



# Piano for Micro:bit User Manual

## INTRODUCTION

Micro: bit is an ARM-based embedded system designed by the BBC of England, which is equipped with Bluetooth, accelerometer, electronic compass, three buttons, and a display consisting of 5x5 LED lattice. It is mainly used for computer education addressed to teenagers.

You can use the BBC micro: bit to build any cool gadgets, whether it's a robot or a musical instrument, you can make anything you can imagine.

This product is a mini piano expansion board which is based on micro:bit. The board features the capacitive touch TTP229 chip, with 13 piano keys and 4 RGB lights. You will be able to play the full octave by touching the piano keys, and the RGB LED will display accordingly.

## FEATURES

Touch Controller: TTP229

Control Interface: I2C

RGB LED: WS2812B

Dimension: 85mm x56mm

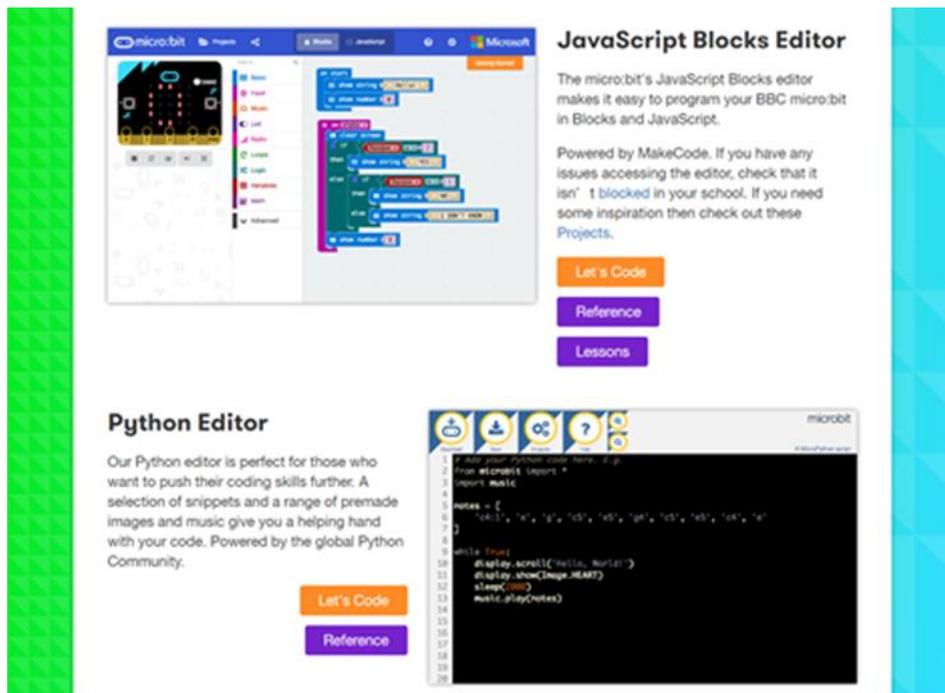
Operating Voltage: 3.3V

## INTERFACE

PIN	Description
VCC	3.3V
GND	Ground
P0	Buzzer
P1	RGB LED
P19	I2C Clock

## USE GUIDE

### HOW TO USE IT



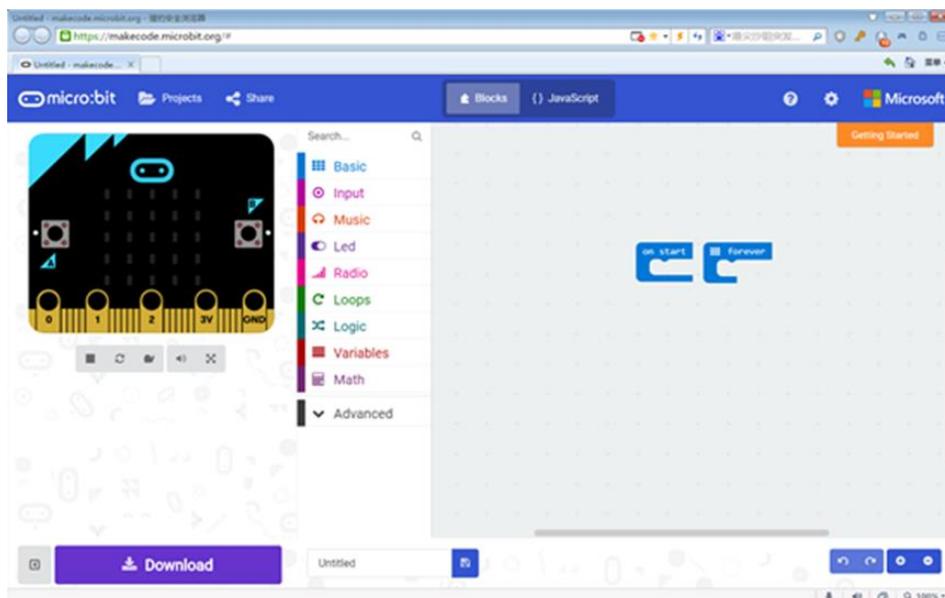
As for its use, micro:bit officially recommends two programming methods:  
<http://microbit.org/code/>

