



# BBC Micro Bit Game Console User Manual

[Home](#) » [BBC](#) » BBC Micro Bit Game Console User Manual 

## Contents

- [1 BBC Micro Bit Game Console](#)
- [2 Product Information](#)
- [3 Frequently Asked Questions \(FAQ\)](#)
- [4 Initialization](#)
- [5 Button monitoring](#)
- [6 Monitoring the joystick](#)
- [7 Controlling the buzzer](#)
- [8 Verifying Demo](#)
- [9 Documents / Resources](#)
  - [9.1 References](#)



**BBC Micro Bit Game Console**



## Product Information

### Specifications

- **Product:** BBC Micro Bit Game Console
- **Website:** <https://makecode.microbit.org/#>
- **Programming Language:** TypeScript
- **Buzzer Control:** Two ways – using provided blocks or micro: bit's music library.

### First upload to Makecode, then download:

If you want to use Micro Python, you can either use the official programming website or download the programming tool Mu.

- Online programming website: <https://codewith.mu/#download>
- Offline programming software: <https://codewith.mu/#download> (also available for download on the resources part of this page)

In the program, you can see the following methods implemented:

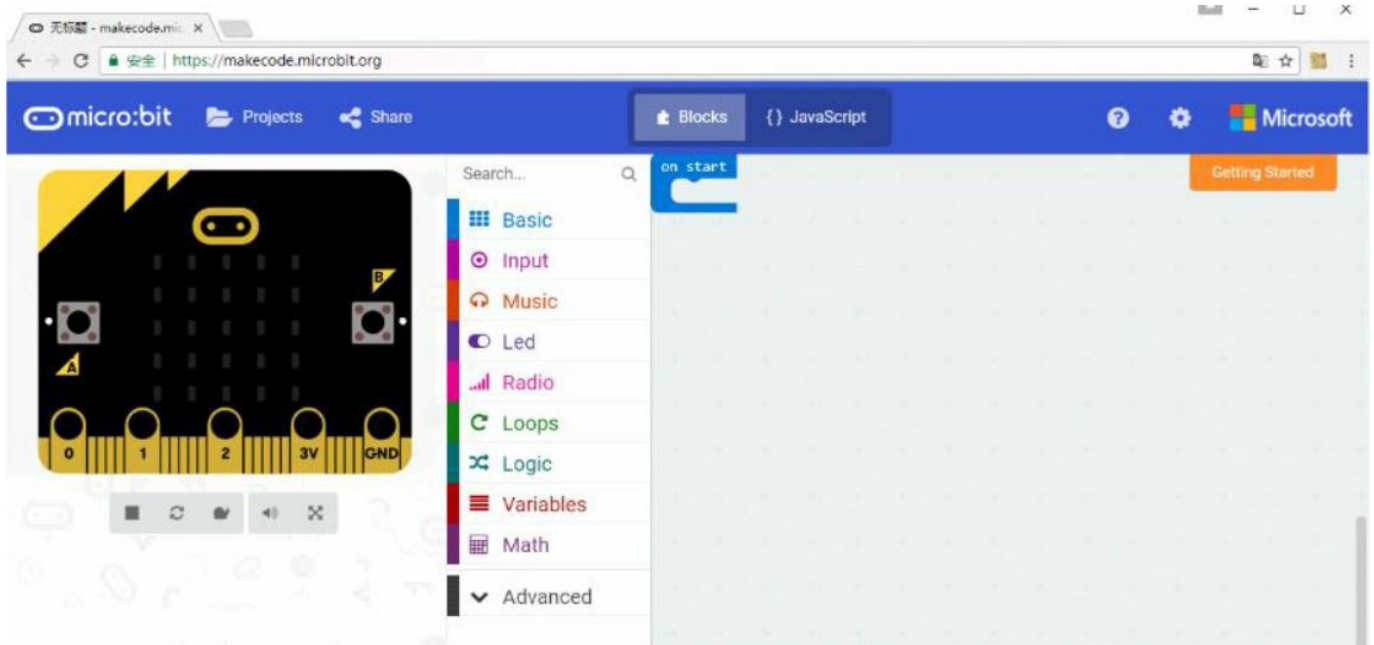
- No initialization is needed when using Micro Python, as it is done during instantiation.
- `Listen_Dir(Dir)`: Monitor the direction of the joystick.
- `Listen_Key(Key)`: Monitor keys.
- `PlayScale(freq)`: Play the sound of a user-defined note.
- `Playmusic(tune)`: Play music/melody.

## Frequently Asked Questions (FAQ)

- **Q:** Where can I find the user manual for the BBC Micro Bit Game Console?

