED migały naprzemiennie.

Umieścić niezbędne elementy na płytce prototypowej, jak pokazano.

Program jest napisany w blokach kodu, online w przeglądarce internetowej. Otworzyć stronę [www.makecode.com](http://www.makecode.com) lub <https://www.microsoft.com/en-us/makecode?rtc=1>, kliknąć ikonę micro:bit i kliknąć **Start Project**.

Microsoft MakeCode brings computer science to life for all students with fun projects, immediate results, and both block and text editors for learners at different levels.

**micro:bit**  
Start coding with micro:bit >

**Circuit Playground Express**  
Start coding with Circuit Playground Express >

**Minecraft**  
Start coding with Minecraft >

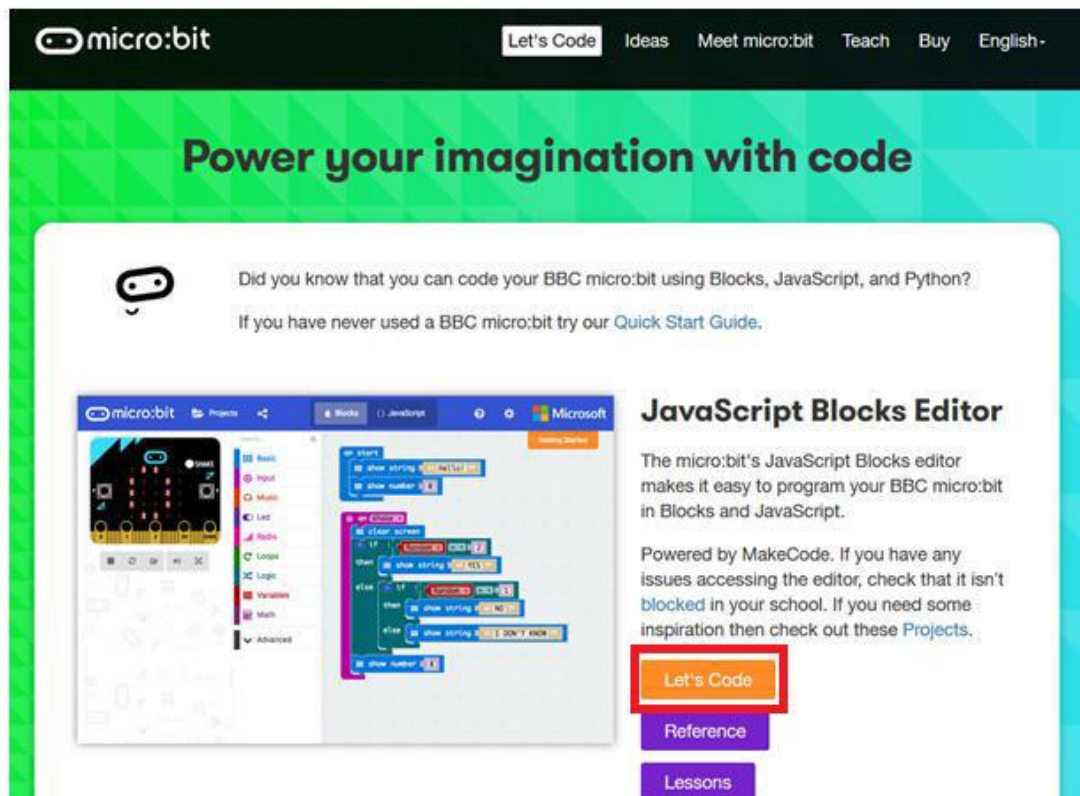
Blok kodu micro:bit otwiera się w nowym oknie. Teraz można rozpocząć składanie kodu za pomocą bloków kodowych, które przeciąga się z szuflady kodów i upuszcza w edytorze kodów.

Przeczytaj poniżej, jak to działa...

## Czym jest MakeCode?

Dawniej PXT (Programming eXperience Toolkit Editor) – edytor do pisania kodów dla micro:bit.

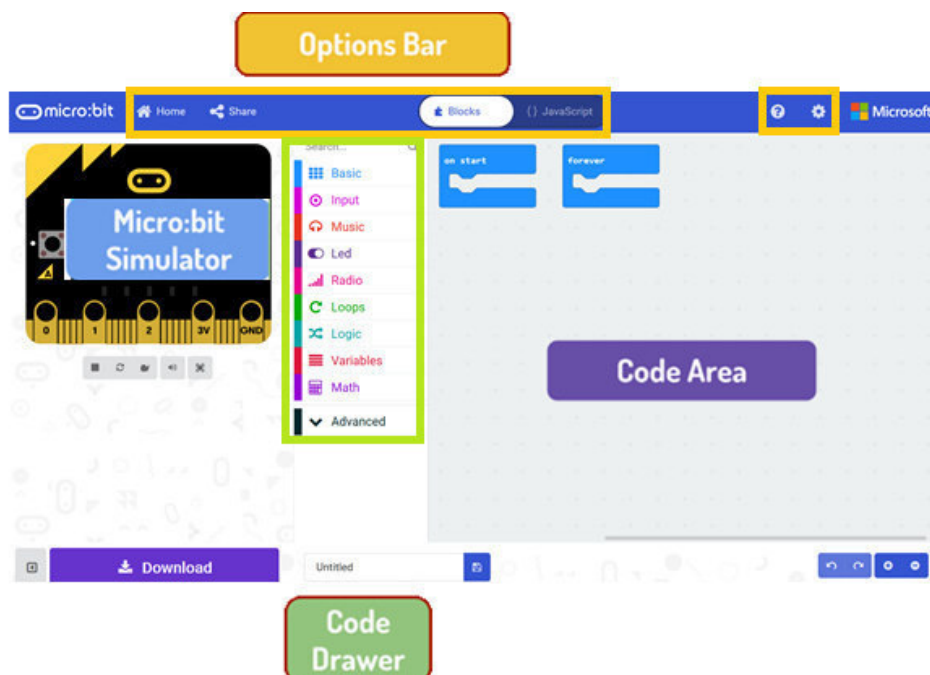
Graficzny i przyjazny dla początkujących edytor kodów typu "przeciągnij i upuść" podobny do Scratch, który pracuje online, w przeglądarce.



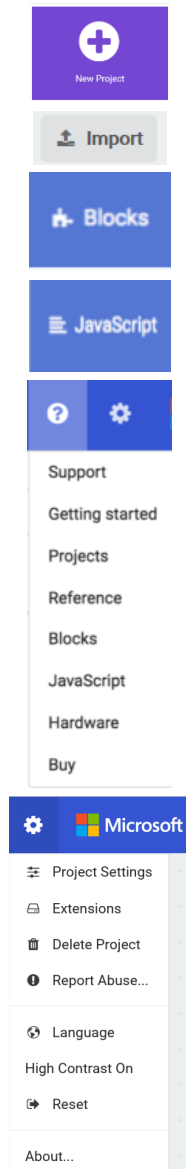
Można zauważyć, że edytor składa się z różnych sekcji. Kod tworzy się w obszarze **Code Area**, przeciągając bloki z szuflady kodów **Code Drawer**. Można od razu zobaczyć wynik działania kodu w symulatorze **micro:bit simulator**. Na dole znajduje się miejsce pobrania i zapisania projektu.



Następnie przyjrzymy się dokładniej, jak używać paska **Options Bar** do robienia innych rzeczy.



## Pasek opcji



Kliknąć, aby stworzyć lub dodać nowy projekt.

Tu można zaimportować własne projekty.

Otworzyć interfejs Blocks, aby pisać skrypty w edytorze bloków.

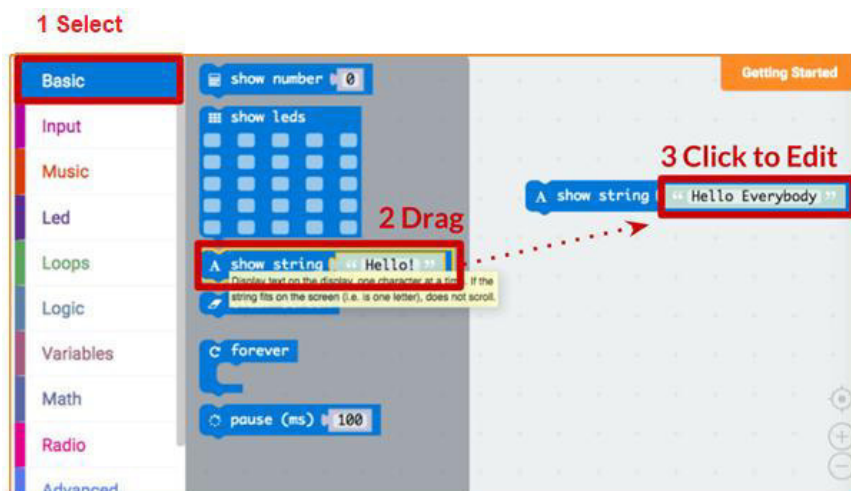
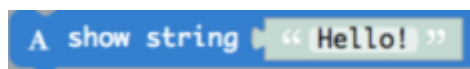
Otworzyć interfejs Javascript, aby przepisać skrypt w języku Javascript.

Pomocne odniesienia przy zapoznawaniu się z działaniem różnych bloków i funkcji.

Skróty do zmiany właściwości projektu. Tutaj można zmienić nazwę oraz usunąć projekt. **Reset** kasuje wszystkie zapisane projekty, dlatego należy zachować szczególną ostrożność. Zazwyczaj najlepiej używać **Delete Project**.

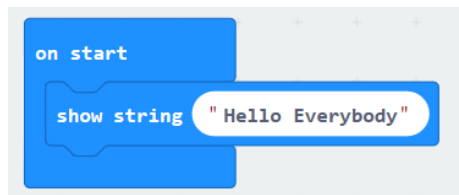
## Używanie paska LED

Rozpoczniemy od napisania kodu! Przy każdym kliknięciu szuflady kodów rozwinie się lista wszystkich dostępnych kodów. Przeciągnąć blok **show string** i kliknąć wewnątrz pola, aby edytować.



- |   |             |
|---|-------------|
| 1 | Wybrać      |
| 2 | Przeciągnąć |

- |   |                       |
|---|-----------------------|
| 3 | Kliknąć, aby edytować |
|---|-----------------------|

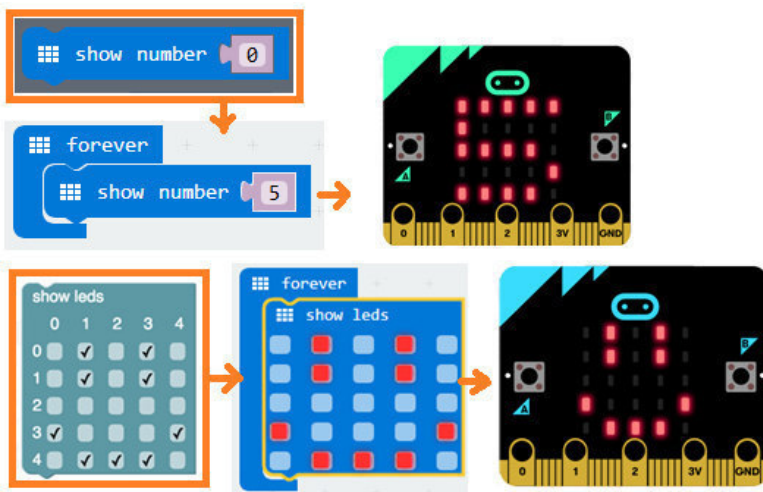


Sprawdź, co się stanie w symulatorze micro:bit!

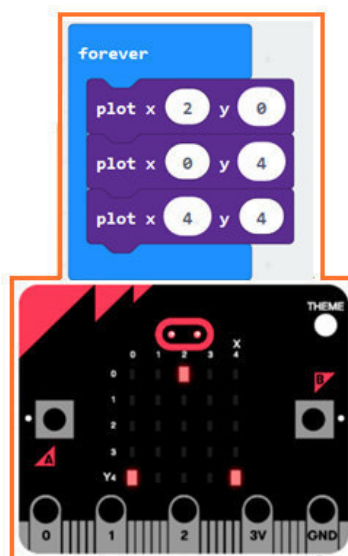
### Say Hello!

Wyświetlane teksty nazywane są ciągami **strings**.

Na ekranie LED można wyświetlić dowolną liczbę całkowitą przy użyciu **show number** lub dowolny obraz 5x5 px przy użyciu **show leds**.



W tym samym czasie można również nanieść jeden punkt LED za pomocą współrzędnych. Współrzędna (0,0) to lewy górny róg.

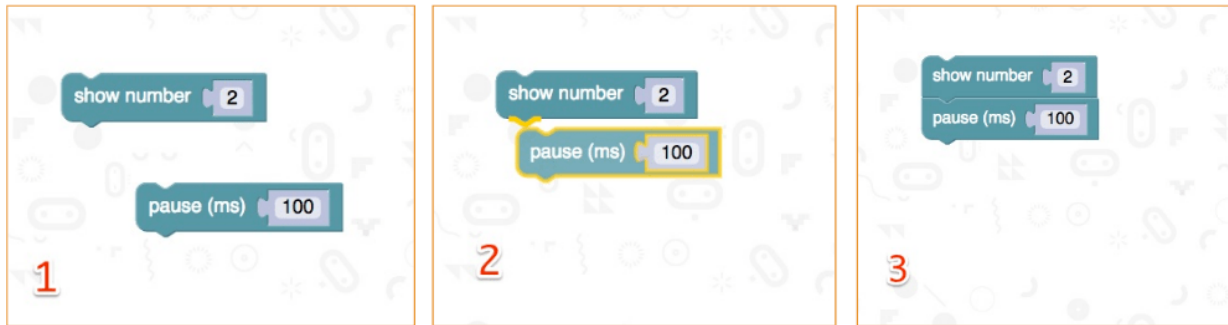


### Łączenie bloków

Kliknąć i przytrzymać blok, który ma być połączony. Przeciągnąć blok do bloku docelowego, aż do momentu podświetlenia się bocznej części bloku docelowego. Puścić - dwa bloki są już połączone!



Kliknięcie na pierwszy blok poruszy drugi; kliknięcie na drugi spowoduje odłączenie go od pierwszego bloku.



### Testowanie na micro:bit

Podłączyć micro:bit do komputera przy użyciu kabla micro-USB.

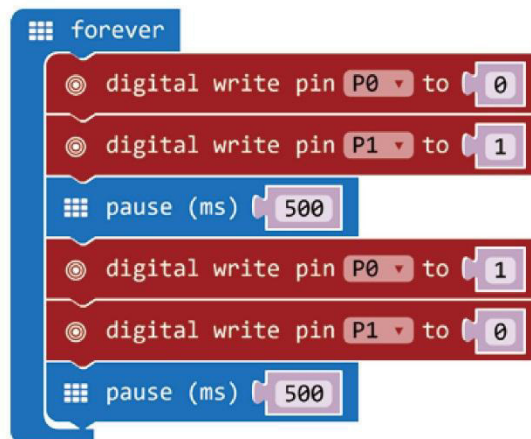
micro:bit przejdzie przez sekwencję domyślną, prosząc o naciśnięcie przycisków i zagranie w grę. Wypróbuj!

Kliknąć **Download**, aby przełożyć skrypt na **plik hex** i pobrać.

Przy prawidłowym ustawieniu Chrome, kolejny krok nie będzie konieczny. Przeciągnąć pobrany plik hex na dysk micro:bit lub kliknąć prawym przyciskiem myszki i wybrać **Send To** w systemie Windows®.

Właśnie zaprogramowałeś własne urządzenie!

Spójrz na poniższy kod.

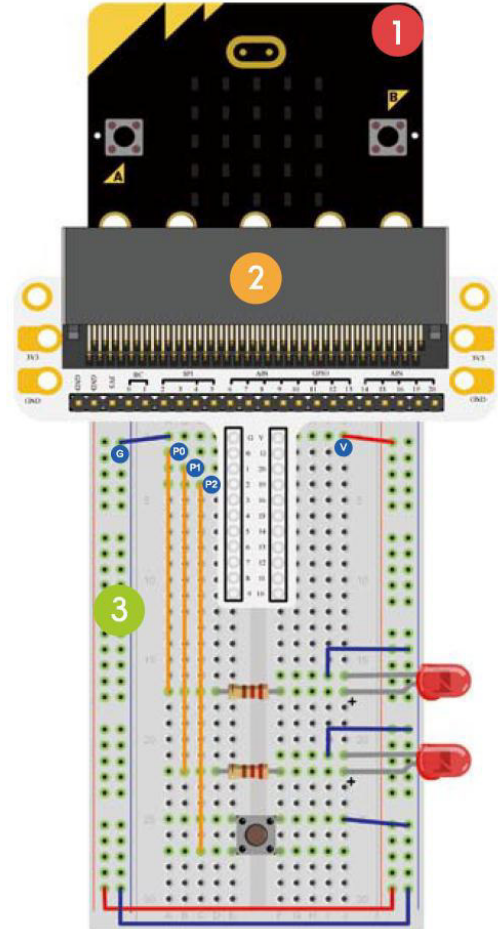
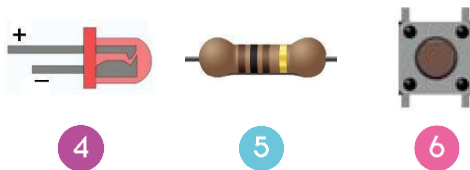


1.	Przeciągnąć bloki kodu i upuścić, aby stworzyć kod, jak pokazano. Blok <b>digital write pin P0 to 0</b> znajduje się w pozycji <b>Advanced</b> → <b>Pins</b> w szufladzie kodów. <b>forever</b> i <b>pause (ms) 100</b> znajdują się w pozycji <b>Basic</b> w szufladzie kodów.
2.	Ustawić wartość P0 na 0. LED0 wył. = niskie napięcie = 0 V = cyfrowe 0. Ustawić wartość P1 na 1. LED1 wł. = wysokie napięcie = 5 V = cyfrowe 1.
3.	Ustawić przerwę <b>pause</b> na 500 ms.
4.	Ustawić wartość P0 na 1. LED0 wł. = wysokie napięcie = 5 V = cyfrowe 1. Ustawić wartość P1 na 0. LED1 wył. = niskie napięcie = 0 V = cyfrowe 0.
5.	Ustawić przerwę <b>pause</b> na 500 ms.
6.	Po ukończeniu skompilujemy program i wygenerujemy plik .hex. Kliknąć przycisk [download] i zapisać plik .hex w folderze pobierania → C:\Pobrane. Plik .hex jest gotowy do załadowania na micro:bit. Podłączyć micro:bit do portu USB. Następnie przeciągnąć plik .hex na urządzenie wymienne micro:bit [removable device], aby załadować program.

Zobaczysz dwie migające naprzemiennie diody LED. A co powiesz na stworzenie sygnalizacji świetlnej RGB?

## 5.2 Przycisk

- 1 1x płytki micro:bit
- 2 1x adapter płytki prototypowej micro:bit
- 3 1x płytki prototypowa
- 4 2x czerwona dioda LED (polaryzacja: anoda (+) = długa nóżka, katoda (-) = krótka nóżka)
- 5 2x rezystor 100  $\Omega$  (brązowy/czarny/brązowy/złoty)
- 6 1x przycisk chwilowy



Przycisk służy do sterowania sekwencją migania diod LED. Po naciśnięciu przycisku diody LED zaczną migać na przemian. Puścić przycisk, aby wyłączyć LED.

Umieścić niezbędne elementy na płytce prototypowej, jak pokazano.

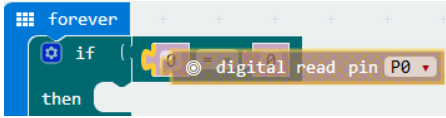
Spójrz na poniższy kod.

```

on start
  set pull pin P2 to up

forever
  if (digital read pin P2 = 0)
  then
    digital write pin P0 to 0
    digital write pin P1 to 1
    pause (ms) 500
    digital write pin P0 to 1
    digital write pin P1 to 0
    pause (ms) 500
  
```

1. Przeciągnąć bloki kodu i upuścić, aby stworzyć kod, jak pokazano. Bloki **digital read pin P0** i **digital write pin P0 to 0** znajdują się w pozycji **Advanced** → **Pins** w szufladzie kodów. **forever** i **pause (ms) 100** znajdują się w pozycji **Basic** w szufladzie kodów. Bloki **if then** i **logic equal function** znajdują się w pozycji **Logic** w szufladzie kodów. Przeciągnąć blok **logic equal function (=)** i upuścić na bloku **true**.

	 <p>Dwa bloki połączą się ze sobą. Przeciagnąć blok <b>digital read pin P0</b> i upuścić na <b>0</b> bloku <b>logic equal function (=)</b>, jak pokazano.</p> 
2.	Ustawić P2 jako pull-up.
3.	Ustawić <b>digital read pin</b> na P2.
4.	Ustawić <b>digital write pin P0 to 0</b> (aktywny niski (0 V)). Ustawić <b>digital write pin P1 to 1</b> (aktywny wysoki (5 V)).
5.	Ustawić przerwę <b>pause</b> na 500 ms.
6.	Ustawić <b>digital write pin P0 to 1</b> (aktywny wysoki (5 V)). Ustawić <b>digital write pin P1 to 0</b> (aktywny niski (0 V)).
7.	Ustawić przerwę <b>pause</b> na 500 ms.
8.	Po ukończeniu skompilujemy program i wygenerujemy plik .hex. Kliknąć przycisk [download] i zapisać plik .hex w folderze pobierania → C:\Pobrane. Plik .hex jest gotowy do załadowania na micro:bit. Podłączyć micro:bit do portu USB. Następnie przeciagnąć plik .hex na urządzenie wymienne micro:bit [removable device], aby załadować program.

Po naciśnięciu przycisku diody LED zaczną migać naprzemiennie. W jaki sposób teraz włączyć czerwoną diodę LED, naciskając przycisk, a zieloną diodę LED, zwalniając przycisk?

### 5.3 Potencjometr dostrojczy – trimpot

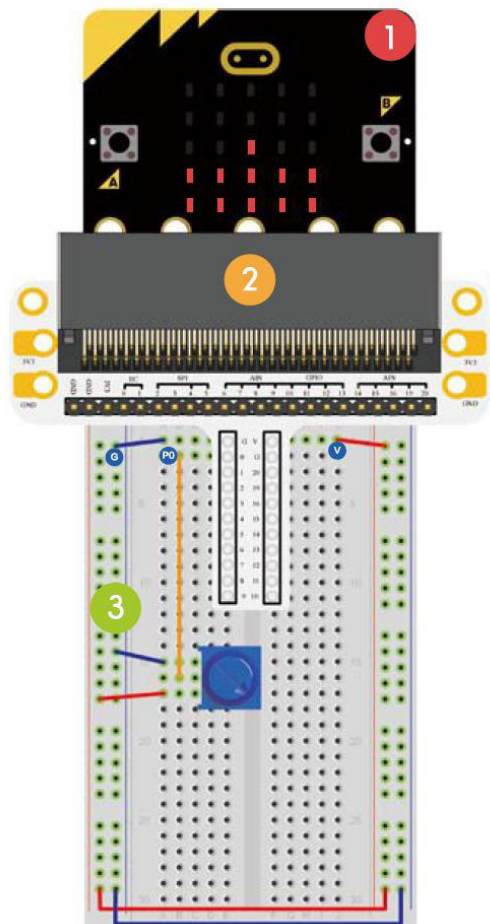
- 1 1x płytki micro:bit
- 2 1x adapter płytki prototypowej micro:bit
- 3 1x płytki prototypowa
- 4 1x trymer 10 kΩ



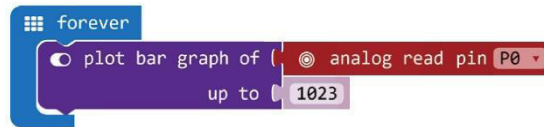
4

Teraz odczytamy napięcie wyjściowe potencjometru dostrojczego i wyświetlimy je na ekranie micro:bit za pomocą wykresu słupkowego.