As you can see, the official recommendation is to use JavaScript and Python. In their website, there are examples of these two programming languages. Click on "Let's Code" to open the corresponding official website.

## JAVASCRIPT

Use any browser to open Micro:bit's website as shown below.

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



Click “Projects”, then “My stuff” and “New Project” to create a new project. Also, you can import previous projects, or refer to other people’s projects and examples;

On the left of micro:bit, there’s an analog display area. When the corresponding blocks are placed in the building blocks area, its dynamic effects will be displayed in response to them.

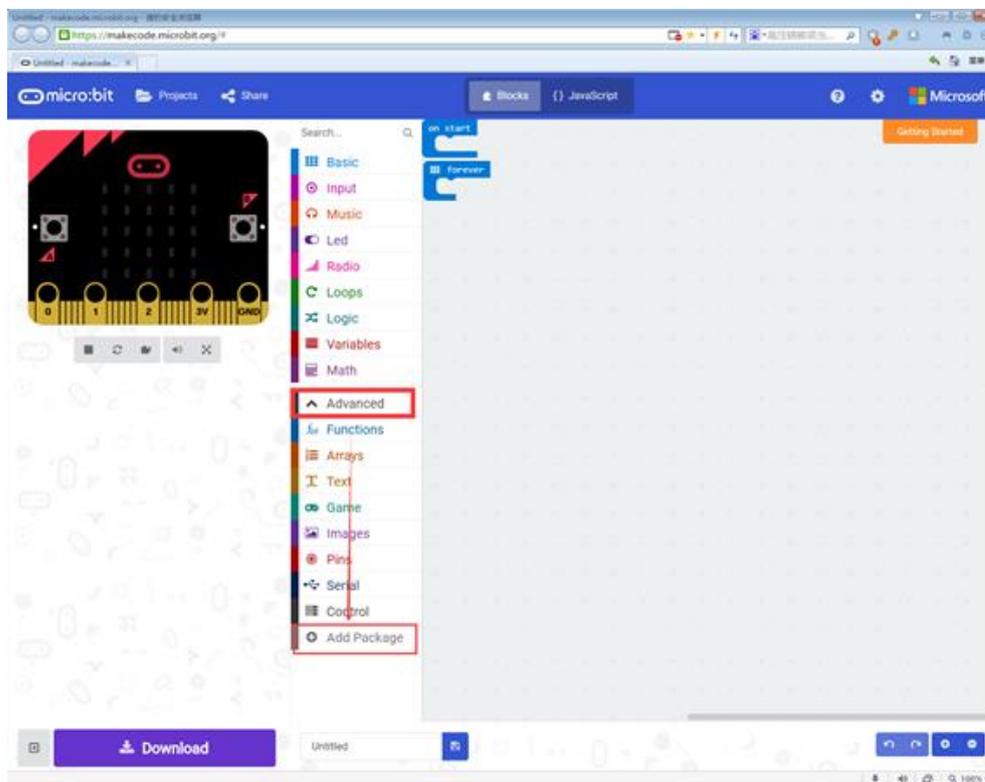
In the middle of the selecting area there are some common building blocks packages. Respectively, these are the basic functions, input, music, LED lights, Radio, loops, logic, variables, and Math and Control. By combining them together it can be achieved any kind of interesting effects. There are also more advanced control blocks further below. Thus, mastering the basic building blocks will allow the student to explore deeper levels of control.