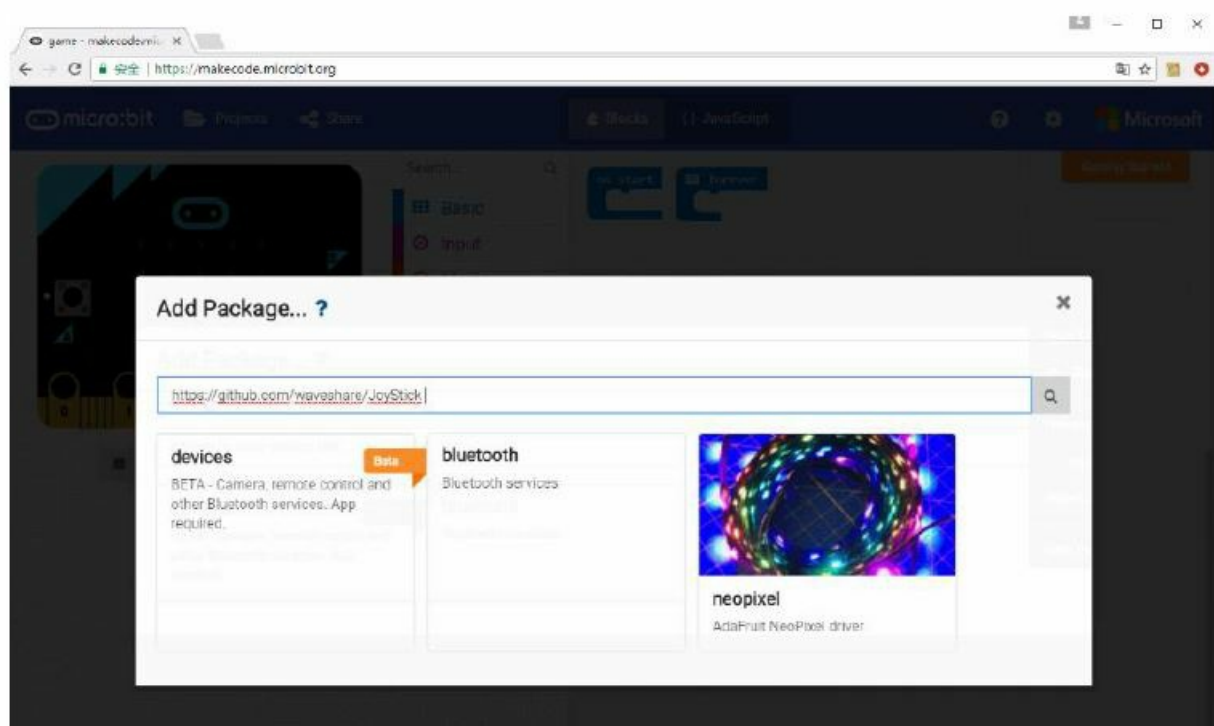
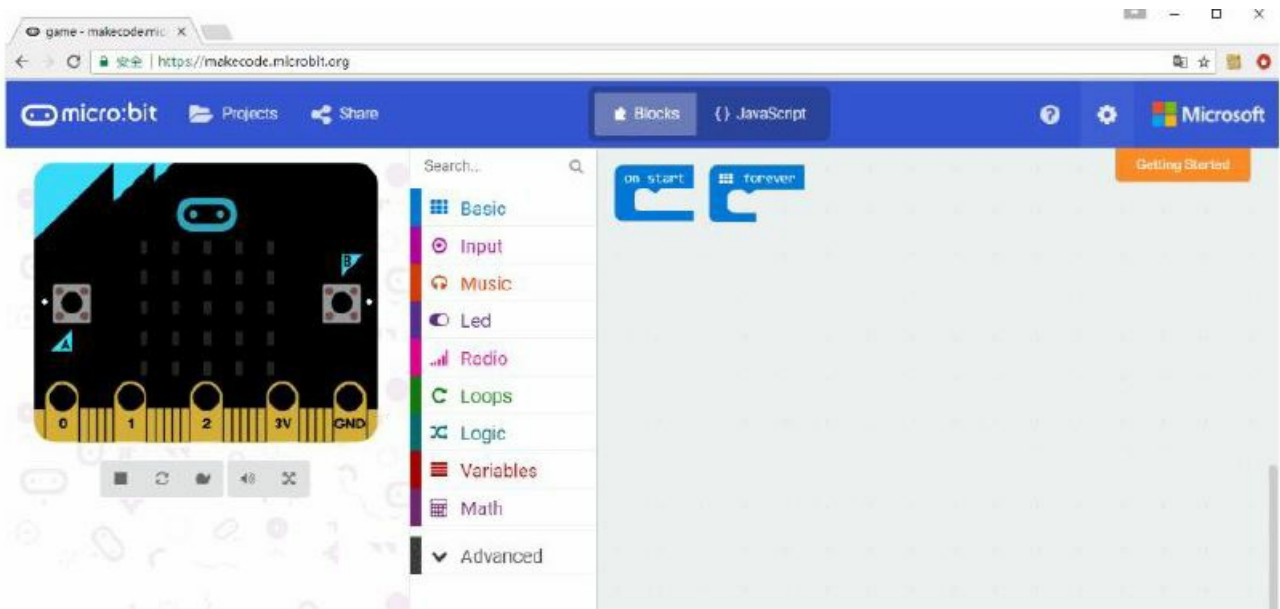
- **A:** The user manual can be found at <https://makecode.microbit.org/#>.
- **Q:** Can I use other blocks besides the ones mentioned in the user manual?
- **A:** Yes, you can explore additional blocks on the programming website or software mentioned in the manual.

