

# Table of Contents

1 Overview .....	2
1.1 What's in the Package: .....	3
1.2 Specification .....	3
1.3 Interface Introduction.....	4
2 Assemble the Joystick .....	6
Step 1 - Assemble the Acrylic Extension Handle .....	6
Step 2 – Install the batteries .....	7
Step 3 – Install the bottom acrylic battery cover .....	8
Step 4 – Attach the button caps.....	9
Step 5 – Turn off the power switch.....	9
Step 5 – Install the micro:bit.....	10
3 Getting Started with MakeCode .....	10
3.1 Create a new project .....	11
3.2 Delete a Project.....	13
3.3 Save a project .....	14
3.4 Import Files .....	14
3.5 Upload code .....	15
3.6 Unpairing Microbit .....	19
3.7 Upload the HEX file to the Micro:bit .....	20
3.8 Learn the basic syntax of Makecode .....	22
4 MakeCode Extension .....	23
4.1 Add Joystick extension .....	23
4.2 Refresh Joystick extension .....	24
4.3 Parsing of Joystick extension statement .....	25
5 Basic Example Projects .....	26
5.1 Button .....	26
5.2 Joystick .....	27
5.3 Vibration motor .....	28
5.4 Battery voltage .....	29
5.5 Servo .....	31
5.6 Wireless control .....	33
6 Extended Projects .....	36
6.1 Control mCar .....	37
6.2 Control Pybit .....	41
6.3 Joystick Control Robot Arm.....	45
7 QA .....	46
7.1 Micro:bit Code Upload Failure .....	46
7.2 Abnormal Micro:bit Drive Presentation .....	46
7.3 Buttons or Joystick Not Responding .....	46
7.4 Cannot Find “Radio” .....	46
7.5 Other Issues .....	47
8 Contact Us .....	47

# 1 Overview

This expansion board is an interactive input peripheral designed for the BBC micro:bit. It integrates a dual-axis joystick, multiple function buttons, and a vibration feedback module, making it suitable for applications such as game controllers, robot remote control, interactive teaching, and more.

## Features:

**Dual-Axis Analog Joystick:** Provides analog input in both X and Y axes, enabling real-time position detection.

**Multi-Function Buttons:** Includes four independent digital buttons (A/B/C/D) in addition to the joystick's built-in button, all programmable for custom actions.

**Vibration Motor:** Equipped with a 3610-type vibration motor for haptic feedback, triggered via program control.

**Expansion Interfaces:** Offers multiple digital I/O pins, servo control pins, and SPI communication support for connecting additional sensors or actuators.

**Independent Power Supply:** Can be powered by 4 AA batteries (nominal 6V), featuring a dedicated power switch—no need to rely on the micro:bit's onboard power.

**Plug-and-Play Design:** Includes a micro:bit edge connector slot for direct insertion without additional wiring.

## Typical Applications:

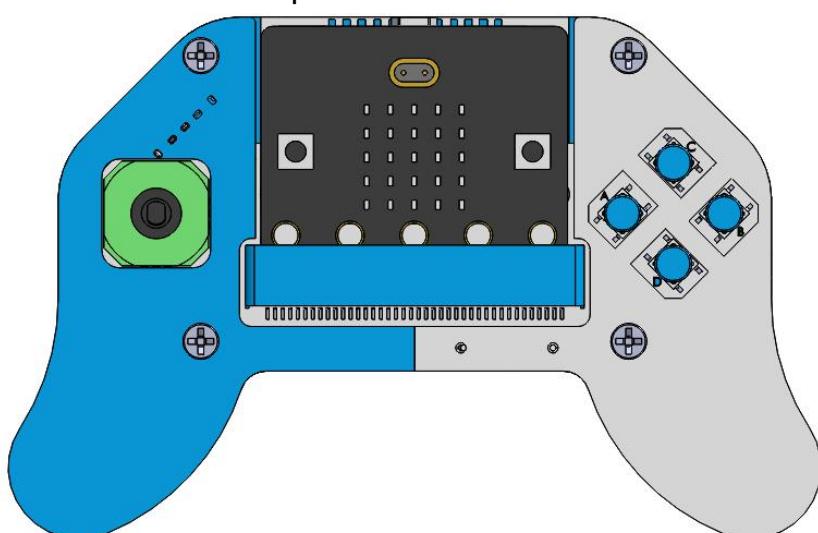
Game controller development

Robotic motion control

STEM education projects

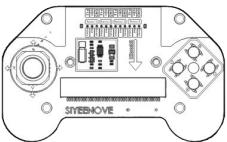
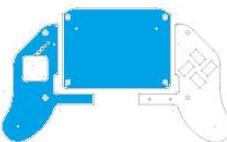
Interactive art installations

Physical feedback simulation experiments



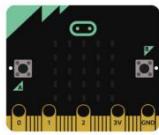
The Micro:bit must be purchased separately!

## 1.1 What's in the Package:

Joystick 1Pcs	3x18mm nylon standoffs 4Pcs	3x9mm screws 8Pcs
		
Acrylic plates 3Pcs	Screwdriver 1Pcs	Button caps 4Pcs
		

Note: The acrylic pieces are white and blue. Please remove the protective film covering their surfaces.

## What you'll need (not included)

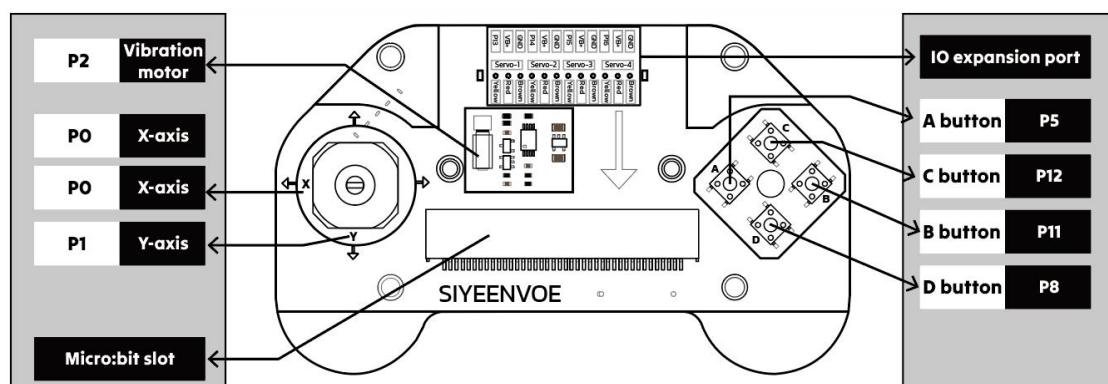
			
Micro:bit V2 1PCS	Micro USB Cable	computer with internet & a USB port	1.5V AA Battery 4 pcs

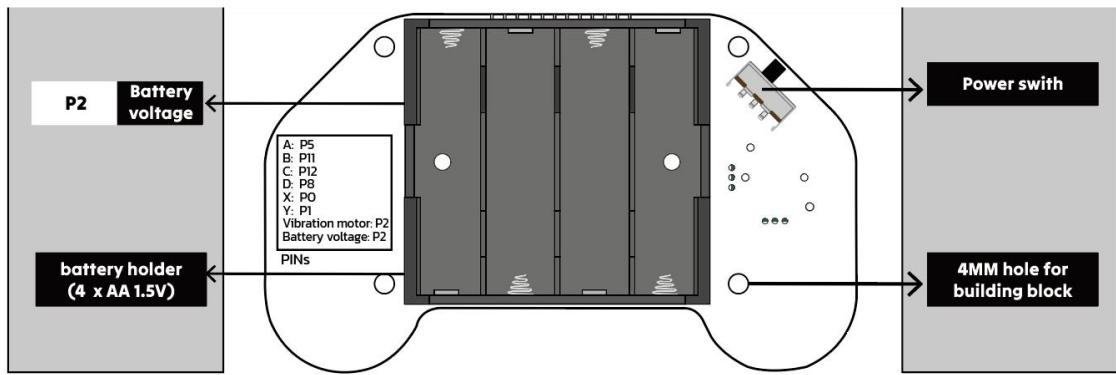
## 1.2 Specification

Kit	
Name	Joystick
SKU	M1K0000
Applicable motherboard	
Micro:bit	

Pins	
Digital I/O Pins	4 (P13, P14, P15, P16)
Servo pins	4 (P13, P14, P15, P16)
Communication	
SPI	Yes (P13, P14, P15, P16)
Power	
Input voltage (nominal)	6V (4 AA batteries, 1.5V/PCS)
VB(+/-)	6V (4 AA batteries, 1.5V/PCS)
Vibration motor	
Model	3610 Vibration motor
Rated voltage/current	2.7V/75mA
Rotation speed	14000±2500RPM
Dimensions	
Width	72.5 mm
Length	120 mm
Height	29mm
Weight	
PCBA	50 g
With Acrylic	82 g

## 1.3 Interface Introduction



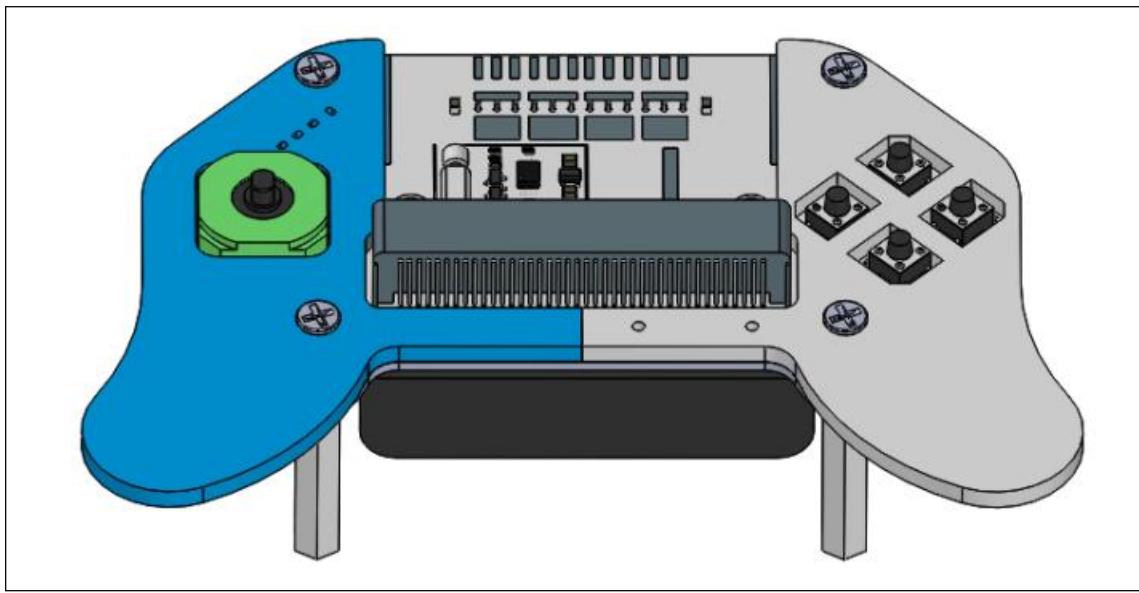


Name	Description
Vibration motor	Provides haptic feedback to the handle, controlled by Micro:bit's
X-axis	The joystick's X-axis, whose analog value can be read through Micro:bit's P0 pin
Y-axis	The joystick's Y-axis, whose analog value can be read through Micro:bit's P1 pin
Micro:bit slot	Designed for inserting the Micro:bit main control board
4 x AA 1.5V battery case	The battery case accommodates 4 batteries, each with standard 1.5V voltage, approximately 14.5mm in diameter and 50.5mm in height
Battery voltage	When 4 AA batteries are installed, the voltage can be read through Micro:bit's P2 pin
IO extension	For connecting external servos or sensors. The VB pin voltage equals the series voltage of 4 AA batteries ( $1.5V \times 4 = 6V$ )
A button	Its value can be read through Micro:bit's P5 pin
B button	Its value can be read through Micro:bit's P12 pin
C button	Its value can be read through Micro:bit's P11 pin
D button	Its value can be read through Micro:bit's P8 pin
Power switch	It controls the main power supply of the handle.
4MM hole	The handle features 4x4mm holes compatible with LEGO building block for expansion

## 2 Assemble the Joystick

### Step 1 - Assemble the Acrylic Extension Handle

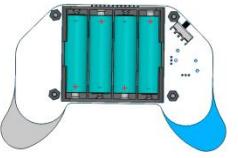
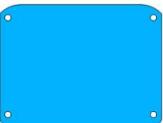
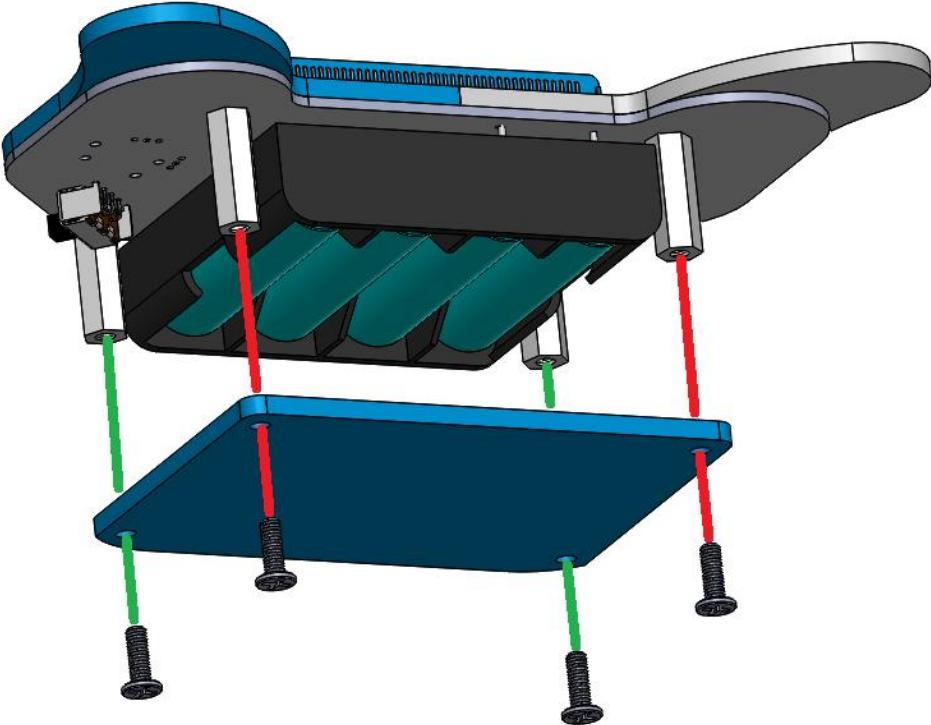
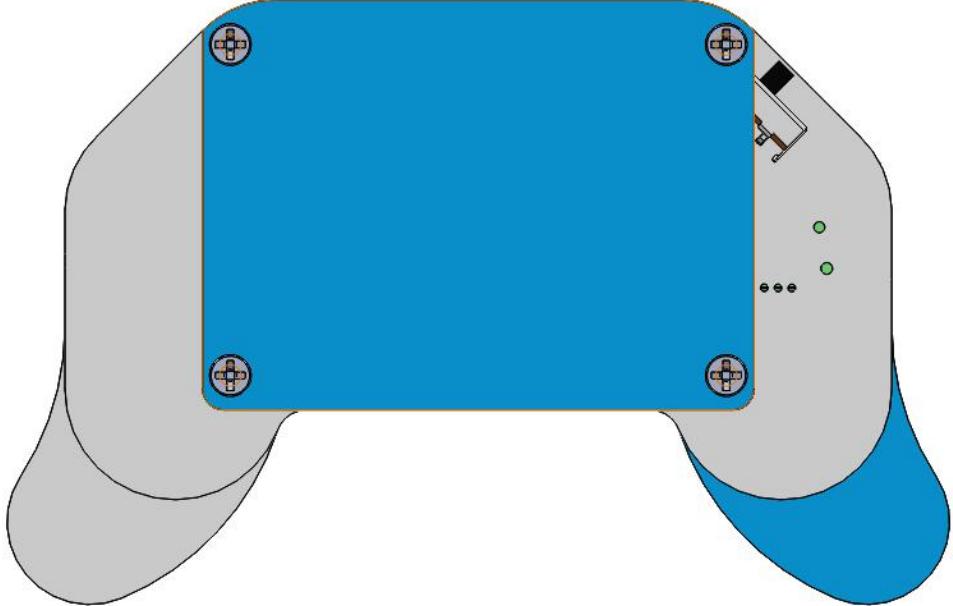
Joystick × 1	Acrylic plate × 1	Acrylic plate × 1
3×18mm nylon standoffs × 4	3×9mm screws × 4	<b>Note</b>
		Please remove the protective film covering the surface of the acrylic plates.