On the right side is the block programming operation area, you can see two options on the top: Blocks and JavaScript. Click on “Blocks” to place blocks inside. Click on “JavaScript” to see generated code. Little students not familiar with JavaScript should better use “Blocks”. By default, in Blocks, you can place the blocks you need in the placement area. Let’s micro:bit display!

## INSTALL A BUILDING BLOCK PACKAGE

After the most basic operations introduction is done, let’s get straight the point. What do we need to do with our new piano module?

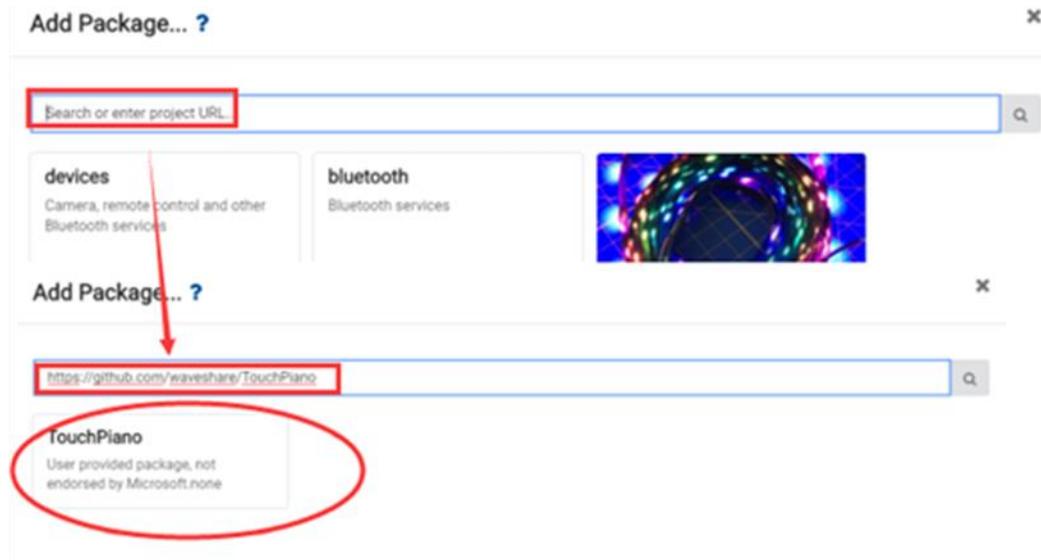
First click “Advanced” and scroll to the bottom to find and click “Add Package”.



In the pop-up box, note that the url should be added with a space:

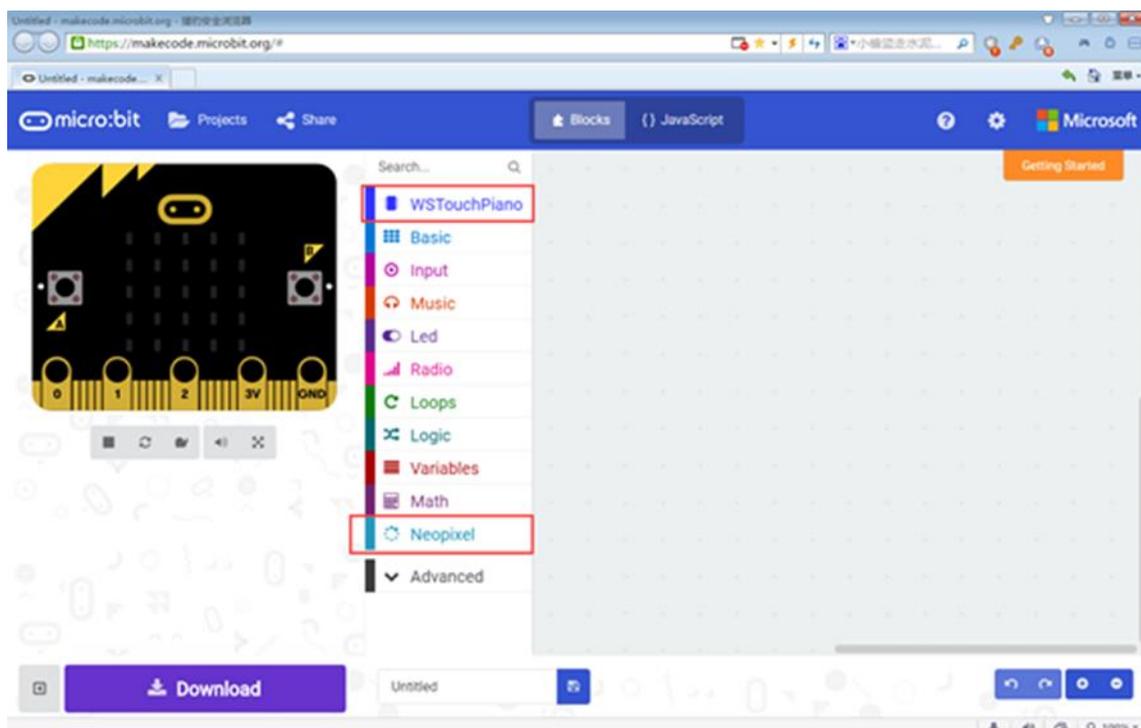
<https://github.com/waveshare/TouchPiano>