Umieścić niezbędne elementy na płytce prototypowej, jak pokazano.



Spójrz na poniższy kod.



1.	Przeciagnąć bloki kodu i upuścić, aby stworzyć kod, jak pokazano. Blok <b>plot bar graph of 0 up to 0</b> znajduje się w pozycji <b>LED</b> w szufladzie kodów.
2.	Ustawić <b>analog read pin</b> na P0. Ten blok znajduje się w pozycji <b>Pins</b> w szufladzie kodów. Ustawić wartość maksymalnie do 1023.
3.	Po ukończeniu skompilujemy program i wygenerujemy plik .hex. Kliknąć przycisk [download] i zapisać plik .hex w folderze pobierania → C:\Pobrane. Plik .hex jest gotowy do załadowania na micro:bit. Podłączyć micro:bit do portu USB. Następnie przeciągnąć plik .hex na urządzenie wymienne micro:bit [removable device], aby załadować program.

Obrócić trymer. Napięcie wyświetli się na ekranie micro:bit w postaci wykresu słupkowego. Jeśli napięcie jest równe 0, na ekranie LED ukaże się jedynie pojedynczy piksel. Przy wartości równej 3,3 V podświetli się cały ekran. W jaki sposób teraz użyć trymera, aby wyregulować jasność LED?

## 5.4 Fotokomórka

- 1 1x płytki micro:bit
- 2 1x adapter płytki prototypowej micro:bit
- 3 1x płytki prototypowa
- 4 1x fotokomórka
- 5 1x rezystor 10 kΩ (brązowy/czarny/czarny/czerwony/brązowy)

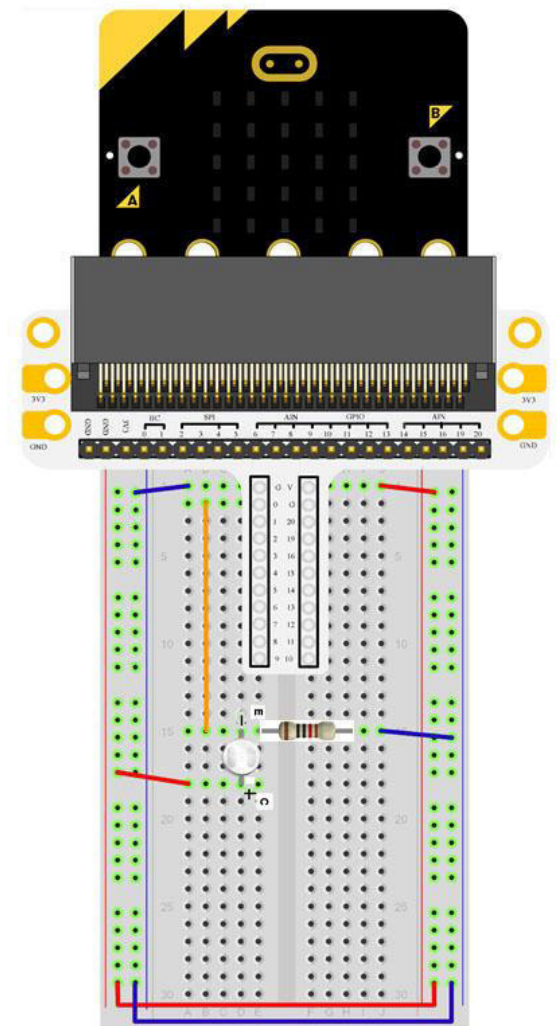


4

Krótką nóżką =  
kolektor  
Anoda dodatnia  
Długą nóżką = emiter  
Katoda ujemna



5



Użyjemy fotokomórki do sterowania jasnością ekranu micro:bit.

Umieścić niezbędne elementy na płytce prototypowej, jak pokazano.

Spójrz na poniższy kod.

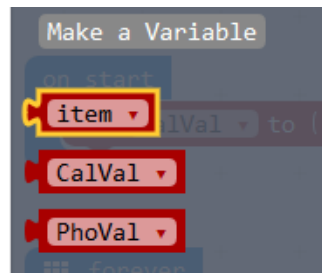
1. Najpierw należy stworzyć dwie zmienne. Przejść do **Variables** w szufladzie kodów i kliknąć **Make a Variable**.

New variable name:



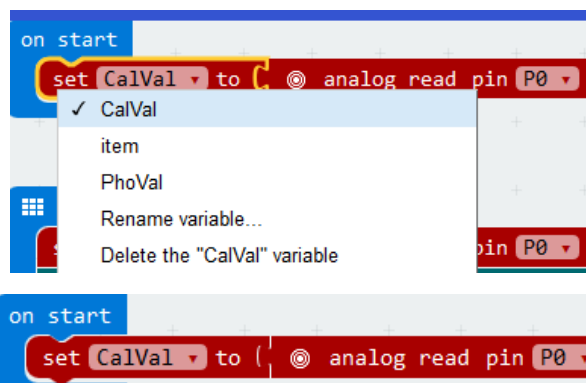

Wprowadzić w okienku **CalVal** i kliknąć **Ok**. Wprowadzić w okienku **PhoVal** i kliknąć **Ok**.

Pod **Variables** w szufladzie kodów pojawią się dwie nowe zmienne.



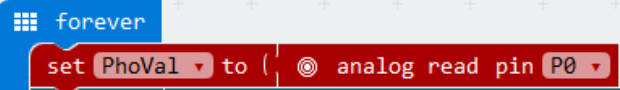
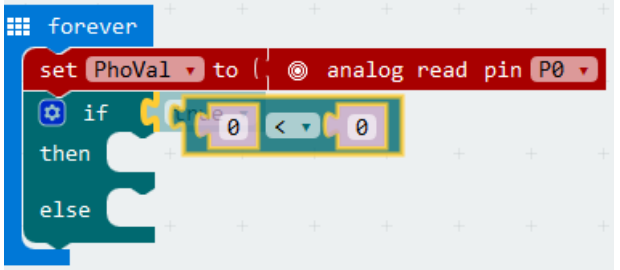
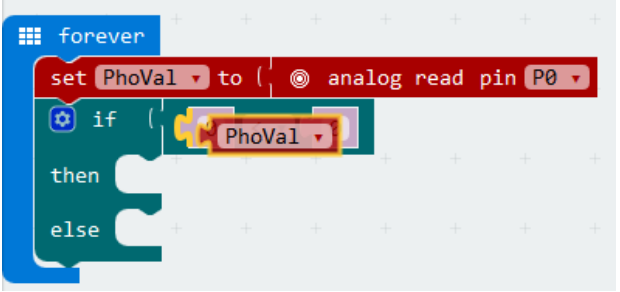
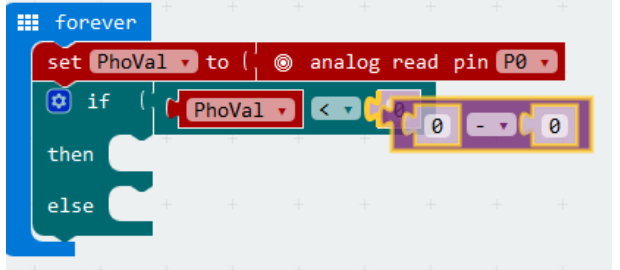
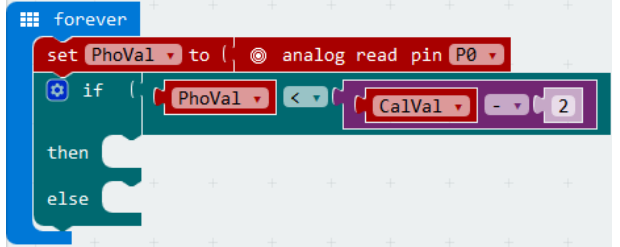
Te zmienne będą potrzebne później w celu zapisania danych w rejestrze.

Bloki **logic smaller than function (<)** oraz **if then else** znajdują się w pozycji **Logic** w szufladzie kodów. Blok **mathematical function minus (-)** można znaleźć w pozycji **Math** w szufladzie kodów. Blok **show icon** znajduje się w pozycji **Basic** w szufladzie kodów. Blok **clear screen** znajduje się w pozycji **Basic** → **More** w szufladzie kodów. Blok **set item to 0** znajduje się w pozycji **Variables** w szufladzie kodów. Kliknąć strzałkę i wybrać **CalVal** lub **PhoVal**.



2. Wybrać zmienną **CalVal** i ustawić **analog read pin** na P0.



3.	<p>W bloku <b>forever</b> wybrać zmienną <b>PhoVal</b> i ustawić <b>analog read pin</b> na P0.</p> 
4.	<p>Przecięgnąć <b>logic smaller than function (&lt;)</b> obok bloku <b>if</b> i upuścić na bloku <b>true</b>.</p>  <p>Następnie przecięgnąć zmienną <b>PhoVal</b> (z <b>Variables</b> w szufladzie kodów) i upuścić na pierwszą wartość 0 bloku <b>logic smaller than function (&lt;)</b>.</p>  <p>Przecięgnąć blok <b>mathematical function minus</b> i upuścić na drugą wartość 0 bloku <b>logic smaller than function (&lt;)</b>.</p>  <p>Następnie przecięgnąć zmienną <b>CalVal</b> (z <b>Variables</b> w szufladzie kodów) i upuścić na pierwszą wartość 0 bloku <b>mathematical function minus (-)</b>. Ustawić drugą wartość 0 bloku <b>mathematical function minus (-)</b> na 2.</p> 
5.	<p>Przecięgnąć blok <b>show icon</b> i upuścić obok bloku <b>then</b>. Przecięgnąć blok <b>clear screen</b> i upuścić obok bloku <b>else</b>.</p>
6.	<p>Jeśli <b>PhoVal</b> jest mniejsza niż <b>CalVal</b> - 2, na wyświetlaczu ukaże się serce. W innym wypadku wyłączy się.</p>



- 7.** Po ukończeniu skompilujemy program i wygenerujemy plik .hex. Kliknąć przycisk [download] i zapisać plik .hex w folderze pobierania → C:\Pobrane. Plik .hex jest gotowy do załadowania na micro:bit.  
Podłączyć micro:bit do portu USB. Następnie przeciągnąć plik .hex na urządzenie wymienne micro:bit [removable device], aby załadować program.

Uwaga: zresetować micro:bit, aby skalibrować wartość referencyjną zgodnie z aktualnym poziomem jasności. Aby właściwie uruchomić program, należy rozpocząć od włączenia oświetlenia.

Gdy oświetlenie jest włączone, nic się nie wyświetli. Gdy oświetlenie jest wyłączone, wyświetli się serce. W jaki sposób teraz można wykorzystać fotokomórkę do sterowania LED?

## 5.5 LED RGB

- 1 1x płytki micro:bit
- 2 1x adapter płytki prototypowej micro:bit
- 3 1x płytki prototypowa
- 4 1x LED RGB (wspólna katoda)
- 5 3x 10 Ω rezystor (brązowy/czarny/brązowy/złoty)

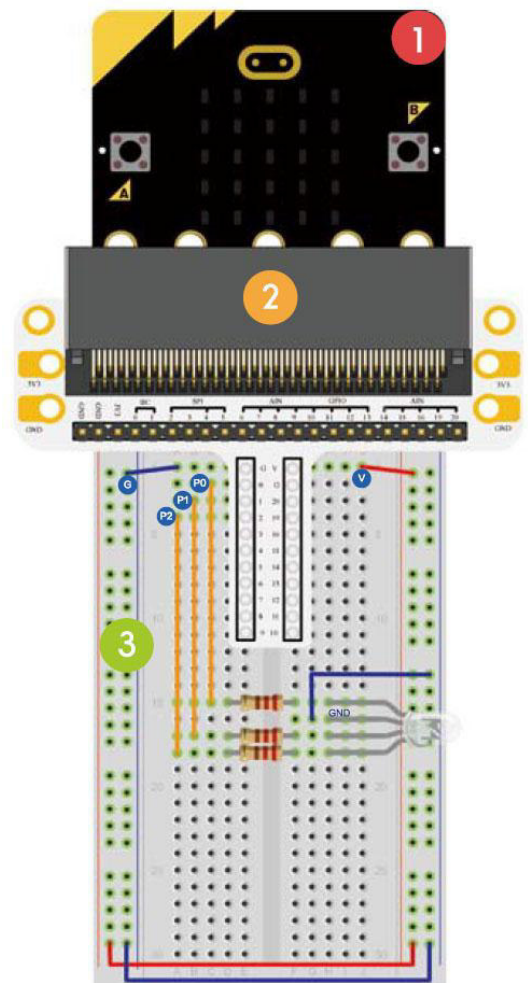


4

1. Czerwona (anoda +)
2. Masa (katoda -) – najdłuższa nóżka
3. Zielona (anoda +)
4. Niebieska (anoda +)



5



Następnie zaprogramujemy stopniową zmianę kolorów LED RGB między czerwonym, zielonym i niebieskim.

Umieścić niezbędne elementy na płytce prototypowej, jak pokazano.

Spójrz na poniższy kod.

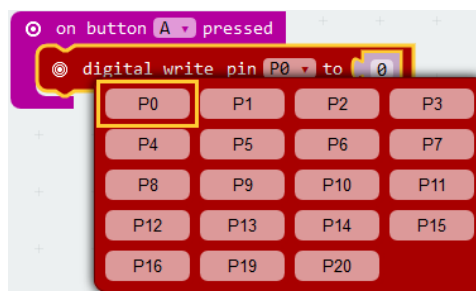
```

on button A pressed
  digital write pin P0 to 1
  digital write pin P1 to 0
  digital write pin P2 to 0

on button B pressed
  digital write pin P0 to 0
  digital write pin P1 to 1
  digital write pin P2 to 0

on button A+B pressed
  digital write pin P0 to 0
  digital write pin P1 to 0
  digital write pin P2 to 1
  
```

- Przeciagnąć bloki kodu i upuścić, aby stworzyć kod, jak pokazano.  
 Blok **on button A pressed** znajduje się w pozycji **Input** w szufladzie kodów. Blok **digital write pin P0 to 0** znajduje się znaleźć w pozycji **Pins** w szufladzie kodów.  
 Wybrać opcję A w bloku **on button A pressed**.  
 Przeciagnąć 3 bloki **digital write pin P0 to 0** i wstawić w blok **on button A pressed**.  
 Ustawić pin P0 w pierwszym bloku **digital write pin P0 to 0** i zmienić wartość 0 na 1 (czerwona dioda LED wł.).  
 Ustawić pin P1 w drugim bloku **digital write pin P0 to 0** i ustawić wartość na 0 (zielona dioda LED wył.).  
 Ustawić pin P2 w trzecim bloku **digital write pin P0 to 0** i ustawić wartość na 0 (niebieska dioda LED wył.).



```

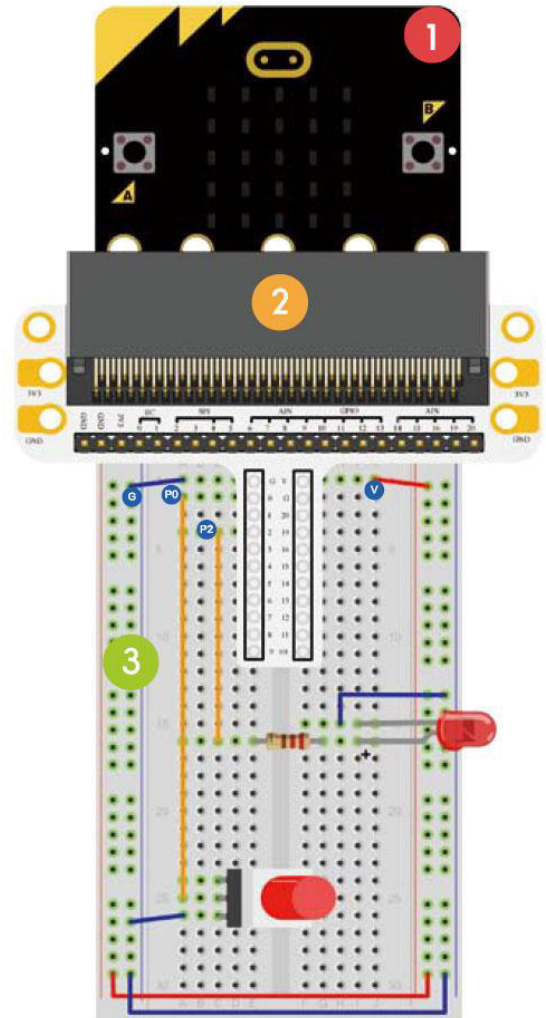
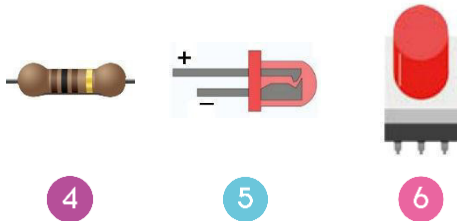
on button A pressed
  digital write pin P0 to 1
  digital write pin P1 to 0
  digital write pin P2 to 0
  
```

2.	Podobnie skompilować dwa bloki <b>on button B/A+B pressed</b> , aby utworzyć pełny kod.
3.	Po ukończeniu skompilujemy program i wygenerujemy plik .hex. Kliknąć przycisk [download] i zapisać plik .hex w folderze pobierania → C:\Pobrane. Plik .hex jest gotowy do załadowania na micro:bit. Podłączyć micro:bit do portu USB. Następnie przeciągnąć plik .hex na urządzenie wymienne micro:bit [removable device], aby załadować program.

Nacisnąć przycisk A, aby zaświecić czerwoną diodę LED, nacisnąć przycisk B, aby zaświecić zieloną diodę LED, nacisnąć jednocześnie przyciski A i B, aby zaświecić niebieską diodę LED. W jaki sposób stworzyć teraz łagodne gradientowe oświetlenie RGB?

## 5.6 Przełącznik samo-blokujący

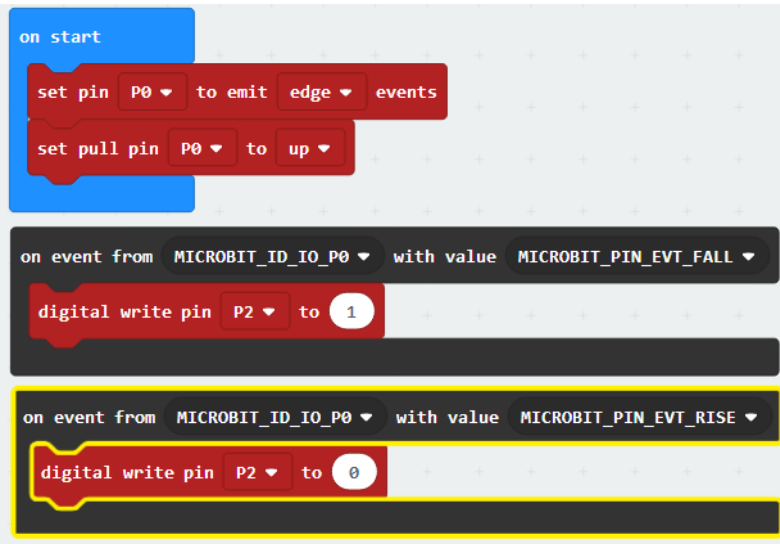
- 1 1x płytki micro:bit
- 2 1x adapter płytki prototypowej micro:bit
- 3 1x płytki prototypowa
- 4 1x rezystor 100  $\Omega$  (brązowy/czarny/brązowy/złoty)
- 5 1x czerwona dioda LED (polaryzacja: anoda (+) = długa nóżka, katoda (-) = krótka nóżka)
- 6 1x przełącznik samoblokujący lub dwustabilny



Zamierzamy odczytać temperaturę otoczenia (dane) analogowego czujnika temperatury i wyświetlić dane na micro:bit.

Umieścić niezbędne elementy na płytce prototypowej, jak pokazano.

Spójrz na poniższy kod.



1.	Przeciągnąć bloki kodu i upuścić, aby stworzyć kod, jak pokazano. Bloki <b>set pin P0 to emit edge events</b> i <b>set pull P0 to up</b> znajdują się w pozycji <b>Pins</b> → <b>More</b> w szufladzie kodów. Bloki <b>on event from MICROBIT...EVT_FALL/RISE</b> znajdują się w pozycji <b>Control</b> w szufladzie kodów.
2.	Ustawić typ zdarzenia na <b>edge</b> . Ustawić pin pull na <b>P0</b> i <b>up</b> .
3.	Teraz należy stworzyć blok typu fall event. Przeciągnąć blok <b>on event from MICROBIT...EVT<sup>o</sup>FALL</b> i upuścić pod blokiem <b>on start</b> . Kliknąć pierwszą strzałkę i wybrać opcję <b>MICROBIT_ID_IO_P0</b> w menu rozwijanym. Kliknąć drugą strzałkę i wybrać opcję <b>MICROBIT_PIN_EVENT_FALL</b> w menu rozwijanym. Przeciągnąć blok <b>digital write pin P0 to 0</b> i upuścić w bloku zdarzeń. Ustawić pin P0 na P2 i zmienić wartość 0 na 1. Blok <b>digital write pin P0 to 0</b> znajduje się w pozycji <b>Pins</b> w szufladzie kodów.
4.	Wykonać te same kroki dla drugiego bloku zdarzeń. Wtedy wystarczy kliknąć drugą strzałkę i wybrać opcję <b>MICROBIT_PIN_EVENT_RISE</b> w menu rozwijanym. Ustawić również pin P0 na P2 bloku <b>digital write pin P0 to 0</b> i ustawić wartość na 0.
5.	Po ukończeniu skompilujemy program i wygenerujemy plik .hex. Kliknąć przycisk [download] i zapisać plik .hex w folderze pobierania → C:\Pobrane. Plik .hex jest gotowy do załadowania na micro:bit. Podłączyć micro:bit do portu USB. Następnie przeciągnąć plik .hex na urządzenie wymienne micro:bit [removable device], aby załadować program.

Nacisnąć przełącznik samoblokujący – włączy się dioda LED. Nacisnąć ponownie, aby wyłączyć diodę LED. W jaki sposób teraz sterować wyświetlaczem micro:bit za pomocą tego przełącznika samoblokującego?



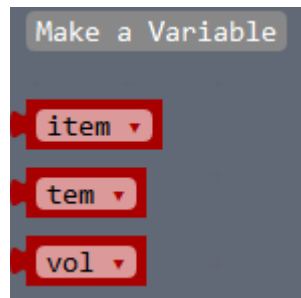
2. Najpierw należy stworzyć dwie zmienne. Przejść do pozycji **Variables** w szufladzie kodów i kliknąć **Make a Variable**.

New variable name:

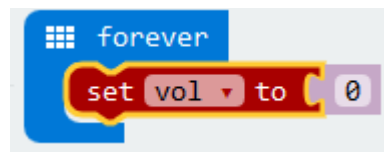
Ok ✓ Cancel ✕

Wprowadzić **vol** w okienku i kliknąć **Ok**. Wprowadzić **tem** w okienku i kliknąć **Ok**.