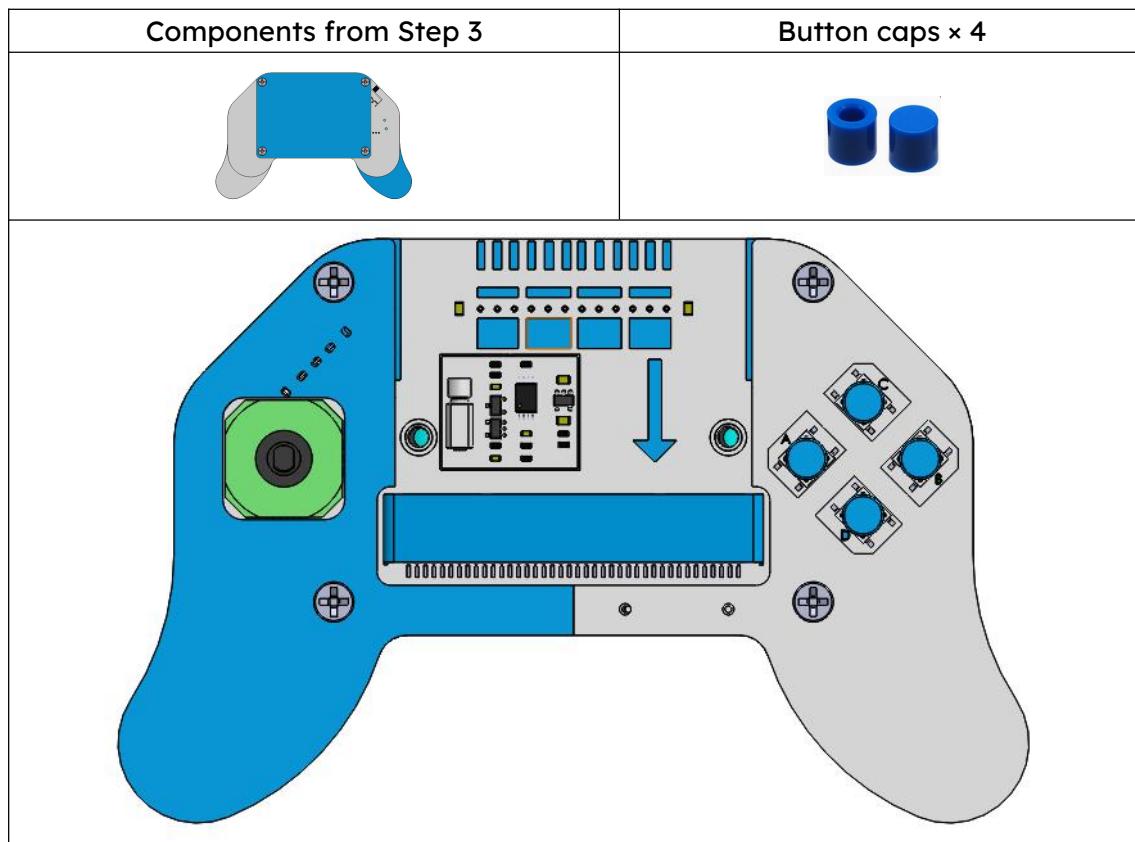
## Step 2 – Install the batteries

Components from Step 1	AA batteries × 4

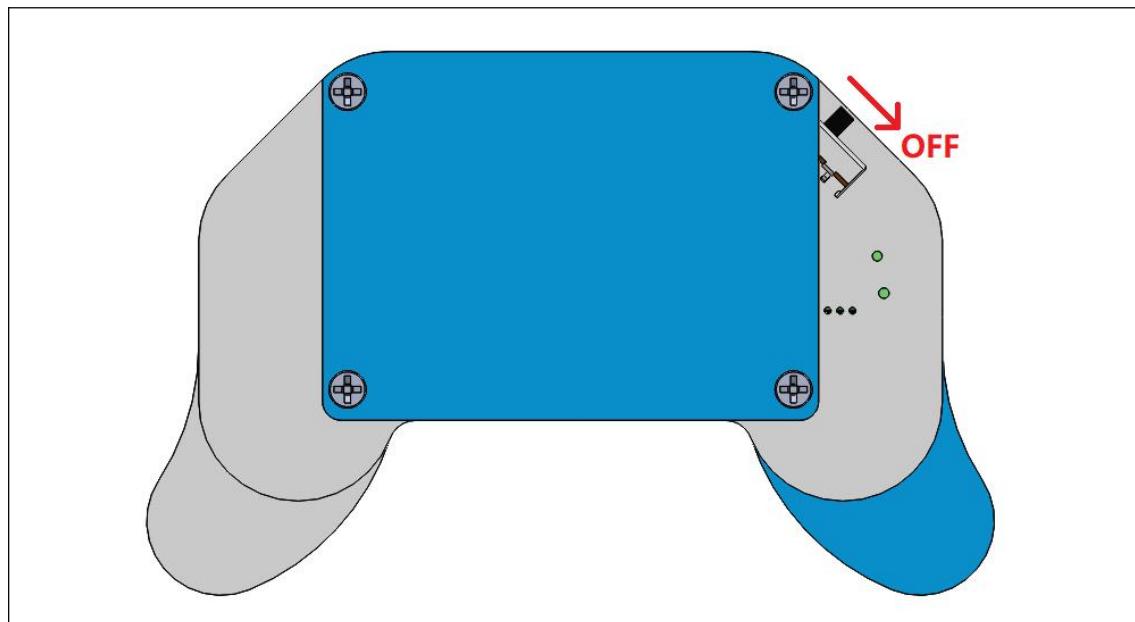
### Step 3 – Install the bottom acrylic battery cover

Components from Step 2	Acrylic plate × 1	Screws × 4
		
		
		

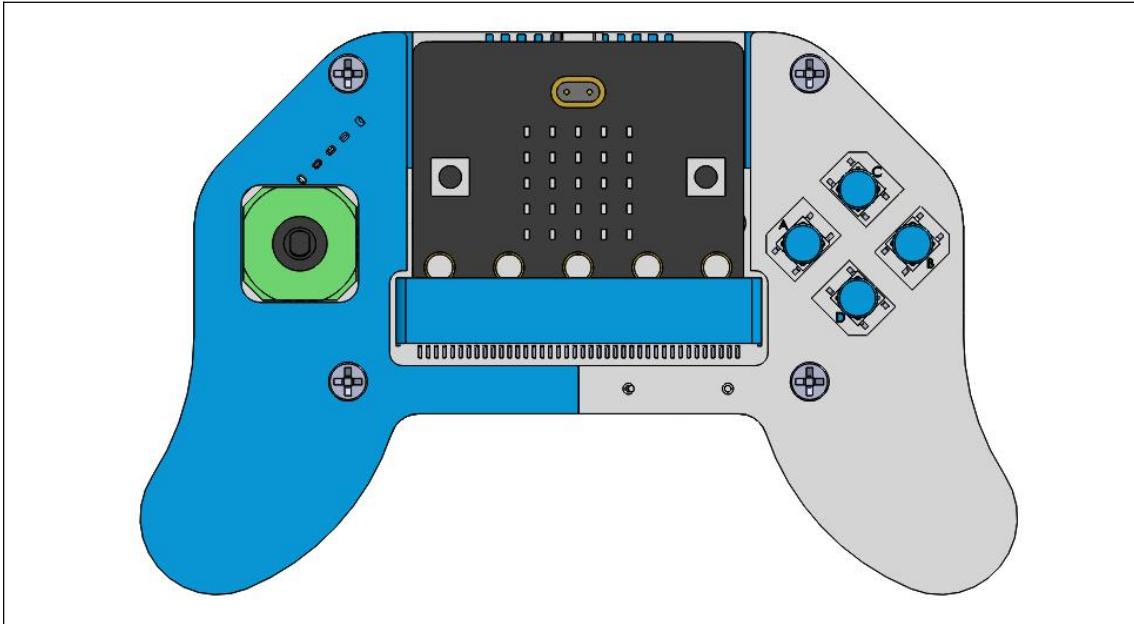
## Step 4 - Attach the button caps



## Step 5 - Turn off the power switch



## Step 6 - Install the micro:bit



## 3 Getting Started with MakeCode

Microsoft's MakeCode editor is the perfect way to start coding with the BBC micro:bit. MakeCode is free and works across all platforms and browsers.

We recommend using Chrome or Edge browsers. WebUSB is a recent and developing web feature that allows you to access a micro:bit directly from a web page. It also lets you directly receive data into the MakeCode editor from the micro:bit. It works in Google Chrome and Microsoft Edge browsers.

### WebUSB support for your micro:bit

If you're not using a current version of the Chrome or Microsoft Edge browsers, make sure they are this version or newer:

Chrome (version 79 and newer) browser for Android, Chrome OS, Linux, macOS and Windows 10.

Microsoft Edge (version 79 and newer) browser for Android, Chrome OS, Linux, macOS and Windows 10.

Link to download the latest Google Chrome:

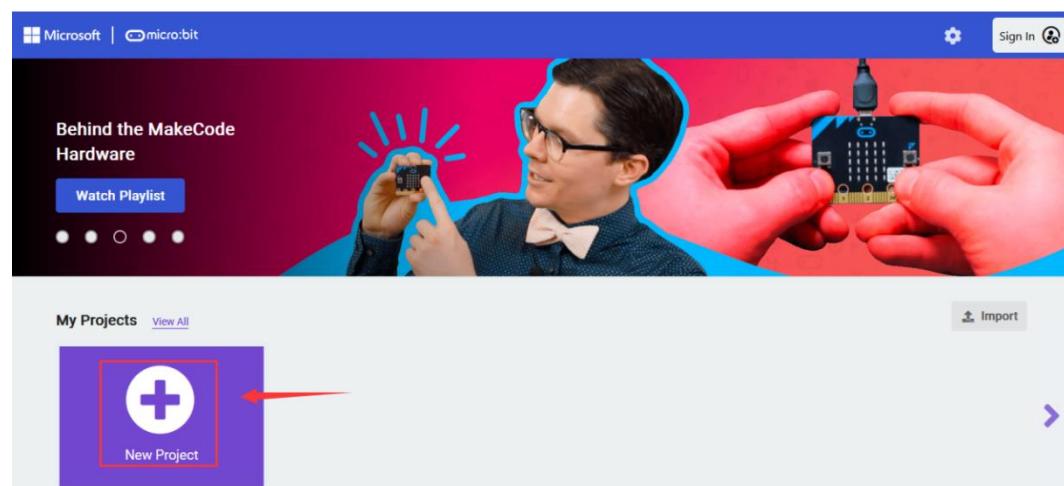
<https://www.google.com/chrome/>

Link to download the latest Microsoft Edge:

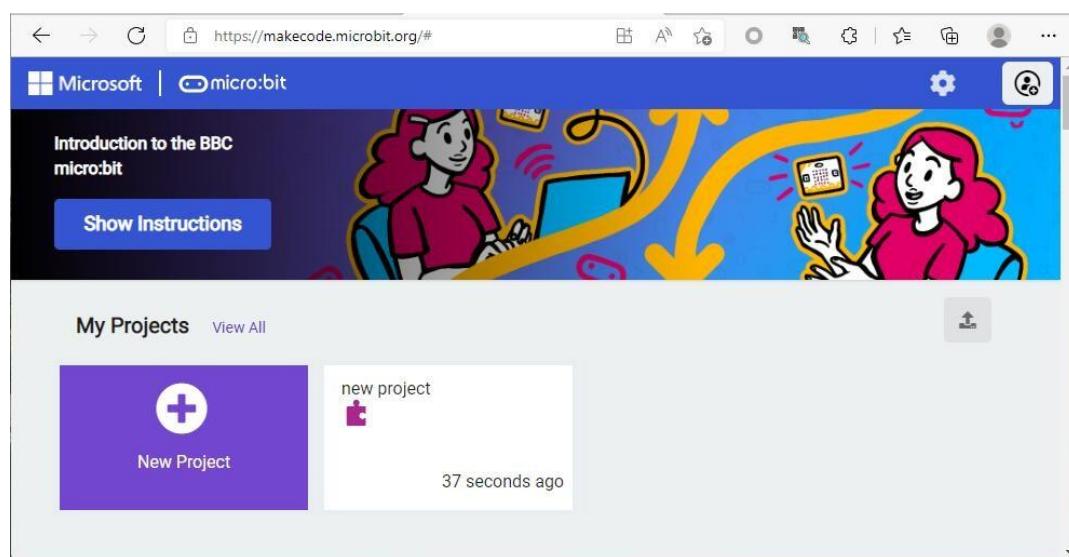
<https://www.microsoft.com/en-us/edge/download>

### 3.1 Create a new project

Open the Makecode editor on your browser: <https://makecode.microbit.org>, click “New Project”, Then you can give a name for your project.

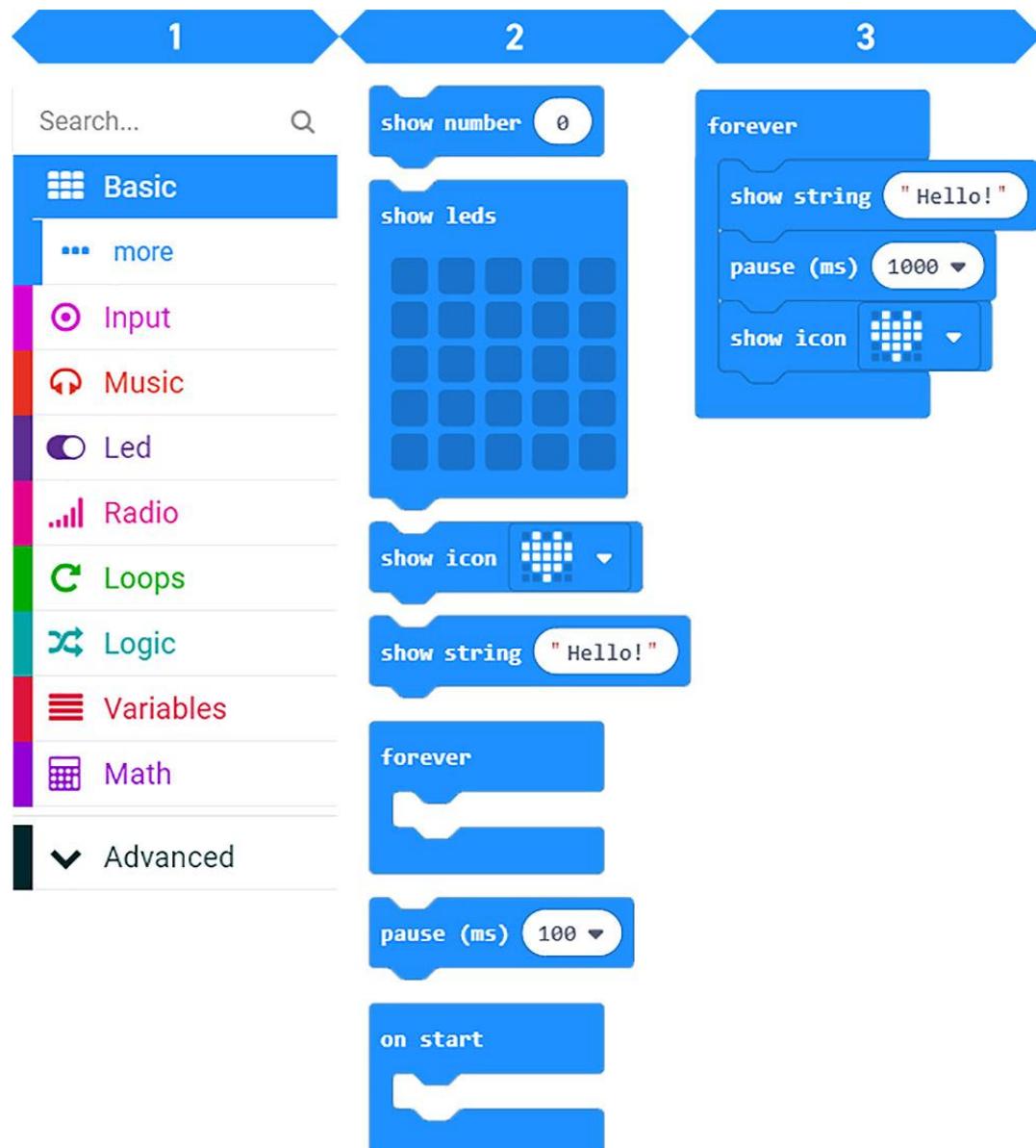


The newly created projects will be saved in the current browser. Just revisit the <https://makecode.microbit.org> website and find them in the project list.



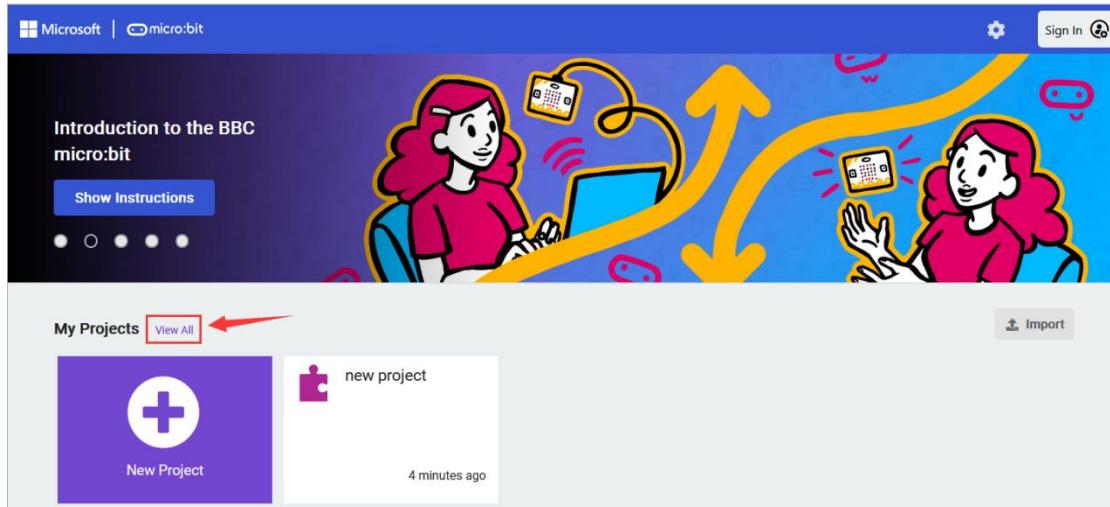
### Selecting Blocks:

1. Select a block category from the list on the left-hand side of the page.
2. Select a block from the selected category, then drag it to the workspace area on the right.
3. Snap new blocks onto existing blocks in the workspace area. As the new blocks are dragged into the workspace, the editor highlights the connecting parts of each block when they are in a valid position to snap to existing blocks. Also, the shape of the blocks gives you an indication of where they might fit into your code blocks.

