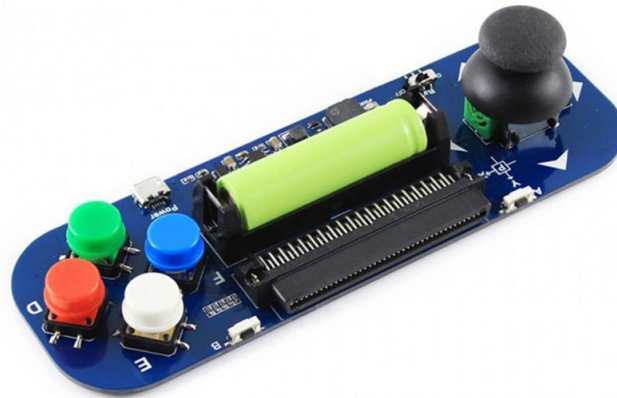
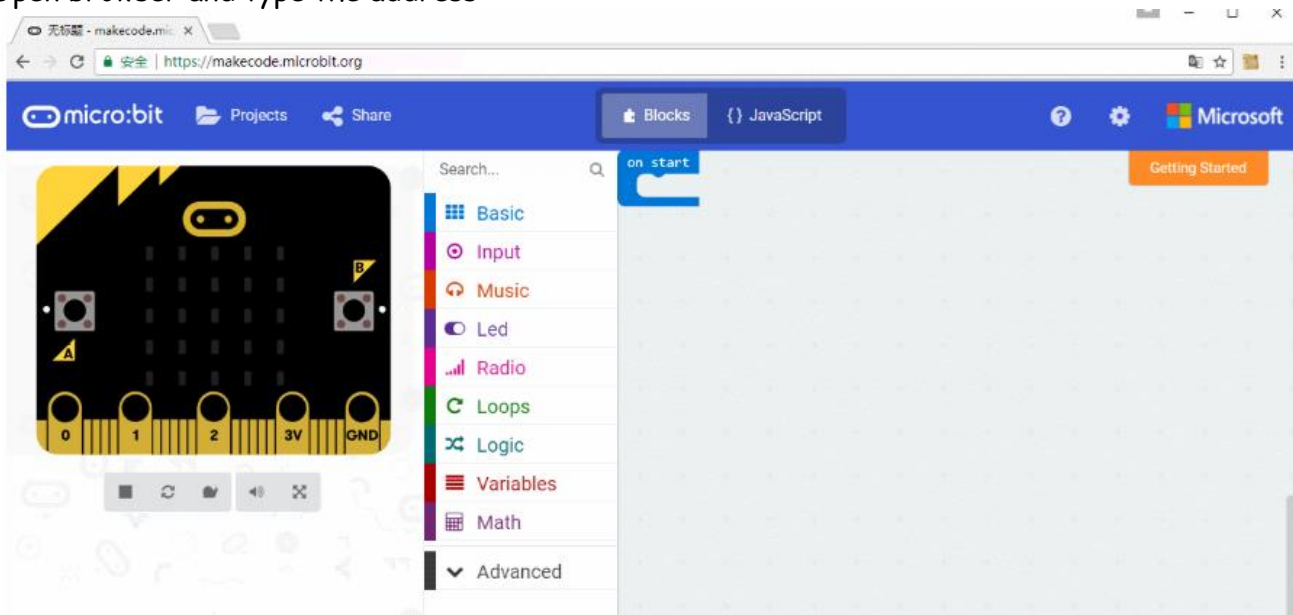