On the searching result, click “piano for micro:bit”. This way you'll have added the blocks to control the piano module.



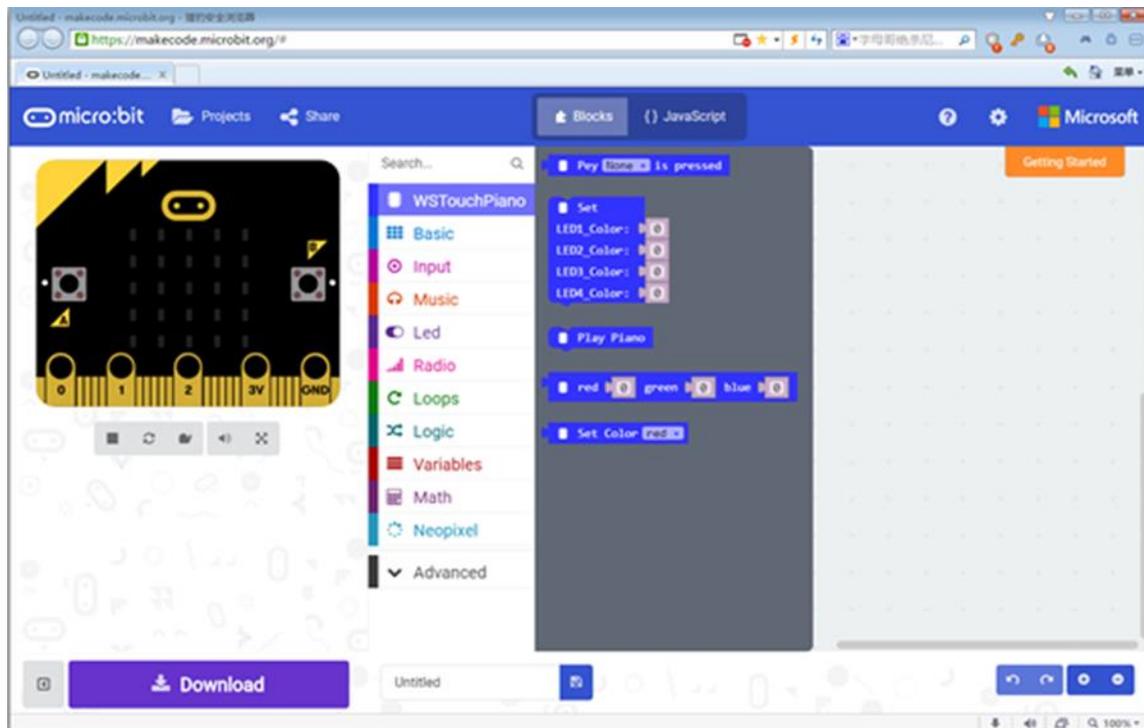
Next, we will continue installing the “Neopixel” block package, which is the control module of the RGB lights. Similarly, click on “Advanced” -> “Add Package”, then input in the popup window the following url: <https://github.com/Microsoft/pxt-neopixel>

The last two installed libraries will appear on the left of the blank space.



## BASIC USE

Clicking on “WSTouchPiano” you should see as below:



The functions of each block are as follows:

Hereby declare: the following are sample tutorials.

