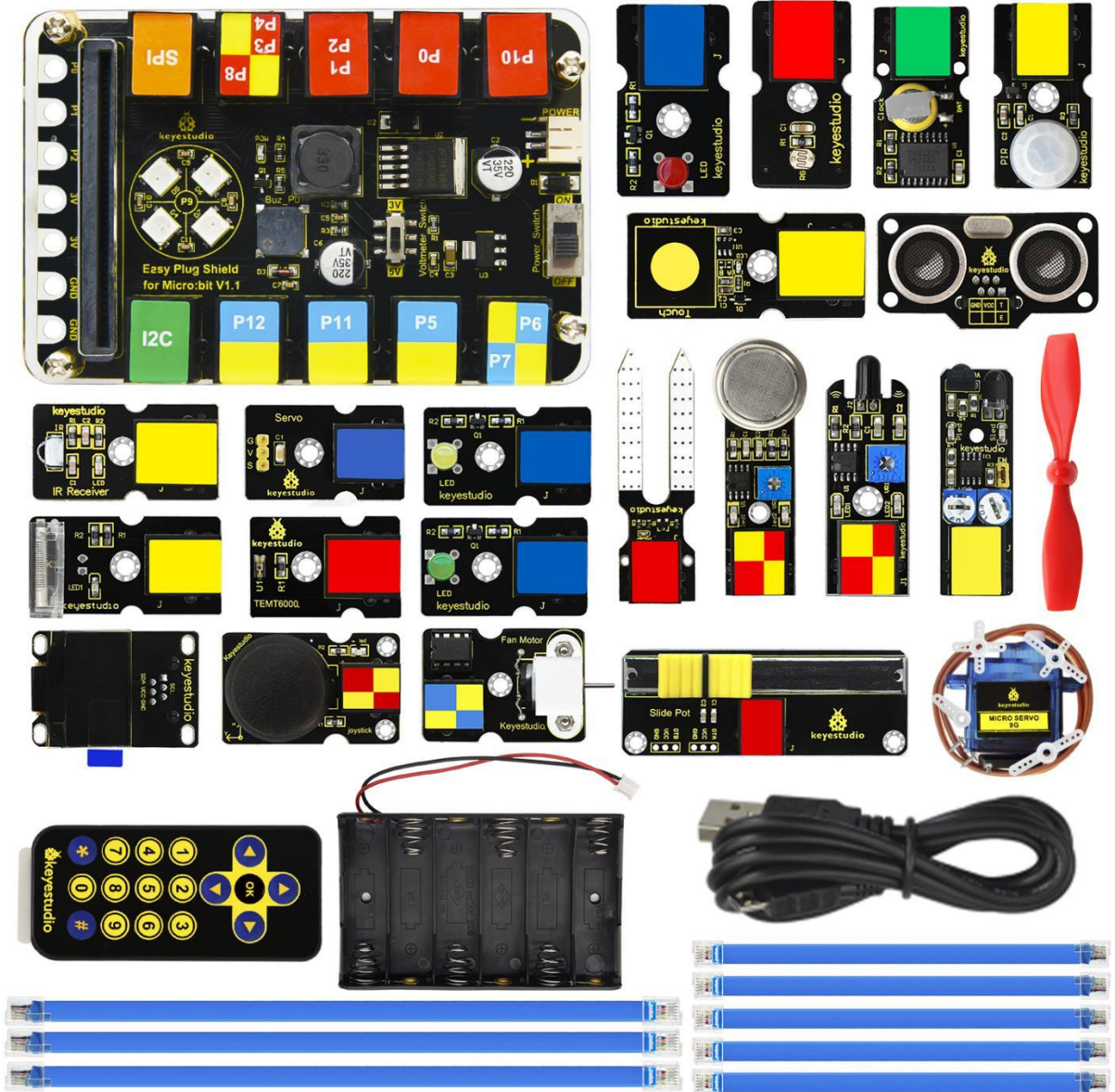


# keystudio

## EASY Plug Ultimate Starter Kit for BBC Micro:bit STEM EDU



# CONTENT

1. Kit Description .....	4
2. Kit List .....	4
3. Micro:bit .....	8
3.1. Install Drive of Micro:bit Board .....	12
4. Code and Programming .....	15
4.1 Makecode .....	错误！未定义书签。
4.2 Quick Download .....	22
4.3 Import Code .....	27
4.4 CoolTerm Installation .....	31
5. Projects .....	错误！未定义书签。
Project 1: Hello world .....	错误！未定义书签。
4.3 Import Code .....	错误！未定义书签。
4.4 CoolTerm Installation .....	错误！未定义书签。
8. Projects .....	35
Project 1: Hello world .....	35
Project 2: Heart beat .....	41
Project 3: Light Up a Single LED .....	43
Project 4: 5×5 LED Dot Matrix .....	47
Project 5: Micro:bit' s Buttons .....	53
Project 6: Temperature Measurement .....	57

Project 7: Micro:bit Compass .....	62
Project 8: Micro:bit Acceleration Sensor .....	72
Project 9: Detect Light Intensity by Micro:bit .....	79
Project 10: Bluetooth Wireless Communication .....	82
Project 11: LED Blink .....	98
Project 12: Breath .....	104
Project 13: Blink and Breath .....	107
Project 14: RGB .....	110
Project 15: Play Music .....	124
Project 16: Knock Sensor .....	130
Project 17: PIR Motion Sensor .....	135
Project 18: Capacitive Touch .....	140
Project 19: Obstacle Avoidance .....	144
Project 20: Servo .....	149
Project 21: Fan Module .....	154
Project 22: Light Brightness .....	159
Project 23: Fire Alarm .....	164
Project 24: Flammable Gas in the Air .....	167
Project 25: Ambient Light .....	173
Project 26: Slide Position .....	178
Project 27: OLED Display .....	183
Project 28: Automatic Water Your Plant .....	190
Project 29: Ultrasonic Ranging .....	197

Project 30: IR Remote Control Decoding .....	205
Project 31: IR Remote Control RGB .....	214
Project 32: Joystick .....	222
Project 33: What Time Is It? .....	226
6. Resource: .....	233

## **1. Kit Description**


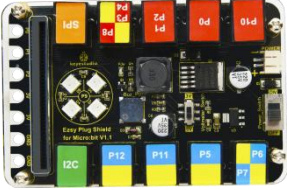

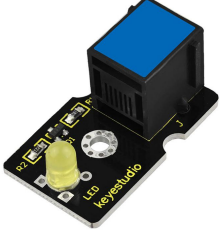

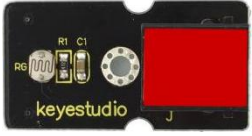
Micro:bit is significantly applied to STEM education for teenagers, as a small microcontroller, which features small in size, easy to carry, and powerful function. At present, innovative technology products, like robots, wearable devices and interactive electronic games can be produced by programming and code.





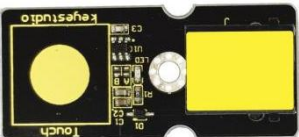
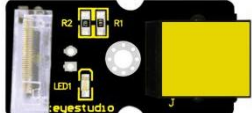

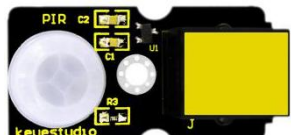
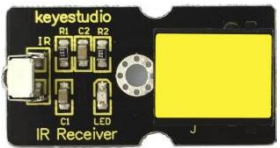
MakeCode is a framework for creating interactive and engaging programming experiences for those new to the world of programming. The platform provides the foundation for a tailored coding experience to create and run user programs on actual hardware or in a simulated target.





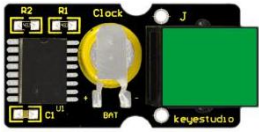



To make you deeply know the micro:bit, we also provide test code and projects.

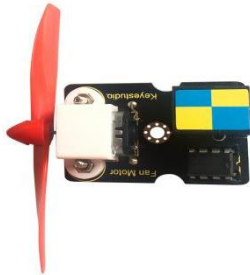




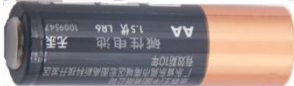
This ultimate starter kit incorporates different sensors and modules such as passive buzzer, 1602 LCD module, RGB, flame sensor and so on. The detailed projects, from simple to difficult will spur your inspiration and bring in the magical programming world.

## **2. Kit List**

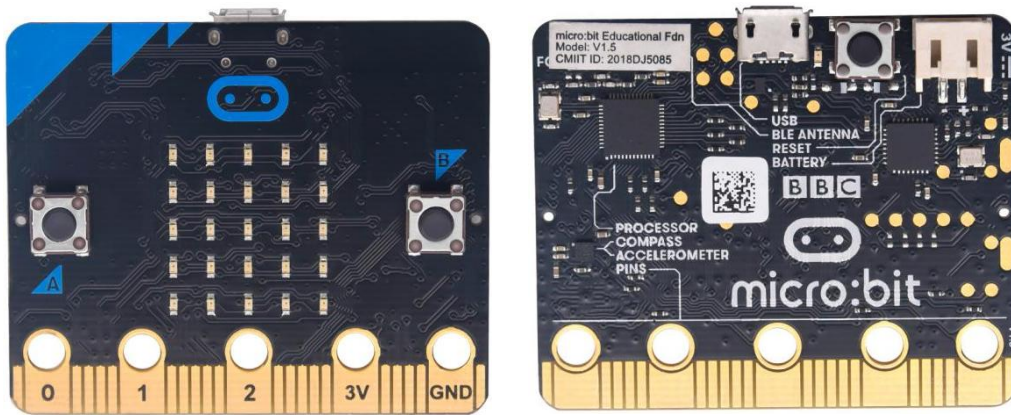
#	Component	QTY	Picture
0	Micro:bit main board <b>is not included</b> in KS4022 Kit		
0	Micro:bit main board <b>is included</b> in KS4023 Kit	1	
1	EASY Plug Shield for micro bit V1.1	1	
2	EASY Plug Green LED Module	1	
3	EASY Plug Yellow LED Module	1	
4	EASY Plug Red LED Module	1	
5	EASY Plug Photoresistor	1	

6	EASY Plug Soil Humidity Sensor	1	
7	EASY Plug Analog Gas Sensor	1	
8	EASY Plug TEMT6000 Ambient Light Sensor	1	
9	EASY Plug Slide Potentiometer Module	1	
10	EASY Plug Capacitive Touch Module	1	
11	EASY Plug Knock Sensor	1	
12	EASY Plug Flame Sensor	1	
13	EASY Plug PIR Motion Sensor	1	
14	EASY Plug IR Receiver	1	

15	Keystudio IR Remote Control	1	
16	EASY Plug Servo Module	1	
17	Keystudio 9G Servo 90°	1	
18	EASY Plug IR Obstacle Avoidance Sensor	1	
19	EASY Plug DS3231 Clock Module	1	
20	EASY Plug Joystick Module	1	
21	EASY PlugSR01 Ultrasonic Sensor	1	
22	EASY Plug OLED Module	1	

23	EASY PlugL9110 Fan Module	1	
24	200mm Blue RJ11 Cable	5	
25	300mm Blue RJ11 Cable	3	
26	Micro USB Cable	1	
27	6-Slot AA Battery Holder	1	
28	1.5V AA Battery	6	

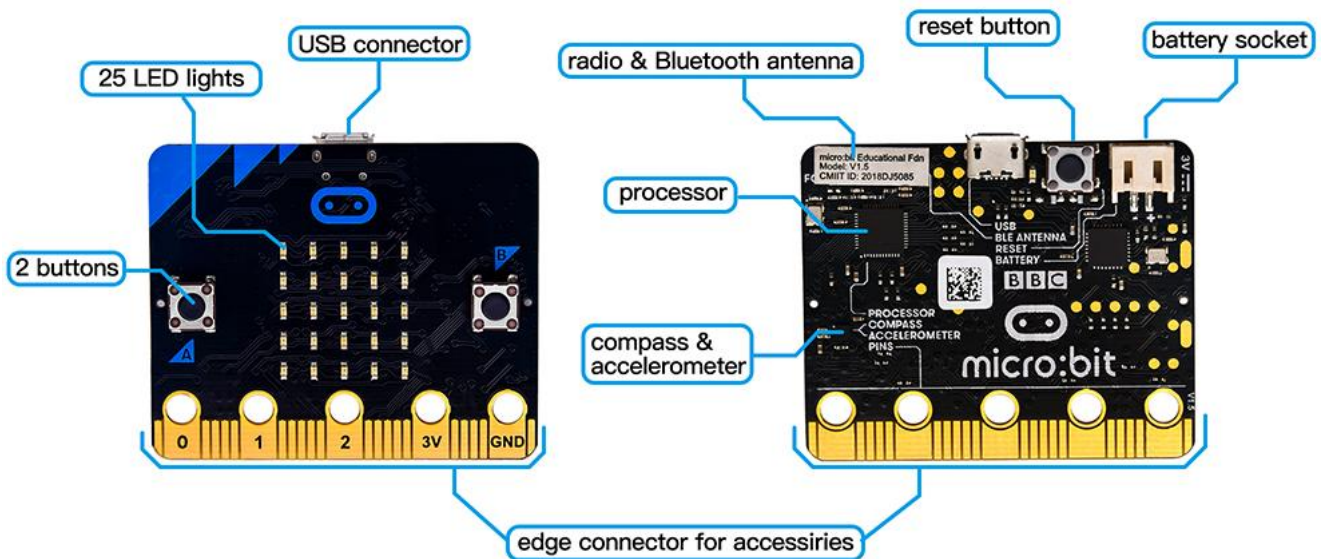
### 3. Micro:bit



For more details, enter the website please:

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

## Function:



## Features:

- NRF51822 processor (16 MHz 32bit, ARM Cortex-M0, Bluetooth 4.0 low consumption/2.4GHz RF wireless, 16kB RAM and 256kB Flash)

- KL26Z micro controller (48 MHz ARM Cortex-M0+ core,128 KB Flash)
- 25 pcs programmable LEDs
- 2 programmable buttons
- Physical pins
- Light and temperature sensors
- Accelerometer(MMA8652 and I2C get the data from accelerator sensor)
- Geomagnetic sensor/compass (MAG3110, I2C obtain three-axis geomagnetic data)
- Wireless communication, by radio and Bluetooth.

### **Micro USB port**

More details, please enter website:

<https://microbit.org/guide/features/>

### **Hardware:**

The details of micro:bit board: <https://tech.microbit.org/hardware/>

### **Micro:bit Pins:**

Before getting started with the following projects, firstly need to figure out each pin of micro:bit main board.

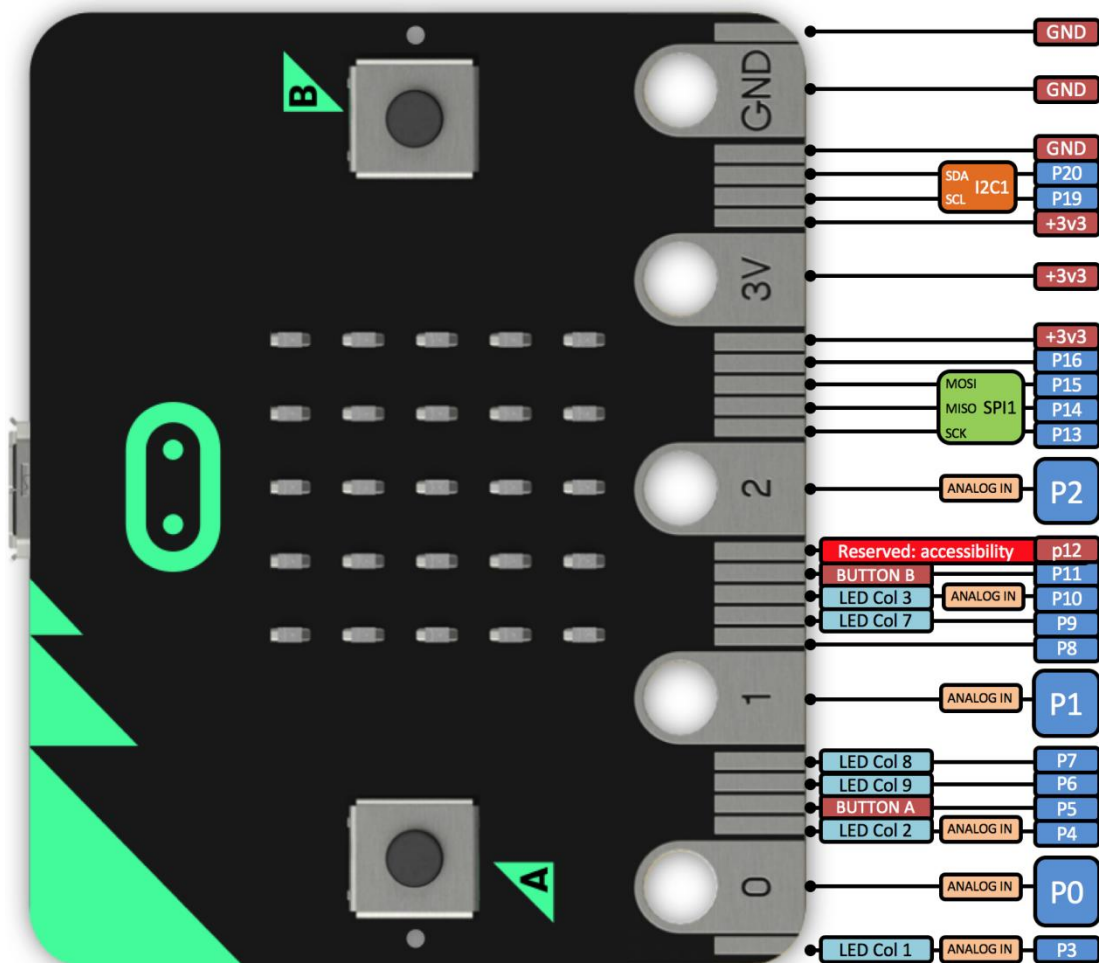
The BBC micro:bit has 25 external connections on the edge connector of the board, which we refer to as “pins” .

The edge connector is the gray area on the right side of the figure below.

There are five large pins, that are also connected to holes in the board labeled: 0, 1, 2, 3V, and GND.

And along the same edge, there are 20 small pins that you can use when plugging the BBC micro:bit into an edge connector.

Please refer to the following diagram shown below.



More details you could refer to the official website:

<https://microbit.org/guide/hardware/pins/>

### 3.1. Install Drive of Micro:bit Board

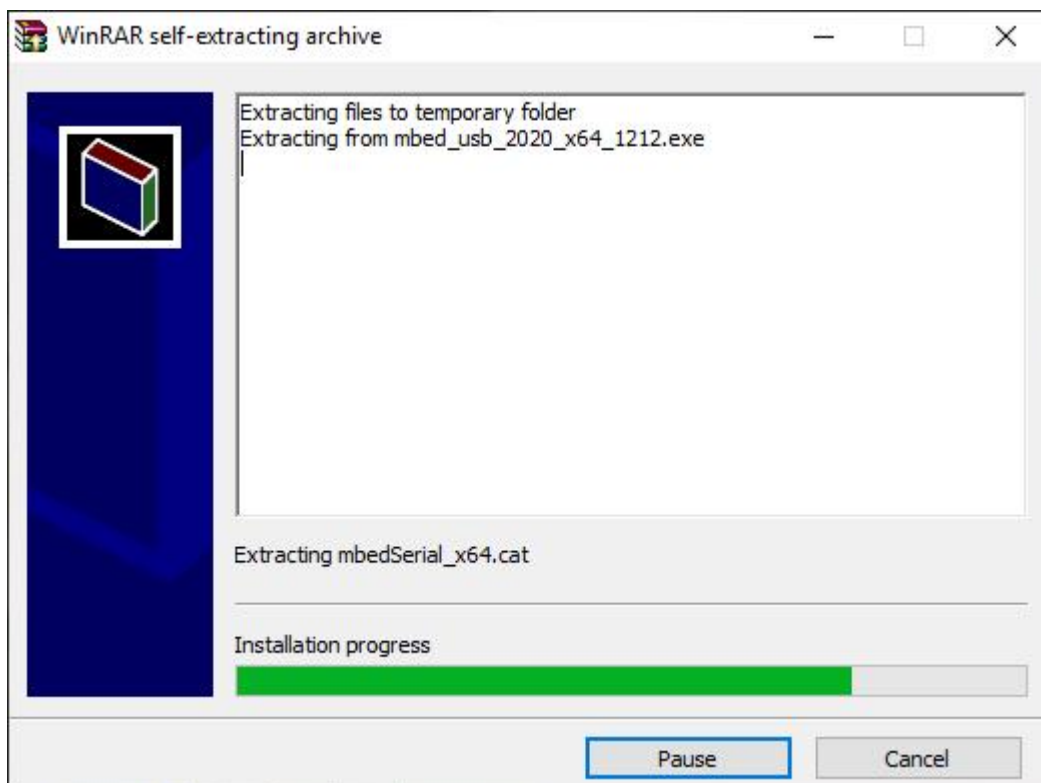
First of all, connect the micro:bit to your computer using a USB cable.

And enter website <https://fs.keyestudio.com/KS4022-4023> to download the

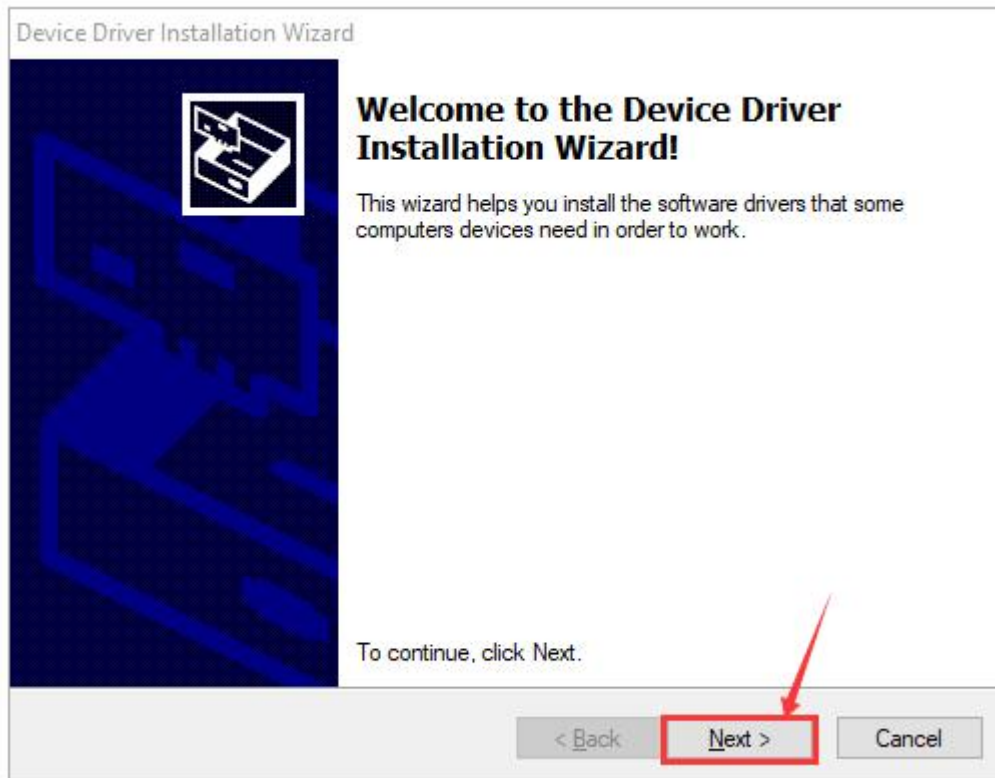


mbed\_usb\_202  
0\_x64\_1212.exe

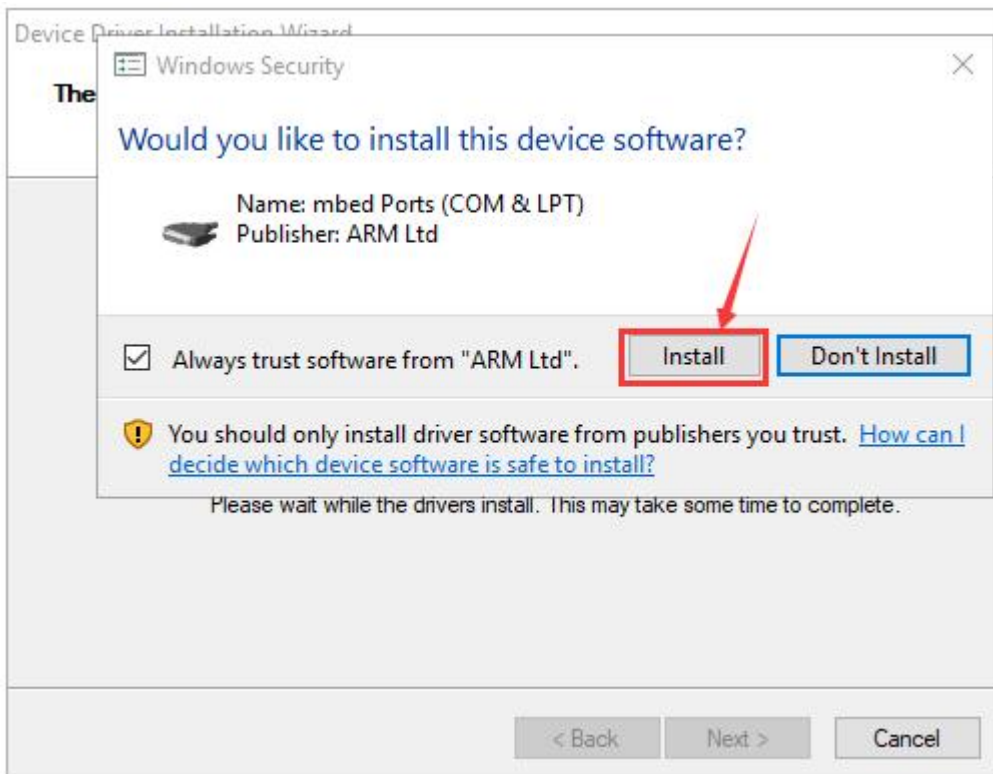
driver file of micro:bit, double-click

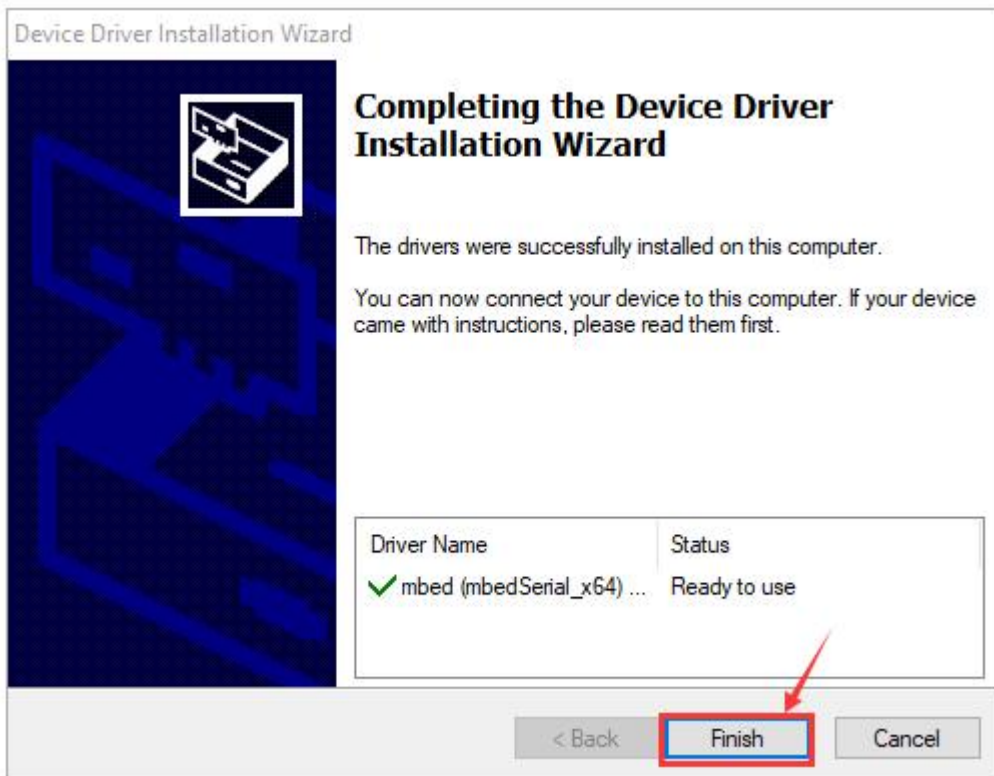


After downloading the driver, then click "Next"

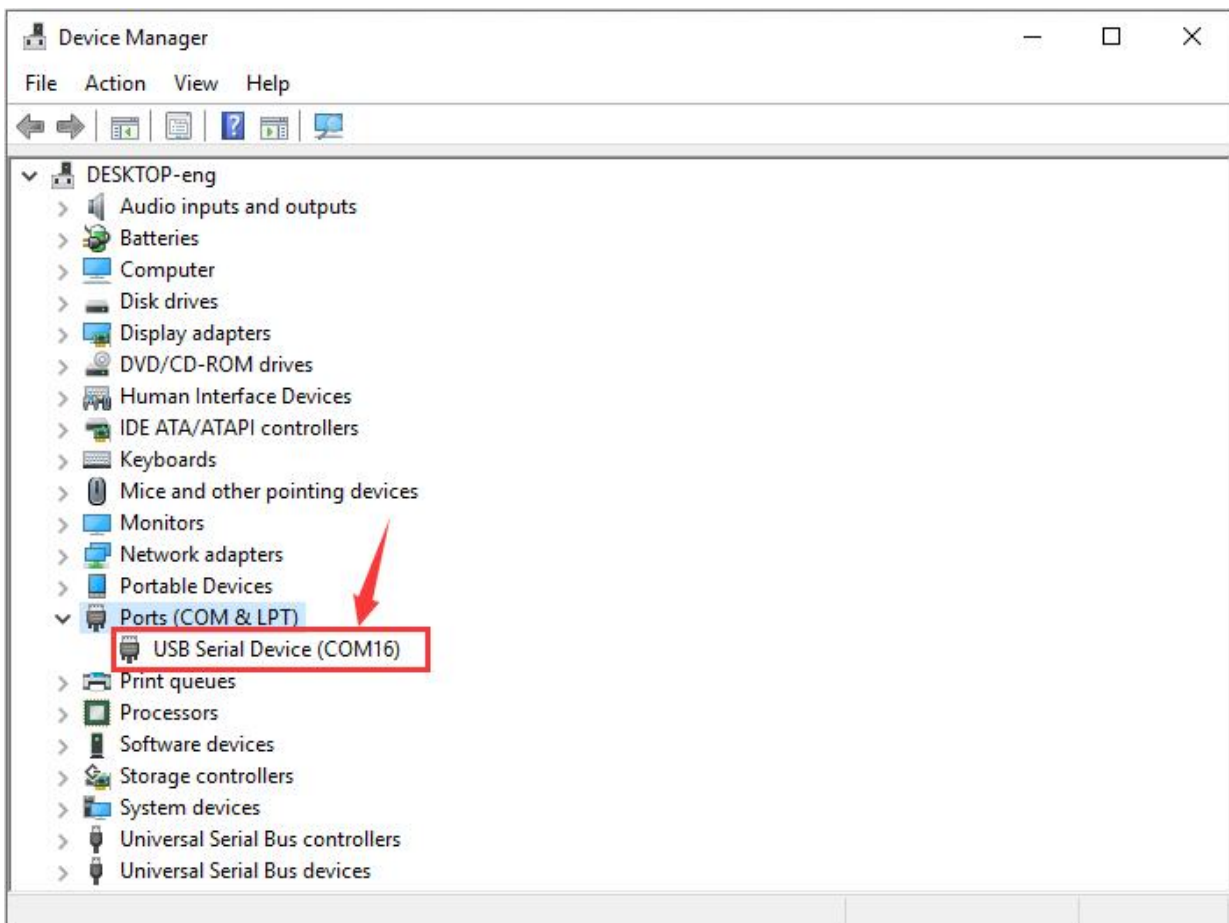


The following page pops up and click "install"





Tap "Finish" . Then click "Computer" —> "Properties" —> "Device manager" , as shown below.



## 4. Code and Programming

We take Windows system as example to show you.

Get started with Micro:bit: <https://microbit.org/guide/quick/>

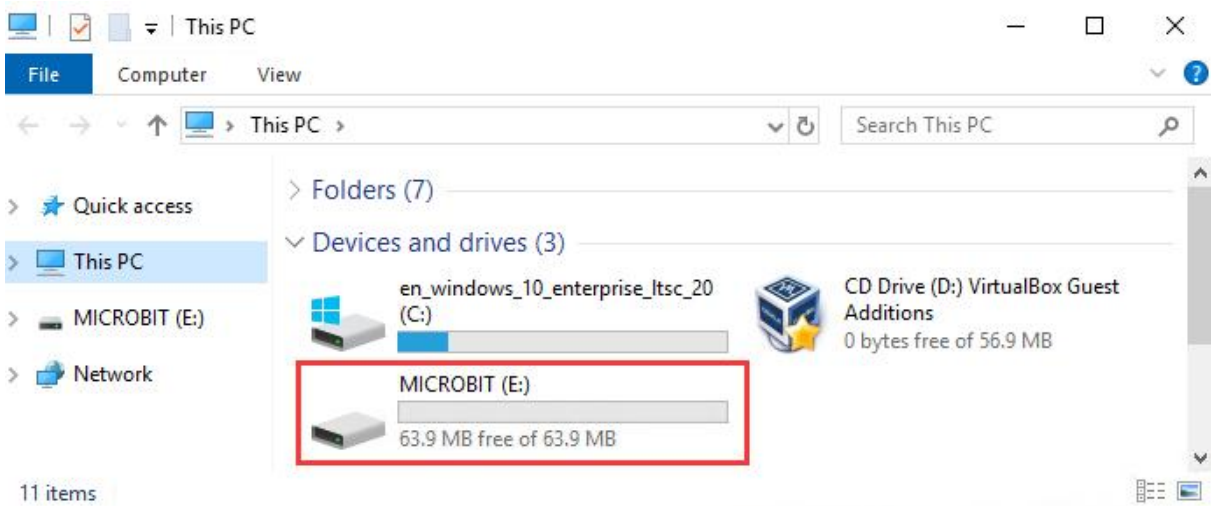
### Step 1: Connect Micro: bit Board

Link micro:bit board to computer with USB cable. (Guide to mobile apps: <https://microbit.org/get-started/user-guide/mobile/>)

Macs ,PCs, Chromebooks and Linux system (including Raspberry Pi) support micro: bit.

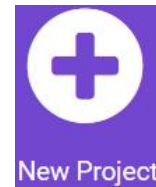


There will be a MICROBIT drive in your computer, as shown below:



## Step 2: Programming

Enter <https://makecode.microbit.org/> then click

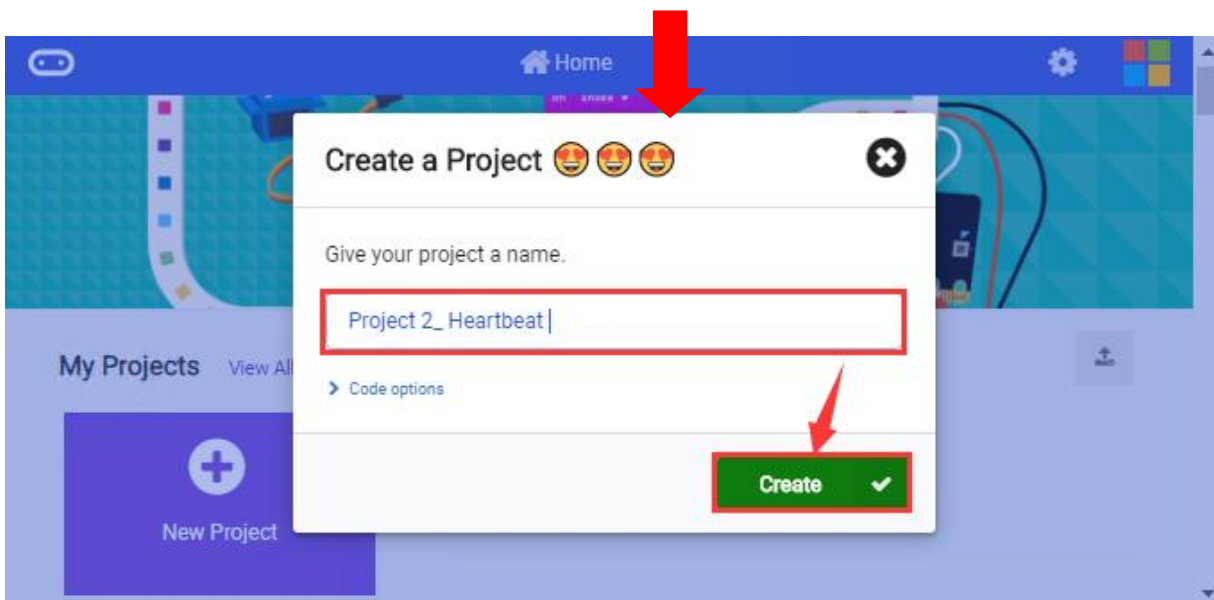
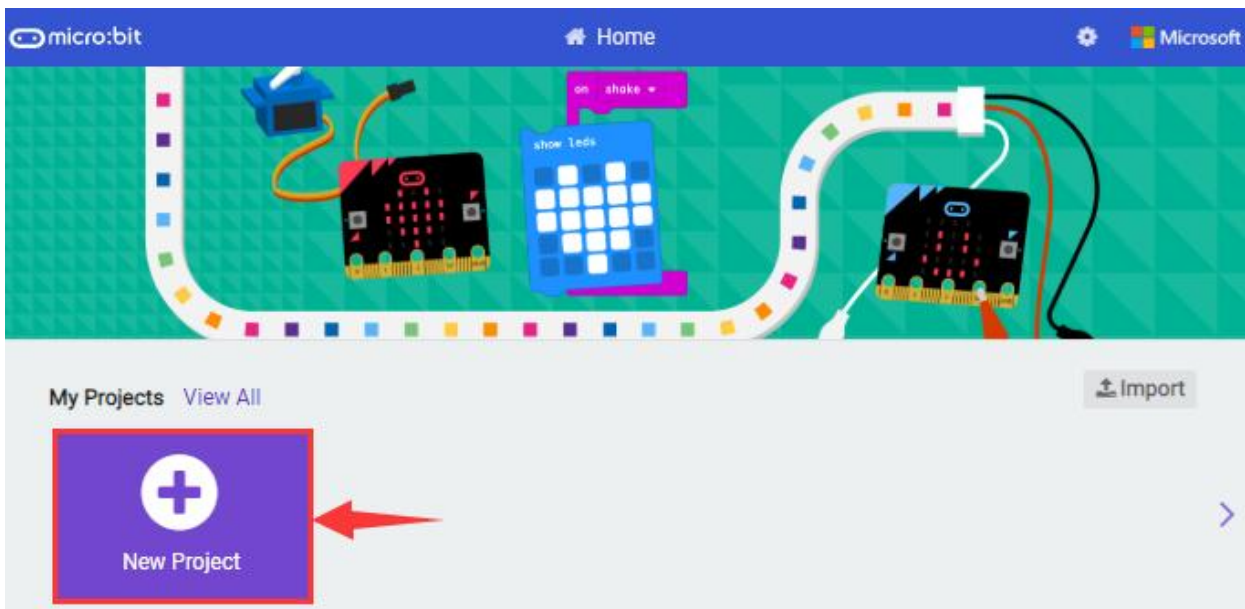


and you

will view the dialog box.

We recommend you to use Google Chrome

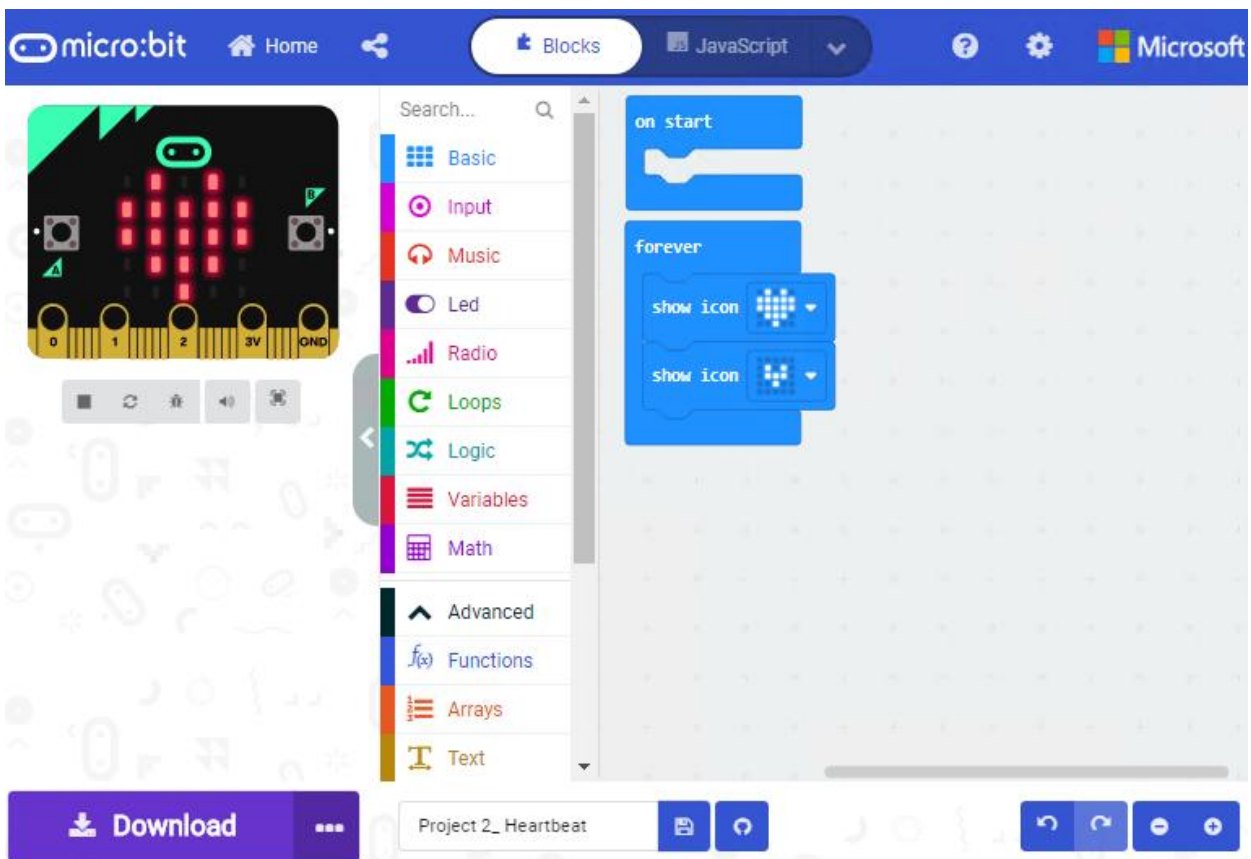
Input "Project 2\_Heartbeat" to name your project and click "Create"



Through MakeCode editor, you just need to drag blocks into code editing area to design program. There is a video to show you how to finish "heartbeat" pattern.

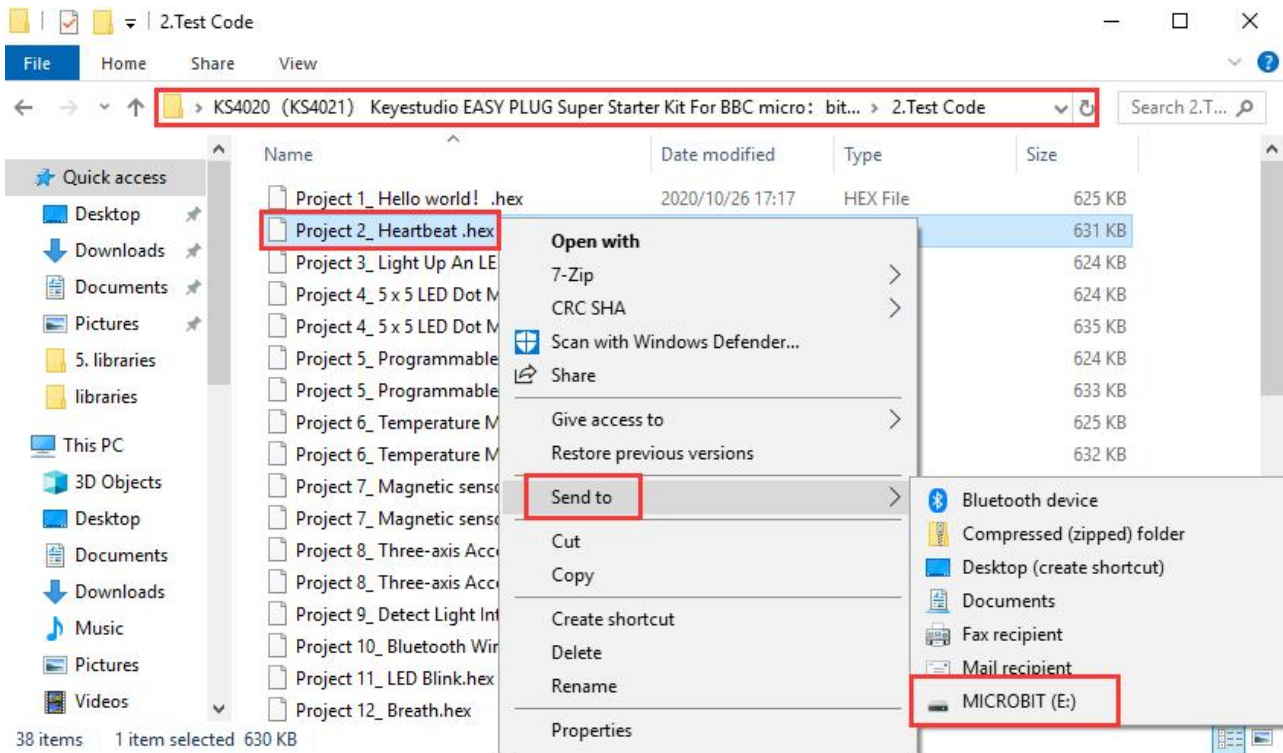
[microbit-heartbeat.mp4](#)

Next, we will introduce Makecode.