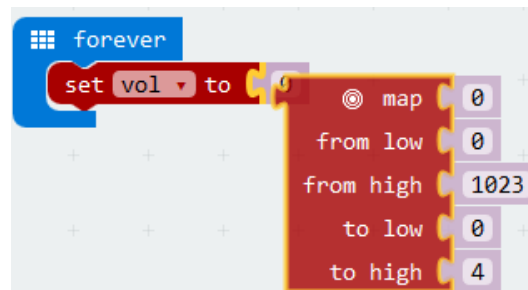
Pod pozycją **Variables** w szufladzie kodów widoczne będą dwie nowe zmienne.



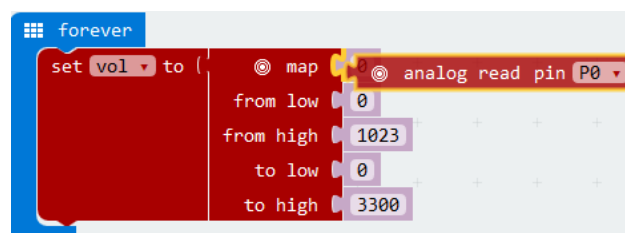
Te zmienne będą potrzebne później w celu zapisania danych w rejestrze. Przeciągnąć i upuścić blok **set item to** w bloku **forever** i wybrać opcję **vol** przy użyciu strzałki.



Następnie przeciągnąć blok **map/from low/from high/to low/to high** i upuścić na 0 obok bloku **set item to**.

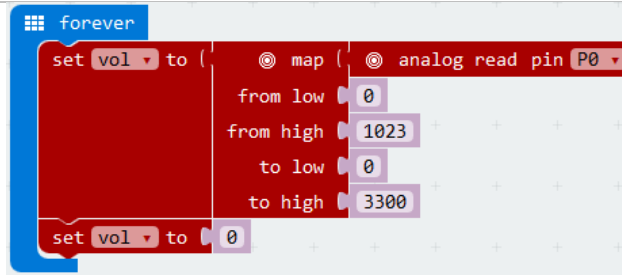


Przeciągnąć i upuścić blok **analog read pin** obok bloku **map/from low/from high/to low/to high** i ustawić wartość **to high** w zakresie od 4 do 3300.

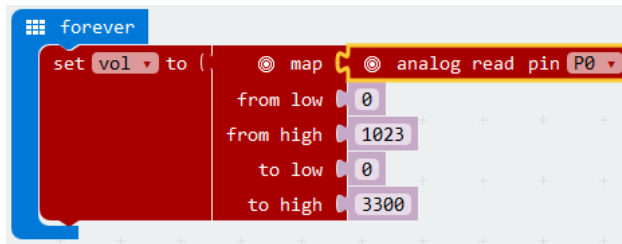


Teraz przeciągnąć blok **set item to** i upuścić pod pierwszym blokiem. Zmieniać zmienną **item** na **vol** przy użyciu strzałki.





Zmierzone napięcie w mV przez **analog read pin P0** jest wartością 8-bitową 0-1023 (0-3,3 V) i jest mapowane od wartości niskiej (0 lub 0 V) do wysokiej (3300 mV lub 3,3 V). Zmierzone napięcie (mV) zapisywane jest w zmiennej **vol**.

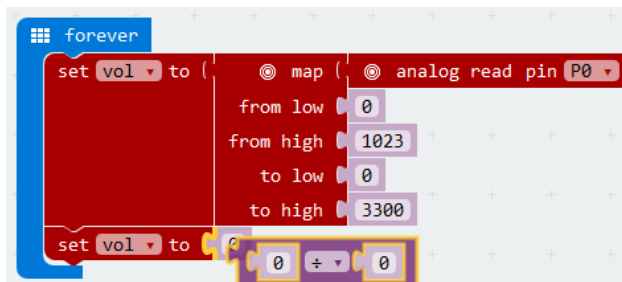


Teraz napiszemy wzór, aby przekonwertować zmierzone napięcie w temperaturę.

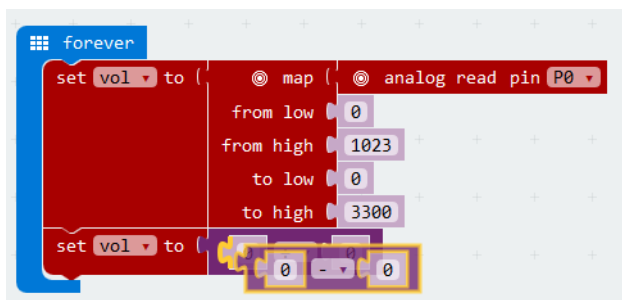
$$\text{Temperature } (^{\circ}\text{C}) = \frac{(\text{Output voltage (mV)} - 500)}{10}$$

W tym wzorze napięcie wyjściowe jest zmienną **vol** (zamierzoną przez czujnik TMP36). Wynik wzoru zostanie zapisany w formie zmiennej **tem**.

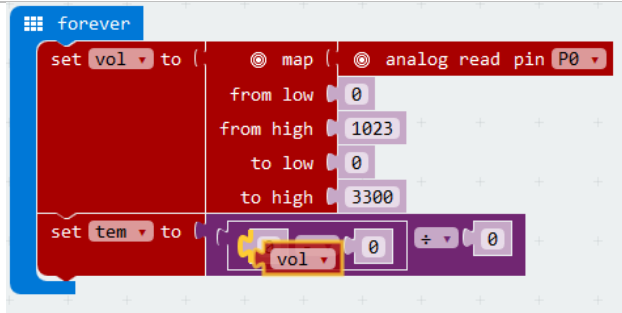
Napiszmy teraz wzór za pomocą bloków. Przeciągnąć **mathematical function divide (÷)** i upuścić na wartości 0 bloku **set vol to**.



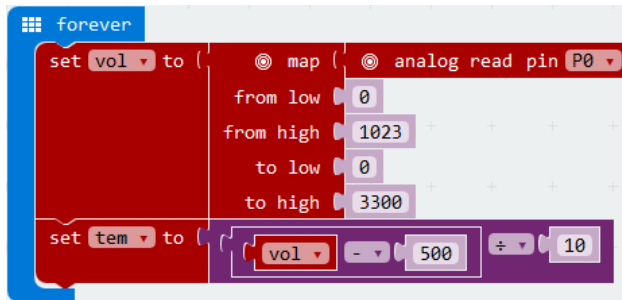
Przeciągnąć **mathematical function minus (-)** i upuścić na pierwszej wartości 0 **mathematical function divide (÷)**.



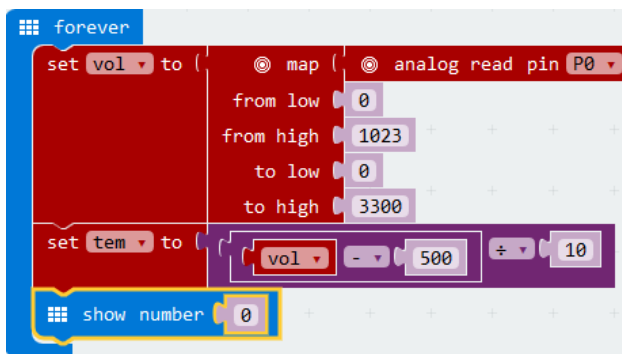
Teraz ustawić drugą zmienną od **vol** do **tem** oraz przeciągnąć i upuścić zmienną **vol** na pierwszej wartości 0 **mathematical function minus (-)**. Zmienna **vol** znajduje się w pozycji **Variables** w szufladzie kodów.



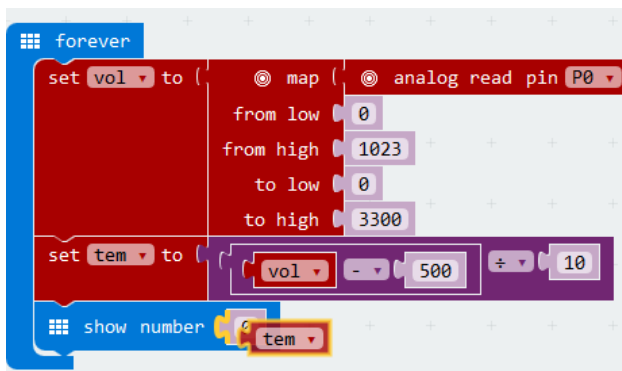
Zmienić wartość 0 **mathematical function minus (-)** na 500. Ustawić również wartość 10 **mathematical function divide (÷)**.



Przecięgnąć blok **show number** i upuścić pod blokiem **set tem to**.



Na koniec przecięgnąć zmienną **tem** i upuścić na wartość 0 bloku **show number**.



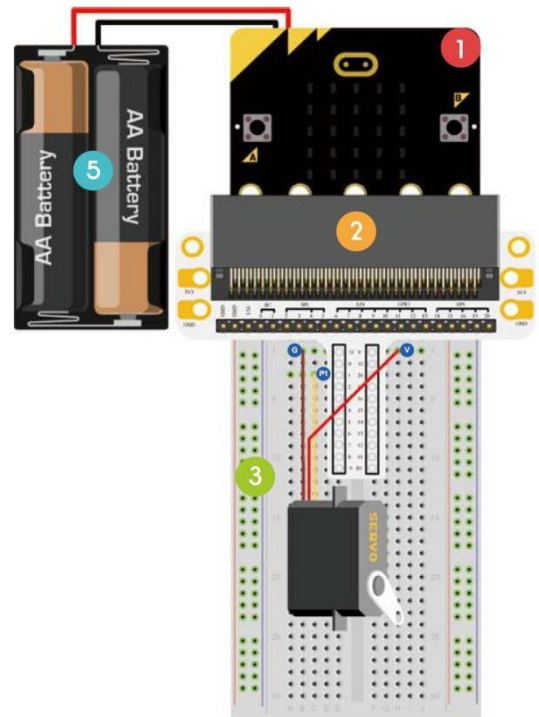
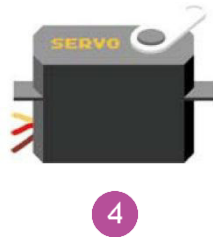
Właśnie napisałeś kod!

3. Po ukończeniu skompilujemy program i wygenerujemy plik `.hex`. Kliknąć przycisk [download] i zapisać plik `.hex` w folderze pobierania → `C:\Pobrane`. Plik `.hex` jest gotowy do załadowania na micro:bit.  
Podłączyć micro:bit do portu USB. Następnie przecięgnąć plik `.hex` na urządzenie wymienne micro:bit [removable device], aby załadować program.

Zobaczysz teraz diody LED migające naprzemiennie. W jaki sposób teraz wyświetlić temperaturę w stopniach Fahrenheita?

## 5.8 Siłownik

- 1 1x płytki micro:bit
- 2 1x adapter płytki prototypowej micro:bit
- 3 1x płytki prototypowa
- 4 1x miniaturowy siłownik
- 5 1x uchwyt baterii z 2 bateriami AA 1,5 V



Teraz stworzymy siłownik obracający się w sposób ciągły w zakresie ruchu (0-180°).

Umieścić niezbędne elementy na płytce prototypowej, jak pokazano.

Spójrz na poniższy kod.

```

forever
  servo write pin P1 to 0
  pause (ms) 2000
  servo write pin P1 to 180
  pause (ms) 2000

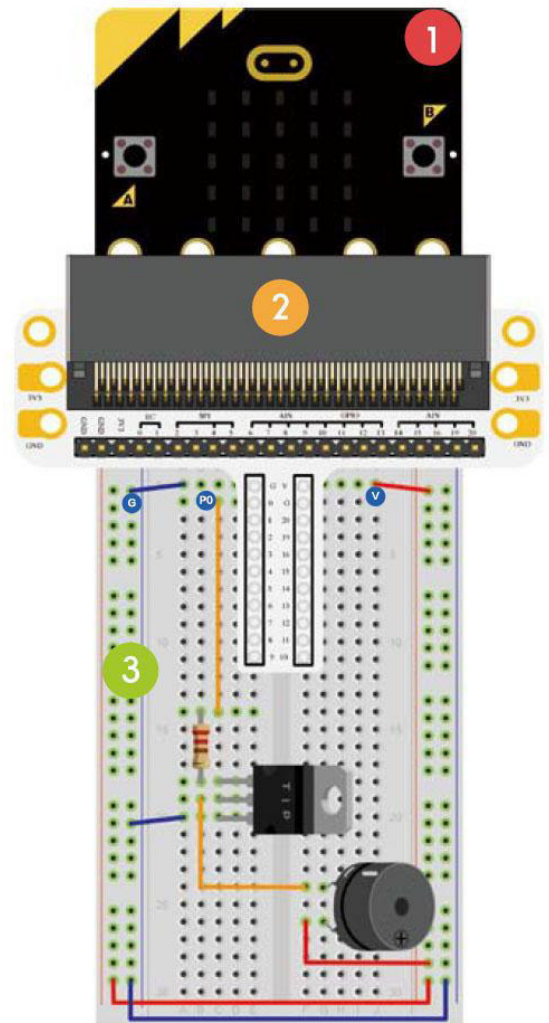
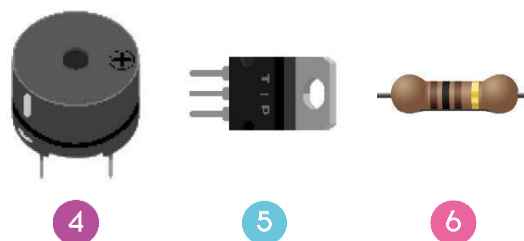
```

1.	Przeciagnąć bloki kodu i upuścić, aby stworzyć kod, jak pokazano. Blok <b>servo write pin P0 to 180</b> znajduje się w pozycji <b>Pins</b> w szufladzie kodów. <b>forever</b> i <b>pause (ms) 100</b> znajdują się w pozycji <b>Basic</b> w szufladzie kodów.
2.	Zmienić <b>P0</b> na <b>P1</b> i ustawić wartość 0.
3.	Ustawić przerwę <b>pause</b> na 2000 ms.
4.	Zmienić <b>P0</b> na <b>P1</b> . Zmienić również wartość z 0 na 180.
5.	Ustawić przerwę <b>pause</b> na 2000 ms.
6.	Po ukończeniu skompilujemy program i wygenerujemy plik .hex. Kliknąć przycisk [download] i zapisać plik .hex w folderze pobierania → C:\Pobrane. Plik .hex jest gotowy do załadowania na micro:bit. Podłączyć micro:bit do portu USB. Następnie przeciagnąć plik .hex na urządzenie wymienne micro:bit [removable device], aby załadować program.

Można zauważyć, że siłownik obraca się od 0 do 180 stopni. W jaki sposób stworzyć teraz termometr tarczowy z czujnikiem temperatury i siłownikiem?

## 5.9 Brzęczyk

- 1 1x płytki micro:bit
- 2 1x adapter płytki prototypowej micro:bit
- 3 1x płytki prototypowa
- 4 1x mini głośnik
- 5 1x N-kanałowy MOSFET
- 6 1x rezystor 100 ? (brązowy/czarny/brązowy/złoty)



Teraz zajmiemy się uruchomieniem brzęczyka.

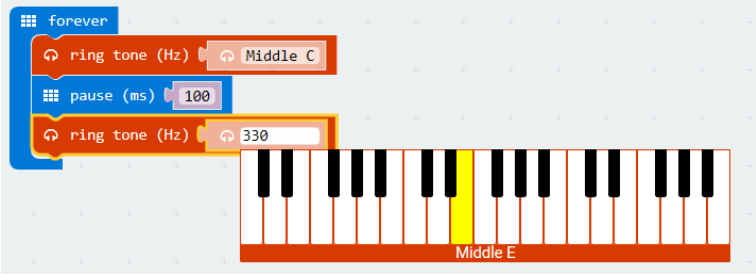
Umieścić niezbędne elementy na płytce prototypowej, jak pokazano.

Spójrz na poniższy kod.

```

forever
  ring tone (Hz) Middle C
  pause (ms) 100
  ring tone (Hz) Middle E
  pause (ms) 100
  ring tone (Hz) Middle G
  pause (ms) 100
  ring tone (Hz) Middle E
  pause (ms) 100
  
```

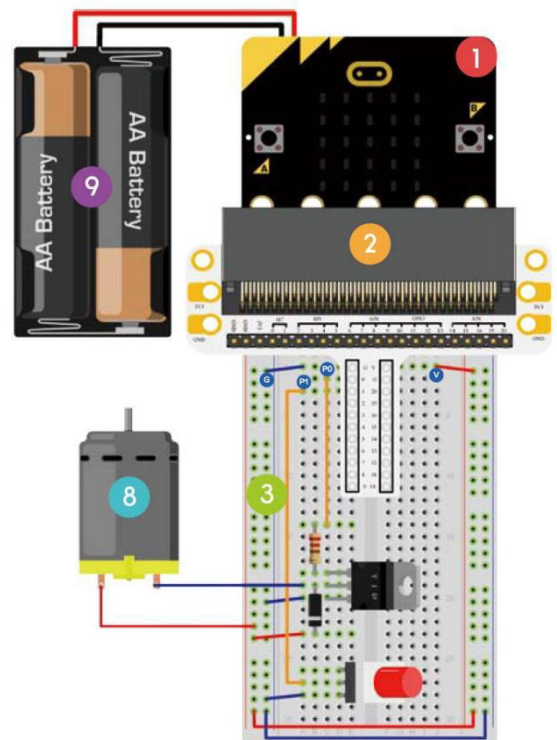
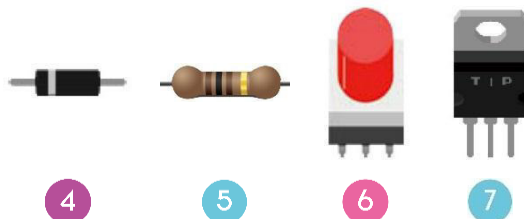
1. Przeciągnąć bloki kodu i upuścić, aby stworzyć kod, jak pokazano. **forever** i **pause (ms) 100** znajdują się w pozycji **Basic** w szufladzie kodów. Blok **ring tone (Hz)** znajduje się w pozycji **Music** w szufladzie kodów.

2.	<p>Ustawić pierwszy sygnał dzwonka <b>ring tone (Hz)</b> na środkowe C <b>middle C</b>. W tym celu kliknąć pole tekstowe i wybrać odpowiadający klawisz fortepianowy.</p>  <p>Wyemitowany zostanie również odpowiedni dźwięk.</p>
3.	Ustawić przerwę <b>pause</b> na 100 ms.
4.	Wykonać powyższe czynności dla pozostałych sygnałów dzwonka i przerw.
5.	<p>Po ukończeniu skompilujemy program i wygenerujemy plik .hex. Kliknąć przycisk [download] i zapisać plik .hex w folderze pobierania → C:\Pobrane. Plik .hex jest gotowy do załadowania na micro:bit.</p> <p>Podłączyć micro:bit do portu USB. Następnie przeciągnąć plik .hex na urządzenie wymienne micro:bit [removable device], aby załadować program.</p>

Rozlegnie się dźwięk brzęczyka. W jaki sposób zaprogramować teraz ulubioną wylicznankę?

## 5.10 Silnik

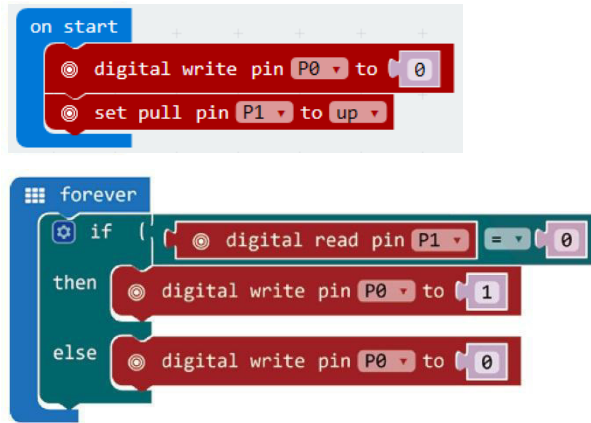
- 1 1x płytki micro:bit
- 2 1x adapter płytki prototypowej micro:bit
- 3 1x płytki prototypowa
- 4 1x dioda
- 5 1x rezystor 100 ?  
(brązowy/czarny/brązowy/złoty)
- 6 1x przełącznik samoblokujący lub dwustabilny
- 7 1x N-kanalowy MOSFET
- 8 1x mini silnik
- 9 1x uchwyt baterii z 2 bateriami AA 1,5 V



Użyjemy przełącznika do sterowania włączaniem i wyłączeniem silnika.

Umieścić niezbędne elementy na płytce prototypowej, jak pokazano.

Spójrz na poniższy kod.



1.	Przeciagnąć bloki kodu i upuścić, aby stworzyć kod, jak pokazano.
2.	Blok <b>on start</b> włączy się tylko raz w celu uruchomienia programu.
3.	Przestawić wartość <b>P0</b> na <b>0</b> .
4.	Ustawić blok <b>pull pin</b> na <b>P1</b> i <b>up</b> .
5.	<p>W bloku <b>forever</b>, ustawić <b>digital read pin</b> na <b>P1</b>. Ustawić <b>mathematical function equal (=)</b> na 0.</p> <p>W rzeczywistości jest to schemat. Nie trzeba dołączać rezystora pull-up jako osprzęt na płytce prototypowej. Funkcja pull-up jest zaprogramowana (patrz krok 3-4) w kodzie i zastępuje potrzebę zastosowania rezystora na płytce prototypowej.</p>
6.	Po naciśnięciu przełącznika ustawić wysokie napięcie na P0. Zmienić wartość 0 na 1 (5 V). Silnik zaczyna pracować.
7.	Po zwolnieniu przycisku ustawić niskie napięcie na P0. Ustawić wartość na 0. Silnik przestaje pracować.
8.	<p>Po ukończeniu skompilujemy program i wygenerujemy plik .hex. Kliknąć przycisk [download] i zapisać plik .hex w folderze pobierania → C:\Pobrane. Plik .hex jest gotowy do załadowania na micro:bit.</p> <p>Podłączyć micro:bit do portu USB. Następnie przeciagnąć plik .hex na urządzenie wymienne micro:bit [removable device], aby załadować program.</p>

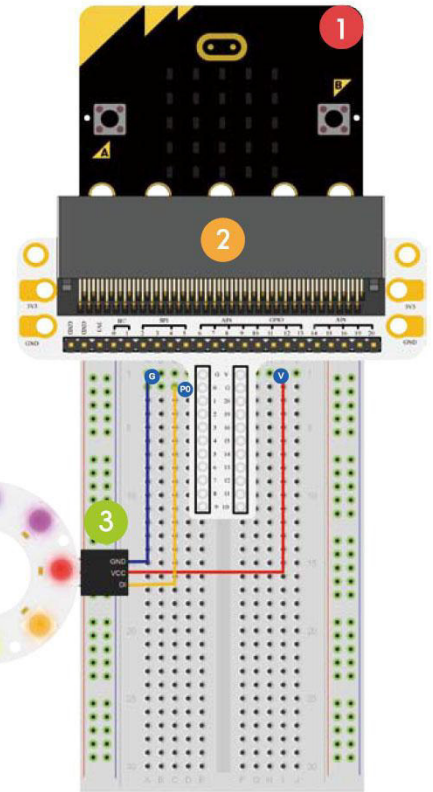
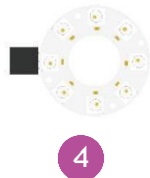
Nacisnąć przycisk, aby uruchomić silnik i zwolnić, aby zatrzymać. W jaki sposób teraz użyć potencjometru dostrojczego (trimpot), aby regulować prędkość silnika?

Uwaga: Napięcie micro:bit wynosi jedynie 3,3 V, co może być niewystarczające do obsługi wentylatora. Obrócenie łopatki może być konieczne do rozpoczęcia pracy wentylatora.