### 1. PLAYING PIANO

#### Placement steps



Go to “Basic”, and place “forever” slot into the blank space on the right. Place the “Play Piano” block of WSTouchPiano inside the “forever” slot. Click “Download” button on the bottom left to save the project, move the .hex file to the micro:bit drive to transfer the code into your micro:bit, and insert micro:bit to Piano's interface module. Now, you can observe our experiment outcome!

Removing blocks and slots:

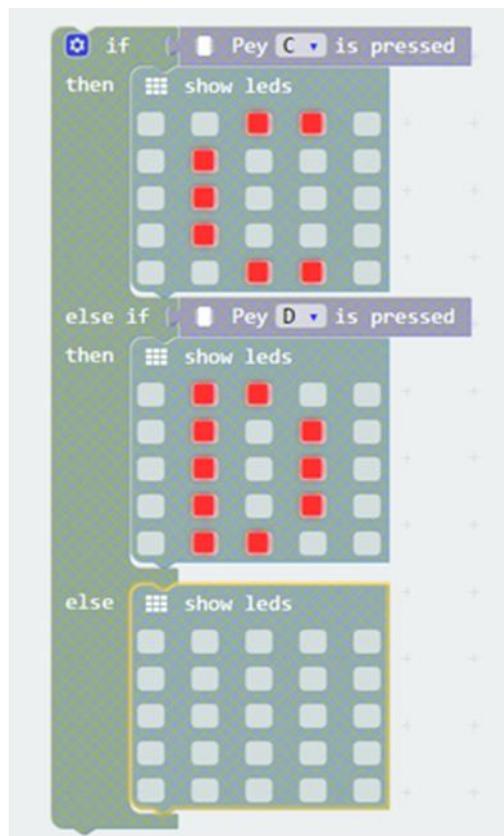
Move back those blocks or slots that are not needed in the programming area to the block selection area on the left.

### Experiment phenomenon

Touch the white piano keys from left to right, and in turn, DO, RE, MI, FA, SOL, LA, SI, DO. On each time you touch, the RGB LED lights up different colors. Of course, when you touch the key is in between of two white keys, it'll produce a tone higher than the key is on its left, and a tone lower than the key is on its right.

---

### 2 DETERMINE KEY PRESSING.



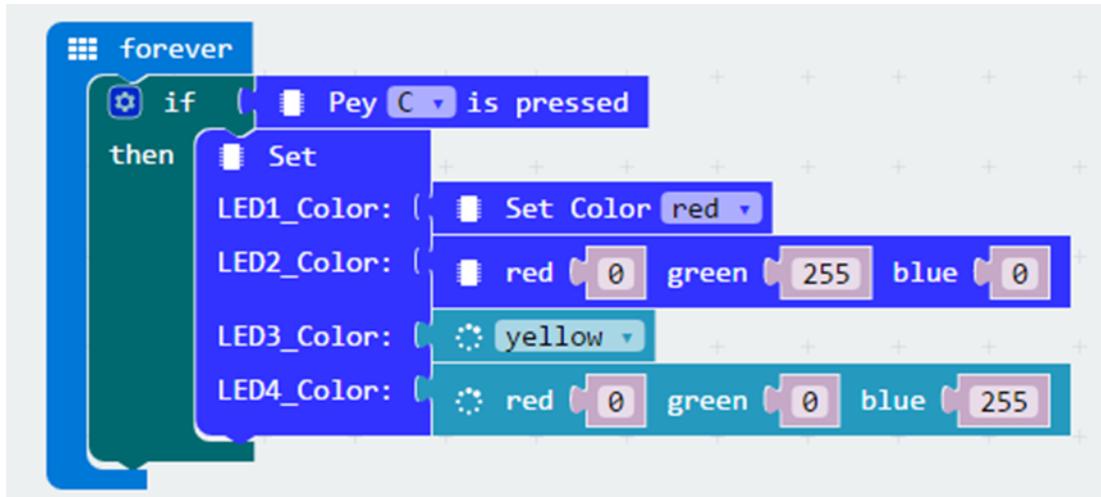
### Placement steps

Within "Logic" package, place "if then" slot to the right area. This block is used to make a logic evaluation. If a given event is true, "then" performs some action. On the "if then" slot, click the blue button next to "if". This will pop up a small box. Drag the "else if" slot from within this small box to the "if then" slot on the right. Go to Basic, and drag the "show leds" block to "then" inside, and click on the corresponding LED for red.