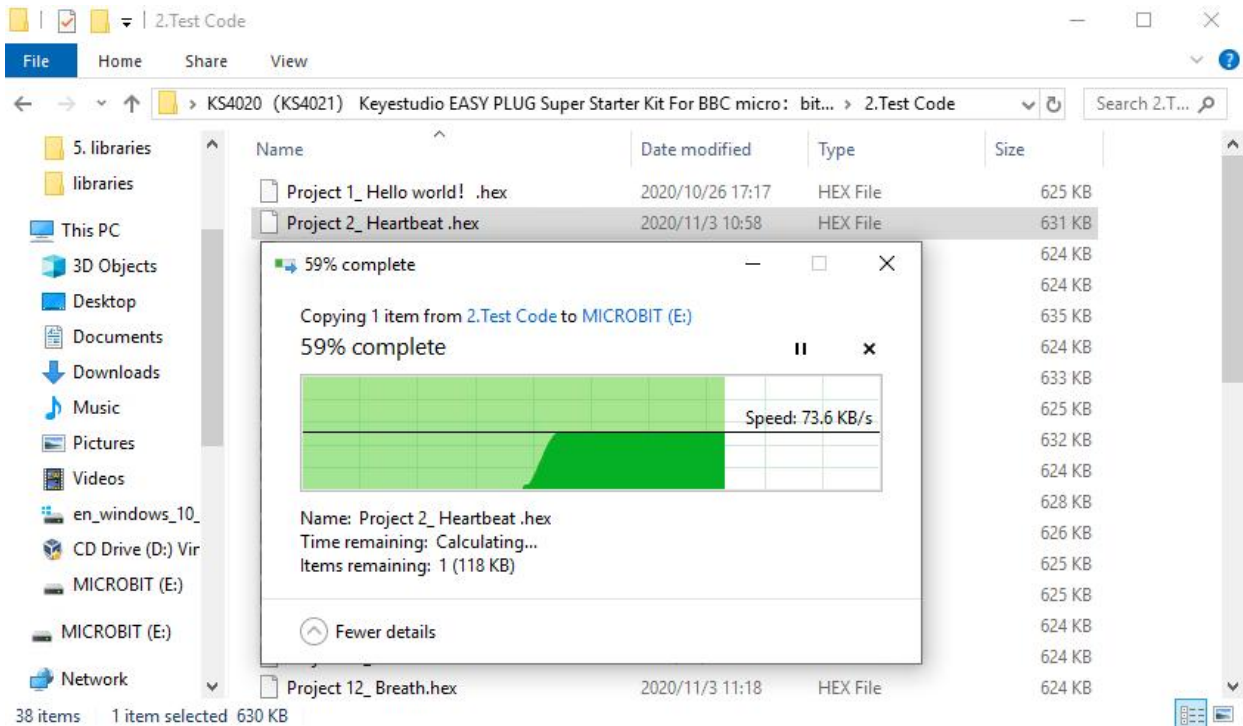
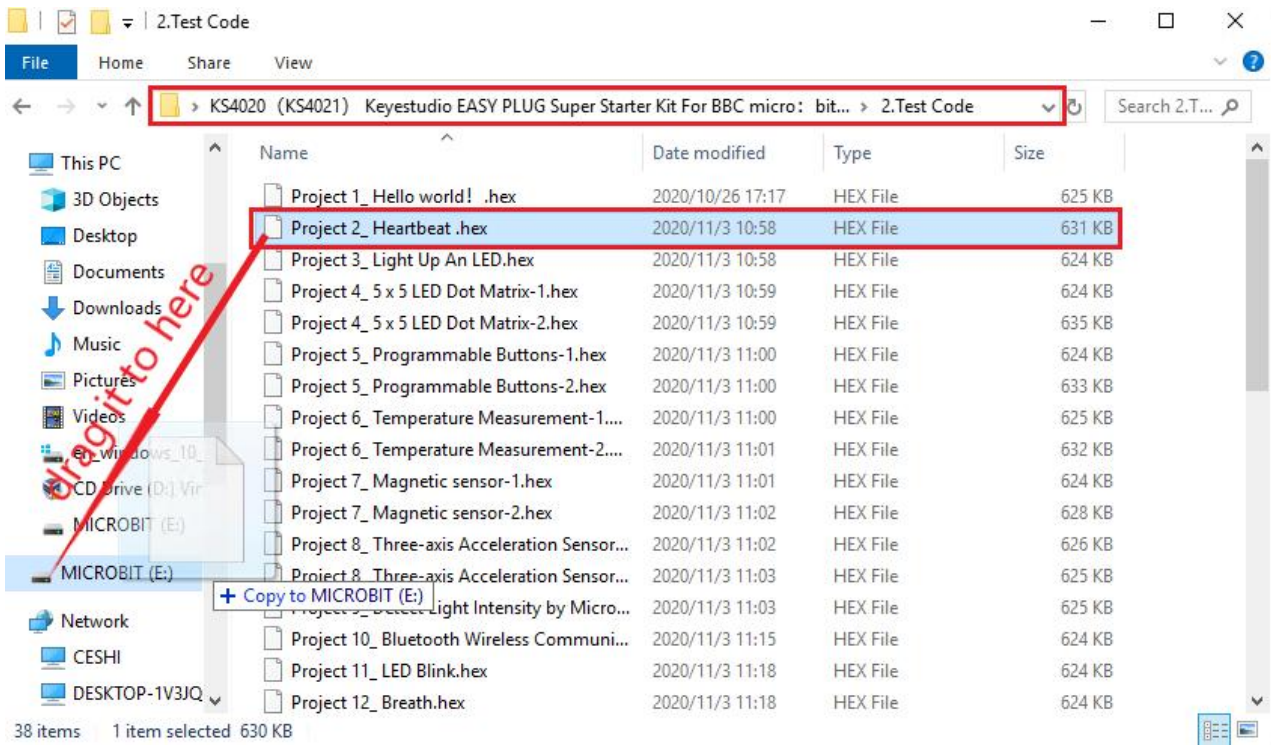
### Step 3: Download Code

Enter Makecode editor, tap “Download” to get a “hex” file. Then copy it into MICROBIT drive.



Or you could find out "hex" file firstly and right-click to select "Send to" MICROBIT (E) " .

Then hex file will be copied on MICROBIT drive.

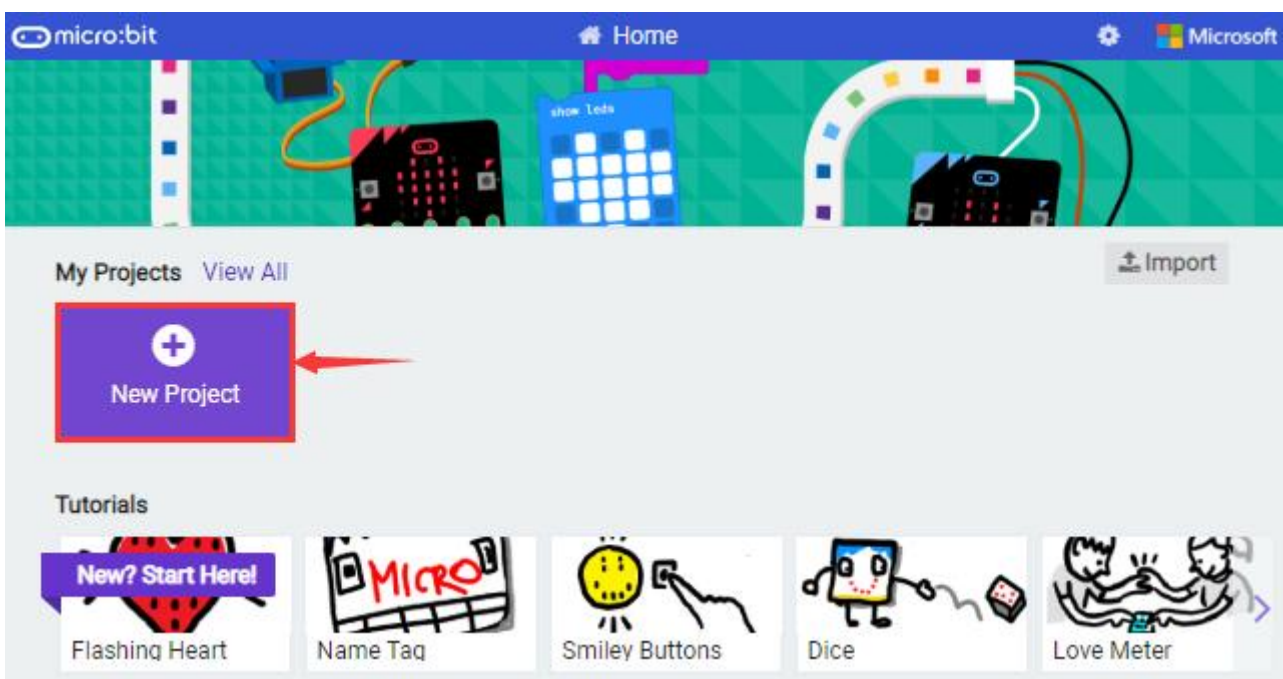




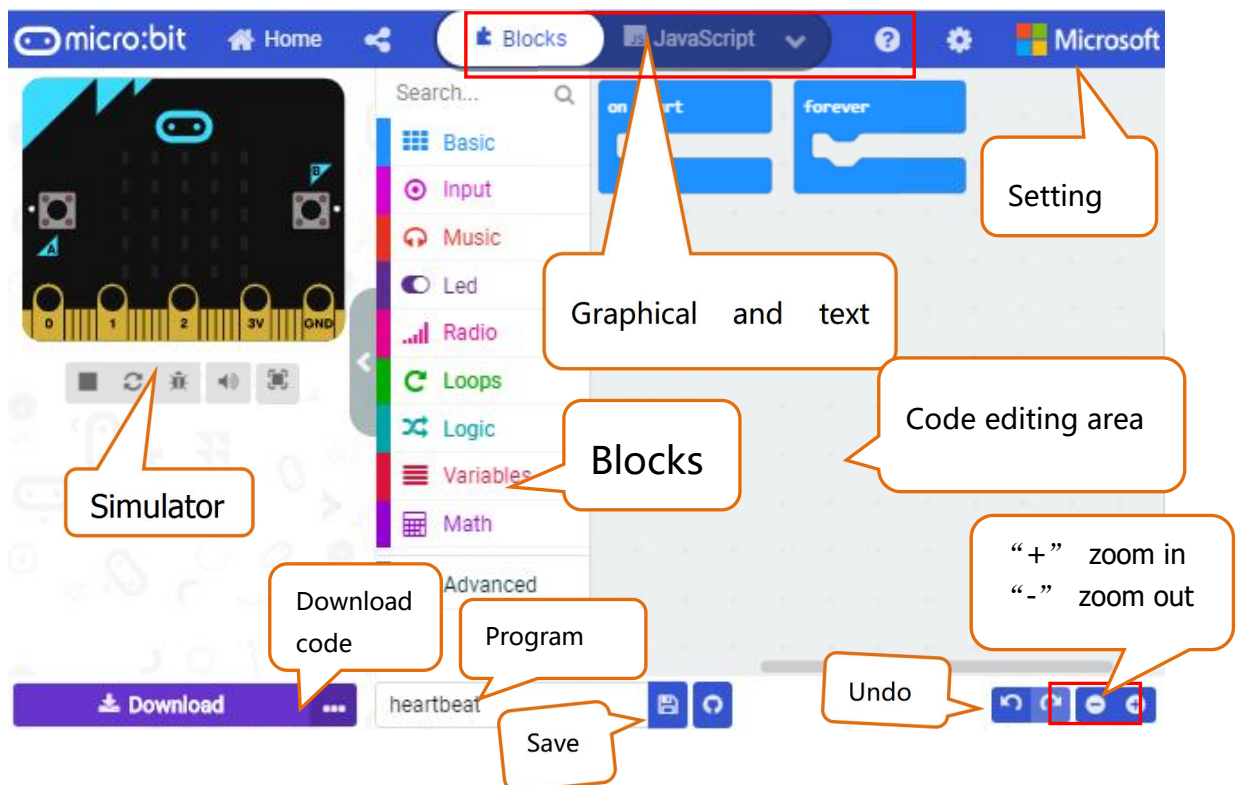
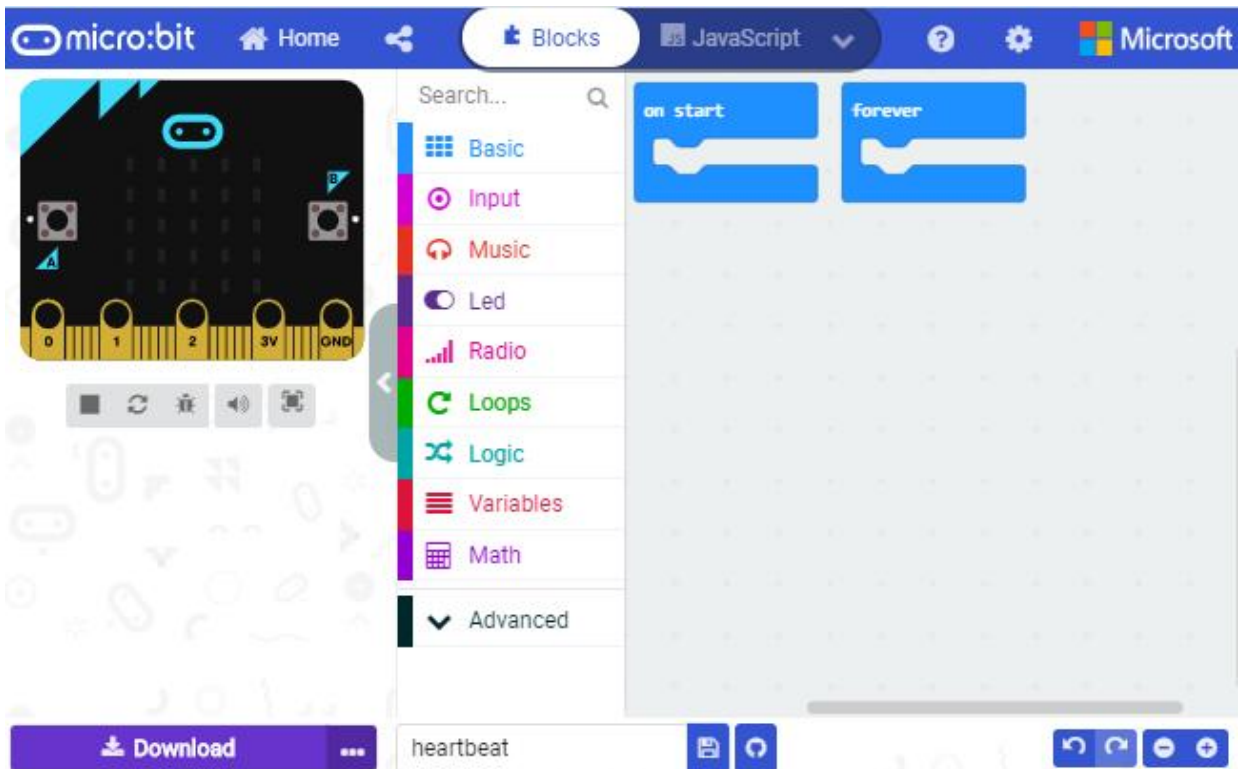
<https://microbit.org/projects/>

## 4.1 Makecode

Browse <https://makecode.microbit.org/> and enter Makecode online editor.



Click "New Project" , and input "heartbeat" , then enter Makecode editor, as shown below:



There are block "on start" and "forever" in the code editing area.

When plugging in power or resetting, “ on start ” means that command blocks in the code are only executed once, “ forever ” implies that code will runs cyclically.

## 4.2 Quick Download

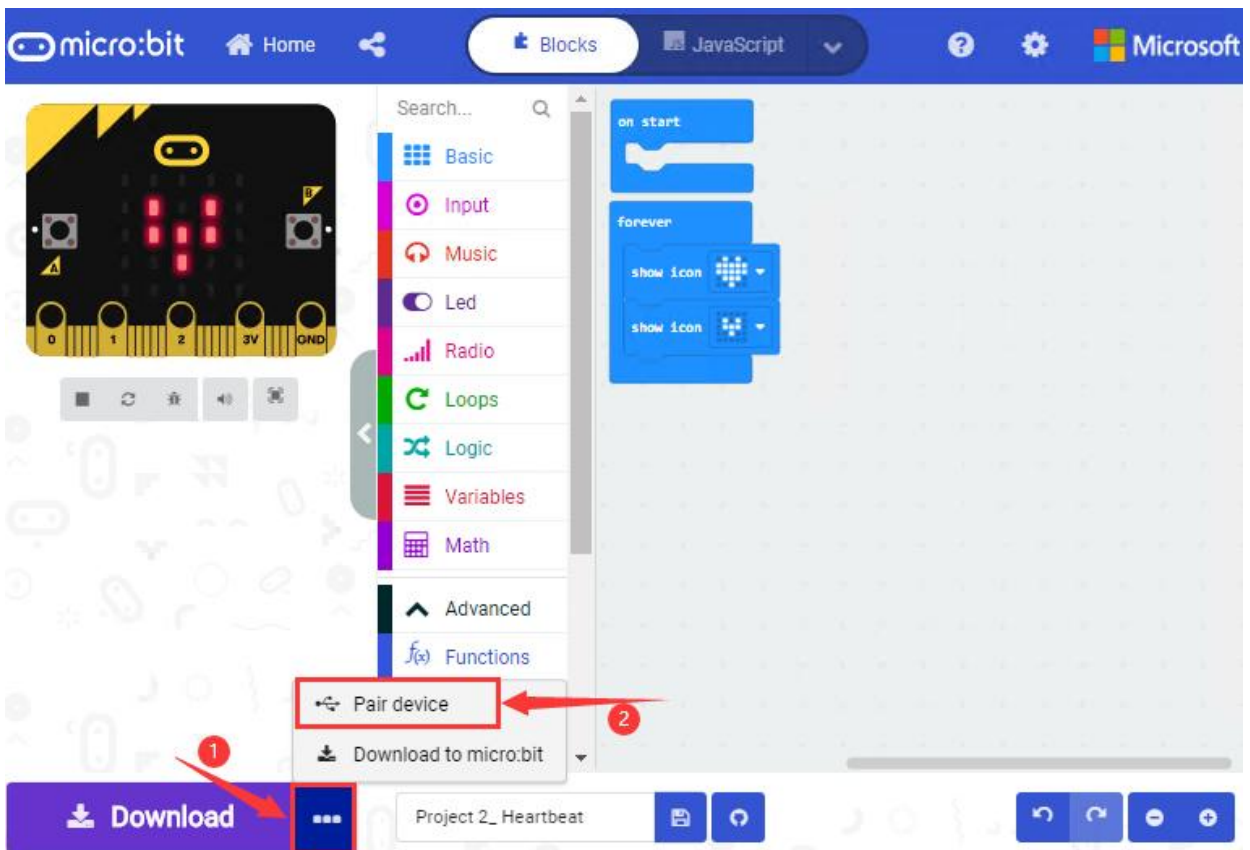
If you use Google Chrome on Android, ChromeOS, Linux, macOS and Windows 10 system, you will achieve the quick download.

We use the webUSB function of Chrome to allow the internet page to access the hardware device connected USB.

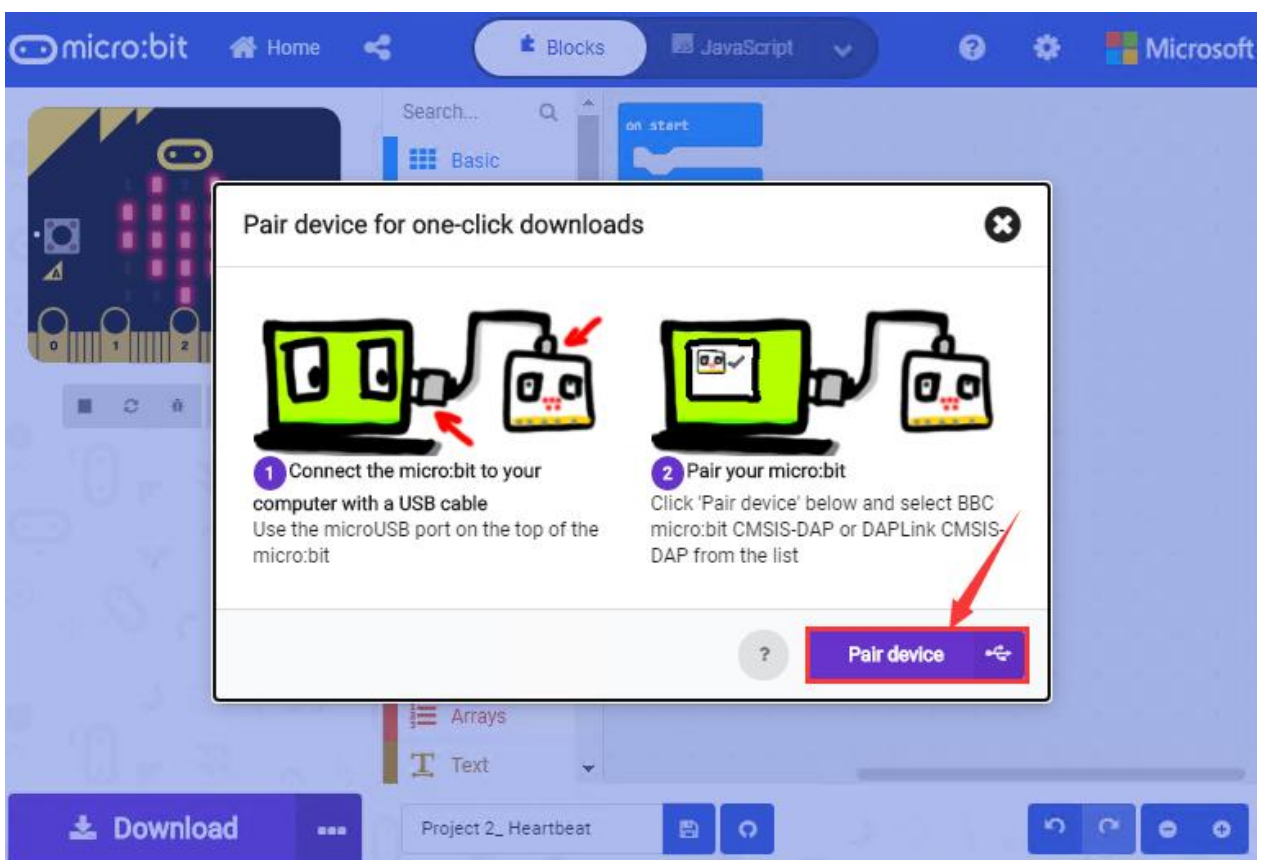
You could refer to the following steps to connect and pair device.

### Pairing device

Connect to computer by USB cable. Click “...” beside “Download” and click “Pair device” .



Continue to tap "Pair device"





Then select the device you want to connect and "connect" in the

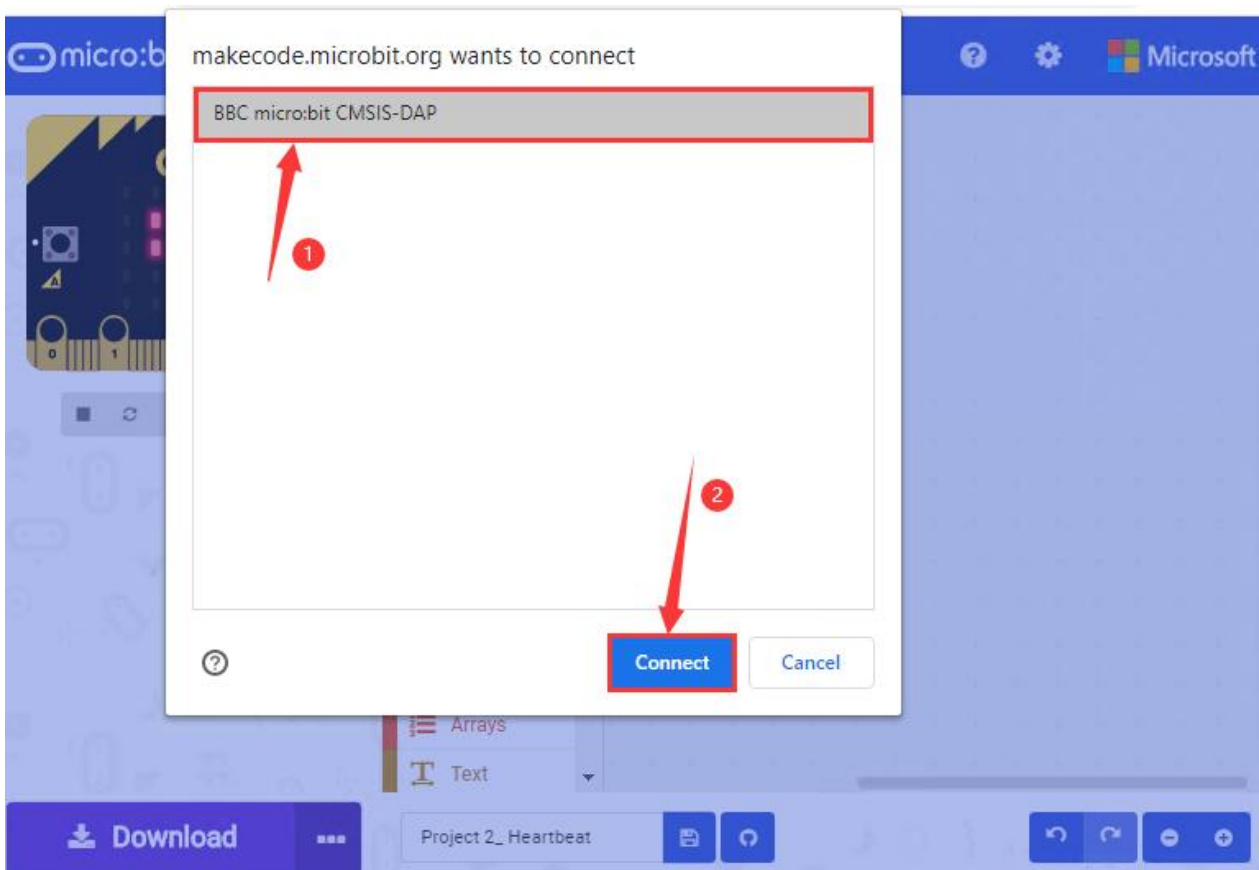
window popping up

If there is no device in the window, please refer the following link:

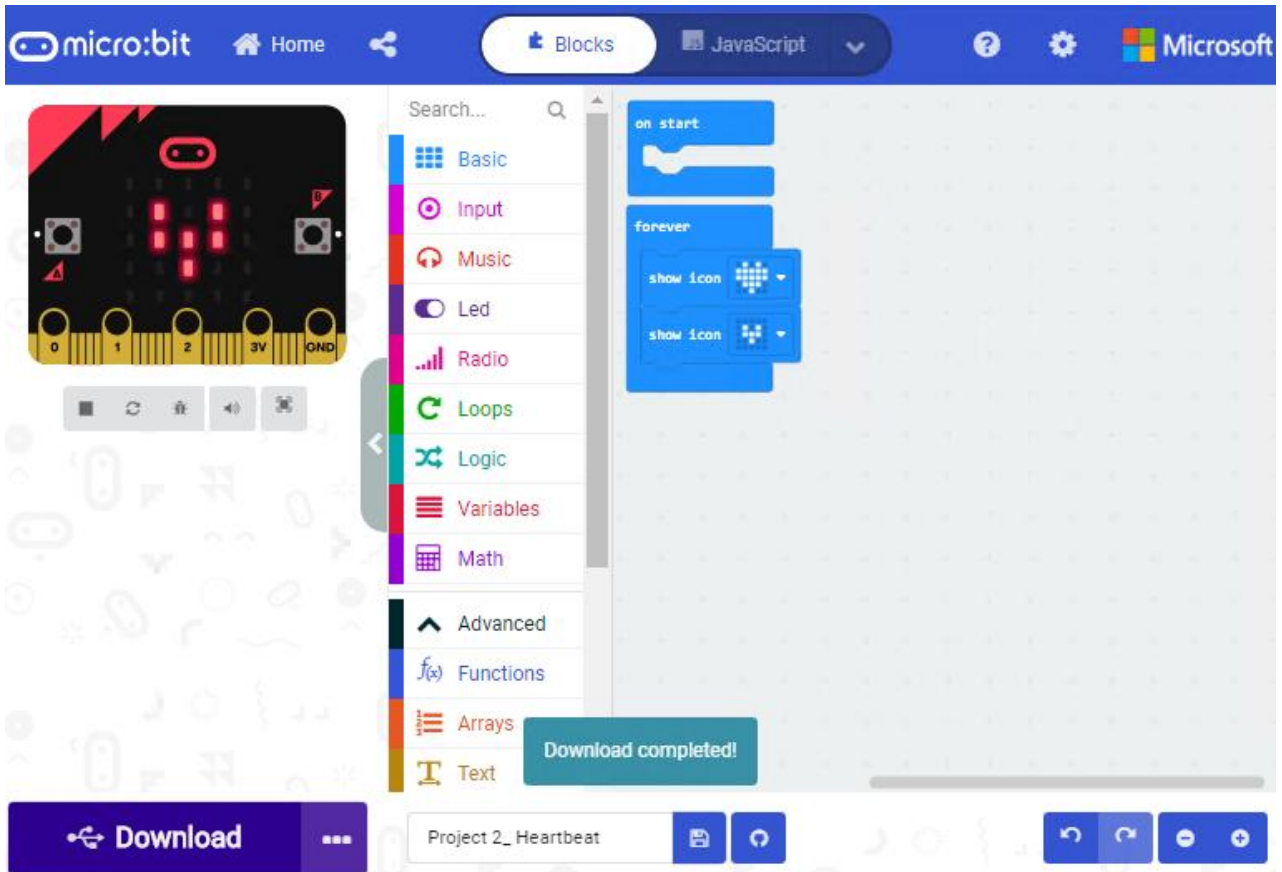
<https://makecode.microbit.org/device/usb/webusb/troubleshoot>

We also provide  6. Troubleshooting-WebUSB in the resource link.

What' s more, if you don' t know how to update the firmware of micro:bit, refer the link: <https://microbit.org/guide/firmware/> or browse  4.Upgrad the Firmware we provide.



After connecting successfully, press buttons and download program to micro:bit board.

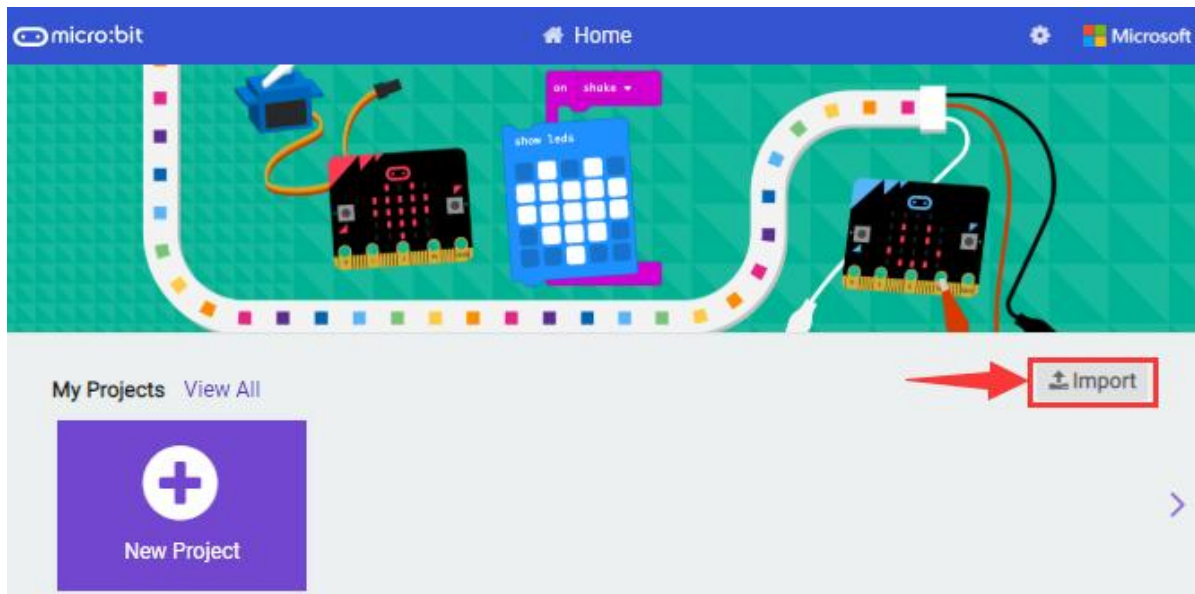


### 4.3 Import Code

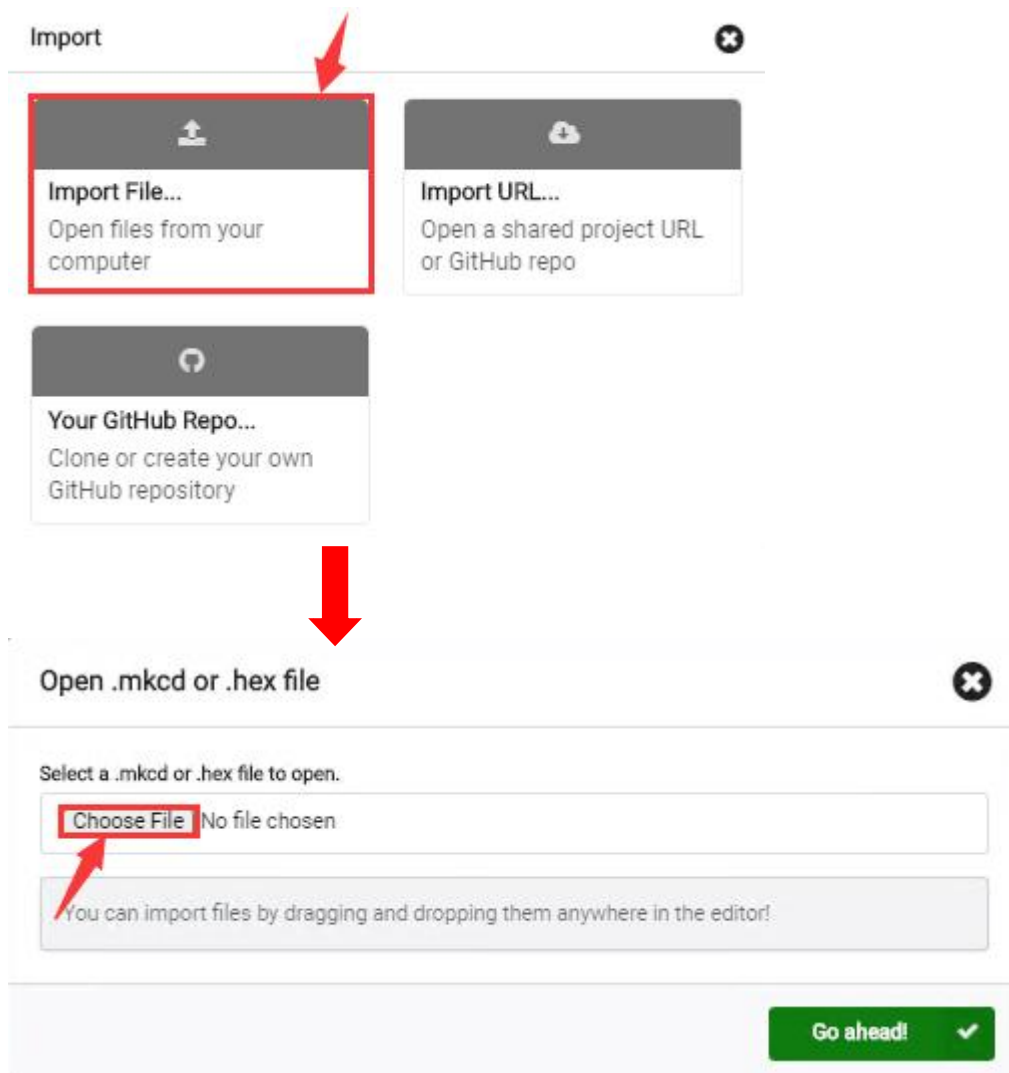
We provide every program with hex file. You could import it directly or program in Makecode blocks area, therefore, the extension library must be added.[\(How to add extension?\)](#)

Next, we will take “heartbeat” as example to introduce how to import code.

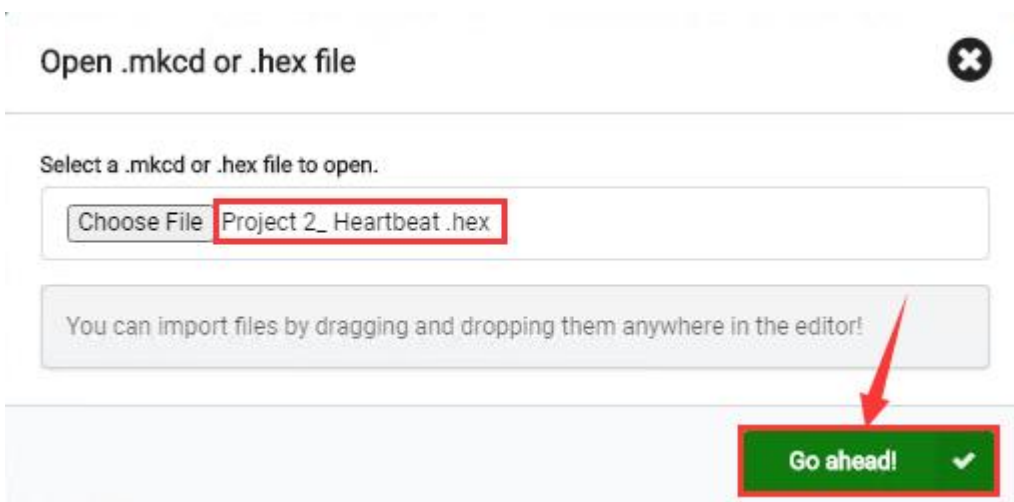
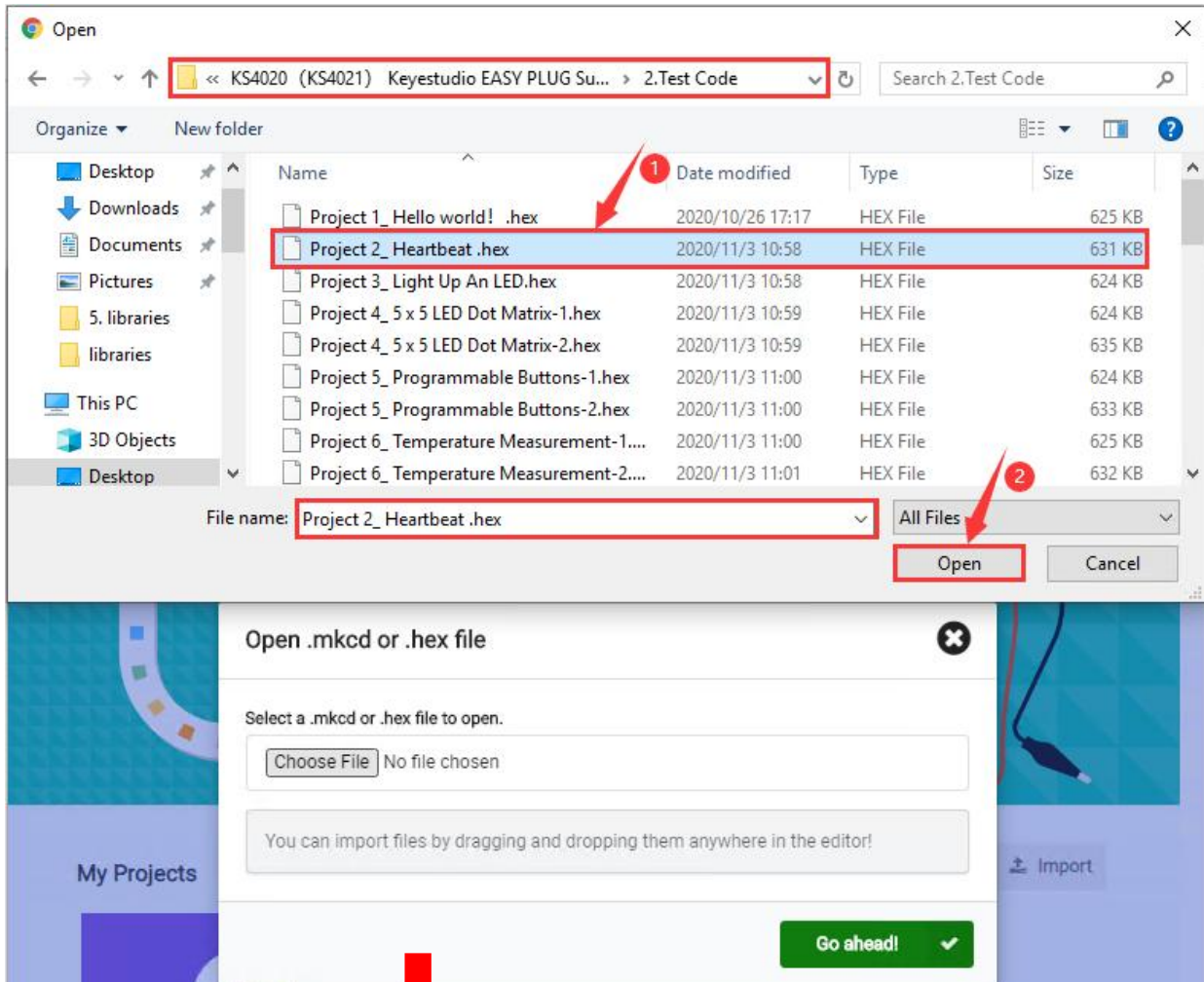
Open Makecode online editor on your computer.



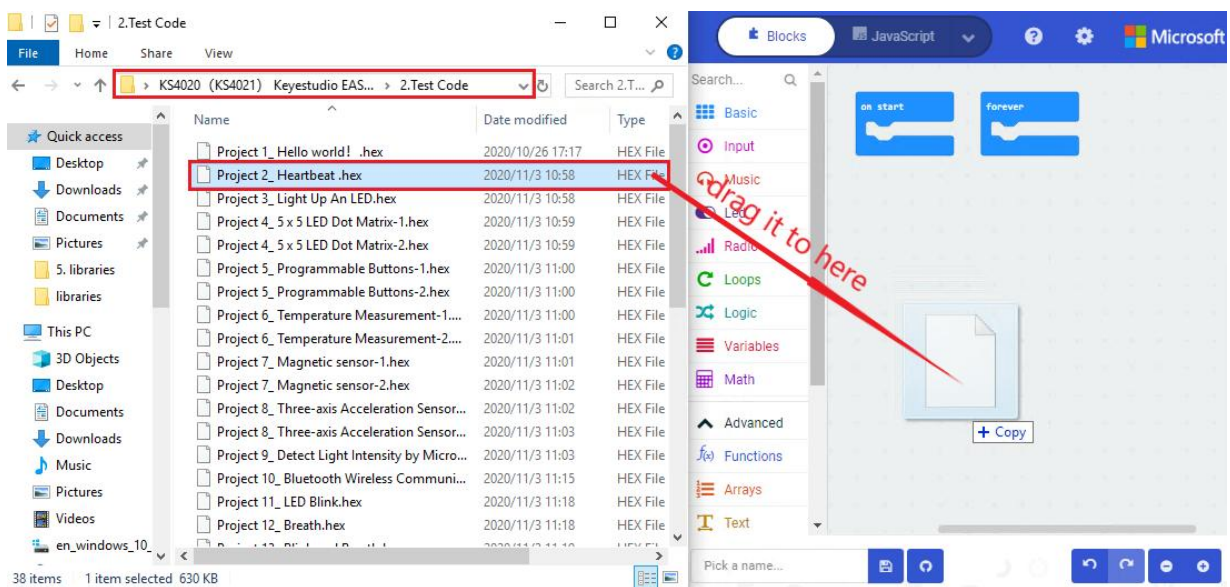
Click "Import" and "Import files"



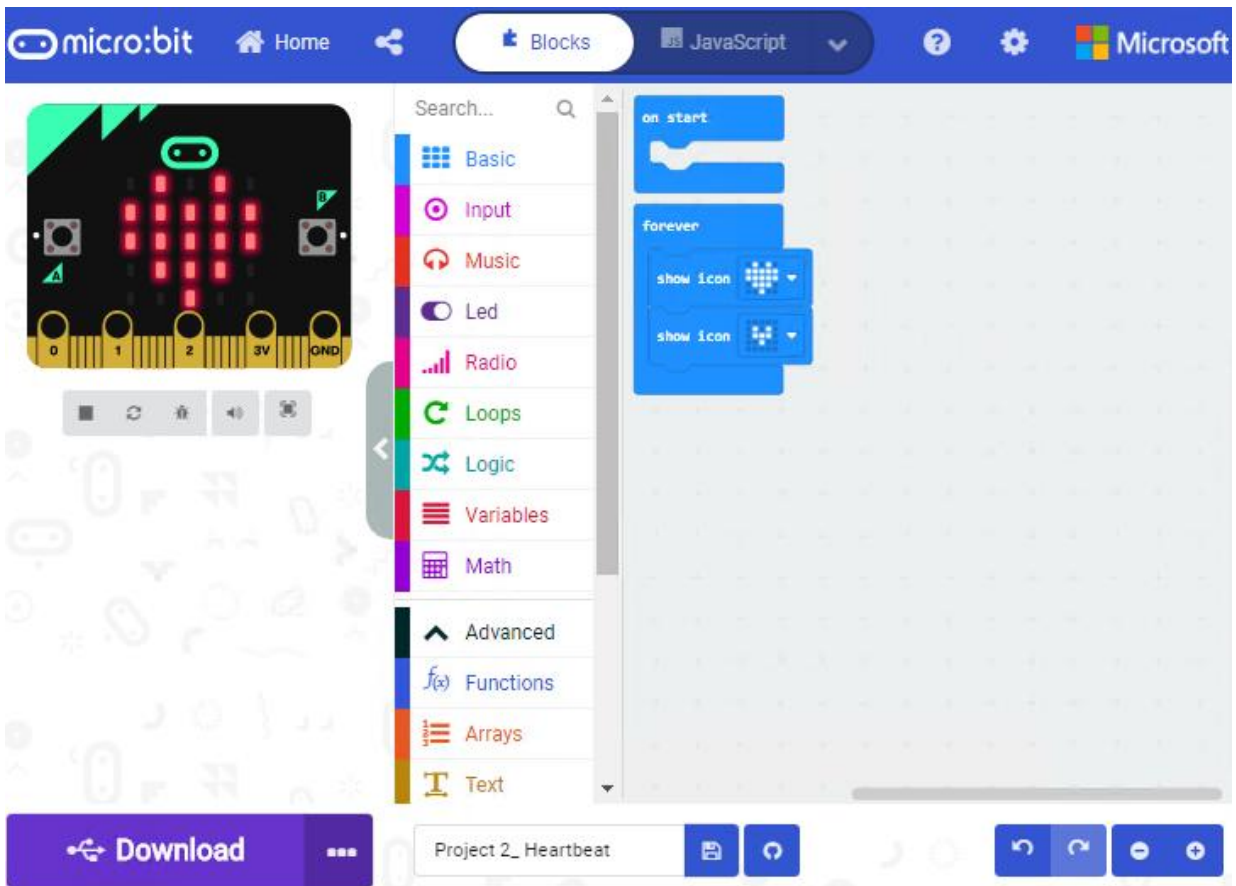
Choose file: `../2.Test Code/Project 2_ Heartbeat.hex` , then tap "Go ahead"



In addition to the above method of importing the provided project code program files directly into the Makecode compiler, you can also drag the provided project code program files directly into the Makecode compiler, as shown in the figure below:



The program is imported successfully after seconds



#### 4.4 CoolTerm Installation

CoolTerm program is used to read the serial communication.

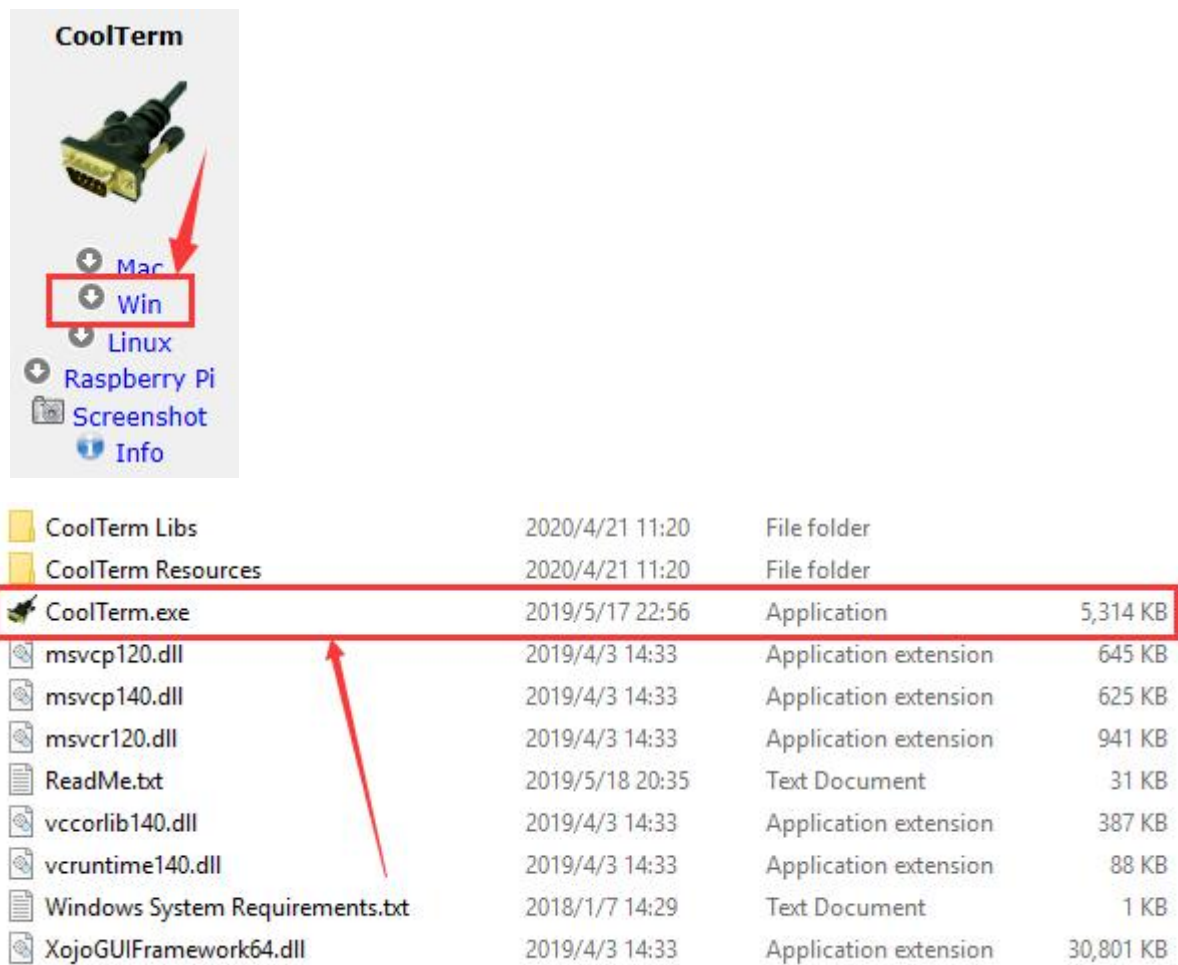
Download CoolTerm program:

<https://freeware.the-meiers.org/>

(1) After the download, we need to install **CoolTerm program file**, the Window system is taken as an example.