# BBC Micro Bit Game Console User's Manual

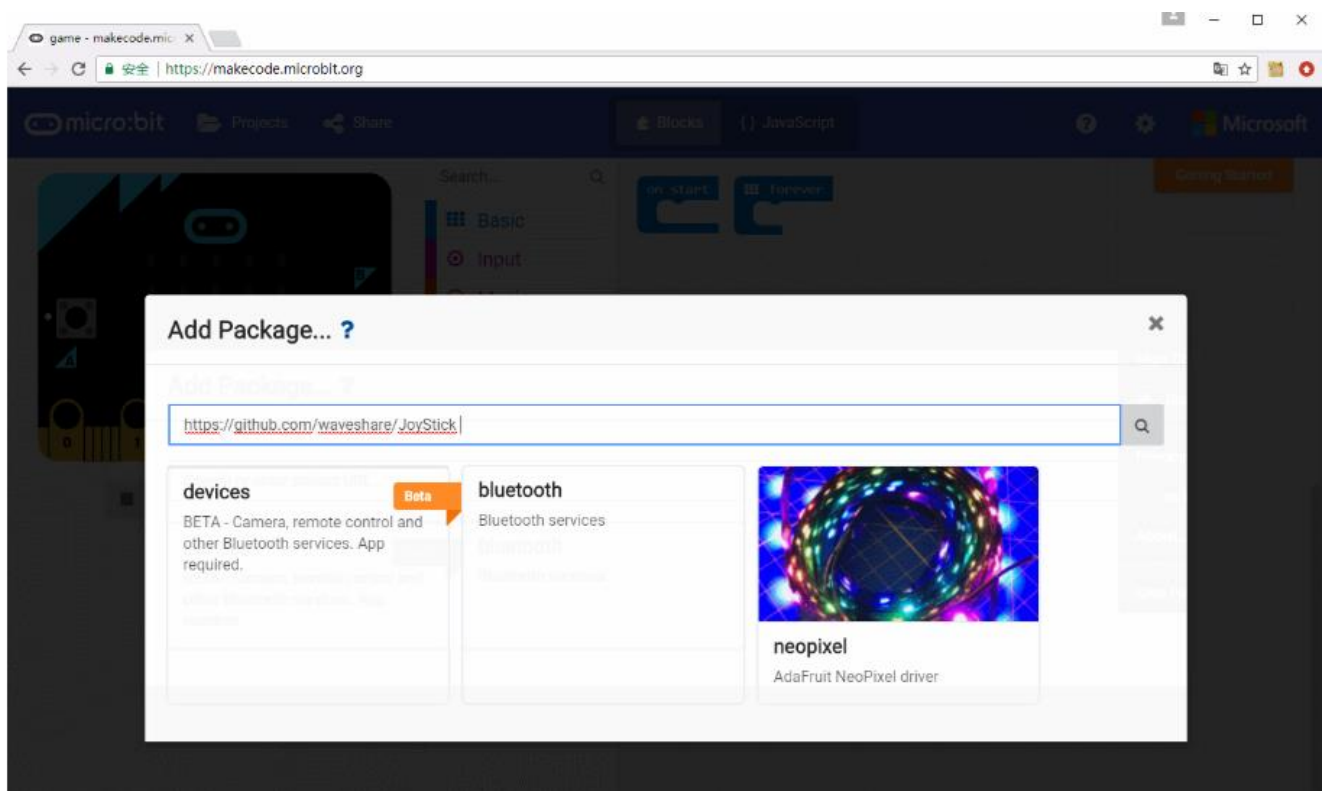