### Experiment phenomenon

When touching the piano key C, micro:bit will display “C”, if you touch D, then micro:bit will also display D, if nothing is touched, it won't displaying anything.

### 3 CONTROLLING RGB LED COLORS



#### Placement steps

Here we will omit some of the basics as they are mentioned above in this tutorial.

Drag the “set LED\_Color” 4-row block directly off the WSTouchPiano. Here the difference is at the back of each LED\_Color. Add to the back of each LED\_color block 4 new blocks as follows. LED1 and LED2, which are in WSTPPiano, and LED3 and LED4 by clicking “... More” just under Neopixel

Note that: right after red, green, blue there's a small field to fill in numbers. In fact this block is made of the composition of the three primary colors called RGB. Each color can set a brightness level of 256, within the range 0-255. By combining 3 different colors range, it can produce 256 x 256 x 256 different colors. You can look up color table on the Internet for each corresponding color combination.

#### Experiment phenomenon

When a piano key C is pressed, the four LEDs on the board will light in red, green, yellow and blue respectively.

## 4 CONTROLLING THE BUZZER



### Placement steps

Drag off the “play tone” block from “Music” and click the first parameter. A piano keyboard will show up. Click the Middle C. The second parameter is used for tone length, which is the beat.

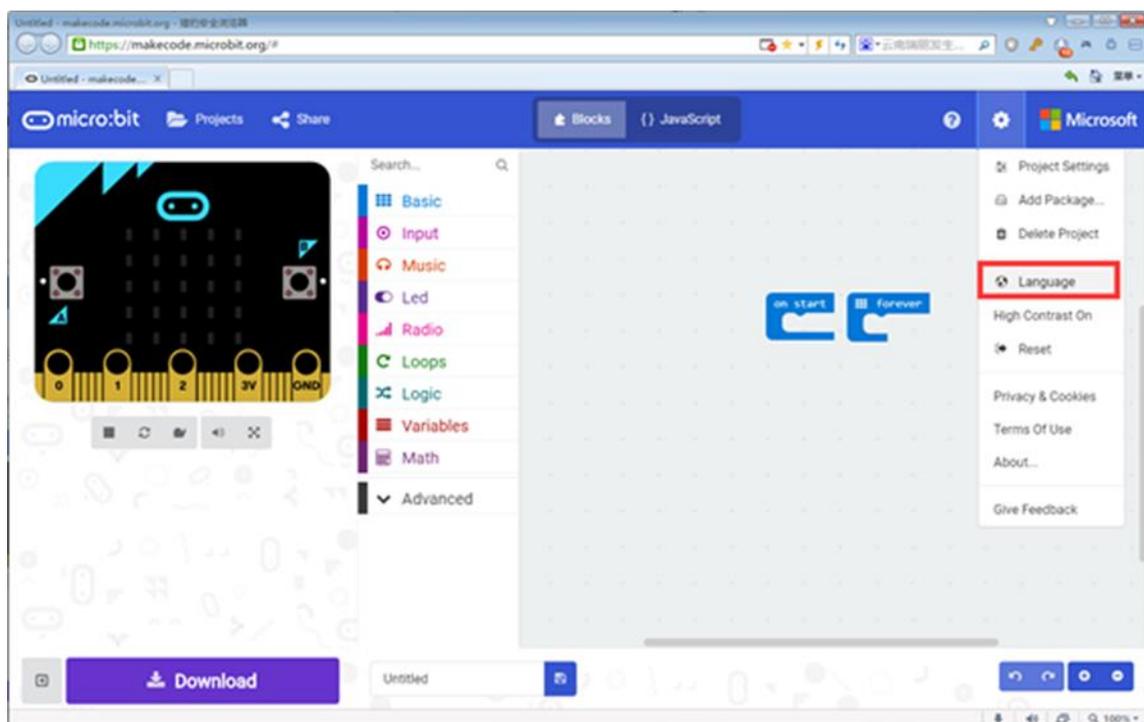
### Experiment phenomenon

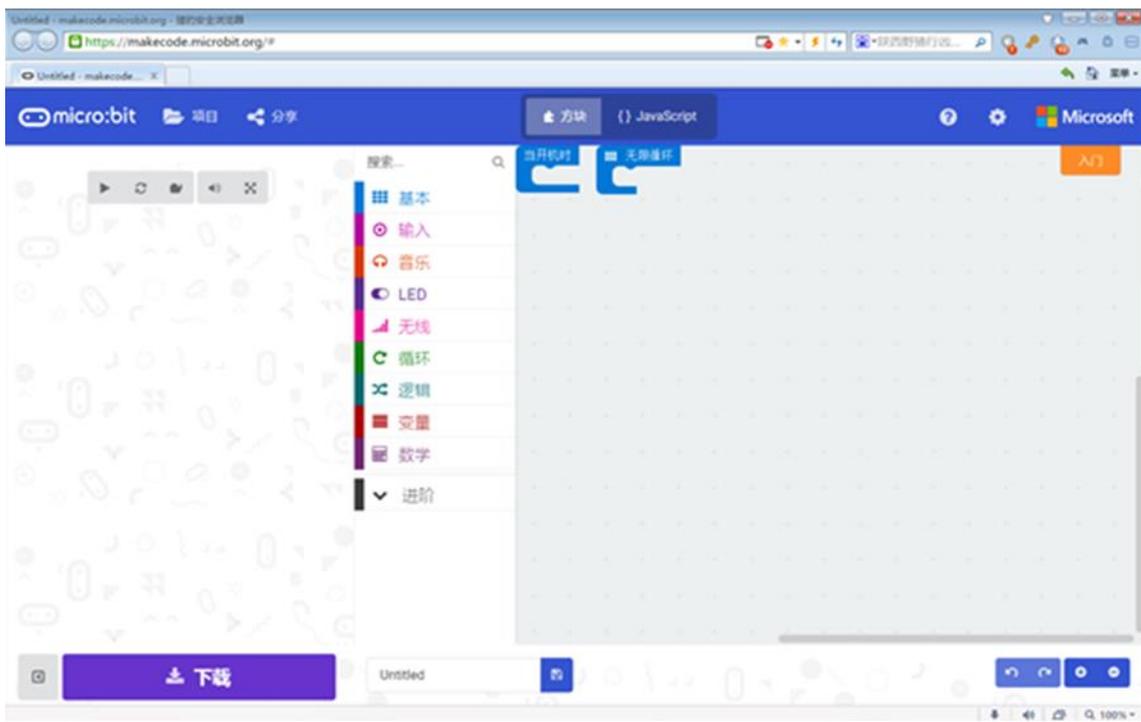
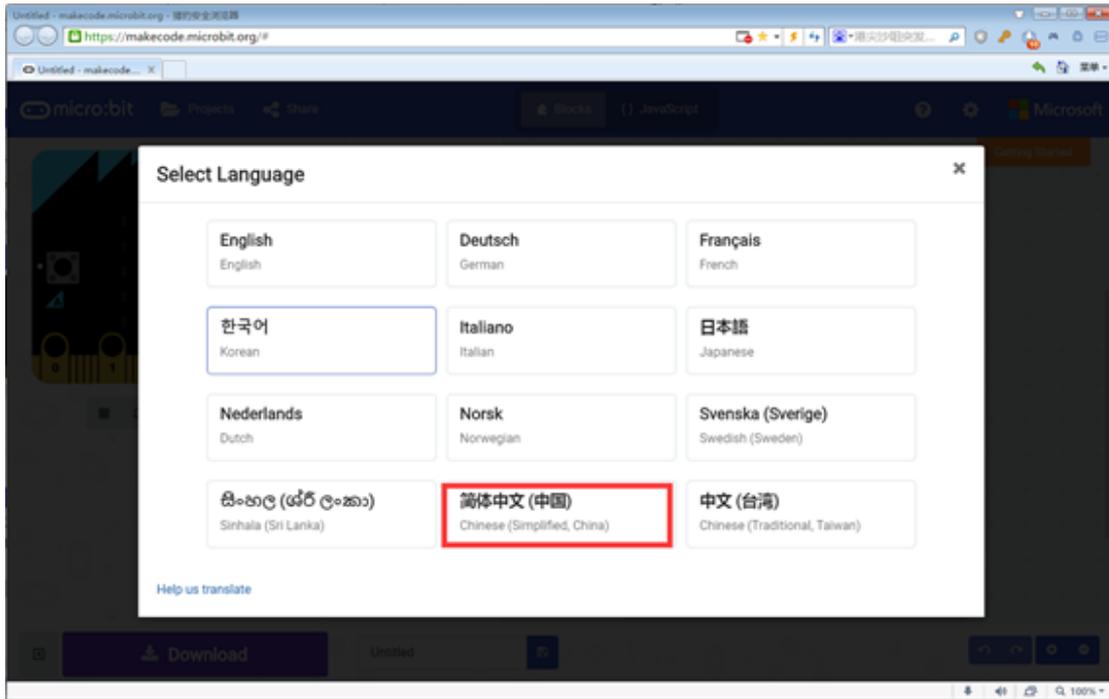
When the C key is pressed, the buzzer will ring one DO beat note.