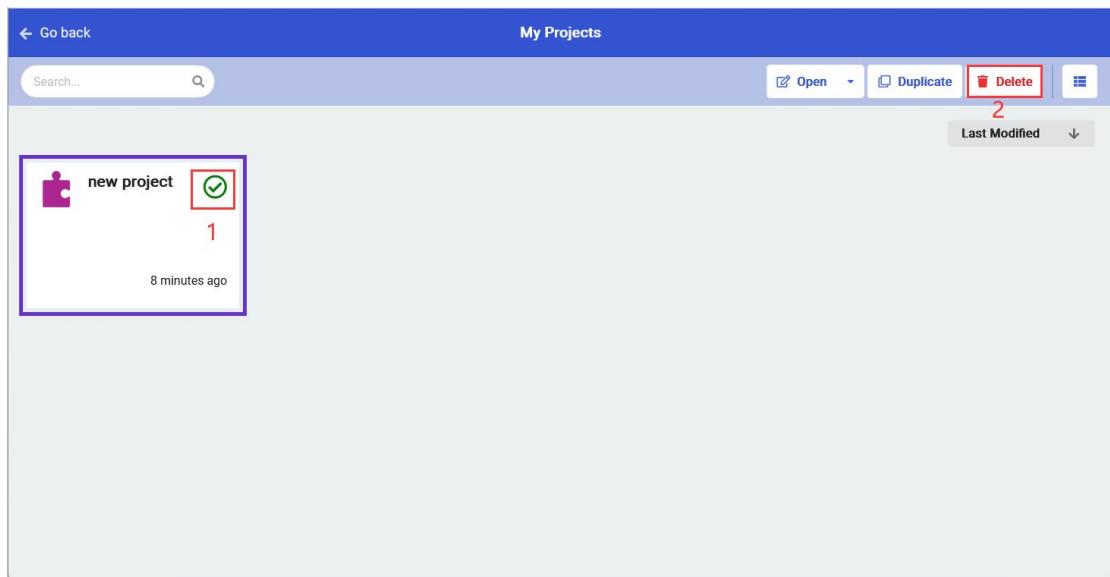


## 3.2 Delete a Project

Click “View All”:

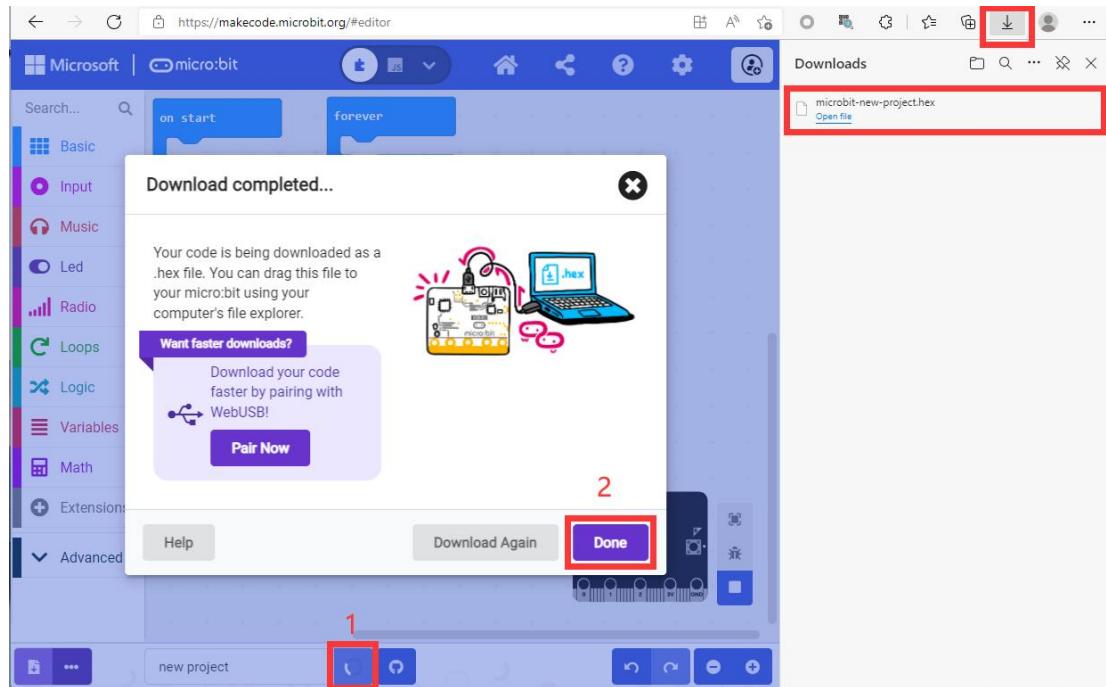


Select the project you want to delete, and click **Delete** button.



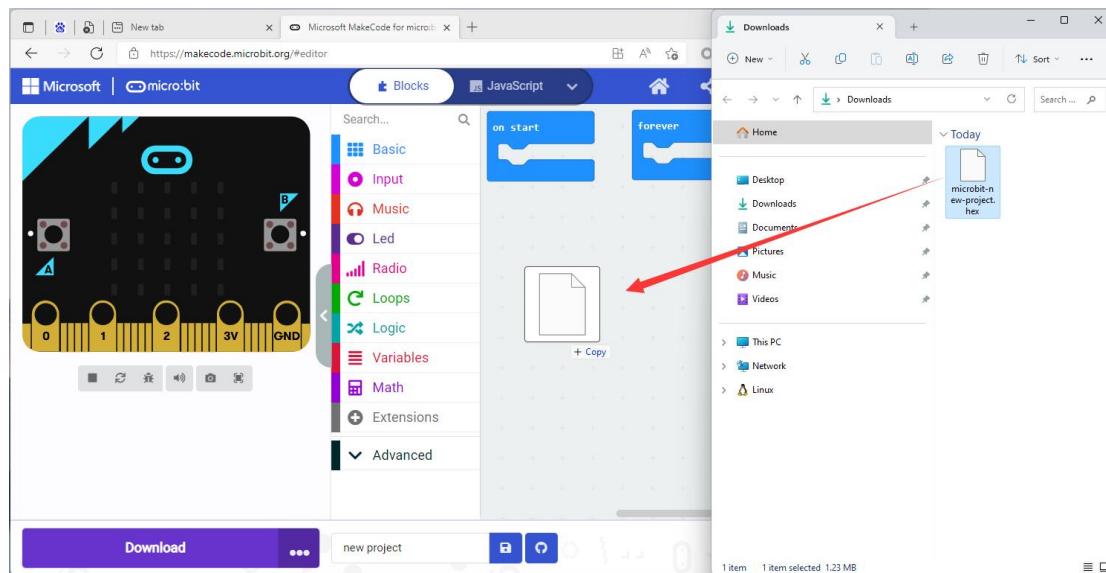
### 3.3 Save a project

Click the "Save" button, and then click the "Done" button to save the project to your computer, as shown below:



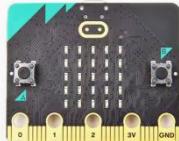
### 3.4 Import Files

Simply drag the local "HEX" project file to the work area of the MakeCode editor, as shown below:

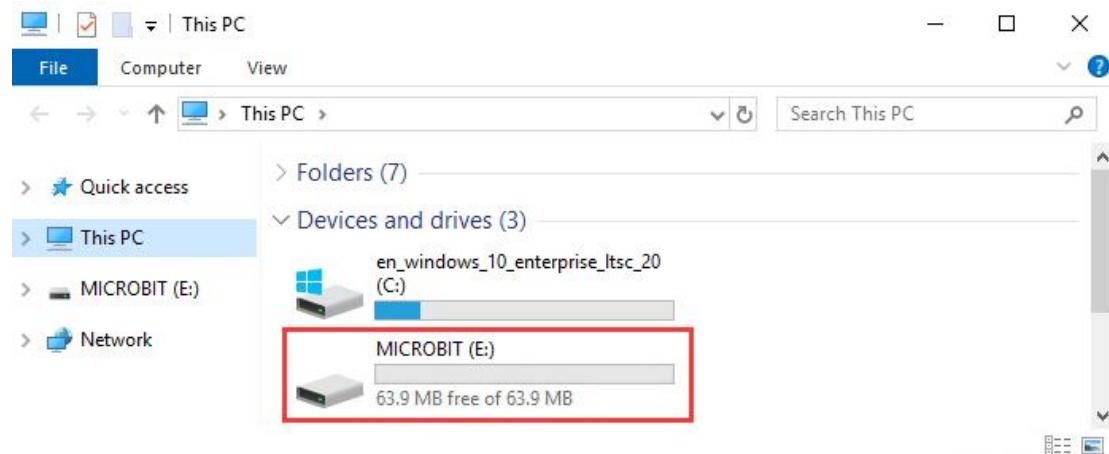
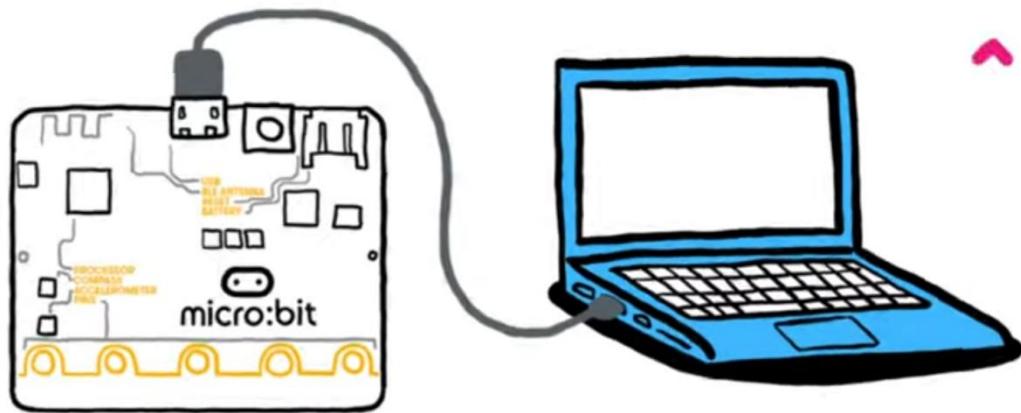


### 3.5 Upload code

Things you need:

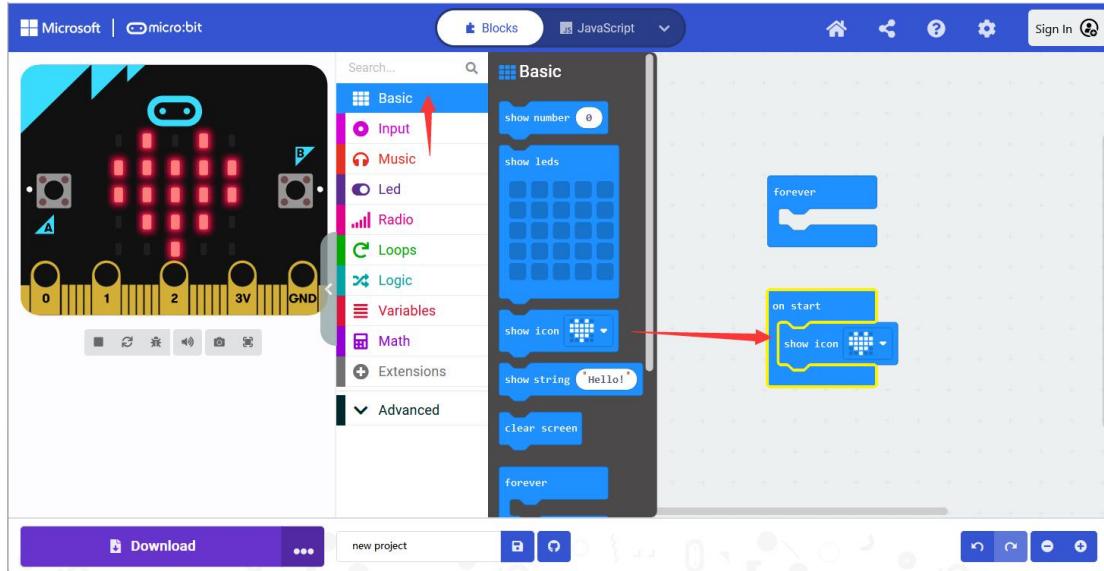
PC	<a href="#">Micro:bit v2.x.x</a>	Micro USB Cable
		

Use a Micro USB cable to connect the micro:bit board to the PC. You will find a new USB disk called **MICROBIT** on the PC:

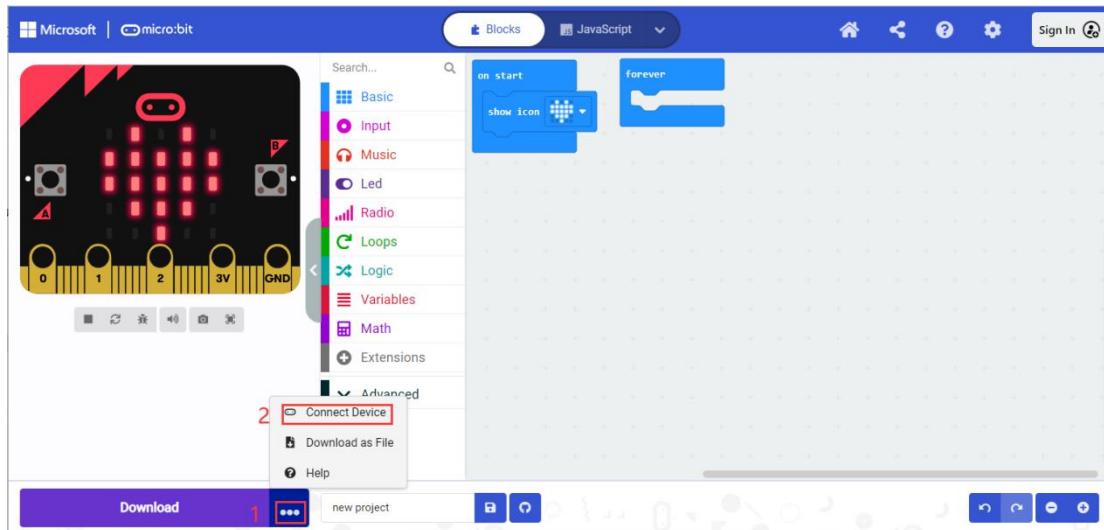


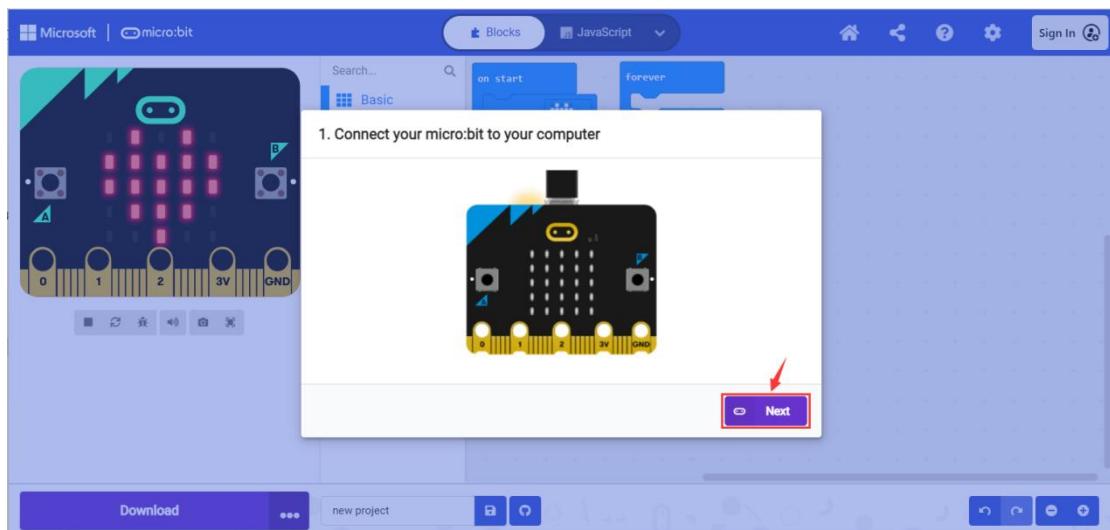
Open the MakeCode editor on your browser, hold down the left mouse button, and drag the **show icon** statement on the left to the working area on the right:

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

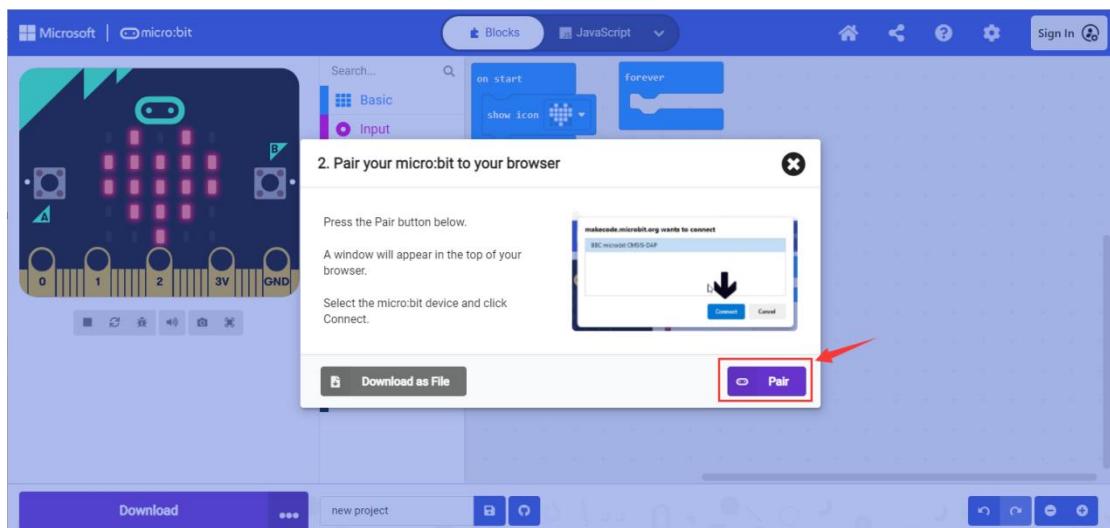


Click on the **three dots** next to the Download button. Then click the "Connect Device", and then click the "Next" , as shown below:

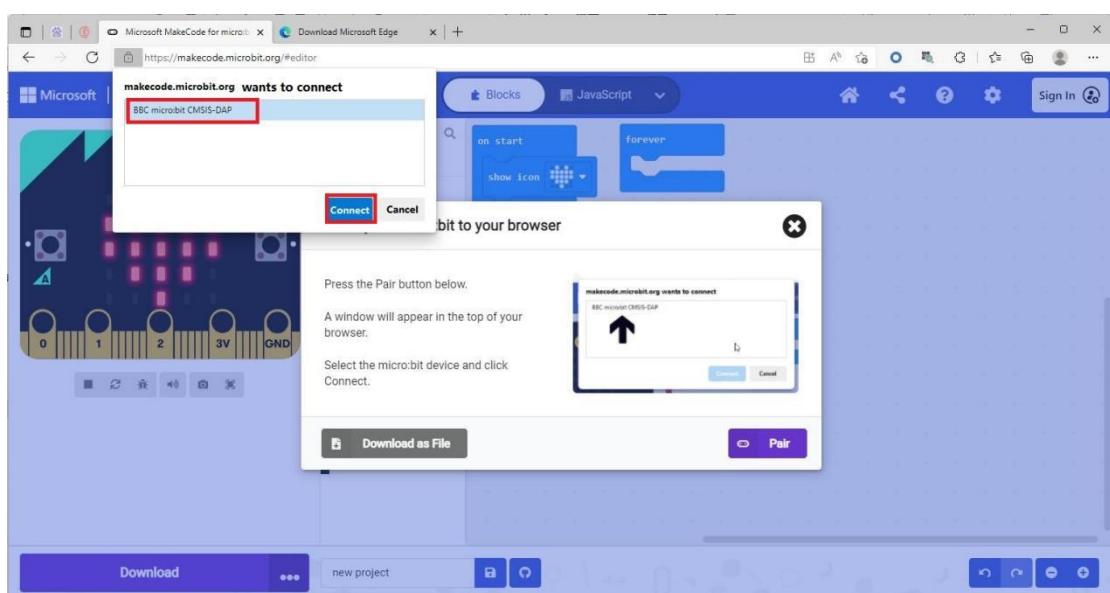




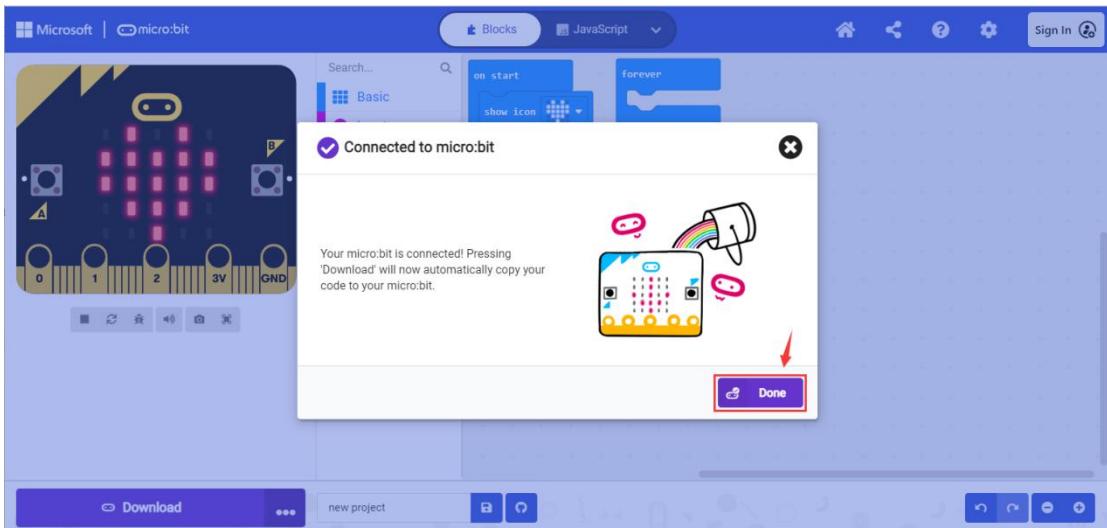
Click the “Pair” button:



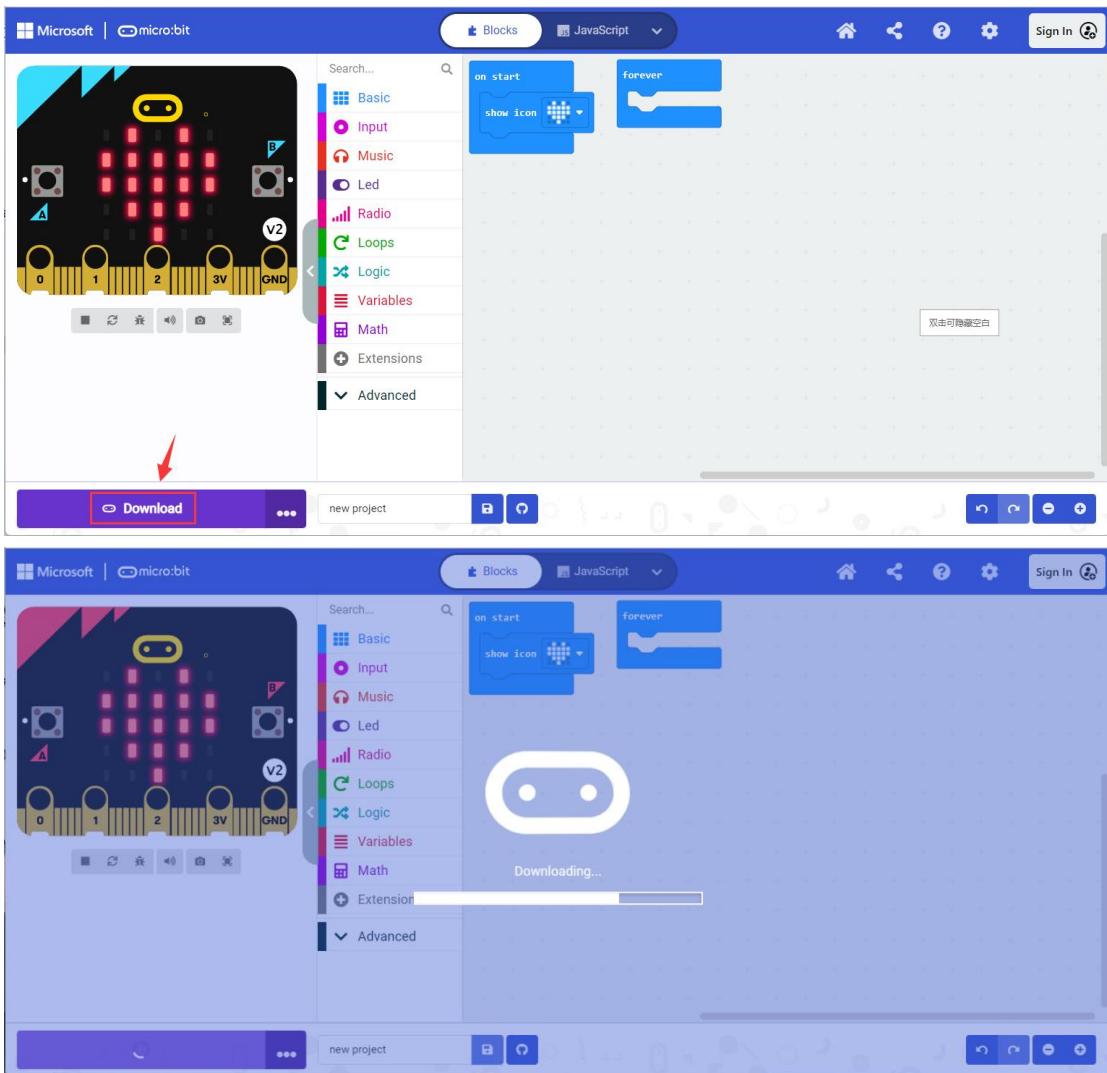
Select the Micro:bit board you want to connect to, and then click "Connect":



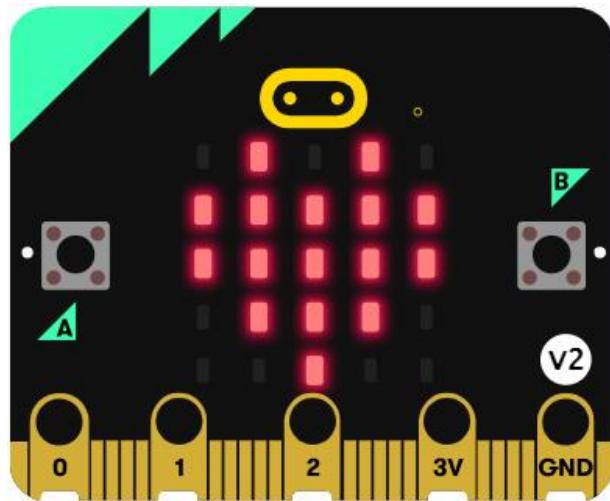
The connection is successful, click “Done”:



Click the "Download" button, you can flash the code to the Micro:bit with WebUSB.



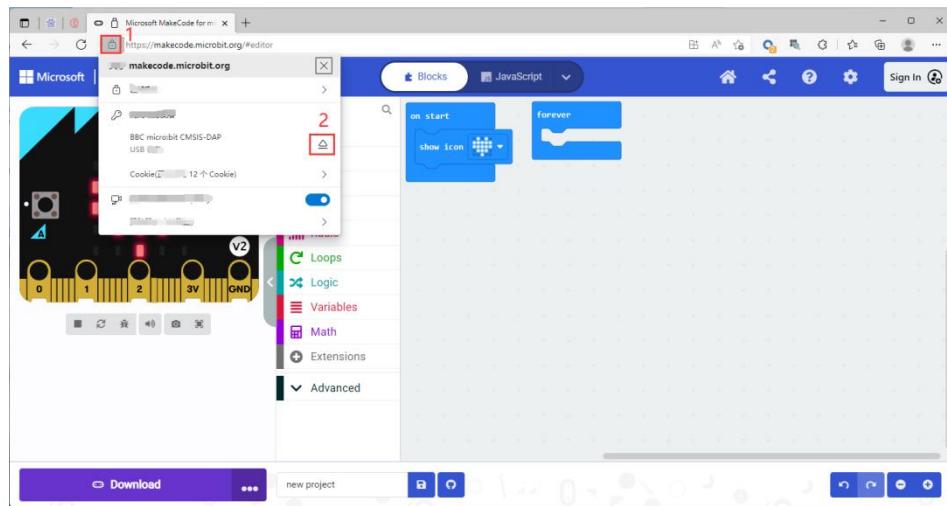
After uploading the code, the dot matrix of the Micro:bit board displays a heart shape:



### Additional Notes (Optional)

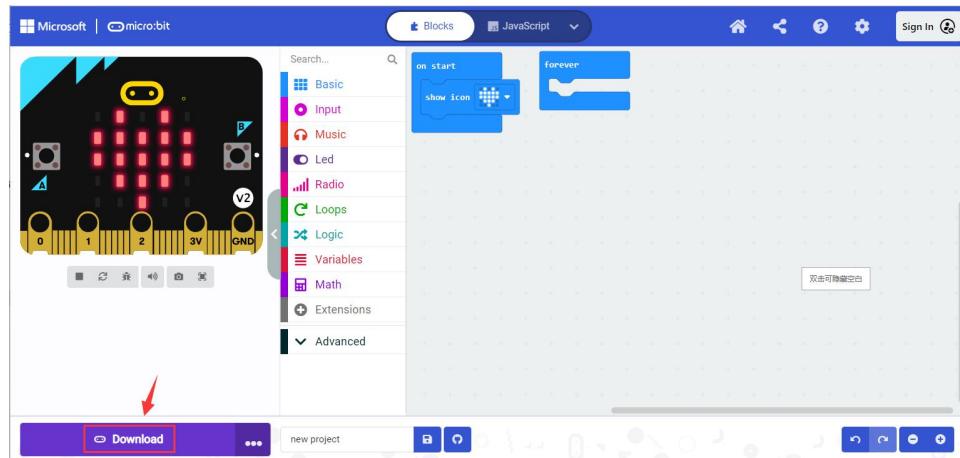
#### 3.6 Unpairing Microbit

- Click the button to the left of your browser's search box.
- Select the Microbit device you want to disconnect and click the button to the right of it.

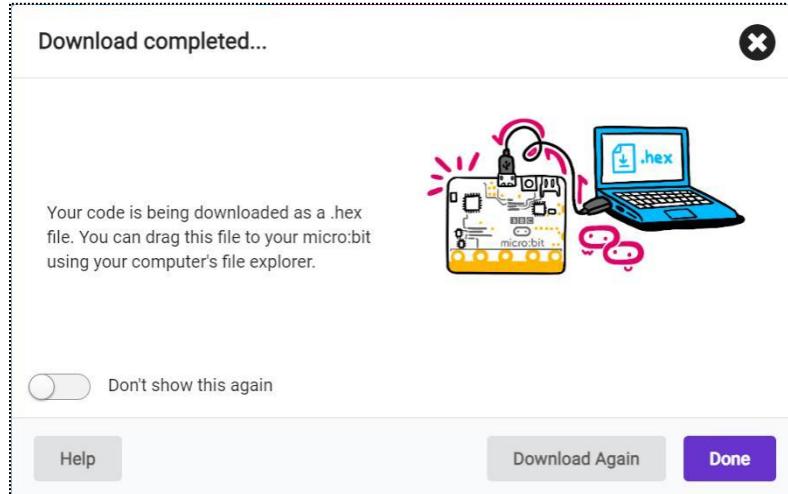


### 3.7 Upload the HEX file to the Micro:bit

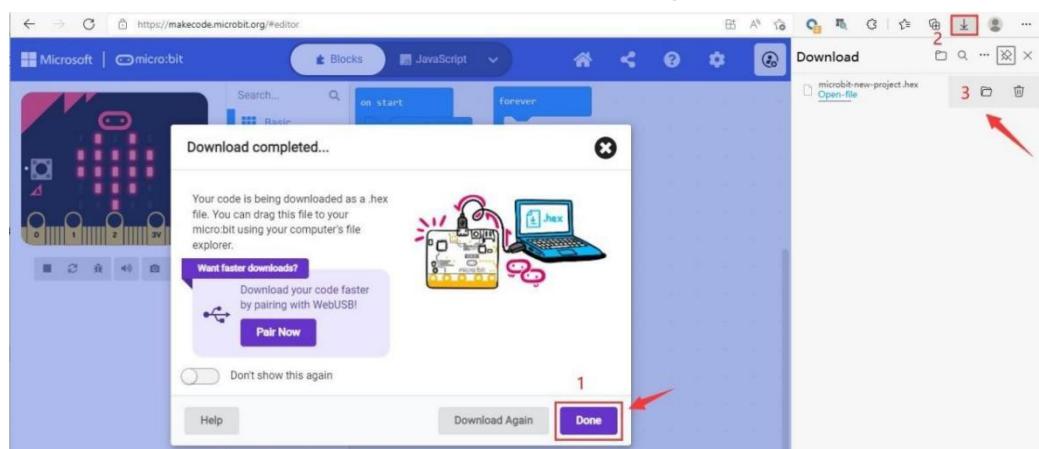
If the Micro:bit is not paired with Microsoft Edge or Google Chrome browser, or if you are using Safari/Firefox/Other browser that may not support WebUSB, directly click the "Download" button, the code won't transfer directly to your microbit, it will be downloaded as a .hexfile. Just like click the save icon to save a copy of the hex file to your computer.



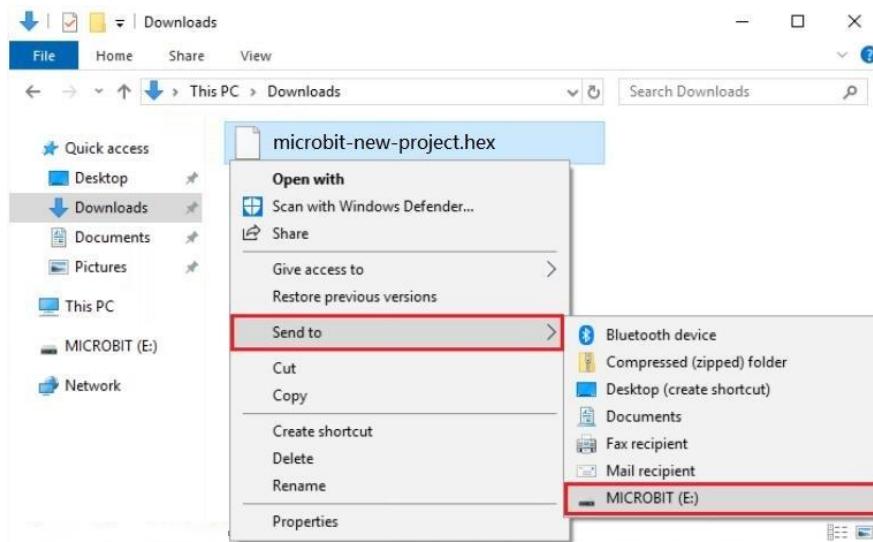
When the following interface appears, click the "X" button and click "Done":



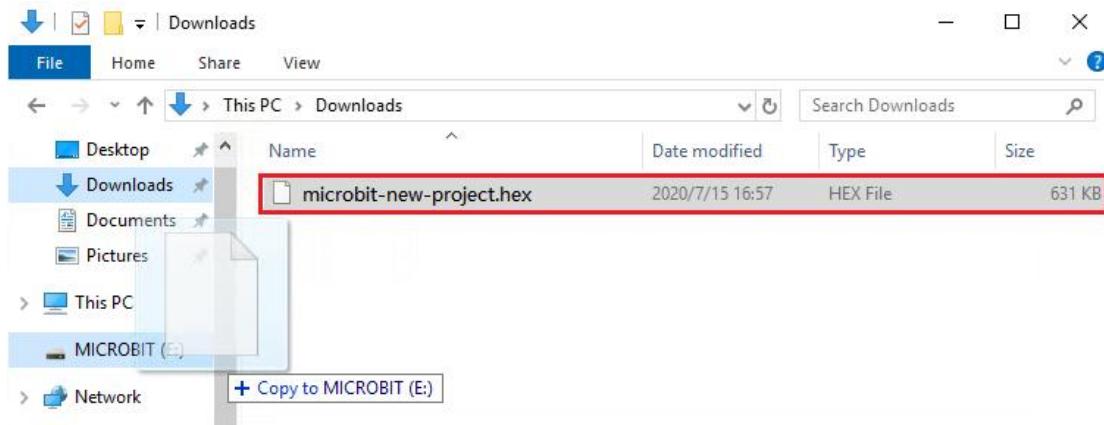
Find the downloaded hex file in the default save path of browser.