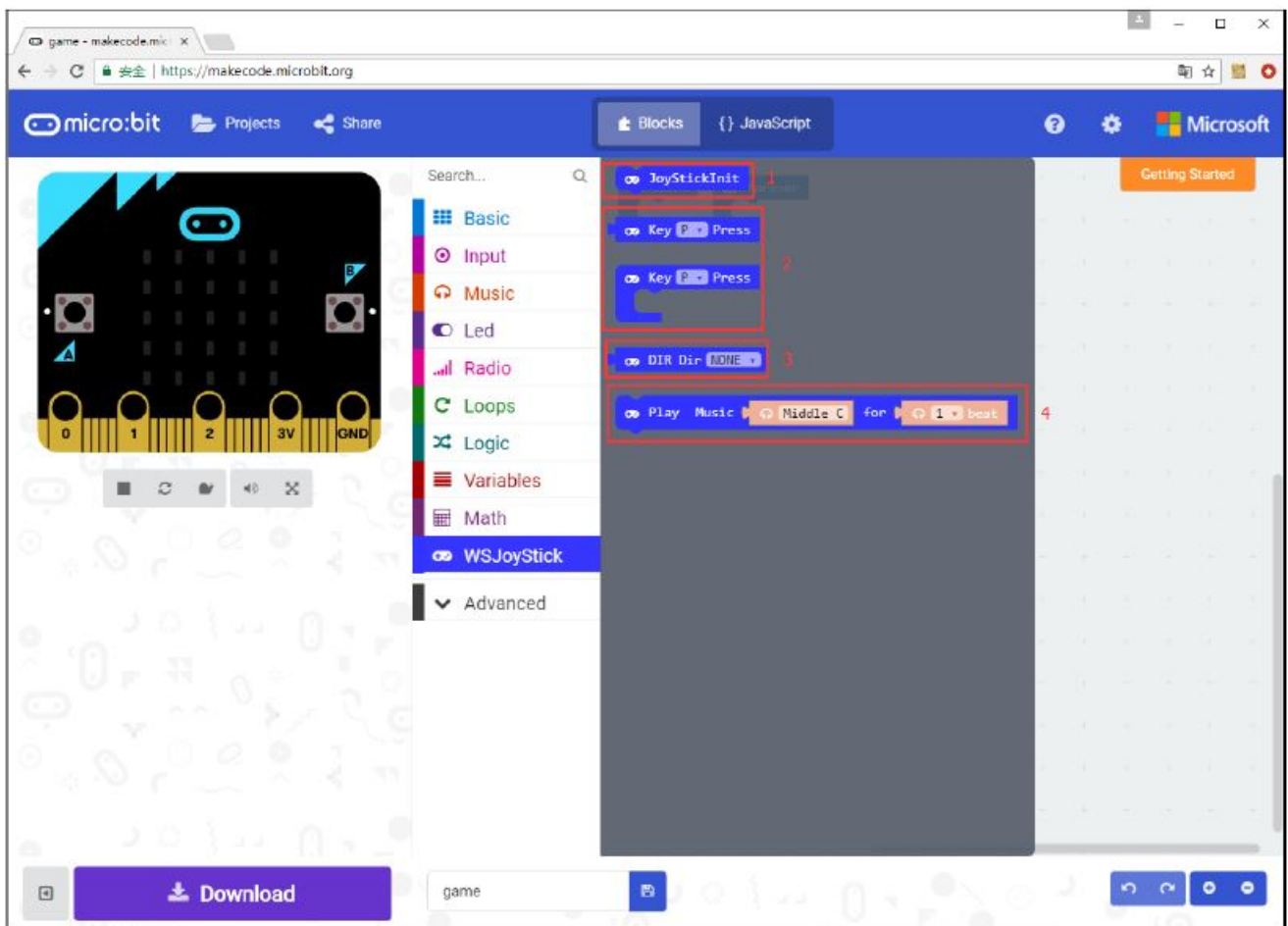
**Getting Started:** The website of typescript: <https://makecode.microbit.org/#> Open the browser and type the address:



1. Create a project: Click on Projects -> New project. Below you will see "Untitled". Click in and rename it to "game". Of course, you can use any name you wish for this project. To add the package, you can download the libraries we provide from GitHub: Click Advanced -> + Add package, or click on the gear icon of the top-right -> Add package. In the pop-up dialogue box, click the search field box to copy the: <https://github.com/waveshare/JoyStick>.