(2) Choose "win"

(3) Unzip file and open it. **(also suitable for Mac and Linux system)**



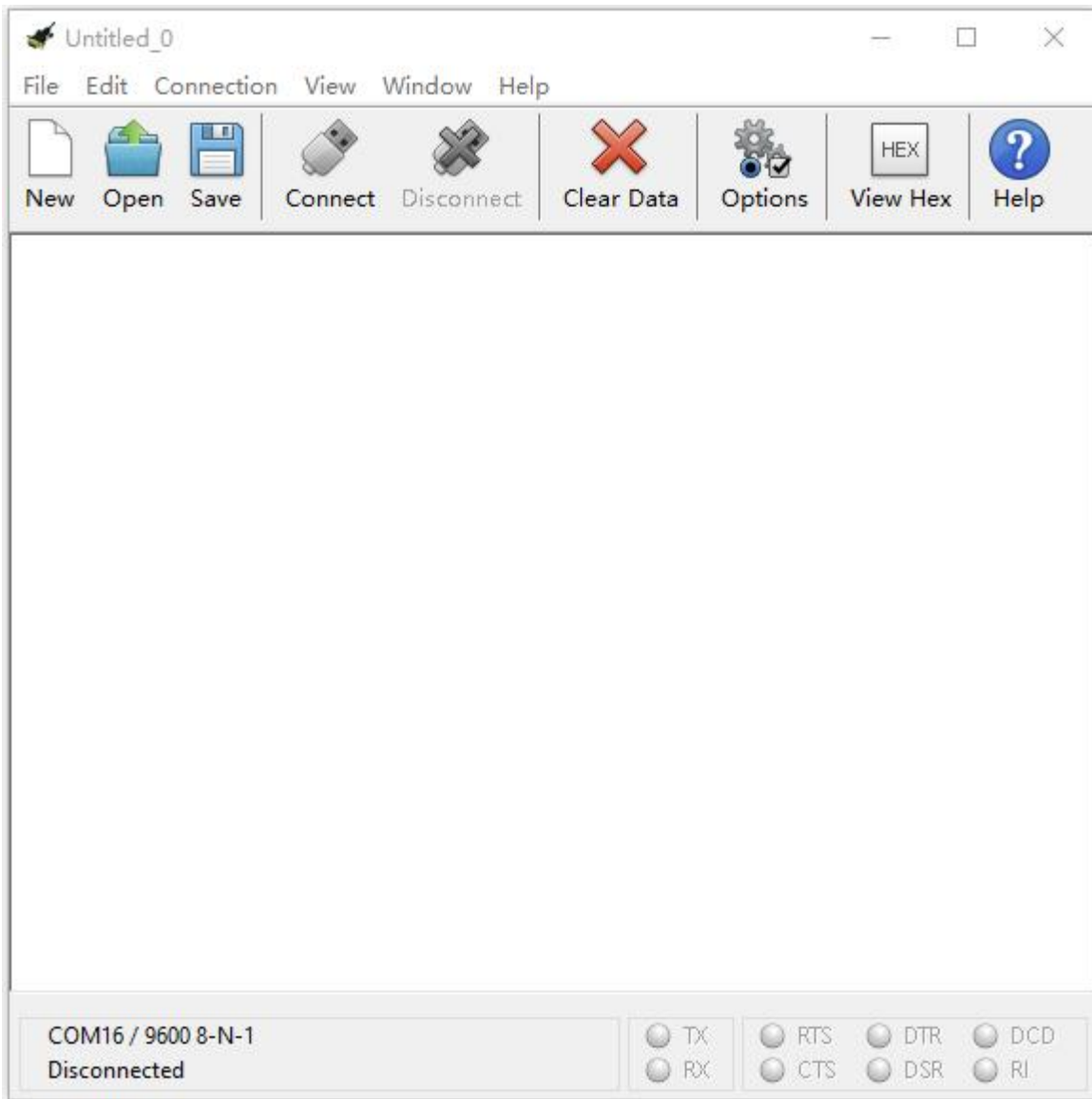
**CoolTerm**

- Mac
- Win**
- Linux
- Raspberry Pi
- Screenshot
- Info

CoolTerm Libs	2020/4/21 11:20	File folder	
CoolTerm Resources	2020/4/21 11:20	File folder	
<b>CoolTerm.exe</b>	2019/5/17 22:56	Application	5,314 KB
msvcp120.dll	2019/4/3 14:33	Application extension	645 KB
msvcp140.dll	2019/4/3 14:33	Application extension	625 KB
msvcr120.dll	2019/4/3 14:33	Application extension	941 KB
ReadMe.txt	2019/5/18 20:35	Text Document	31 KB
vccorlib140.dll	2019/4/3 14:33	Application extension	387 KB
vcruntime140.dll	2019/4/3 14:33	Application extension	88 KB
Windows System Requirements.txt	2018/1/7 14:29	Text Document	1 KB
XojoGUIFramework64.dll	2019/4/3 14:33	Application extension	30,801 KB




(4) Double-click  CoolTerm.exe







Note: you have to install the driver of micro:bit and connect micro:bit to computer



The functions of each button on the Toolbar are listed below:



 <p>New</p>	<p>Opens up a new Terminal</p>
 <p>Open</p>	<p>Opens a saved Connection</p>
 <p>Save</p>	<p>Saves the current Connection to disk</p>

 Connect	Opens the Serial Connection
 Disconnect	Closes the Serial Connection
 Clear Data	Clears the Received Data
 Options	Opens the Connection Options Dialog
 View Hex	Displays the Terminal Data in Hexadecimal Format
 Help	Displays the Help Window

## 5. Projects

### Project 1: Hello world

#### 1. Description

In first project, we will demonstrate the easiest experiment to make you acquire the basic knowledge of micro:bit.

We make micro:bit display "Hello word!" . In addition, we need a serial communication software CoolTerm as well. It sounds simple, right? Let' s get started!

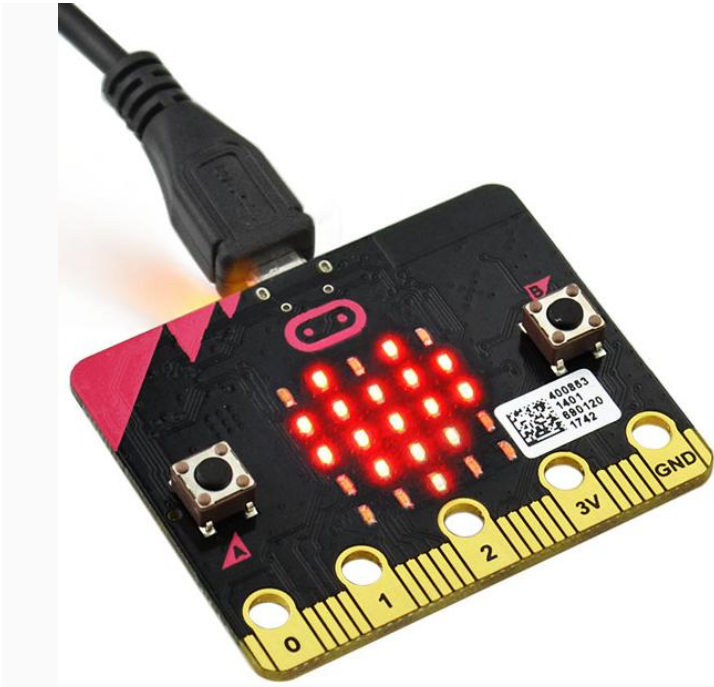
#### 2. What You Need:

Micro:bit Board\*1

Micro USB Cable\*1

#### 3.Wiring Up:

Connect micro:bit to your computer using micro USB cable.



#### 4. Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.

```

on start
  led enable false

serial on data received "R"
  serial write line "Hello,world!"
  pause (ms) 500
  
```

"on start" : command block only runs once to start program.

Turn off dot matrix on micro:bit

Serial transmits data when receiving "R"

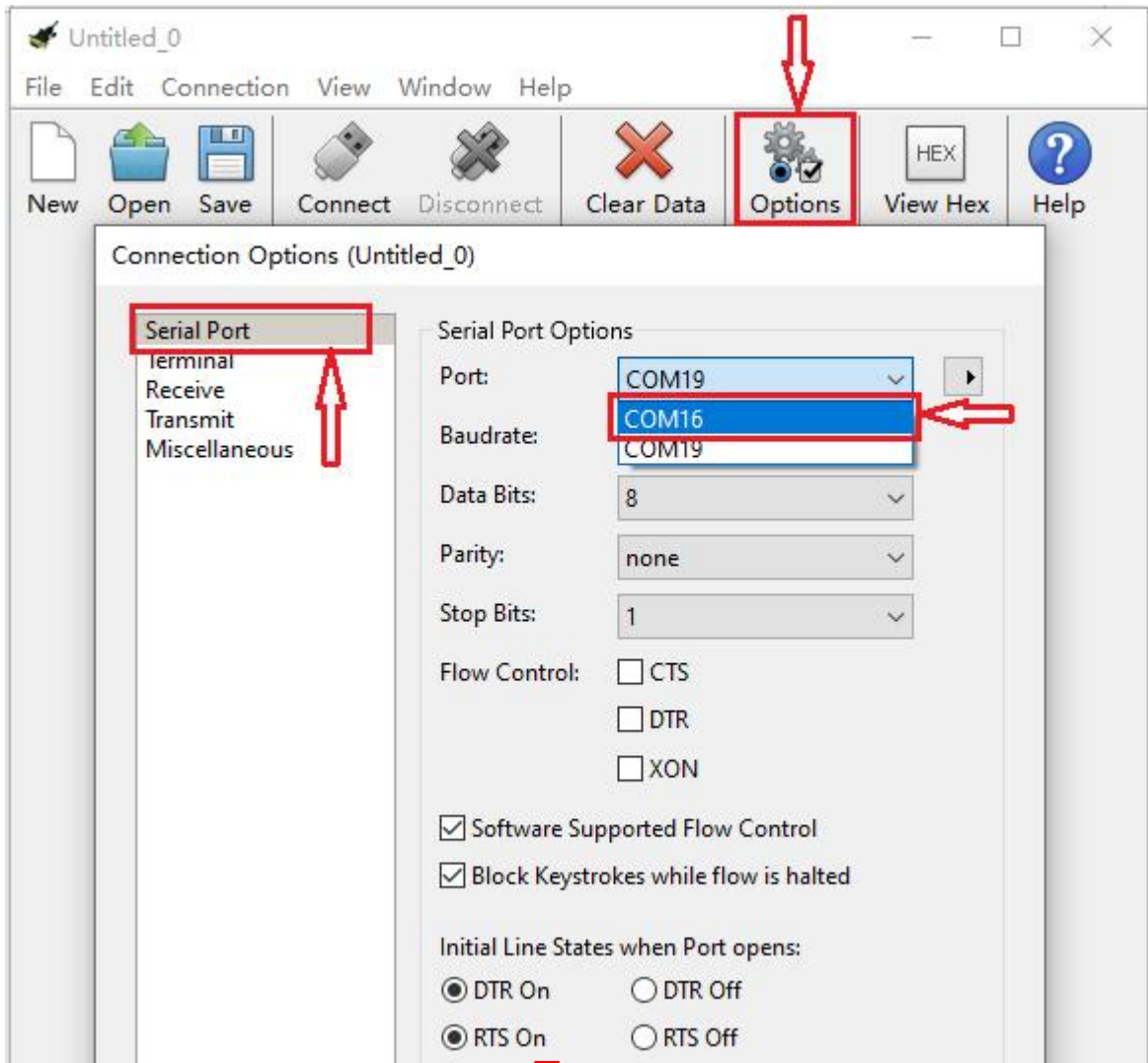
Serial writes "Hello World! "

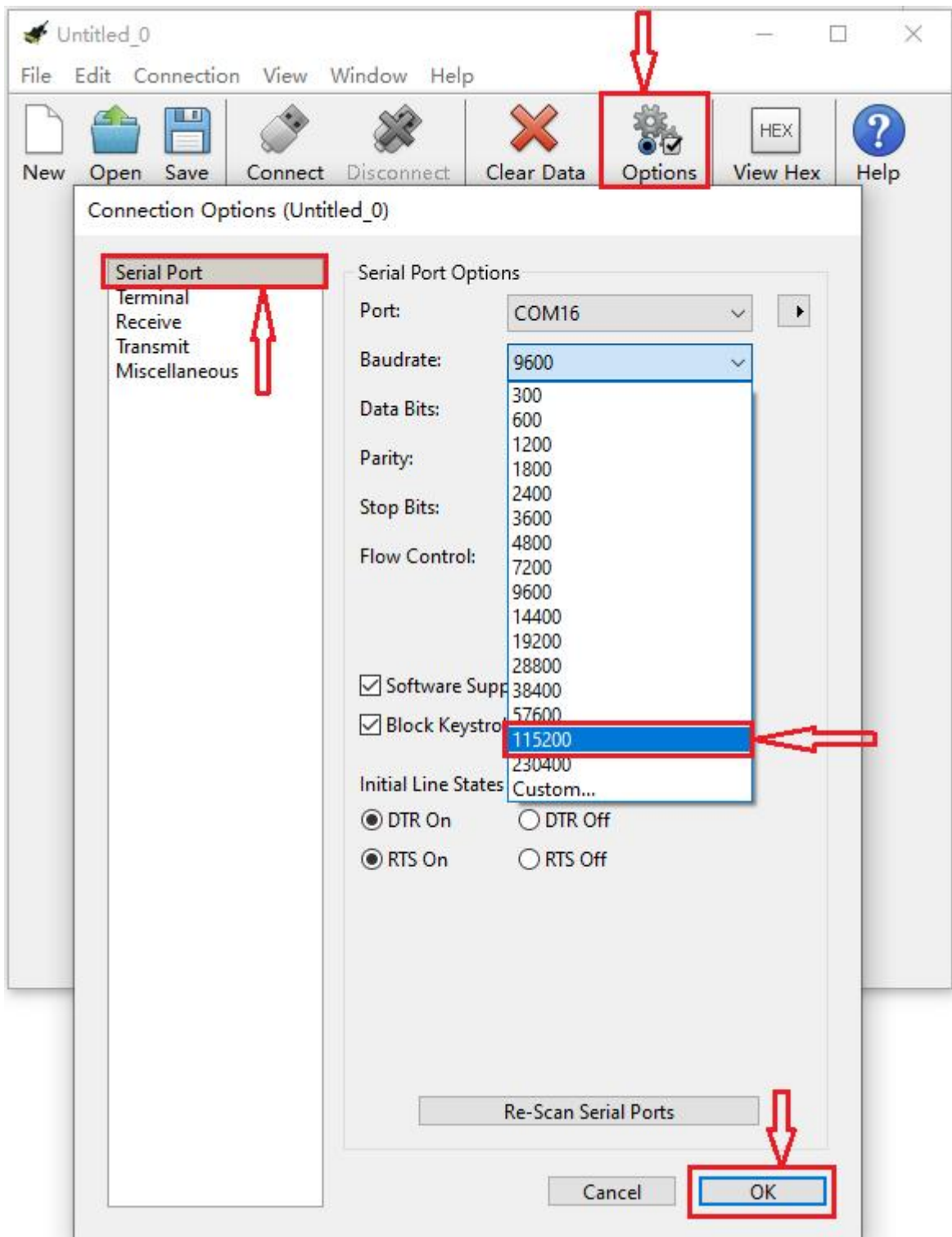
Delay in 500ms

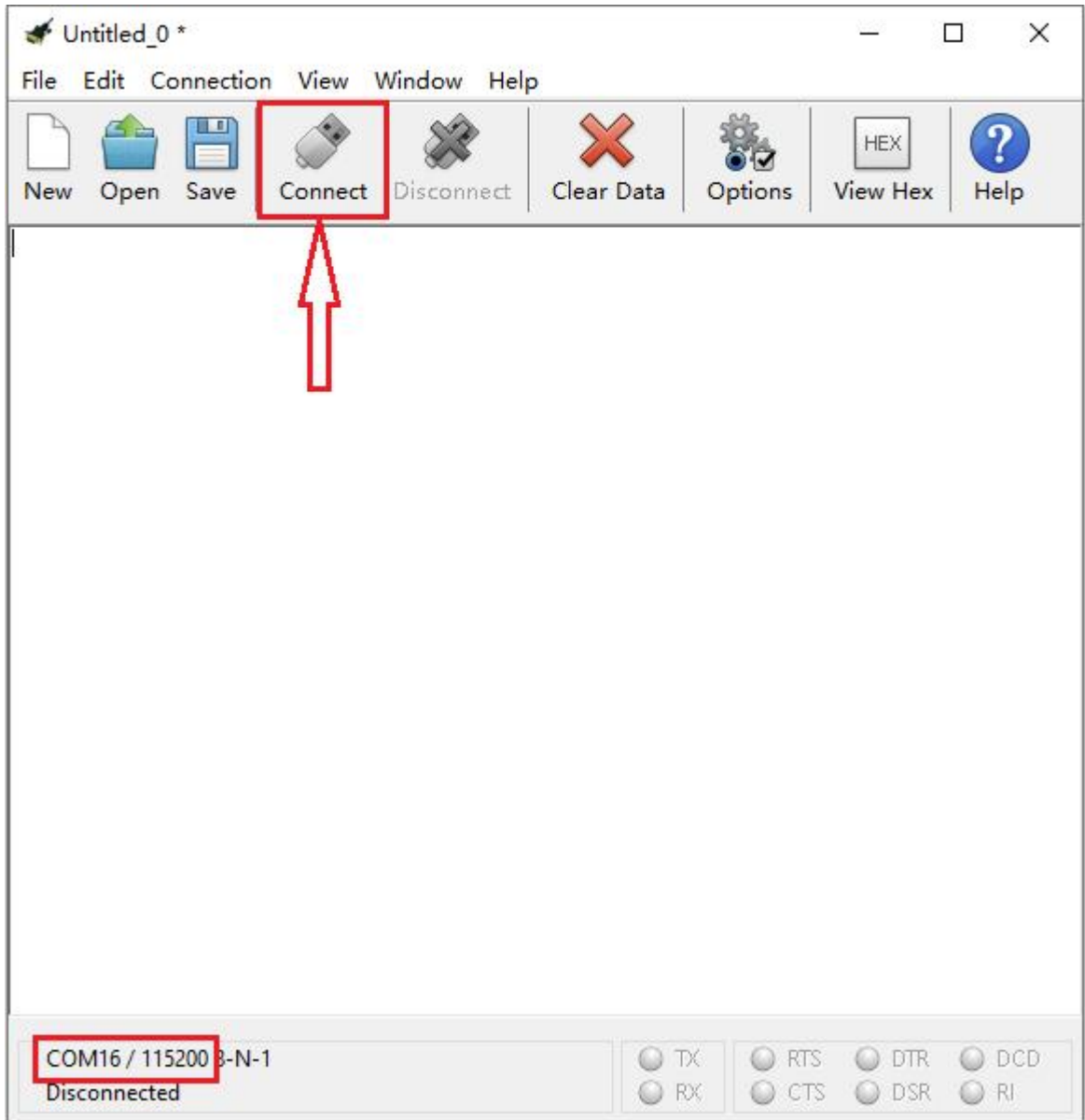
## 5.Test Result:

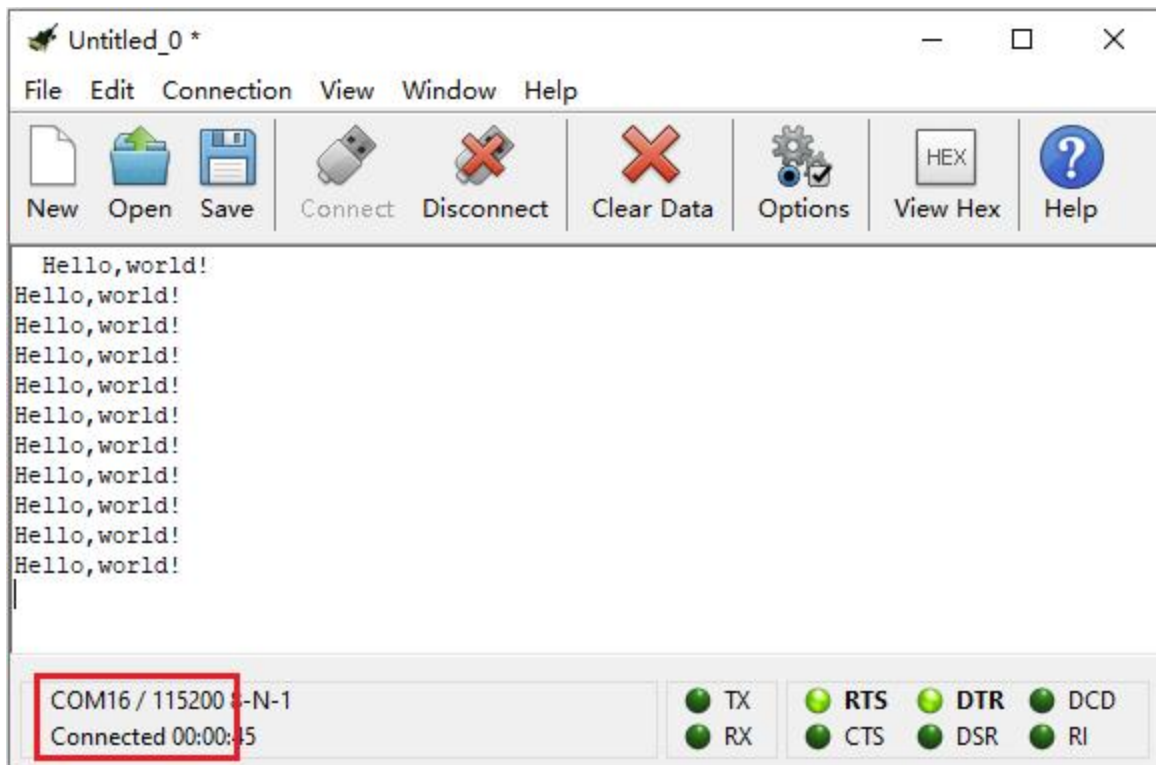
According to wiring diagram, upload code and plug micro:bit to power. Open CoolTerm to select "SerialPort" , set COM port and baud rate to 115200 (the baud rate of USB serial communication of micro:bit is 115200 through the test). Click "OK" and "Connect" . Press "Capslock" and "R" on keyboard, "Hello World! " will be shown on CoolTerm monitor, as shown below:

Please disconnect USB cable and reconnect it if there is something wrong with downloading code.

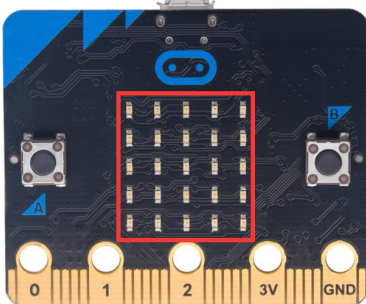








## Project 2: Heart beat



### 1. Description

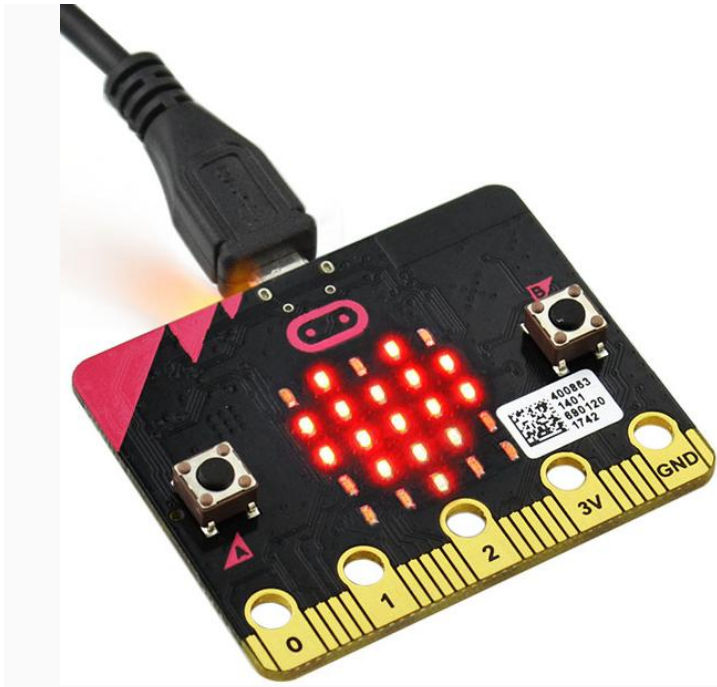
Next, we will make micro:bit display "Heart Beat" pattern.

### 2.What You Need:

- Micro:bit Board\*1
- Micro USB Cable\*1

### 3. Wiring Up:


Connect micro:bit to your computer using micro USB cable



### 4. Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.



"on start" : command block only runs once to start program.

The program under the block "forever" runs cyclically.

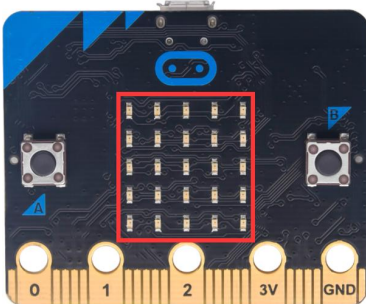
LED screen displays "♥"

Micro:bit shows "📊"

## 5. Test Result:

Download code to micro:bit and keep USB connected. Plug in power, micro:bit shows "♥" and "📊" ceaselessly.

## Project 3: Light Up a Single LED

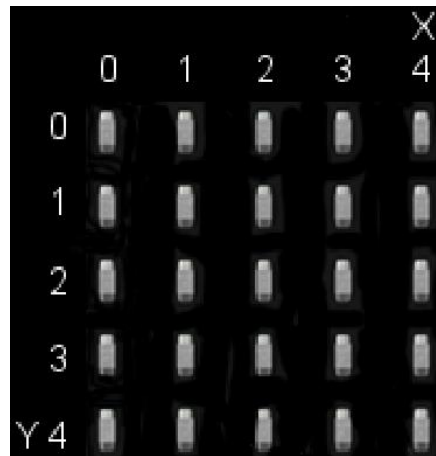


### 1. Description

Micro:bit motherboard consists of 25 light-emitting diodes, 5 pcs in a group. They correspond to x and y axis. Then the 5\*5 matrix is formed. Moreover, every diode locates at the point of x and y axis.

Virtually, we could control a LED by setting coordinate points. For instance, set coordinate point (0, 0) to turn on the LED at row 1 and

column 1; light up LED at the row 1 and column 3, we could set (2, 0) and so on.

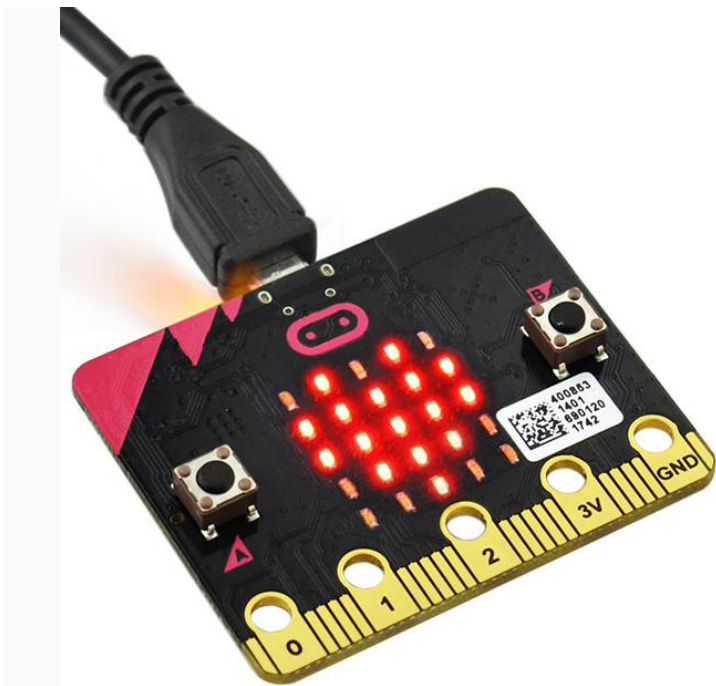


## 2.What You Need:

- Micro:bit Board\*1
- Micro USB Cable\*1

## 3.Wiring Up:

Connect micro:bit to your computer using micro USB cable



#### 4.Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.



“on start” : command block only runs once to start program.

open LED dot matrix on micro:bit

The program under the block “forever” runs cyclically.

Toggle the LED brightness at coordinate point “x 1 y 0” .

Delay in 500ms

Toggle the LED brightness at coordinate point “x 1 y 0” .

Delay in 500ms

Turn on the LED at coordinate point “x3, y4” .

Delay in 500ms

Turn off the LED at coordinate point “x3 y4” .

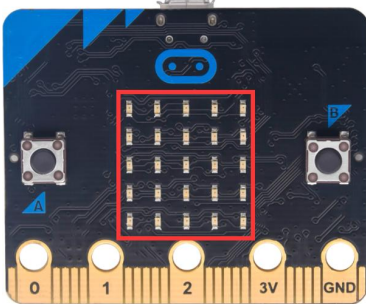
Delay in 500ms

## 5.Test Result:

Download code to micro:bit and keep USB connected. We will view

the LED at coordinate point (1,0) flashes for 0.5s, and the LED at (3,4) flashes for 0.5s alternately.

## Project 4: 5×5 LED Dot Matrix



### 1. Description

Dot matrix gains popularity in our life, such as LED screen, bus station and the mini TV in the lift.

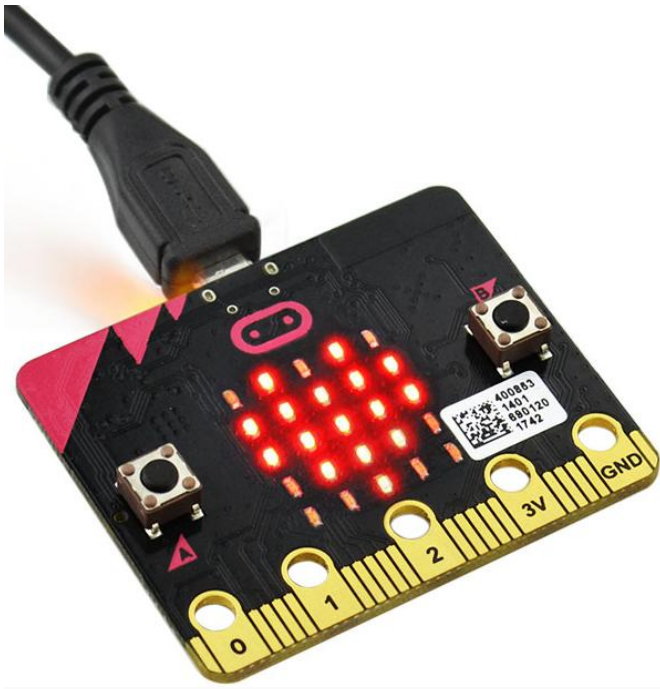
The dot matrix of Micro:bit board consists of 25 light emitting diodes. In previous lesson, we control LED of Micro:bit board to form patterns, numbers and character strings by setting the coordinate points. Moreover, we could adopt another way to complete the display of patterns, numbers and character strings.

### 2. What You Need:

- Micro:bit Board\*1
- Micro USB Cable\*1

### 3. Wiring Up:

Connect micro:bit to your computer using micro USB cable.

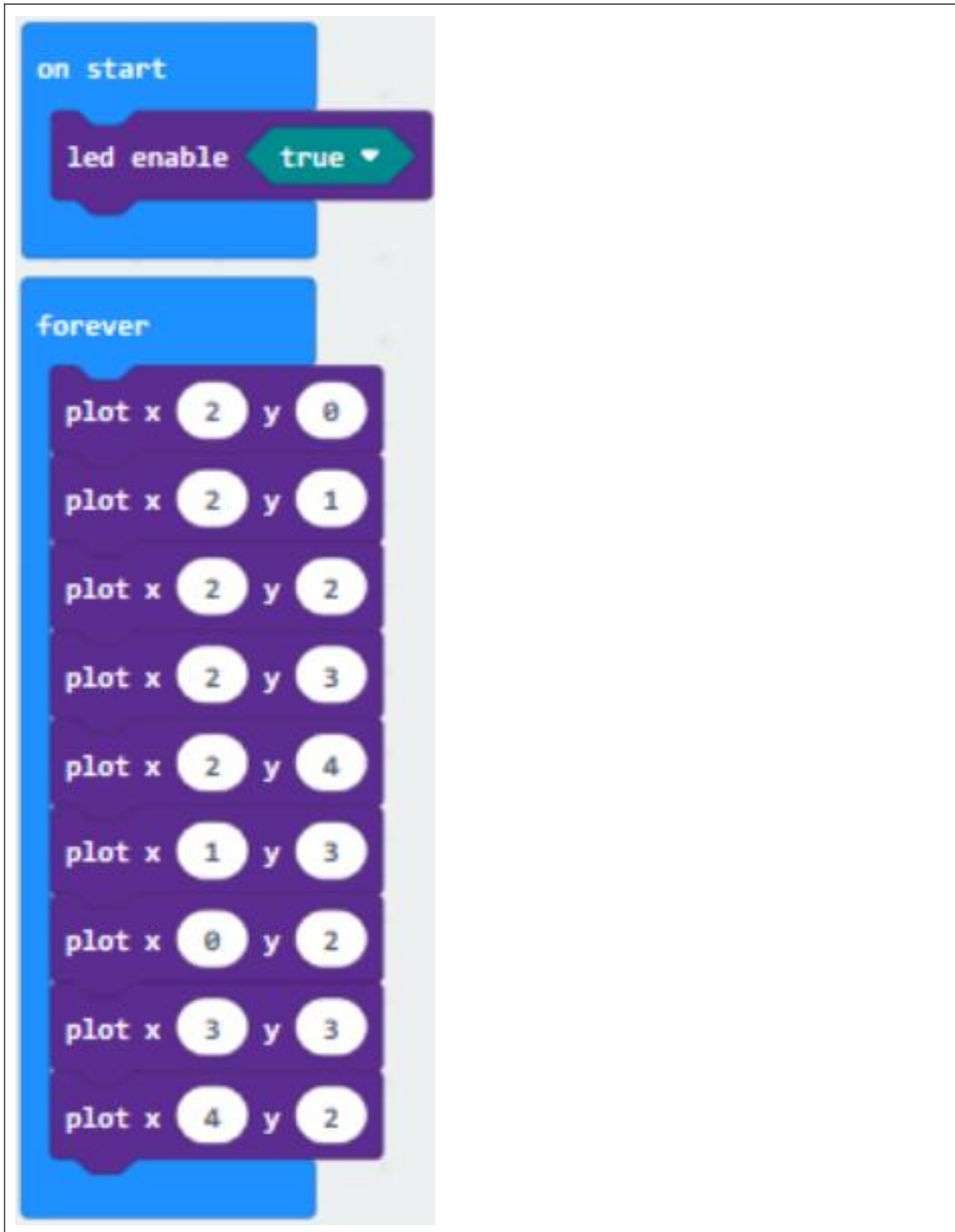


#### 4. Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.

#### Code 1:



"on start" : command block only runs once to start program.

open LED dot matrix on micro:bit

The program under the block "forever" runs cyclically.

.Toggle the LED brightness at coordinate point "x 2, y 0" , "x 2, y 1" , "x 2, y 2" , "x 2, y 3" , "x 2, y 4" , "x 1, y 3" , "x 0, y 2" , "x 3, y 3" and "x 4, y 2"

## Code 2:

The image shows a Scratch script for a simulation. The script starts with an 'on start' block, followed by five 'show number' blocks with values 1, 2, 3, 4, and 5. Next is a 'show leds' block with a 5x5 grid of LEDs, where the center and four surrounding LEDs are lit. This is followed by a 'show string' block with the text 'Hello!'. Then, a 'show icon' block with a 5x5 grid icon. Next are four 'show arrow' blocks with directions North East, South East, South West, and North West. This is followed by a 'clear screen' block and a 'pause (ms)' block with a value of 500.

```
on start
  show number 1
  show number 2
  show number 3
  show number 4
  show number 5
  show leds
  show string "Hello!"
  show icon
  show arrow North East
  show arrow South East
  show arrow South West
  show arrow North West
  clear screen
  pause (ms) 500
```

"on start" : command block only runs once to start program.

LED dot matrix displays 1,2,3,4,5

Dot matrix shows the "↓" pattern

Under the block "forever" , program runs cyclically.

Dot matrix shows the "↓" pattern

Dot matrix scrolls to show "Hello!"

"♥" is shown on dot matrix

LED dot matrix displays "North East" arrow.

The "South East" arrow shows up on LED dot matrix

The "South West" arrow appears up on LED dot matrix

The "North West" arrow is displayed on LED dot matrix

Clear the screen


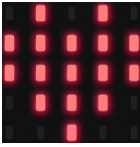
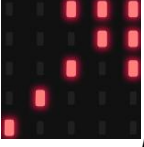
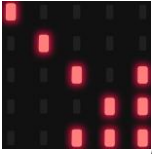

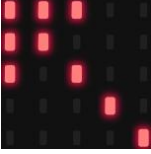
Delay in 500ms

## 5.Test Result:

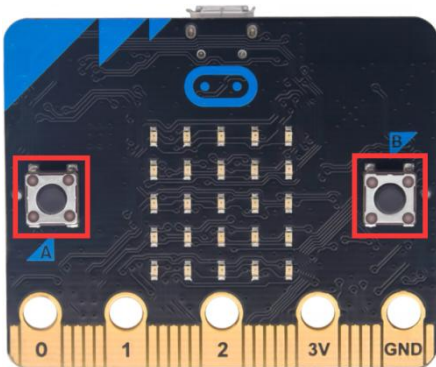
Upload code 1 and power on , keep USB cable connected, we will see

the  icon.

Transfer code 2 to micro:bit. Micro: bit starts showing number 1, 2, 3,

4, and 5, then cyclically display , "Hello!" , , , ,  and  patterns.

## Project 5: Micro:bit' s Buttons



### 1. Description

The buttons are common input devices. You will have used a power button to switch off the mobile or tablet, or volume buttons to turn the audio level up or down. The buttons on the BBC micro:bit are input devices that can be pressed to trigger an action. You can write a program to tell the micro:bit what actions should happen when the

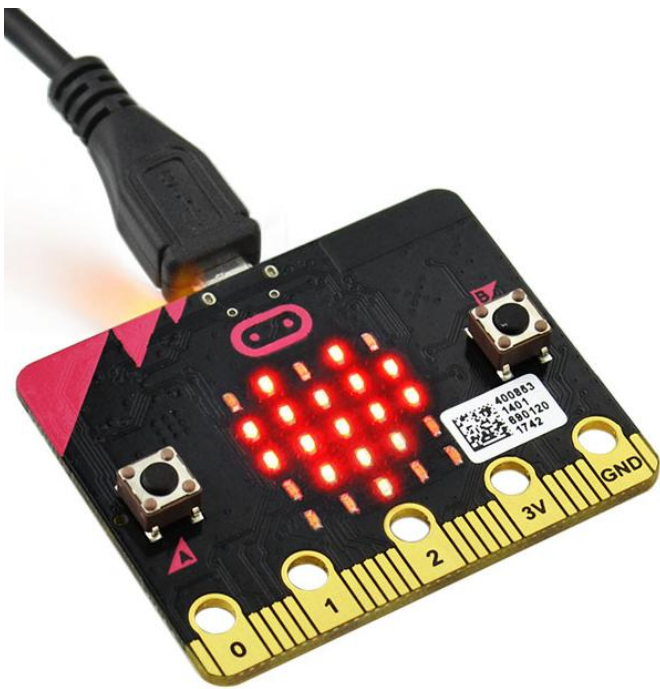
button is pressed, and there are lots of you can program. There are two buttons on the micro:bit, button A and button B. The button can be pressed separately or together. There is a third button on the other side, that's for resetting your micro:bit and start your program from the beginning.

## 2. What You Need:

- Micro:bit Board\*1
- Micro USB Cable\*1

## 3.Wiring Up:

Connect micro:bit to your computer using micro USB cable.




## 4.Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have

access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.

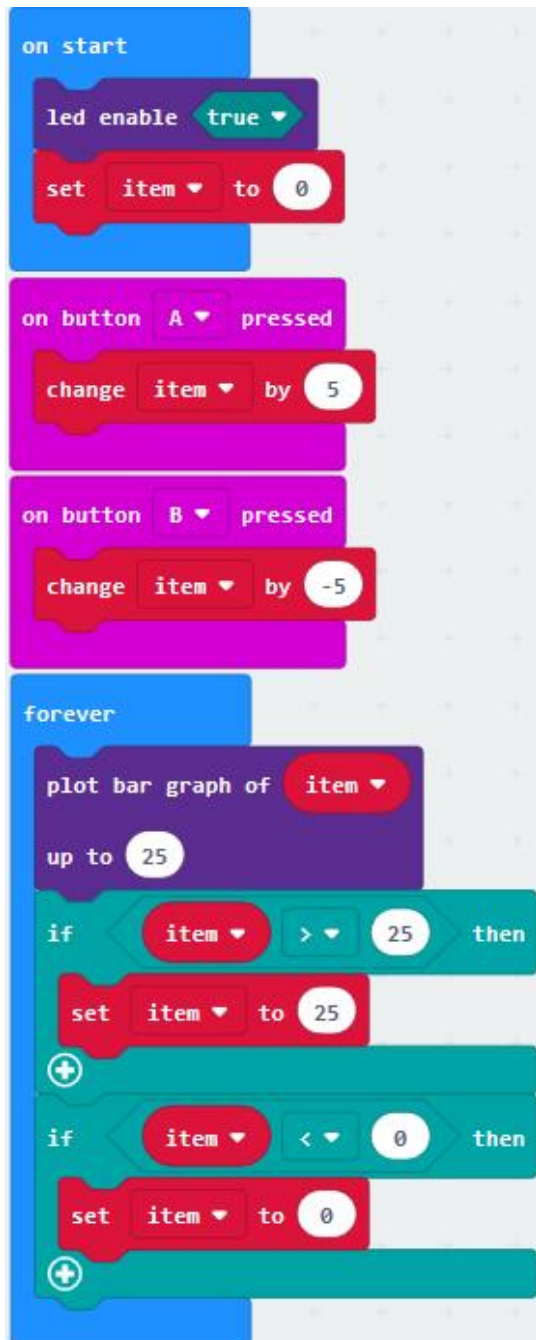
### Code 1:



The image shows three Scratch code blocks. Each block starts with an 'on button pressed' event. The first block is for button 'A' and contains a 'show string' block with the value 'A'. The second block is for button 'B' and contains a 'show string' block with the value 'B'. The third block is for button 'A+B' and contains a 'show string' block with the value 'AB'.

on button A pressed	Press button A on micro: bit main board Show the character "A"
on button B pressed	Press button B on micro: bit main board Show the character "B"
on button A+B pressed	Press button A and B at same time Display the character "AB"

### Code 2



“on start”: command block runs once

to start program.

Turn on LED dot matrix

Set the initial value of item to 0

Press button A on Micro:bit board

Change item by 5

Press button B on Micro:bit board

Change item by -5

The program under the block

“forever” runs cyclically. Light on

LED in dot matrix to draw bar

graph, light up up to 25 LEDs

If item is greater than 25

Then set item to 25

If item is less than 0

Then Set item to 0

## 5. Test Result:

Download code 1 to micro:bit and keep USB connected. Press “A” on micro:bit board, character “A” will be displayed; in case that B is pressed, letter “B” will appear. So will “AB” if you press A and B

buttons simultaneously.

Then download code 2 in same way. Press button A, a row of LEDs are added, when B is pressed, a row of LEDs are deducted.

## Project 6: Temperature Measurement

### 1. Description

Micro:bit main board contains temperature sensor, as micro:bit is small pocket-sized computer, it has a micro processor. That means this is a tiny electronic component with less power than normal computer processor, which is one of the reason your micro: bit is so small and can run on batteries.

The micro:bit processor runs the program you create for your micro:bit so when you write a program in online editors, then transfer it to your micro:bit. The processor is where your program runs.

**Note: the temperature sensor is included in the processor**



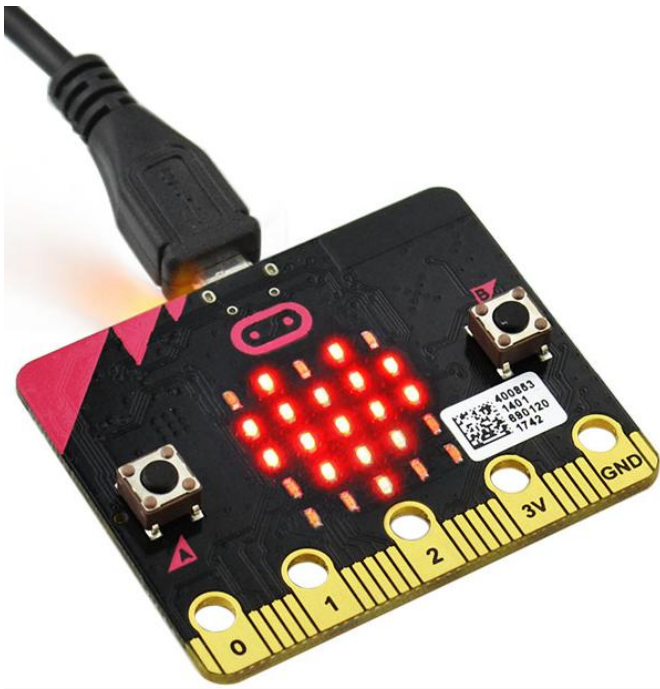
## 2.What You Need:

Micro:bit Board\*1

Micro USB Cable\*1

## 3.Wiring Up:

Connect micro:bit to your computer using micro USB cable.




## 4.Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.

### Code 1:



The image shows a Scratch script with the following blocks:

- on start** block containing a **serial redirect to USB** block.
- forever** loop block containing:
  - serial write value** block with the text "Temperature" and a variable input for **temperature (°C)**.
  - pause (ms)** block with the value 500.

“on start” : command block only runs once to start program.