## 5.11 Tęcza LED

- 1 1x płytki micro:bit
- 2 1x adapter płytki prototypowej micro:bit
- 3 1x płytki prototypowa
- 4 1x pierścień LED RGB



Teraz zrobimy tęczę z ośmiu diod RGB ustawionych w pierścień.

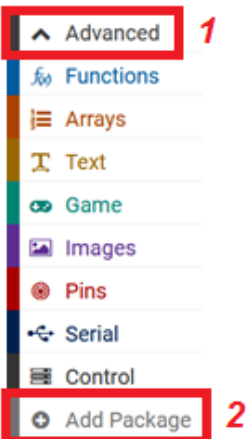
Umieścić niezbędne elementy na płytce prototypowej, jak pokazano.

Spójrz na poniższy kod.

```

on start
  set strip to NeoPixel at pin P0 with 8 leds as RGB (GRB format)
  strip show rainbow from 1 to 360

forever
  strip show
  strip rotate pixels by 1
  pause (ms) 100
  
```

<b>1.</b>	Przeciagnąć bloki kodu i upuścić, aby stworzyć kod, jak pokazano.
<b>2.</b>	<p>Wyszukać i dodać bibliotekę NeoPixel. Przejsć do Advanced → Add Package (Extensions) i wprowadzić NeoPixel.</p> 

Add Package... ?

1 neopixel

2

Q

devices

BETA - Camera, remote control and other Bluetooth services. App required.

Beta

bluetooth

Bluetooth services



neopixel

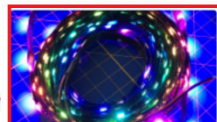
AdaFruit NeoPixel driver

Następnie wybrać bibliotekę NeoPixel.

Add Package... ?

neopixel

Q



neopixel

A Neo-Pixel package for pxt-microbit



bitbot

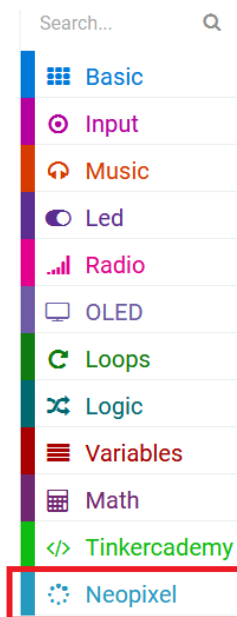
A BitBot package for pxt-microbit.



BitBot

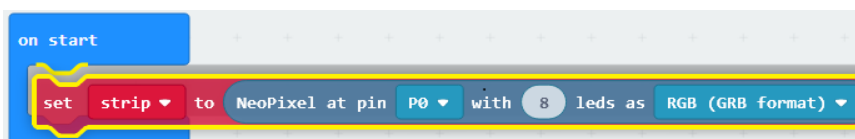
Official Microsoft MakeCode package for the BitBot robot. <https://4tronix.co.uk/bitbot/>

Biblioteka jest już pobrana i dodana do szuflady kodów.



Bloki **NeoPixel at pin P0 with 24 leds**, **item show rainbow from 1 to 360**, **item show** oraz **items rotate pixels by 1** znajdują się w pozycji **NeoPixel** w szufladzie kodów.

3. Przeciągnąć blok **set strip to NeoPixel at pin P0 with 24 leds as RGB (GRB format)** i upuścić w bloku **on start**. Ustawić pin na **P0**, a wartość 24 zmienić na 8 (8 diod LED w pierścieniu LED).



4.	<p>Teraz przeciągnąć blok <b>set strip to rainbow from 1 to 360</b> i upuścić pod blokiem <b>set strip to NeoPixel at pin P0</b>.</p> 
5.	Uzupełnić kod zgodnie z przykładem.
6.	<p>Po ukończeniu skompilujemy program i wygenerujemy plik .hex. Kliknąć przycisk [download] i zapisać plik .hex w folderze pobierania → C:\Pobrane. Plik .hex jest gotowy do załadowania na micro:bit.</p> <p>Podłączyć micro:bit do portu USB. Następnie przeciągnąć plik .hex na urządzenie wymienne micro:bit [removable device], aby załadować program.</p>

Na pierścieniu LED widzimy obracającą się tęczę. W jaki sposób teraz sprawić, że pierścień będzie mrugał jak oko?

**Należy używać wyłącznie oryginalnych akcesoriów. Firma Velleman nv nie ponosi odpowiedzialności za uszkodzenia lub urazy wynikające z (niewłaściwego) korzystania z niniejszego urządzenia. Aby uzyskać więcej informacji dotyczących produktu oraz najnowszą wersję niniejszej instrukcji, należy odwiedzić naszą stronę internetową [www.velleman.eu](http://www.velleman.eu). Informacje zawarte w niniejszej instrukcji obsługi mogą ulec zmianie bez wcześniejszego powiadomienia.**

#### © INFORMACJA O PRAWACH AUTORSKICH

**Właścicielem praw autorskich do niniejszej instrukcji jest firma Velleman nv. Wszelkie prawa są zastrzeżone na całym świecie.** Żadna część niniejszej instrukcji nie może być kopiowana, powielana, tłumaczona ani przenoszona na jakikolwiek nośnik elektroniczny (lub w inny sposób) bez wcześniejszej pisemnej zgody właściciela praw autorskich.

# MANUAL DO UTILIZADOR

## 1. Introdução

### Aos cidadãos da União Europeia

#### Importantes informações sobre o meio ambiente no que respeita a este produto



Este símbolo no aparelho ou na embalagem indica que, enquanto desperdícios, poderão causar danos no meio ambiente. Não coloque a unidade (ou as pilhas) no depósito de lixo municipal; deve dirigir-se a uma empresa especializada em reciclagem. Devolva o aparelho ao seu distribuidor ou ao posto de reciclagem local. Respeite a legislação local relativa ao meio ambiente.

**Em caso de dúvidas, contacte com as autoridades locais para os resíduos.**

Obrigada por escolher a Velleman®! Leia atentamente as instruções do manual antes de usar o aparelho. Caso o aparelho tenha sofrido algum dano durante o transporte não o instale e entre em contacto com o seu distribuidor.

## 2. Instruções de segurança



Este aparelho pode ser usado por crianças com idade igual ou superior a 8 anos e pessoas com capacidades físicas, mentais e sensoriais reduzidas, ou com falta de experiência e conhecimentos acerca do aparelho, desde que sejam supervisionadas e informadas acerca da utilização do aparelho e possíveis acidentes. As crianças não devem brincar com o aparelho. A limpeza e manutenção não devem ser feitas por crianças sem a devida supervisão.



Usar apenas em interiores.

Proteger o aparelho contra a chuva e humidade ou qualquer tipo de salpicos ou gotas.

## 3. Normas gerais



- Consulte a Garantia de Serviço e Qualidade Velleman® na parte final deste manual do utilizador.
- Familiarize-se com o funcionamento do aparelho antes de o utilizar.
- Por razões de segurança, estão proibidas quaisquer modificações do aparelho desde que não autorizadas. Os danos causados por modificações não autorizadas do aparelho não estão cobertos pela garantia.
- Utilize o aparelho apenas para as aplicações descritas neste manual. Uma utilização incorreta anula a garantia completamente.
- Danos causados pelo não cumprimento das normas de segurança referidas neste manual anulam a garantia e o seu distribuidor não será responsável por quaisquer danos ou outros problemas daí resultantes.
- Nem Velleman NV nem os seus distribuidores podem ser responsabilizados por quaisquer danos (extraordinário, incidental ou indireto) - de qualquer natureza decorrentes (financeira, física...) a partir da posse, uso ou falha do produto.
- Devido às contantes melhorias do produto, o produto actual pode diferir das imagens apresentadas.
- As imagens do produto têm apenas uma função ilustrativa.
- Não ligue o aparelho depois de exposto a variações de temperatura. Para evitar danos, espere que o aparelho atinge a temperatura ambiente.
- Guarde este manual para posterior consulta.

## 4. Descrição

Este kit de iniciação é um kit educacional baseado no micro:bit. Inclui componentes eletrônicos básicos, uma placa de ensaio, fios de ligação e um micro:bit.

## 5. Exemplos

### 5.1 LED

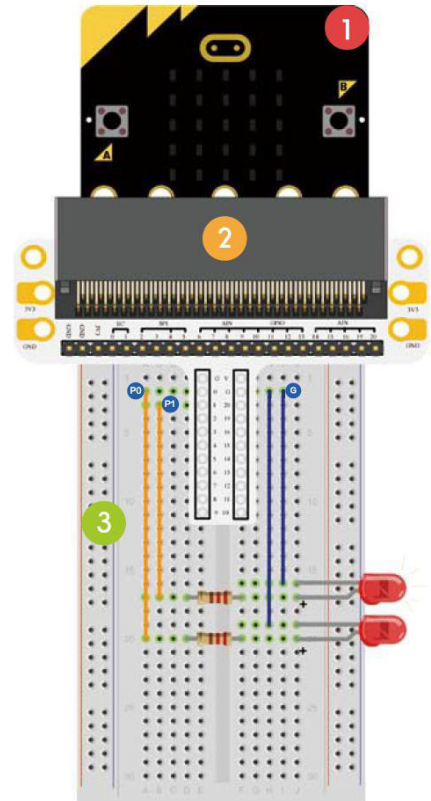
- 1 1x placa micro:bit
- 2 1x adaptador de placa de ensaio micro:bit
- 3 1x placa de ensaio
- 4 2x LED vermelho (polarização: ânodo (+) = perna comprida, cátodo (-) = perna curta)
- 5 2x resistor 100  $\Omega$  (castanho/preto/castanho/dourado)



4



5



Neste curso vamos utilizar o micro: bit para fazer dois LEDs brilharem alternadamente.

Disponha os componentes necessários na placa de ensaio, conforme indicado.

O programa é escrito em blocos de códigos, online no navegador da web. Acesse ao website [www.makecode.com](http://www.makecode.com) ou <https://www.microsoft.com/en-us/makecode?rtc=1> clique no ícone micro:bit e depois clique em **Start Project**.

Microsoft Office Windows Surface Xbox Deals Support More

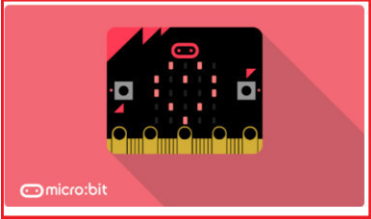
MakeCode About Get started Resources

Search Microsoft.com Sign in

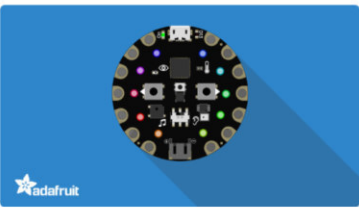
### Hands on computing education

Microsoft MakeCode brings computer science to life for all students with fun projects, immediate results, and both block and text editors for learners at different levels.


*Klik dit icoon aan*



micro:bit  
[Start coding with micro:bit >](#)



Circuit Playground Express  
[Start coding with Circuit Playground Express >](#)



Minecraft  
[Start coding with Minecraft >](#)

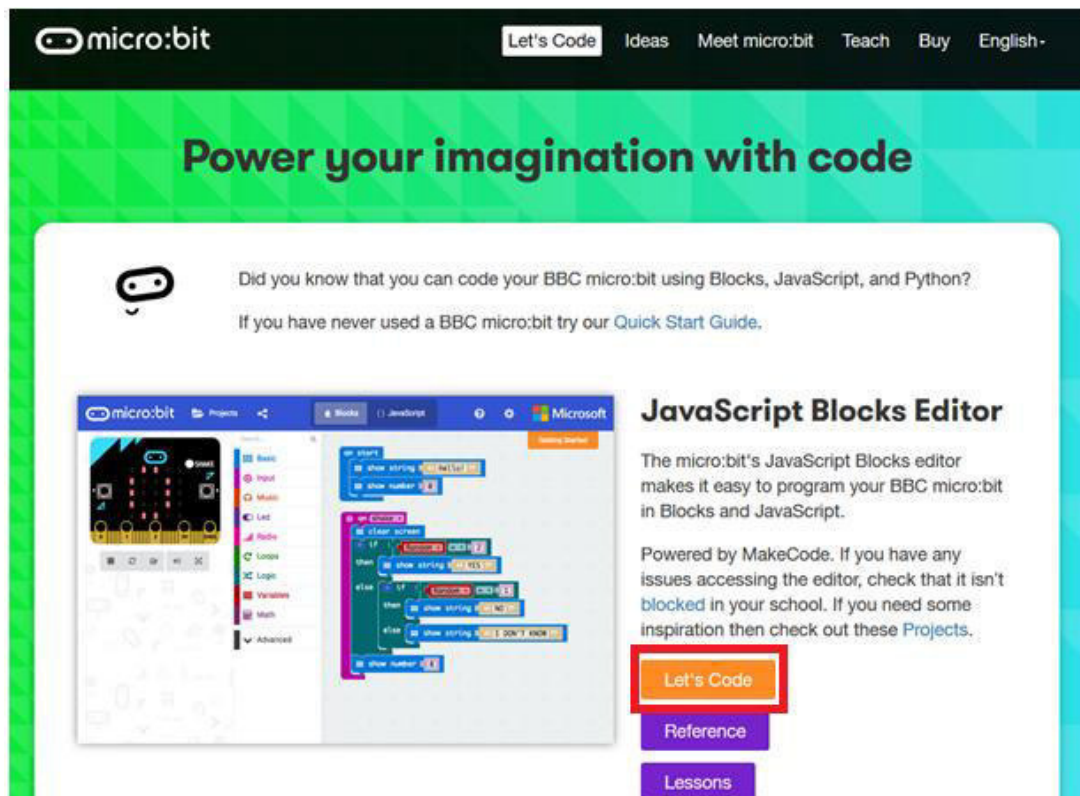
O bloco de código micro:bit é aberto numa nova janela. Agora, pode começar a compor o código com os blocos, que são arrastados desde o desenhador de código para o editor de código.

Leia mais abaixo como funciona...

## O que é o MakeCode?

Formerly PXT – Programming eXperience Toolkit Editor para escrever o código para o micro:bit.

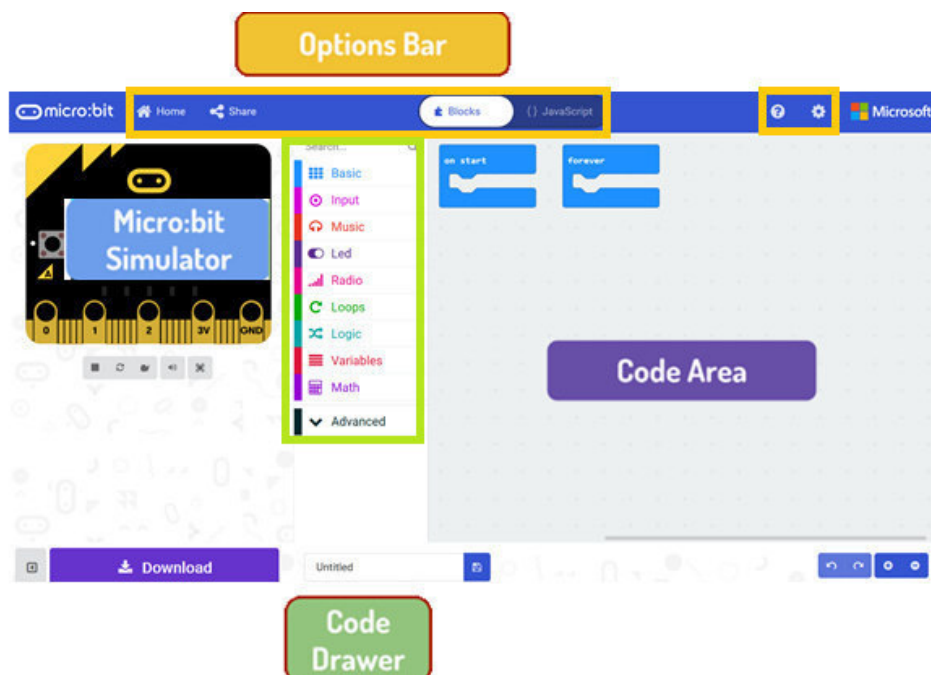
Um editor de códigos de arrastar-e-soltar, gráfico e indicado para principiantes, semelhante ao Scratch. Funciona online, no navegador.



Pode verificar que o editor é composto de diferentes seções. Você cria o seu código na **Code Area**, arrastando os blocos a partir do **Code Drawer**. Pode ver o resultado de imediato no **micro:bit simulator**. Na parte inferior, pode fazer o download e salvar o seu projeto.

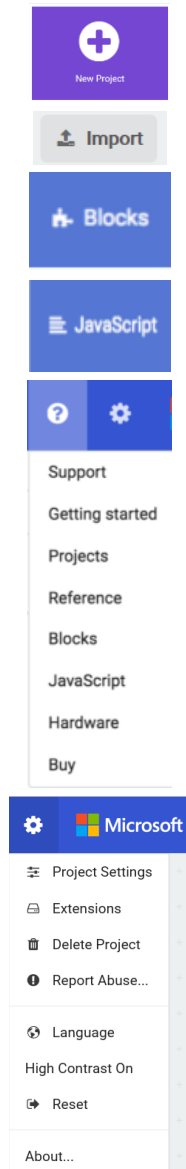


Em seguida, vamos ver mais em pormenor como pode usar a **Options Bar** para fazer outras coisas.





## Barra de Opções



Clique para criar ou adicionar um novo projeto.

Importe os seus projetos aqui.

Abra o interface Blocos para escrever os seus scripts no editor de bloco.

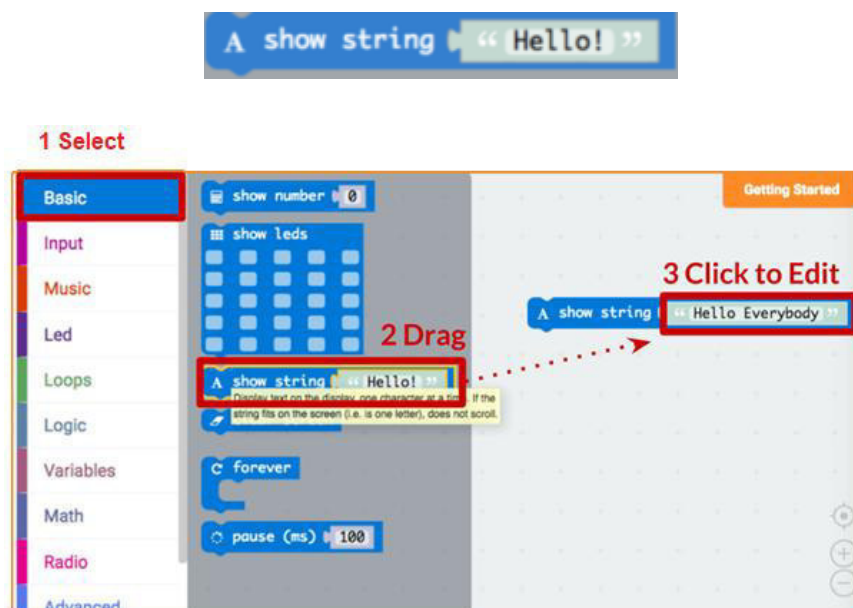
Abra a interface Javascript para digitar seu script em linguagem Javascript.

Muito útil sempre que não tem a certeza sobre como os vários blocos e funções funcionam.

Atalhos para alterar as propriedades do projeto. Altere o nome e apague os seus projetos aqui. **Reset** apaga todos os projetos que tenha guardado, por isso seja muito cuidadoso. Muitas vezes, apenas procura a opção **Delete Project**.

## Usar a Barra LED

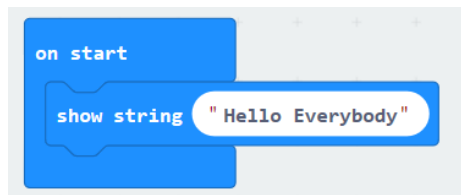
Começamos por escrever um código! Sempre que seleciona a partir do Desenhador de Código, serão apresentados todos os códigos disponíveis. Arraste o bloco **show string** e clique dentro da caixa para editá-lo.



1 Selecionar

2 Arrastar

3 Clicar para editar

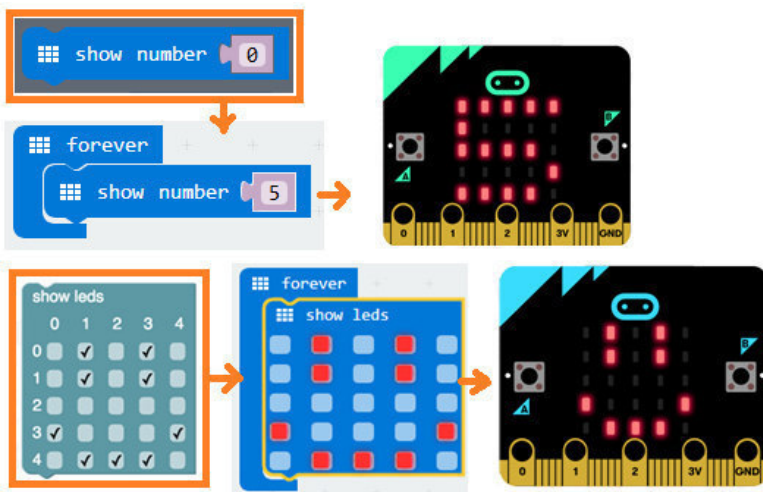


Veja o que acontece no simulador micro:bit!

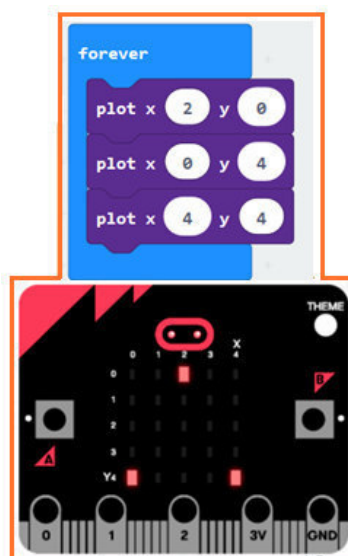
### Diga Olá!

O texto que acabou de apresentar é chamado de **strings**.

Pode exibir um número inteiro no ecrã LED usando **show number** ou qualquer imagem de 5x5 pixels usando **show leds**.



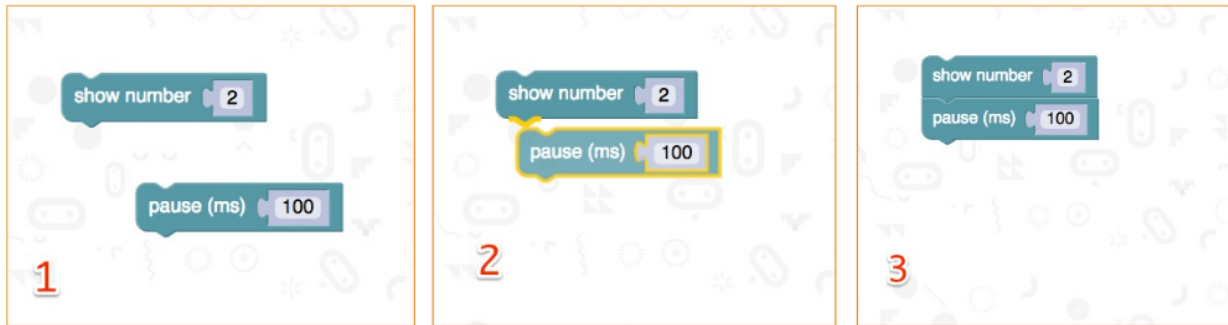
Também pode traçar um LED de ponto de cada vez usando coordenadas. Coordenar (0,0) é o canto superior esquerdo.