**Note:** Note that the end of the link needs to be added a space, otherwise it may not be indexed:

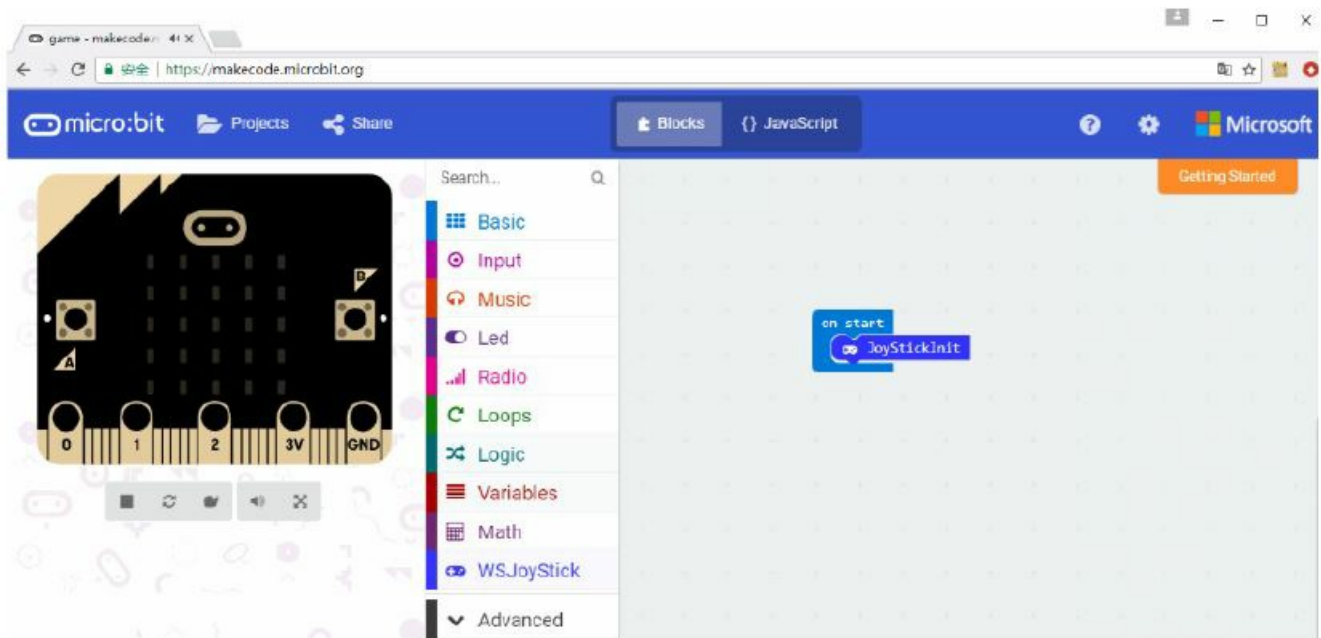




The functions of each block are as follows

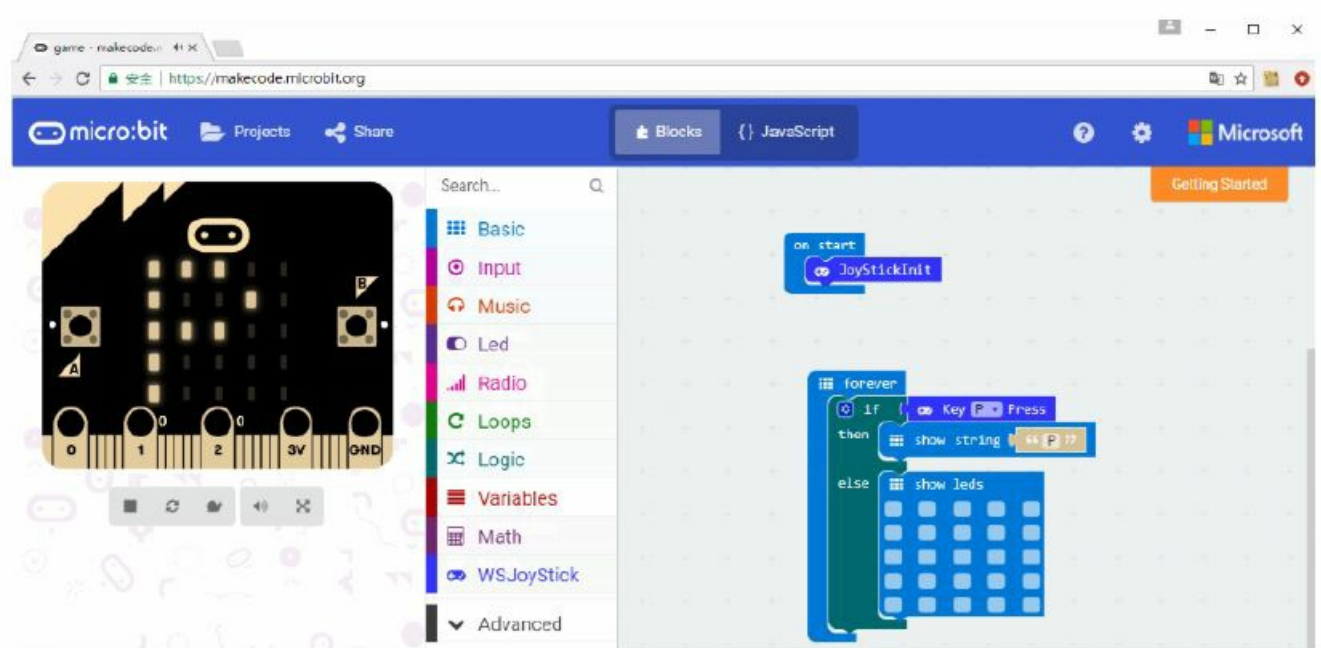
## Initialization

- This module requires the previous initialization of the block.
- In this block, there are five keys (except for the A key) that execute pull-up and read the joystick state.
- This state value is used to test any current operation done on the joystick position.
- If the initialization process is not complete, at moving the joystick, it might not judge a current position state.
- To fix this, don't move the joystick and reset the micro: bit to restore it.



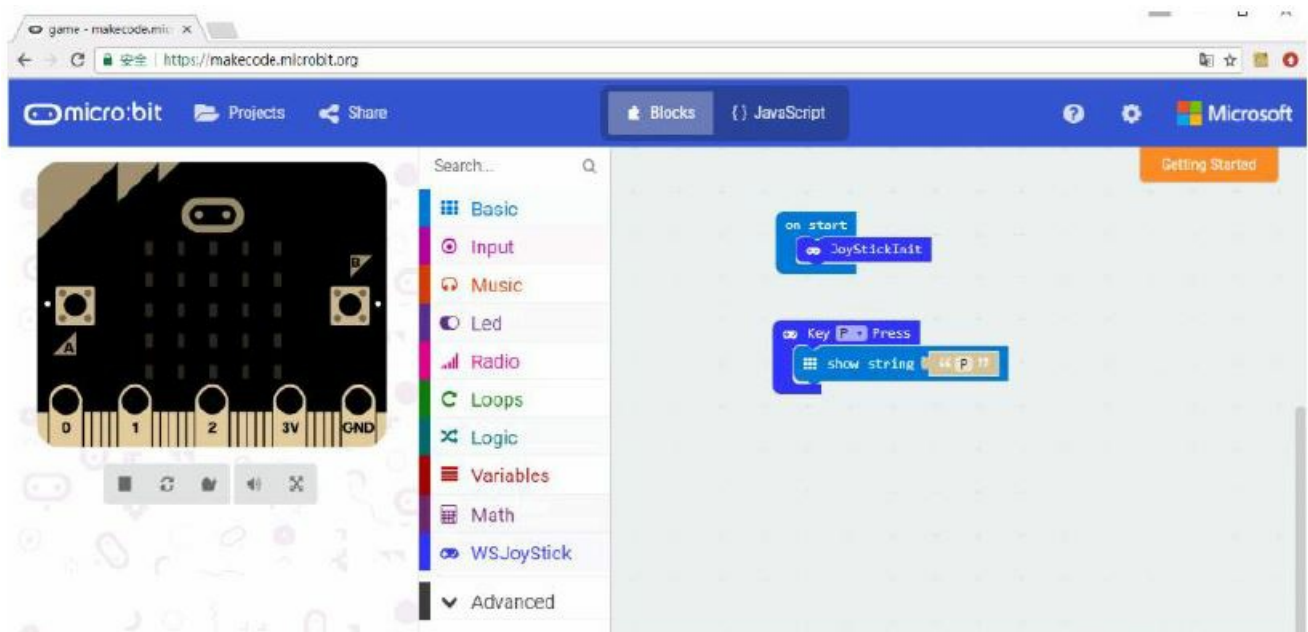
## Button monitoring

- We provide two ways of monitoring, each of which has its advantages. The first one is used with “if” which processes non-real-time events.
- This kind of event usually has delays.