Serial redirect to USB

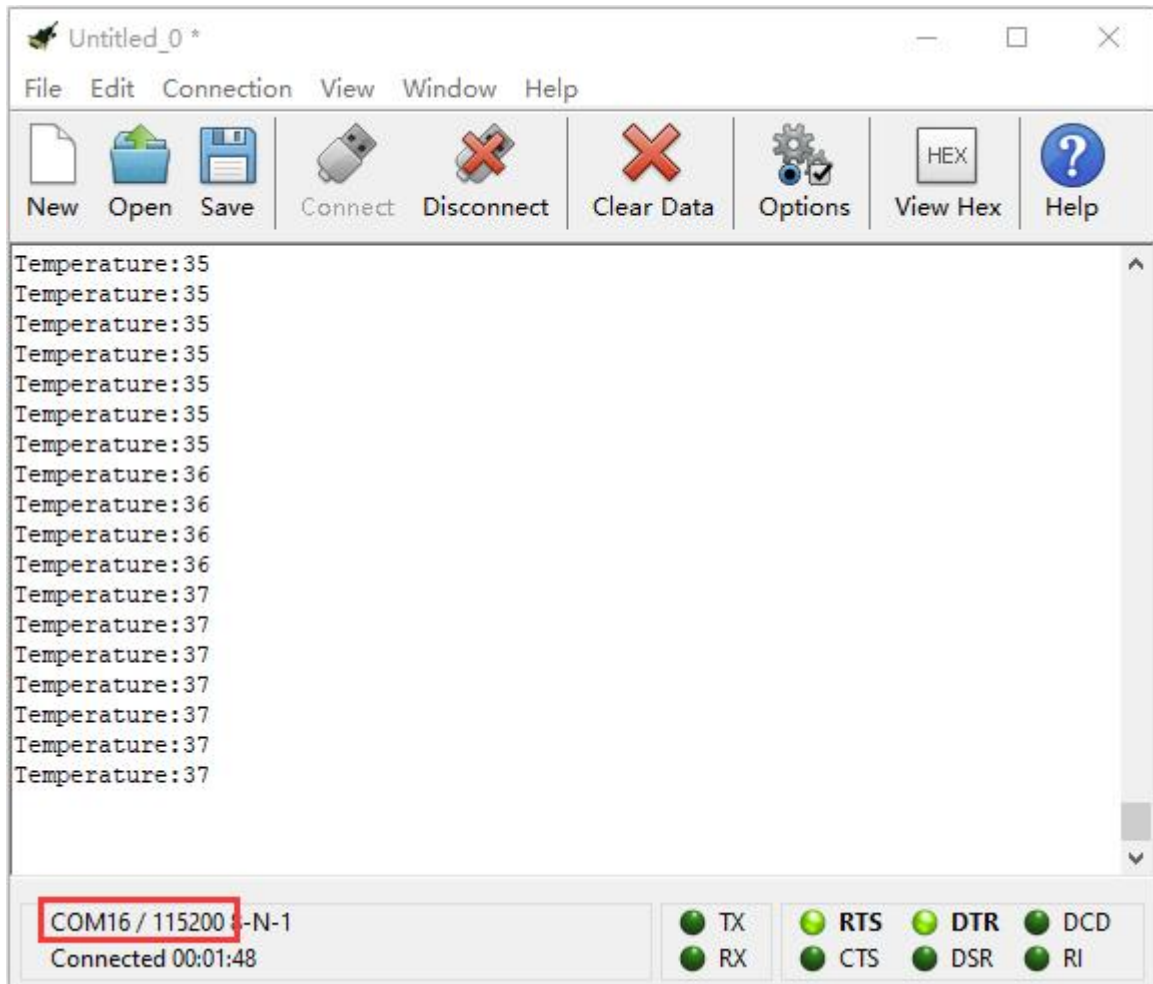
The program under the block “forever” runs cyclically.

Serial writes Temperature=Temperature (°C)

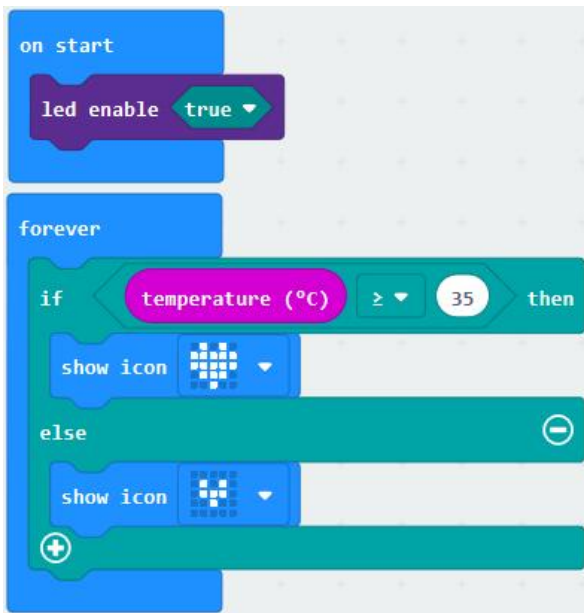
Delay in 500ms

According to wiring diagram, upload code and plug micro:bit to power. Open CoolTerm to select “SerialPort” , set COM port and baud rate to 115200 (the baud rate of USB serial communication of micro:bit is 115200 through the test).

Click “OK” and “Connect” .



**Code 2 (Note: temperature value can be adjusted)**



"on start" : command block runs once to start program.

Turn on LED dot matrix

Under the block "forever" , program runs cyclically.

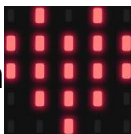
If the detected temperature  $\geq 35^\circ$  , the next program will be executed.

Dot matrix shows "♥"

## 5. Test Result:

Then transfer code 2 in micro:bit, when the ambient temperature is

less than  $35^\circ\text{C}$ , micro:bit will show  . When the temperature is

equivalent to or greater than  $35^\circ\text{C}$ , the pattern  will appear.

## Project 7: Micro:bit Compass



### 1. Description

This project mainly introduces the use of the micro:bit magnetic sensor. In addition to detecting the strength of the magnetic field, it can also be used as an electronic compass to determine the direction, an important part of the heading and attitude reference system (AHRS) as well.

It uses FreescaleMAG3110 three-axis magnetometer. Its I2C interface communicates with the outside, the range is  $\pm 1000\mu\text{T}$ , the maximum data update rate is 80Hz. Combined with accelerometer, it can calculate the position. Additionally, it is applied to magnetic detection and compass blocks.

Then we could read the value detected by magnetic sensor to determine the location. We need to calibrate the micro:bit board when magnetic sensor works.

The correct calibration method is to rotate the micro:bit board.

In addition, the objects nearby may affect the accuracy of readings and calibration.

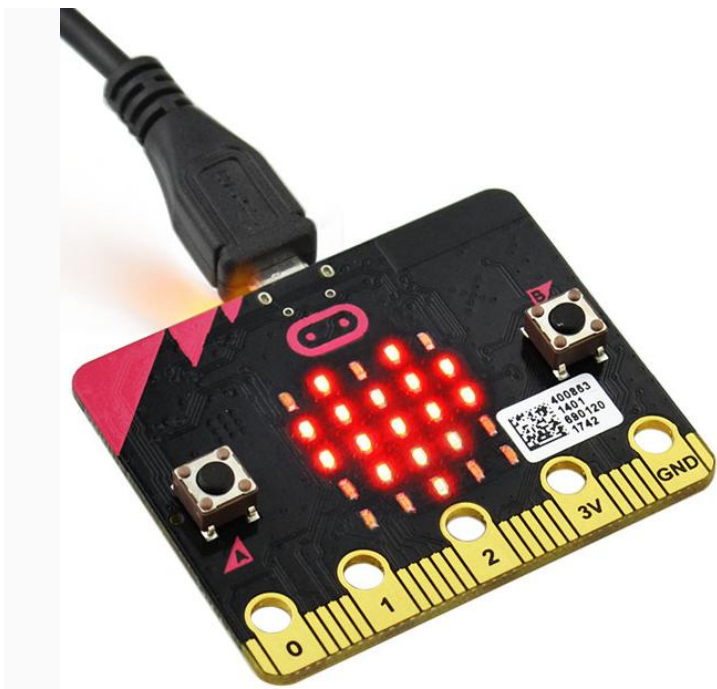
## 2.What You Need:

Micro:bit Board\*1

Micro USB Cable\*1

## 3.Wiring Up:

Connect micro:bit to your computer using micro USB cable.

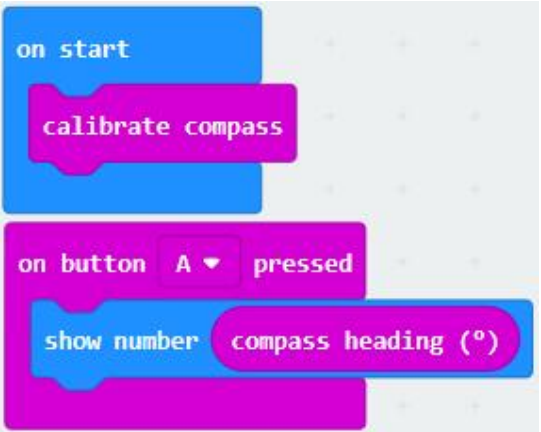


## 4.Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.

## Code 1:



The image shows a Scratch-style code editor with two event blocks. The first block is 'on start' (blue) containing a 'calibrate compass' block (purple). The second block is 'on button A pressed' (purple) containing a 'show number' block (blue) with the value 'compass heading (°)' (purple).

“on start” : command block only runs once to start program.

Calibrate compass

Press button A on Micro:bit main board

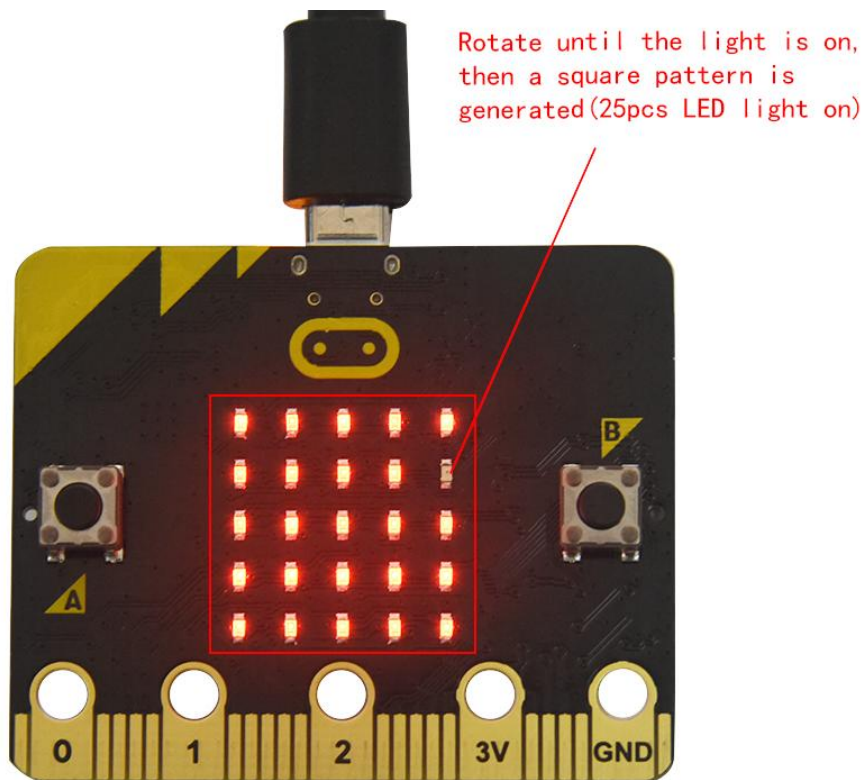
Dot matrix shows the direction of compass heading

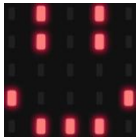
## Code Explanation:

Calibrate micro:bit first due to different magnetic field everywhere. Micro:bit will prompt the need of calibration if it is the first time to use it.

Download code 1 to micro:bit and keep USB cable connected. Press

button A, LED dot matrix will show "TILT TO FILL SCREEN" . Then enter the calibration interface, rotate the micro:bit board to make a full square (25 LEDs are on) on dot matrix, as shown below:



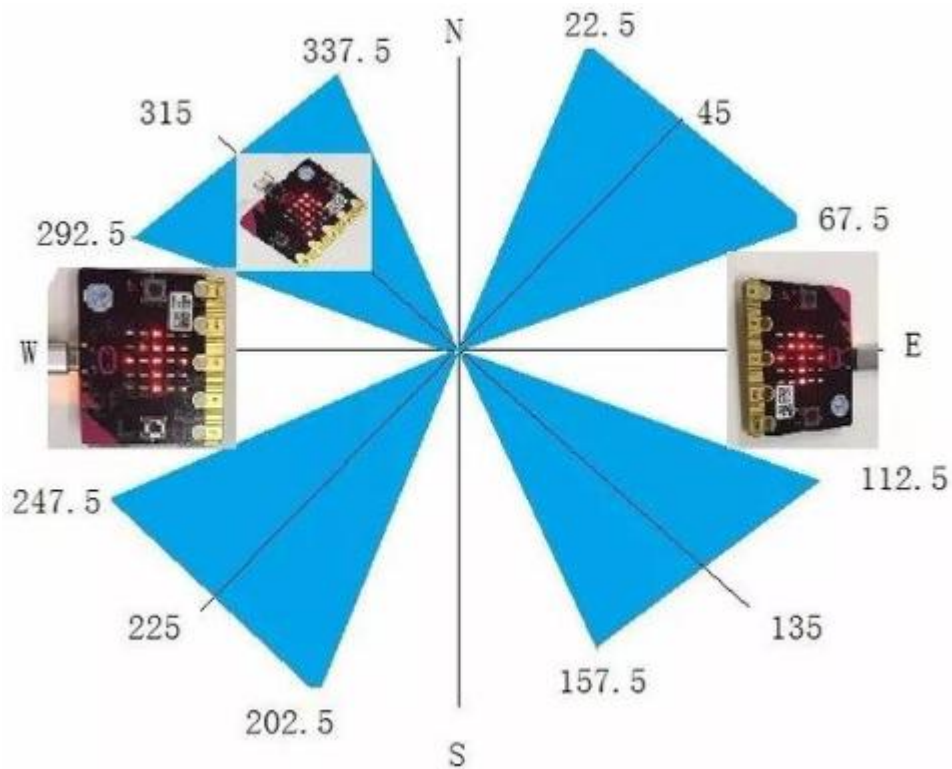
A smile  will appear if calibration is finished.

After calibrating, press A each time, the serial monitor will show  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  and  $270^\circ$  (correspond to North, East, South, West)

## Code 2



This code string complies that we read the value detected incessantly and determine the direction by the value range. The arrow is toward North.



For the above picture, the arrow pointing to the upper right when the value ranges from 292.5 to 337.5. 0.5 can't be input in the code, thereby, the values we get are 293 and 338

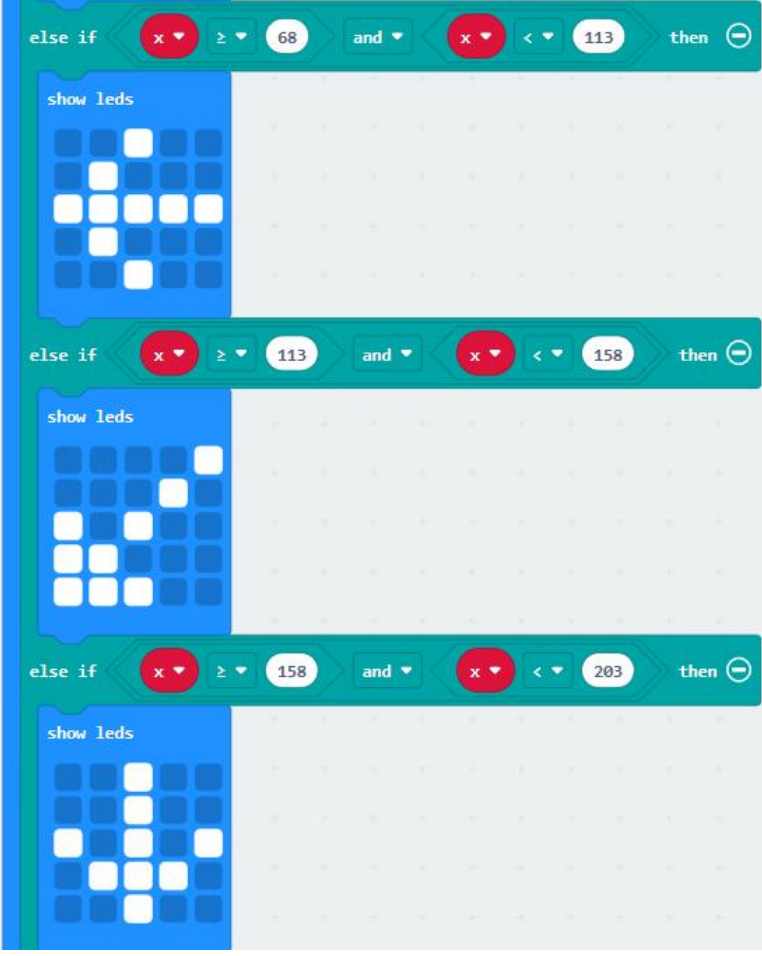
**Complete Code:**

The image shows a Scratch script for a compass calibration program. The script is as follows:

```
on start
  calibrate compass

forever
  set x to compass heading (°)
  if x >= 293 and x < 338 then
    show leds
  else if x >= 23 and x < 68 then
    show leds
```

The script starts with an "on start" block containing a "calibrate compass" block. This is followed by a "forever" loop. Inside the loop, the variable "x" is set to the "compass heading (°)". An "if" block checks if "x" is greater than or equal to 293 and less than 338. If true, it triggers a "show leds" block. An "else if" block checks if "x" is greater than or equal to 23 and less than 68. If true, it also triggers a "show leds" block.



The image shows three 'else if' code blocks in a Scratch script, each followed by a 'show leds' block. The 'show leds' blocks display a 5x5 grid of LEDs with varying patterns of lit (white) and unlit (blue) LEDs.

The first 'else if' block has the condition:  $x \geq 68$  and  $x < 113$ . The 'show leds' block shows a pattern where the top row has 4 lit LEDs, the second row has 3, the third row has 2, the fourth row has 1, and the fifth row has 0 lit LEDs.

The second 'else if' block has the condition:  $x \geq 113$  and  $x < 158$ . The 'show leds' block shows a pattern where the top row has 3 lit LEDs, the second row has 2, the third row has 1, the fourth row has 0, and the fifth row has 0 lit LEDs.

The third 'else if' block has the condition:  $x \geq 158$  and  $x < 203$ . The 'show leds' block shows a pattern where the top row has 2 lit LEDs, the second row has 1, the third row has 0, the fourth row has 0, and the fifth row has 0 lit LEDs.

The image shows a Scratch code editor with a script area containing three conditional blocks, each followed by a "show leds" block. The first block is an "else if" block with the condition  $x \geq 203$  and  $x < 248$ . The second block is an "else if" block with the condition  $x \geq 248$  and  $x < 293$ . The third block is an "else" block. Each "show leds" block displays a 5x5 grid of LEDs with a specific pattern of lit (white) and unlit (blue) LEDs. The first pattern has lit LEDs at (1,1), (1,2), (2,3), (3,4), (4,5), (4,4), (3,3), (2,2), and (1,1). The second pattern has lit LEDs at (1,3), (1,4), (2,4), (3,5), (4,4), (4,3), (3,2), (2,1), and (1,1). The third pattern has lit LEDs at (1,2), (1,3), (2,3), (3,4), (4,5), (4,4), (3,3), (2,2), and (1,1). A plus sign icon is visible at the bottom left of the script area.

"on start" : command block only runs once to start program.


Calibrate compass

The program under the block "forever" runs cyclically.


Store the angle of the compass heading into the variable x

Save the angle of


When  $293 \leq x < 338$ , execute the program under then block

Micro:bit shows 


When  $23 \leq x < 68$ , execute the program under then block

Micro:bit shows 


When  $68 \leq x < 113$ , execute the program under then block

Micro" bit shows "  "


When  $113 \leq x < 158$ , execute the program under then block

Micro" bit shows "  "


When  $158 \leq x < 203$ , execute the program under then block

Micro" bit shows "  "


When  $203 \leq x < 248$ , execute the program under then block

Micro:bit shows "  "

When  $248 \leq x < 293$ , execute the program under then block

Micro:bit shows "  "

When x is not among the above rang, the next program will be executed under else block

Micro:bit shows "  "

## 5. Test Result:

Download code 2 to micro:bit and keep USB connected.

After calibration, tilt micro:bit board, the LED dot matrix displays the direction signs.

## Project 8: Micro:bit Acceleration Sensor



### 1. Description

The micro:bit board has a built-in Freescale MMA8653FC three-axis acceleration sensor (accelerometer). Its I2C interface works on external communication, the range can be set to  $\pm 2g$ ,  $\pm 4g$ , and  $\pm 8g$ , and the maximum data update rate can reach 800Hz.

When the micro:bit is stationary or moving at a constant speed, the accelerometer only detects the gravitational acceleration; when the micro:bit is slightly shaken, the acceleration detected is much smaller than the gravitational acceleration and can be ignored. Therefore, in the process of using micro:bit, the main purpose is to detect the

changes of the gravitational acceleration on the x, y, and z axes when the attitude changes.

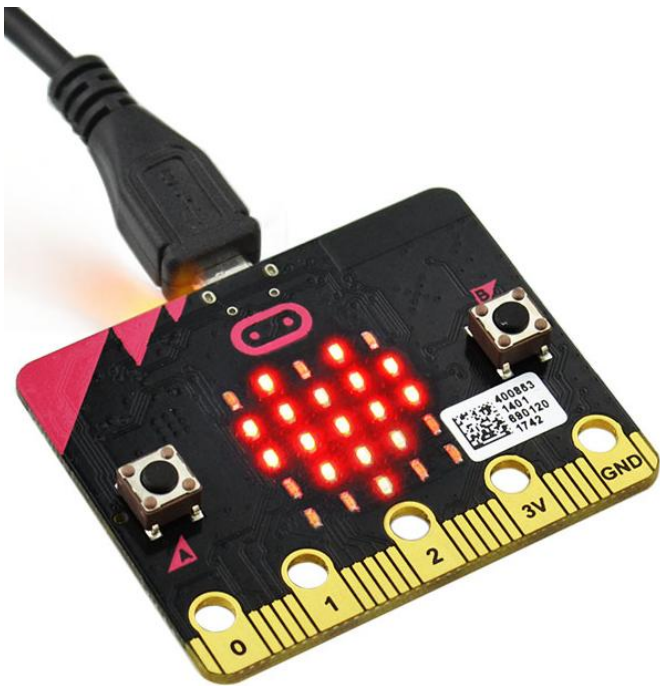
For this project, we will introduce the detection of several special postures by the accelerometer.

## 2.What You Need:

- Micro:bit Board\*1
- Micro USB Cable\*1

## 3.Wiring Up:

Connect micro:bit to your computer using micro USB cable.

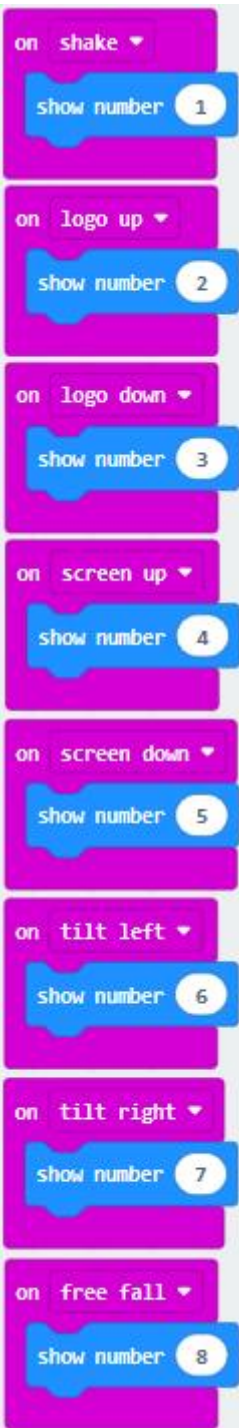


## 4.Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.


### Code 1:



The image shows a vertical stack of eight Scratch code blocks. Each block is a pink 'when triggered' block with a blue 'show number' block attached to it. The triggers are: 'on shake', 'on logo up', 'on logo down', 'on screen up', 'on screen down', 'on tilt left', 'on tilt right', and 'on free fall'. The numbers shown are 1 through 8 respectively.

on shake	1	Shake micro:bit
on logo up	2	Micro:bit shows 1 Micro:bit' s logo is upward
on logo down	3	Micro:bit shows 2 Micro:bit' s logo is download
on screen up	4	Micro:bit shows 3 Dot matrix of micro:bit is upward
on screen down	5	Micro:bit shows 4 Dot matrix of micro:bit is downward
on tilt left	6	Micro:bit shows 5 micro:bit tilts left
on tilt right	7	Micro:bit shows 6 micro:bit tilts right
on free fall	8	Micro:bit shows 7 When the Micro:bit board is free fall
		Micro:bit shows 8

## Code 2:



```

on start
  serial redirect to USB

forever
  serial write value "X" = acceleration (mg) x
  pause (ms) 100
  serial write value "Y" = acceleration (mg) y
  pause (ms) 100
  serial write value "Z" = acceleration (mg) z
  pause (ms) 100

```

"on start" : command block only runs once to start program.

The program under the block "forever" runs cyclically.

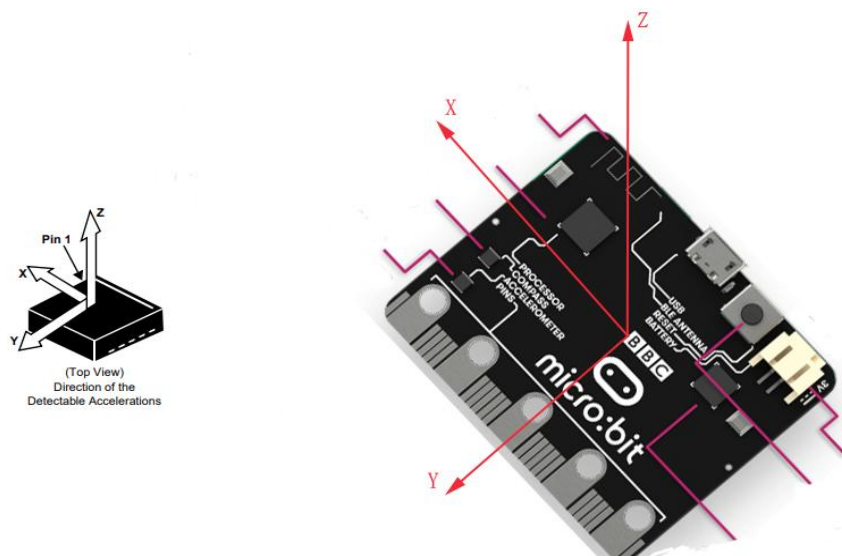
Serial writes value X=acceleration value of x axis  
Delay in 100ms

Serial writes value Y=acceleration value of Y axis  
Delay in 100ms

Serial writes value Z=acceleration value of Z axis  
Delay in 100ms

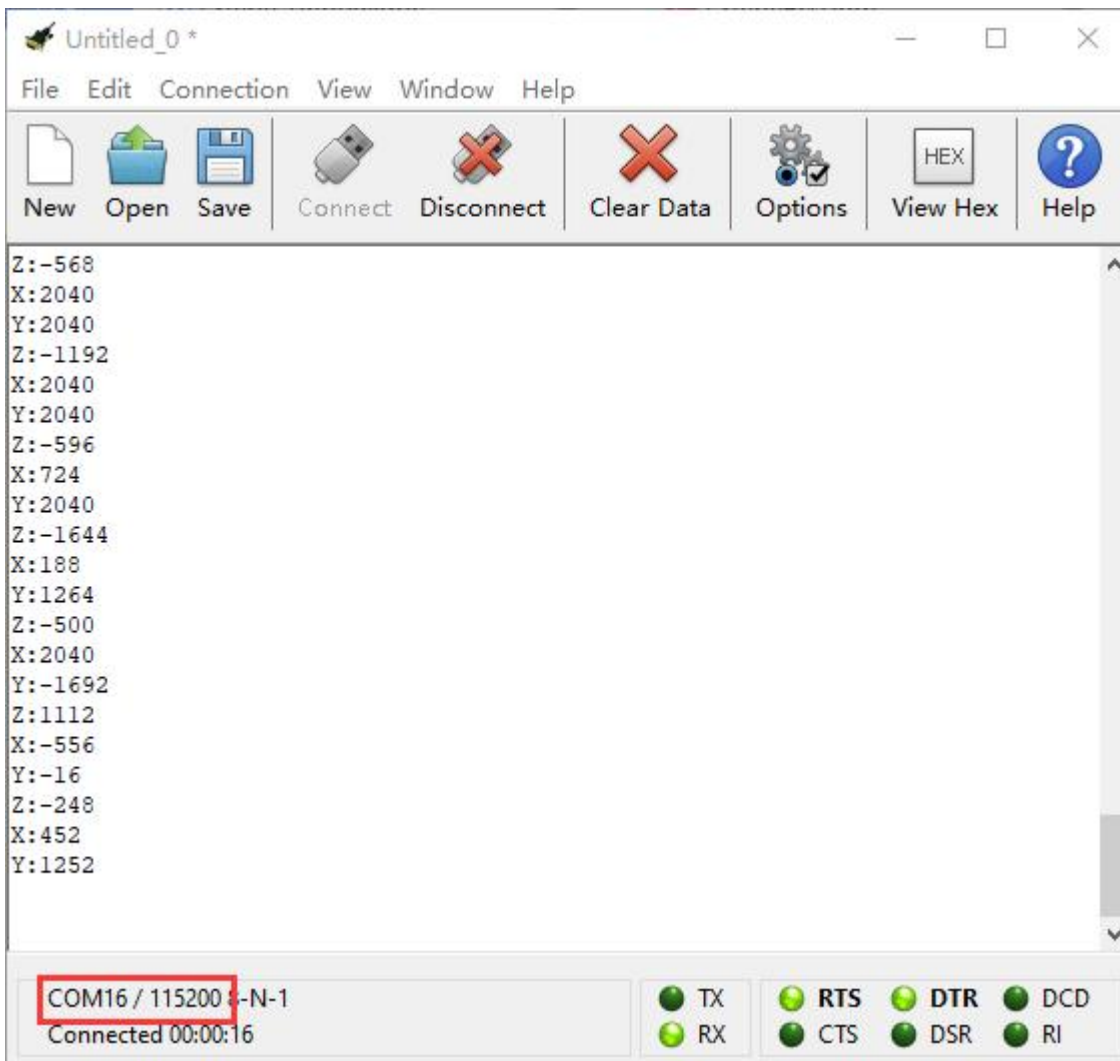
Download code 2 to micro:bit, keep USB cable connected.

The coordinates of the micro:bit accelerometer are shown in the following figure:



Hook the components up, upload code, power on and open CoolTerm, click Options to select SerialPort. Set COM port and baud rate is 115200(the baud rate of USB serial communication of micro:bit is 115200 through the test). Click "OK" and "Connect" .

The serial monitor separately displays the decomposition value of acceleration on the X-axis, Y-axis, and Z-axis, as well as the synthesis value of acceleration (the synthesis of gravitational acceleration and other external forces).

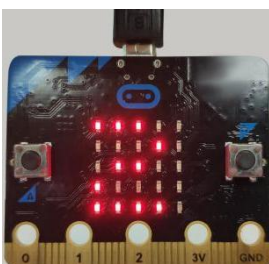


## 5. Test Result:

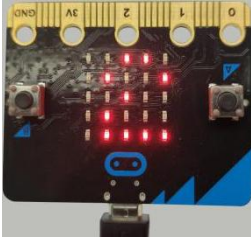
Download code 1 to micro:bit and plug in power with USB cable.

Shake micro:bit then the number 1 appears.

When the logo points to the North, the number 2 is displayed:



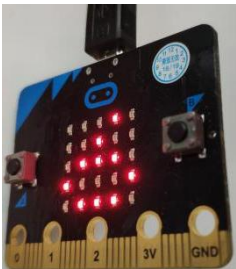
When the logo points to the South, the number 3 is displayed:



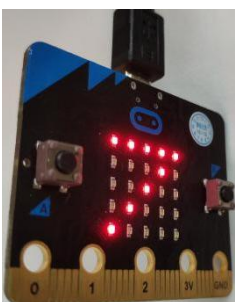
When the screen is upward, the number 4 is shown.

On the contrary, the number 5 is displayed when the screen is downward.

When micro:bit board is tilt to the left, number 6 is shown.

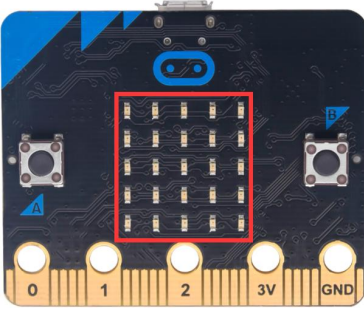


When micro:bit board is inclined to the right, number 7 is displayed.



When it is free fall (accidentally making it fall), number 8 appear on dot matrix. (Note: we don't recommend you to make it free fall, it will cause board damage)

## Project 9: Detect Light Intensity by Micro:bit



### 1. Description

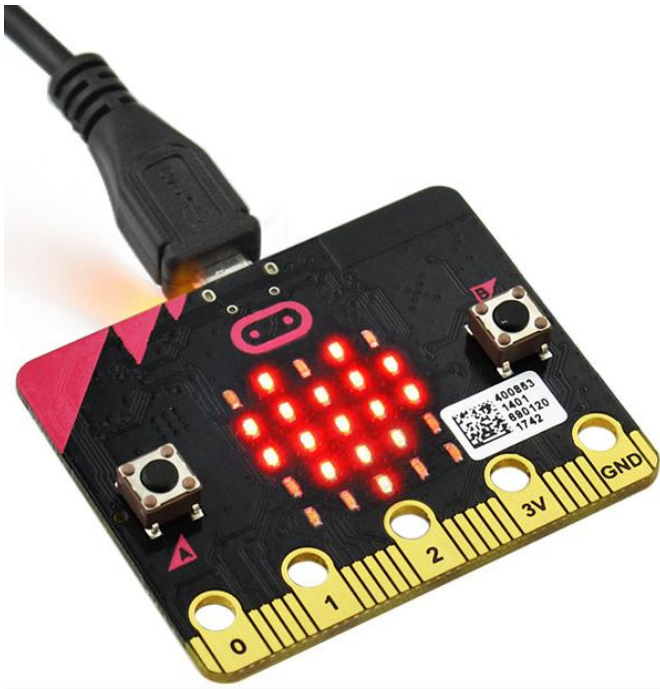
This project will introduce how micro:bit detects the external light intensity. Since micro:bit doesn't come with photosensitive sensor, the detection of light intensity is completed through the LED matrix. When the light irradiates the LED matrix, the voltage change will be produced. Therefore, we could determine the light intensity by voltage change.

### 2. What You Need:

- Micro:bit Board\*1
- Micro USB Cable\*1

### 3. Wiring Up:


Connect micro:bit to your computer using micro USB cable.



#### 4. Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.



“on start”: command block runs once to start program.

Serial redirects to USB

The program under the block “forever” runs cyclically.

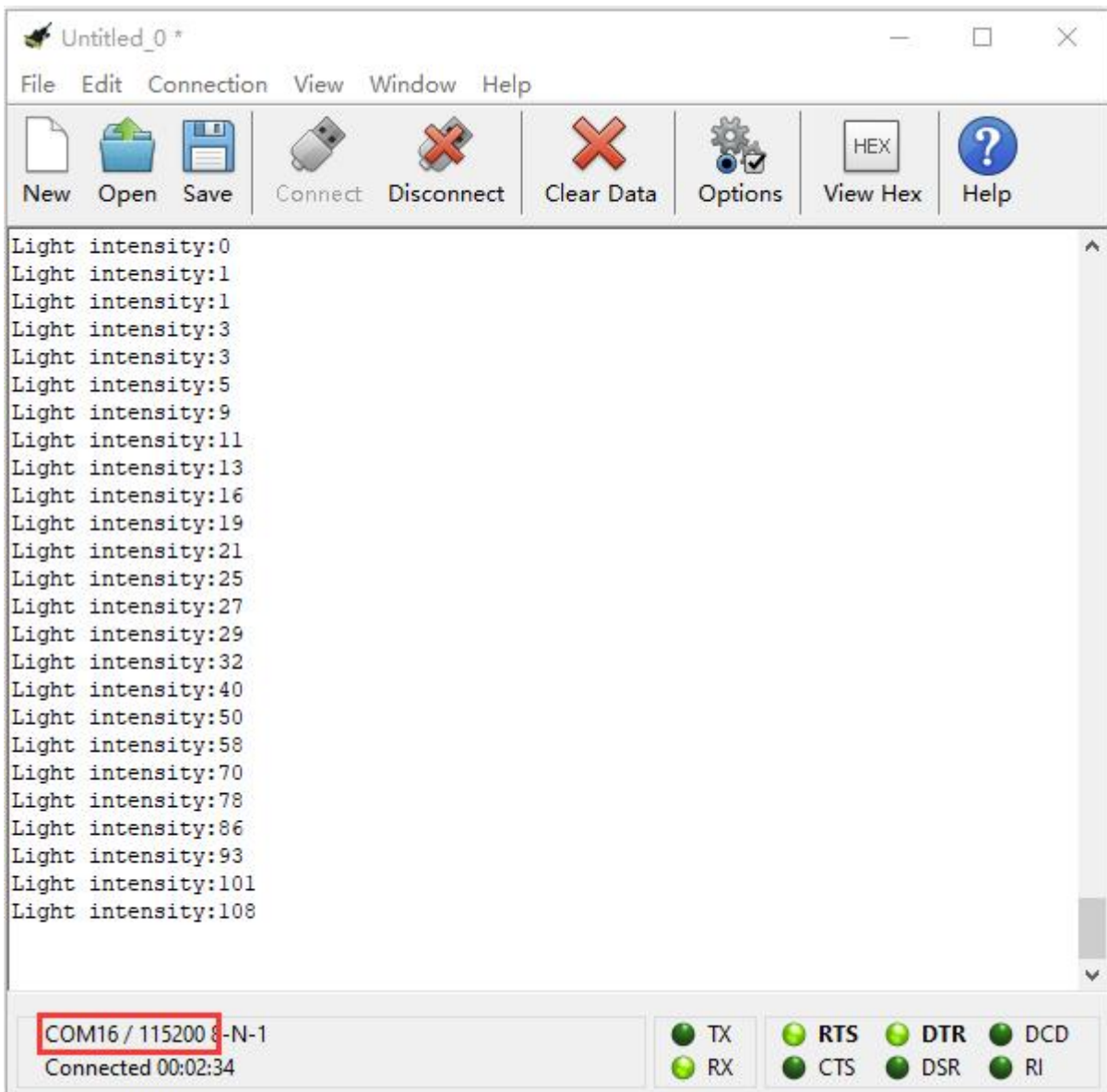
Serial write value “Light intensity” = light level

Delay in 100ms

## 5. Test Result:

Wire the components up, download code to micro:bit, keep USB cable connected and open CoolTerm, click Options to select SerialPort. Set COM port and baud rate is 115200(the baud rate of USB serial communication of micro:bit is 115200 through the test). Click “OK” and “Connect” .

CoolTerm monitor displays the light intensity value. When covering LED dot matrix by hand, the light intensity is 0; when putting it under the sun, the intensity value varies with the light intensity.



## Project 10: Bluetooth Wireless Communication



### 1. Description

Micro:bit board comes with NRF51822 processor, Bluetooth and 2.4GHz RF antenna, which work with Bluetooth and 2.4G wireless communication.

In this project, we connect cellphone to micro:bit motherboard to complete the wireless connection.

## 2. What You Need:

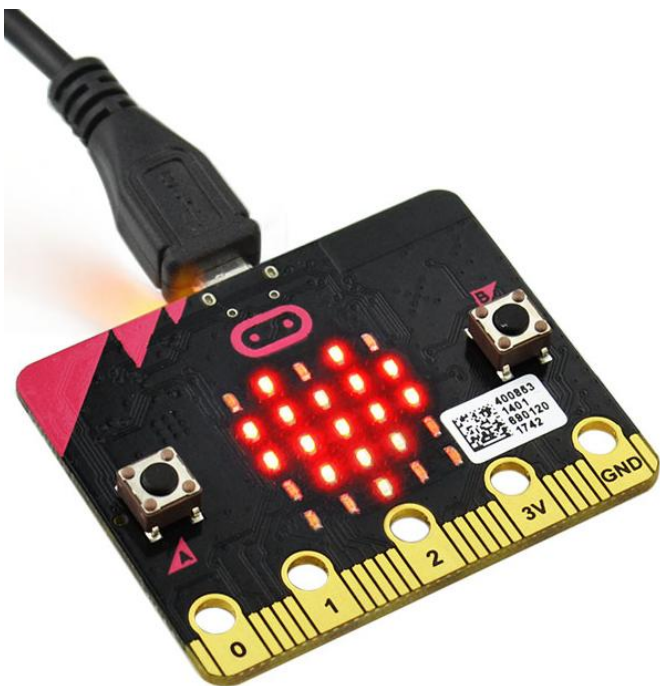
Micro:bit Board\*1

Micro USB Cable\*1

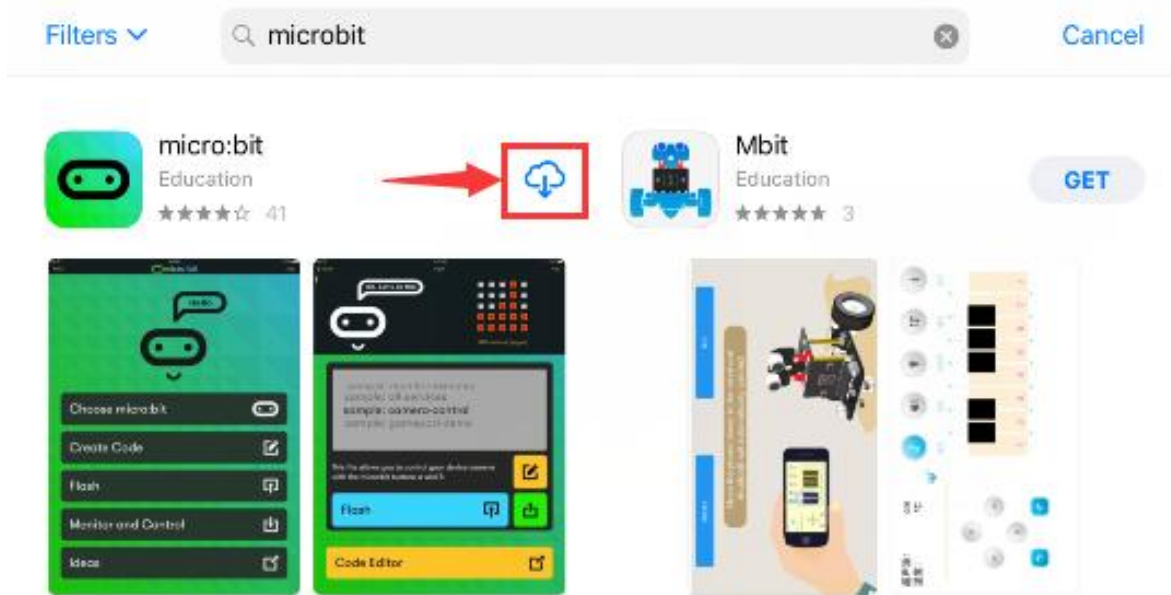
- Cellphone\*1

## 3. Wiring Up:

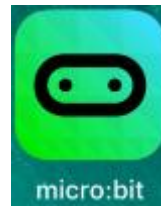
Connect micro:bit to your computer using micro USB cable

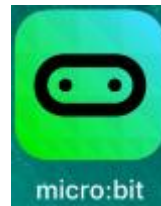


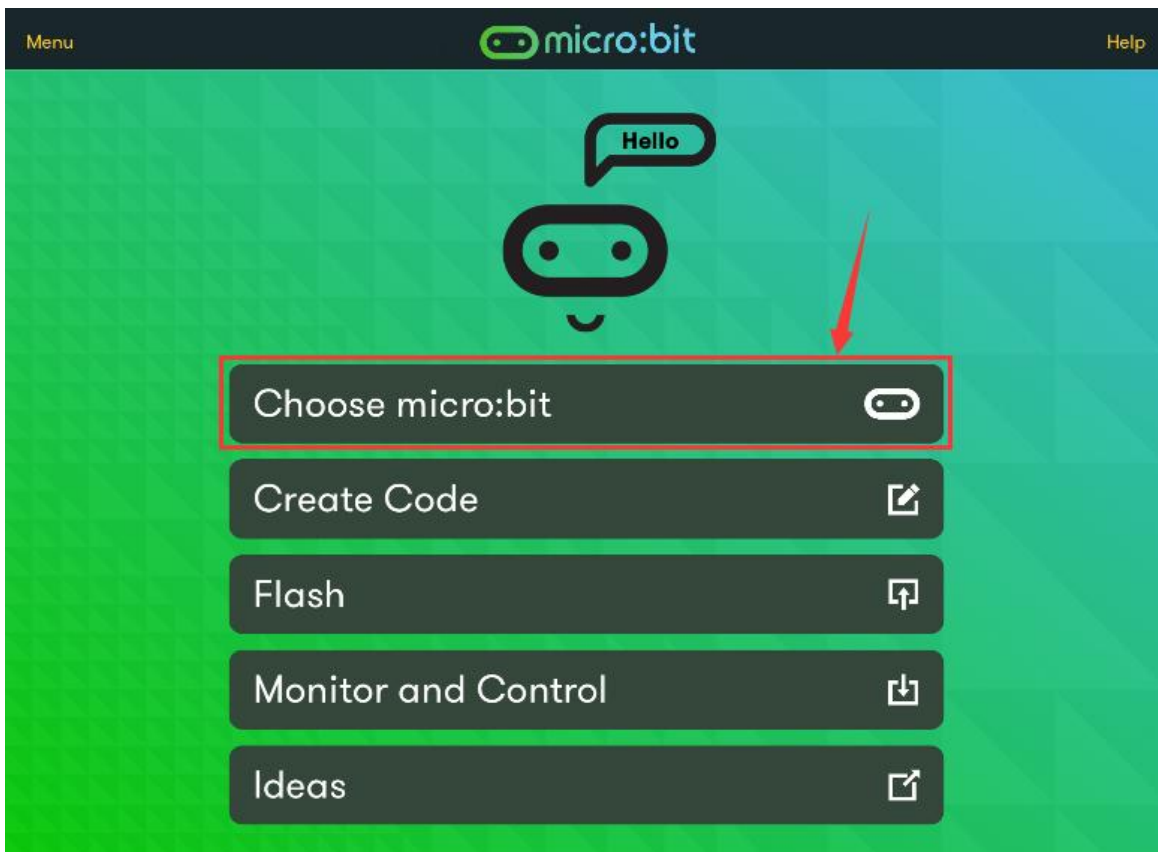
Search micro bit in App store and install it.



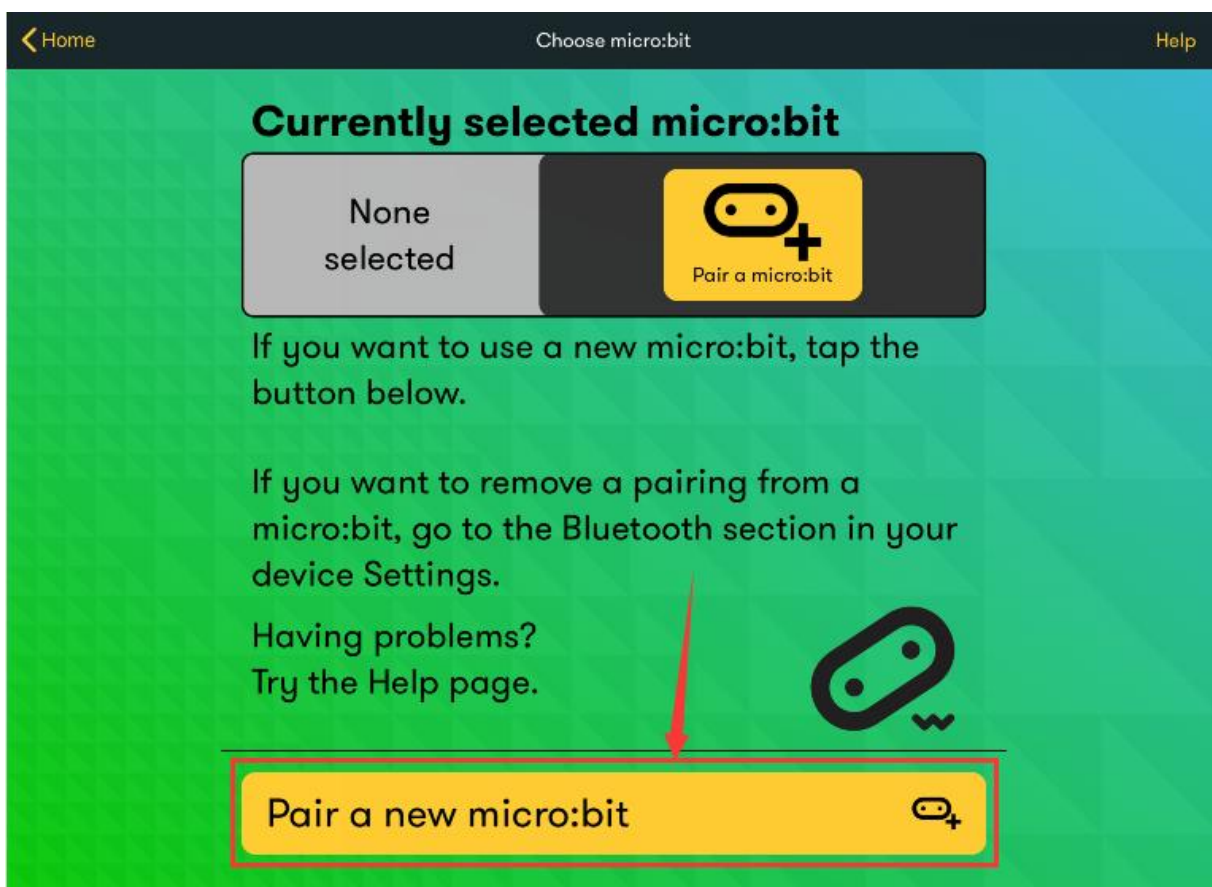
Pair BBC micro:bit with cellphone or iPad



a. After installing APP successfully, click  to enter app. plug in micro:bit , click “Choose Micro:bit” to start Bluetooth pairing.



b. Click "Pair a new Micro:bit"



c. Hold down button A and B, then press and release the reset button. LED dot matrix will show a pattern. At last, release the button A and B and click "Next" .