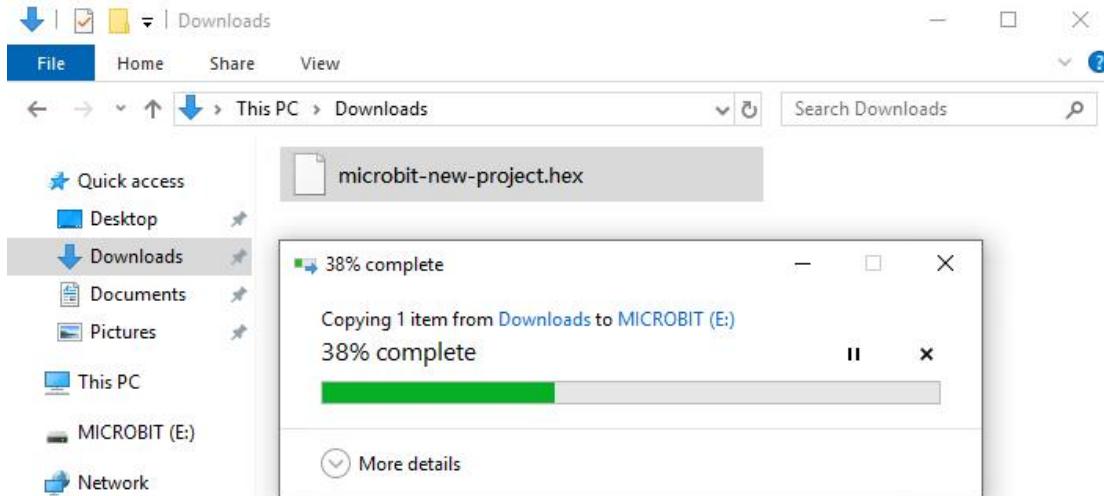
Then select the downloaded hex file, right click the mouse and click "Send to", then you can send the hex file to your Microbit:



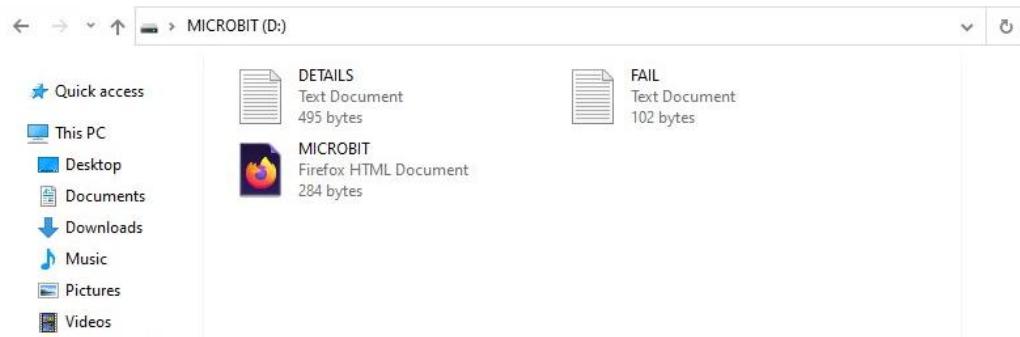
If you do not use the "Send to" button, you can directly drag the hex file to the Micro:bit:



The following interface indicates that the code is being uploaded, at the same time, the yellow LED on the back of the Microbit will also flash rapidly until the code upload is complete.



After the code upload is complete, the Micro Bit will disconnect and reconnect. If you look at the contents of the MICROBIT drive, you will not see the .hex file listed, this is normal, but your hex file will start automatically.



### 3.8 Learn the basic syntax of Makecode

The Micro:bit platform provides official MakeCode API and device usage documents for your reference.

To use APIs: <https://makecode.microbit.org/reference>

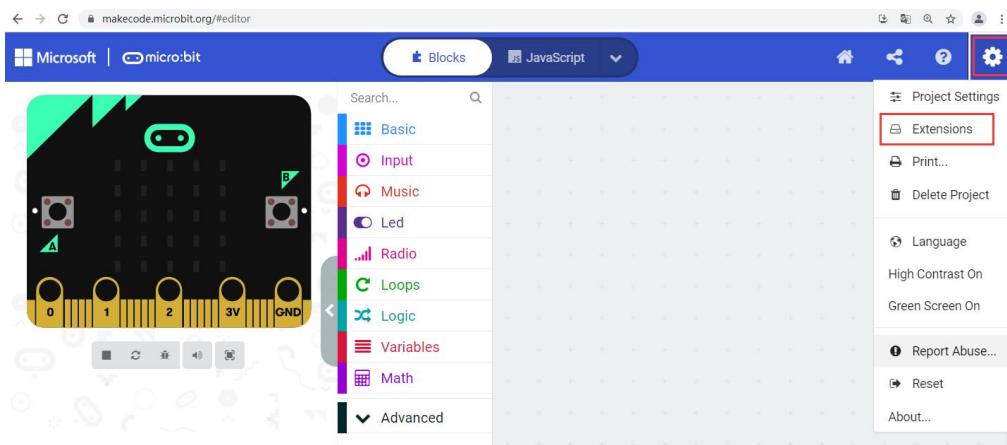
To use device: <https://makecode.microbit.org/device>

Logic and data types: <https://makecode.microbit.org/blocks>

# 4 MakeCode Extension

We have developed an extension for Joystick, which makes it easier for us to use MakeCode to program Joystick.

## 4.1 Add Joystick extension



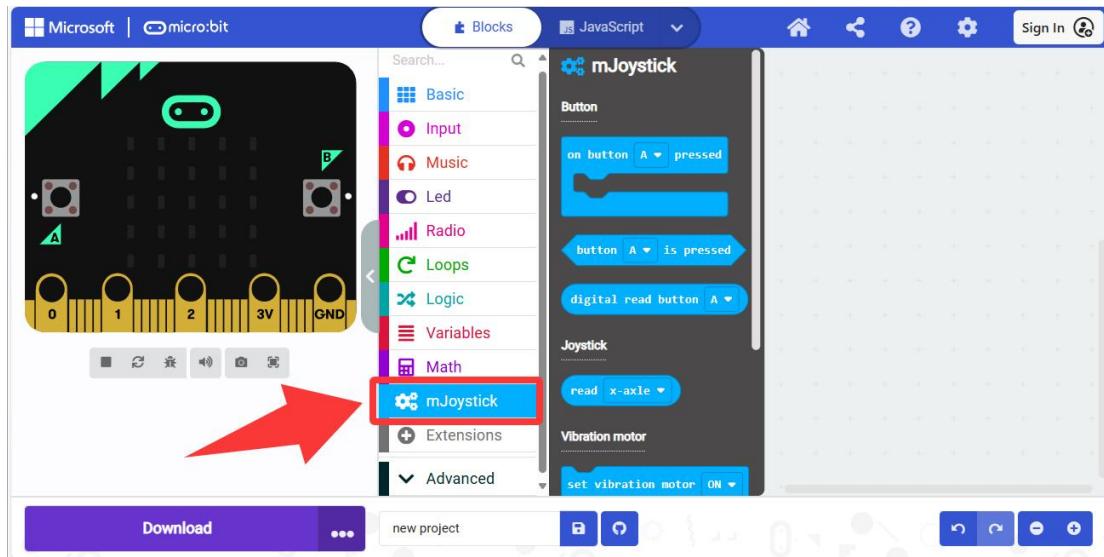
Extension link for Joystick: [https://github.com/siyeenove/pxt\\_mjoystick](https://github.com/siyeenove/pxt_mjoystick)

Copy the above link into the search box on the extension page and click the search button on the right.

Click on the extension named pxt\_Joystick in the search results.

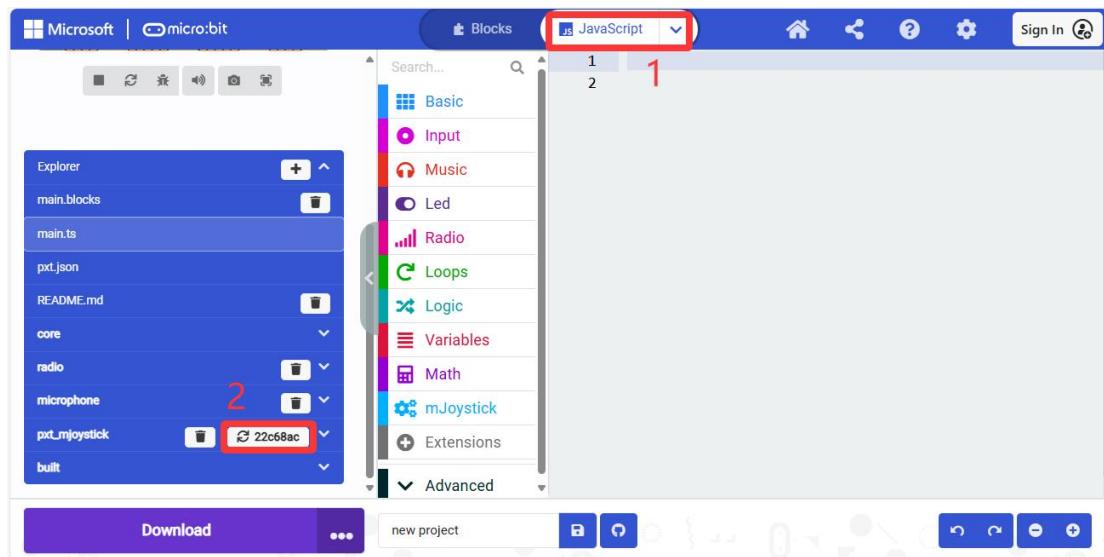
A screenshot of the MakeCode extension search results. At the top, there's a search bar with the URL 'https://github.com/siyeenove/pxt\_mjoystick' and a magnifying glass icon. Below the search bar are category filters: Lights and Display, Software, Science, Robotics, Gaming, and Networking. The main area shows a single extension result: 'pxt\_mjoystick' by Siyeenove. This result is highlighted with a red box. A small red number '3' is visible near the bottom of the extension card. At the bottom of the card, a note says 'User-provided extension, not endorsed by Microsoft.' with a 'Learn More' link. There's also a 'Import File' button at the top right of the extension card.

After a few seconds the page will jump to the Makecode main interface, and you will see the added Joystick extension in the toolbox list.



## 4.2 Refresh Joystick extension

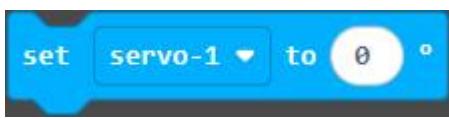
Open the project with the Joystick extension added, and switch to the JavaScript programming interface to refresh the extension:



After refreshing, switch back to the "Blocks" interface.

## 4.3 Parsing of Joystick extension statement

All Makecode statements based on Joystick are integrated in the Joystick extension package. The statement analysis is as follows:

		This is a Micro:bit background thread that can be triggered by pressing Button A, B, C, or D. Each press triggers the execution once.
Buttons		Read the value of Button A, B, C, or D on the joystick. The return value is 0 or 1.
		Detect whether Button A, B, C, or D on the joystick is pressed. The return value is true or false.
		Read the value of the joystick's X-axis and Y-axis. The return value ranges from -100 to +100.
Vibration motor		Start or stop the vibration motor on the joystick. (Shares the P2 pin)
Battery		Read the battery level of the 4 AA batteries. The return value ranges from 0% to 100%. (Shares the P2 pin)
Servo		

# 5 Basic Example Projects

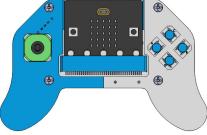
All example codes are saved in the [Code → Basics code folder](#).



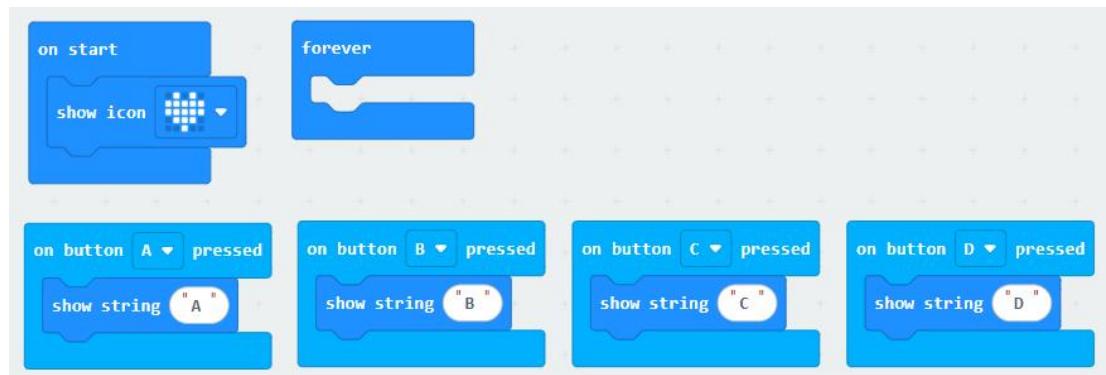
Basics code

## 5.1 Button

Required Hardware:

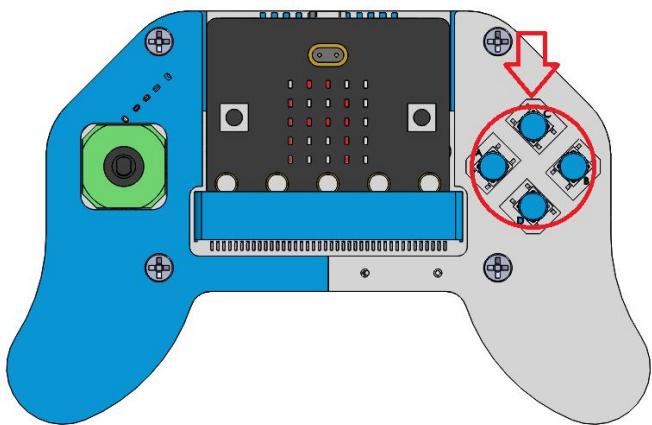
PC	Joystick with micro:bit	Micro USB cable
		

Code:



- When the micro:bit is powered on, the LED matrix displays a heart icon (the “[on start](#)” block is executed once upon power-up).
- The “[forever](#)” block runs in an infinite loop after the micro:bit is powered on; the code above is an empty loop.
- When button A on the joystick is pressed, the “[on button A pressed](#)” block is executed, and the micro:bit LED matrix displays the letter A.
- When button B on the joystick is pressed, the “[on button B pressed](#)” block is executed, and the micro:bit LED matrix displays the letter B.
- When button C on the joystick is pressed, the “[on button C pressed](#)” block is executed, and the micro:bit LED matrix displays the letter C.
- When button D on the joystick is pressed, the “[on button D pressed](#)” block is executed, and the micro:bit LED matrix displays the letter D.

Result:



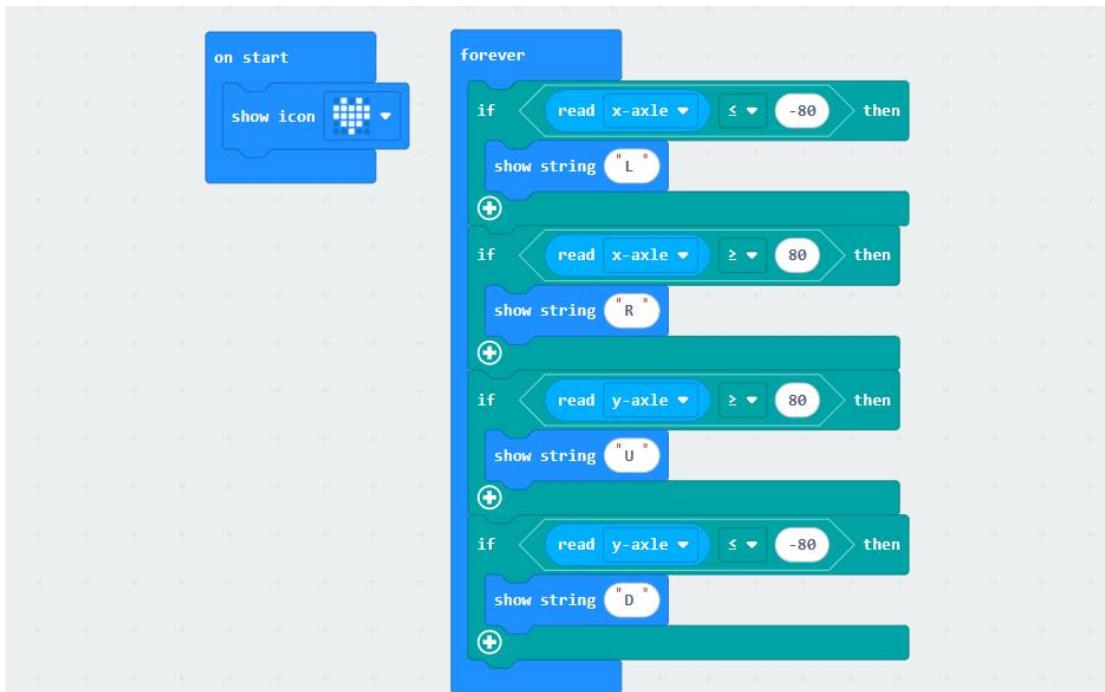
When powered on, the Micro:bit LED matrix displays a heart icon. Pressing button A, B, C, or D on the Joystick will cause the Micro:bit LED matrix to display "A", "B", "C", or "D", respectively.