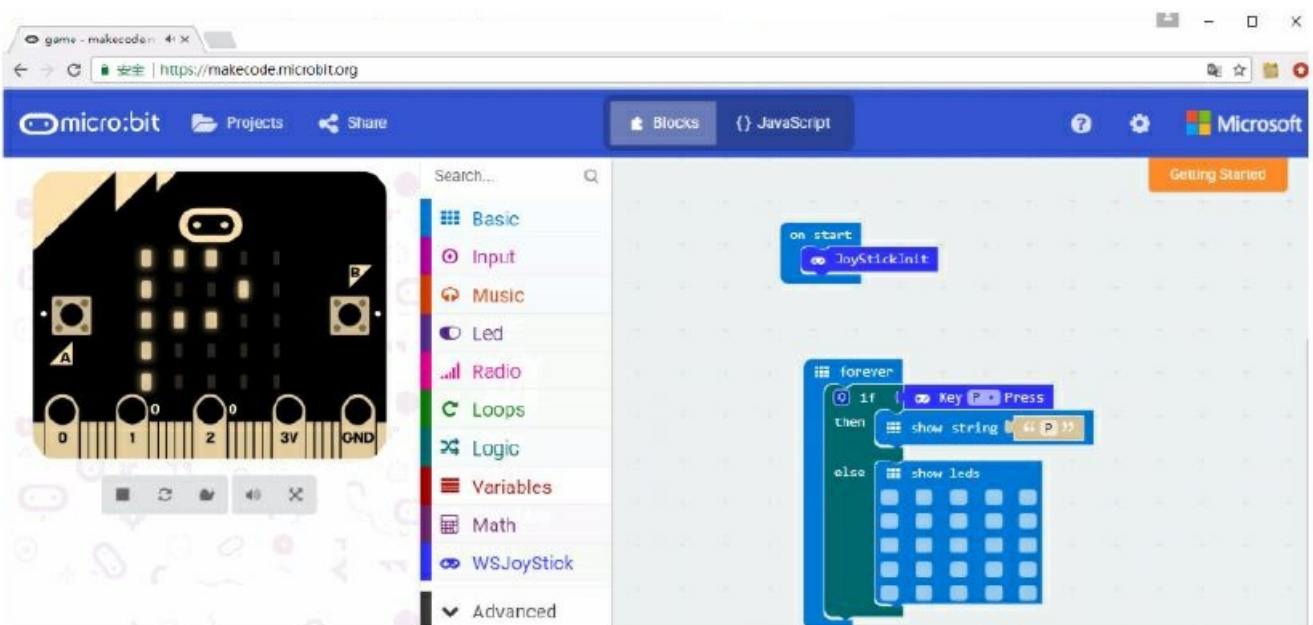


- The second one does not need “if”.
- It is similar to the “on button A pressed” block of the input category.
- This is an interrupt handling mechanism, that cannot be delayed, and the real-time performance is relatively strong.



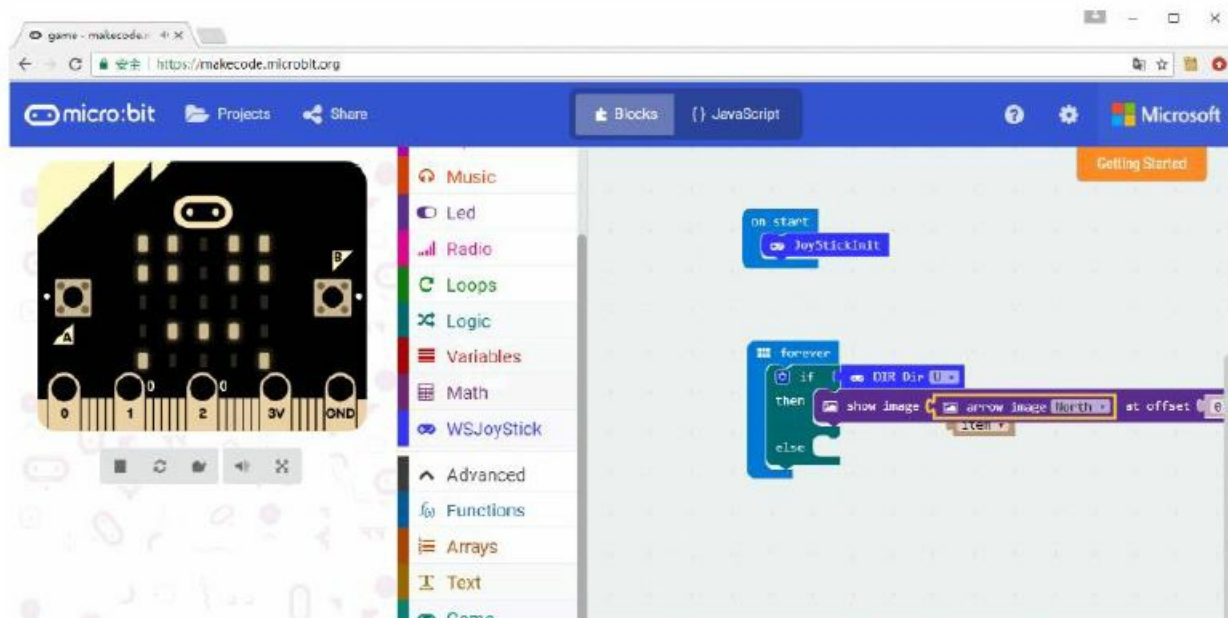


- **Expected result:** When pressing the joystick, the micro: bit will light up a “P” letter.