**How to pair your micro:bit**

**HOLD**  
A + B

**PRESS**  
RESET

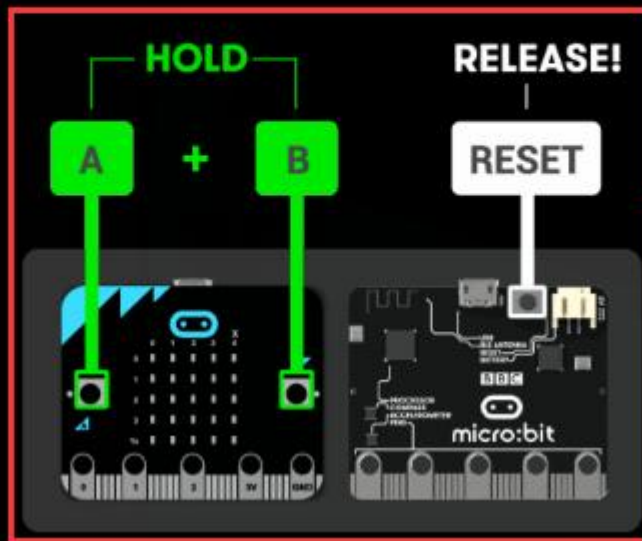
Let's do this

Step 1  
HOLD the A and B buttons and  
PRESS and RELEASE RESET

Cancel X

Next >

## How to pair your micro:bit



Step 1

HOLD the A and B buttons and  
PRESS and RELEASE RESET

Let's do this

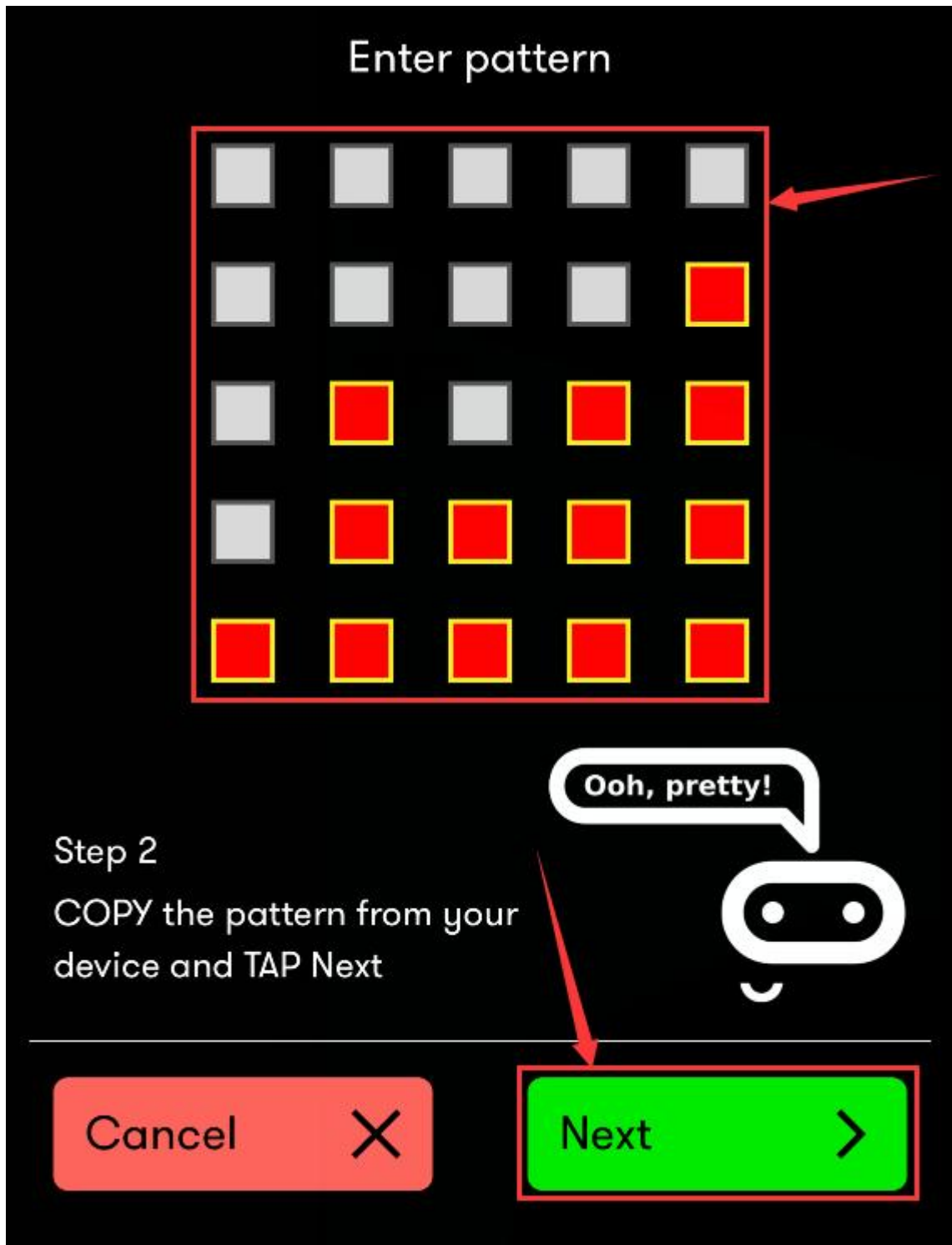


Cancel

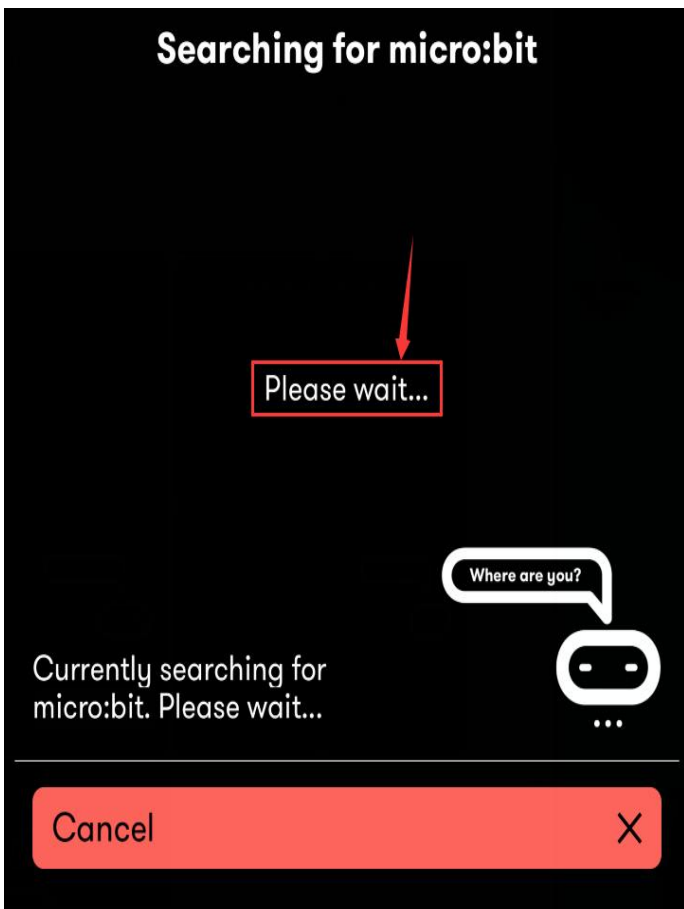
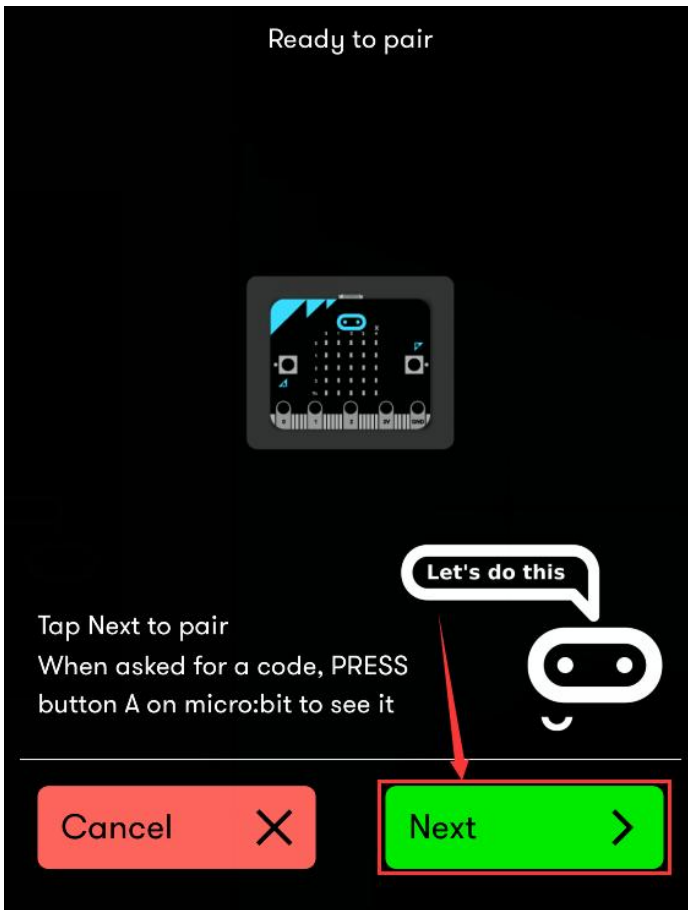


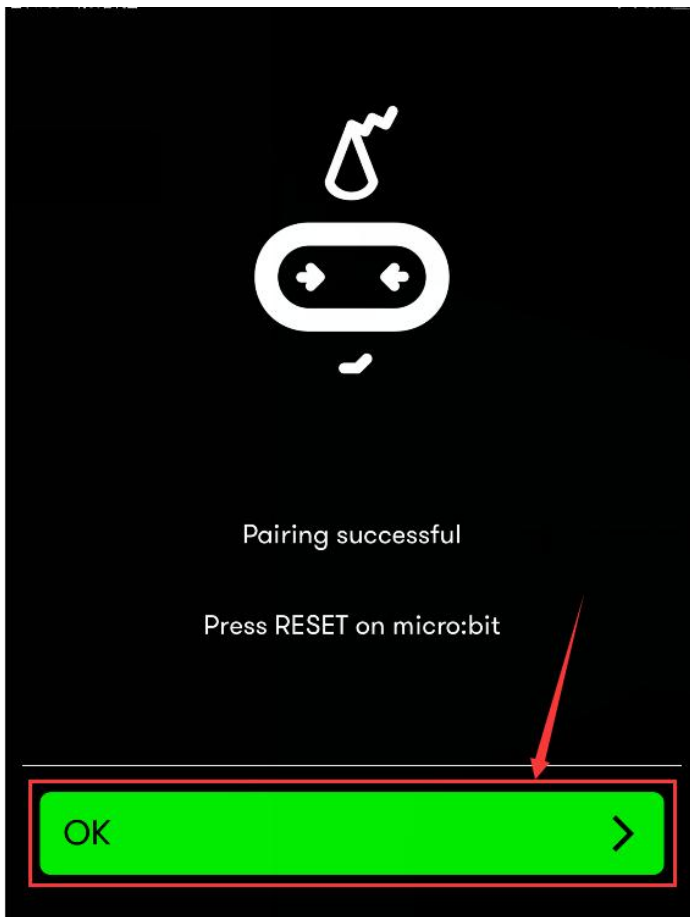
Next






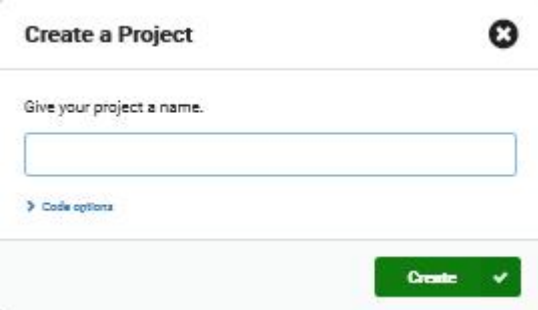

Tap "Next" to pair.

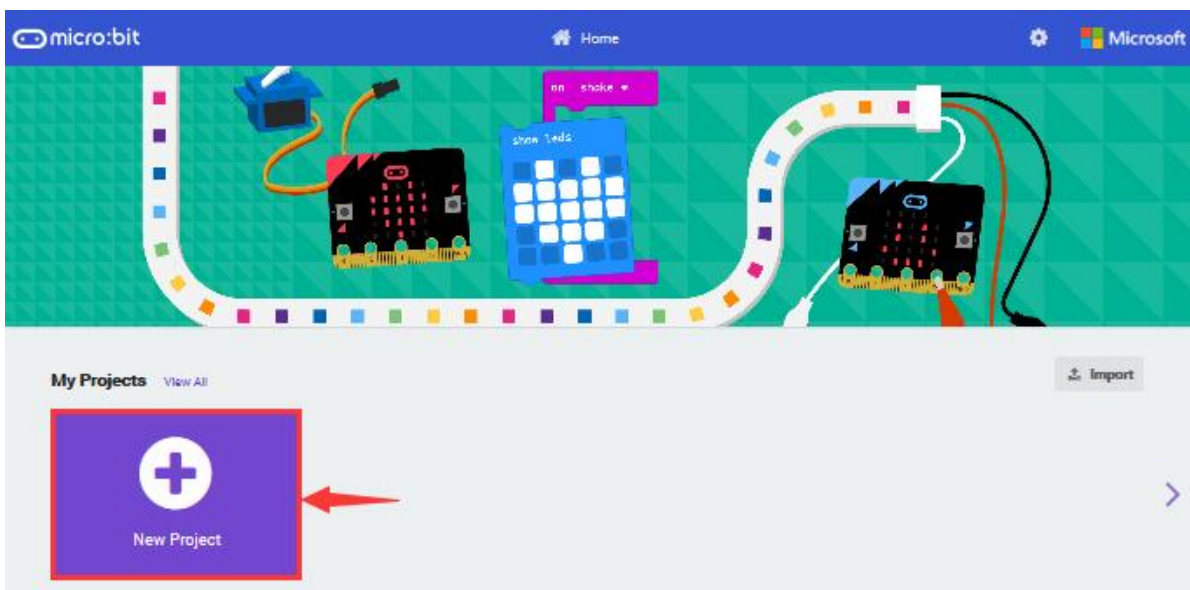
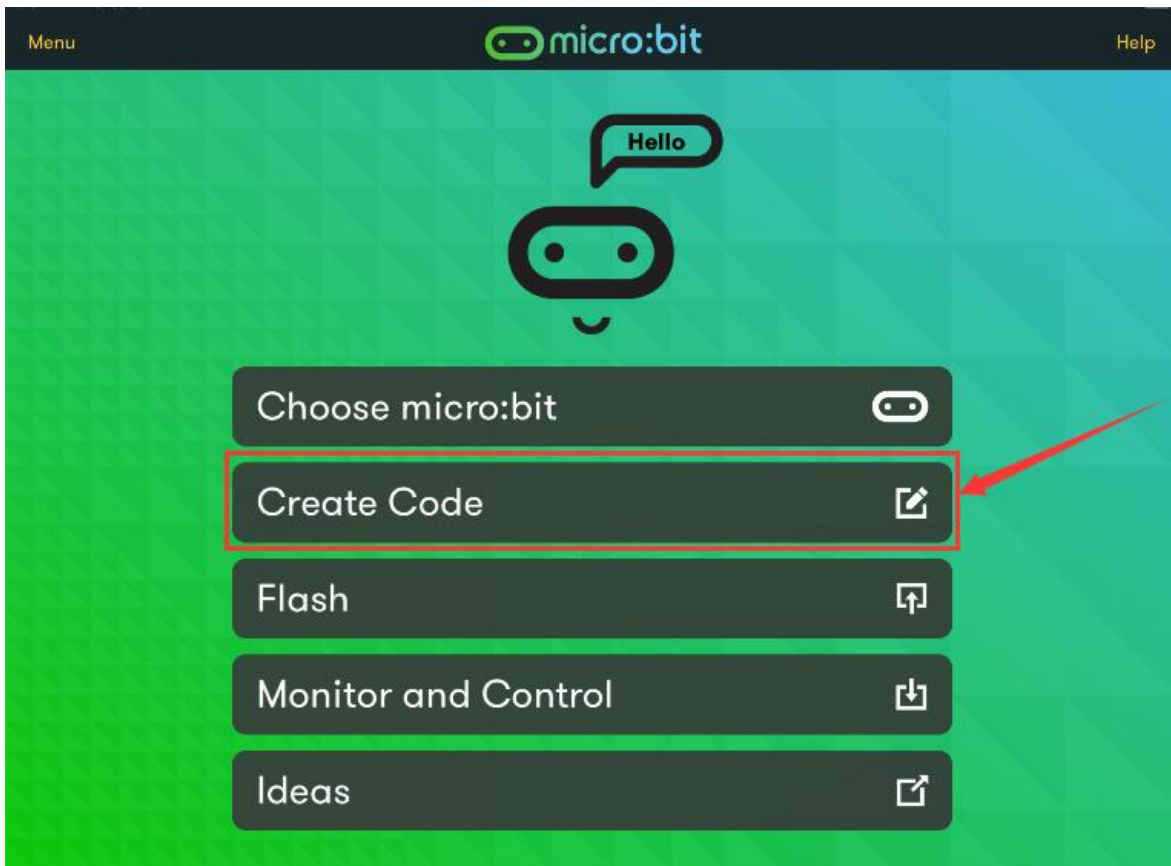


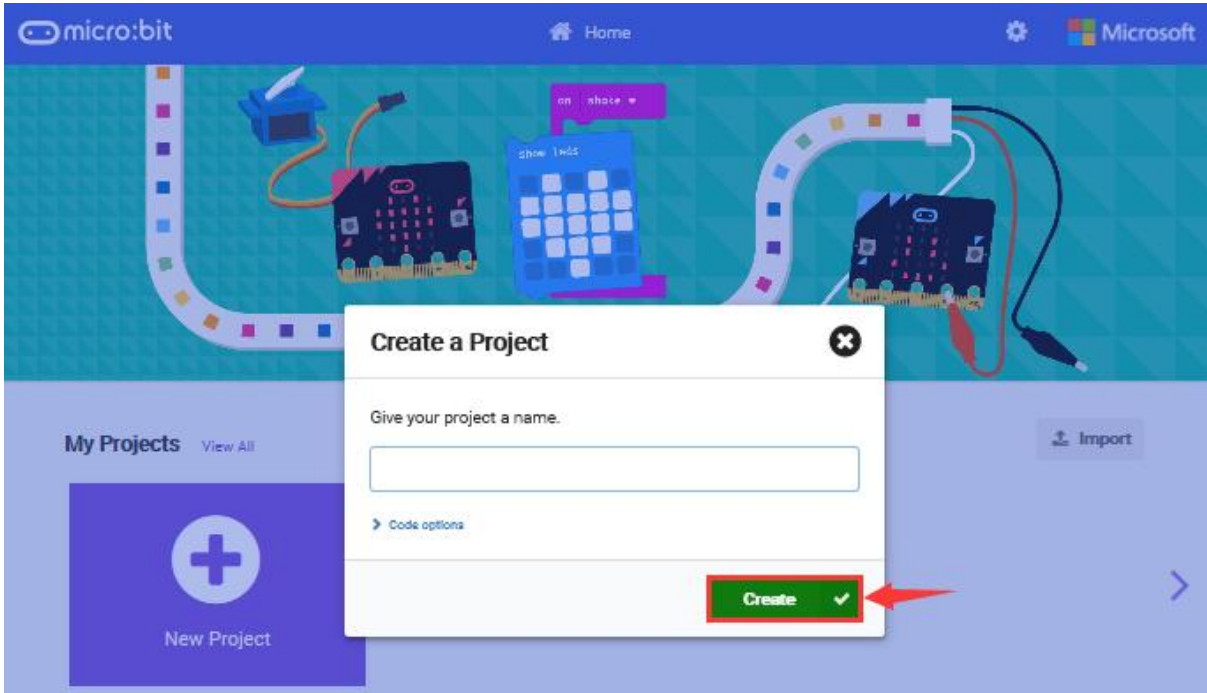


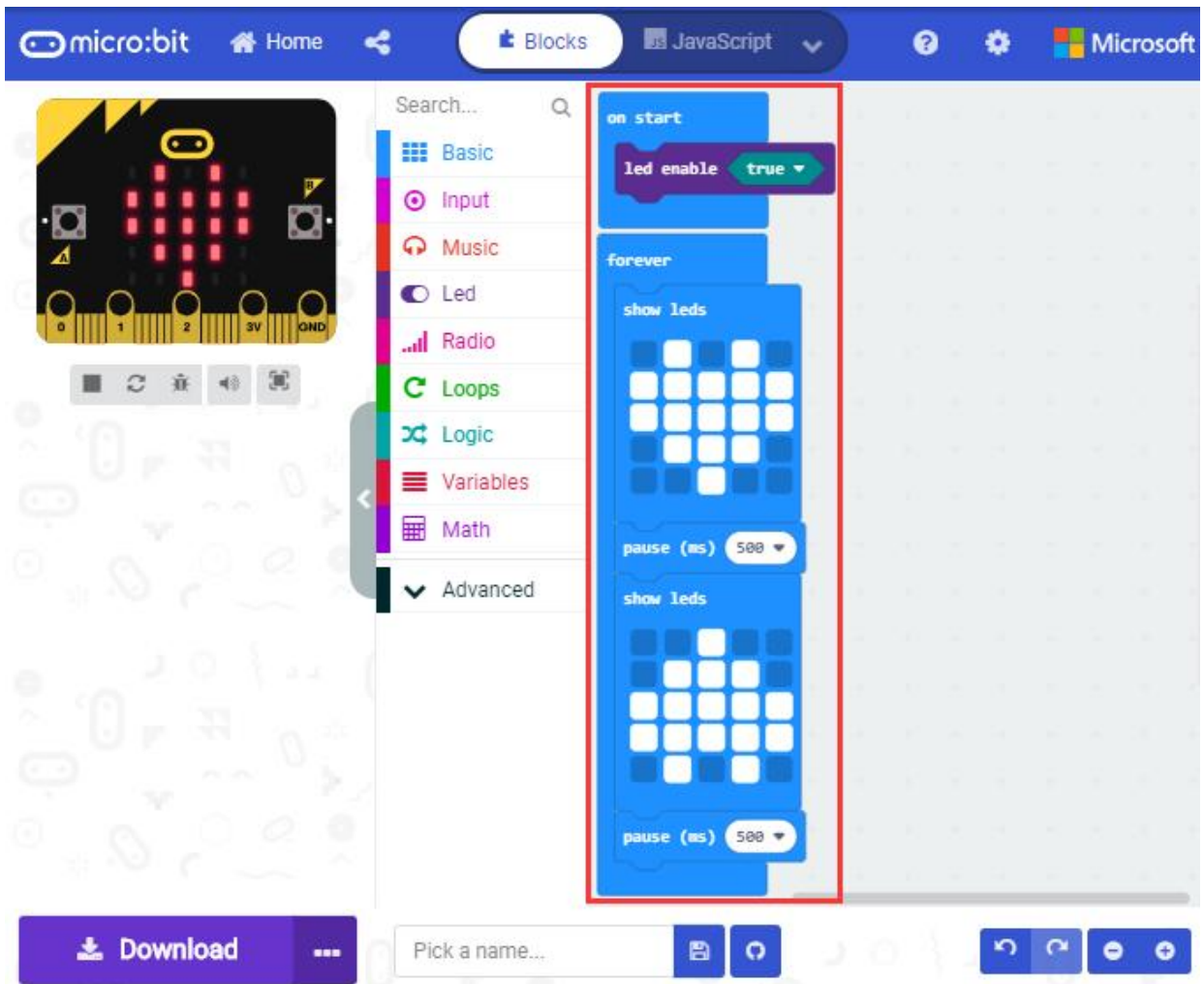
After pairing successfully, edit code via APP and upload code.

Select "Create Code" (Click  and appear the dialog

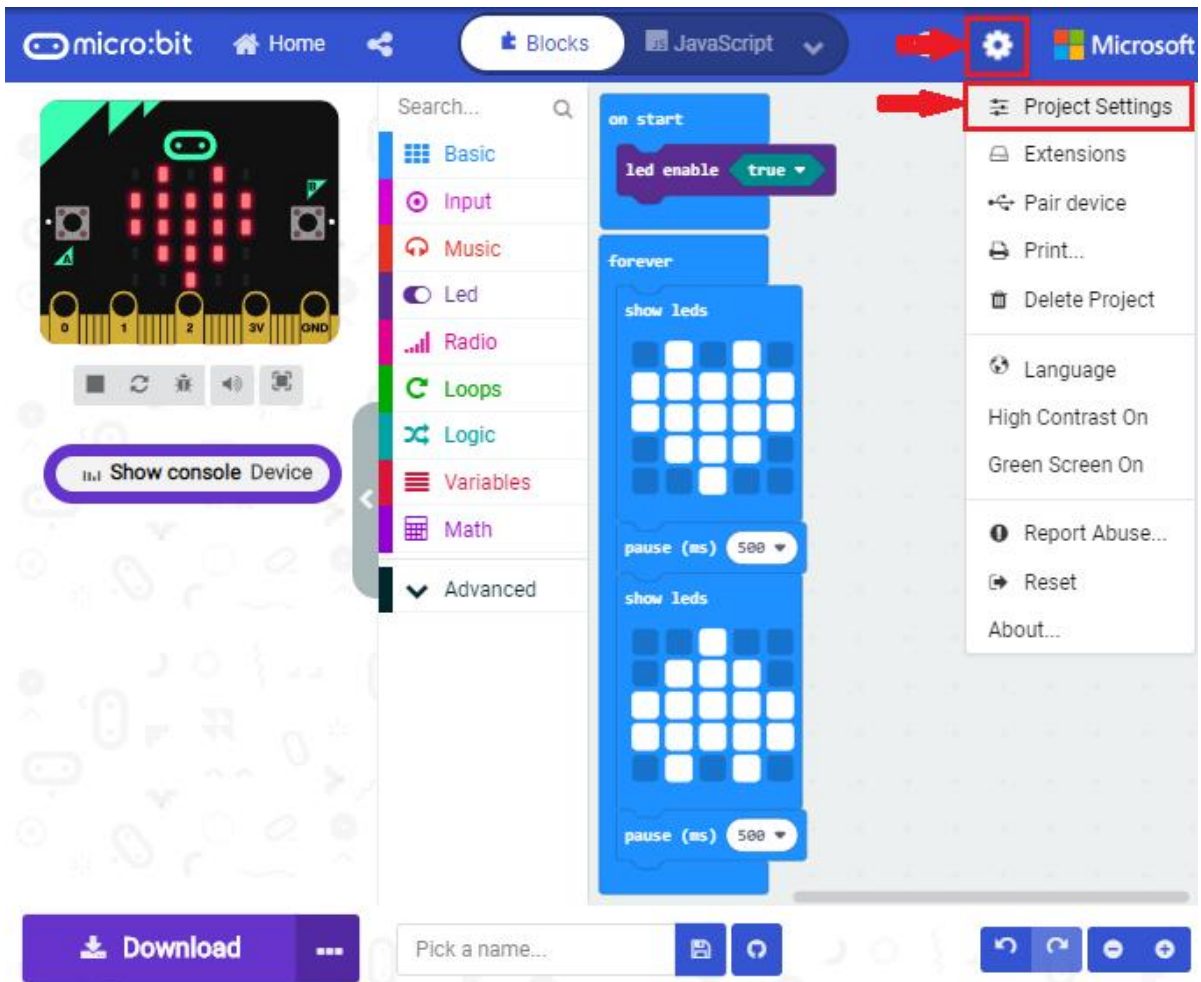
box , click icon  to enter the programming interface.




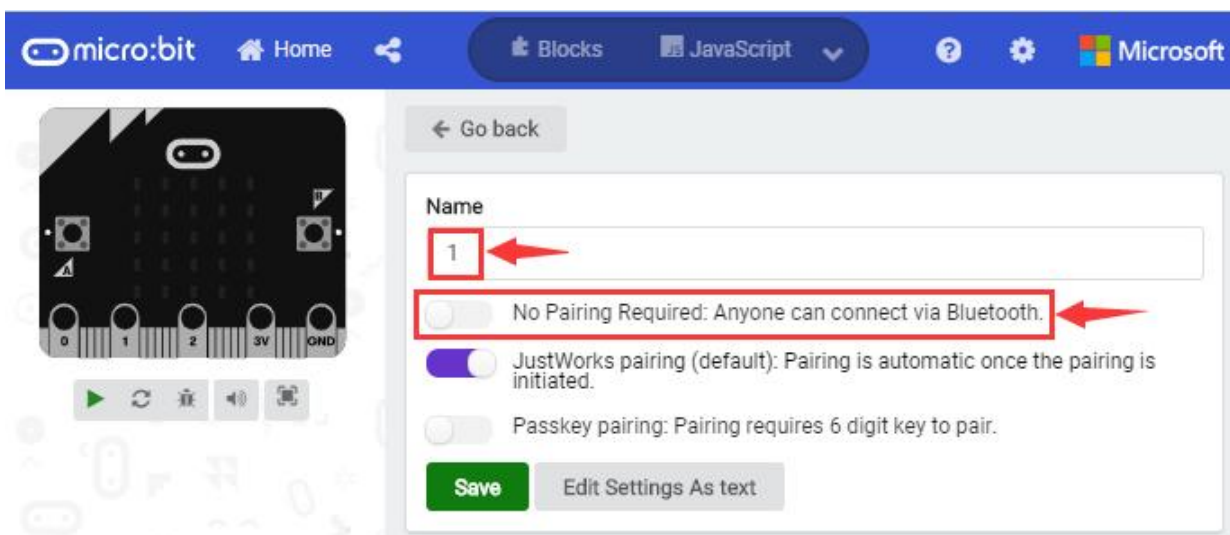


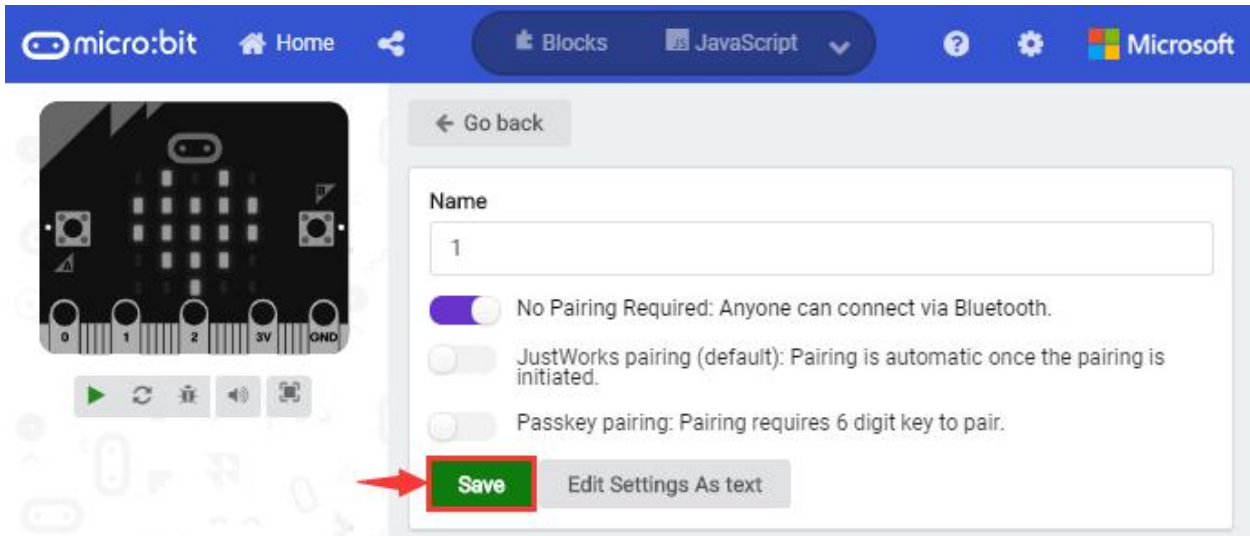


After finish programming, click  to select "Project Setting" .




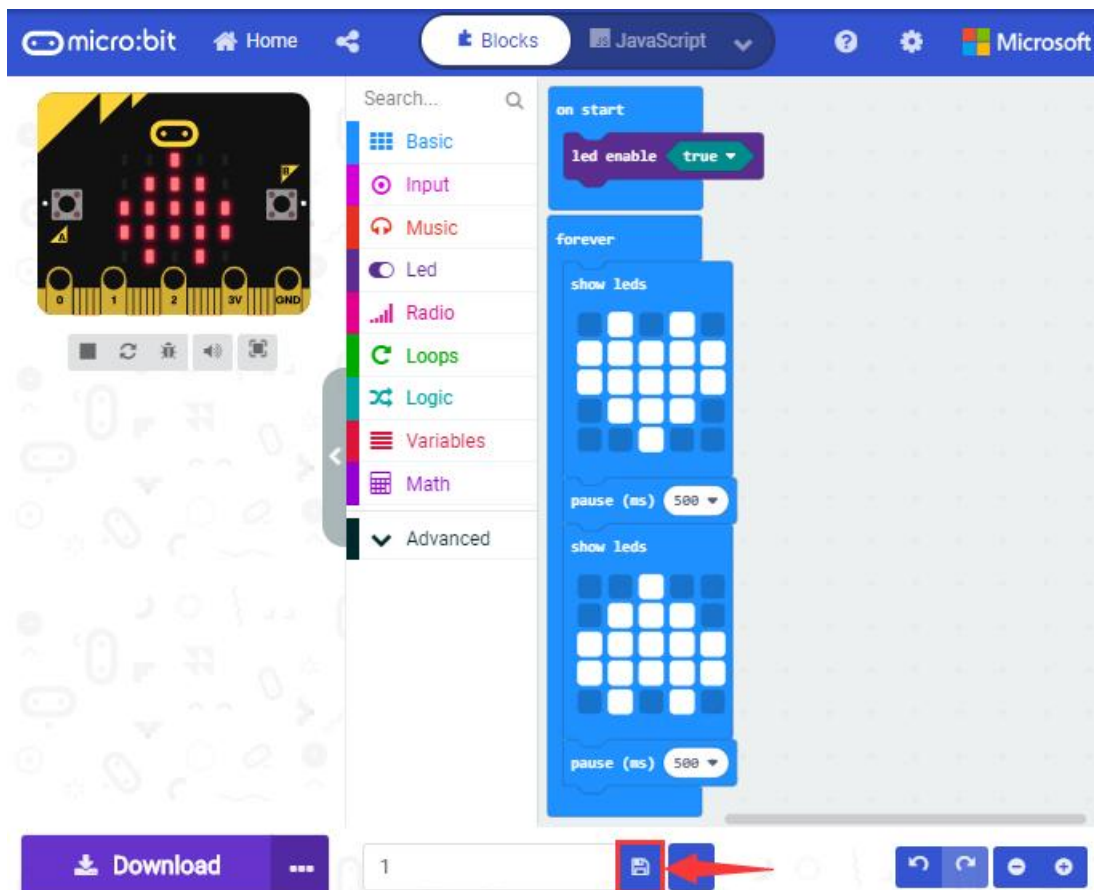
Input 1 in the blank box to name your project. Enable button  No Pairing Required: Anyone can connect via Bluetooth. The page will return to the initial page. Then click the button  and "Project Setting" to come back and tap "save"





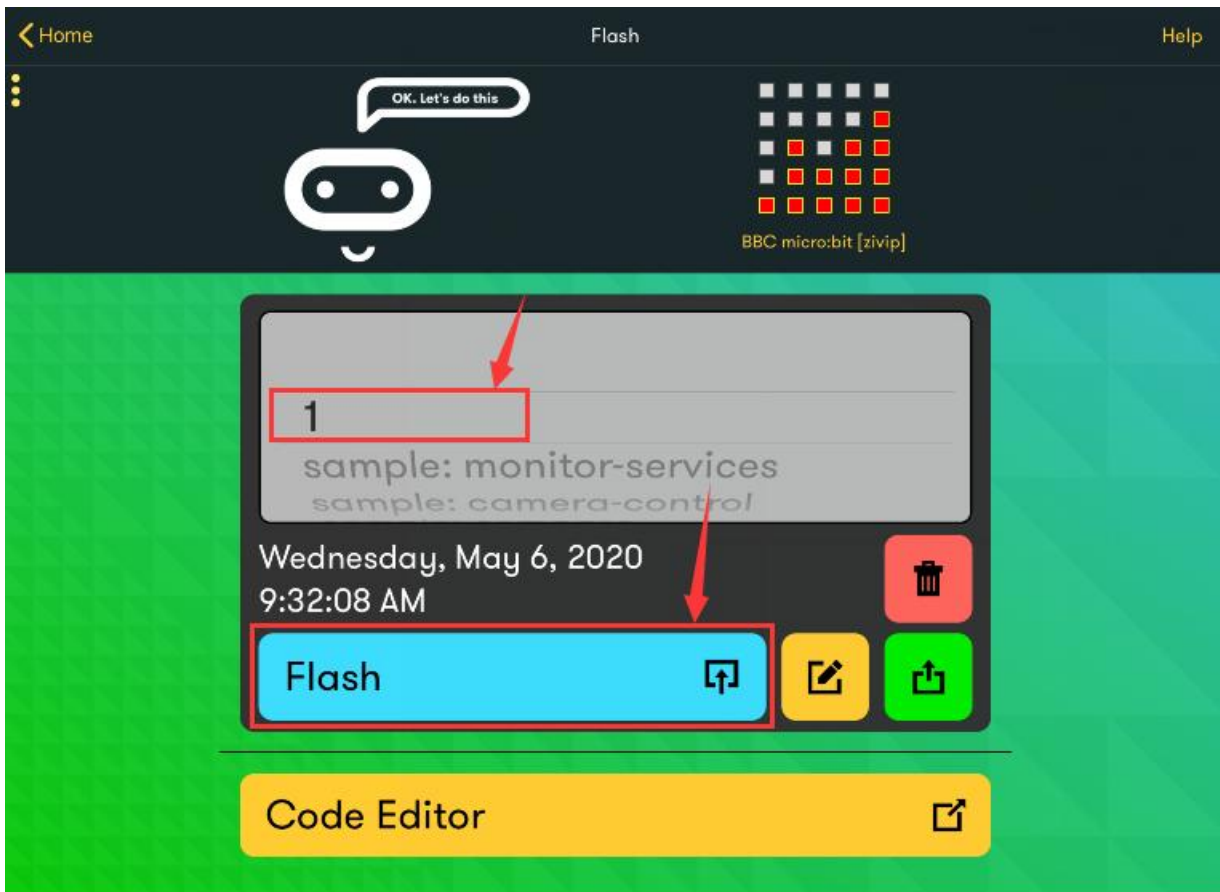
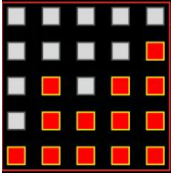
The program "1" is set well, click

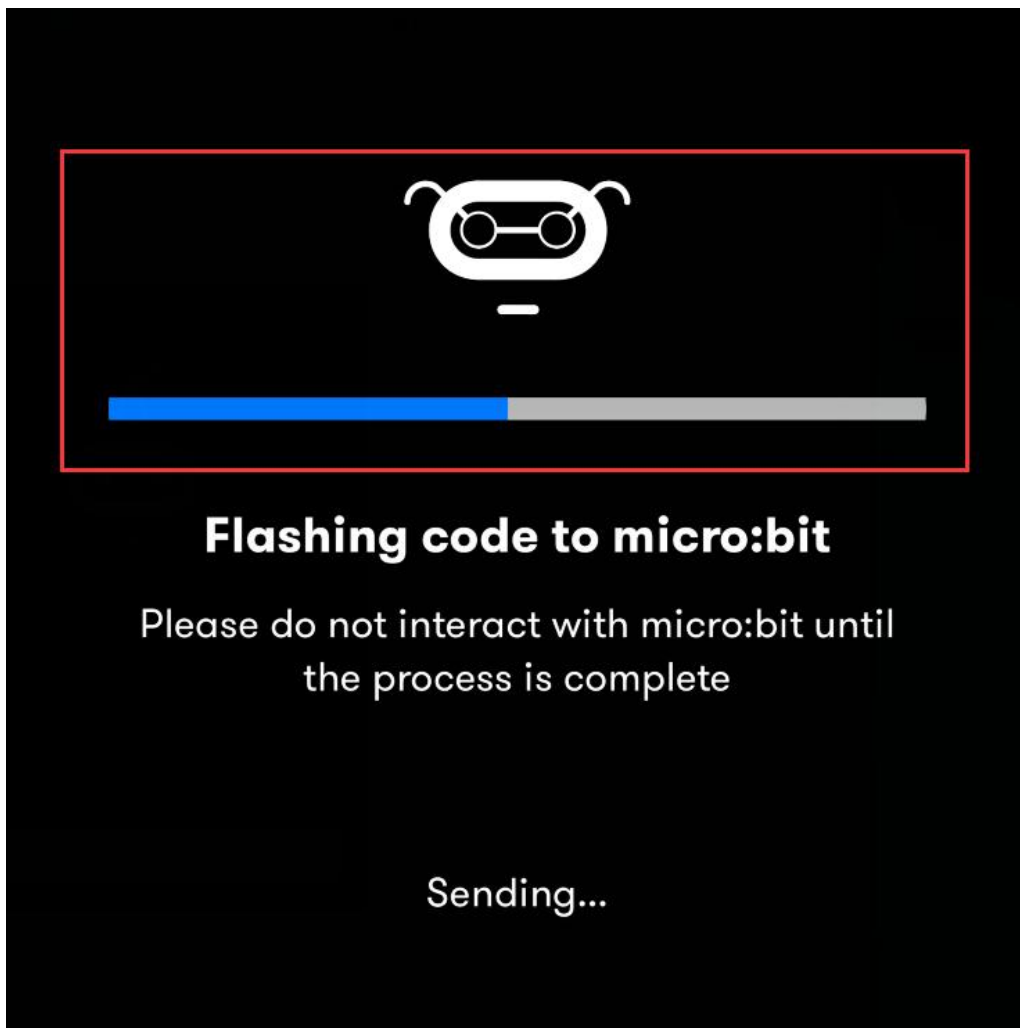
icon   to save the program.



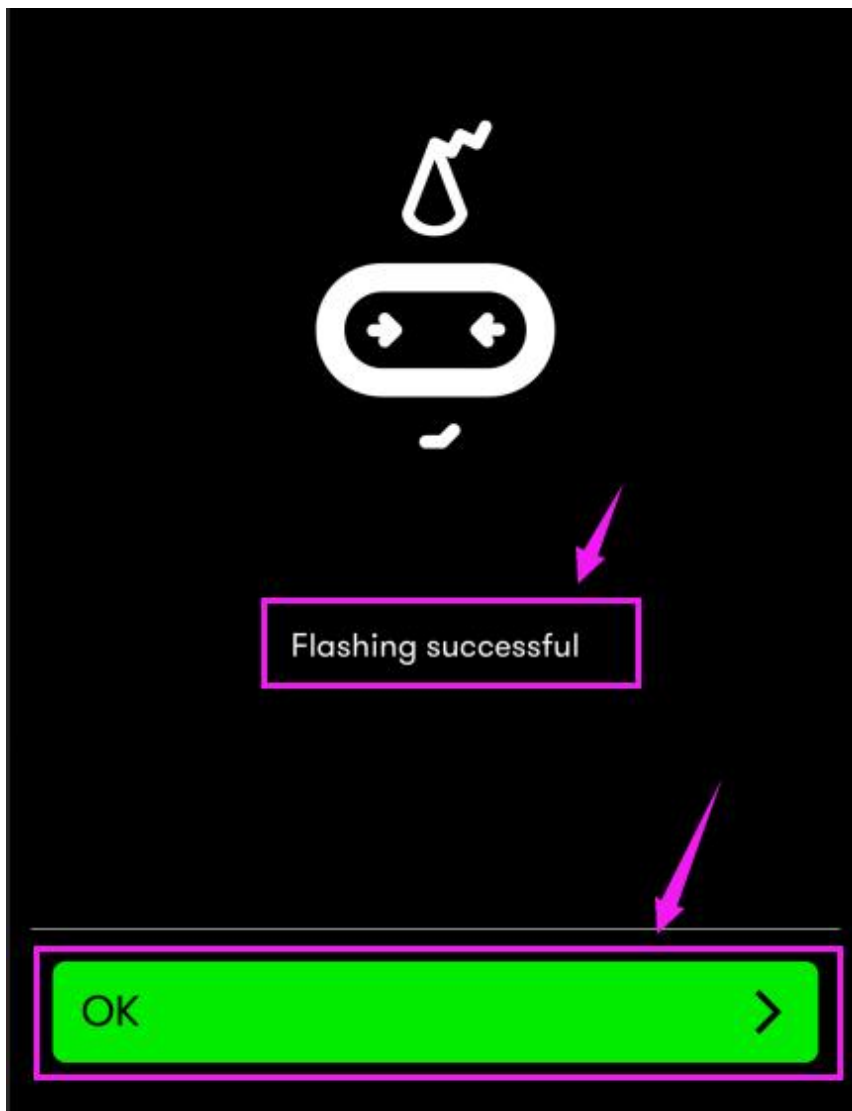
Then download the code to micro:bit directly(default program name is 1) and click "Flash" .

Note: the code will be downloaded after micro:bit shows the





Upload the program successfully, as shown below:



## Project 11: LED Blink

### 1. Description

LED blink is a basic experiment. You will learn how to make white LED blink through code. Please turn off dot matrix on micro:bit before testing.

## 2. What You Need:

Micro:bit Board\*1

EASY Plug Shield for micro bit V1.1\*1

Micro USB Cable\*1

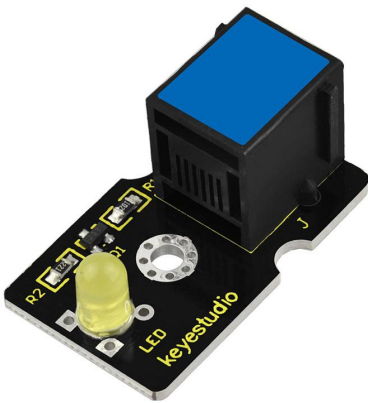
EASY Plug Yellow LED Module\*1

RJ11 Cable\*1

6-Slot AA Battery Holder\*1

1.5V AA Battery\*6

### EASY Plug Yellow LED Module



The LED light modules have shiny colors, ideal for Arduino starters. It can be easily connected to IO/Sensor shield.

**Note:** this module needs to be used together with EASY Plug Shield for micro bit V1.1. You can also choose other LED to emit different color of light like white, blue, green, yellow and red.

## 3. Specification:

- Interface: Easy plug
- Sensor type: Digital
- Working voltage: 5V
- LED color: yellow

- Easy to use
- Useful for light projects

### **EASY Plug Shield for micro bit V1.1**

Micro:bit is a basic development board designed by the British Broadcasting Corporation for youth programming education. It supports the PXT graphical programming interface developed by Microsoft, without the need to download an additional compiler, and can be used under Windows, macOS, IOS, Android and other operating systems.

We combine EASY Plug shield with micro:bit due to the inconvenience of wiring up micro:bit .

The golden finger interfaces ,as well as 10 pcs easy plug ports (RJ11 6P6C interfaces)could be connected to other modules and sensors, therefore, you don' t need to worry about wiring up components incorrectly.

The shield comes with 4 pcs WS2812 LEDs controlled by P9, P0 controls passive buzzer; and two dial switches--Power\_Switch and Voltmeter\_Switch(3.3V, 5V).