## 5.2 Joystick

Required Hardware:

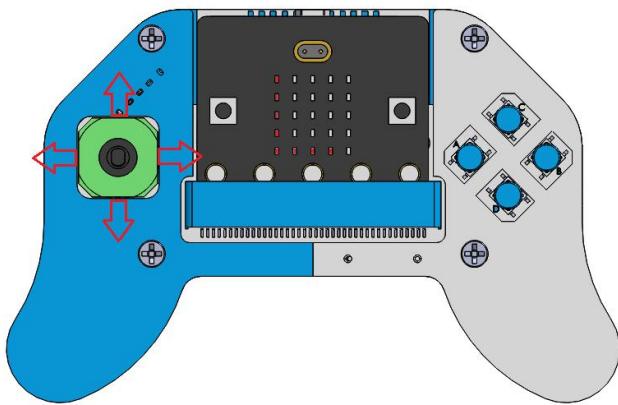
PC	Joystick with micro:bit	Micro USB cable

Code:



- ▶ When the Micro:bit is powered on, the LED matrix displays a heart icon (the “[on start](#)” block is executed once upon power-up).
- ▶ The “[forever](#)” block runs in an infinite loop after the Micro:bit is powered on.
- ▶ Determine whether the value of the joystick’s X-axis is  $\leq -80$ . If true, the Micro:bit LED matrix displays the letter L.
- ▶ Determine whether the value of the joystick’s X-axis is  $\geq 80$ . If true, the Micro:bit LED matrix displays the letter R.
- ▶ Determine whether the value of the joystick’s Y-axis is  $\geq 80$ . If true, the Micro:bit LED matrix displays the letter U.
- ▶ Determine whether the value of the joystick’s Y-axis is  $\leq -80$ . If true, the Micro:bit LED matrix displays the letter D.

Result:



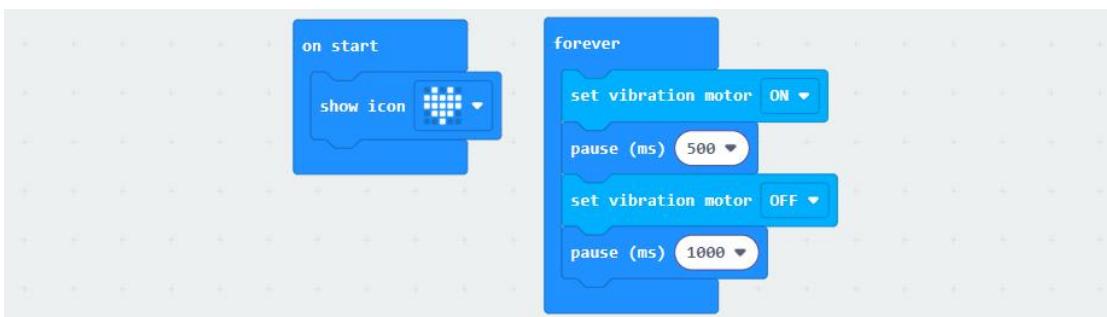
When powered on, the Micro:bit LED matrix displays a heart icon. Pushing the joystick on the Joystick up, down, left, or right will cause the Micro:bit LED matrix to display L, R, U, or D, respectively.

### 5.3 Vibration motor

**Required Hardware:**

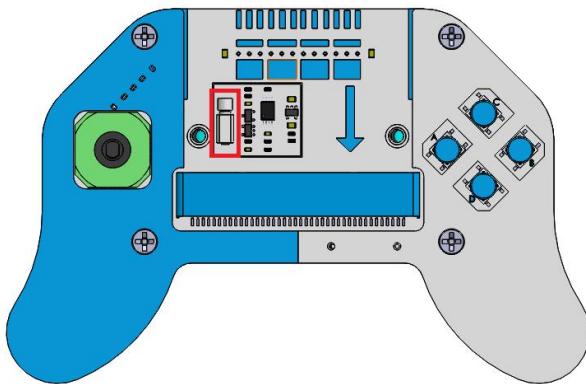
PC	Joystick with micro:bit	Micro USB cable

Code:



- ▶ When the Micro:bit is powered on, the LED matrix displays a heart icon (the "on start" block is executed once upon power-up).
- ▶ The "forever" block runs in an infinite loop after the Micro:bit is powered on.
- ▶ Start the vibration motor of the joystick.
- ▶ Delay for 500 milliseconds.
- ▶ Turn off the vibration motor of the joystick.
- ▶ Delay for 1000 milliseconds.

Result:



When powered on, the Micro:bit LED matrix displays a heart icon, and the vibration motor on the Joystick vibrates for 0.5 seconds every 1 second.

## 5.4 Battery voltage

Required Hardware:

PC	Joystick with micro:bit	Micro USB cable

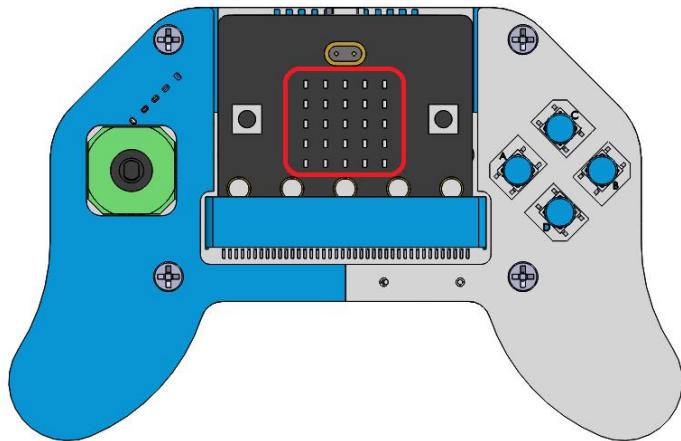
Code:



- ▶ When the Micro:bit is powered on, no statement is executed.
- ▶ The "forever" block runs in an infinite loop after the Micro:bit is powered on.
- ▶ Read the battery level of the 4 AA batteries and display the battery value.
- ▶ Delay for 1000 milliseconds.

### Result:

Note: The remote joystick must be equipped with 4 AA batteries and the power switch must be turned on!



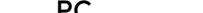
The Micro:bit LED matrix displays the battery level value respectively, within a range of 0-100

### Reminder:

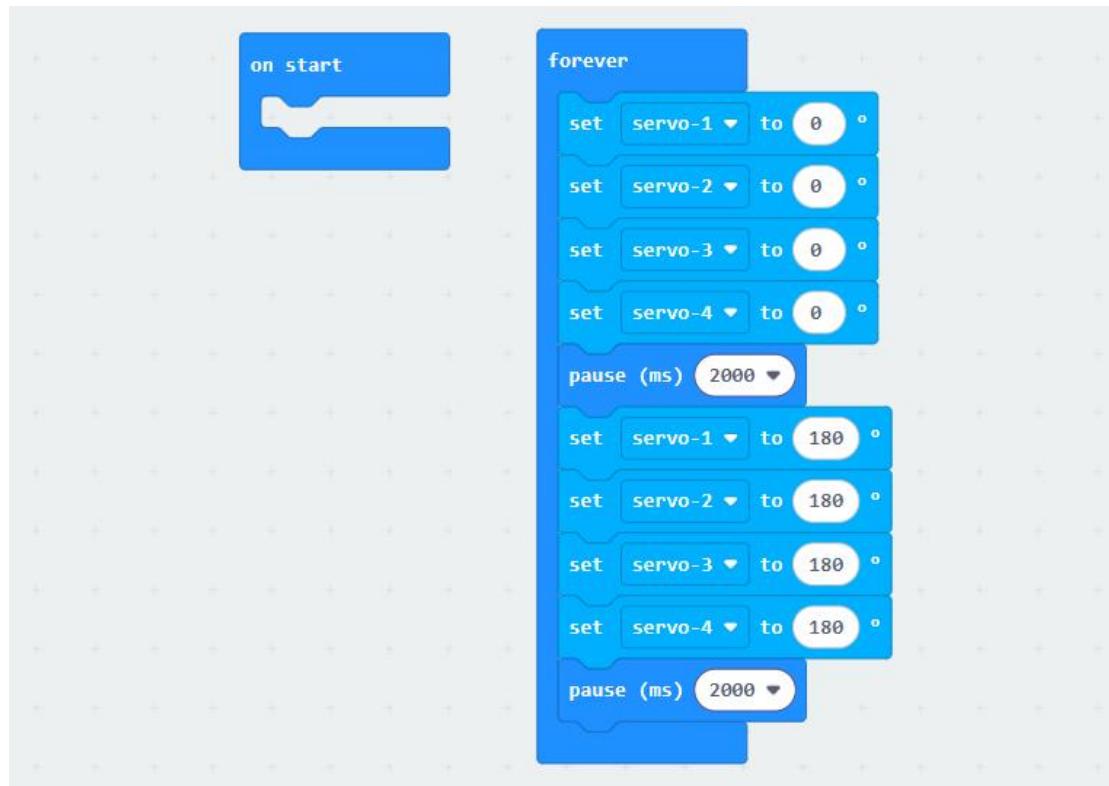
Reading the battery voltage and controlling the vibration motor share the same P2 pin on the Micro:bit. Do not switch too frequently between reading the battery voltage and controlling the vibration motor, as this may affect the accuracy of the battery voltage reading. It is recommended to read the voltage continuously multiple times after controlling the vibration motor and take the last value.

## 5.5 Servo

## **Required Hardware:**

PC	Joystick with micro:bit	Micro USB cable
		

**Code:**



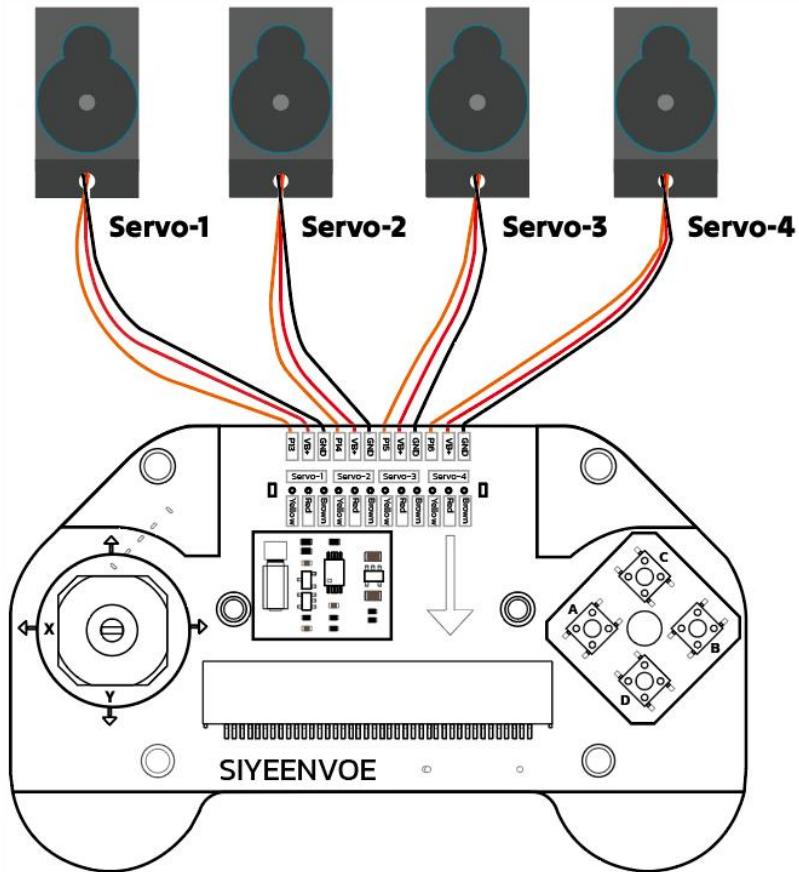
- ▶ When the Micro:bit is powered on, the "on start" block is executed once, displaying a smiley face on the LED matrix.
  - ▶ The "forever" block runs in an infinite loop after the Micro:bit is powered on.
  - ▶ Set the angles of servo-1, servo-2, servo-3, and servo-4 to 0 degrees.
  - ▶ Delay for 2000 milliseconds.
  - ▶ Set the angles of servo-1, servo-2, servo-3, and servo-4 to 180 degrees.
  - ▶ Delay for 2000 milliseconds.

The four servos swing simultaneously to 180 degrees every second in a loop.

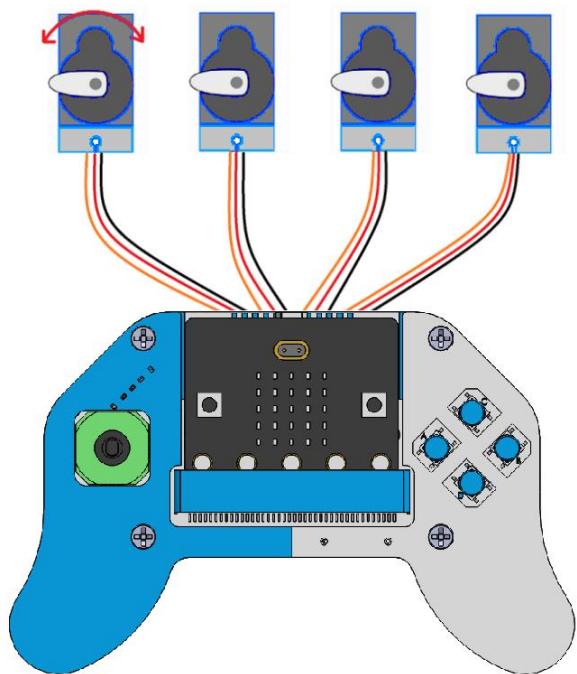
**Note:** The remote joystick must be equipped with 4 AA batteries and the power switch must be turned on!

**Wiring:**

Servo 1			Servo 2			Servo 3			Servo 4		
P13	VB+	GND	P14	VB+	GND	P15	VB+	GND	P16	VB+	GND



**Result:**

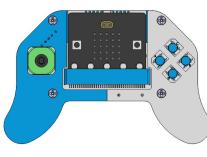


The four servos move synchronously, completing one full cycle every 4 seconds ( $0^\circ \rightarrow$  hold for 2 seconds  $\rightarrow 180^\circ \rightarrow$  hold for 2 seconds  $\rightarrow$  return to  $0^\circ$ ), then return to step 1 and repeat indefinitely in an infinite loop.

**Note:** The remote joystick must be equipped with 4 AA batteries and the power switch must be turned on!

## 5.6 Wireless control

### Required Hardware:

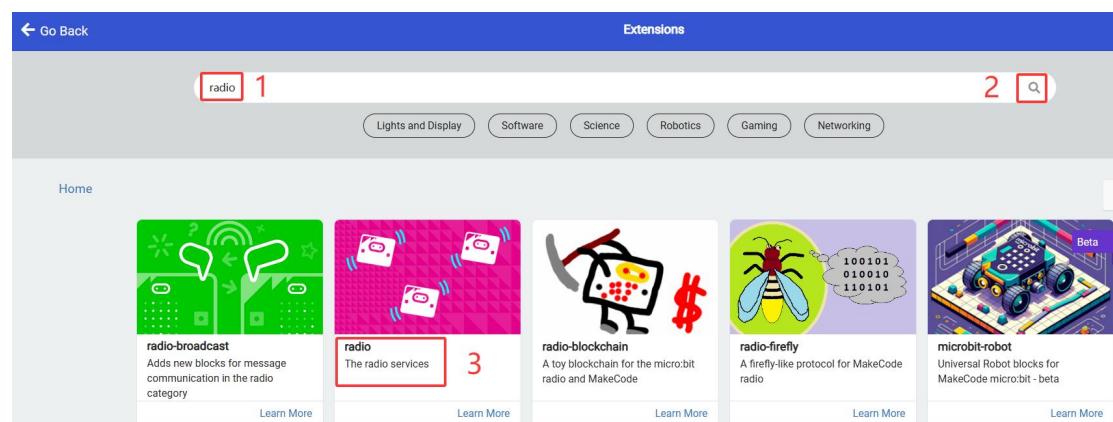
PC	Joystick with Micro:bit V2	Micro USB Cable	Micro:bit V2
			

### Reminder!

If you cannot find "Radio" on the left side of MakeCode:

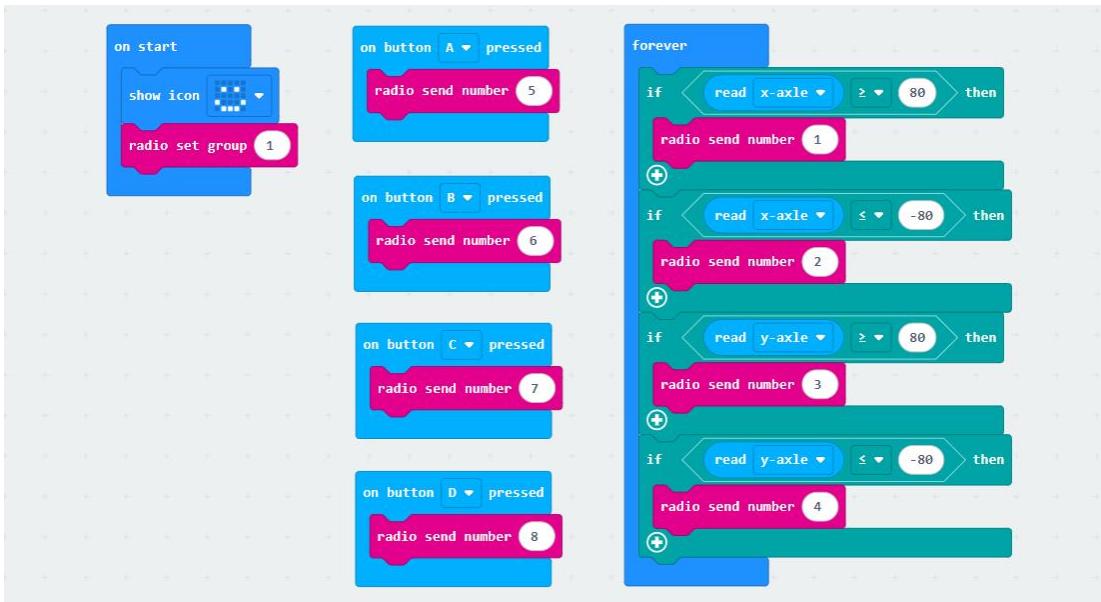


You can add it in "Extensions".

A screenshot of the MakeCode Extensions page. At the top, there is a search bar with the word 'radio' and a magnifying glass icon. Below the search bar, there are several category tabs: Lights and Display, Software, Science, Robotics, Gaming, and Networking. The main area displays five extension cards: 'radio-broadcast' (green card with a radio icon), 'radio' (pink card with a radio icon), 'radio-blockchain' (white card with a blockchain icon), 'radio-firefly' (purple card with a firefly icon), and 'micrombit-robot' (blue card with a robot icon). A large yellow arrow points from the search bar down to the 'radio' extension card.