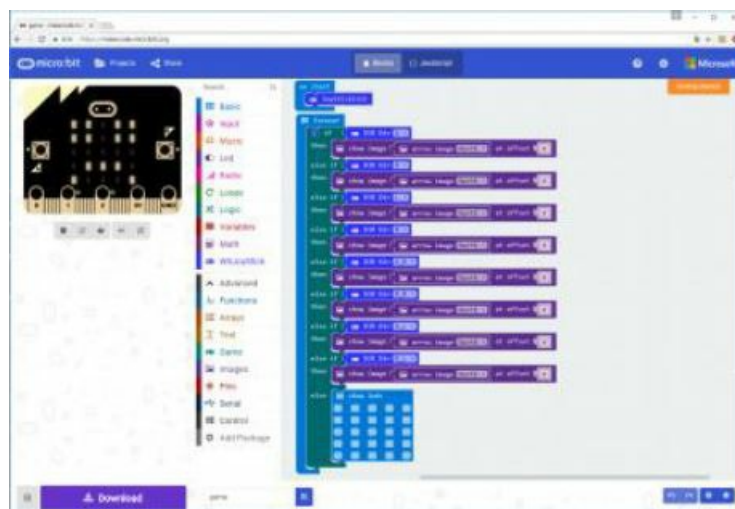


## Monitoring the joystick

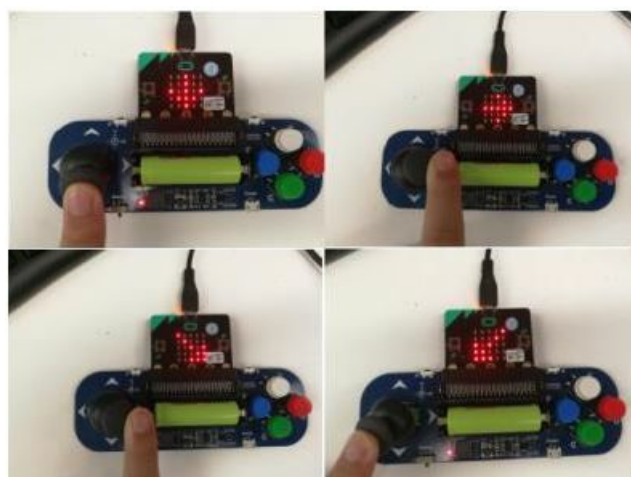
- If initialization is done before the block is used, at moving the stick to a direction, this will return its corresponding logic value TRUE.



- Place in sequence 8 directions as follows for judging each direction,



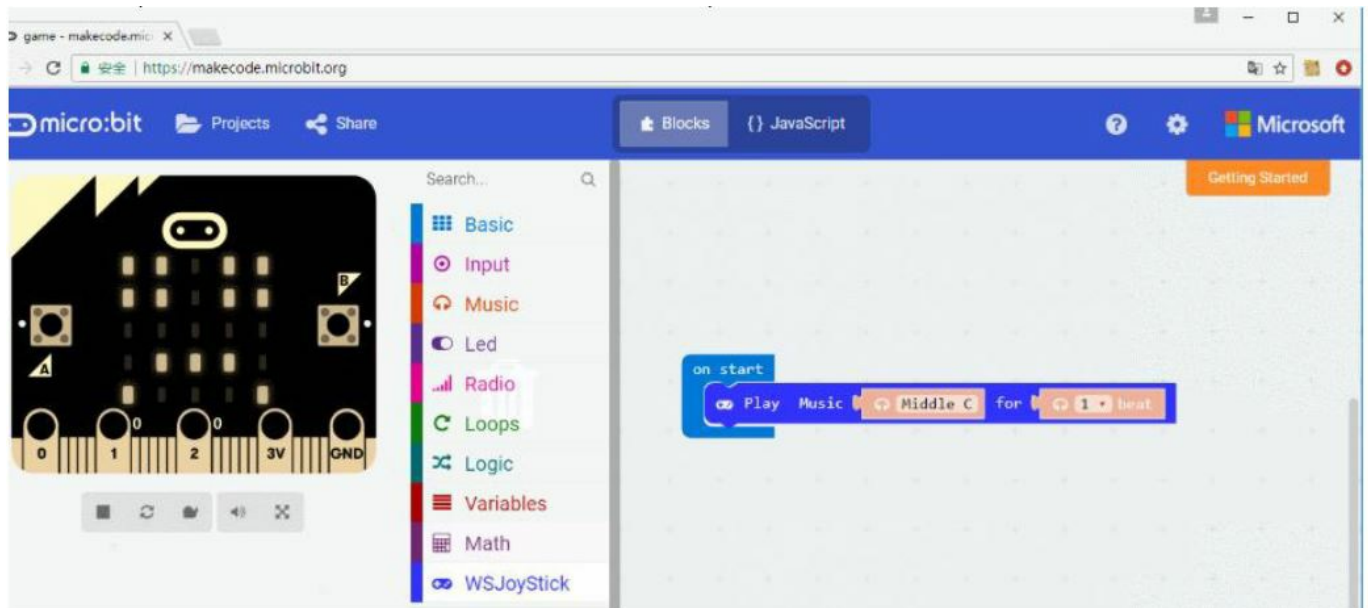
- Expected result: As you push the joystick, the micro: bit display will show an arrow corresponding to the colored direction