The voltage of power supply is DC 6-10V

**The Easy Plug port only supports the sensors and modules with RJ11 6P6C port.**

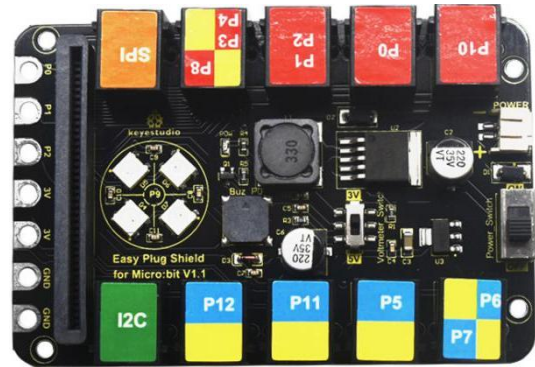
#### 4. Specification:

Power supply: DC 6-10V

Output current: 1.5A

Interface: RJ11 6P6C interface and golden finger interface

Size: 98\*65\*17mm



#### 5. Interface Description:

G: GND

V: Voltmeter\_Switch control, dial to 5V end, 5V; dial to 3V end, 3.3V

I2C Communication Port

G	V	SDA
	SCL	

SDA: P20

SCL: P19

SPI Communication Port

G	V	P16
SCK	MISO	MOSI

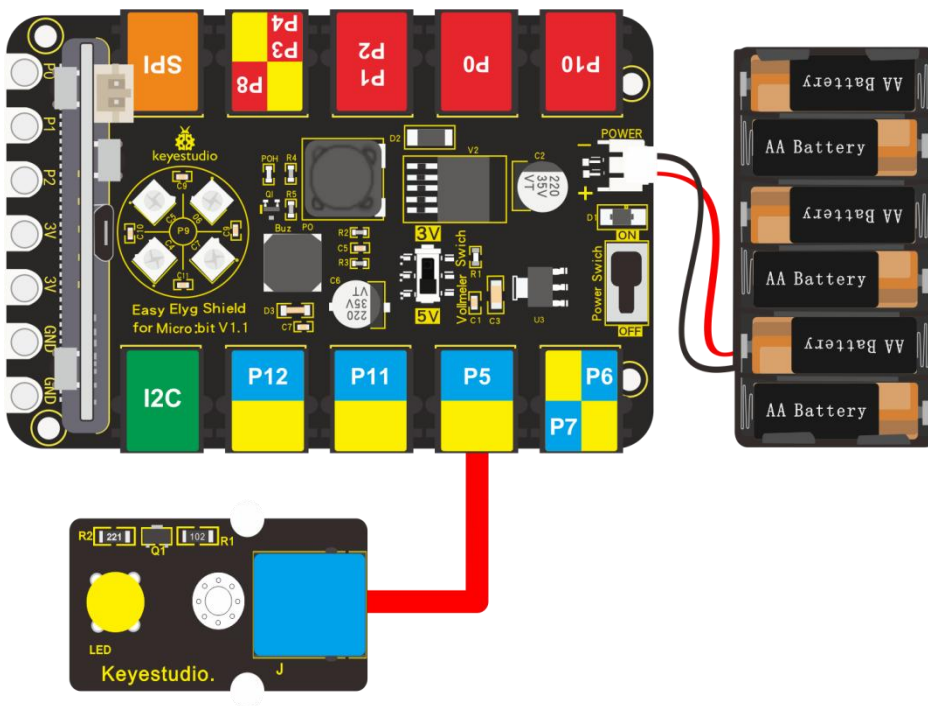
MOSI: P15

MISO: P14

SCK: P13

## 6. Wiring Up:

Insert micro:bit onto EASY Plug shield, link white LED module with P5 port of shield and plug in power.



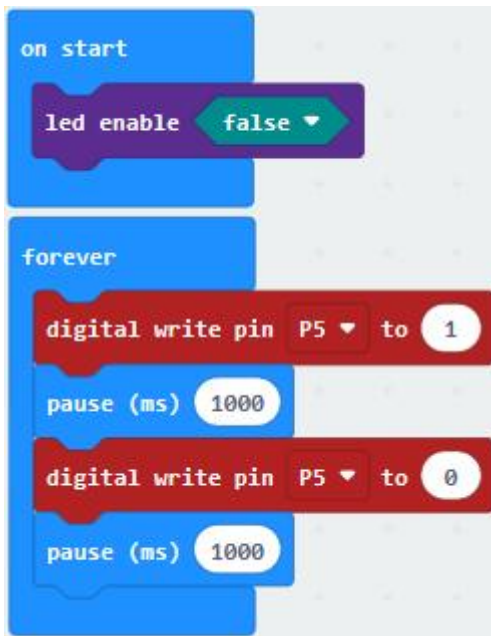
Note: Dial Voltmeter\_Switch to 3V end

## 7. Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program.

The following test code is as for your reference.



"on start" : command block only runs once to start program.

Turn off dot matrix on micro:bit

The program under the block "forever" runs cyclically.

Set P5 to high level(1), turn on LED

Delay in 1000ms

Set P5 to low level (0) , turn off LED

Delay in 1000ms

## 8. Test Result:

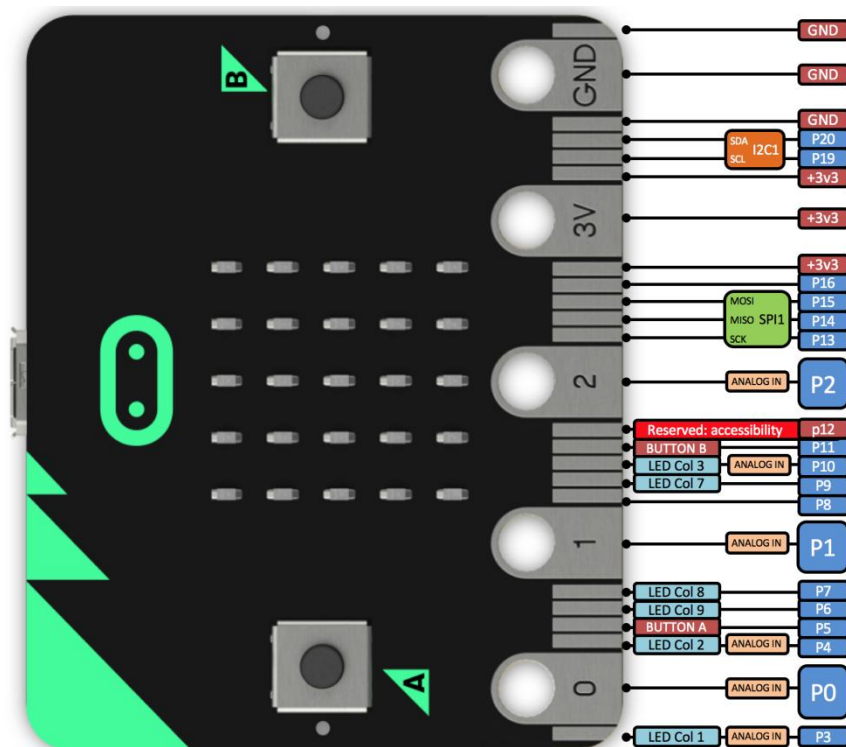
Wire up, dial Voltmeter\_Switch to 5V end, plug in external power and

dial Power\_Switch to ON end. Upload code to micro:bit and you will view LED flashing, with interval of 1s.

## Project 12: Breath

### 1. Description

The light breath experiment is a little bit similar to the previous project. This time we connect the EASY Plug Red Led module to the EASY Plug Shield for micro bit V1.1. Connect the pin of LED module to P10 of micro:bit. From the Pinout diagram of micro:bit, you can get the P10 can be used as Analog IN. This lesson you will learn how to control the brightness of LED on the module, gradually becoming brighter and dimming, just like the LED is breathing.

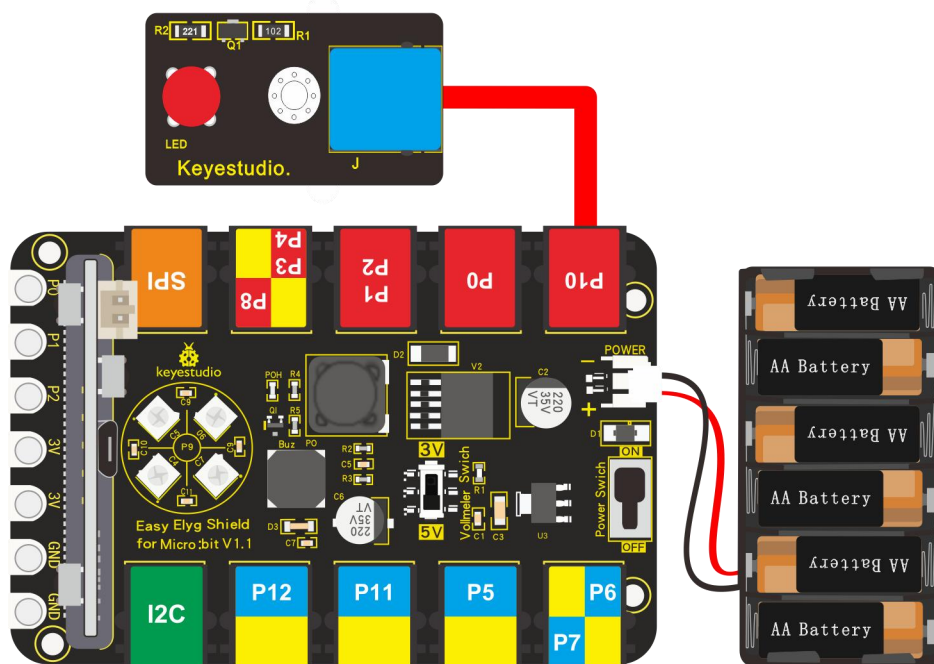


## 2. What You Need:

- Micro:bit Board\*1
- EASY Plug Shield for micro bit V1.1\*1
- Micro USB Cable\*1
- EASY Plug Red LED Module\*1
- RJ11 Cable\*1
- 6-Slot AA Battery Holder\*1
- 1.5V AA Battery\*6

## 3. Wiring Up:

Insert micro:bit onto EASY Plug shield, connect red LED module to P10 of shield with a RJ11 cable, and plug in external power.




Note: Dial Voltmeter\_Switch to 3V end

#### 4. Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.



“on start” : command block only runs once to start program.  
Turn off dot matrix on micro:bit  
The program under the block “forever” runs cyclically.  
When val<1024, run the program in the do block  
Set val to val+1  
Set analog value of P10 to val  
Delay in 5ms  
When val>0, run the program in the do block  
Set val to val-1  
Set the analog value of P10 to val.  
Delay in 5ms

#### 5. Test Result:

Wire up, dial Voltmeter\_Switch to 3V end, plug in external power and dial Power\_Switch to ON end and upload code to micro:bit.

You will find LED of module get brighter then darker, like human breath.

## **Project 13: Blink and Breath**

### **1. Description**

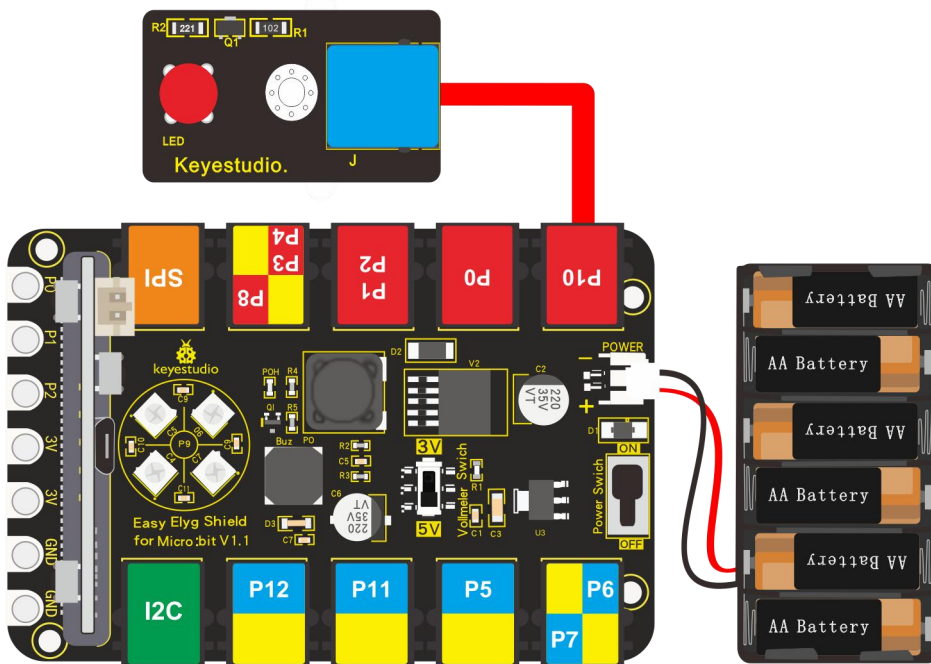
In this project, we will combine LED flash and breathing effect together.

### **2. What You Need:**

- Micro:bit Board\*1
- EASY Plug Shield for micro bit V1.1\*1
- Micro USB Cable\*1
- EASY Plug Red LED Module\*1
- RJ11 Cable\*1
- 6-Slot AA Battery Holder\*1 1.5V AA Battery\*6

### **3. Wiring Up:**

Insert micro:bit onto EASY Plug shield, connect red LED module to P10 of shield with a RJ11 cable and connect external power.



Note: Dial Voltmeter\_Switch to 3V end

#### 4.Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.

```
on start
  led enable false

forever
  repeat 2 times
    do
      digital write pin P10 to 1
      pause (ms) 1000
      digital write pin P10 to 0
      pause (ms) 1000

  repeat 2 times
    do
      while val < 1024
        do
          set val to val + 1
          analog write pin P10 to val
          pause (ms) 5

      while val > 0
        do
          set val to val - 1
          analog write pin P10 to val
          pause (ms) 5
```

The code is written in a Scratch-style block-based language. It starts with an 'on start' block containing an 'led enable' block set to 'false'. This is followed by a 'forever' loop. Inside the 'forever' loop, there are two 'repeat 2 times' blocks. The first 'repeat' block contains a 'do' loop with four blocks: 'digital write pin P10 to 1', 'pause (ms) 1000', 'digital write pin P10 to 0', and 'pause (ms) 1000'. The second 'repeat' block contains a 'do' loop with three blocks: a 'while val < 1024' loop containing 'set val to val + 1', 'analog write pin P10 to val', and 'pause (ms) 5'; followed by a 'while val > 0' loop containing 'set val to val - 1', 'analog write pin P10 to val', and 'pause (ms) 5'.

"on start" : command block only runs once to start program.

Turn off dot matrix on micro:bit

The program under the block "forever" runs cyclically.

Repeat the program in the do block twice

Set P10 to high level (1) , turn on LED

Delay in 1000ms

Set P10 to low level (0) , turn off LED

Delay in 1000ms

Repeat the program in the do block twice

When val < 1024, run the program in the do block

Set val to val+1

Set the analog value of P10 to val

Delay in 5ms

## 5. Test Result:

Wire up, dial Voltmeter\_Switch to 3V end, plug in power and dial Power\_Switch to ON end. Upload program to micro:bit, LED flashes twice and shows breathing effect twice ceaselessly.

## Project 14: RGB

### 1. Description

EASY Plug shield comes with 2812 2x2 full color RGB, we will finish three experiments with 2812 2x2 full color RGB.

In this project, we will demonstrate how to play music with passive buzzer. Easy Plug shield comes with one. Let's get started. (Passive buzzer is connected to P9 on Easy Plug shield)

## 2.What You Need:

- Micro:bit Board\*1
- EASY Plug Shield for micro bit V1.1\*1
- Micro USB Cable\*1
- 6-Slot AA Battery Holder\*1
- 1.5V AA Battery\*6

## 2812 2x2 full color RGB



2812 2x2 full color RGB module is a smart external control LED light source that integrates control circuit and lighting circuit.

Each LED has the same appearance as a 5050 LED bead, and each component is a pixel point.

The pixel point includes an intelligent digital interface data latch signal shaping and amplifying driving circuit, as well as a high-precision internal oscillator and a 12V high-voltage programmable constant current control part, which effectively ensures that the color of the pixel point light is highly uniform.

The data protocol adopts the single-line return-to-zero code

communication mode. After power-on and reset the pixel point, the S pin receives the data transmitted from the controller. And the 24-bit data are extracted by the first pixel and then sent to the data latch inside the pixel point.

LED has advantages of low voltage drive, environmental protection and energy saving, high brightness, wide scattering angle, good consistency, ultra low power, long life and so on.

### **3. Specification:**

Working voltage: DC 5V

Power: 0.1W

Light source: SMD 5050 RGB

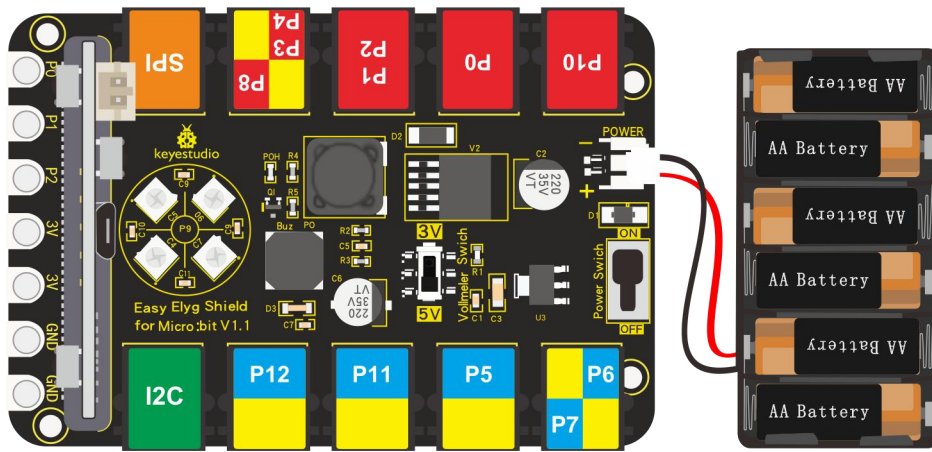
IC model: 4pcs/WS2811

Gray level: 256 levels

Beam angle: 180°

Luminous color: can be adjusted to white, red, yellow, blue, green, etc.  
by the controller

### **4. Wiring Up:**



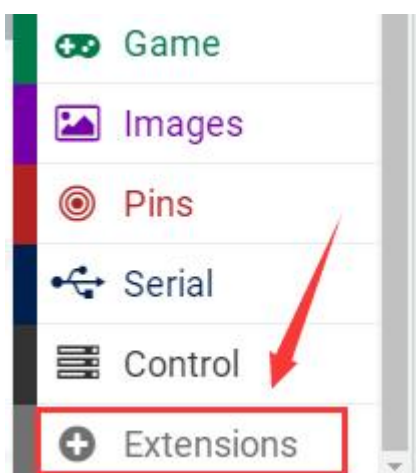
Note: Dial Voltmeter\_Switch to 3V end

## 5. Test Code:

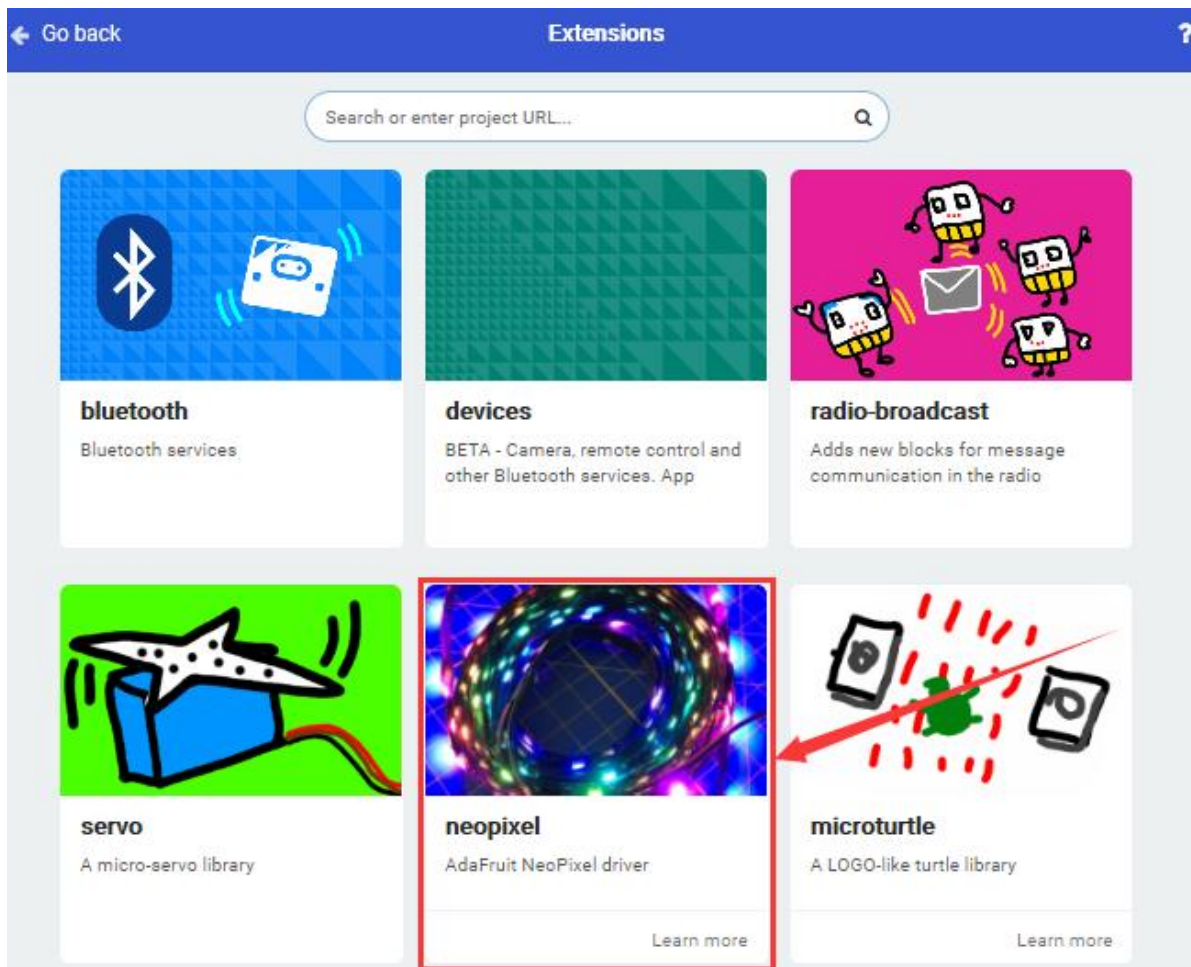
You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.

We need to set test code in library file, and add the library of "neopixe" .



Click "Extensions" → "neopixel" , click to download

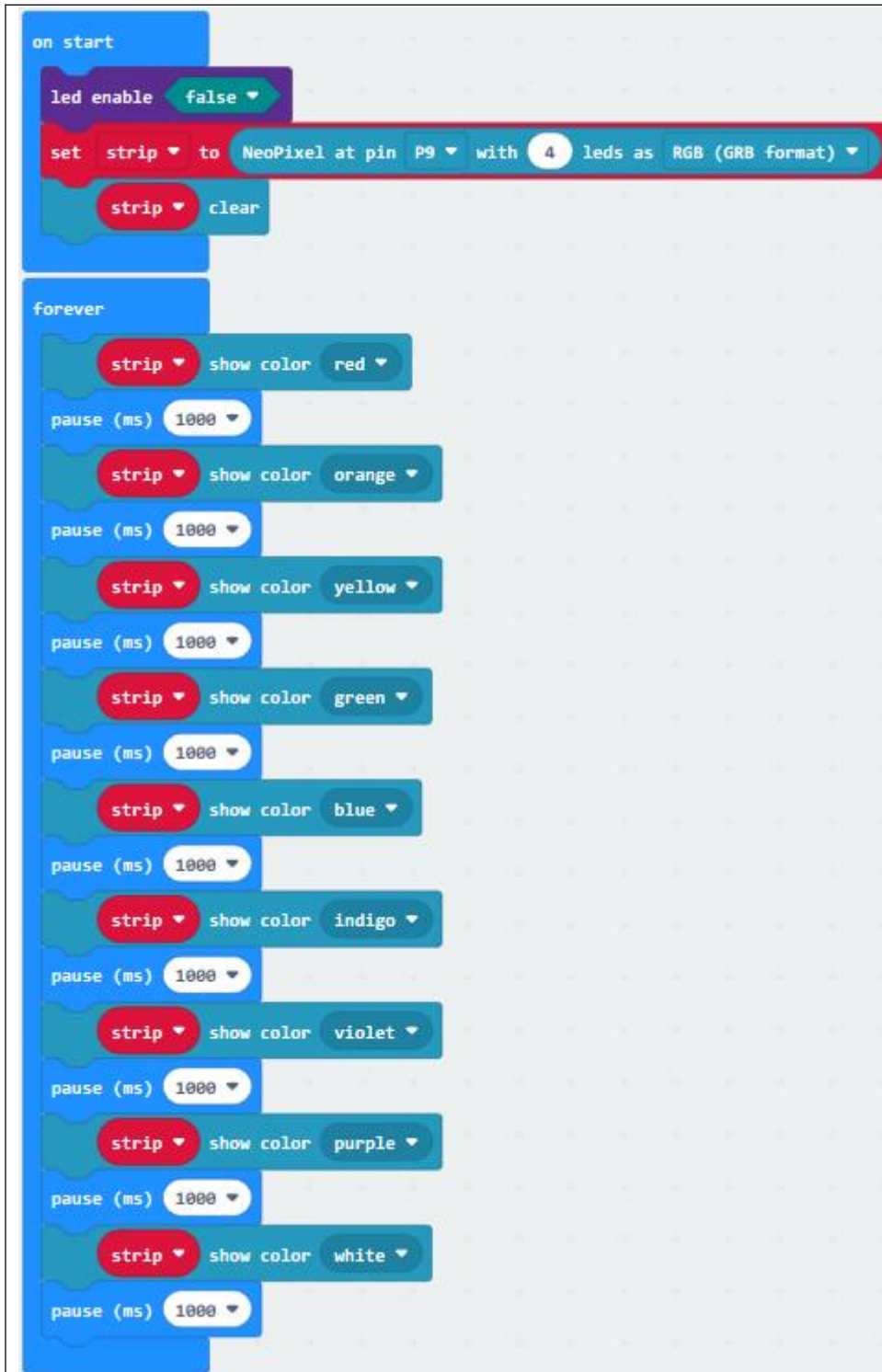


You will view library "neopixel" in the editing blocks, as shown below:

The image shows a screenshot of a block-based programming environment, likely Scratch, with the Neopixel library selected. The left sidebar contains a search bar and a list of categories: Basic, Input, Music, Led, Radio, Loops, Logic, Variables, Math, Neopixel (highlighted with a red box and a red arrow), and more. Below these are Advanced, Functions, Arrays, Text, and Game. The main workspace is titled 'Neopixel' and contains a sequence of blocks:

- set strip to NeoPixel at pin P0 with 24 leds as RGB (GRB format)
- set range to strip range from 0 with 4 leds
- strip show rainbow from 1 to 360
- strip show color red
- strip show bar graph of 0 up to 255
- strip show
- strip clear
- hue 0 saturation 0 luminosity 0
- strip shift pixels by 1
- strip rotate pixels by 1

**Code 1:**



The image shows a Scratch script for controlling an LED strip. It starts with an 'on start' block containing three steps: 'led enable' set to 'false', 'set strip' to 'NeoPixel at pin P9 with 4 leds as RGB (GRB format)', and 'strip clear'. This is followed by a 'forever' loop with 12 steps, each consisting of a 'strip show color' block and a 'pause (ms) 1000' block. The colors shown in sequence are red, orange, yellow, green, blue, indigo, violet, purple, and white.

```
on start
  led enable false
  set strip to NeoPixel at pin P9 with 4 leds as RGB (GRB format)
  strip clear

forever
  strip show color red
  pause (ms) 1000
  strip show color orange
  pause (ms) 1000
  strip show color yellow
  pause (ms) 1000
  strip show color green
  pause (ms) 1000
  strip show color blue
  pause (ms) 1000
  strip show color indigo
  pause (ms) 1000
  strip show color violet
  pause (ms) 1000
  strip show color purple
  pause (ms) 1000
  strip show color white
  pause (ms) 1000
```

“on start” : command block only runs once to start program.

Turn off dot matrix on micro:bit

Set strip to NeoPixel at pin P9 with 4 leds as RGB

Turn off 4 pcs WS2812 RGB lights

The program under the block “forever” runs cyclically.

make all RGB show red color

make all RGB show red color

Delay in 1000ms

make all RGB show orange color

Delay in 1000ms

make all RGB show yellow color

Delay in 1000ms

make all RGB show green color

Delay in 1000ms

All of RGB show blue color

Delay in 1000ms

make all RGB show indigo color

Delay in 1000ms

make all RGB show violet color

Delay in 1000ms

make all RGB show purple color

Delay in 1000ms

make all RGB show white color

Delay in 1000ms

## Code 2

```
on start
  led enable false
  set strip to NeoPixel at pin P9 with 4 leds as RGB (GRB format)

forever
  for index from 0 to 3
  do
    strip clear
    strip set pixel color at index to red
    strip show
    pause (ms) 100

  for index from 0 to 3
  do
    strip clear
    strip set pixel color at index to orange
    strip show
    pause (ms) 100

  for index from 0 to 3
  do
    strip clear
    strip set pixel color at index to yellow
    strip show
    pause (ms) 100
```

“on start” : command block only runs once to start program.

Turn off dot matrix on micro:bit

Set strip to NeoPixel at pin P9 with 4 leds as RGB

The program under the block “forever” runs cyclically.

For index from 0 to 3, execute the program under do block

Turn off 4 pcs WS2812 RGB lights

Set pixel color of 4 pcs WS2812 RGB lights to red color

Strip show

Delay in 100ms

For index from 0 to 3, execute the program under do block

Turn off 4 pcs WS2812 RGB lights

Set pixel color of 4 pcs WS2812 RGB lights to orange color

Strip show

Delay in 100ms

For index from 0 to 3, execute the program under do block

Turn off 4pcs WS2812 RGB lights

Set pixel color of 4 pcs WS2812 RGB lights to yellow color

Strip show

Delay in 100ms



The image displays two sections of Scratch code blocks. Each section consists of four identical blocks, each representing a loop iteration for a different color. The first section contains loops for green, blue, indigo, and violet. The second section contains loops for purple and white. Each loop block includes the following steps: a 'for' loop from index 0 to 3, a 'do' block containing 'strip clear', 'strip set pixel color at index to [color]', and 'strip show', followed by a 'pause (ms) 100' block.

```
for index from 0 to 3
do
strip clear
strip set pixel color at index to green
strip show
pause (ms) 100

for index from 0 to 3
do
strip clear
strip set pixel color at index to blue
strip show
pause (ms) 100

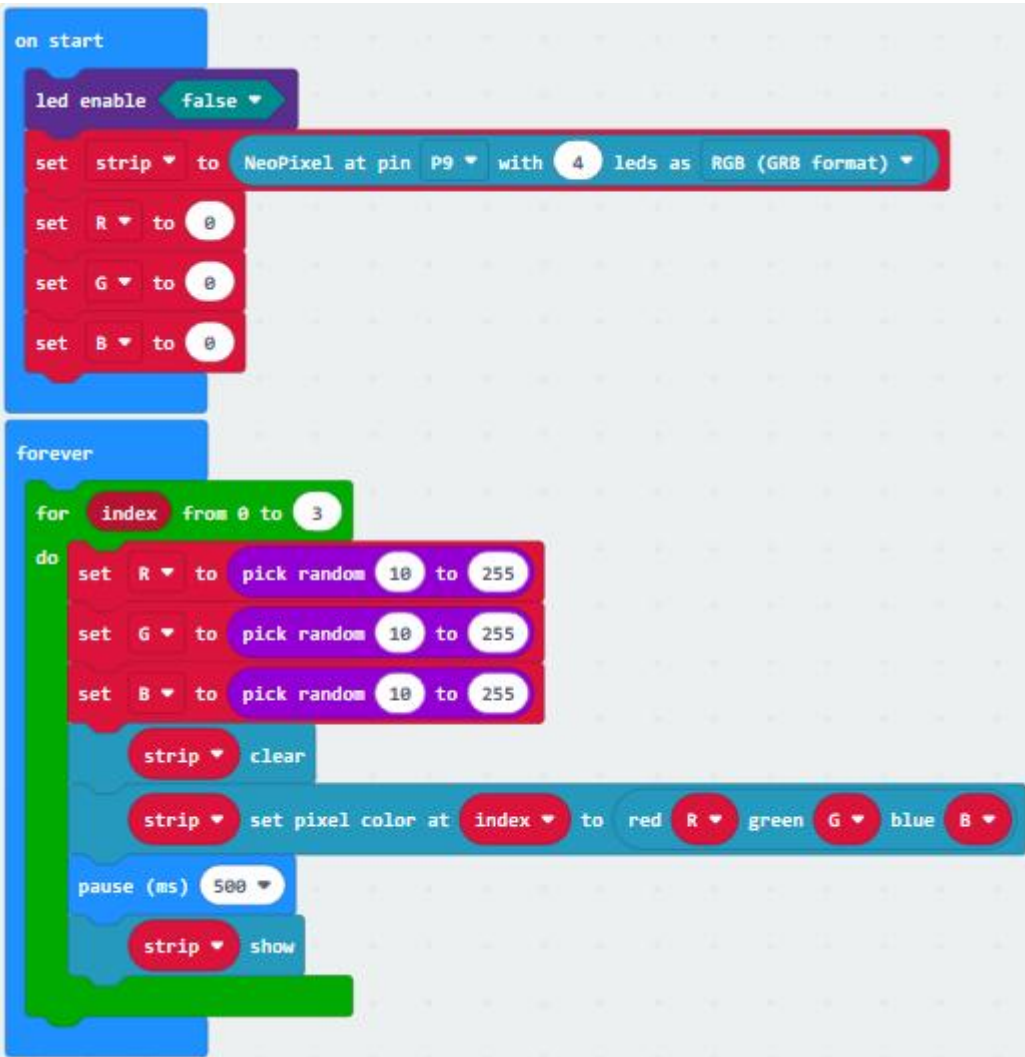
for index from 0 to 3
do
strip clear
strip set pixel color at index to indigo
strip show
pause (ms) 100

for index from 0 to 3
do
strip clear
strip set pixel color at index to violet
strip show
pause (ms) 100

for index from 0 to 3
do
strip clear
strip set pixel color at index to purple
strip show
pause (ms) 100

for index from 0 to 3
do
strip clear
strip set pixel color at index to white
strip show
pause (ms) 100
```

For index from 0 to 3, execute the program under do block  
Turn off 4 pcs WS2812 RGB lights  
Set pixel color of 4 pcs WS2812 RGB lights to green color  
Strip show  
Delay in 100ms  
For index from 0 to 3, execute the program under do block  
Turn off 4 pcs WS2812 RGB lights  
Set pixel color of 4 pcs WS2812 RGB lights to blue color  
Strip show  
Delay in 100ms  
For index from 0 to 3, execute the program under do block  
Turn off 4 pcs WS2812 RGB  
Set pixel color of 4 pcs WS2812 RGB lights to indigo color  
Strip show  
Delay in 100ms  
For index from 0 to 3, execute the program under do block  
Turn off 4 pcs WS2812 RGB lights  
Set pixel color of 4 pcs WS2812 RGB lights to violet color  
Strip show  
Delay in 100ms  
For index from 0 to 3, execute the program under do block  
Turn off 4 pcs WS2812 RGB  
Set pixel color of 4 pcs WS2812 RGB lights to violet color  
Strip show  
Strip refreshes to display  
Delay in 100ms  
For index from 0 to 3, execute the program under do block  
  
Turn off 4 pcs WS2812 RGB  
Turn off 4 pcs  
Set pixel color of 4 pcs WS2812 RGB lights to white color  
  
Strip show  
Delay in 100ms

**Code 3:**

```
on start
  led enable false
  set strip to NeoPixel at pin P9 with 4 leds as RGB (GRB format)
  set R to 0
  set G to 0
  set B to 0

forever
  for index from 0 to 3
  do
    set R to pick random 10 to 255
    set G to pick random 10 to 255
    set B to pick random 10 to 255
    strip clear
    strip set pixel color at index to red R green G blue B
    pause (ms) 500
    strip show
```

The code is written in a Scratch-like block-based language. It starts with an 'on start' block containing: 'led enable' set to 'false', 'set strip' to 'NeoPixel at pin P9 with 4 leds as RGB (GRB format)', and three 'set' blocks for 'R', 'G', and 'B' all set to '0'. A 'forever' loop contains a 'for' loop for 'index' from 0 to 3. Inside the 'for' loop, there is a 'do' block with three 'set' blocks for 'R', 'G', and 'B' each set to 'pick random 10 to 255'. This is followed by 'strip clear', 'strip set pixel color at index to red R green G blue B', 'pause (ms) 500', and 'strip show'.

“on start” : command block only runs once to start program.

Turn off micro:bit LED dot matrix

Set strip to NeoPixel at pin P9 with 4 leds as RGB

Set strip to initialization o

Set variable R to 0

Set variable G to 0

Set variable B to 0

The program under the block “forever” runs cyclically.

When value of variable index is in 0-3, execute the program in the do block

Set variable R to random number in 10~255

Set variable G to random number in 10~255

Set variable B to random number in 10~255

Turn off all RGB lights on the strip

Set pixel color of 4 pcs WS2812 RGB lights to RGB

Set 4pcs WS2812 RGB to

Delay in 500ms

strip refreshes to display

## 6. Test Result:

Wiring up, dial Voltmeter\_Switch to 3V end, plug in external power and dial Power\_Switch to ON end and download code 1 to micro:bit, WS2812RGB lights display different color.

Download code 2 to micro:bit, WS2812RGB show same color like flow light.

Download code 3 to micro:bit, each WS2812RGB shows random color like flow light.

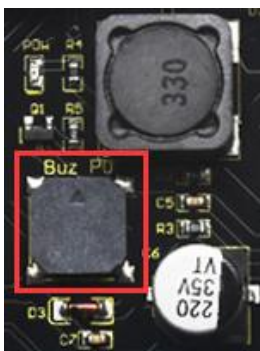
## Project 15: Play Music

### 1. Description

In this project, we will demonstrate how to play music with passive buzzer. Easy Plug shield comes with one. Let's get started. (Passive buzzer is connected to P0 on Easy Plug shield)

### 2. What You Need:

- Micro:bit Board\*1
- EASY Plug Shield for micro bit V1.1\*1
- Micro USB Cable\*1
- Slot AA Battery Holder\*1
- 1.5V AA Battery\*6



**Passive Buzzer Module:**

The buzzer includes active buzzer and passive buzzer.

The difference between them a built-in vibration source, therefore, it will make a sound when power is plugged in.

We need 2K-5K square wave to drive passive buzzer because the buzzer on EASY Plug Shield doesn't come with this kind of source.

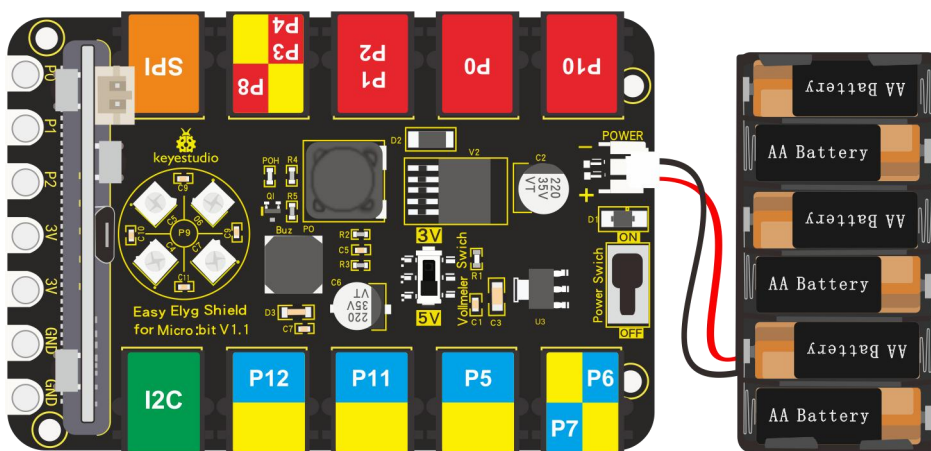
Different frequencies produce different sounds. You can use the micro:bit to compose a simple, interesting and melodic song.

### 3. Specification:

Working voltage: 3.3-5V

Interface type: Digital

### 4. Wiring Up:



**Note:** Dial Voltmeter\_Switch to 3V end.

### 5. Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.

## Code 1:

“on start” : command block only runs once to start program.

Turn off dot matrix on micro:bit

The program under the block “forever” runs cyclically.

When  $i < 80$ , run the program in the block

Set P0 to low level(0) to make passive buzzer silent

Delay in 1ms

Set P0 to high level(1) to make passive buzzer emit sound

Delay in 1ms

Set variable i to i+1

Set variable i to 0

Delay in 100ms

When  $i < 100$ , run the program in the do block

Set P0 to high level(1) to make passive buzzer emit sound

Delay in 2ms

Set P0 to low level(0) to make passive buzzer silent

Delay in 2ms

Set variable i to i+1

## Code 2:

```

on start
  led enable false
forever
  play tone High E for 2 beat
  play tone High F for 1 beat
  play tone High G for 2 beat
  play tone High F for 1 beat
  play tone High E for 1 beat
  play tone High D for 1 beat
  play tone High C for 2 beat
  play tone High D for 1 beat
  play tone High E for 2 beat
  play tone High E for 1/2 beat
  play tone High D for 1/2 beat
  play tone High D for 2 beat
  play tone High E for 2 beat
  play tone High E for 1/2 beat
  play tone High D for 1/2 beat
  play tone High D for 2 beat
  play tone High E for 2 beat
  play tone High F for 1 beat
  play tone High G for 2 beat
  play tone High F for 1 beat
  play tone High E for 1 beat
  play tone High D for 1 beat
  play tone High C for 2 beat
  play tone High D for 1 beat
  play tone High E for 1 beat
  play tone High D for 1 beat
  
```

“on start” : command block only runs once to start program.

Turn off dot matrix on micro:bit

The program under the block “forever” runs cyclically.

Play tone High E for 2 beats

Play tone High E for 1 beat

Play tone high G for 2 beats

Play tone high F for 1 beat

Play tone high E for 1 beat

Play tone high D for 1 beat

Play tone high C for 2 beats

Play tone high C for 2 beats

Play tone high D for 1 beat

Play tone high E for 2 beats

Play tone high E for 1/2 beat

Play tone high D for 1/2 beat

Play tone high D for 2 beats

Play tone high E for 2 beats

Play tone high F for 1 beat

Play tone high G for 2 beats

Play tone high F for 1 beat

Play tone high E for 1 beat

Play tone high D for 1 beat

Play tone high C for 2 beats

Play tone high D for 1 beat

Play tone high E for 1 beat

Play tone high D for 1 beat

play tone High D for 1/2 beat

play tone High C for 1/2 beat

play tone High C for 2 beat

play tone High D for 2 beat

play tone High E for 1 beat

play tone High C for 1 beat

play tone High D for 1 beat

play tone High E for 1/2 beat

play tone High F for 1/2 beat

play tone High E for 1 beat

play tone High C for 1 beat

play tone High D for 1 beat

play tone High E for 1/2 beat

play tone High F for 1/2 beat

play tone High E for 1 beat

play tone High D for 1 beat

play tone High E for 1/2 beat

play tone High F for 1/2 beat

play tone High E for 1 beat

play tone High D for 1 beat

play tone High C for 1 beat

play tone High D for 1 beat

play tone Middle G for 1 beat

play tone Low E for 1 beat

play tone High E for 2 beat

play tone High F for 1 beat

play tone High G for 2 beat

play tone High F for 1 beat

play tone High E for 1 beat

play tone High F for 1/2 beat

Play tone high D for 1/2 beat

Play tone high D for 1/2 beat

Play tone high C for 1 /2 beat

Play tone high C for 1 beat

Play tone high D for 2 beats

Play tone high E for 1 beat

Play tone high C for 1 beat

Play tone high D for 1 beat

Play tone high E for 1 /2 beat

Play tone high F for 1 /2 beat

Play tone high E for 1 beat

Play tone high C for 1 beat

Play tone high D for 1 beat

Play tone high E for 1/2 beat

Play tone high F for 1 /2 beat

Play tone high E for 1 beat

Play tone high D for 1 beat

Play tone high C for 1 beat

Play tone high D for 1 beat

Play tone high G for 1 beat

Play tone high E for 1 beat

Play tone high E for 2 beats

Play tone high F for 1 beat

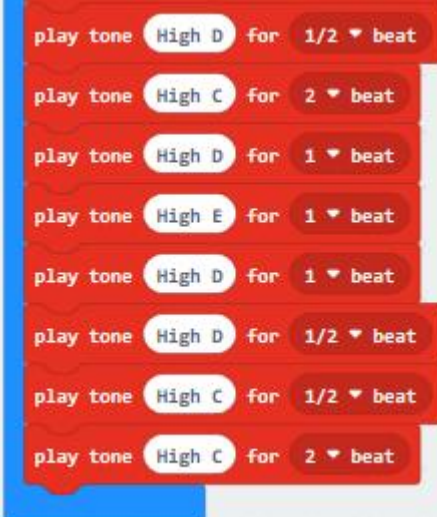
Play tone high G for 2 beats

Play tone high F for 1 beat

Play tone high E for 1 beat


Play tone high F for 1 /2 beat

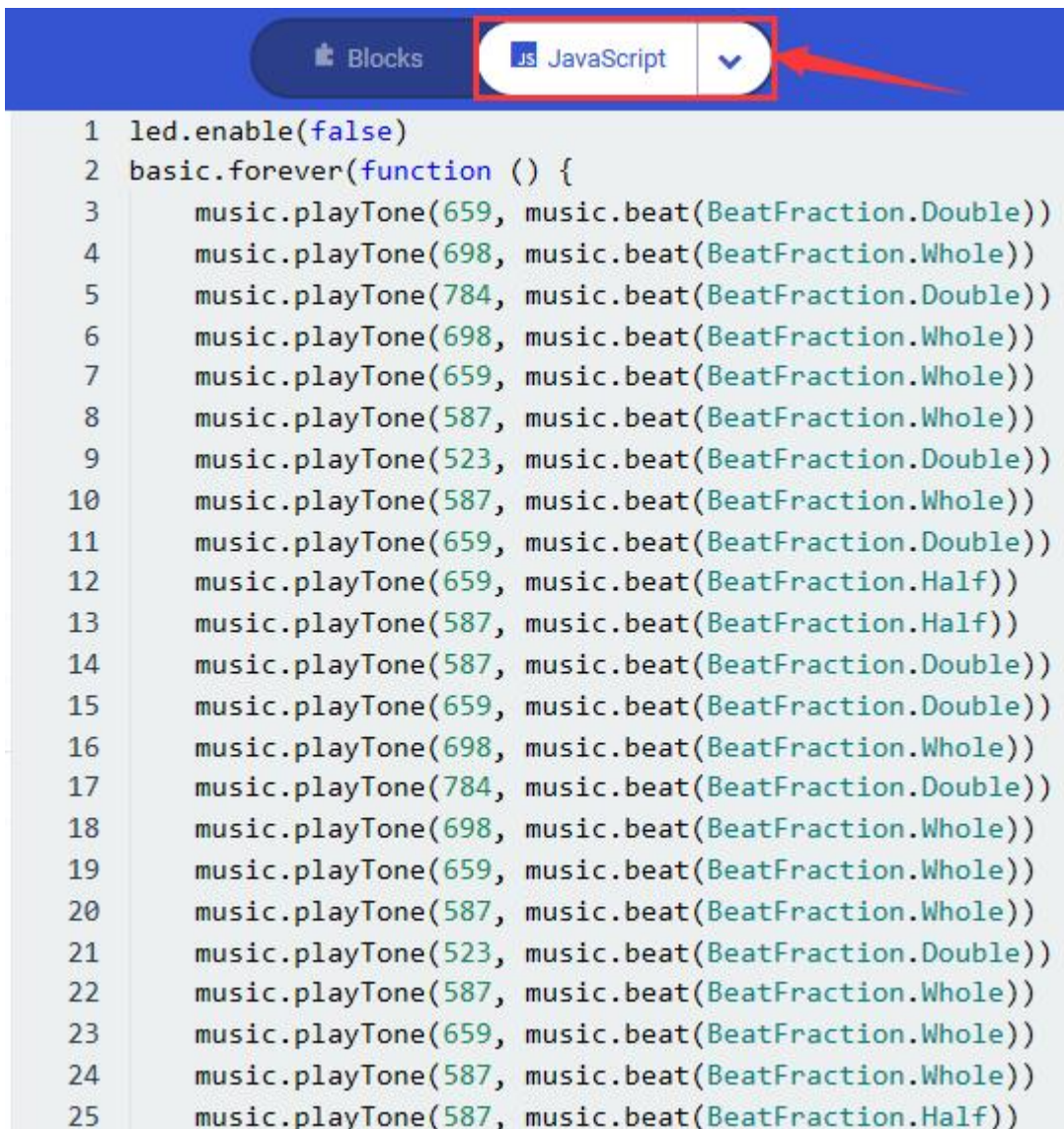
Play tone high E for 2 beats



The image shows a vertical stack of eight red Scratch 'play tone' blocks. Each block contains a tone name in a white oval, the word 'for', a duration value in a white oval, and the word 'beat'. The blocks are: High D for 1/2 beat, High C for 2 beat, High D for 1 beat, High E for 1 beat, High D for 1 beat, High D for 1/2 beat, High C for 1/2 beat, and High C for 2 beat.