### Juntar Blocos

Clique e segure o bloco que gostaria de juntar. Arraste o bloco para o bloco alvo até que um dos lados do bloco alvo fique iluminado. Liberte e agora os dois blocos já estão juntos!

Clicar no primeiro bloco irá mover o segundo; clicar no segundo bloco irá separá-lo do primeiro.



### Testar no micro:bit

Ligue o micro:bit ao seu computador usando o cabo USB.

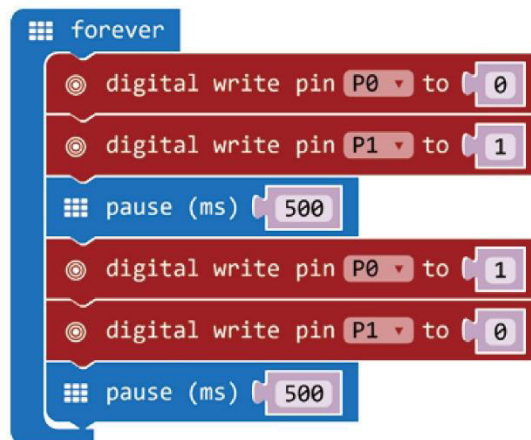
Este passará por uma sequência padrão, pedindo-lhe que pressione os botões e jogue um jogo. Experimente!

Clique em **Download** para traduzir seu script para um **hex file** e para o descarregar.

Se configurar o Google Chrome corretamente, não precisará desta próxima etapa. Arraste o arquivo hexadecimal descarregado para a drive do micro:bit ou clique com o botão direito do rato e **Send To** no Windows®.

Acabou de programar o seu próprio dispositivo!

Dê uma olhadela no código abaixo.



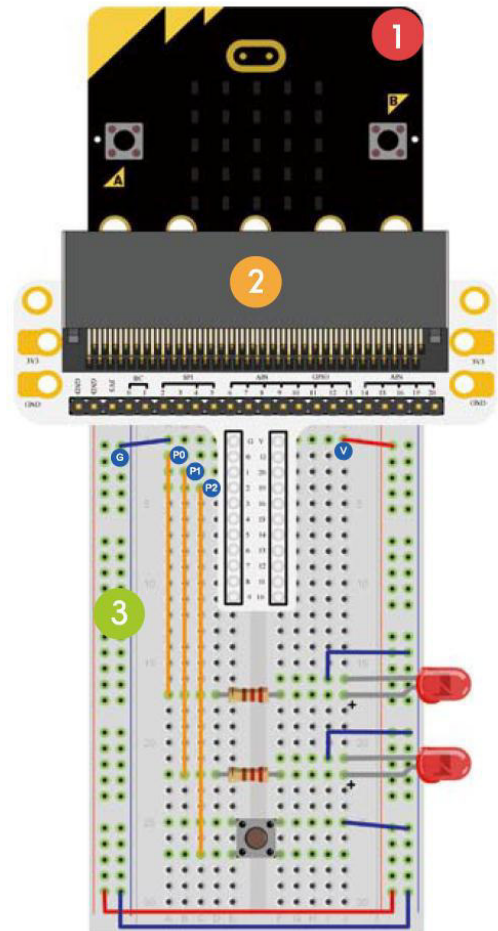
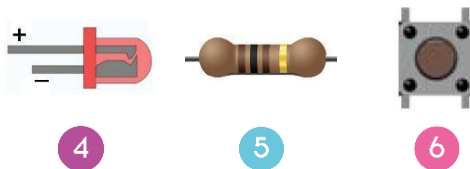
1.	Arraste e largue os blocos de código para formar o código conforme indicado. Pode encontrar o bloco de <b>digital write pin P0 a 0</b> em <b>Advanced</b> → <b>Pins</b> no Code Drawer. Pode encontrar <b>forever</b> e <b>pause (ms) 100</b> em <b>Basic</b> no Code Drawer.
2.	Defina o valor de P0 para 0. LED0 desligado = baixa tensão = 0 V = digital 0. Defina o valor de P1 para 1. LED1 ligado = alta tensão = 5 V = digital 1.
3.	Defina <b>pause</b> para 500 ms.
4.	Defina o valor de P0 para 1. LED0 ligado = alta tensão = 5 V = digital 1. Defina o valor de P1 para 0. LED1 desligado = baixa tensão = 0 V = digital 0.
5.	Defina <b>pause</b> para 500 ms.
6.	Quando terminado, compilamos o programa e geramos o ficheiro .hex. Clique no botão de download e guarde o ficheiro .hex na pasta Downloads → C:\downloads Este ficheiro .hex está pronto a ser carregado para o micro:bit.

Ligue o micro:bit a uma porta USB. Em seguida, arraste e largue o ficheiro .hex no dispositivo amovível do micro: bit para carregar o programa.

Verá os dois LEDs piscarem alternadamente. E agora, por que não fazer um semáforo RGB?

## 5.2 Botão

- 1 1x placa micro:bit
- 2 1x adaptador de placa de ensaio micro:bit
- 3 1x placa de ensaio
- 4 2x LED vermelho (polarização: ânodo (+) = perna comprida, cátodo (-) = perna curta)
- 5 2x resistor de 100  $\Omega$  (castanho/preto/castanho/dourado)
- 6 1x botão de pressão momentânea



Usamos um botão para controlar o piscar do LED. Pressione o botão para fazer o LED piscar por turnos. Liberte o botão para desligar o LED.

Disponha os componentes necessários na placa de ensaio, conforme indicado.

Dê uma olhadela no código abaixo.

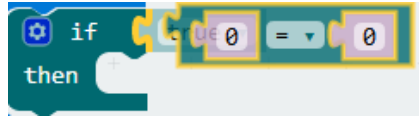
```

on start
  set pull pin P2 to up

forever
  if (digital read pin P2 = 0)
  then
    digital write pin P0 to 0
    digital write pin P1 to 1
    pause (ms) 500
    digital write pin P0 to 1
    digital write pin P1 to 0
    pause (ms) 500
  
```

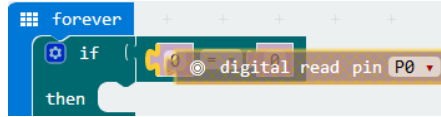
1. Arraste e largue os blocos de código para formar o código conforme indicado. Pode encontrar os blocos do **digital read pin P0** e do **digital write pin P0 a 0** em **Advanced** → **Pins** no Code Drawer. Pode encontrar **forever** e **pause (ms) 100** em **Basic** no Code Drawer. Você pode encontrar os blocos **if then** e **logic equal function** em **Logic** no Code Drawer.

Arraste o bloco **logic equal function** e solte-o em cima do bloco **true**.



Os dois blocos encaixam.

Arraste o bloco **digital read pin P0** e solte-o sobre o **0** do bloco **logic equal function** conforme indicado.



2.	Configure P2 para ser um "pull-up".
3.	Configure o <b>digital read pin</b> em P2.
4.	Configure o <b>digital write pin P0 para 0</b> (ativo baixo (0 V)). Configure o <b>digital write pin P1 para 1</b> (ativo alto (5 V)).
5.	Defina <b>pause</b> para 500 ms.
6.	Configure o <b>digital write pin P0 para 1</b> (ativo alto (5 V)). Configure o <b>digital write pin P1 para 0</b> (ativo baixo (0 V)).
7.	Defina <b>pause</b> para 500 ms.
8.	Quando terminado, compilamos o programa e geramos o ficheiro .hex. Clique no botão de download e guarde o ficheiro .hex na pasta Downloads → C:\downloads Este ficheiro .hex está pronto a ser carregado para o micro:bit.  Ligue o micro:bit a uma porta USB. Em seguida, arraste e largue o ficheiro .hex no dispositivo amovível do micro: bit para carregar o programa.

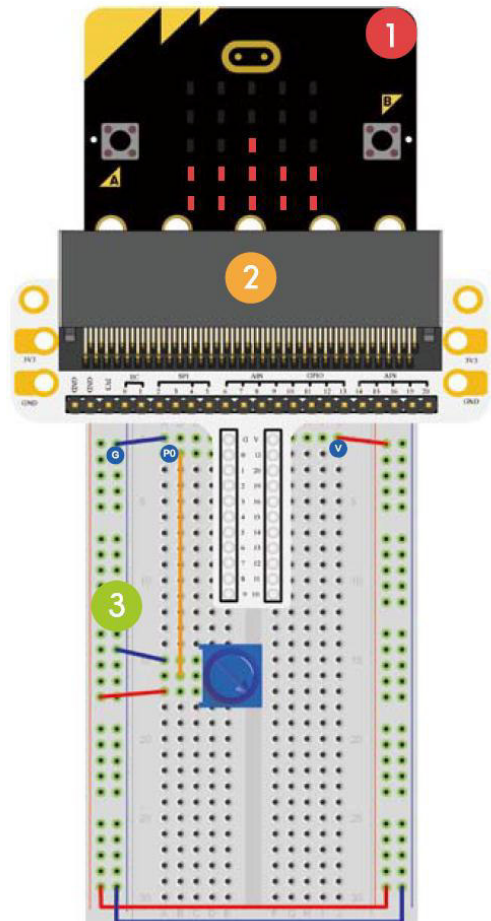
Pressione o botão e verá o LED piscar alternadamente. Agora, como acender o LED vermelho com o botão pressionado e acender o LED verde com o botão liberto?

### 5.3 Trimpot

- 1 1x placa micro:bit
- 2 1x adaptador de placa de ensaio micro:bit
- 3 1x placa de ensaio
- 4 1x trimmer 10 kΩ



4

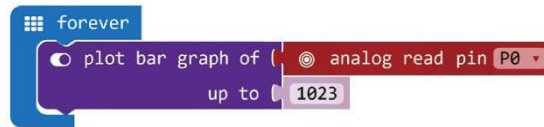


Vamos ler a tensão de saída do trimpot e apresentá-la no ecrã do micro:bit com um gráfico de barras.

Disponha os componentes necessários na placa de ensaio, conforme indicado.



Dê uma olhadela no código abaixo.



1.	Arraste e largue os blocos de código para formar o código conforme indicado. Pode encontrar o bloco do <b>plot bar graph of 0 up to 0</b> em <b>LED</b> no Code Drawer.
2.	Ajuste o <b>analog read pin</b> para P0. Pode encontrar este bloco em <b>Pins</b> no Code Drawer. Defina o valor para 1023.
3.	Quando terminado, compilamos o programa e geramos o ficheiro .hex. Clique no botão de download e guarde o ficheiro .hex na pasta Downloads → C:\downloads Este ficheiro .hex está pronto a ser carregado para o micro:bit. Ligue o micro:bit a uma porta USB. Em seguida, arraste e largue o ficheiro .hex no dispositivo amovível do micro: bit para carregar o programa.

Gire o trimmer. A tensão será apresentada no ecrã do micro:bit através de um gráfico de barras. Quando a tensão é 0, o ecrã LED exibe apenas um ponto de pixel. Quando é de 3.3 V, todo o ecrã ficará iluminado. E agora, como usaria o trimmer para ajustar o brilho do LED?

## 5.4 Fotocélula

- 1 1x placa micro:bit
- 2 1x adaptador de placa de ensaio micro:bit
- 3 1x placa de ensaio
- 4 1x fotocélula
- 5 1x resistor 10 kΩ (castanho/preto/preto/vermelho/castanho)

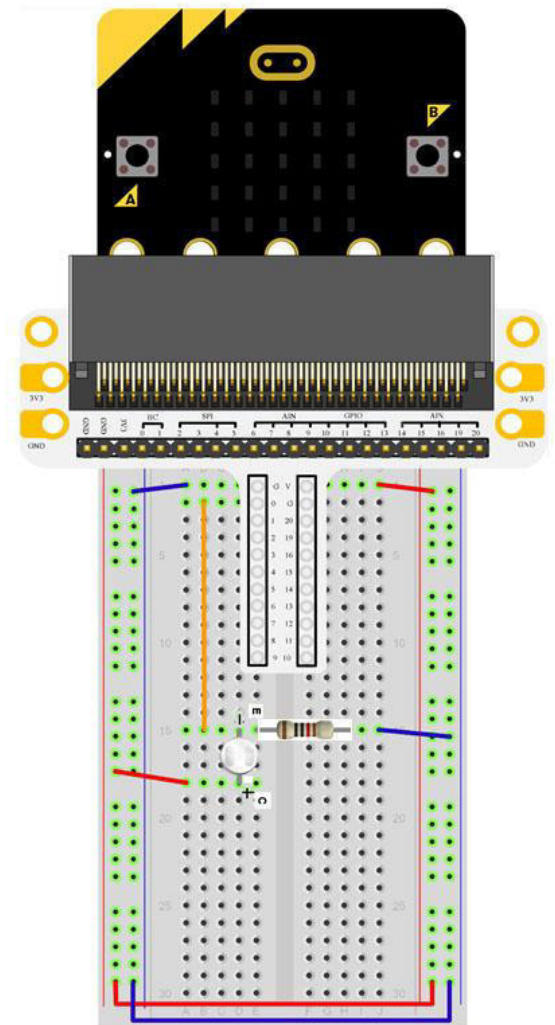


4

Perna curta = coletor  
Ânodo positivo  
Perna longa = emissor  
Cátodo negativo



5

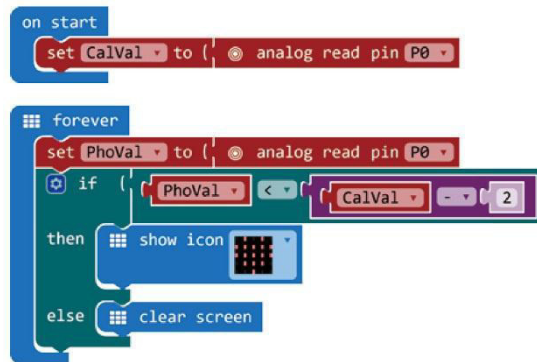


Vamo usar uma fotocélula para controlar o brilho do ecrã do micro:bit.

Disponha os componentes necessários na placa de ensaio, conforme indicado.



Dê uma olhadela no código abaixo.



1. Primeiro, criamos duas variáveis. Vá a **Variables** no Code Drawer e clique em **Make a Variable**.

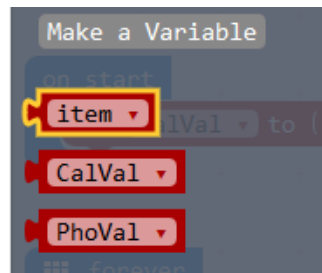
New variable name:

CalVal

Ok ✓ Cancel ✕

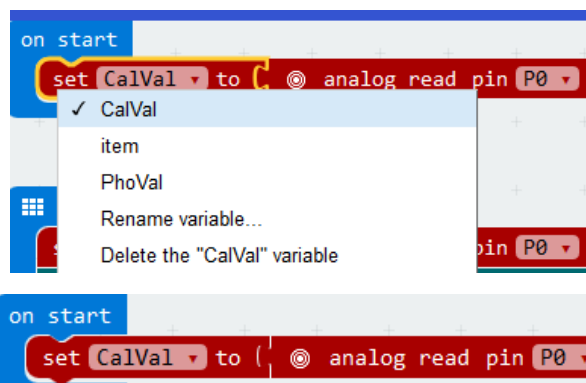
Entre em **CalVal** na janela e clique em **Ok**. Introduza **PhoVal** na janela e clique em **Ok**.

Verá as duas novas variáveis por baixo de **Variables** no Code Drawer.



Vamos precisar destas variáveis mais tarde para salvar os dados num registo.

Pode encontrar os blocos **logic smaller than function if then else** em **Logic** no Code Drawer. Pode encontrar o bloco da **mathematical function minus** em **Math** no Code Drawer. Pode encontrar o bloco **show icon** em **Basic** no Code Drawer. Pode encontrar o bloco **clear screen** em **Basic** → **More** no Code Drawer. Pode encontrar o bloco **set item to 0** em **Variables** no Code Drawer. Clique na seta e selecione **CalVal** ou **PhoVal**.



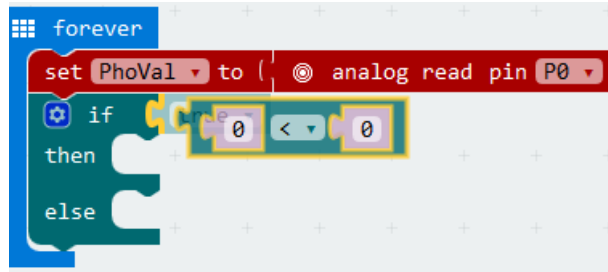
2. Selecione a variável **CalVal** e configure o **analog read pin** para P0.



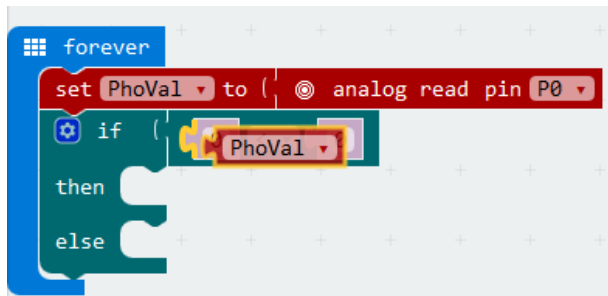
3. No bloco **forever**, selecione a variável **PhoVal** e configure o **analog read pin** para P0.



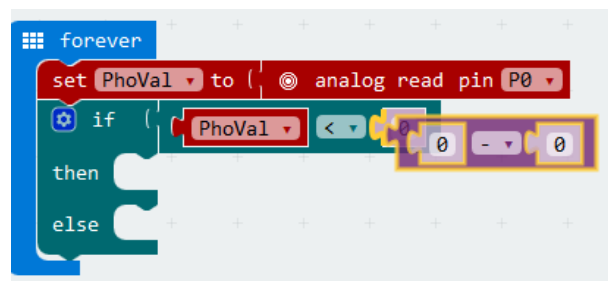
4. Arraste a **logic smaller than function** para junto do bloco **if** e largue-o em cima do bloco **true**.



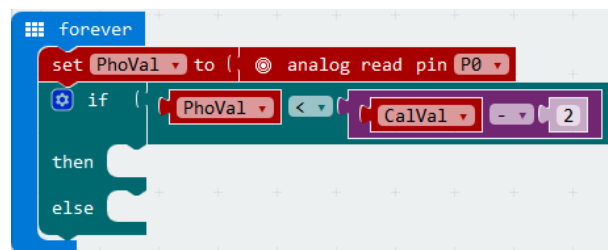
Em seguida, arrastamos e largamos a variável **PhoVal** (a partir de **Variables** no Code Drawer) e largamos em cima do primeiro 0 da **logic smaller than function**.



Arraste o bloco **mathematical function minus** e largue-a sobre o segundo 0 de **logic smaller than function**.



Em seguida, arrastamos e largamos a variável **CalVal** (a partir de **Variables** no Code Drawer) e largamos em cima do primeiro 0 do bloco **mathematical function minus**. Configure o segundo 0 do bloco de **mathematical function minus** para 2.



5. Arraste e solte o bloco **show icon** para junto do bloco **then**. Arraste e solte o bloco **clear screen** para junto do bloco **else**.
6. Sempre que **PhoVal** é inferior a **CalVal** - 2, o ecrã exibe um coração. Caso contrário, desligar-se-á.

- 7.** Quando terminado, compilamos o programa e geramos o ficheiro .hex. Clique no botão de download e guarde o ficheiro .hex na pasta Downloads → C:\downloads Este ficheiro .hex está pronto a ser carregado para o micro:bit.
- Ligue o micro:bit a uma porta USB. Em seguida, arraste e largue o ficheiro .hex no dispositivo amovível do micro: bit para carregar o programa.

Observação: redefinir o micro:bit para calibrar o valor de referência de acordo com o brilho atual. Para executar o programa devidamente, temos de iniciar com a luz ligada.

Quando a luz está ligada não aparece nada. Quando a luz está desligada, aparece o coração. Agora, como podemos usar a fotocélula para controlar um LED?

## 5.5 LED RGB

- 1 1x placa micro:bit
- 2 1x adaptador de placa de ensaio micro:bit
- 3 1x placa de ensaio
- 4 1x LED RGB (cátodo comum)
- 5 3x resistor 10 Ω (castanho/preto/castanho/dourado)

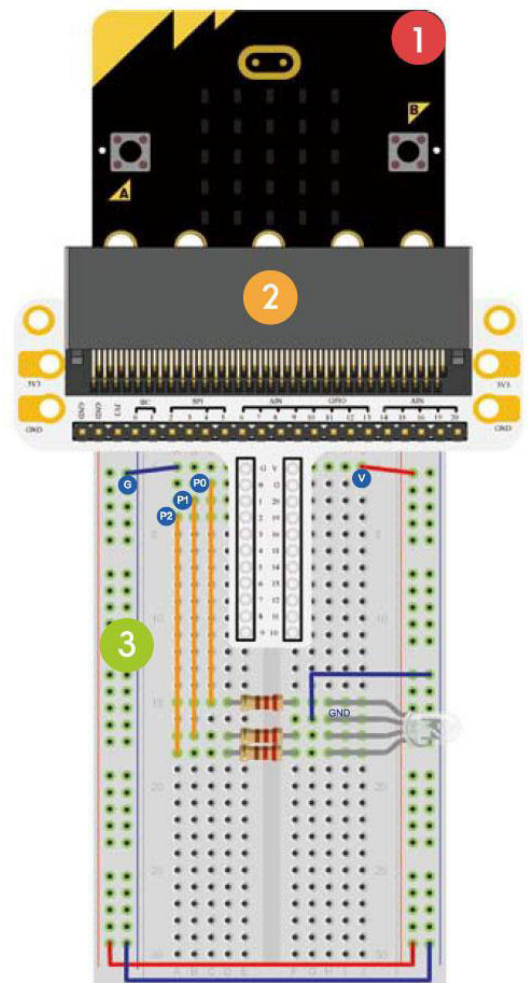


4

1. Vermelho (ânodo +)
2. Terra (cátodo -) – perna mais longa
3. Verde (ânodo +)
4. Azul (ânodo +)



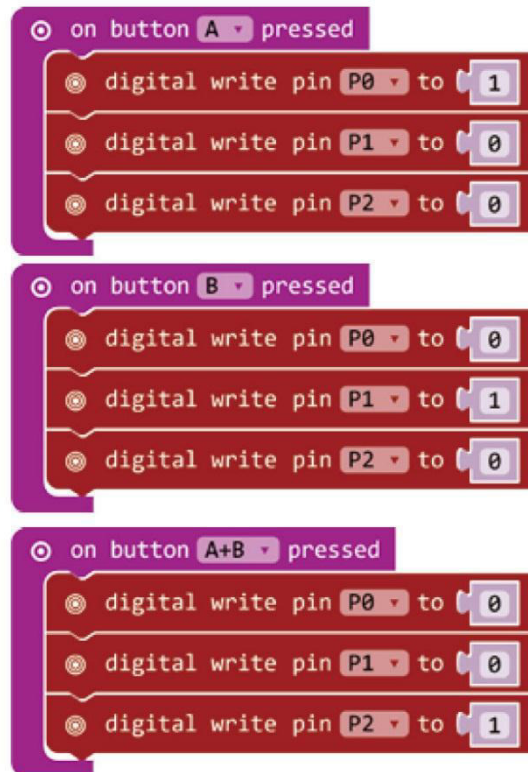
5



Vamos fazer um LED RGB mudar a luz gradualmente entre o vermelho, o verde e o azul.