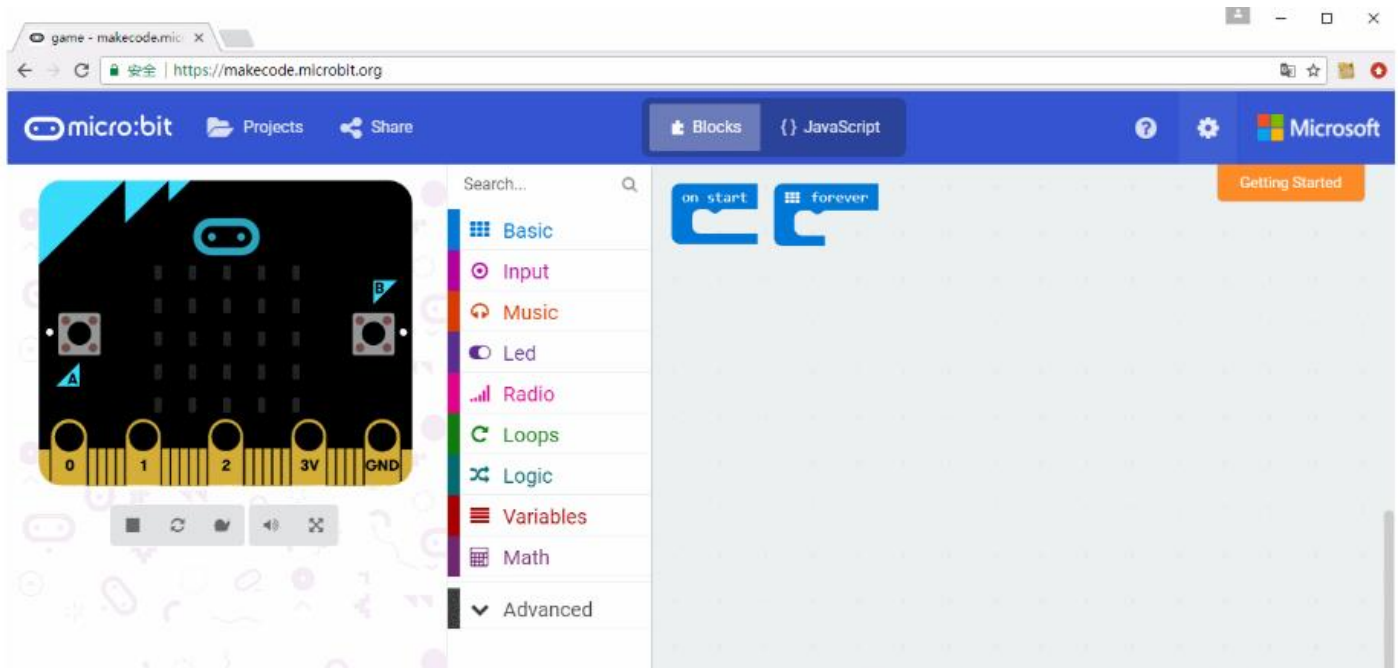


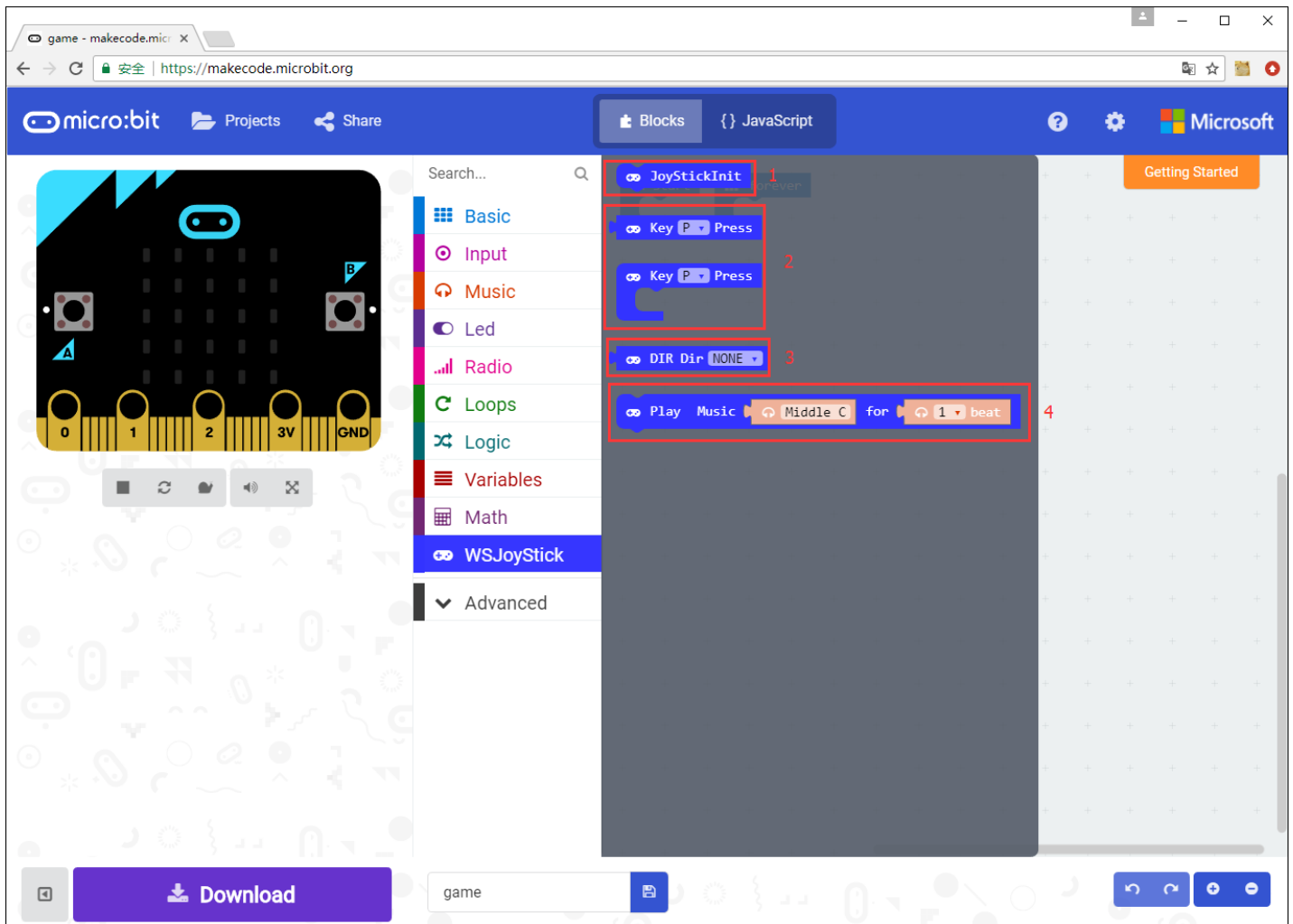
## Getting Started:

The website of typescript: <https://makecode.microbit.org/#>  
Open 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 dialog box, click the search field box to copy the  
URL: <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 to:

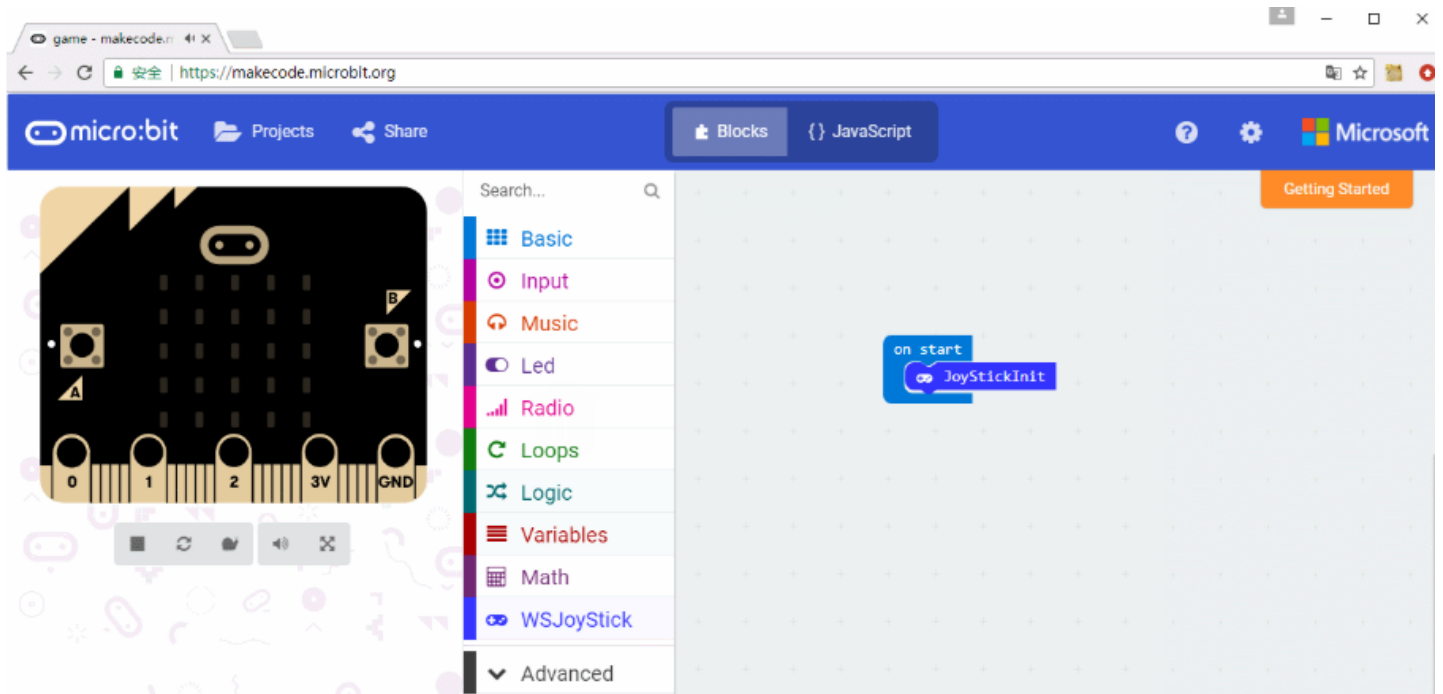




### The functions of each block are as follows:

#### 1. Initialization:

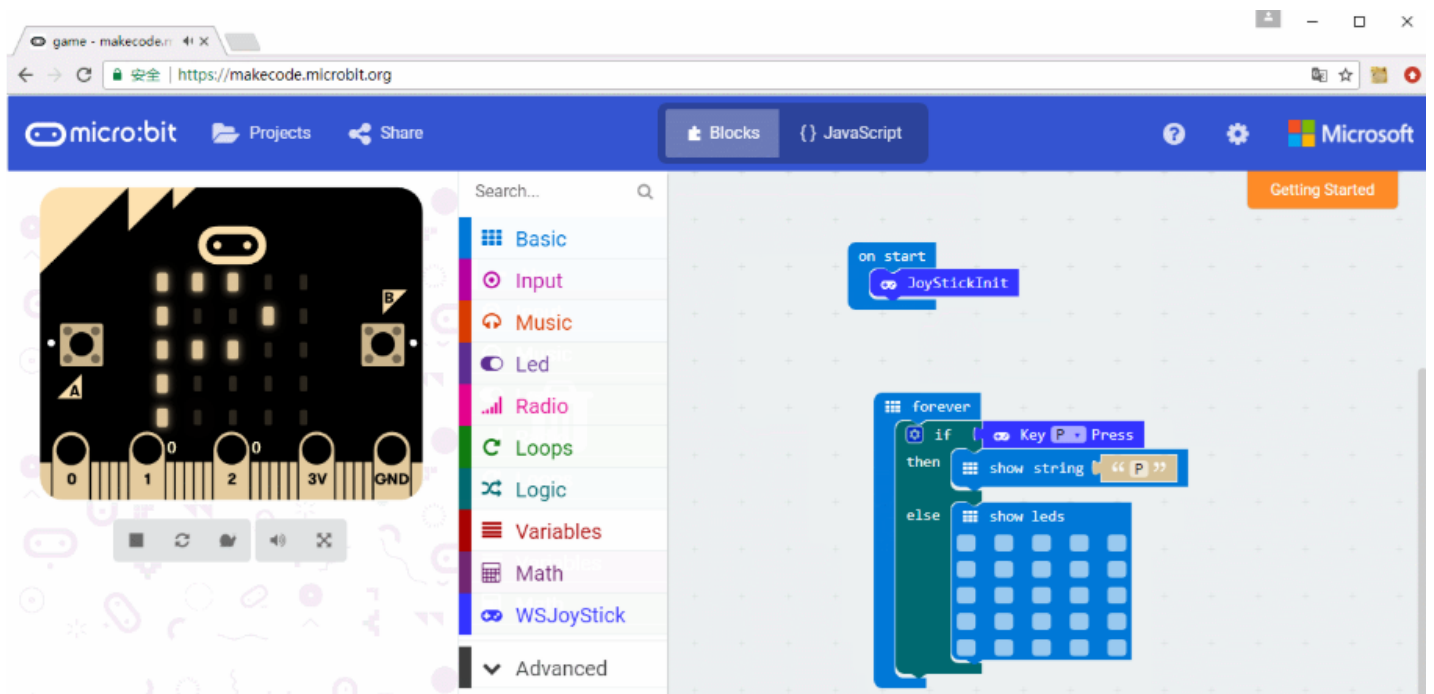
This module requires previous initialization of the block. In this block there are five keys (except for A key) which executes pull-up and read joystick state. This state value is used to test any current operation done on the joystick position. If the initialization process is not complete, at moving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