Tone	Duration
High D	1/2 beat
High C	2 beats
High D	1 beat
High E	1 beat
High D	1 beat
High D	1/2 beat
High C	1/2 beat
High C	2 beats

Note: Click  to switch into JavaScript code, each frequency and beat of tone is shown below:



```

1 led.enable(false)
2 basic.forever(function () {
3     music.playTone(659, music.beat(BeatFraction.Double))
4     music.playTone(698, music.beat(BeatFraction.Whole))
5     music.playTone(784, music.beat(BeatFraction.Double))
6     music.playTone(698, music.beat(BeatFraction.Whole))
7     music.playTone(659, music.beat(BeatFraction.Whole))
8     music.playTone(587, music.beat(BeatFraction.Whole))
9     music.playTone(523, music.beat(BeatFraction.Double))
10    music.playTone(587, music.beat(BeatFraction.Whole))
11    music.playTone(659, music.beat(BeatFraction.Double))
12    music.playTone(659, music.beat(BeatFraction.Half))
13    music.playTone(587, music.beat(BeatFraction.Half))
14    music.playTone(587, music.beat(BeatFraction.Double))
15    music.playTone(659, music.beat(BeatFraction.Double))
16    music.playTone(698, music.beat(BeatFraction.Whole))
17    music.playTone(784, music.beat(BeatFraction.Double))
18    music.playTone(698, music.beat(BeatFraction.Whole))
19    music.playTone(659, music.beat(BeatFraction.Whole))
20    music.playTone(587, music.beat(BeatFraction.Whole))
21    music.playTone(523, music.beat(BeatFraction.Double))
22    music.playTone(587, music.beat(BeatFraction.Whole))
23    music.playTone(659, music.beat(BeatFraction.Whole))
24    music.playTone(587, music.beat(BeatFraction.Whole))
25    music.playTone(587, music.beat(BeatFraction.Half))

```

## 6. Test Result:

Wiring up, dial Voltmeter\_Switch to 3V end, plug in external power and dial Power\_Switch to ON end and upload code 1 to micro:bit, you will hear the buzzer emit two kind of sounds; if download code 2 to micro:bit, the song "Ode-to- Joy" will be played.

## Project 16: Knock Sensor

### 1. Description

Sensor can detect the data, sense the signal and control the devices. We are familiar with light sensor, temperature, humidity and sound sensors. A great deal of experiments could be finished with these sensors and modules.

We will control the LED by knock sensor in the experiment.

## 2. What You Need:

- Micro:bit Board\*1
- EASY Plug Shield for micro bit V1.1\*1
- Micro USB Cable\*1
- EASY Plug Knock Sensor\*1
- EASY Plug Red LED Module\*1
- RJ11 Cable\*2
- 6-Slot AA Battery Holder\*1
- 1.5V AA Battery\*6



### **EASY Plug Knock Sensor:**

The knock sensor is mainly composed of SW-280 vibration switch, which is an inductive proximity switch.

It is an electronic switch that transmits

the sensing result to the circuit device and induces the circuit to start working when the vibration force is induced.

The module comes with a positioning hole for you to fix it to other devices.

You can make full use of it with creative thinking, like electronic drum, and so on.

This module should be used together with EASY plug control board.

### **3. Specification:**

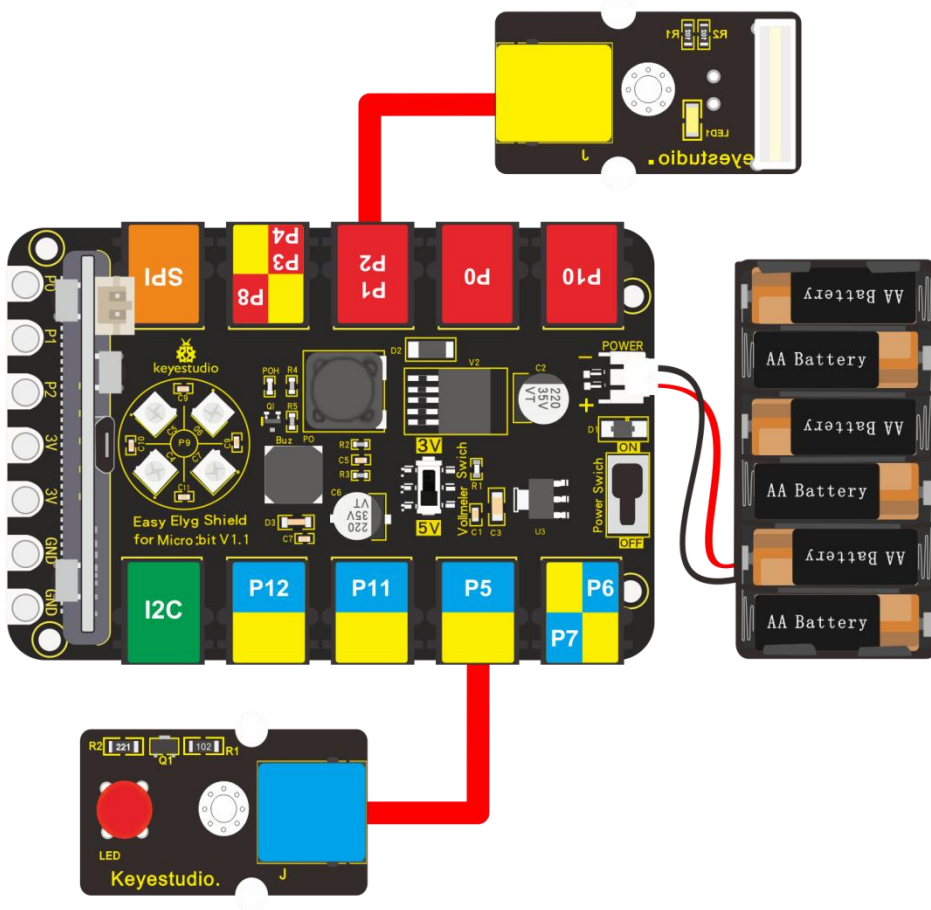
Interface: Easy plug

Working voltage: 5V

Sensor type: Digital

### **4. Wiring Up:**

Insert micro:bit onto EASY Plug shield, connect knock sensor and red LED module to P1 and P5 port of shield with two RJ11 cables.

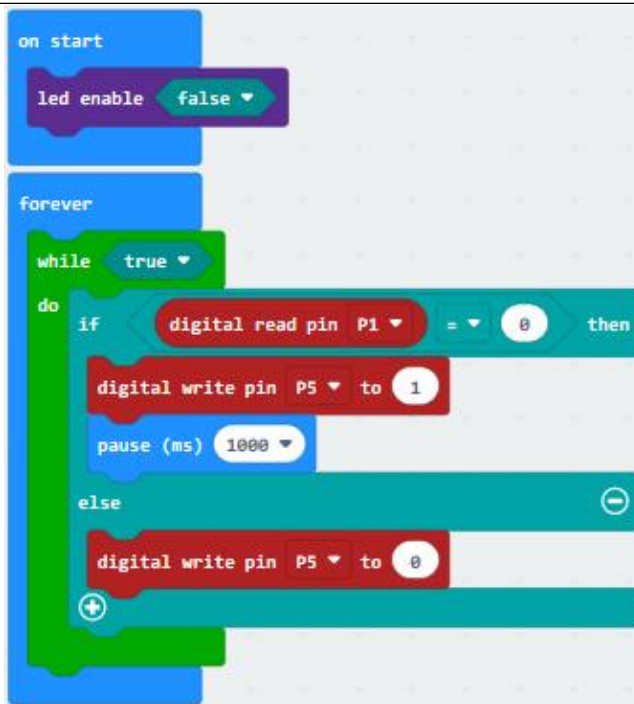


Note: Dial Voltmeter\_Switch to 3V end.

## 5.Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.



"on start" : command block only runs once to start program.

Turn off dot matrix on micro:bit

The program under the block "forever" runs cyclically.

If it is true, execute the program under do block

If the digital signal of P1 is 0, module is knocked, execute the program under then block

Set P5 to high level(1), turn on LED

Delay in 1000ms

If the digital signal of P1 is 1, module is not knocked, execute the program under else block

Set P5 to low level(0), turn off LED

## 6.Test Result:

Wiring up, dial Voltmeter\_Switch to 3V end, plug in external power

and dial Power\_Switch to ON end and upload code to micro:bit.

When the module is knocked, LED turns on for 1s; if not, LED is off.

## **Project 17: PIR Motion Sensor**

### **1. Description**

In this experiment, we connect EASY Plug PIR motion sensor to micro:bit and detect the object moving , the detected digital signals will be displayed on serial monitor.

### **2. What You Need:**

- Micro:bit Board\*1
- EASY Plug Shield for micro bit V1.1\*1
- Micro USB Cable\*1
- EASY Plug PIR Motion Sensor\*1
- EASY Plug Red LED Module\*1
- RJ11 Cable\*2
- 6-Slot AA Battery Holder\*1

- 1.5V AA Battery\*6

### **EASY Plug PIR Motion Sensor:**



PIR stands for Pyroelectric Infrared (many times, they are also referred as Passive Infrared sensors). This is because their principle of operation is based on the detection of infrared energy emitted by a moving body.

The PIR sensor can detect infrared signals from a moving person or moving animal, outputting switching signals.

One important thing to mention is that when motion is detected, the output will stay high for 2.3 to 3 seconds after the motion stops.

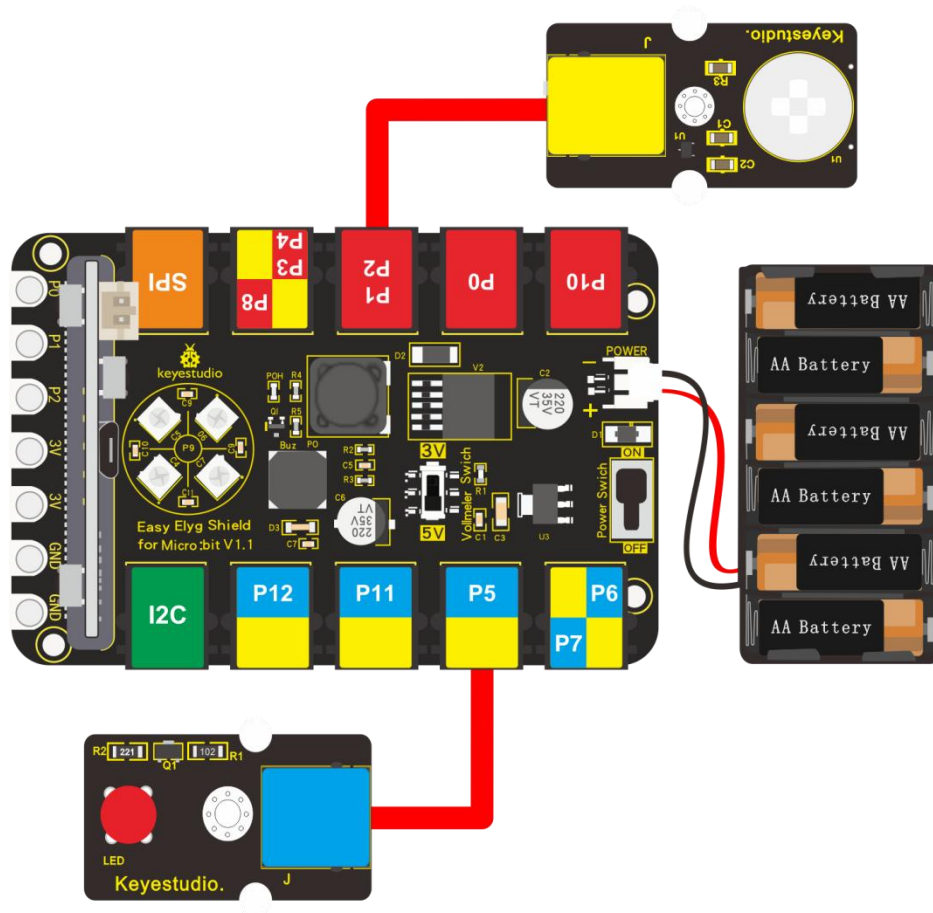
Regarding the power supply, it can work with voltages of both 3.3V and 5V. The device has a detection range of 7 meters and a detection angle of 100°.

### **3. Specification:**

- Connector: Easy plug
- Input Voltage: 3.3 ~ 5V, Maximum 6V
- Working Current: 15uA
- Working Temperature: -20 ~ 85°C
- Output Voltage: High 3V, Low 0V
- Output Delay Time (High Level): about 2.3 to 3 Seconds
- Detection angle: 100°
- Detection distance: 7 meters
- Output Indicator LED (When output HIGH, it will be ON)
- Pin limit current: 100mA

#### **4. Wiring Up:**

Insert micro:bit onto EASY Plug shield, connect PIR motion sensor and red LED to P1 and P5 port of shield.

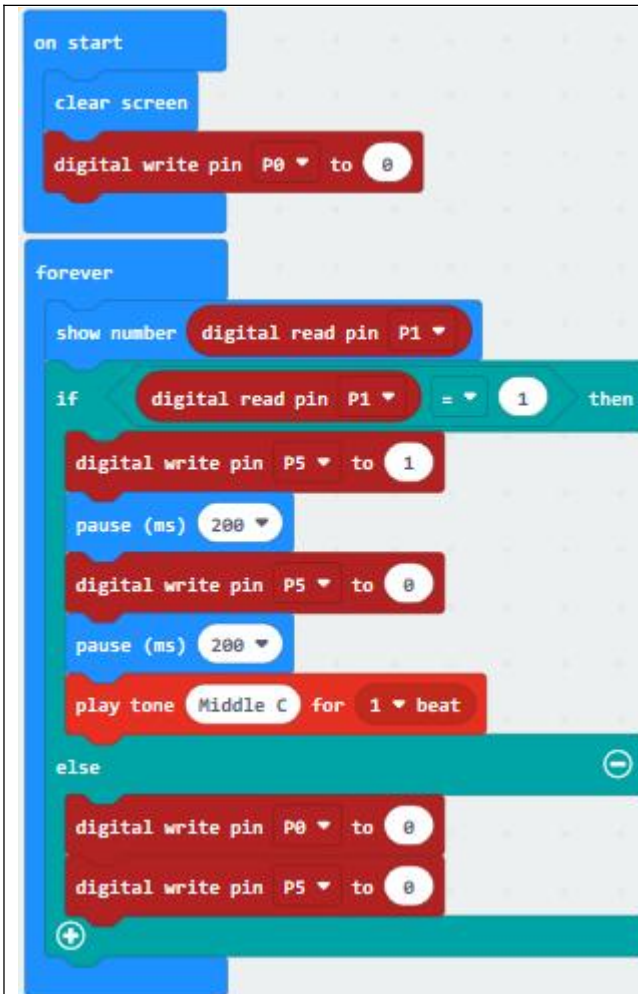


Note: Dial Voltmeter\_Switch to 3V end.

## 5.Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.



“on start” : command block only runs once to start program.

Clear LED dot matrix

Set P0 to low level(0), turn off passive buzzer

The program under the block “forever” runs cyclically.

Micro:bit shows the digital signal read by PIR motion sensor.

If digital signal read by P1, detect people’s motion, execute the program under then block.

Set P5 to high level(1), LED turns on

Delay in 200ms

Set P5 to low level(0), LED is off

Delay in 200ms

Play tone C for 1 beat, passive buzzer emits sound.

When digital signal read by P1 is 0, nobody is detected, execute the program under else block

Set P0 to low level(0), passive buzzer doesn’t sound

Set P5 to low level(0), LED is off

## 6. Test Result:

Wiring up, dial Voltmeter\_Switch to 3V end, plug in external power and dial Power\_Switch to ON end and upload code to micro:bit.

When PIR motion sensor detects the movement of PIR motion sensor. LED flashes, passive buzzer emits sound and micro:bit shows high level(1); by contrast, micro:bit shows low level(0), LED is off and buzzer doesn't emit sound.

## Project 18: Capacitive Touch

### 1. Description

Are you tired of mechanical buttons? Try the capacitive touch module. In this lesson, we will replace button switch with capacitive touch module and demonstrate how to control passive buzzer with capacitive buzzer.

### 2. What You Need:

- Micro:bit Board\*1

- EASY Plug Shield for micro bit V1.1\*1
- Micro USB Cable\*1
- EASY Plug Capacitive Touch Module \*1
- RJ11 Cable\*1
- 6-Slot AA Battery Holder\*1
- 1.5V AA Battery\*6

### **EASY Plug Capacitive Touch Module:**

Based on touch detection chip, the touch area of capacitive touch module is applied widely on a plenty of touch sensors. This touch sensor features the one-key button function and adopts the most popular capacitive sensing technology. In addition, it can "feel" people and metal touch and feedback a high/low voltage level. Even isolated by some cloth and paper, it can still feel the touch. The sensitivity will decrease as the isolation getting thick.

This sensor can tackle the traditional button problems, with the characteristics of low consumption and wide working voltage.

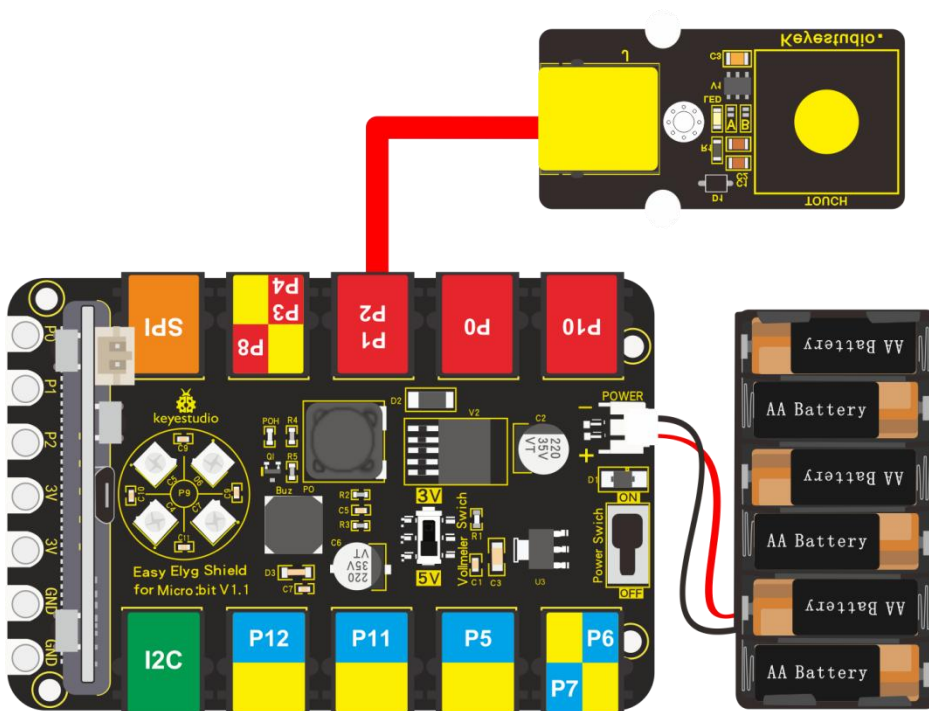
### **3. Specification:**

- Jog type: the initial state is low, high touch, do not touch is low (similar touch of a button feature);

- Low power consumption;
- Power supply for 3.3 ~ 5V DC;
- Smooth touch surface

#### 4. Wiring Up:

Insert micro:bit onto EASY Plug shield, connect capacitive touch module to P1 port of shield.

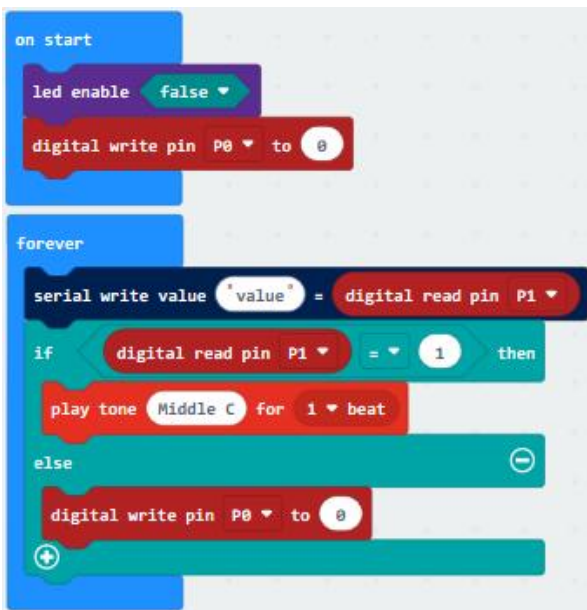


Note: Dial Voltmeter\_Switch to 3V end

#### 5. Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.



"on start" : command block only runs once to start program.

Turn off dot matrix on micro:bit

Set P0 to low level(0), turn off passive buzzer

The program under the block "forever" runs cyclically.

Serial writes the digital signals read by sensor

If the digital signal read by P1 is 1, touch sensor, execute the program under then block

Play tone C for 1 beat, make passive buzzer emit sound

If the digital signal read by P1 is 0, sensor is not touched, execute the program under else block

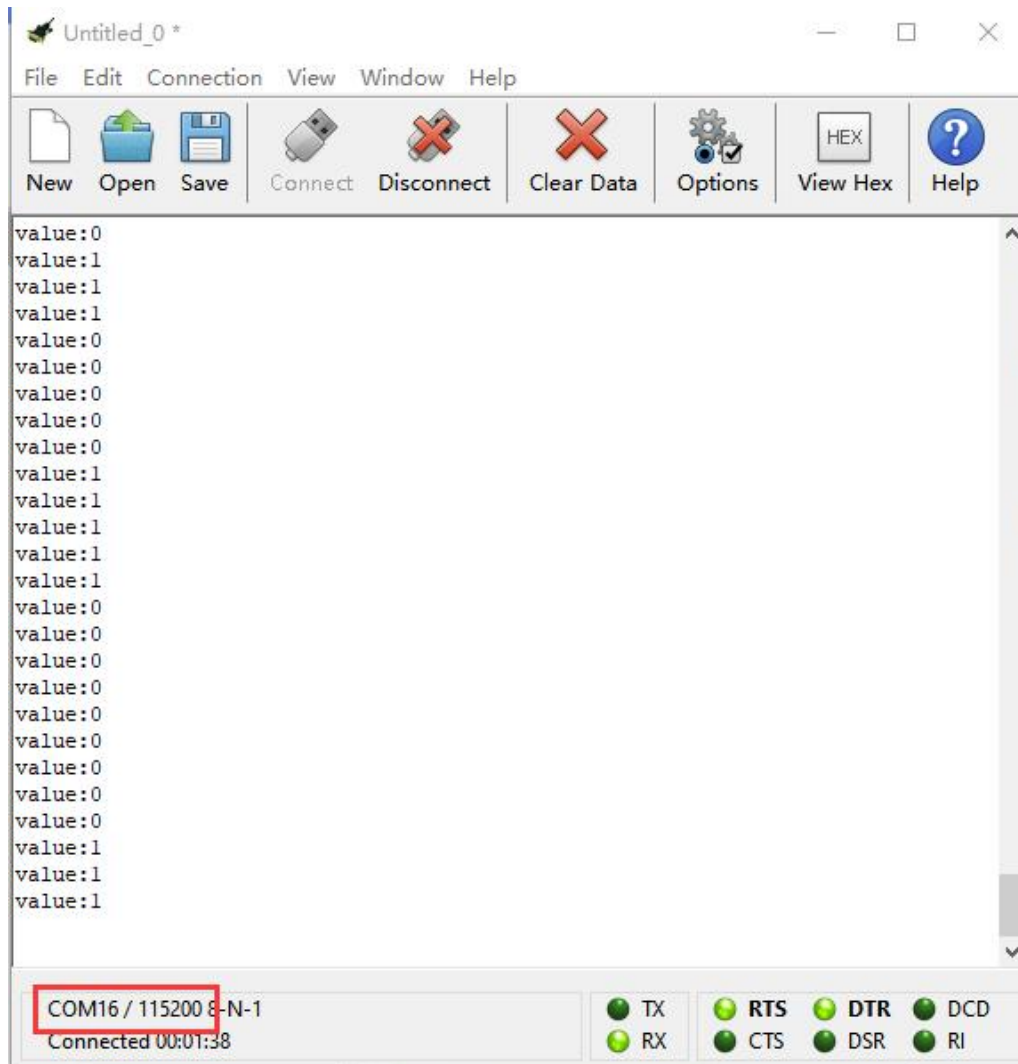
Set P0 to low level (0), turn off passive

## 6.Test Result:

Wiring up, dial Voltmeter\_Switch to 3V end, plug in external power and dial Power\_Switch to ON end and upload code to micro:bit.

Open CoolTerm, click Options and select SerialPort, set COM port and baud rate, set baud rate to 115200. Tap OK and Connect.

When touch area is touched, CoolTerm monitor shows 1 and passive buzzer emits sound; otherwise, 0 is shown on CoolTerm monitor and passive buzzer doesn't emit sound, as shown below:



## Project 19: Obstacle Avoidance

### 1. Description

Have you ever seen a smart car avoid the obstacle itself?

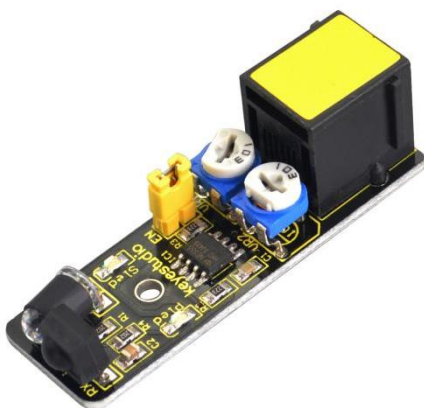
Do you know why? Let me explain to you in this chapter.

We will use a passive buzzer, red LED module and an obstacle avoidance sensor in the experiment

## 2. What You Need:

- Micro:bit Board\*1
- EASY Plug Shield for micro bit V1.1\*1
- Micro USB Cable\*1
- EASY Plug Obstacle Avoidance Sensor\*1
- EASY Plug Red LED Module\*1
- RJ11 Cable\*2
- 6-Slot AA Battery Holder\*1
- 1.5V AA Battery\*6

### Obstacle Avoidance Sensor:



Infrared obstacle avoidance sensor is equipped with distance adjustment function and is especially designed for wheeled robots.

This sensor has strong adaptability to ambient light and is of high precision. It has a pair of infrared transmitting and receiving tube.

When infrared ray launched by the transmitting tube encounters an obstacle (its reflector), the infrared ray is reflected to the receiving tube, after a comparator circuit processing, the indicator will light up.

You can adjust the detection distance by rotating the potentiometer knob, the effective distance range of 2~40cm.

They can be widely used in robot obstacle avoidance, avoidance car, line count, and black and white line tracking and many other occasions.

### **3. Specification:**

Working voltage: DC 3.3V-5V

Working current:  $\geq 20\text{mA}$

Working temperature:  $-10^{\circ}\text{C}$  to  $+50^{\circ}\text{C}$

IO interface: EASY Plug(G(-)/V(+)/out/EN)

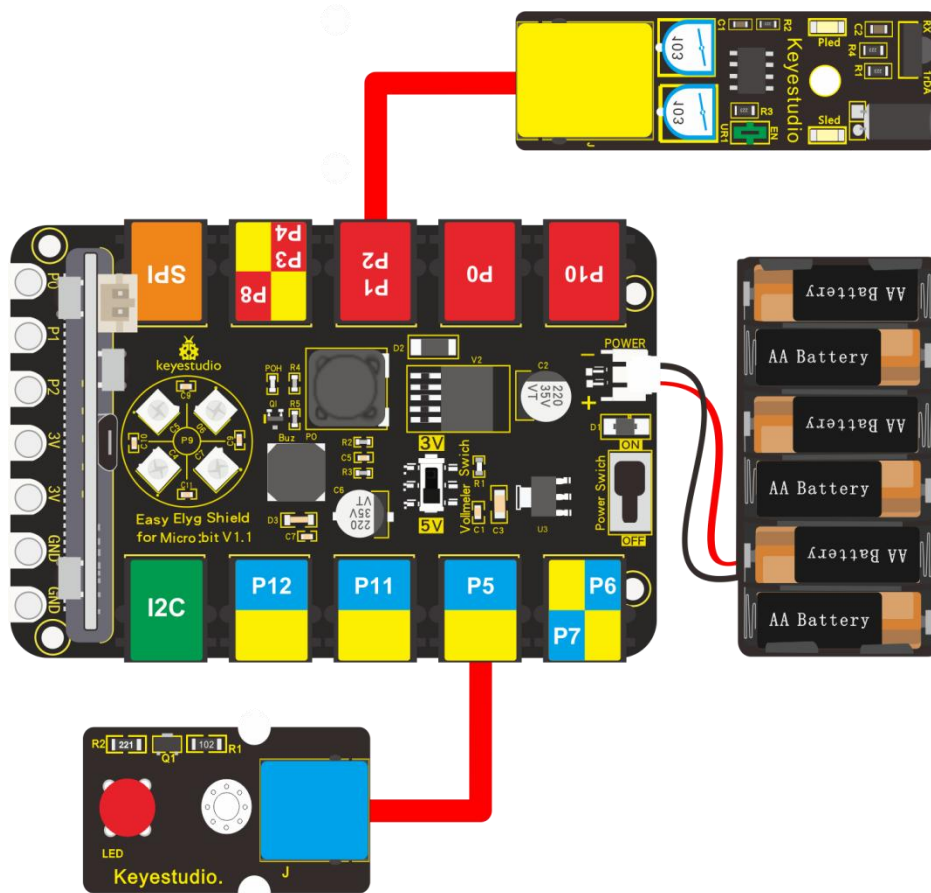
Signal output: TTL voltage

Detection distance: 2-40cm

### **4. Wiring Up:**

Insert micro:bit onto slot of EASY Plug shield, connect obstacle

avoidance sensor and red LED module to P1 and P5 of Easy Plug shield.

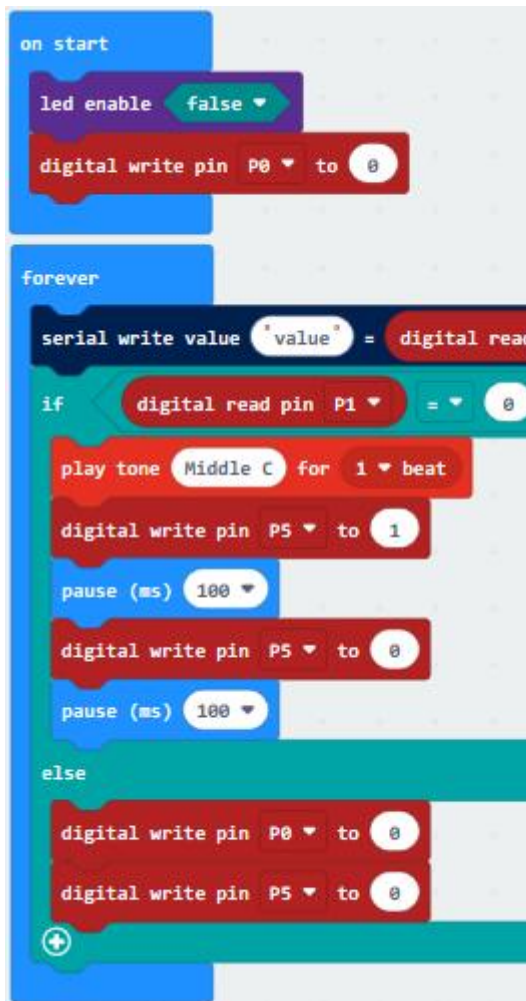


Note: Dial Voltmeter\_Switch to 3V end.

## 5.Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.



"on start": command block only runs once to start program.

Turn off dot matrix on micro:bit

Set P0 to low level(0), turn off passive buzzer

The program under the block "forever" runs cyclically.

Serial writes the digital signal read by obstacle avoidance sensor

If the digital signal=0, detect the obstacle, execute the program under then block

Play tone C for 1 beat to make passive buzzer emit sound

Set P5 to high level(1), turn on LED

delay in 100ms

Set P5 to low level(0), turn off LED

delay in 100ms

If the digital signal=1, detect no obstacle, execute the program under else block

Set P0 to low level(0), passive buzzer doesn't emit sound

Set P5 to low level(0), turn off LED

## 6.Test Result:

Wiring up, dial Voltmeter\_Switch to 3V end, plug in external power and dial Power\_Switch to ON end and upload code to micro:bit.

Open CoolTerm, click Options and select SerialPort, set COM port and baud rate, set baud rate to 115200. Tap OK and Connect.



In this chapter, we will illustrate the principle and application of servo.

## 2. What You Need:

- Micro:bit Board\*1
- EASY Plug Shield for micro bit V1.1\*1
- Micro USB Cable\*1
- EASY Plug Servo Module\*1
- Keystudio Servo\*1
- RJ11 Cable\*1
- 6-Slot AA Battery Holder\*1
- 1.5V AA Battery\*6

### EASY Plug Servo Module



If you want to use the Micro Servo and EASY PLUG control board to do some experiments, you need to use the EASY Plug Servo extension module.

The EASY Plug Servo module is extended into Registered Jack, so you can connect it to EASY PLUG control board using only a RJ11 cable.

The Servo module also comes with 3pins of 2.54mm pin pitch, fully compatible with servo pins.

## Keystudio Servo



Servo motor comes with many specifications. But all of them have three connection wires, distinguished by brown, red, orange colors. Brown one is for ground, red one for power positive, orange one for signal line.

Included with your Micro Servo you will find a variety of white motor mounts that connect to

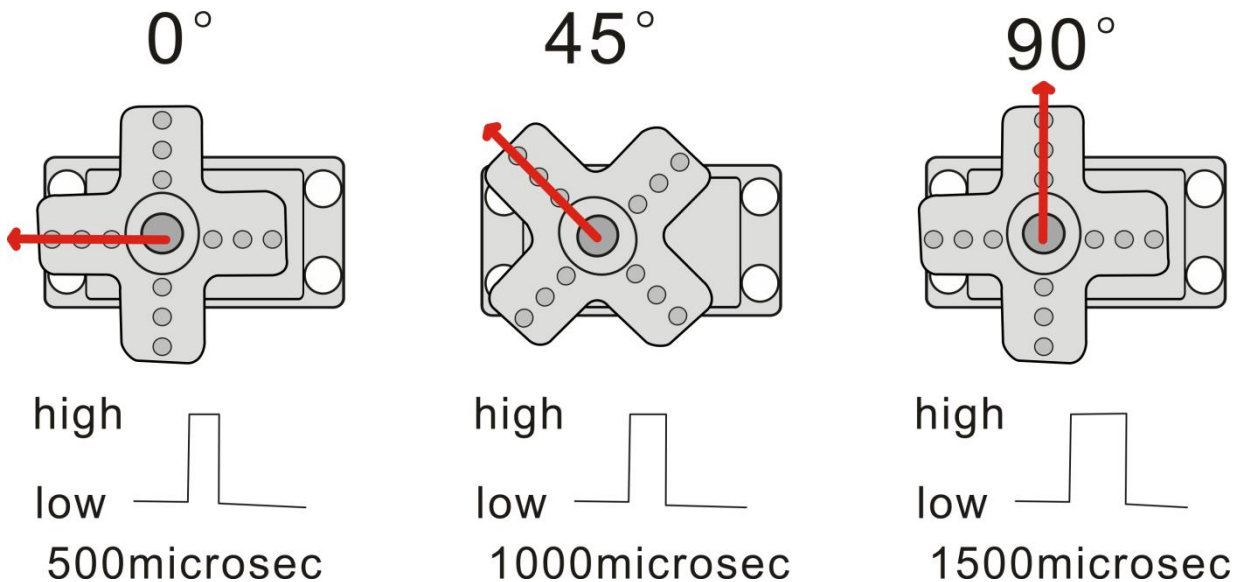
the shaft of your servo.

You may choose to attach any mount you wish for the circuit. It will serve as a visual aid, making it easier to see the servo spin.

The rotation angle of servo is controlled by regulating the duty cycle of the PWM(Pulse-Width Modulation) signal.

The standard cycle of the PWM signal is fixed at 20ms (50 Hz), and the pulse width is distributed between 1ms-2ms.

The pulse width corresponds to the rotation angle (  $0^\circ \sim 90^\circ$  ) of servo.



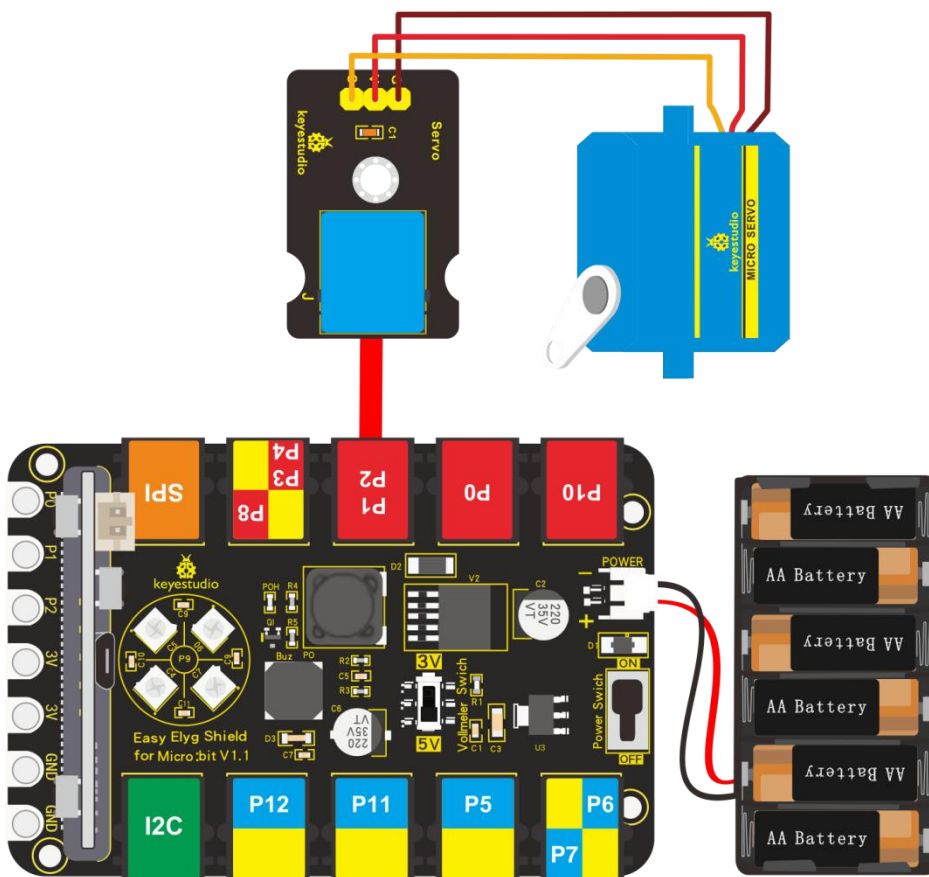
### 3. Specification:

- Operating voltage: DC 4.8V~6V
- Angle range: about 180°(in 500→2500μsec)
- Pulswidth range: 500→2500μsec
- No-load speed:  $0.12 \pm 0.01$  sec/60 (DC 4.8V) ;  $0.1 \pm 0.01$  sec/60 (DC 6V)
- No-load current:  $200 \pm 20$ mA (DC 4.8V) ;  $220 \pm 20$ mA (DC 6V)
- Stop torque:  $1.3 \pm 0.01$ kg/cm (DC 4.8V) ;  $1.5 \pm 0.1$ kg/cm (DC 6V)
- Stop current:  $\leq 850$ mA (DC 4.8V) ;  $\leq 1000$ mA (DC 6V)
- Standby current:  $3 \pm 1$ mA (DC 4.8V) ;  $4 \pm 1$ mA (DC 6V)
- Operation temperature:  $-10^{\circ}\text{C} \sim 50^{\circ}\text{C}$
- Save temperature:  $-20^{\circ}\text{C} \sim 60^{\circ}\text{C}$
- Motor wire length:  $250 \pm 5$  mm
- Dimensions: 22.9mm\*12.2mm\*30mm

- Weight:  $9 \pm 1$  g (without servo mounts)

#### 4.Wiring Up:

Insert micro:bit into EASY Plug shield, link servo with servo module. Brown line is connected to G, red line is linked with V and orange line is connected to S. Connect servo module to P1 port of shield.



Dial Voltmeter\_Switch to 5V end.

#### 5.Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.



"on start" : command block only runs once to start program.  
 Turn off dot matrix on micro:bit  
 Rotate P1 to 0°  
 Delay in 100ms  
 The program under the block "forever" runs cyclically.  
 Rotate P1 to 0°  
 Delay in 1000ms  
 Rotate P1 to 180°  
 Delay in 1000ms

## 6.Test Result:

Wiring up, dial Voltmeter\_Switch to 5V end, plug in external power and dial Power\_Switch to ON end and upload code to micro:bit.

You will view the servo rotate from 0 ° to 180 °

## Project 21: Fan Module

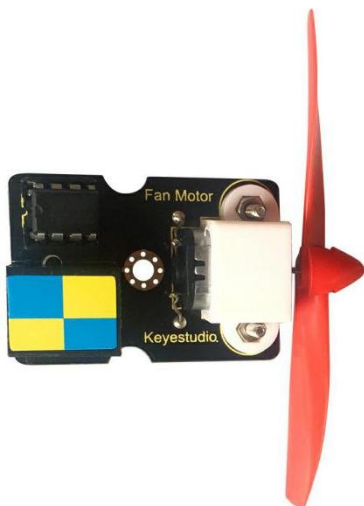
## 1. Description:

We will make fan module turn clockwise, anticlockwise and stop.

## 2. What You Need:

- Micro:bit Board\*1
- EASY Plug Shield for micro bit V1.1\*1
- Micro USB Cable\*1
- EASY PlugL9110 Fan Module \* 1
- RJ11 Cable\*1
- 6-Slot AA Battery Holder\*1
- 1.5V AA Battery\*6

### EASY Plug Fan Module



EASY PlugL9110 fan module cooperates GND, VCC, INA and INB pin. Pin INA and INB can control the speed and direction of fan.

This fan control module adopts L9110 motor control chip. It can control the rotation direction of the motor, hence the fan. The module is designed with mounting hole, compatible with servo motor control.

The module is of high efficiency, with the high quality fan, it can easily blow out flame of a light in 20cm distance.

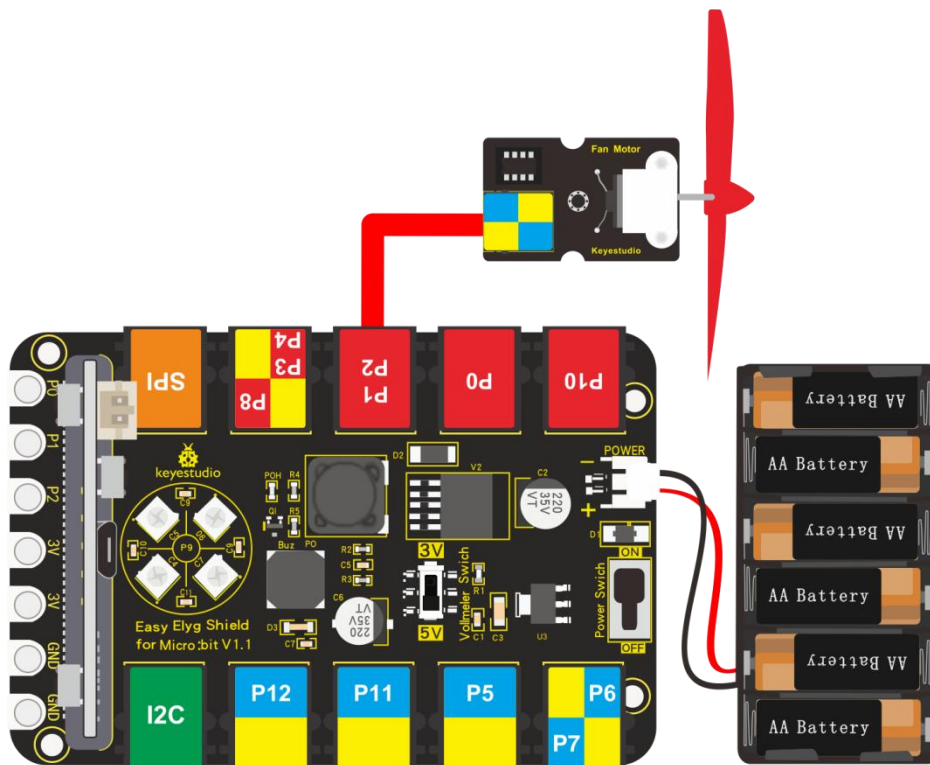
It is widely applied in air propeller, cooling system and spinning frame.

### 3. Specification:

- Fan blade diameter: 75mm
- Interface type: dual analog I/O port interface
- Working voltage: 5V

### 4.Wiring Up:

Insert micro:bit onto EASY Plug shield, connect fan module to P1-P2 port of shield.



Note: dial Voltmeter\_Switch to 5V end.

### 5.Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.



```
on start
  led enable false

function forward speed1
  digital write pin P1 to 0
  analog write pin P2 to speed1

function reversal speed2
  digital write pin P2 to 0
  analog write pin P1 to speed2
```

"on start" : command block only runs once to start program.

Turn off dot matrix on micro:bit

Set subfunction forward speed1

Set P1 to low level(0)

Set the analog value of P2 to speed1

Set subfunction reversal speed2

Set P2 to low level(0)

Set analog value of P1 to speed2



Subfunction stop  
 Set P1 to low level(0)  
 Set P2 to low level(0)  
 The program under the block "forever" runs cyclically.  
 Call subfunction forward, set speed to 800  
 Delay in 1000ms  
 Call subfunction stop, stop  
 Delay in 2000ms  
 Call subfunction reversal, set speed to 500  
 Delay in 1000ms  
 Set subfunction stop, stop  
 Call the sub  
 Delay in 2000ms

## 6.Test Result:

Wiring up, dial Voltmeter\_Switch to 5V end, plug in external power and dial Power\_Switch to ON end and upload code to micro:bit.

You will view that fan module rotate clockwise for 1s, and stop for 2s, then anticlockwise for 1s, and stop for 2s.

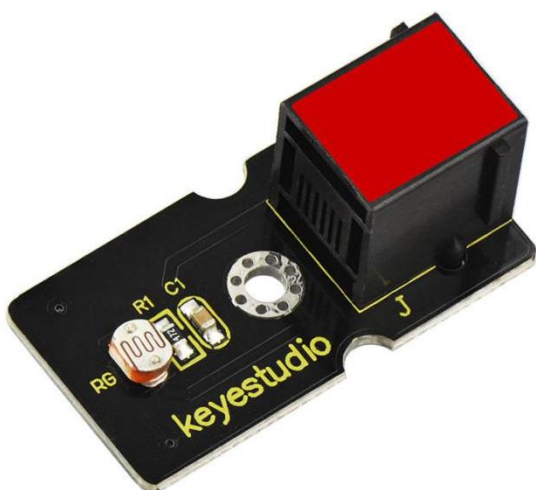
## Project 22: Light Brightness

### 1. Description

Sensors are everywhere in our life, street lights will turn on at night but off at day, why? In fact, it is because the photorsensitive element that can sense the ambient light brightness. In this program, we will control the light brightness by photoresistor module.

### 2.What You Need:

- Micro:bit Board\*1
- EASY Plug Shield for micro bit V1.1\*1
- Micro USB Cable\*1
- EASY Plug Photoresistor\*1
- EASY Plug Red LED Module\*1
- RJ11 Cable\*2
- 6-Slot AA Battery Holder\*1
- 1.5V AA Battery\*6



### EASY Plug Photoresistor:

A photoresistor or light-dependent resistor (LDR) or photocell is a light-controlled variable resistor. The

principal is very simple. The resistance of photoresistor varies with incident light intensity. If the incident light intensity is high, the resistance decreases; if the light intensity is low, the resistance increases.