Disponha os componentes necessários na placa de ensaio, conforme indicado.

Dê uma olhadela no código abaixo.



- Arraste e solte os blocos de código para formar o código, conforme indicado.

Pode encontrar o bloco **on button A pressed** em **Input** no Code Drawer. Pode encontrar o bloco **digital write pin P0 to 0** em **Pins** no Code Drawer.

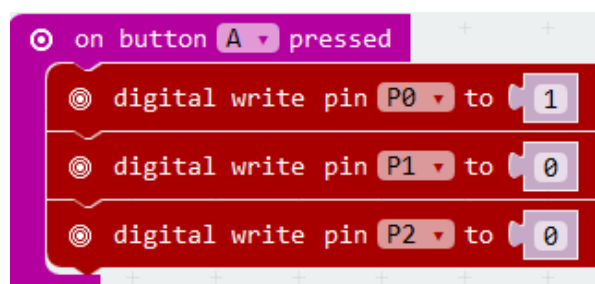
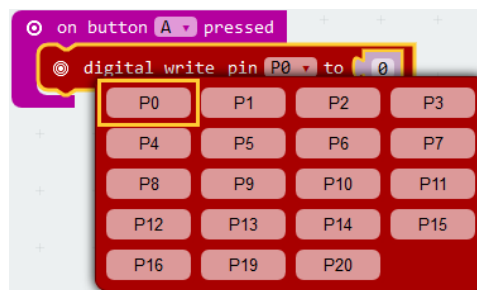
Selecione a opção A no bloco do **botão A pressionado**.

Arraste 3 blocos **digital write pin P0 to 0** e insira-os no bloco **on button A pressed**.

Coloque o pino P0 no primeiro bloco **digital write pin P0 to 0**, e coloque o valor 0 em 1 (LED vermelho ligado).

Coloque o pino P1 no segundo bloco **digital write pin P0 to 0**, e coloque o valor em 0 (LED verde desligado).

Coloque o pino P2 no terceiro bloco **digital write pin P0 to 0**, e coloque o valor em 0 (LED azul desligado).

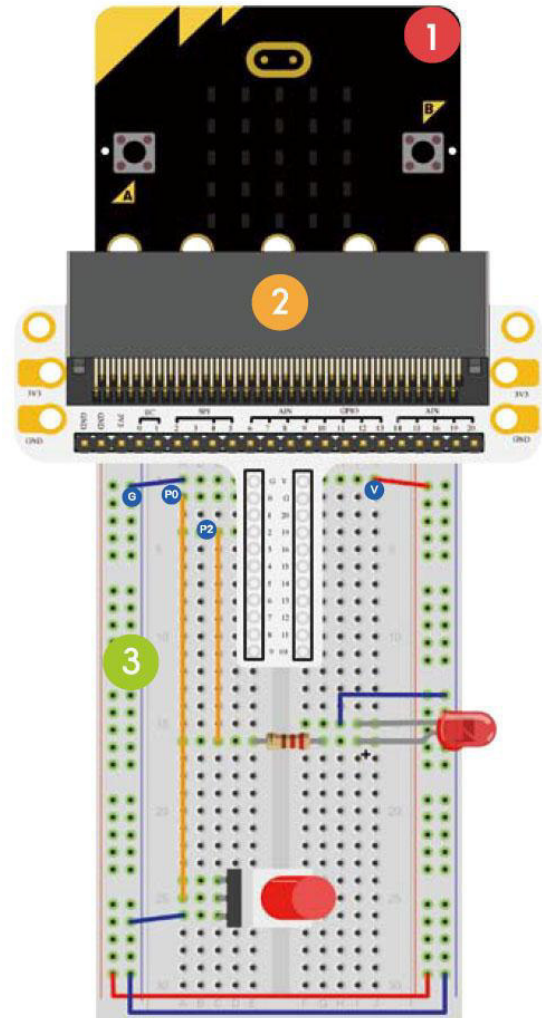
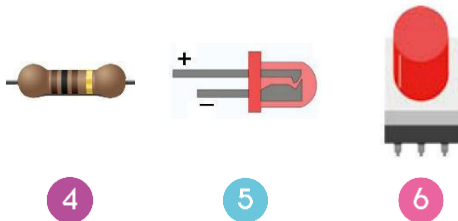


2.	De modo semelhante, compile os dois blocos <b>on button B/A+B pressed</b> para formar o código completo.
3.	Quando terminado, compilamos o programa e geramos o ficheiro .hex. Clique no botão de download e guarde o ficheiro .hex na pasta Downloads → C:\downloads Este ficheiro .hex está pronto a ser carregado para o micro:bit. Ligue o micro:bit a uma porta USB. Em seguida, arraste e largue o ficheiro .hex no dispositivo amovível do micro: bit para carregar o programa.

Pressione o botão A para acender o LED vermelho, pressione o botão B para acender o LED verde, pressione os botões A e B em simultâneo para acender o LED azul. Agora, como criaria um luz RGB de gradiente suave?

## 5.6 Interruptor de Travamento Automático

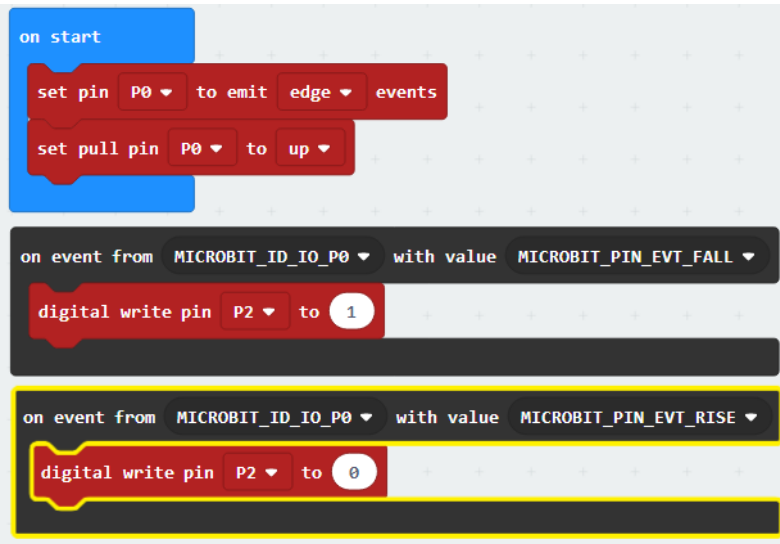
- 1 1x placa micro:bit
- 2 1x adaptador placa de ensaio micro:bit
- 3 1x placa de ensaio
- 4 1x resistor de 100  $\Omega$  (castanho/preto/castanho/dourado)
- 5 1x LED vermelho (polarização: ânodo (+) = perna longa, cátodo (-) = perna curta)
- 6 1x interruptor com bloqueio automático ou biestável



Vamos ler a temperatura ambiente (dados) a partir do sensor de temperatura analógico e exibir os dados no micro:bit.

Disponha os componentes necessários na placa de ensaio, conforme indicado.

Dê uma olhadela no código abaixo.



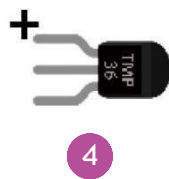
1.	Arraste e largue os bloco de código para formar o código conforme indicado. Pode encontrar os blocos <b>set pin P0 to emit edge events</b> e <b>set pull P0 to up</b> em <b>Pins</b> → <b>More</b> no Code Drawer. Pode encontrar os blocos <b>on event from MICROBIT...EVT_FALL/RISE</b> em <b>Control</b> no Code Drawer.
2.	Defina o tipo de eventos na <b>borda</b> . Defina o pino de puxar para <b>P0</b> e para <b>cima</b> .
3.	Agora, crie um bloco de evento. Arraste e largue o bloco <b>on event from MICROBIT...EVT°FALL</b> para baixo do bloco <b>on start</b> . Clique na primeira seta e selecione a opção <b>MICROBIT_ID_IO_P0</b> a partir do menu. Clique na segunda seta e selecione a opção <b>MICROBIT_PIN_EVENT_FALL</b> a partir do menu. Arraste e largue o bloco <b>digital write pin P0 to 0</b> no bloco de evento. Defina P0 para P2 e o 0 para 1. Pode encontrar o bloco <b>digital write pin P0 to 0</b> em <b>Pins</b> no Code Drawer.
4.	Faça o mesmo para o segundo bloco de eventos. Em vez disso, clique na segunda seta e selecione a opção <b>MICROBIT_PIN_EVENT_RISE</b> a partir do menu. Do mesmo modo, defina o pino P0 para P2 do bloco <b>digital write pin P0 to 0</b> e defina o valor para 0.
5.	Quando terminado, compilamos o programa e geramos o ficheiro .hex. Clique no botão de download e guarde o ficheiro .hex na pasta Downloads → C:\downloads Este ficheiro .hex está pronto a ser carregado para o micro:bit. Ligue o micro:bit a uma porta USB. Em seguida, arraste e largue o ficheiro .hex no dispositivo amovível do micro: bit para carregar o programa.

Pressione o interruptor de travamento automático e o LED acenderá. Pressione novamente para desligar o LED. E agora, como controlaria o ecrã do micro:bit com esse interruptor de travamento automático?



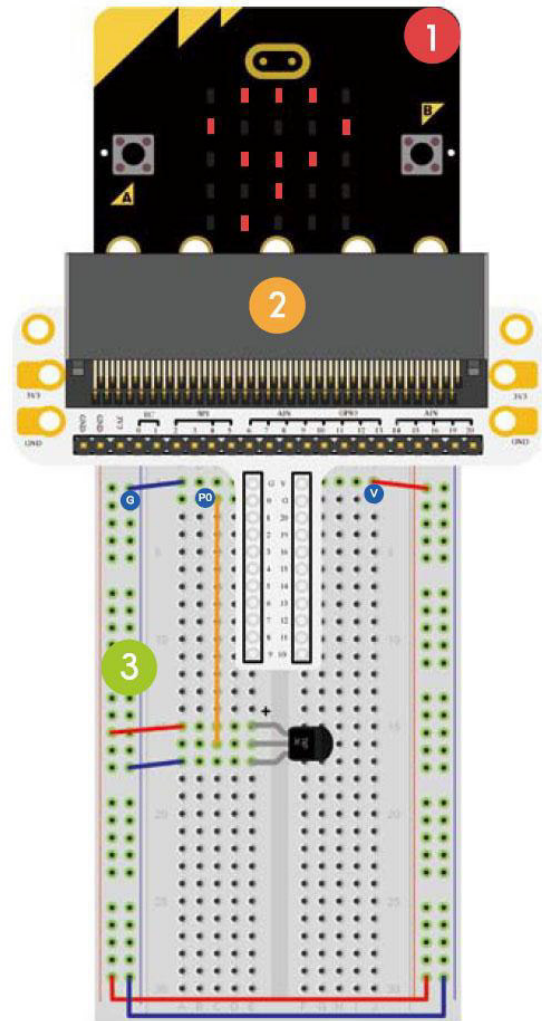
## 5.7 Sensor de Temperatura

- 1 1x placa micro:bit
- 2 1x adaptador de placa de ensaio micro:bit
- 3 1x placa de ensaio
- 4 1x sensor de temperatura TMP26

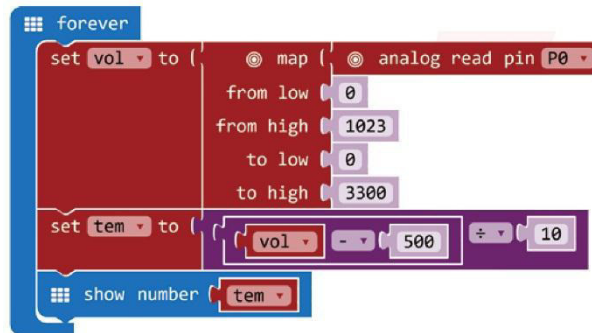


Vamos ler a temperatura ambiente (dados) do sensor de temperatura analógico e exibir os dados no micro:bit.

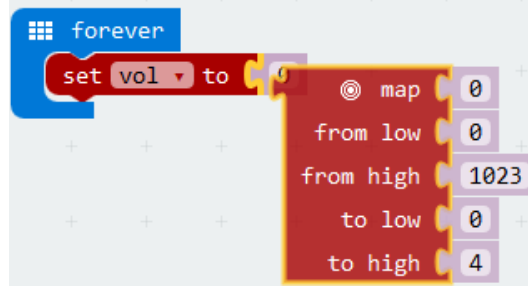
Disponha os componentes necessários na placa de ensaio, conforme indicado.



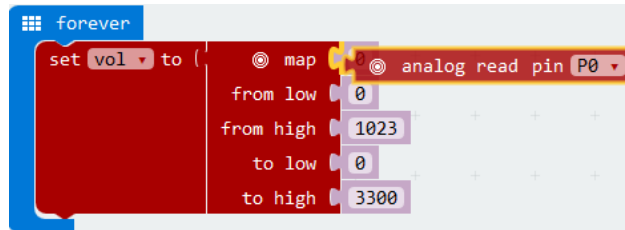
Dê uma olhadela no código abaixo.



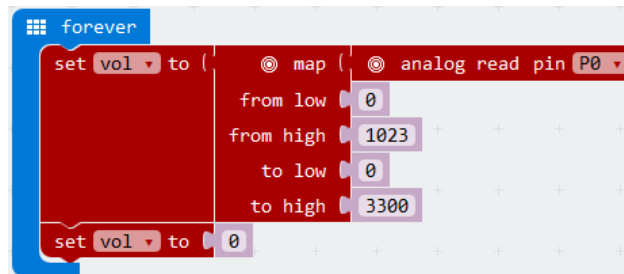
<p><b>1.</b></p>	<p>Arraste e solte os blocos de código para formar o código, conforme indicado.</p> <p>Pode encontrar o bloco <b>set item to</b> em <b>Variables</b> no Code Drawer. Pode encontrar os blocos <b>map/from low/from high/to low/to high</b> e <b>analog read pin</b> em <b>Pins</b> no Code Drawer. Pode encontrar <b>mathematical function minus</b> e <b>divide</b> em <b>Math</b> no Code Drawer. Pode encontrar o bloco <b>show number</b> em <b>Basic</b> no Code Drawer.</p>
<p><b>2.</b></p>	<p>Primeiro, criamos duas variáveis. Vá a <b>Variáveis</b> no Desenhador de Código e clique em <b>Criar Variável</b>.</p> <p>New variable name:</p> <div data-bbox="496 927 1230 1077" style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <input style="width: 400px;" type="text" value="CalVa"/> <div style="display: flex; justify-content: flex-end; gap: 10px; margin-top: 5px;"> <span style="background-color: #28a745; color: white; padding: 2px 10px; border-radius: 3px;">Ok ✓</span> <span style="background-color: #6c757d; color: white; padding: 2px 10px; border-radius: 3px;">Cancel ✕</span> </div> </div> <p>Insira <b>vol</b> na janela e clique em <b>Ok</b>. Insira <b>tem</b> na janela e clique em <b>Ok</b>. Poderá ver as duas novas variáveis em <b>Variables</b> no Code Drawer.</p> <div data-bbox="711 1173 1015 1469" style="border: 1px solid #ccc; padding: 5px; margin: 10px 0; background-color: #343a40; color: white;"> <p style="text-align: center; margin: 0;">Make a Variable</p> <div style="margin-top: 10px;"> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px; display: flex; align-items: center;"> <span style="font-size: 0.8em; margin-right: 5px;">item</span> <span style="font-size: 0.8em;">▼</span> </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px; display: flex; align-items: center;"> <span style="font-size: 0.8em; margin-right: 5px;">tem</span> <span style="font-size: 0.8em;">▼</span> </div> <div style="border: 1px solid #ccc; padding: 2px; display: flex; align-items: center;"> <span style="font-size: 0.8em; margin-right: 5px;">vol</span> <span style="font-size: 0.8em;">▼</span> </div> </div> </div> <p>Vamos precisar destas variáveis mais tarde para salvar os dados no registo. Arraste e largue o bloco <b>set item to</b> no bloco <b>forever</b> e selecione a opção <b>vol</b> usando a seta.</p> <div data-bbox="671 1615 1058 1760" style="border: 1px solid #ccc; padding: 5px; margin: 10px 0; background-color: #343a40; color: white;"> <div style="display: flex; align-items: center; margin-bottom: 5px;"> <span style="font-size: 0.8em; margin-right: 5px;">forever</span> <span style="font-size: 0.8em;">▼</span> </div> <div style="border: 1px solid #ccc; padding: 2px; display: flex; align-items: center;"> <span style="font-size: 0.8em; margin-right: 5px;">set vol</span> <span style="font-size: 0.8em;">▼</span> <span style="font-size: 0.8em; margin: 0 5px;">to</span> <span style="font-size: 0.8em;">0</span> </div> </div> <p>Em seguida, arrastamos e largamos o bloco <b>map/from low/from high/to low/to high</b> para cima do 0 junto do bloco <b>set item to</b>.</p>



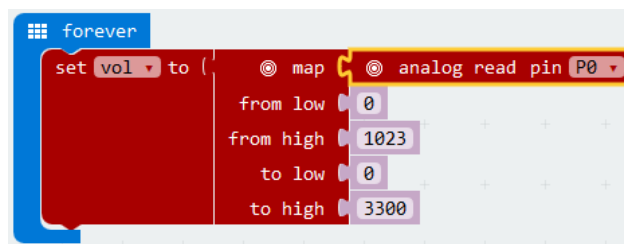
Arraste e largue o bloco **analog read pin** para junto do bloco **map/from low/from high/to low/to high** e defina o valor **to high** 4 para 3300.



Agora, arraste e largue o bloco **set item to** para baixo do primeiro bloco. Altere a variável **item** para **vol** usando a seta.



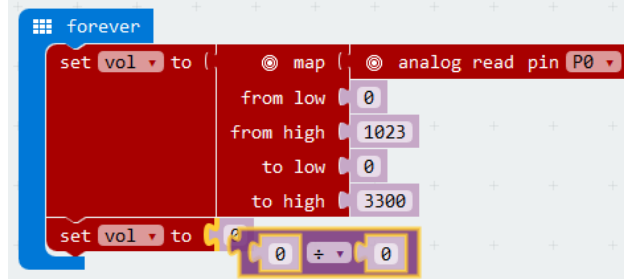
A tensão medida em mV via **analog read pin P0** é um valor de 8-bit de 0-1023 (0-3.3 V) e é mapeado a partir de um valor baixo (0 ou 0 V) para um valor alto (3300 mV ou 3.3 V). A tensão medida (mV) é guardada na variável **vol**.



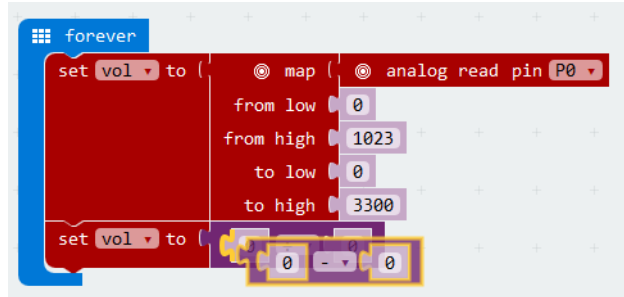
Agora, criamos uma fórmula para converter a tensão medida em temperatura:

$$Temperature (^{\circ}C) = \frac{(Output\ voltage\ (mV) - 500)}{10}$$

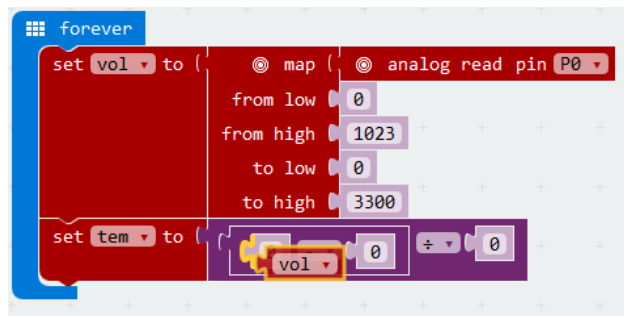
Nesta fórmula, a tensão de saída é a variável **vol** (medida através do sensor TMP36). O resultado da fórmula será guardado na variável **tem**. Vamos lá criar a fórmula usando os blocos. Arraste e largue **mathematical function divide** para cima do valor 0 do bloco **set vol to**.



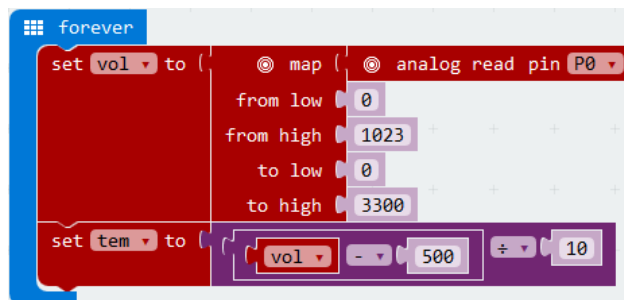
Araste e largue **mathematical function minus** em cima do primeiro 0 de **mathematical function divide**.



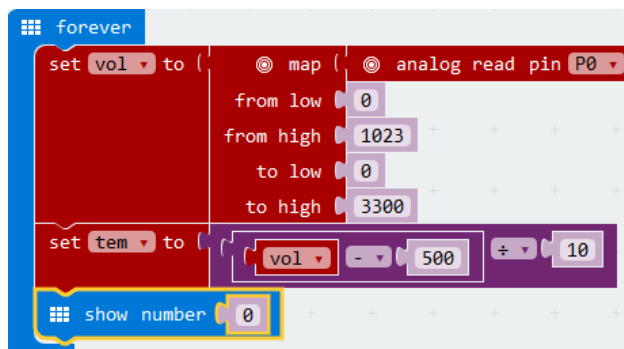
Agora, defina a segunda variável de **vol** para **tem**, arraste e largue a variável **vol** para cima do primeiro 0 de **mathematical function minus**. A variável **vol** pode ser encontrada em **Variables** no Code Drawer.



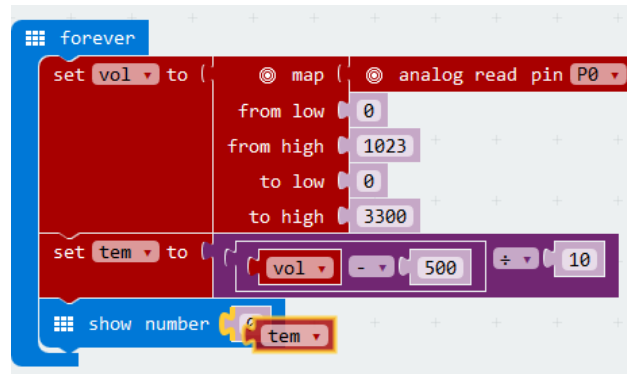
Defina o valor 0 de **mathematical function minus** para 500. Do mesmo modo, defina o valor de **mathematical function divide** para 10



Arraste e largue o bloco **show number** sob o bloco **set tem to**.



Finalmente, arraste e largue a variável **tem** sobre o valor 0 do bloco **show number**.