## Controlling the buzzer



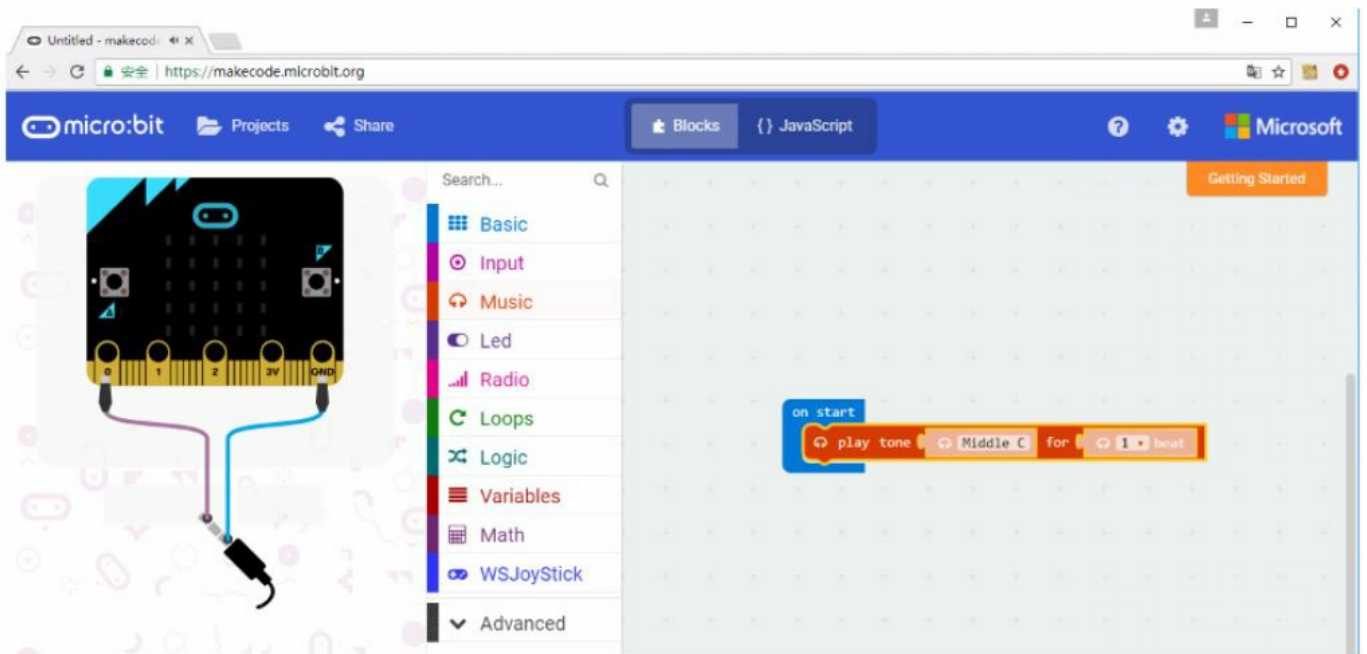
- There are two ways to control the buzzer. The first is to use the blocks we provide, and the second one is to use Micro: bit's music library.
- At first, we'll use our block, which is quite the same as micro: bit. The first parameter selects the note, and the second parameter selects the beat.



- Place them in turn as follows:



- **Expected result:** Download the program to the module, which will make the onboard speaker sound.
- The second one is about using micro: bit's music blocks, which are compatible with the pins.
- It's the same as the one above.



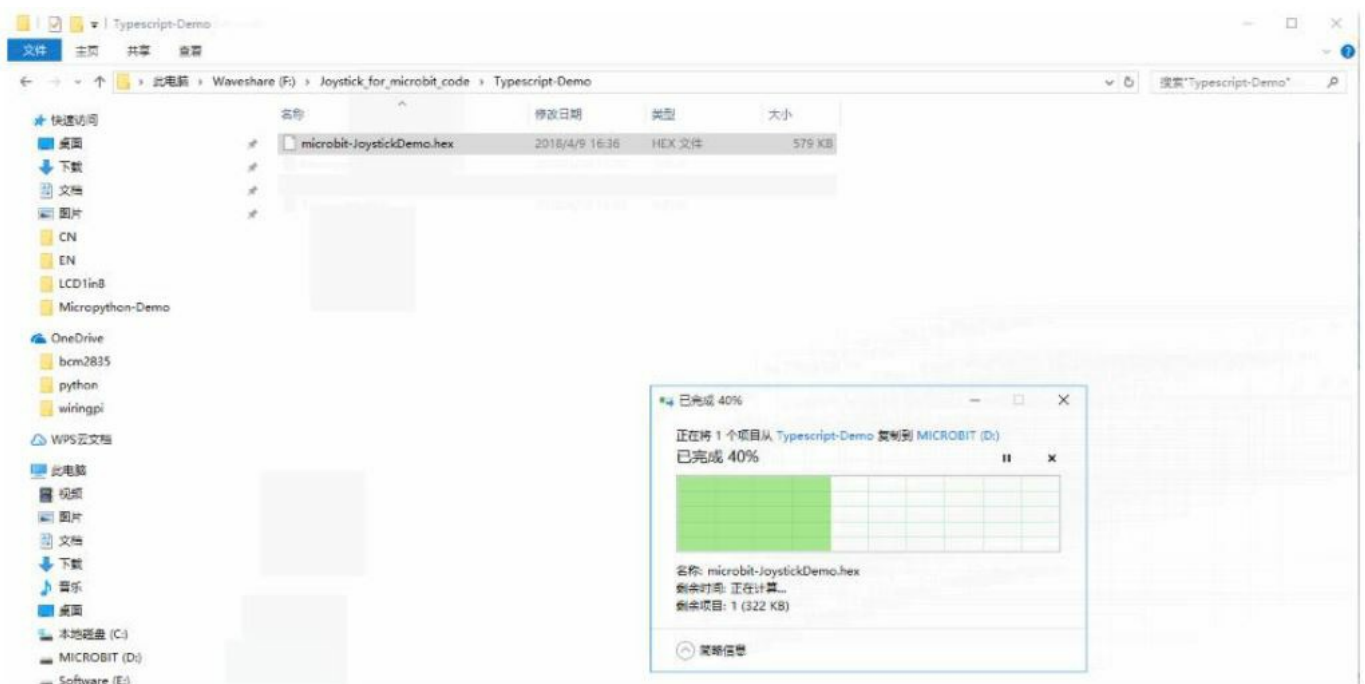
- You might be willing to use other blocks as well, next, we show you more blocks as follows.