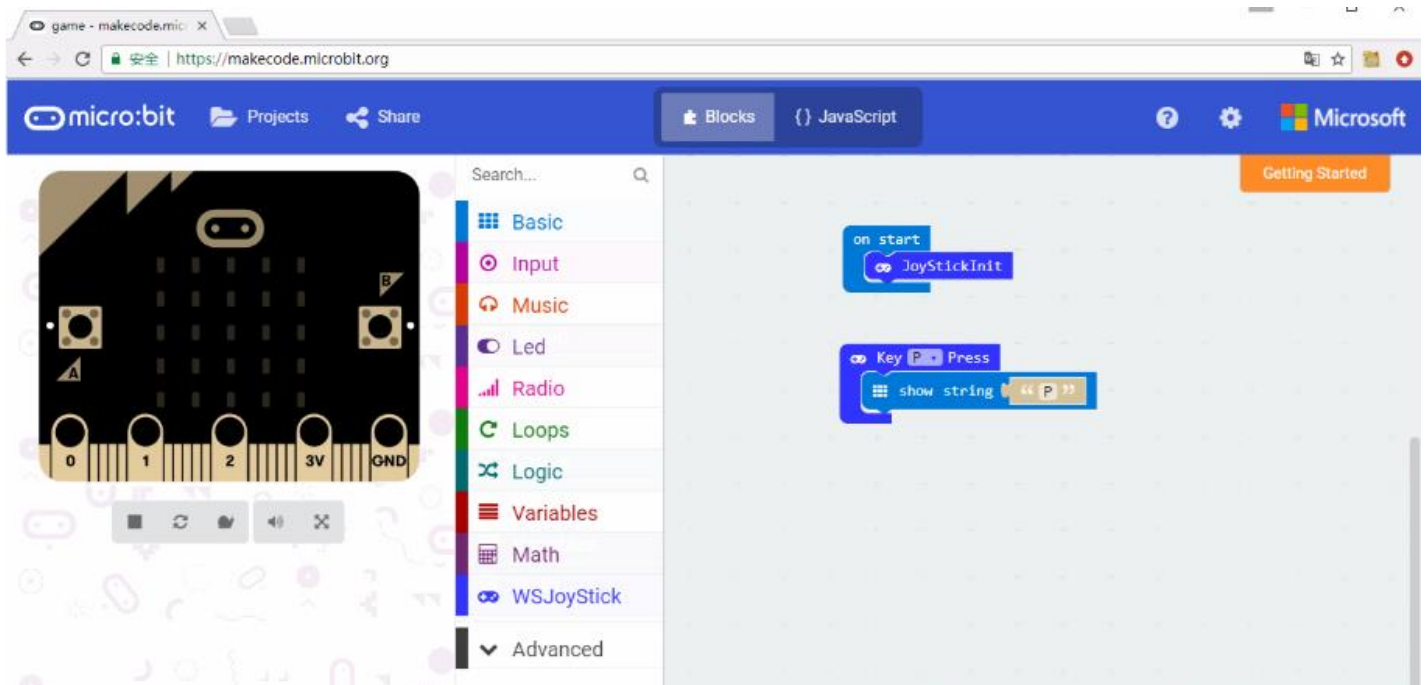
## 2. Button monitoring

We provide two ways of monitoring, each of which has its own advantages

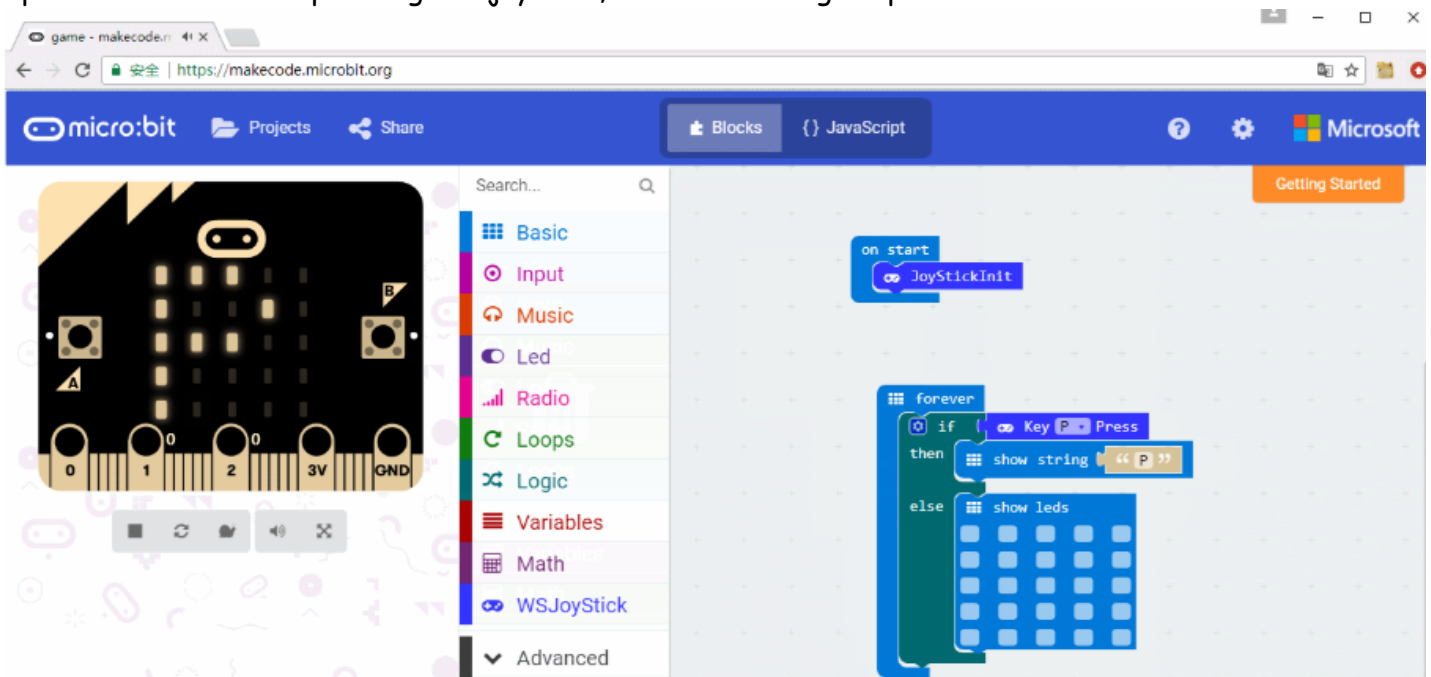
The first one is used with "if" that process non-real-time events. These kind of events usually have delays



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



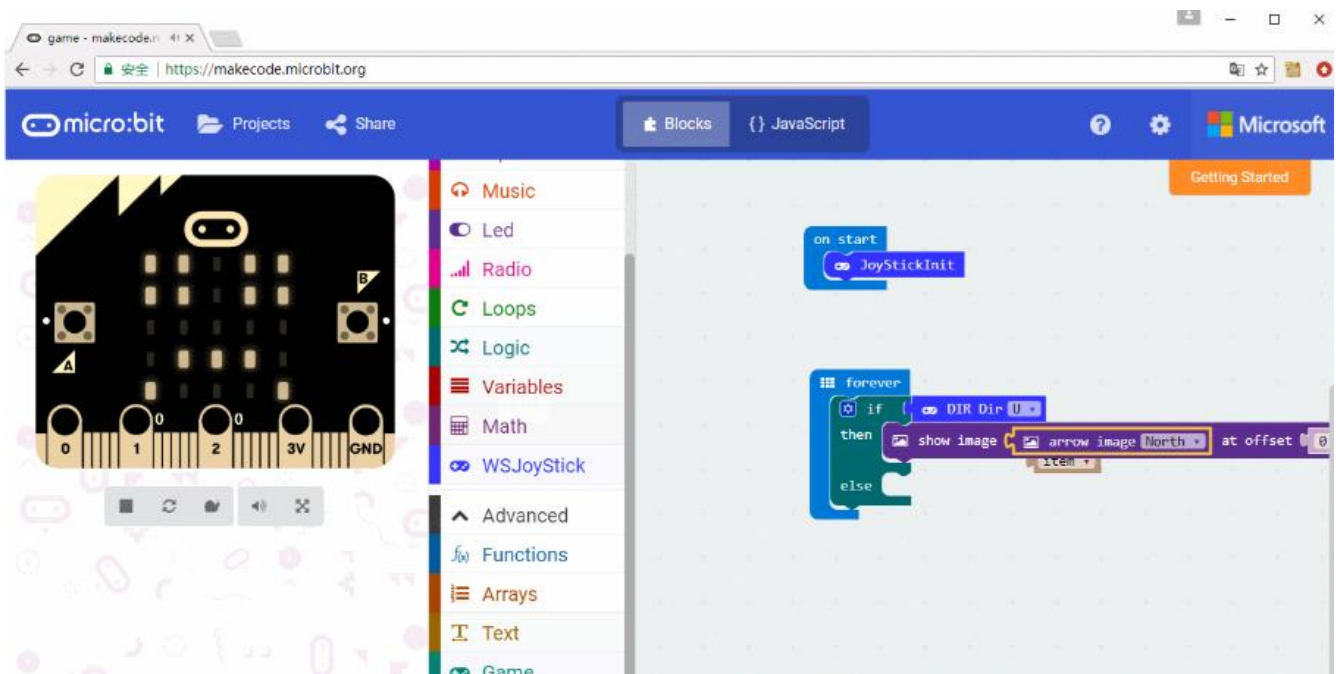
Expected result: When pressing the joystick, micro:bit will light up a "P" letter.