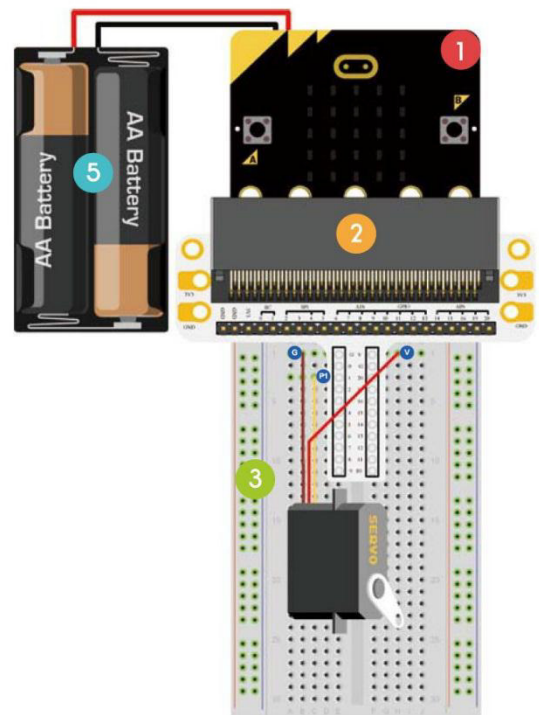
Acabou de criar o código!

- 3.** Quando terminado, compilamos o programa e geramos o ficheiro .hex. Clique no botão de download e guarde o ficheiro .hex na pasta Downloads → C:\downloads Este ficheiro .hex está pronto a ser carregado para o micro:bit.
- Ligue o micro:bit a uma porta USB. Em seguida, arraste e largue o ficheiro .hex no dispositivo amovível do micro: bit para carregar o programa.

Verá as bolinhas LED piscarem alternadamente. Agora, como indicaria a temperatura em graus Fahrenheit?

## 5.8 Servo

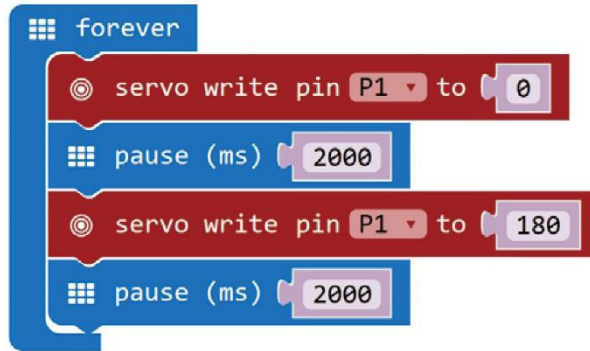
- 1 1x placa micro:bit
- 2 1x adaptador placa de ensaio micro:bit
- 3 1x placa de ensaio
- 4 1x mini servo
- 5 1x suporte com 2 pilhas AA de 1.5 V



Vamos fazer um servo girar continuamente dentro de um alcance de deslocação (0-180 °).

Disponha os componentes necessários na placa de ensaio, conforme indicado.

Dê uma espreitadela ao código abaixo.



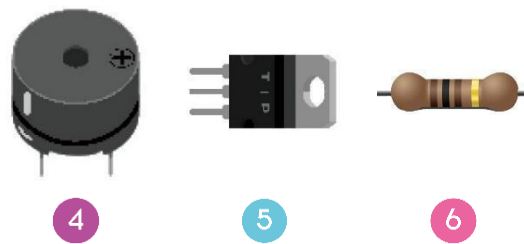
1.	Arraste e solte os blocos de código para formar o código, conforme indicado. Pode encontrar o bloco <b>servo write pin P0 a180</b> em <b>Pins</b> no Code Drawer. Pode encontrar <b>forever</b> e <b>pause (ms) 100</b> em <b>Basic</b> no Code Drawer.
2.	Coloque <b>P0</b> em <b>P1</b> e defina o valor para 0.
3.	Configure <b>pause</b> para 2000 ms.
4.	Coloque <b>P0</b> em <b>P1</b> . Configure também o valor de 0 para 180.
5.	Configure <b>pause</b> para 2000 ms.
6.	Quando terminado, compilamos o programa e geramos o ficheiro .hex. Clique no botão de download e guarde o ficheiro .hex na pasta Downloads → C:\downloads Este ficheiro .hex está pronto a ser carregado para o micro:bit. Ligue o micro:bit a uma porta USB. Em seguida, arraste e largue o ficheiro .hex no dispositivo amovível do micro: bit para carregar o programa.



Podemos ver o servo rodar de 0 a 180 graus. Agora, como faria um termómetro de discagem com sensor de temperatura e servo?

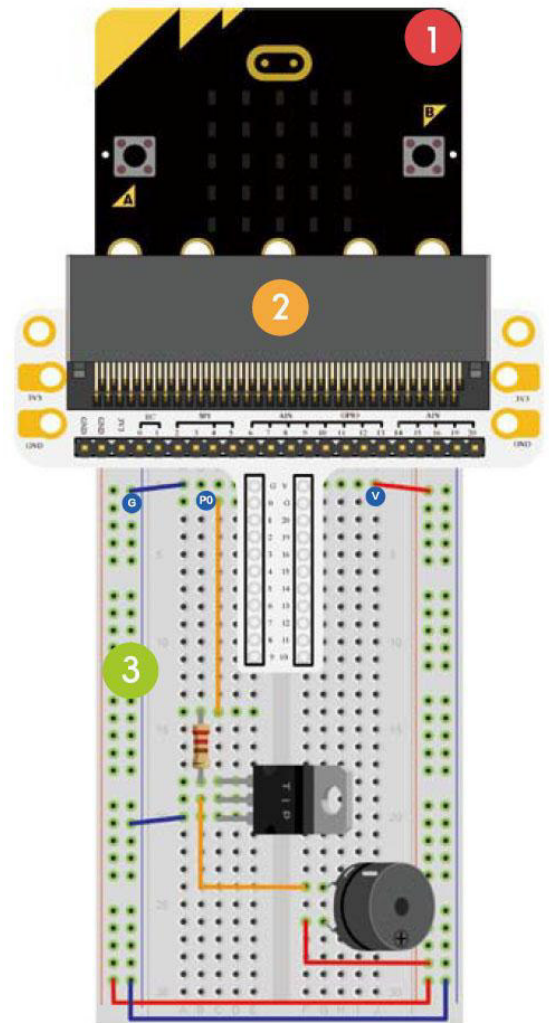
## 5.9 Campainha

- 1 1x placa micro:bit
- 2 1x adaptador de palca de ensaio micro:bit
- 3 1x placa de ensaio
- 4 1x mini coluna de som
- 5 1x MOSFET canal-N
- 6 1x resistor de 100  $\Omega$   
(castanho/preto/castanho/dourado)

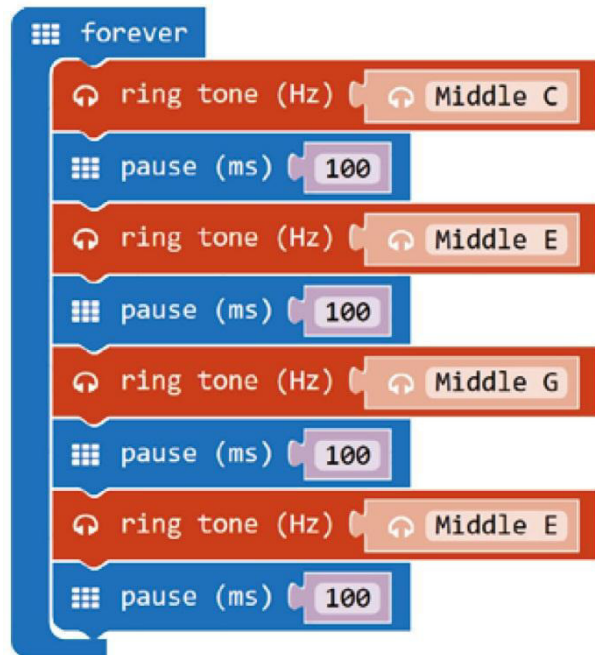


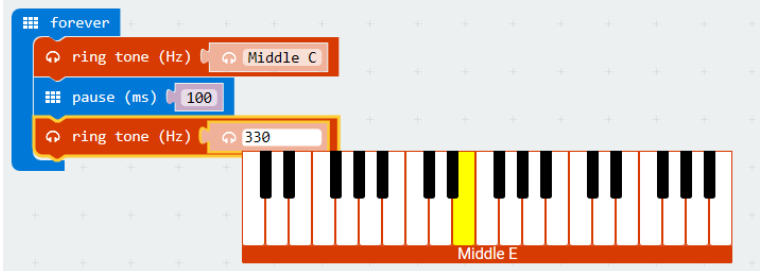
Vamos agora controlar uma campainha.

Disponha os componentes necessários na placa de ensaio, conforme indicado.



Dê uma espreitadela ao código abaixo.

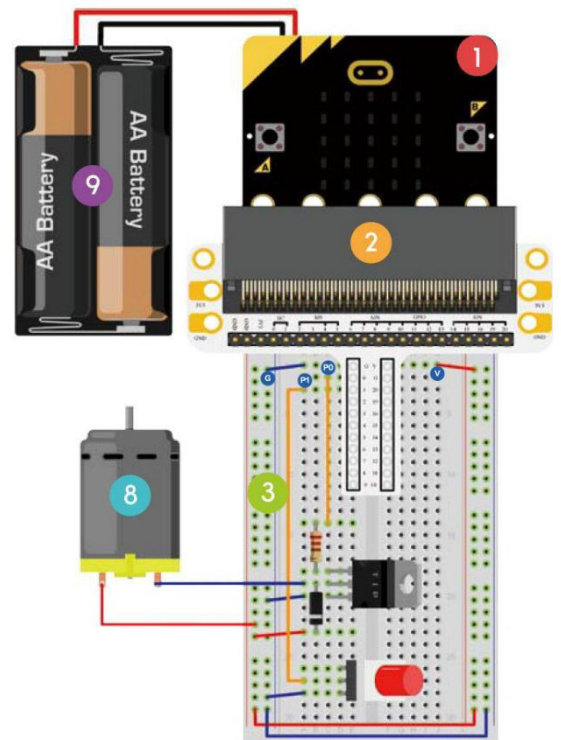
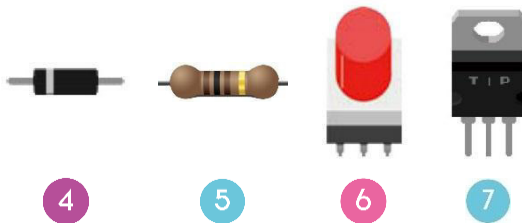


<p><b>1.</b></p>	<p>Arraste e solte os blocos de código para formar o código, conforme indicado.</p> <p>Pode encontrar <b>forever</b> e <b>pause (ms) 100</b> em <b>Basic</b> no Code Drawer. Pode encontrar o bloco <b>ring tone (Hz)</b> em <b>Music</b> no Code Drawer.</p>
<p><b>2.</b></p>	<p>Configure o primeiro <b>ring tone (Hz)</b> como <b>médio C</b>. Para tal, clique na caixa de texto e selecione a tecla de piano correspondente.</p>  <p>Irá também ouvir o toque correspondente.</p>
<p><b>3.</b></p>	<p>Defina a <b>pause</b> para 100 ms.</p>
<p><b>4.</b></p>	<p>Proceda da mesma forma para o resto dos toques e pausas.</p>
<p><b>5.</b></p>	<p>Quando terminado, compilamos o programa e geramos o ficheiro .hex. Clique no botão de download e guarde o ficheiro .hex na pasta Downloads → C:\downloads Este ficheiro .hex está pronto a ser carregado para o micro:bit.</p> <p>Ligue o micro:bit a uma porta USB. Em seguida, arraste e largue o ficheiro .hex no dispositivo amovível do micro: bit para carregar o programa.</p>

Podemos ouvir um som vindo da campainha. Agora, como programaria sua canção de embalar preferida?

## 5.10 Motor

- 1 1x placa micro:bit
- 2 1x adaptador placa de ensaio micro:bit
- 3 1x placa de ensaio
- 4 1x diodo
- 5 1x resistor de 100  $\Omega$   
(castanho/preto/castanho/dourado)
- 6 1x interruptor com bloqueio automático ou biestável
- 7 1x MOSFET canal-N
- 8 1x mini motor
- 9 1x suporte de pilhas com 2 pilhas AA de 1.5 V



Vamos usar um interruptor para controlar o arranque e paragem de um motor.

Disponha os componentes necessários na placa de ensaio, conforme indicado.

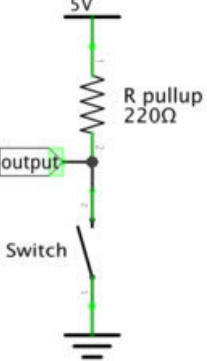
Dê uma espreitadela ao código abaixo.

```

on start
  digital write pin P0 to 0
  set pull pin P1 to up

forever
  if (digital read pin P1 = 0)
  then
    digital write pin P0 to 1
  else
    digital write pin P0 to 0
  
```

1.	Arraste e solte os blocos de código para formar o código, conforme indicado.
2.	O bloco <b>on start</b> é executado apenas uma vez para iniciar o programa.
3.	Defina o valor de <b>P0</b> para <b>0</b> .
4.	Defina o bloco <b>pull pin</b> para <b>P1</b> e <b>acima</b> .

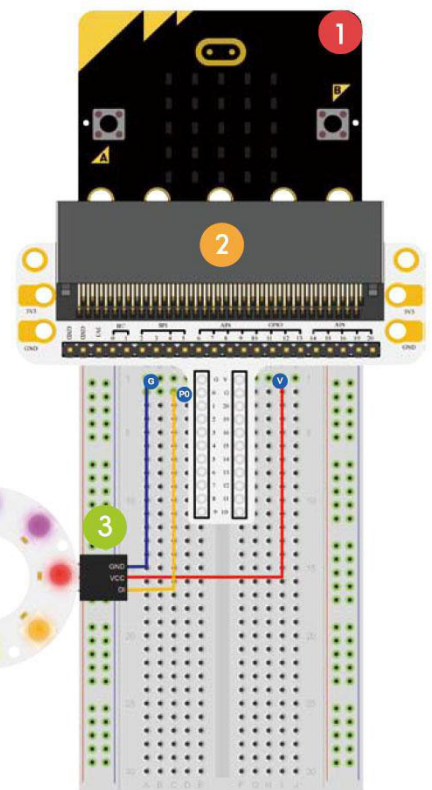
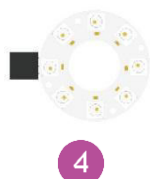
<p><b>5.</b></p>	<p>Em <b>forever</b>, defina o <b>digital read pin</b> para <b>P1</b>. Defina <b>mathematical function equal</b> para 0.</p> <p>Na realidade, este é o esquema. O resistor de elevação não deve ser adicionado como hardware à placa de ensaio. A função de elevação está programada (ver passos 3-4) no código e substitui a necessidade do resistor na placa de ensaio.</p> 
<p><b>6.</b></p>	<p>Assim que o interruptor for pressionado, ajuste a tensão alta para P0. Defina o valor para 0 a 1 (5 V) O motor começa a funcionar.</p>
<p><b>7.</b></p>	<p>Assim que o interruptor for libertado, ajuste a tensão baixa para P0. Defina o valor para 0. O motor começa a funcionar.</p>
<p><b>8.</b></p>	<p>Quando terminado, compilamos o programa e geramos o ficheiro .hex. Clique no botão de download e guarde o ficheiro .hex na pasta Downloads → C:\downloads Este ficheiro .hex está pronto a ser carregado para o micro:bit.</p> <p>Ligue o micro:bit a uma porta USB. Em seguida, arraste e largue o ficheiro .hex no dispositivo amovível do micro: bit para carregar o programa.</p>

Pressione o interruptor para fazer funcionar o motor e liberte-o para parar. Agora, como usaria um trimpot para controlar a velocidade do motor?

Observação: Uma vez que a tensão do micro:bit é de 3.3V, esta poderá não ser suficiente para suportar a ventoinha. Para fazer a ventoinha funcionar, poderá ter que girar a lâmina para ajudar a arrancar.

### 5.11 LED Arco-íris

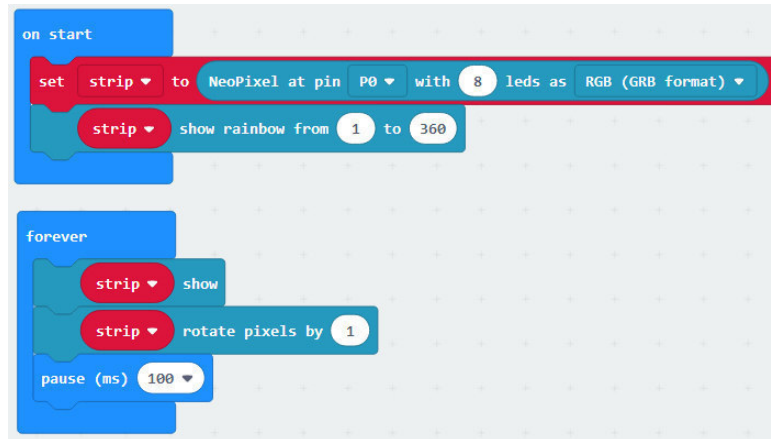
- 1 1x placa micro:bit
- 2 1x adaptador de placa de ensaio micro:bit
- 3 1x placa de ensaio
- 4 1x anel LED RGB



Vamos controlar oito LEDs RGB num anel e criar um arco-íris gradual.

Disponha os componentes necessários na placa de ensaio, conforme indicado.

Dê uma espreitadela no código abaixo.

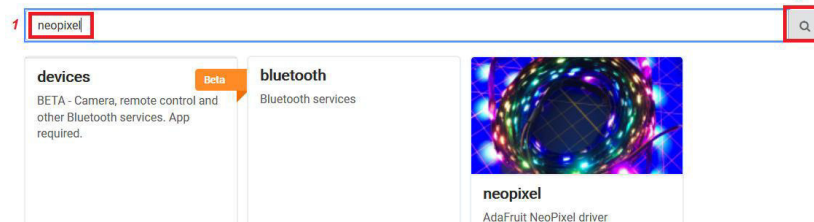


1. Arraste e solte os blocos de código para formar o código conforme indicado.

2. Procure e adicione a biblioteca NeoPixel.  
Vá para Advanced → Add Package (Extensions) e digite NeoPixel.

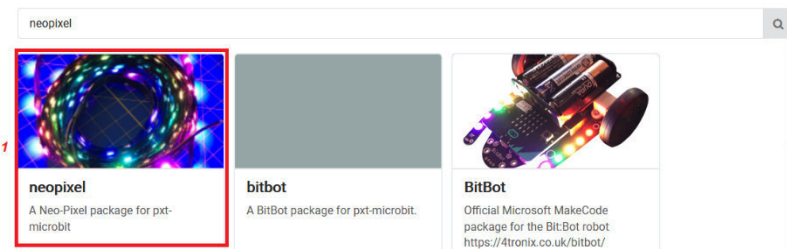


Add Package... ?

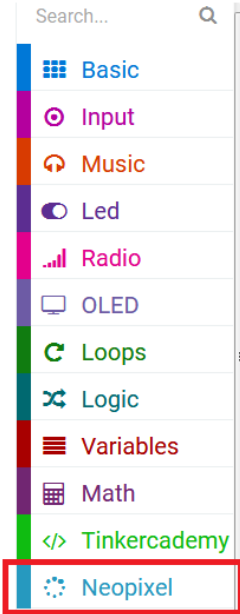

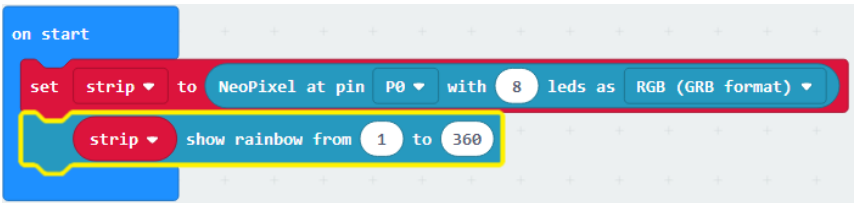


Em seguida, selecione a biblioteca NeoPixel.

Add Package... ?



Agora, a biblioteca foi descarregada e adicionada ao Code Drawer.

	 <p>Pode encontrar os blocos <b>NeoPixel at pin P0 with 24 leds, item show rainbow from 1 to 360, item show</b> e <b>items rotate pixels by 1</b> em <b>NeoPixel</b> no Code Drawer.</p>
3.	<p>Arraste e largue o bloco <b>set strip to NeoPixel noP0 com 24 leds RGB (formato GRB)</b> no bloco <b>on start</b>. Ajuste o pino para <b>P0</b> e o valor de 24 a 8 (temos 8 LEDs no anel LED).</p> 
4.	<p>Agora, arraste e solte o bloco <b>set strip to rainbow from 1 to 360</b> por baixo do bloco <b>set strip to NeoPixel at pin P0</b>.</p> 
5.	<p>Complete o código conforme o exemplo.</p>
6.	<p>Quando terminado, compilamos o programa e geramos o ficheiro .hex. Clique no botão de download e guarde o ficheiro .hex na pasta Downloads → C:\downloads Este ficheiro .hex está pronto a ser carregado para o micro:bit.</p> <p>Ligue o micro:bit a uma porta USB. Em seguida, arraste e largue o ficheiro .hex no dispositivo amovível do micro: bit para carregar o programa.</p>

Podemos ver o arco-íris girar no anel LED. Agora, como faria o anel piscar como se fosse um olho?

**Utilize este aparelho apenas com acessórios originais. A Velleman NV não será responsável por quaisquer danos ou lesões causados pelo uso (indevido) do aparelho. Para mais informação sobre este produto e para aceder à versão mais recente deste manual do utilizador, visite a nossa página [www.velleman.eu](http://www.velleman.eu). Podem alterar-se as especificações e o conteúdo deste manual sem aviso prévio.**

#### © DIREITOS DE AUTOR

**A Velleman NV detém os direitos de autor deste manual do utilizador. Todos os direitos mundiais reservados.** É estritamente proibido reproduzir, traduzir, copiar, editar e gravar este manual do utilizador ou partes deste sem prévia autorização escrita por parte da detentora dos direitos.



**EN****Velleman® Service and Quality Warranty**

Since its foundation in 1972, Velleman® acquired extensive experience in the electronics world and currently distributes its products in over 85 countries. All our products fulfil strict quality requirements and legal stipulations in the EU. In order to ensure the quality, our products regularly go through an extra quality check, both by an internal quality department and by specialized external organisations. If, all precautionary measures notwithstanding, problems should occur, please make appeal to our warranty (see guarantee conditions).

**General Warranty Conditions Concerning Consumer Products (for EU):**

- All consumer products are subject to a 24-month warranty on production flaws and defective material as from the original date of purchase.
- Velleman® can decide to replace an article with an equivalent article, or to refund the retail value totally or partially when the complaint is valid and a free repair or replacement of the article is impossible, or if the expenses are out of proportion.

You will be delivered a replacing article or a refund at the value of 100% of the purchase price in case of a flaw occurred in the first year after the date of purchase and delivery, or a replacing article at 50% of the purchase price or a refund at the value of 50% of the retail value in case of a flaw occurred in the second year after the date of purchase and delivery.

**• Not covered by warranty:**

- all direct or indirect damage caused after delivery to the article (e.g. by oxidation, shocks, falls, dust, dirt, humidity...), and by the article, as well as its contents (e.g. data loss), compensation for loss of profits;
- consumable goods, parts or accessories that are subject to an aging process during normal use, such as batteries (rechargeable, non-rechargeable, built-in or replaceable), lamps, rubber parts, drive belts... (unlimited list);
- flaws resulting from fire, water damage, lightning, accident, natural disaster, etc....;
- flaws caused deliberately, negligently or resulting from improper handling, negligent maintenance, abusive use or use contrary to the manufacturer's instructions;
- damage caused by a commercial, professional or collective use of the article (the warranty validity will be reduced to six (6) months when the article is used professionally);
- damage resulting from an inappropriate packing and shipping of the article;
- all damage caused by modification, repair or alteration performed by a third party without written permission by Velleman®.
- Articles to be repaired must be delivered to your Velleman® dealer, solidly packed (preferably in the original packaging), and be completed with the original receipt of purchase and a clear flaw description.
- Hint: In order to save on cost and time, please reread the manual and check if the flaw is caused by obvious causes prior to presenting the article for repair. Note that returning a non-defective article can also involve handling costs.
- Repairs occurring after warranty expiration are subject to shipping costs.
- The above conditions are without prejudice to all commercial warranties.