## LANGUAGE

If you are not used to read in English, you can set to other language like Chinese as follows:

At the top, click on the “gear symbol” -> Language, in the pop-up window, scroll down and select simplified Chinese (China). That will set up the interface to Chinese language.

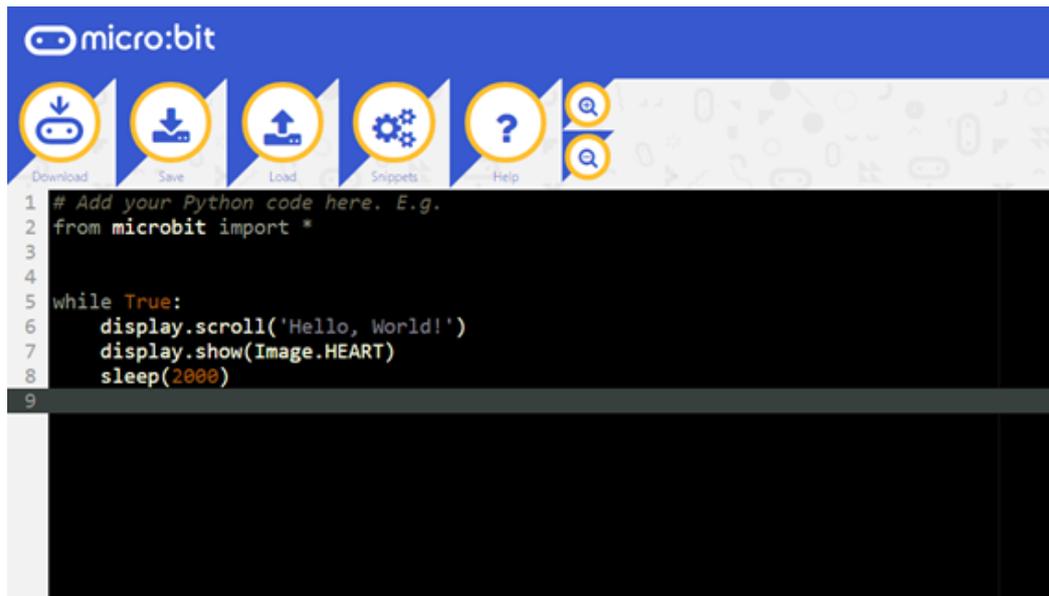




## USING THE PYTHON

Visit the official python programming website of micro:bit: <http://python.microbit.org/v/1>

You can see the following interface.



In turn, written above you can see the corresponding picture:

**Download:** Download the program to micro:bit.

**Save:** Save the python program to the computer.

**Load:** upload the local python program to the editor.

**Snippets:** These are the definitions and usages of common statements.

**Help:** Ask for Help with a more detailed description of the editor's usage.

You can download the program from the product information corresponding to our website, and upload the WSTouchPiano is in the folder to the editor.

Within Visual Programming there are the following functions.

**TouchPress()** : Listens to the key press and return the corresponding key value.

**ShowRGB (RGB):** Displays the color of four corresponding lights, where RGB are four tuples with three elements each, that is (0, 255, 255), (0, 255, 255), (0, 255, 255), (0, 255, 255).

**PlayPiano():** Keeps listening to key pressing. When the key is pressed, the four RGB lights up any random colors, as well as the buzzer emits the corresponding sound at the same time.

For learning more about control functions, please refer to micro:bit official python library, If you don't know the python library, you can find micro:bit's python library in the following places:

<https://github.com/bbcmicrobit/micropython>

<http://microbit-micropython.readthedocs.io/en/latest/index.html>