## Code for the Joystick (Transmitter)

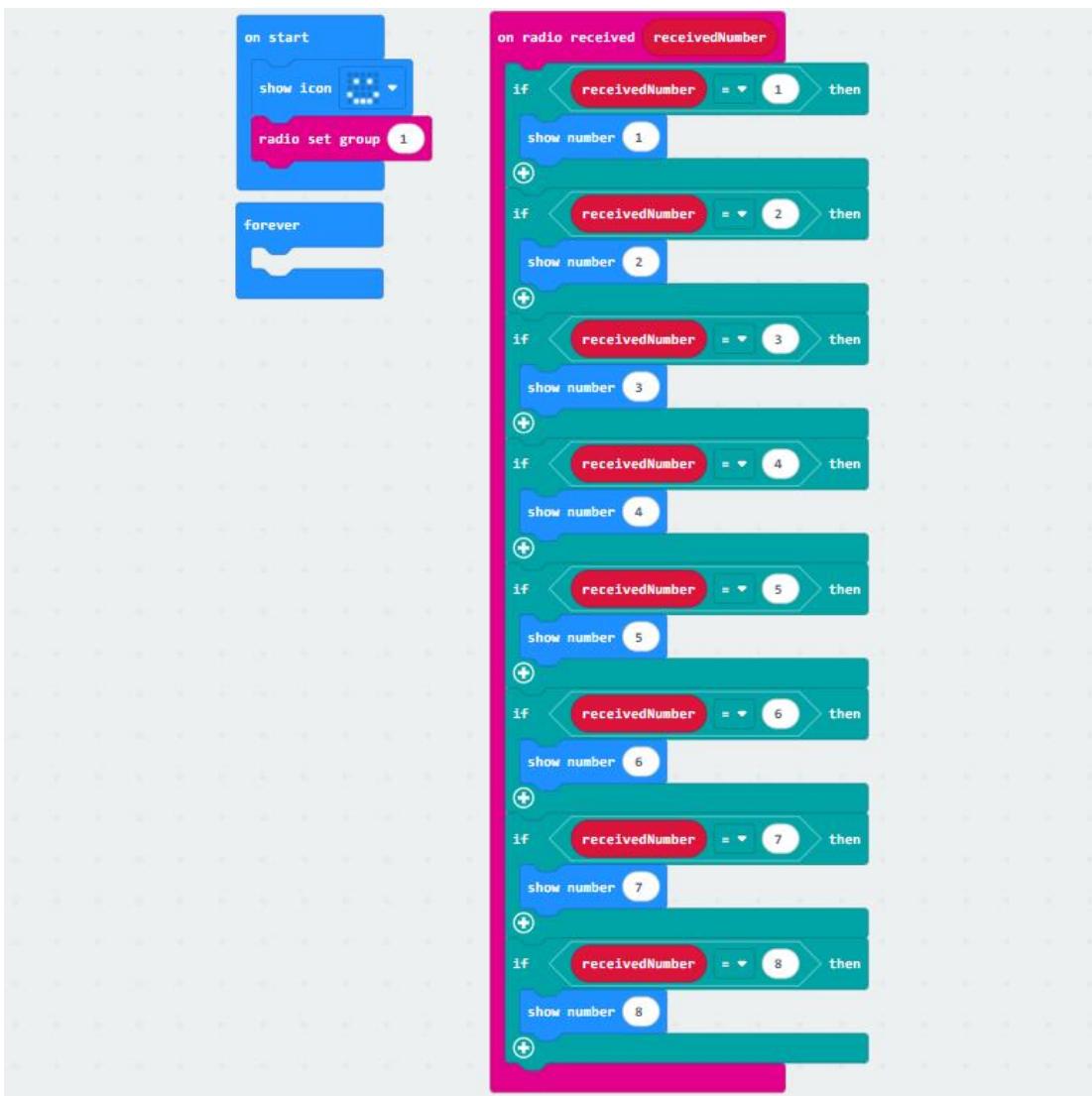
Upload this program to the Micro:bit in the Joystick.



- ▶ When the Micro:bit is powered on, the "on start" block is executed once, displaying a smiley face on the LED matrix and setting the radio group to 1.
- ▶ The "forever" block runs in an infinite loop after the Micro:bit is powered on.
- ▶ If the value of the joystick's X-axis is greater than 80, send the number 1 via radio.
- ▶ If the value of the joystick's X-axis is less than -80, send the number 2 via radio.
- ▶ If the value of the joystick's Y-axis is greater than 80, send the number 3 via radio.
- ▶ If the value of the joystick's Y-axis is less than -80, send the number 4 via radio.
- ▶ Delay for 200 milliseconds.
- ▶ If button A is pressed, send the number 5 via radio.
- ▶ If button B is pressed, send the number 6 via radio.
- ▶ If button C is pressed, send the number 7 via radio.
- ▶ If button D is pressed, send the number 8 via radio.

Upload the code to the Micro:bit in the Joystick.

### Receiver code:

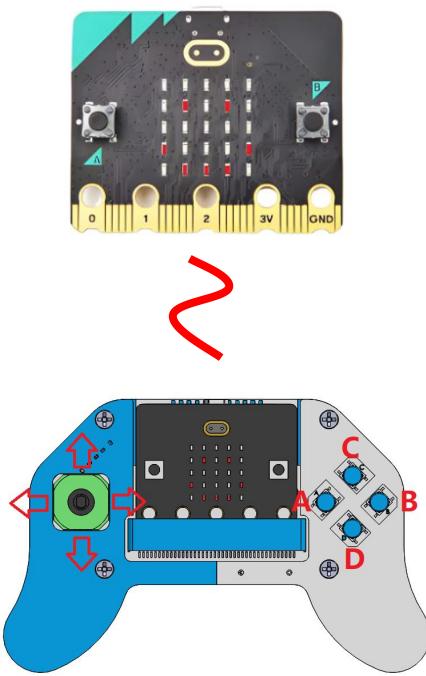


- ▶ When the Micro:bit is powered on, the "**on start**" block is executed once, displaying a smiley face on the LED matrix and setting the radio group to **1**.
- ▶ The "**forever**" block runs in an infinite loop after the Micro:bit is powered on; in this program, it is an empty loop.
- ▶ Continuously read values from radio group 1 in the background.
- ▶ If the received radio value is **1**, display **1** on the LED matrix.
- ▶ If the received radio value is **2**, display **2** on the LED matrix.
- ▶ If the received radio value is **3**, display **3** on the LED matrix.
- ▶ If the received radio value is **4**, display **4** on the LED matrix.
- ▶ If the received radio value is **5**, display **5** on the LED matrix.
- ▶ If the received radio value is **6**, display **6** on the LED matrix.
- ▶ If the received radio value is **7**, display **7** on the LED matrix.
- ▶ If the received radio value is **8**, display **8** on the LED matrix.

Upload the code above to another micro:bit

Result:

Transmitter Action	Receiver Display
Push joystick right	1
Push joystick left	2
Push joystick up	3
Push joystick down	4
Press button A	5
Press button B	6
Press button C	7
Press button D	8



**Note:** The remote joystick must be equipped with 4 AA batteries and the power switch must be turned on!

## 6 Extended Projects

All example codes are saved in the “Code” → “Extended code” folder in the tutorial package.



Extended code

## 6.1 Control mCar

### Required Hardware:

PC	Joystick with Micro:bit V2	mCar M1C0000 with Micro:bit V2	Micro USB Cable
			

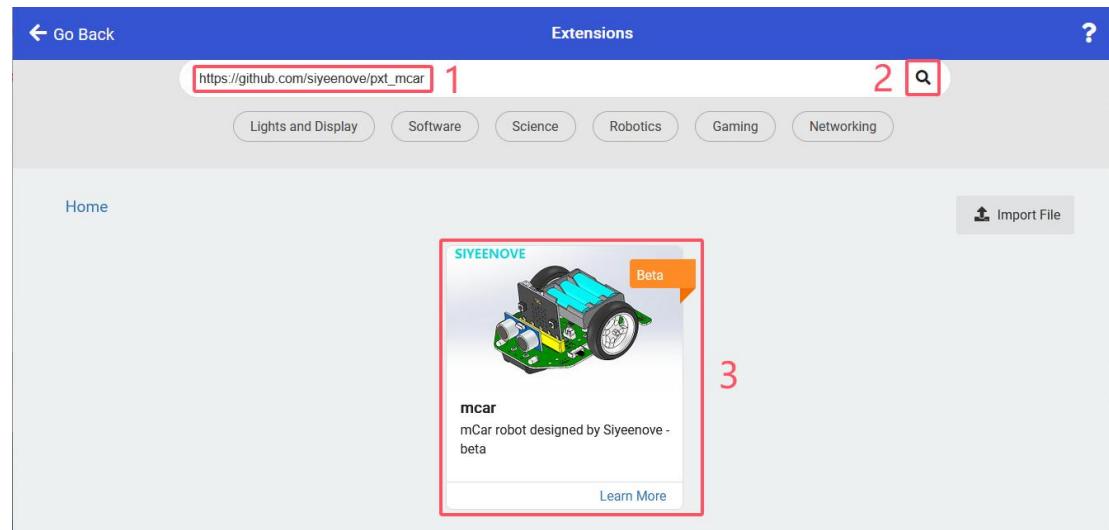
The following section will introduce how to use this Joystick as a controller to operate our mCar robot(**Micro:bit V2 not included**, Model No.: M1C0000). If you are interested, you may contact us to purchase it.

The tutorial for the mCar can be downloaded here:

<https://siyeenove.com/tutorial/>

SKU	Product-Name	View on GitHub	Download from Github	Download from Cloud
A1D0000	Leonardo R3	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>
E1R0000	ESP32 C3 eArm	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>
M1C0000	Micro:bit mCar	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>
M1C0001	Micro:bit Pybit	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>
M1E0001	Micro:bit uShield	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>
M1E0002	Micro:bit mShield	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>
M1K0000	Micro:bit mJoystick	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>
M1R0000	Micro:bit mArm	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>

Add the mCar Extension: [https://github.com/siyeenove/pxt\\_mcar](https://github.com/siyeenove/pxt_mcar)



Extensions

https://github.com/siyeenove/pxt\_mcar 1

2

3

Home

Import File

SIEENOVE Beta

mcar

mCar robot designed by Siyienove - beta

Learn More

## Code for the Joystick (Transmitter)

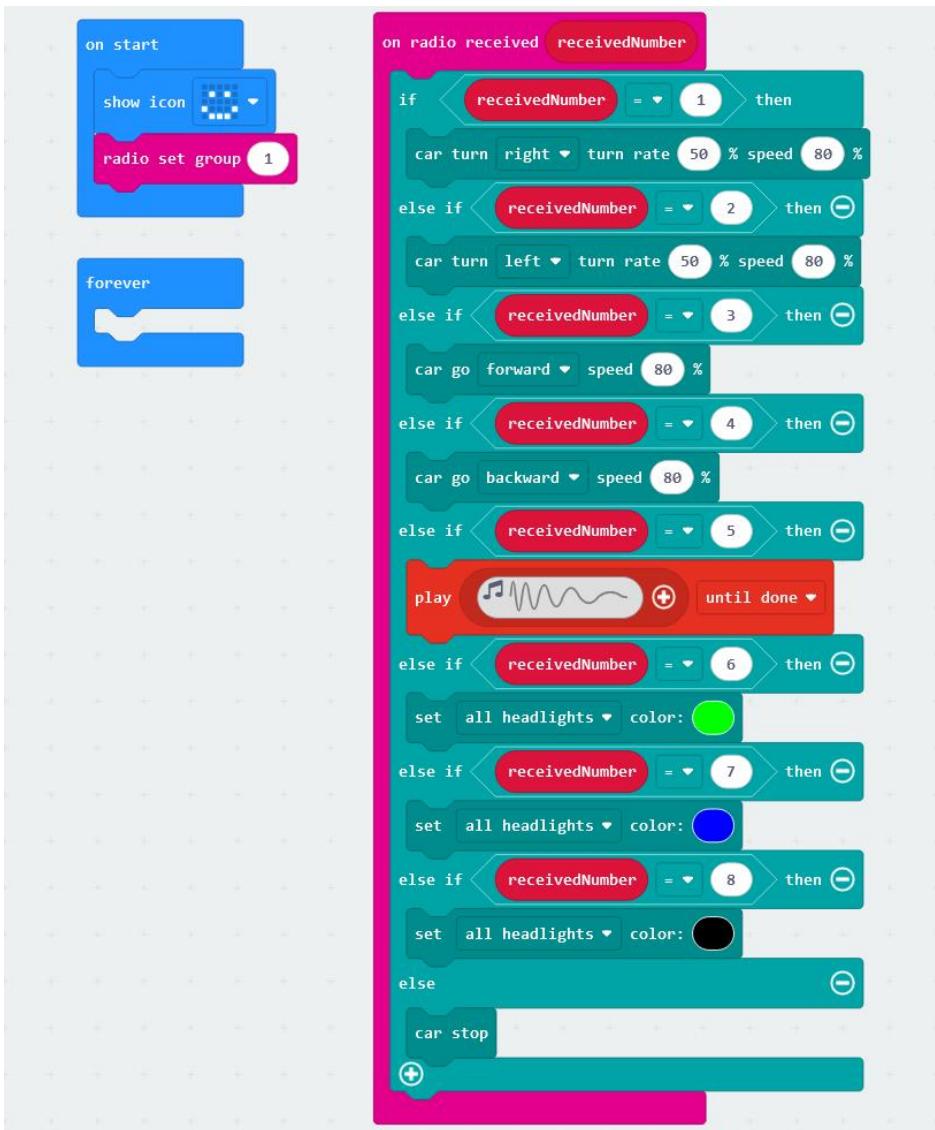
Upload this program to the Micro:bit in the Joystick.



- ▶ When the Micro:bit is powered on, the "on start" block is executed once, displaying a smiley face on the LED matrix and setting the radio group to 1.
- ▶ The "forever" block runs in an infinite loop after the Micro:bit is powered on.
- ▶ If the value of the joystick's X-axis is  $\geq 60$ , send the number 1 via radio.
- ▶ If the value of the joystick's X-axis is  $\leq -60$ , send the number 2 via radio.
- ▶ If the value of the joystick's Y-axis is  $\geq 60$ , send the number 3 via radio.
- ▶ If the value of the joystick's Y-axis is  $\leq -60$ , send the number 4 via radio.
- ▶ If button A is pressed, send the number 5 via radio.
- ▶ If button B is pressed, send the number 6 via radio.
- ▶ If button C is pressed, send the number 7 via radio.
- ▶ If button D is pressed, send the number 8 via radio.
- ▶ If no action is detected, send the number 0 via radio.
- ▶ Delay for 200 milliseconds.

Upload the code to the Micro:bit in the Joystick.

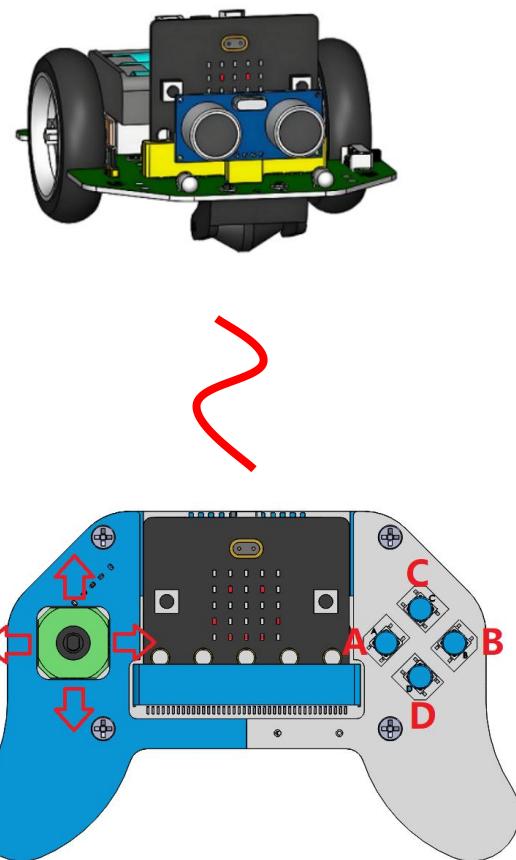
Code for mCar:



- When the Micro:bit is powered on, the "on start" block is executed once, displaying a smiley face on the LED matrix and setting the radio group to 1.
- The "forever" block runs in an infinite loop after the Micro:bit is powered on; in this program, it is an empty loop.
- Continuously read values from radio group 1 in the background.
- If the received radio value is 1, the car turns right at 80% speed with a 50% turn rate.
- If the received radio value is 2, the car turns left at 80% speed with a 50% turn rate.
- If the received radio value is 3, the car moves forward at 80% speed.
- If the received radio value is 4, the car moves backward at 80% speed.
- If the received radio value is 5, the car beeps.
- If the received radio value is 6, the car's headlights turn green.
- If the received radio value is 7, the car's headlights turn blue.

- If the received radio value is 8, the car's headlights turn off.
  - Otherwise, the car stops.
- Upload the code to the Micro:bit in the mCar robot.