**The above enumeration is subject to modification according to the article (see article's manual).**

**NL****Velleman® service- en kwaliteitsgarantie**

Velleman® heeft sinds zijn oprichting in 1972 een ruime ervaring opgebouwd in de elektronica wereld en verdeelt op dit moment producten in meer dan 85 landen. Al onze producten beantwoorden aan strikte kwaliteitseisen en aan de wettelijke bepalingen geldig in de EU. Om de kwaliteit te waarborgen, ondergaan onze producten op regelmatige tijdstippen een extra kwaliteitscontrole, zowel door onze eigen kwaliteitsafdeling als door externe gespecialiseerde organisaties. Mocht er ondanks deze voorzorgen toch een probleem optreden, dan kunt u steeds een beroep doen op onze waarborg (zie waarborgvoorwaarden).

**Algemene waarborgvoorwaarden consumentengoederen (voor Europese Unie):**

- Op alle consumentengoederen geldt een garantieperiode van 24 maanden op productie- en materiaalfouten en dit vanaf de oorspronkelijke aankoopdatum.
- Indien de klacht gegrond is en een gratis reparatie of vervanging van een artikel onmogelijk is of indien de kosten hiervoor buiten verhouding zijn, kan Velleman® beslissen het desbetreffende artikel te vervangen door een gelijkwaardig artikel of de aankoopsum van het artikel gedeeltelijk of volledig terug te betalen. In dat geval krijgt u een vervangend product of terugbetaling ter waarde van 100% van de aankoopsum bij ontdekking van een gebrek tot één jaar na aankoop en levering, of een vervangend product tegen 50% van de kostprijs of terugbetaling van 50% bij ontdekking na één jaar tot 2 jaar.

**• Valt niet onder waarborg:**

- alle rechtstreekse of onrechtstreekse schade na de levering veroorzaakt aan het toestel (bv. door oxidatie, schokken, val, stof, vuil, vocht...), en door het toestel, alsook zijn inhoud (bv. verlies van data), vergoeding voor eventuele winstderving.
- verbruiksgoederen, onderdelen of hulpstukken die onderhevig zijn aan veroudering door normaal gebruik zoals bv. batterijen (zowel oplaadbare als niet-oplaadbare, ingebouwd of vervangbaar), lampen, rubberen onderdelen, aandrijfriemen... (onbeperkte lijst).
- defecten ten gevolge van brand, waterschade, bliksem, ongevallen, natuurrampen, enz.
- defecten veroorzaakt door opzet, nalatigheid of door een onoordeelkundige behandeling, slecht onderhoud of abnormaal gebruik of gebruik van het toestel strijdig met de voorschriften van de fabrikant.
- schade ten gevolge van een commercieel, professioneel of collectief gebruik van het apparaat (bij professioneel gebruik wordt de garantieperiode herleid tot 6 maand).
- schade veroorzaakt door onvoldoende bescherming bij transport van het apparaat.
- alle schade door wijzigingen, reparaties of modificaties uitgevoerd door derden zonder toestemming van Velleman®.
- Toestellen dienen ter reparatie aangeboden te worden bij uw Velleman®-verdelers. Het toestel dient vergezeld te zijn van het oorspronkelijke aankoopbewijs. Zorg voor een degelijke verpakking (bij voorkeur de originele verpakking) en voeg een duidelijke foutomschrijving bij.
- Tip: alvorens het toestel voor reparatie aan te bieden, kijk nog eens na of er geen voor de hand liggende reden is waarom het toestel niet naar behoren werkt (zie handleiding). Op deze wijze kunt u kosten en tijd besparen. Denk eraan dat er ook voor niet-defecte toestellen een kost voor controle aangerekend kan worden.
- Bij reparaties buiten de waarborgperiode zullen transportkosten aangerekend worden.
- Elke commerciële garantie laat deze rechten onverminderd.

**Bovenstaande opsomming kan eventueel aangepast worden naargelang de aard van het product (zie handleiding van het betreffende product).**

**FR****Garantie de service et de qualité Velleman®**

Depuis 1972, Velleman® a gagné une vaste expérience dans le secteur de l'électronique et est actuellement distributeur dans plus de 85 pays. Tous nos produits répondent à des exigences de qualité rigoureuses et à des dispositions légales en vigueur dans l'UE. Afin de garantir la qualité, nous soumettons régulièrement nos produits à des contrôles de qualité supplémentaires, tant par notre propre service qualité que par un service qualité externe. Dans le cas improbable d'un défaut malgré toutes les précautions, il est possible d'invoquer notre garantie (voir les conditions de garantie).

**Conditions générales concernant la garantie sur les produits grand public (pour l'UE) :**

- tout produit grand public est garanti 24 mois contre tout vice de production ou de matériaux à dater du jour d'acquisition effective ;
- si la plainte est justifiée et que la réparation ou le remplacement d'un article est jugé impossible, ou lorsque les coûts s'avèrent disproportionnés, Velleman® s'autorise à remplacer ledit article par un article équivalent ou à rembourser la totalité ou une partie du prix d'achat. Le cas échéant, il vous sera consenti un article de remplacement ou le remboursement complet du prix d'achat lors d'un défaut dans un délai de 1 an après l'achat et la livraison, ou un article de remplacement moyennant 50% du prix d'achat ou le remboursement de 50% du prix d'achat lors d'un défaut après 1 à 2 ans.

**• sont par conséquent exclus :**

- tout dommage direct ou indirect survenu à l'article après livraison (p.ex. dommage lié à l'oxydation, choc, chute, poussière, sable, impureté...) et provoqué par l'appareil, ainsi que son contenu (p.ex. perte de données) et une indemnisation éventuelle pour perte de revenus ;
- toute pièce ou accessoire nécessitant un remplacement causé par un usage normal comme p.ex. piles (rechargeables comme non rechargeables, intégrées ou remplaçables), ampoules, pièces en caoutchouc, courroies... (liste illimitée) ;
- tout dommage qui résulte d'un incendie, de la foudre, d'un accident, d'une catastrophe naturelle, etc. ;
- out dommage provoqué par une négligence, volontaire ou non, une utilisation ou un entretien incorrect, ou une utilisation de l'appareil contraire aux prescriptions du fabricant ;
- tout dommage à cause d'une utilisation commerciale, professionnelle ou collective de l'appareil (la période de garantie sera réduite à 6 mois lors d'une utilisation professionnelle) ;
- tout dommage à l'appareil qui résulte d'une utilisation incorrecte ou différente que celle pour laquelle il a été initialement prévu comme décrit dans la notice ;
- tout dommage engendré par un retour de l'appareil emballé dans un conditionnement non ou insuffisamment protégé.
- toute réparation ou modification effectuée par une tierce personne sans l'autorisation explicite de SA Velleman® ; - frais de transport de et vers Velleman® si l'appareil n'est plus couvert sous la garantie.
- toute réparation sera fournie par l'endroit de l'achat. L'appareil doit nécessairement être accompagné du bon d'achat d'origine et être dûment conditionné (de préférence dans l'emballage d'origine avec mention du défaut) ;
- tuyau : il est conseillé de consulter la notice et de contrôler câbles, piles, etc. avant de retourner l'appareil. Un appareil retourné jugé défectueux qui s'avère en bon état de marche pourra faire l'objet d'une note de frais à charge du consommateur ;
- une réparation effectuée en-dehors de la période de garantie fera l'objet de frais de transport ;
- toute garantie commerciale ne porte pas atteinte aux conditions susmentionnées.

**La liste susmentionnée peut être sujette à une complémentarité selon le type de l'article et être mentionnée dans la notice d'emploi.**

**ES****Garantía de servicio y calidad Velleman®**

Desde su fundación en 1972 Velleman® ha adquirido una amplia experiencia como distribuidor en el sector de la electrónica en más de 85 países. Todos nuestros productos responden a normas de calidad rigurosas y disposiciones legales vigentes en la UE. Para garantizar la calidad, sometemos nuestros productos regularmente a controles de calidad adicionales, tanto a través de nuestro propio servicio de calidad como de un servicio de calidad externo. En el caso improbable de que surgieran problemas a pesar de todas las precauciones, es posible recurrir a nuestra garantía (véase las condiciones de garantía).

**Condiciones generales referentes a la garantía sobre productos de venta al público (para la Unión Europea):**

- Todos los productos de venta al público tienen un período de garantía de 24 meses contra errores de producción o errores en materiales desde la adquisición original;
- Si la queja está fundada y si la reparación o sustitución de un artículo no es posible, o si los gastos son desproporcionados, Velleman® autoriza reemplazar el artículo por un artículo equivalente o reembolsar la totalidad o una parte del precio de compra. En este caso, usted recibirá un artículo de recambio o el reembolso completo del precio de compra si encuentra algún fallo hasta un año después de la compra y entrega, o un artículo de recambio al 50% del precio de compra o el reembolso del 50% del precio de compra si encuentra un fallo después de 1 año y hasta los 2 años después de la compra y entrega.

**Por consiguiente, están excluidos entre otras cosas:**

- todos los daños causados directa o indirectamente al aparato (p.ej. por oxidación, choques, caída,...) y a su contenido (p.ej. pérdida de datos) después de la entrega y causados por el aparato, y cualquier indemnización por posible pérdida de ganancias;
- partes o accesorios, que estén expuestos al desgaste causado por un uso normal, como por ejemplo baterías (tanto recargables como no recargables, incorporadas o reemplazables), bombillas, partes de goma, etc. (lista ilimitada);
- defectos causados por un incendio, daños causados por el agua, rayos, accidentes, catástrofes naturales, etc.;
- defectos causados a conciencia, descuido o por malos tratos, un mantenimiento inapropiado o un uso anormal del aparato contrario a las instrucciones del fabricante;
- daños causados por un uso comercial, profesional o colectivo del aparato (el período de garantía se reducirá a 6 meses con uso profesional);
- daños causados por un uso incorrecto o un uso ajeno al que está previsto el producto inicialmente como está descrito en el manual del usuario;
- daños causados por una protección insuficiente al transportar el aparato.
- daños causados por reparaciones o modificaciones efectuadas por una tercera persona sin la autorización explícita de Velleman® ;
- se calcula gastos de transporte de y a Velleman® si el aparato ya no está cubierto por la garantía.
- Cualquier artículo que tenga que ser reparado tendrá que ser devuelto a su distribuidor Velleman®. Devuelva el aparato con la factura de compra original y transportélo en un embalaje sólido (preferentemente el embalaje original). Incluya también una buena descripción del fallo;
- Consejo: Lea el manual del usuario y controle los cables, las pilas, etc. antes de devolver el aparato. Si no se encuentra un defecto en el artículo los gastos podrían correr a cargo del cliente;
- Los gastos de transporte correrán a carga del cliente para una reparación efectuada fuera del período de garantía.
- Cualquier gesto comercial no disminuye estos derechos.

**La lista previamente mencionada puede ser adaptada según el tipo de artículo (véase el manual del usuario del artículo en cuestión).**

**DE****Velleman® Service- und Qualitätsgarantie**

Seit der Gründung in 1972 hat Velleman® sehr viel Erfahrung als Verteiler in der Elektronikwelt in über 85 Ländern aufgebaut. Alle Produkte entsprechen den strengen Qualitätsforderungen und gesetzlichen Anforderungen in der EU. Um die Qualität zu gewährleisten werden unsere Produkte regelmäßig einer zusätzlichen Qualitätskontrolle unterworfen, sowohl von unserer eigenen Qualitätsabteilung als auch von externen spezialisierten Organisationen. Sollten, trotz aller Vorsichtsmaßnahmen, Probleme auftreten, nehmen Sie bitte die Garantie in Anspruch (siehe Garantiebedingungen).

**Allgemeine Garantiebedingungen in Bezug auf Konsumgüter (für die Europäische Union):**

- Alle Produkte haben für Material- oder Herstellungsfehler eine Garantieperiode von 24 Monaten ab Verkaufsdatum.
- Wenn die Klage berechtigt ist und falls eine kostenlose Reparatur oder ein Austausch des Gerätes unmöglich ist, oder wenn die Kosten dafür unverhältnismäßig sind, kann Velleman® sich darüber entscheiden, dieses Produkt durch ein gleiches Produkt zu ersetzen oder die Kaufsumme ganz oder teilweise zurückzuzahlen. In diesem Fall erhalten Sie ein Ersatzprodukt oder eine Rückzahlung im Werte von 100% der Kaufsumme im Falle eines Defektes bis zu 1 Jahr nach Kauf oder Lieferung, oder Sie bekommen ein Ersatzprodukt im Werte von 50% der Kaufsumme oder eine Rückzahlung im Werte von 50 % im Falle eines Defektes im zweiten Jahr.

**• Von der Garantie ausgeschlossen sind:**

- alle direkten oder indirekten Schäden, die nach Lieferung am Gerät und durch das Gerät verursacht werden (z.B. Oxidation, Stöße, Fall, Staub, Schmutz, Feuchtigkeit, ...), sowie auch der Inhalt (z.B. Datenverlust), Entschädigung für eventuellen Gewinnausfall.
- Verbrauchsgüter, Teile oder Zubehörteile, die durch normalen Gebrauch dem Verschleiß ausgesetzt sind, wie z.B. Batterien (nicht nur aufladbare, sondern auch nicht aufladbare, eingebaute oder ersetzbare), Lampen, Gummiteile, Treibriemen, usw. (unbeschränkte Liste).
- Schäden verursacht durch Brandschaden, Wasserschaden, Blitz, Unfälle, Naturkatastrophen, usw.
- Schäden verursacht durch absichtliche, nachlässige oder unsachgemäße Anwendung, schlechte Wartung, zweckentfremdete Anwendung oder Nichtbeachtung von Benutzerhinweisen in der Bedienungsanleitung.
- Schäden infolge einer kommerziellen, professionellen oder kollektiven Anwendung des Gerätes (bei gewerblicher Anwendung wird die Garantieperiode auf 6 Monate zurückgeführt).
- Schäden verursacht durch eine unsachgemäße Verpackung und unsachgemäßen Transport des Gerätes.
- alle Schäden verursacht durch unautorisierte Änderungen, Reparaturen oder Modifikationen, die von einem Dritten ohne Erlaubnis von Velleman® vorgenommen werden.
- Im Fall einer Reparatur, wenden Sie sich an Ihren Velleman®-Verteiler. Legen Sie das Produkt ordnungsgemäß verpackt (vorzugsweise die Originalverpackung) und mit dem Original-Kaufbeleg vor. Fügen Sie eine deutliche Fehlerbeschreibung hinzu.
- Hinweis: Um Kosten und Zeit zu sparen, lesen Sie die Bedienungsanleitung nochmals und überprüfen Sie, ob es keinen auf der Hand liegenden Grund gibt, ehe Sie das Gerät zur Reparatur zurückschicken. Stellt sich bei der Überprüfung des Gerätes heraus, dass kein Geräteschaden vorliegt, könnte dem Kunden eine Untersuchungspauschale berechnet.
- Für Reparaturen nach Ablauf der Garantiefrist werden Transportkosten berechnet.
- Jede kommerzielle Garantie lässt diese Rechte unberührt.

**Die oben stehende Aufzählung kann eventuell angepasst werden gemäß der Art des Produktes (siehe Bedienungsanleitung des Gerätes).****PL****Velleman® usługi i gwarancja jakości**

Od czasu założenia w 1972, Velleman® zdobył bogate doświadczenie w dziedzinie światowej elektroniki. Obecnie firma dystrybuje swoje produkty w ponad 85 krajach.

Wszystkie nasze produkty spełniają surowe wymagania jakościowe oraz wypełniają normy i dyrektywy obowiązujące w krajach UE. W celu zapewnienia najwyższej jakości naszych produktów, przechodzą one regularne oraz dodatkowo wyrównane badania kontroli jakości, zarówno naszego wewnętrznego działu jakości jak również wyspecjalizowanych firm zewnętrznych. Pomimo dołożenia wszelkich starań czasem mogą pojawić się problemy techniczne, prosimy odwołać się do gwarancji (patrz warunki gwarancji).

**Ogólne Warunki dotyczące gwarancji:**

- Wszystkie produkty konsumenckie podlegają 24-miesięcznej gwarancji na wady produkcyjne i materiałowe od daty zakupu.
- W przypadku, gdy usterka jest niemożliwa do usunięcia lub koszt usunięcia jest nadmiernie wysoki Velleman® może zdecydować o wymianie artykułu na nowy, wolny od wad lub zwrócić zapłaconą kwotę. Zwrot gotówki może jednak nastąpić z uwzględnieniem poniższych warunków:
  - zwrot 100% ceny zakupu w przypadku, gdy wada wystąpiła w ciągu pierwszego roku od daty zakupu i dostawy
  - wymiana wadliwego artykułu na nowy, wolny od wad z odpłatnością 50% ceny detalicznej lub zwrot 50% kwoty ceny nabycia w przypadku gdy wada wystąpiła w drugim roku od daty zakupu i dostawy.
- **Produkt nie podlega naprawie gwarancyjnej:**
  - gdy wszystkie bezpośrednie lub pośrednie szkody spowodowane są działaniem czynników środowiskowych lub losowych (np. przez utlenianie, wstrząsy, upadki, kurz, brud, ...), wilgotności;
  - gwarant nie ponosi odpowiedzialności za szkody wynikających z utraty danych;
  - produkty konsumenckie, części zamienne lub akcesoria podatne na proces starzenia, wynikającego z normalnego użytkowania, np: baterie (ładowalne, nieladowalne, wbudowane lub wymienne), żarówki, paski napędowe, gumowe elementy napędowe... (nieograniczona lista);
  - usterka wynika z działania pożaru, zalania wszelkimi cieczami, uderzenia pioruna, upadku lub kłęski żywiołowej, itp.;
  - usterka wynika z zaniedbań eksploatacyjnych tj. umyślne bądź nieumyślne zaniechanie czyszczenia, konserwacji, wymiany materiałów eksploatacyjnych, niedbalstwa lub z niewłaściwego obchodzenia się lub niezgodnego użytkowania z instrukcją producenta;
  - szkody wynikające z nadmiernego użytkowania gdy nie jest do tego celu przeznaczony tj. działalność komercyjna, zawodowa lub wspólne użytkowanie przez wiele osób - okres obowiązywania gwarancji zostanie obniżony do 6 (sześć) miesięcy;
  - Szkody wynikające ze źle zabezpieczonej wysyłki produktu;
  - Wszelkie szkody spowodowane przez nieautoryzowaną naprawę, modyfikację, przeróbkę produktu przez osoby trzecie jak również bez pisemnej zgody firmy Velleman®.
- Uszkodzony produkt musi zostać dostarczony do sprzedawcy® Velleman, solidnie zapakowany (najlepiej w oryginalnym opakowaniu), wraz z wyposażeniem z jakim produkt został sprzedany. W przypadku wysyłki towaru w opakowaniu innym niż oryginalnym ryzyko usterki produktu oraz tego skutki przechodzą na właściciela produktu. Wraz z niesprawnym produktem należy dołączyć jasny i szczegółowy opis jego usterki, wady;
- Wskazówka: Aby zaoszczędzić na kosztach i czasie, proszę szczegółowo zapoznać się z instrukcją obsługi; czy przyczyna wady są okoliczności techniczne czy też wynikają wyłącznie z nieznaności obsługi produktu. W przypadku wysyłki sprawnego produktu do serwisu nabywca może zostać obciążony kosztami obsługi oraz transportu.

- W przypadku napraw pogwarancyjnych lub odpłatnych klient ponosi dodatkowo koszt wysyłki produktu do i z serwisu.
- wymienione wyżej warunki są bez uszczerbku dla wszystkich komercyjnych gwarancji.

**Powwyższe postanowienia mogą podlegać modyfikacji w zależności od wyrobu (patrz art obsługi).****PT****Garantia de serviço e de qualidade Velleman®**

Desde a sua fundação em 1972 Velleman® tem adquirido uma ampla experiência no sector da eletrónica com uma distribuição em mais de 85 países. Todos os nossos produtos respondem a exigências rigorosas e a disposições legais em vigor na UE. Para garantir a qualidade, submetemos regularmente os nossos produtos a controlos de qualidade suplementares, com o nosso próprio serviço qualidade como um serviço de qualidade externo. No caso improvável de um defeito mesmo com as nossas precauções, é possível invocar a nossa garantia. (ver as condições de garantia).

**Condições gerais com respeito a garantia sobre os produtos grande público (para a UE):**

- qualquer produto grande público é garantido 24 mês contra qualquer vício de produção ou materiais a partir da data de aquisição efectiva;
  - no caso da reclamação ser justificada e que a reparação ou substituição de um artigo é impossível, ou quando os custos são desproporcionados, Velleman® autoriza-se a substituir o dito artigo por um artigo equivalente ou a devolver a totalidade ou parte do preço de compra. Em outro caso, será consentido um artigo de substituição ou devolução completa do preço de compra no caso de um defeito no prazo de 1 ano depois da data de compra e entrega, ou um artigo de substituição pagando o valor de 50% do preço de compra ou devolução de 50% do preço de compra para defeitos depois de 1 a 2 anos.
  - **estão por consequência excluídos:**
    - todos os danos directos ou indirectos depois da entrega do artigo (p.ex. danos ligados a oxidação, choques, quedas, poeiras, areias, impurezas...) e provocado pelo aparelho, como o seu conteúdo (p.ex. perda de dados) e uma indemnização eventual por perda de receitas;
    - consumíveis, peças ou acessórios sujeitos a desgaste causado por um uso normal, como p.ex. pilhas (recarregáveis, não recarregáveis, incorporadas ou substituíveis), lâmpadas, peças em borracha correias... (lista ilimitada);
    - todos os danos que resultem de um incêndio, raios, de um acidente, de uma catastrophe natural, etc.;
    - danos provocados por negligencia, voluntária ou não, uma utilização ou manutenção incorrecta, ou uma utilização do aparelho contrária as prescrições do fabricante;
    - todos os danos por causa de uma utilização comercial, profissional ou colectiva do aparelho ( o período de garantia será reduzido a 6 meses para uma utilização profissional);
    - todos os danos no aparelho resultando de uma utilização incorrecta ou diferente daquela inicialmente prevista e descrita no manual de utilização;
    - todos os danos depois de uma devolução não embalada ou mal protegida ao nível do acondicionamento.
  - todas as reparações ou modificações efectuadas por terceiros sem a autorização de SA Velleman®;
  - despesas de transporte de e para Velleman® se o aparelho não estiver coberto pela garantia.
  - qualquer reparação será fornecida pelo local de compra. O aparelho será obrigatoriamente acompanhado do talão ou factura de origem e bem acondicionado (de preferência dentro da embalagem de origem com indicação do defeito ou avaria);
  - dica: aconselha-mos a consulta do manual e controlar cabos, pilhas, etc. antes de devolver o aparelho. Um aparelho devolvido que estiver em bom estado será cobrado despesas a cargo do consumidor;
  - uma reparação efectuada fora da garantia, será cobrado despesas de transporte;
  - qualquer garantia comercial não prevalece as condições aqui mencionadas.
- A lista pode ser sujeita a um complemento conforme o tipo de artigo e estar mencionada no manual de utilização.**