## Verifying Demo

- Open the Typescript-Demo which holds the microbit-joystickdemo.Hex file. You can copy it directly to the micro: bit connected to the computer. You can also download it from the last edition of MakeCode.

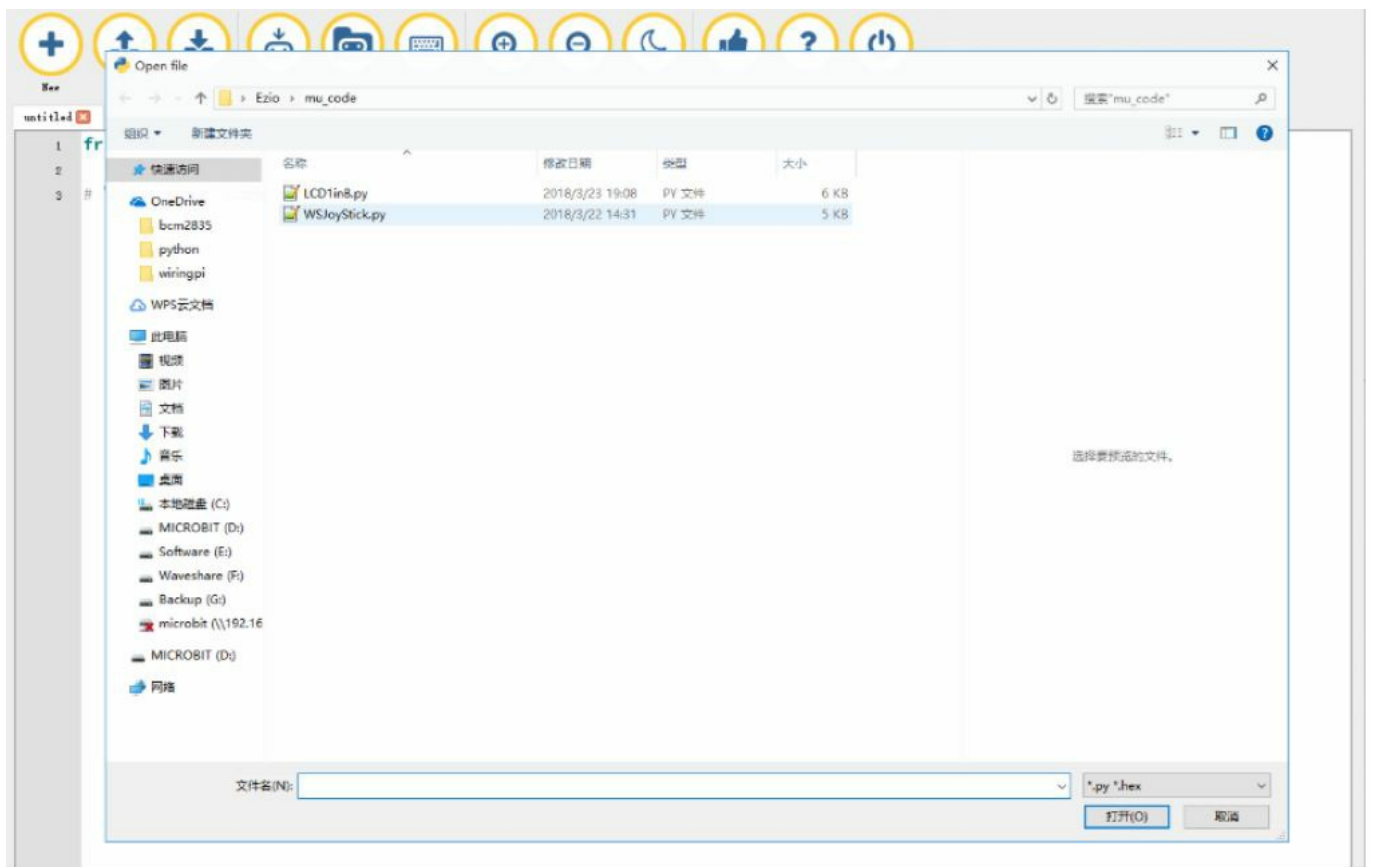
- Download directly to the micro:bit:
- Connected Micro: bit to the computer by USB cable. Your computer will recognize a USB flash drive as a MICROBIT of about 8MB space. Now copy the microbit-joystickdemo.Hex file to this USB flash disk.



**First upload to Makecode, then download**



**Micro Python** is this kind of program, you can use the official programming website or download the programming tool Mu. The online programming website: is <https://codewith.mu/#download> The outline programming software: is <https://codewith.mu/#download> you can also download it on the resources part of this page Open the software.



In the program, you can see the following methods implemented: No initialization is needed when using Python because this step is done when instantiation happens.






- Listen\_Dir (Dir): monitor the direction of the joystick.
- Listen\_Key (Key): monitor keys
- PlayScale (freq): playing the sound of a user-defined note
- Playmusic (tune): play music/melody

## Documents / Resources

|  |  |
|--|--|
|  | <p><b>BBC Micro Bit Game Console</b> [pdf] User Manual<br/>         Micro Bit Game Console, Micro, Bit Game Console, Game Console, Console</p> |
|--|--|

## References

- [Code With Mu](#)
- [GitHub - waveshareteam/JoyStick](#)
- [Microsoft MakeCode for micro:bit](#)

-  [Microsoft MakeCode for micro:bit](#)
-  [File:Blocks music.png - Waveshare Wiki](#)
-  [File:Place music.png - Waveshare Wiki](#)
-  [File:PlaceDir All.png - Waveshare Wiki](#)
-  [File:WpsF913.tmp.png - Waveshare Wiki](#)
- [User Manual](#)