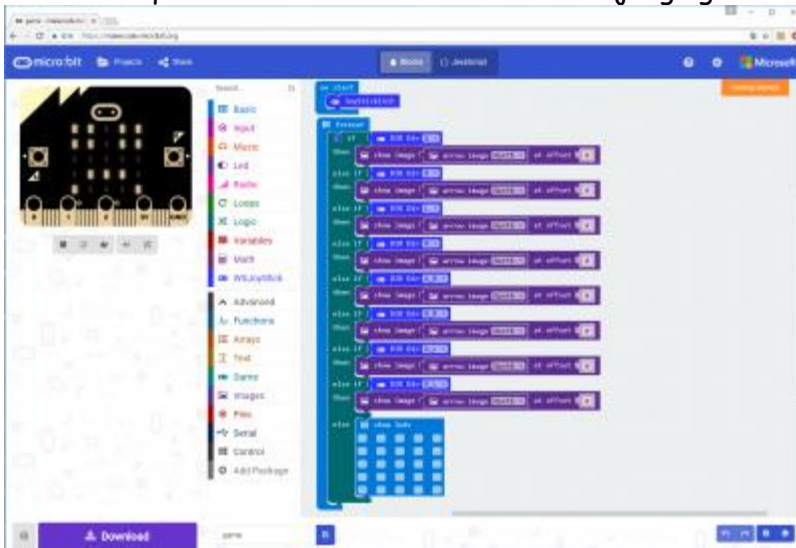
### 3. Monitoring the joystick

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

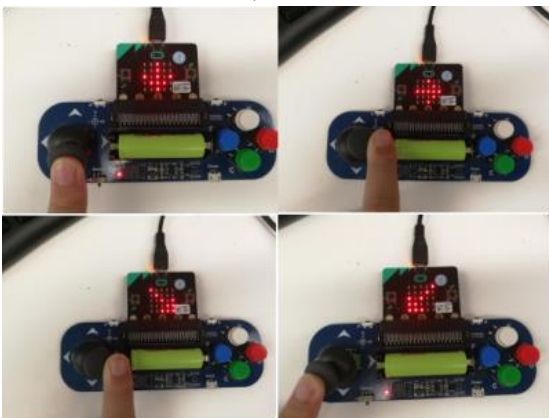




Place in sequence 8 directions as follows for judging each direction,



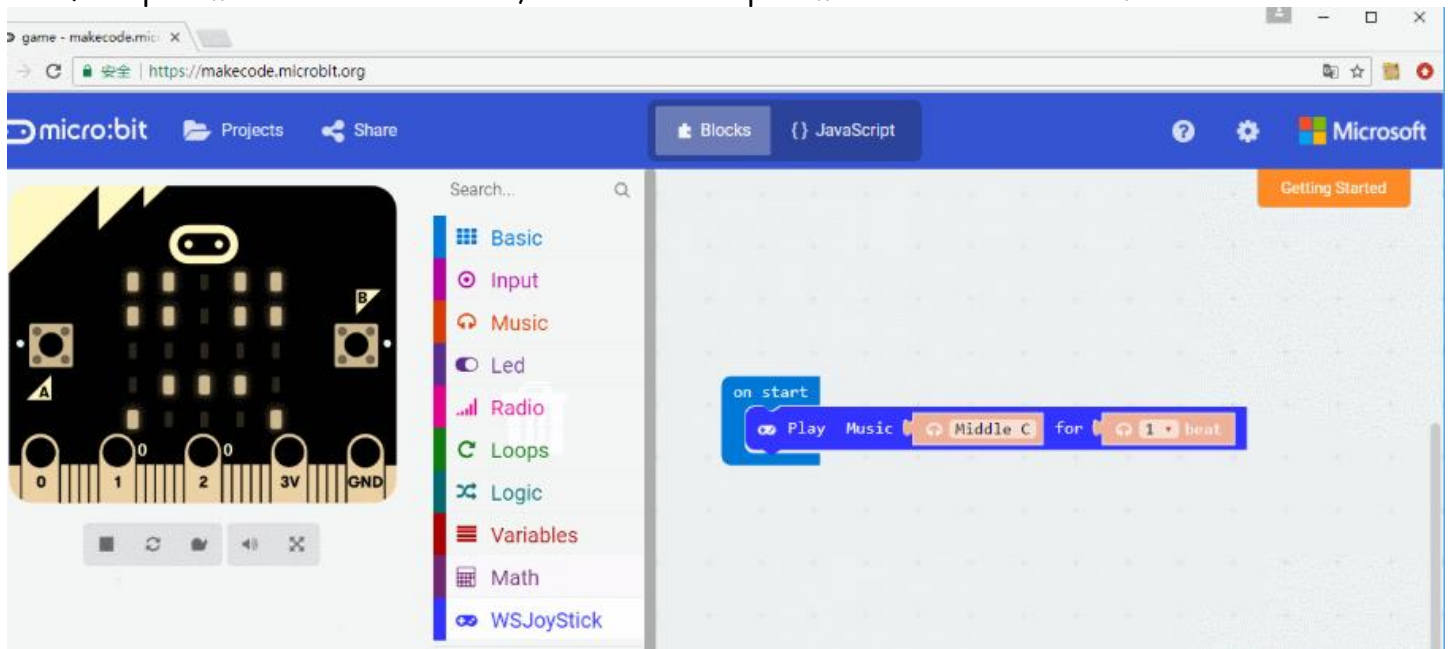
Expected result: As you push the joystick, micro:bit display will show an arrow corresponding to the selected direction.