**Result:**



Joystick Action	mCar Function
Push joystick up	Car moves forward
Push joystick down	Car moves backward
Push joystick left	Car turns left
Push joystick right	Car turns right
Press button A	Car beeps
Press button B	Headlights turn green
Press button C	Headlights turn blue
Press button D	Headlights turn off
No action	Car stops

**Note:** The joystick must be equipped with 4 AA batteries and the power switch must be turned on!

## 6.2 Control Pybit

### Required Hardware:

PC	Joystick with Micro:bit V2	Pybit M1C0001 with Micro:bit V2	Micro USB Cable
			

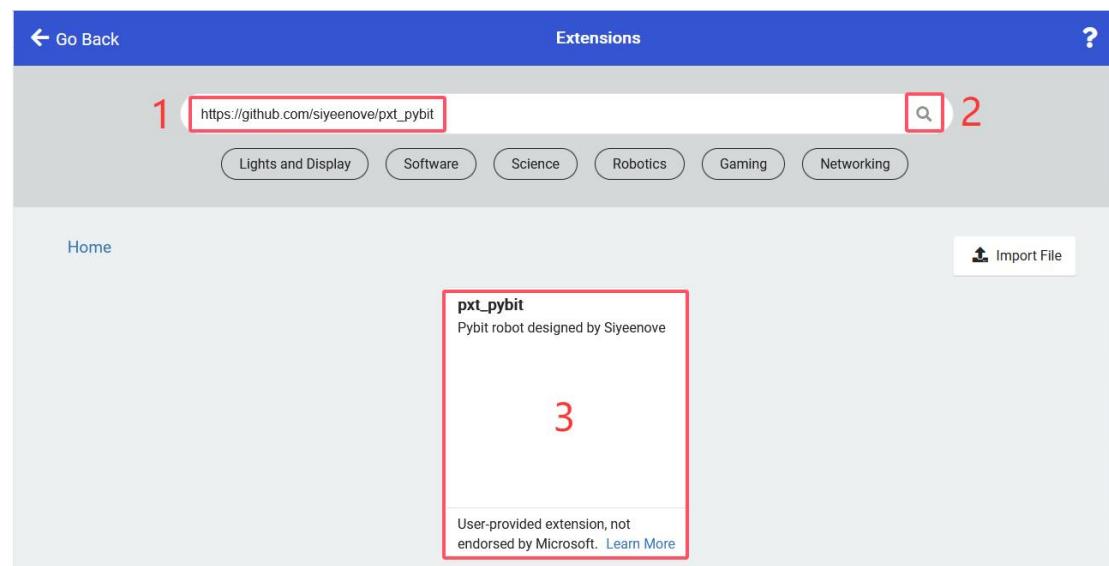
The following section will introduce how to use this Joystick as a controller to operate our Pybit robot(**Micro:bit V2 not included**, Model No.: M1C0001). If you are interested, you may contact us to purchase it.

The tutorial for the Pybit can be downloaded here:

<https://siyeenove.com/tutorial/>

SKU	Product-Name	View on GitHub	Download from Github	Download from Cloud
A1D0000	Leonardo R3	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>
E1R0000	ESP32 C3 eArm	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>
M1C0000	Micro:bit mCar	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>
M1C0001	Micro:bit Pybit	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>
M1E0001	Micro:bit uShield	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>
M1E0002	Micro:bit mShield	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>
M1K0000	Micro:bit mJoystick	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>
M1R0000	Micro:bit mArm	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>

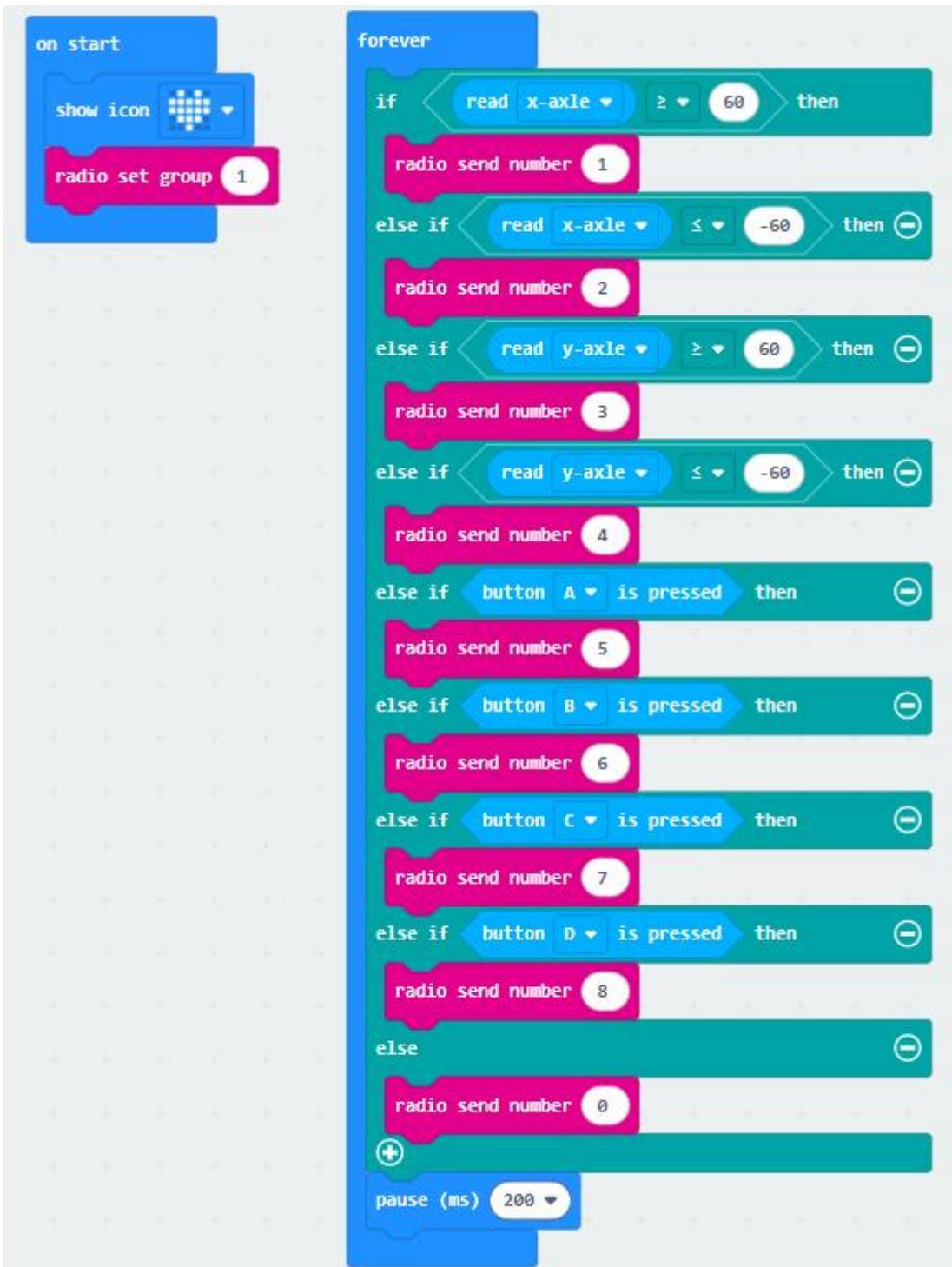
Add the Pybit Extension: [https://github.com/siyeenove/pxt\\_pybit](https://github.com/siyeenove/pxt_pybit)



The screenshot shows the Microsoft MakeCode Extensions page. At the top, there is a search bar with the URL "https://github.com/siyeenove/pxt\_pybit" and a magnifying glass icon. Below the search bar, there are category filters: Lights and Display, Software, Science, Robotics, Gaming, and Networking. In the center, there is a card for the "pxt\_pybit" extension. The card has a red border and contains the text "pxt\_pybit" and "Pybit robot designed by Siyeenove". A large red number "3" is overlaid on the bottom right corner of the card. At the bottom of the card, there is a note: "User-provided extension, not endorsed by Microsoft. [Learn More](#)". On the left side of the card, there is a "Home" link and an "Import File" button with a file icon. On the right side, there is a question mark icon.

## Code for the Joystick (Transmitter)

Upload this program to the Micro:bit in the Joystick.

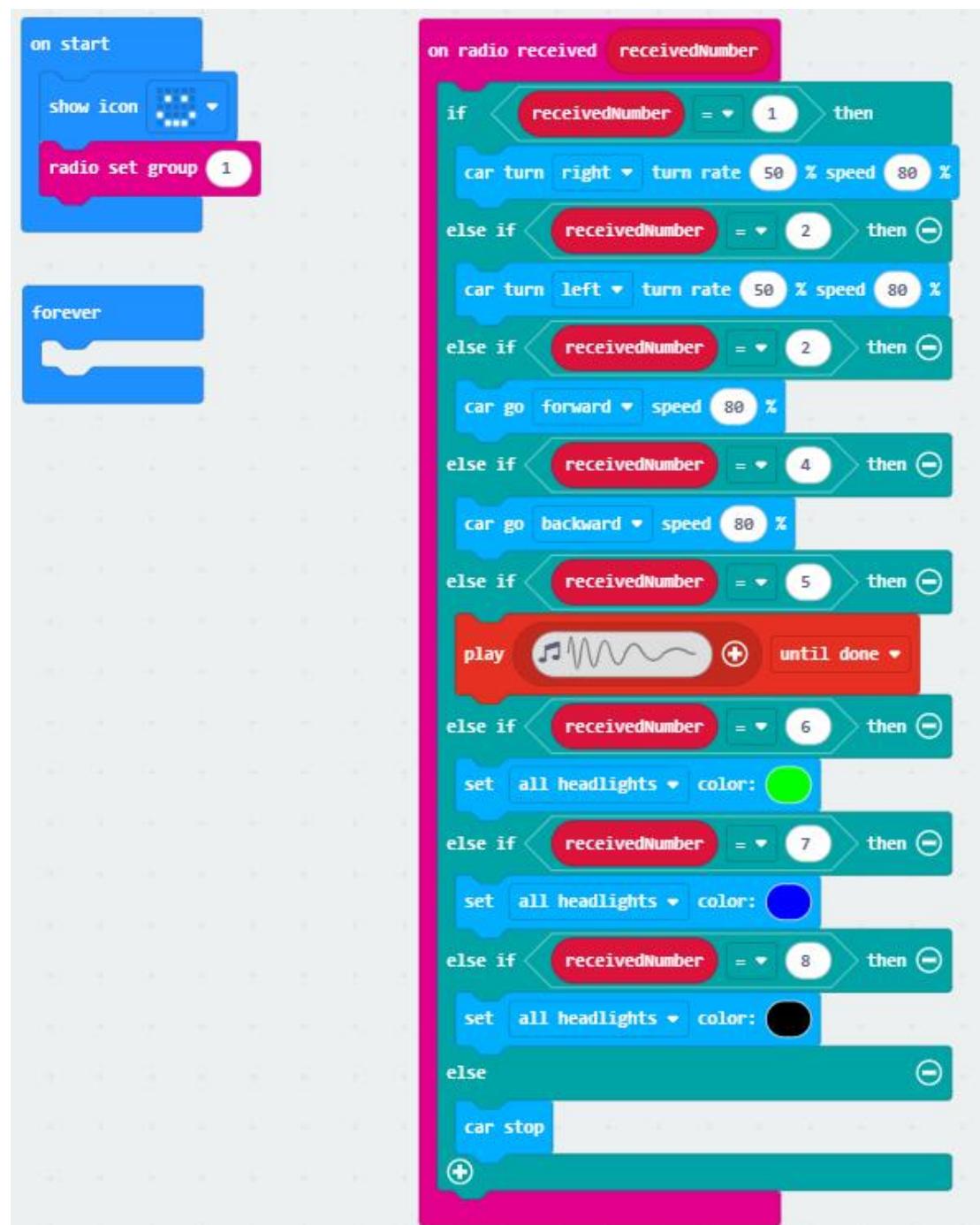


- When the Micro:bit is powered on, the "on start" block is executed once, displaying a smiley face on the LED matrix and setting the radio group to 1.
- The "forever" block runs in an infinite loop after the Micro:bit is powered on.
- If the value of the joystick's X-axis is  $\geq 60$ , send the number 1 via radio.
- If the value of the joystick's X-axis is  $\leq -60$ , send the number 2 via radio.

- If the value of the joystick's Y-axis is  $\geq 60$ , send the number 3 via radio.
- If the value of the joystick's Y-axis is  $\leq -60$ , send the number 4 via radio.
- If button A is pressed, send the number 5 via radio.
- If button B is pressed, send the number 6 via radio.
- If button C is pressed, send the number 7 via radio.
- If button D is pressed, send the number 8 via radio.
- If no action is detected, send the number 0 via radio.
- Delay for 200 milliseconds.

Upload the code to the Micro:bit in the Joystick.

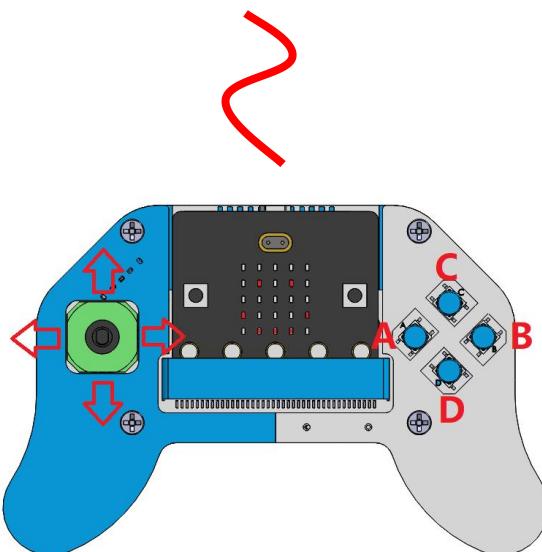
### Code for Pybit:



- ▶ When the Micro:bit is powered on, the "on start" block is executed once, displaying a smiley face on the LED matrix and setting the radio group to 1.
- ▶ The "forever" block runs in an infinite loop after the Micro:bit is powered on; in this program, it is an empty loop.
- ▶ Continuously read values from radio group 1 in the background.
- ▶ If the received radio value is 1, the car turns right at 80% speed with a 50% turn rate.
- ▶ If the received radio value is 2, the car turns left at 80% speed with a 50% turn rate.
- ▶ If the received radio value is 3, the car moves forward at 80% speed.
- ▶ If the received radio value is 4, the car moves backward at 80% speed.
- ▶ If the received radio value is 5, the car beeps.
- ▶ If the received radio value is 6, the car's headlights turn green.
- ▶ If the received radio value is 7, the car's headlights turn blue.
- ▶ If the received radio value is 8, the car's headlights turn off.
- ▶ Otherwise, the car stops.

Upload the code above to one Micro:bit and then insert it into the PyBit.

### **Result:**

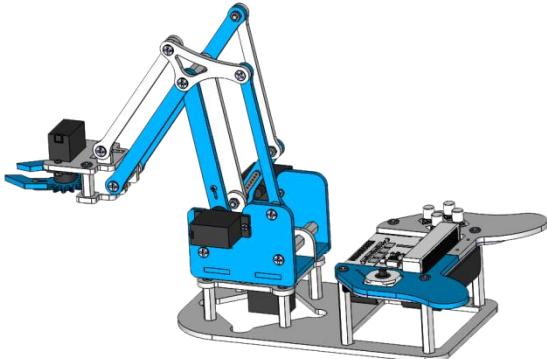


Joystick Action	Pybit Function
Push joystick up	Car moves forward
Push joystick down	Car moves backward
Push joystick left	Car turns left
Push joystick right	Car turns right
Press button A	Car beeps
Press button B	Headlights turn green
Press button C	Headlights turn blue
Press button D	Headlights turn off
No action	Car stops

**Note:** The joystick must be equipped with 4 AA batteries and the power switch must be turned on!

### 6.3 Joystick Control Robot Arm

The Joystick can also be used to control a robotic arm. Here is the mArm 4DF Robotic Arm we developed that includes this joystick (**Micro:bit V2 not included**, Model No.: M1R0000). You can contact us to purchase the mArm kit or the accessories of it excluding the joystick.



The tutorial for the mArm robot arm can be downloaded here:

<https://siyeneove.com/tutorial/>

SKU	Product-Name	View on GitHub	Download from Github	Download from Cloud
A1D0000	Leonardo R3	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>
E1R0000	ESP32 C3 eArm	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>
M1C0000	Micro:bit mCar	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>
M1C0001	Micro:bit Pybit	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>
M1E0001	Micro:bit uShield	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>
M1E0002	Micro:bit mShield	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>
M1K0000	Micro:bit mJoystick	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>
M1R0000	Micro:bit mArm	<a href="#">Go to view</a>	<a href="#">Download</a>	<a href="#">Download</a>

# 7 QA

## 7.1 Micro:bit Code Upload Failure

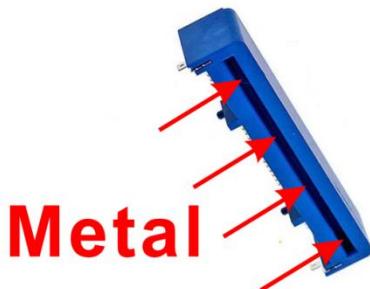
- ☛ Verify USB cable supports data transmission (charge-only cables will not work)
- ☛ Inspect USB cable and ports for physical damage or connectivity issues

## 7.2 Abnormal Micro:bit Drive Presentation

- ☛ Drive labeled "MAINTENANCE" (instead of "MICROBIT") indicates accidental entry to firmware recovery mode. Resolution:
  - ☛ Reflash firmware using official tool:  
<https://microbit.org/get-started/user-guide/firmware/>
  - ☛ Prevent recurrence by avoiding reset button depression during power cycling

## 7.3 Buttons or Joystick Not Responding

- ☛ Ensure the Micro:bit is fully inserted into the mJoystick slot.
- ☛ Ensure the edge connector of the Micro:bit is clean.
- ☛ Ensure the mJoystick slot is clean



## 7.4 Cannot Find “Radio”



You can add it in “Extensions”:



Since Bluetooth and Radio cannot run simultaneously on the micro:bit in MakeCode, adding either one will automatically remove the other from MakeCode.

## 7.5 Other Issues

- ☛ Please check whether the batteries have sufficient power.
- ☛ Please check whether the batteries used meet the required specifications.

# 8. Contact Us

If you couldn't find a solution above, please contact our support team.

**To help us assist you quickly, please have the following information ready:**

Your order number.

Product model (e.g., Robot Arm Kit M1R0000) and software version

A detailed description of the issue or your question.

Steps you have already tried.

Any relevant error message screenshots, photos, or code snippets.

### Other Inquiries?

Please feel free to reach out for:

**Tutorial Errors & Feedback:** Help us make our documentation better.

**Product Ideas & Suggestions:** We'd love to hear your great ideas.

**Partnerships & Collaboration:** Interested in working together? Let's talk.

**Discounts & Promotions:** Inquire about educational or bulk pricing.

**Anything Else:** For all other non-technical questions.

 [support@siyeenove.com](mailto:support@siyeenove.com)

 <http://siyeenove.com>