#### 4. 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 own music library.

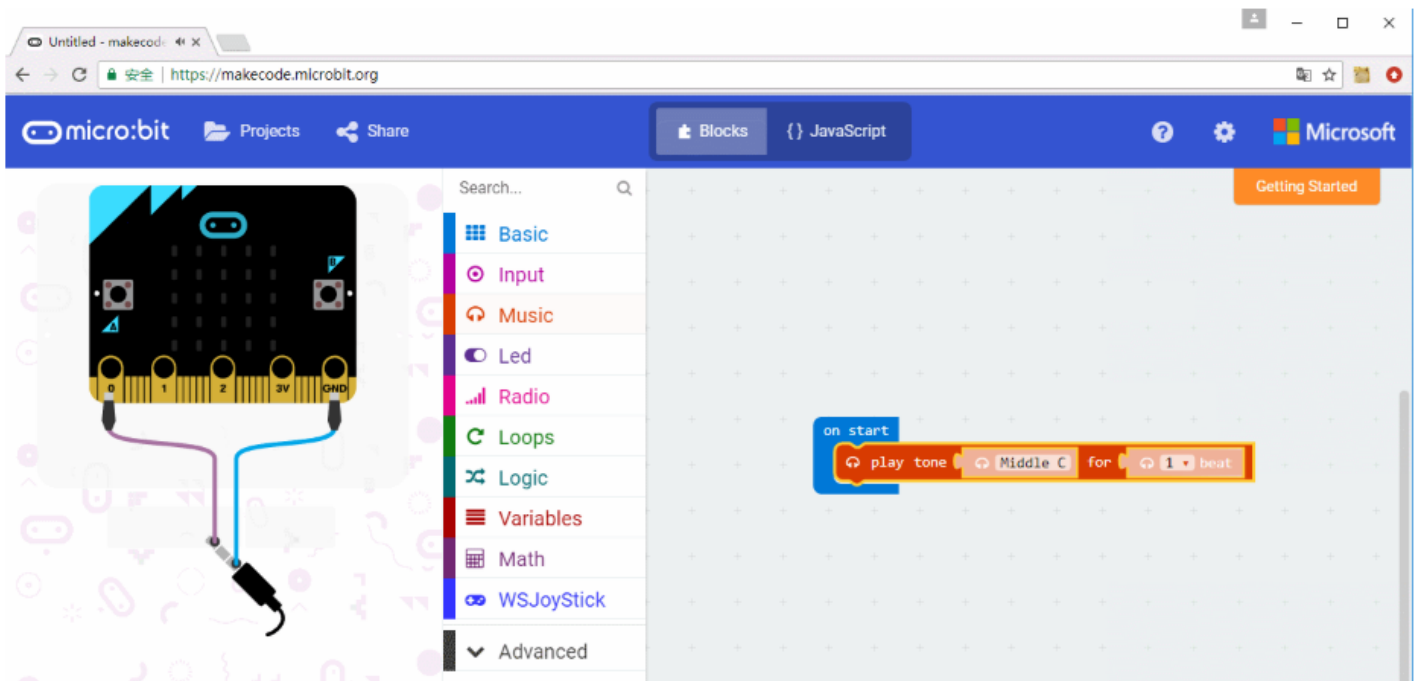
At first we'll use our block, which is quite the same as micro:bit.  
The first parameter selects the note, and the second parameter selects the beat.



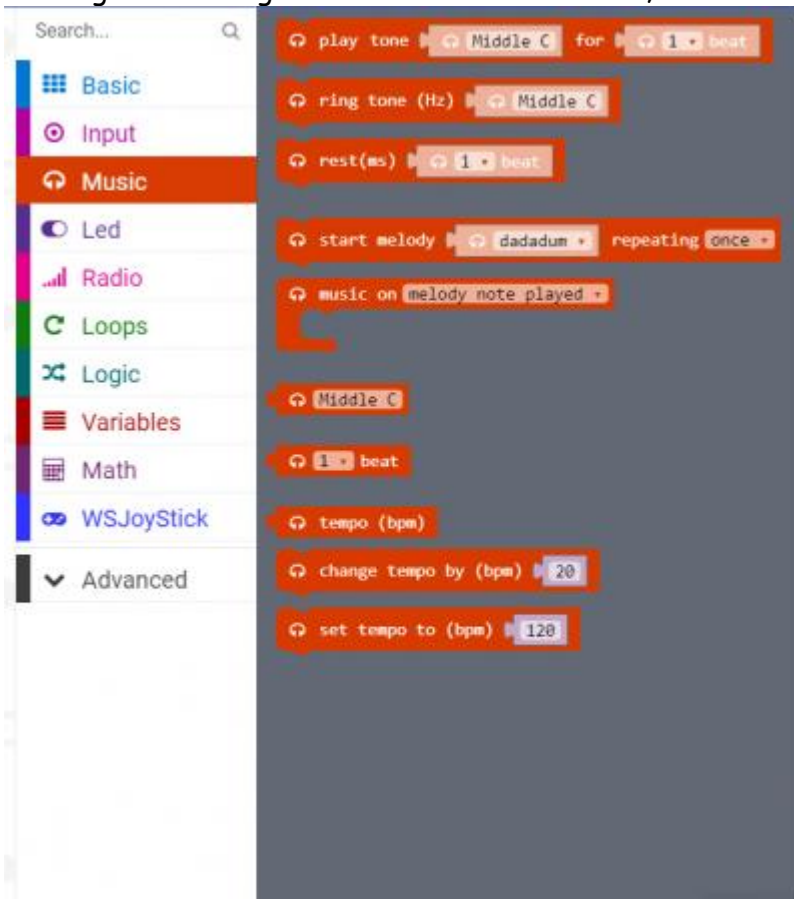
Place them in turn as follows:



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



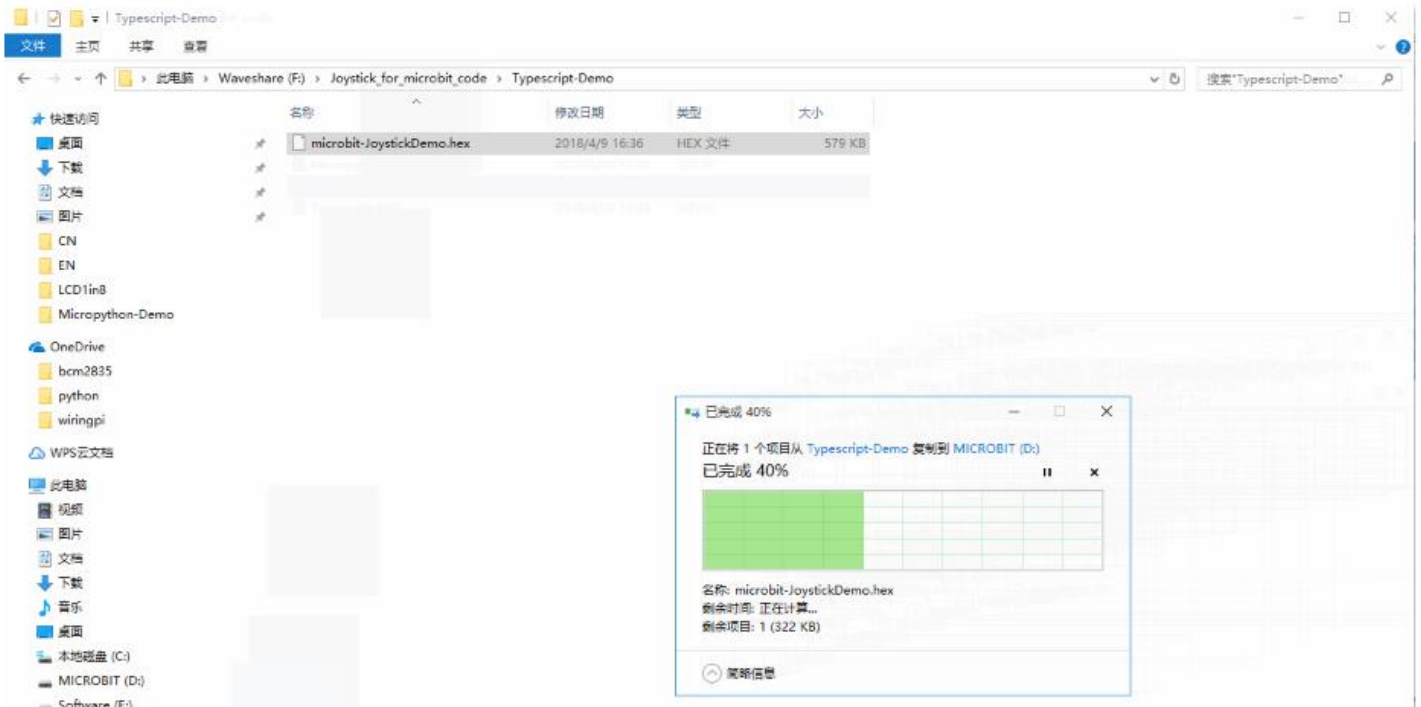
You might be willing to use other blocks as well, next we show you more blocks are as follows





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



First upload to Makecode, then download:



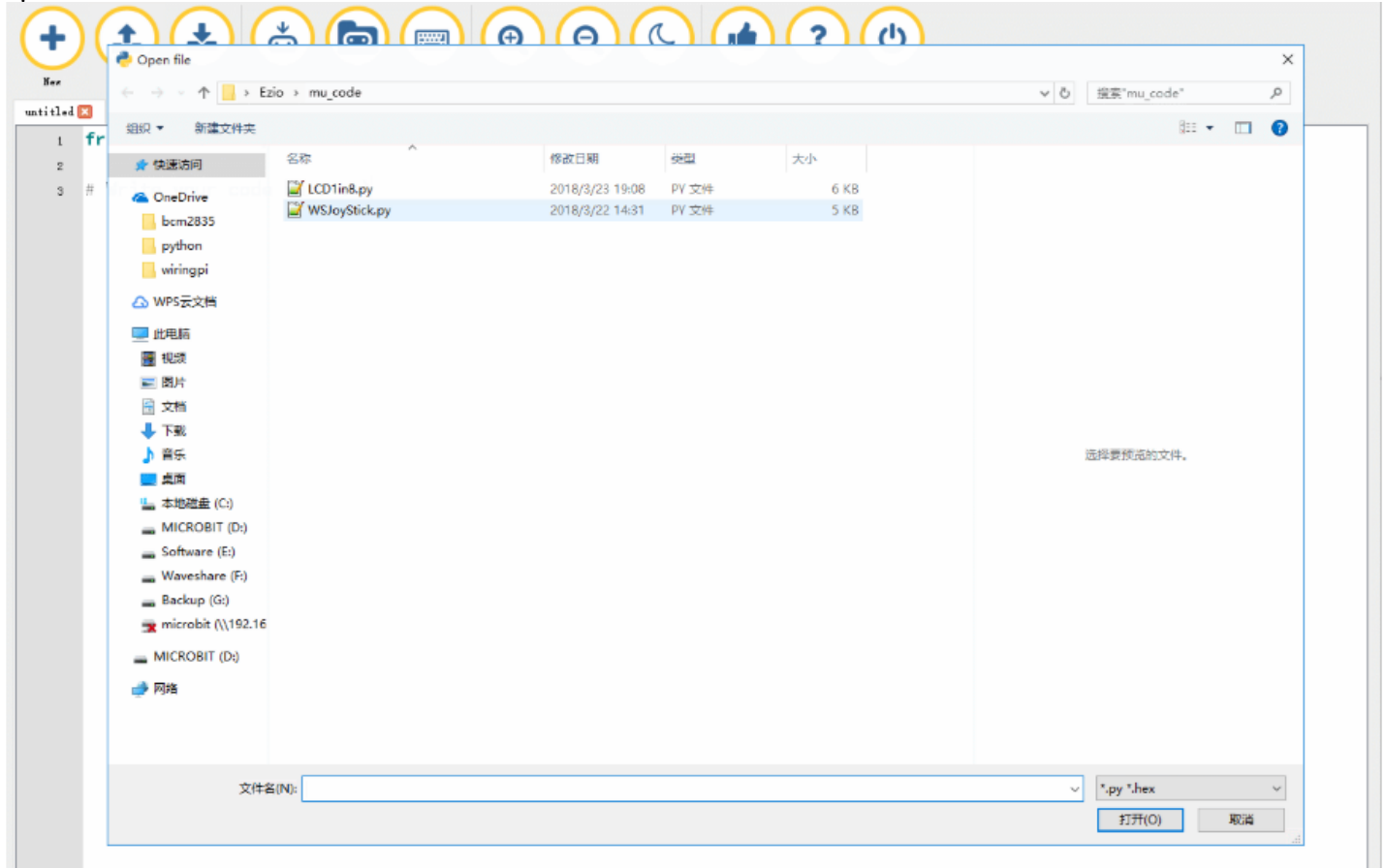
## Micro python

This kind of program, you can use the official programming website or download the programming tool Mu.

The online programming website: <https://codewith.mu/#download>

The outline programming software: <https://codewith.mu/#download> (you can also download it on resources part this page)

Open the software



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

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