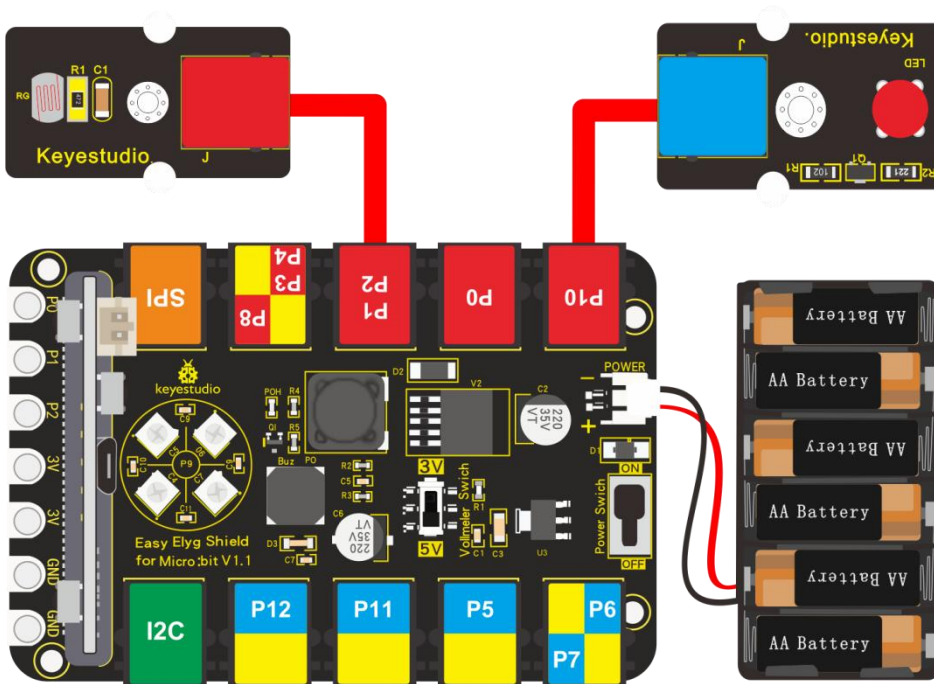
keyestudio EASY Plug photocell sensor is a semiconductor, integrated with a photoresistor, easy to use and very convenient for wiring.

It has features of high sensitivity, quick response, spectral characteristic and R-value consistence.

It can be applied in light-sensitive detector circuits, intelligent switch design and light- and dark-activated switching circuits.

#### **4.Wiring Up:**

Insert micro:bit onto EASY Plug shield, connect photoresistor and red LED module to P1 and P10 port of shield with RJ11 cables.



Note: Dial Voltmeter\_Switch to 3V end.

## 5. Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.



“on start” : command block only runs once to start program.

Turn off dot matrix on micro:bit

The program under the block “forever” runs cyclically.

Serial writes the analog signals detected by photoresistor

Delay in 100ms

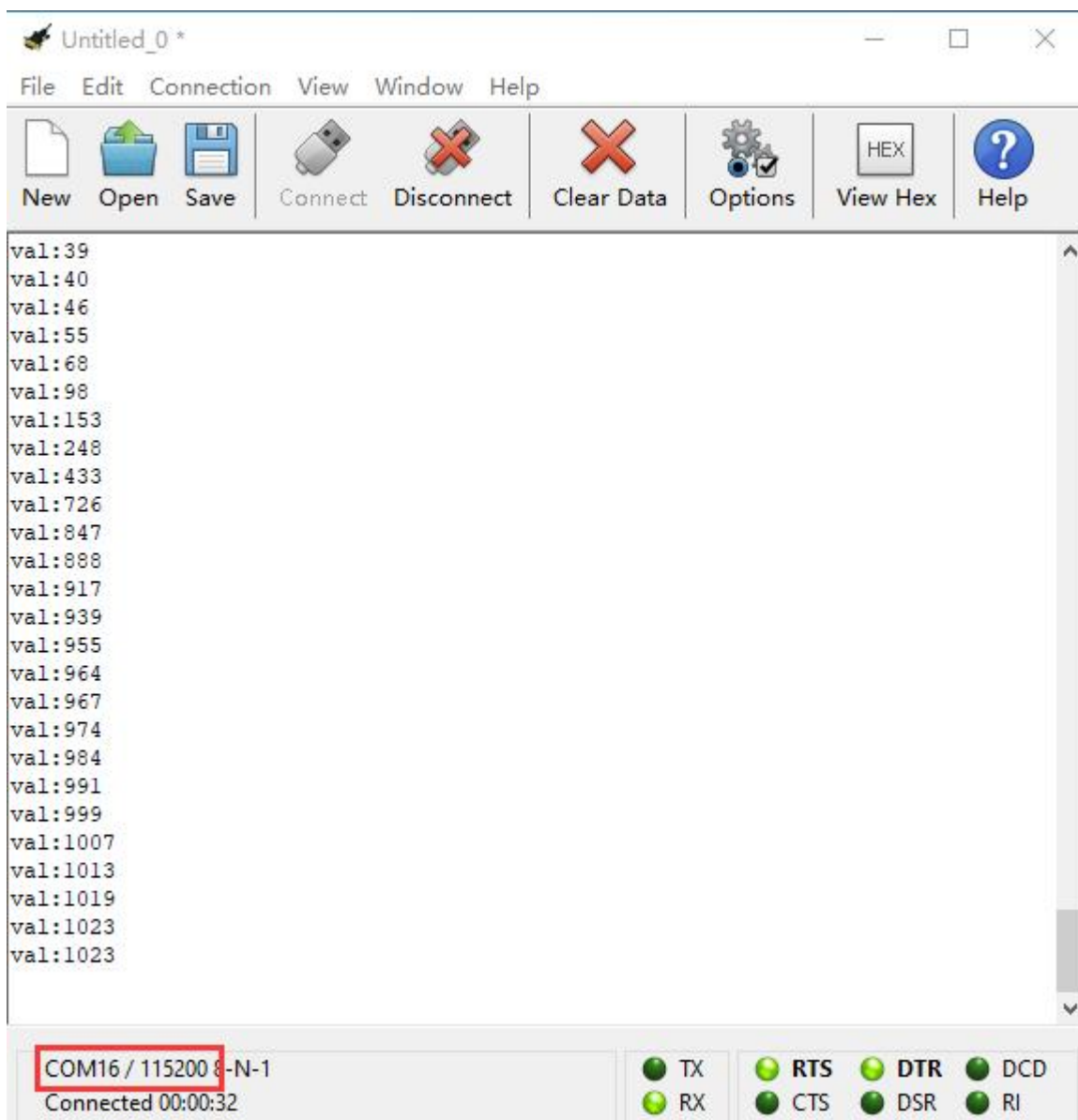
Set the detected analog signal(0-1023) by photoresistor and map the analog value of LED connected P10.

## 6.Test Result:

Wiring up, dial Voltmeter\_Switch to 3V end, plug in external power and dial Power\_Switch to ON end and upload code to micro:bit.

Open CoolTerm, click Options and select SerialPort, set COM port and baud rate, set baud rate to 115200. Tap OK and Connect.

As the ambient light brightness increases, so does the analog value on CoolTerm monitor, red LED gradually gets brighter; by contrast, analog value reduces and LED gets dimmer.



## Project 23: Fire Alarm

### 1.Description:

The violent fire will cause huge economic and human loss if without any effective measures.

You only need a flame sensor that can alarm if sensing the fire.

In this program, we will imitate the fire alarm system with flame sensor, passive buzzer and red LED module.

### 2.What You Need:

- Micro:bit Board\*1
- EASY Plug Shield for micro bit V1.1\*1
- Micro USB Cable\*1
- EASY Plug Flame Sensor\*1
- EASY Plug Red LED Module\*1
- RJ11 Cable\*2
- 6-Slot AA Battery Holder\*1
- 1.5V AA Battery\*6

## EASY Plug Flame Sensor:



The flame sensor can be used to detect fire or other wavelength at 760nm ~1100nm light.

In the fire-fighting robot game, the flame plays an important role in the probe, which can be used as the robot's eyes to find fire source.

The potentiometer on the flame sensor can be used to adjust the sensitivity.

### 3. Specification:

Working voltage: 3.3V to 5V

Detection angle: about 60°

Detection range: 20cm (4.8V) ~ 100cm (1V)

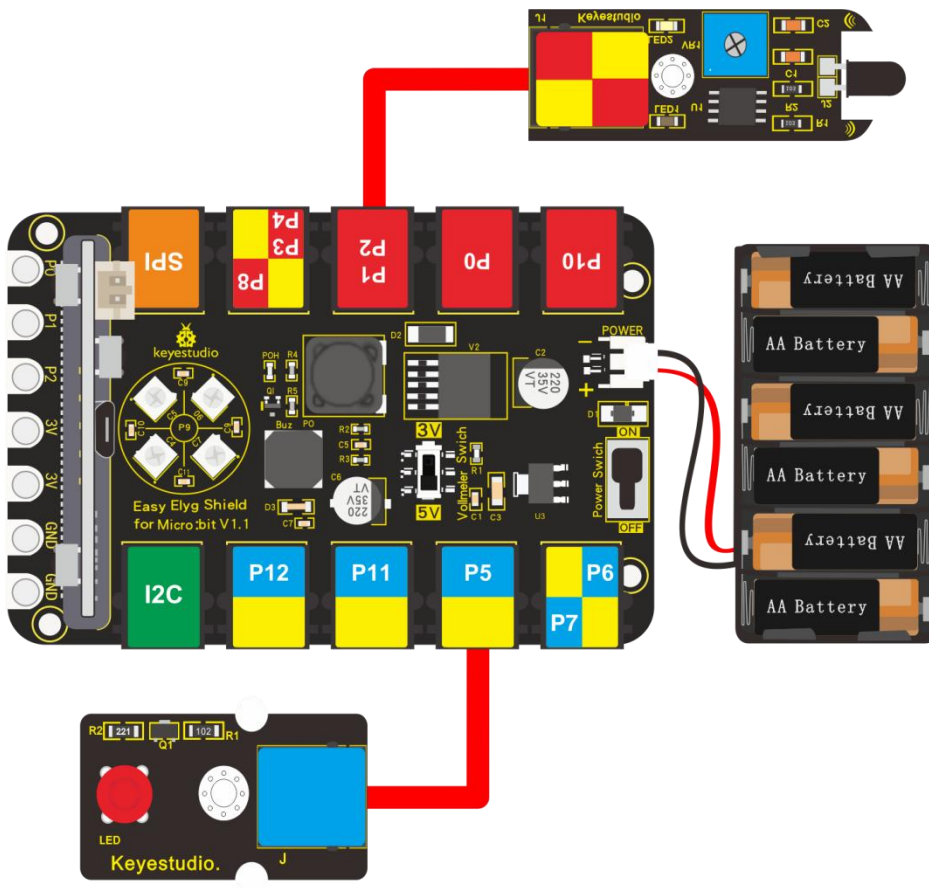
Spectral bandwidth: 760nm to 1100nm

Working temperature: -25°C to 85°C

Sensor type: digital

## 4. Wiring Up:

Insert micro:bit onto Easy Plug shield , connect flame sensor and red LED module to P1 and P5 port of shield with RJ11 cables.



Note: Dial Voltmeter\_Switch to 3V end

## 5. Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.



“on start” : command block only runs once to start program.  
 Turn off dot matrix  
 Set P0 to low level(0), turn off passive buzzer  
 The program under the block “forever” runs cyclically.  
 Micro:bit shows digital signals  
 If the digital signal is 0, there is flame, execute the program under then block  
 Play tone C for 1 beat, passive buzzer emits sound  
 Set P5 to high level(1), turn on LED  
 If the digital signal is 1, there is no flame, execute the program under else block  
 Set P5 to low level(0), turn off passive buzzer  
 Set P0 to low level(0), turn off passive buzzer

## 6.Test Result:

Wiring up, dial Voltmeter\_Switch to 3V end, plug in external power and dial Power\_Switch to ON end and upload code to micro:bit show 0(low level), LED is on and passive buzzer emits sound; by contrast 1 appears on micro:bit, LED is off and buzzer doesn't emit sound

## Project 24: Flammable Gas in the Air

### 1. Description

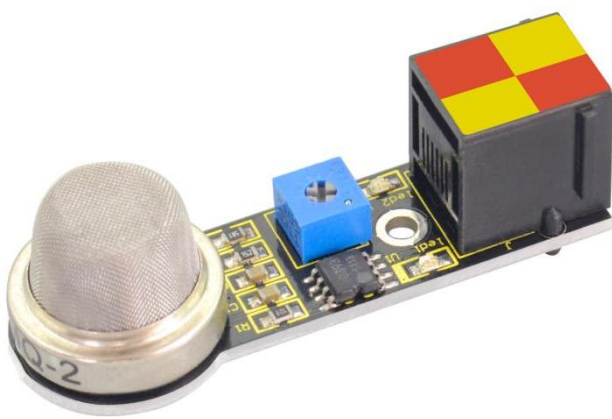
This gas sensor is used in gas leakage detecting equipment in consumer electronics and industrial markets.

This sensor is suitable for detecting LPG, I-butane, propane, methane, alcohol, Hydrogen and smoke. It has high sensitivity and quick response.

We will show you how to detect the flammable gas in the air with gas sensor.

## 2.What You Need:

- Micro:bit Board\*1
- EASY PlugShield for micro bit V1.1\*1
- Micro USB Cable\*1
- EASY Plug Analog Gas Sensor\*1
- EASY Plug Red LED Module\*1
- RJ11 Cable\*2
- 6-Slot AA Battery Holder\*1
- 1.5V AA Battery\*6



### EASY Plug Analog Gas Sensor:

This is a robust Gas sensor

[www.keyestudio.com](http://www.keyestudio.com)

suitable for sensing LPG, Smoke, Alcohol, Propane, Hydrogen, Methane and Carbon Monoxide concentrations in the air. If you are planning on creating an indoor air quality monitoring system; breath checker or early fire detection system, Easy Plug gas Sensor Module is a great choice.

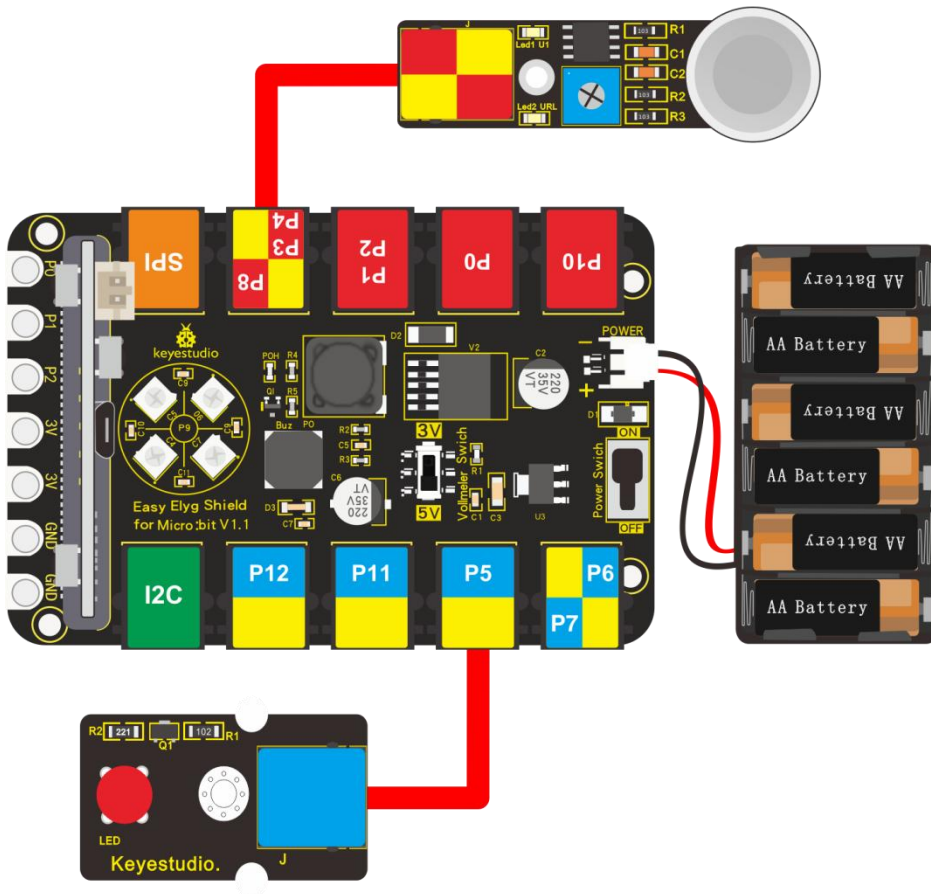
It has two signal terminals-pin A0 and D0. The value of A0 will rise up as the concentration of flammable gas.

### **3. Specification:**

- Port: Easy plug
- Working Voltage: 5V
- Interface Type: Digital and Analog
- Wide detecting scope
- Simple drive circuit
- Stable and long lifespan
- Quick response and High sensitivity

### **4. Wiring Up:**

Insert micro:bit onto EASY Plug shield, connect analog gas sensor and red LED module to P4 and P5 port of shield with RJ11 cables.



Note: dial Voltmeter\_Switch to 5V end

## 5.Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.

(Note: the gas analog value could be adjusted)



"on start" : command block only runs once to start program.

Turn off dot matrix on micro:bit

Set P0 to low level(0), turn off passive buzzer

The program under the block "forever" runs cyclically.

Serial writes analog signals read by analog gas sensor

If the analog signal read by P4 is greater than 600, execute the program under then block

Play C tone for 1 beat, passive buzzer emits sound

Set P5 to high level(1), turn on LED

If the analog signals read by P4 is less than and equivalent to 600, execute the program under else block

Set P5 to low level(0), turn off LED

Set P0 to low level(0), turn off passive buzzer

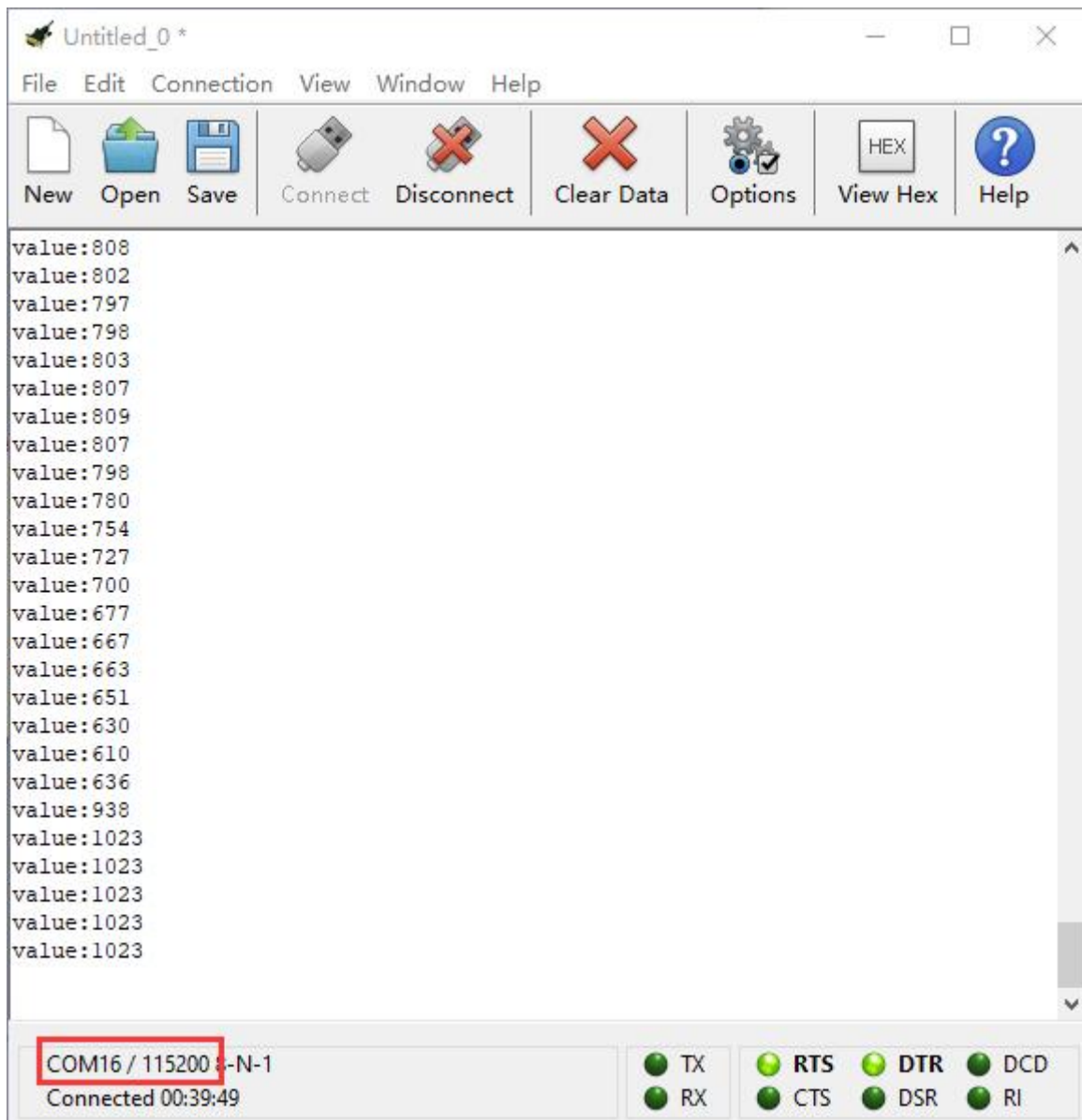
## 6.Test Result:

Wiring up, dial Voltmeter\_Switch to 5V end, plug in external power and dial Power\_Switch to ON end and upload code to micro:bit.

LED2 on gas sensor will be on, you could adjust the sensitivity(make LED at on-and-off state) with blue potentiometer

Open CoolTerm, click Options and select SerialPort, set COM port and baud rate, set baud rate to 115200. Tap OK and Connect.

Putt the firelighter close to analog gas sensor, the analog value on CoolTerm monitor will get larger and LED1 will be on. When sensor detects the analog value more than 600, buzzer will emit sound; otherwise, buzzer is silent and LED is off.



## Project 25: Ambient Light

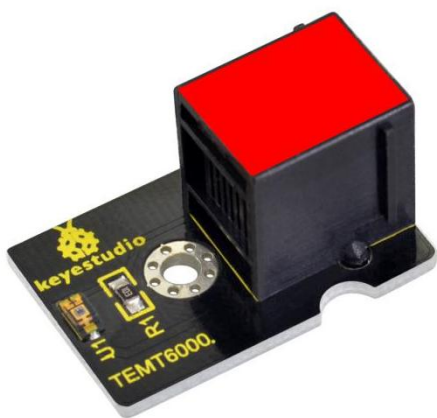
### 1. Description

In this project, we will detect the ambient brightness with EASY PlugTEMTE6000 light sensor.

## 2. What You Need:

- Micro:bit Board\*1
- EASY PlugShield for micro bit V1.1\*1
- Micro USB Cable\*1
- EASY Plug TEMT6000 Ambient Light Sensor\*1
- RJ11 Cable\*1
- 6-Slot AA Battery Holder\*1
- 1.5V AA Battery\*6

### **EASY Plug TEMT6000** Ambient Light Sensor:



The TEMT6000 was designed as an ambient light detector for automatically controlling the backlight dimming of cell phones, laptops, car dashboards and

similar items. It can be used in many applications where it is desirable to measure the relative brightness of the light falling on the sensor.

The sensor is designed to mainly detect the light spectrum visible to the human eye with peak sensitivity at 570nm which is in the green spectrum. The full range spans 440nm to 800nm.

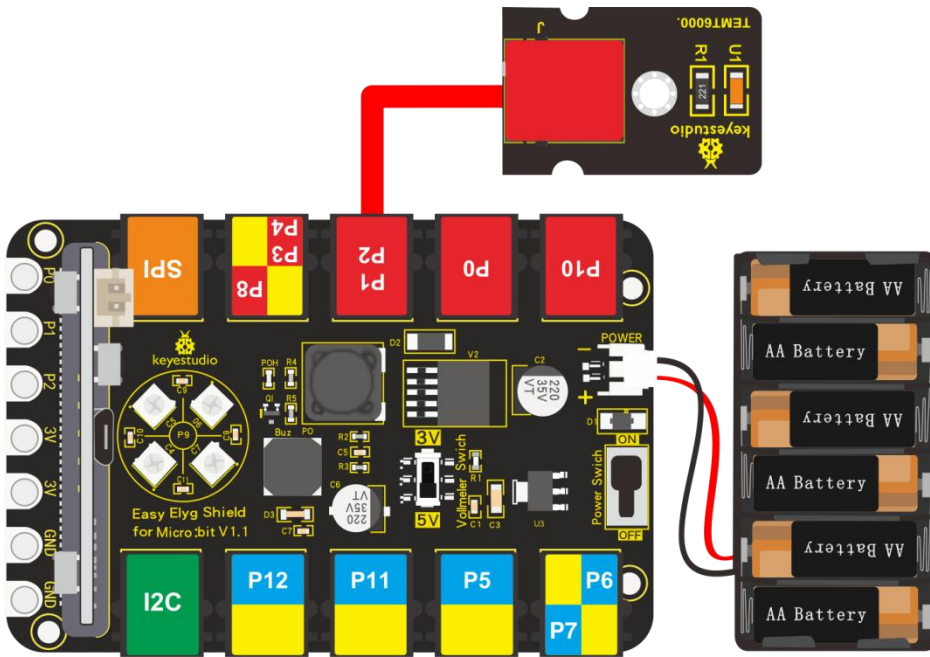
It does react well to very small changes in a wide range of brightness, however, it does not react well to IR or UV light. The sensor can help you to detect the light density.

### **3. Specification:**

- Working voltage: DC +5v
- Communication interface: analog voltage
- Only sensitive to visible light, no need for additional filters
- Recognizable light intensity range: 1 – 1000 Lux
- Good linear output

### **4.Wiring Up:**

Insert micro:bit onto EASY Plug shield, connect EASY Plug TEMT6000 ambient sensor to P1 port of shield with a RJ11 cable.

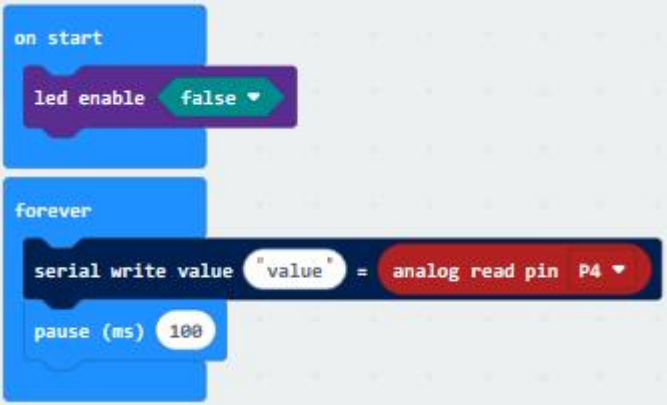


Note: Dial Voltmeter\_Switch to 5V end.

## 5.Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.



"on start" : command block only runs once to start program.

Turn off dot matrix on micro:bit

The program under the block "forever" runs cyclically.

Serial writes analog signals read by alcohol sensor

Delay in 100ms

## 6.Test Result:

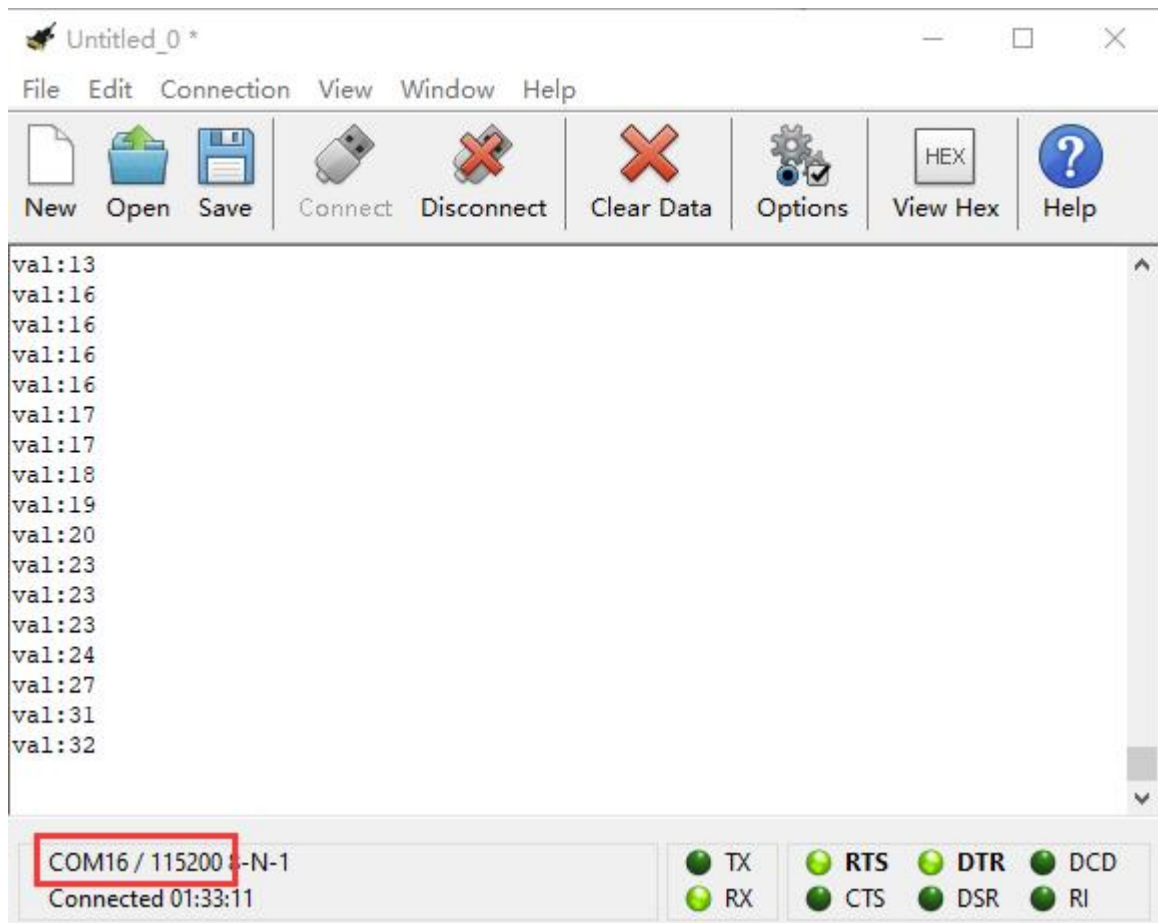
Wiring up, dial Voltmeter\_Switch to 5V end, plug in external power and dial Power\_Switch to ON end and upload code to micro:bit.(LED2 of sensor shows green color, and you could adjust potentiometer to keep LED on module in off-and-on state(the sensitivity is highest)

Open CoolTerm, click Options and select SerialPort, set COM port and baud rate, set baud rate to 115200. Tap OK and Connect.

The stronger the light is , the larger the analog value is; by contrast,

[www.keyestudio.com](http://www.keyestudio.com)

the smaller the analog value is, as shown below:



## Project 26: Slide Position

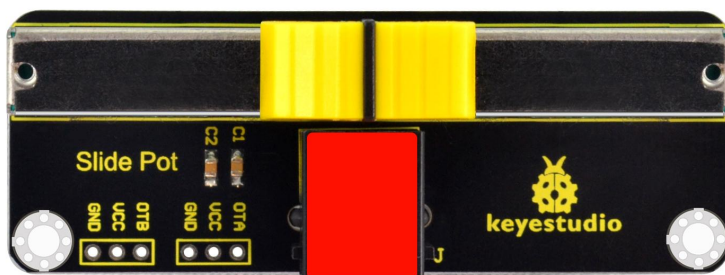
### 1. Description

You will learn how to use slide potentiometer to control LED and servo.

## 2.What You Need:

- Micro:bit Board\*1
- EASY Plug Shield for micro bit V1.1\*1
- Micro USB Cable\*1
- EASY Plug Slide Potentiometer\*1
- EASY Plug Red LED Module\*1
- EASY Plug Servo\*1
- Keystudio Servo\*1
- RJ11 Cable\*3
- 6-Slot AA Battery Holder\*1
- 1.5V AA Battery\*6

### EASY Plug Slide Potentiometer



The EASY plug slide potentiometer uses high-quality sliding appliances for stable

and reliable performance. It is a dual analog output that outputs a 0-VCC analog voltage signal.

The module pins are extended into Registered jack, so you can easily connect it to EASY Plug control board using a RJ11 cable.

There are 6 pad interfaces on the module. So you can solder two 3pin

headers with a pitch of 2.54mm on the module.

It can be used to connect with other MCUs. The signal terminal outputs two analog values. The sum of the two analog values is 1023.

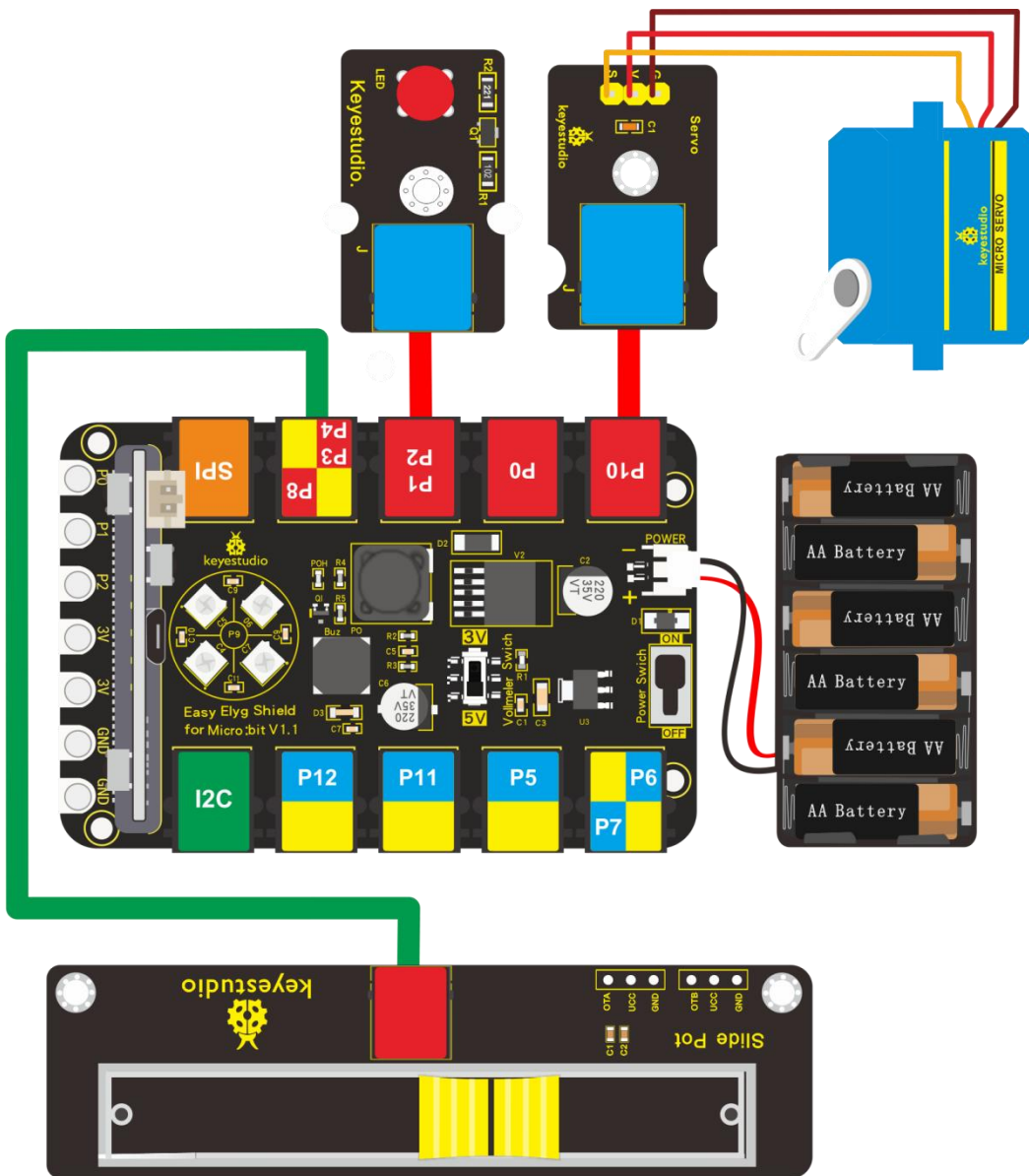
### **3. Specification:**

- Working voltage: DC 3.3V-5V
- Resistance: 10K $\Omega$
- Interface: analog interface
- Attribute: ROHS

### **4.Wiring Up:**

Insert micro:bit onto EASY Plug shield, connect slide potentiometer, red LED module and servo module to P3-P4, P1 and P10 port of shield with RJ11 cables.

The brown line, red line and orange line of servo are respectively connected to G, V and S

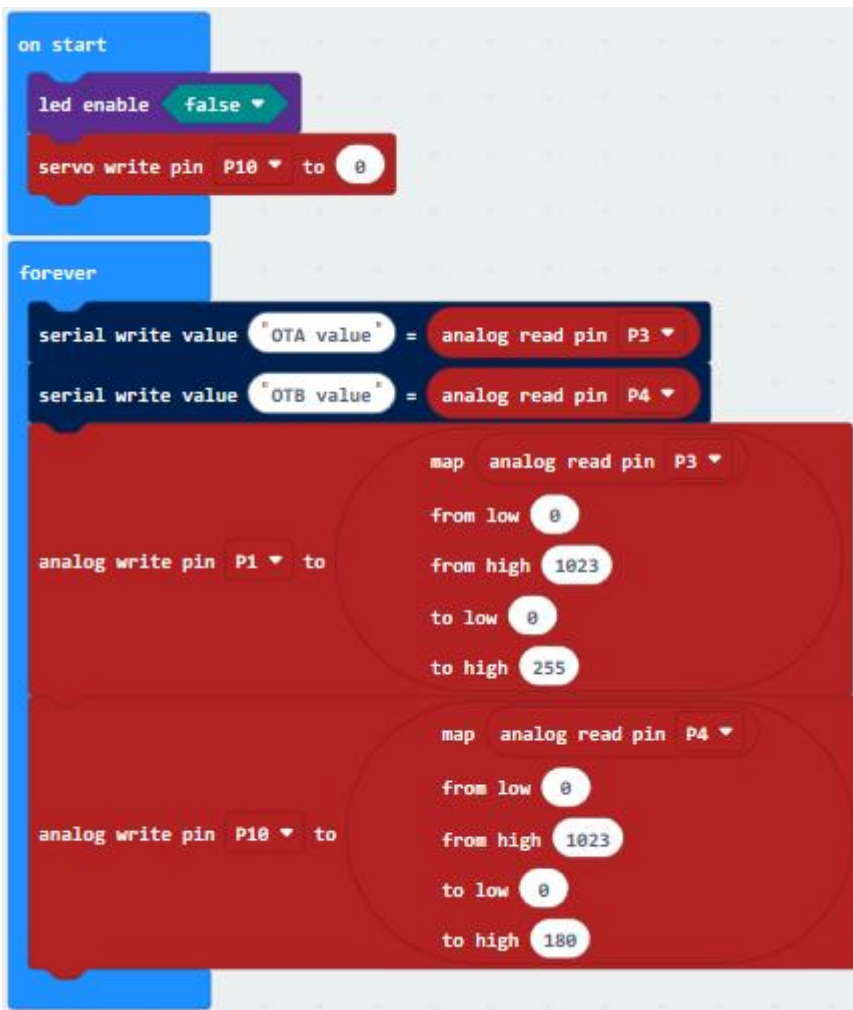


Note: Dial Voltmeter\_Switch to 5V end

## 5.Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.

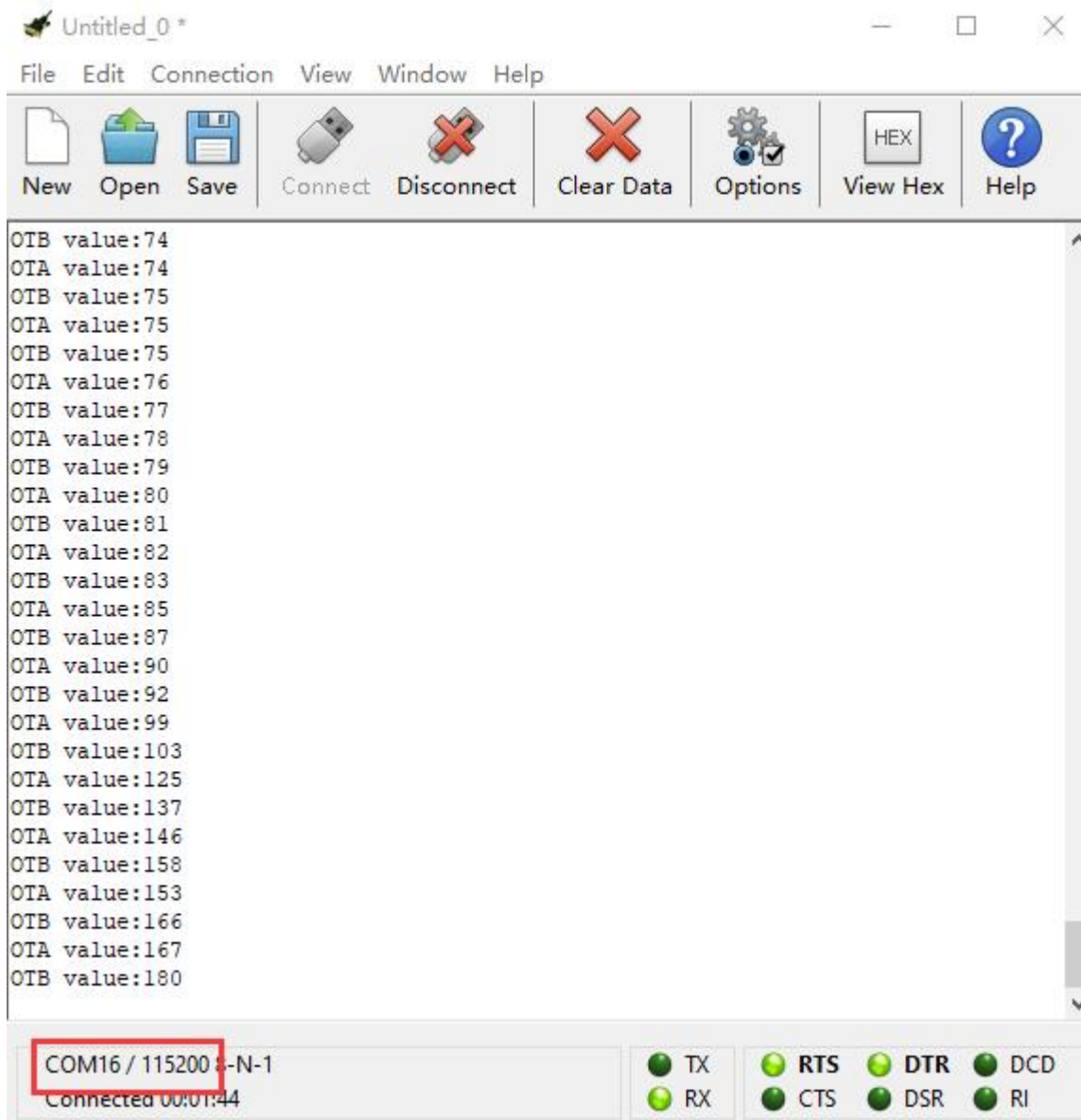


## 6.Test Result:

Wiring up, dial Voltmeter\_Switch to 5V end, plug in external power and dial Power\_Switch to ON end and upload code to micro:bit.

Open CoolTerm, click Options and select SerialPort, set COM port and baud rate, set baud rate to 115200. Tap OK and Connect.

CoolTerm monitor displays analog value of pin P3 and P4, if slide the potentiometer, value will alter in 0-1023, and servo will rotate in the range of 0°-180°, as shown below:



## Project 27: OLED Display

### 1. Description:

OLED module is applied widely in mobile devices, in this project, we will demonstrate how to display numbers, context and pictures with OLED module.

## 2. What You Need:

- Micro:bit Board\*1
- EASY Plug Shield for micro bit V1.1\*1
- Micro USB Cable\*1
- EASY Plug OLED Module\*1
- RJ11 Cable\*1
- 6-Slot AA Battery Holder\*1
- 1.5V AA Battery\*6



### EASY Plug OLED Module

OLED is short for organic light emitting diode. On the microscopic level, an OLED display is a matrix of organic LEDs that light up when they emit energy.

Our EASY Plug OLED displays are perfect when you need a small display with vivid,

high-contrast color.

The visible portion of the OLED measures 0.96" diagonal and contains 128 x 64 pixels.

An OLED display works without a backlight. Thus, it can display deep black levels and can be thinner and lighter than a liquid crystal display (LCD).

In low ambient light conditions such as a dark room an OLED screen can achieve a higher contrast ratio than an LCD.

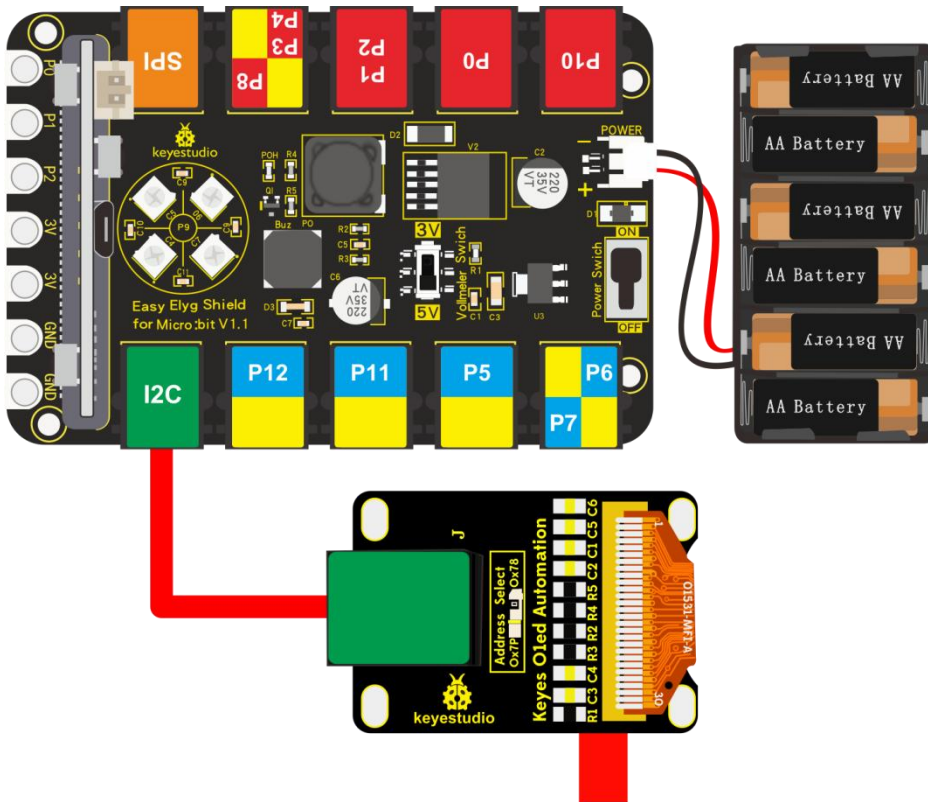
OLED technology is used in commercial applications such as displays for mobile phones and portable digital media players, car radios and digital cameras among others.

### **3. Specification:**

- 0.96" diagonal OLED
- Pixels: 128 × 64
- Color Depth: Monochrome (White)
- 5V power
- Brightness (cd/m<sup>2</sup>): 100 (Typ)

### **4.Wiring Up:**

Insert micro:bit onto EASY Plug shield, connect OLED module to I2C port of shield with a RJ11 cable.



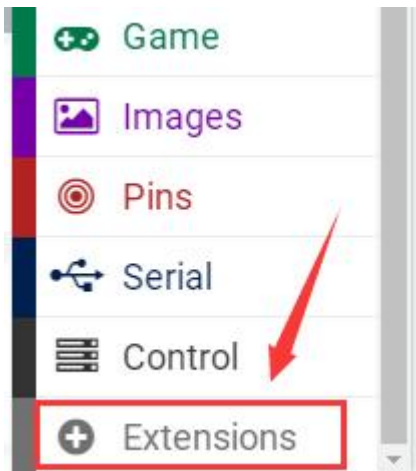
Note: Dial Voltmeter\_Switch to 5V end.

## 5.Test Code:

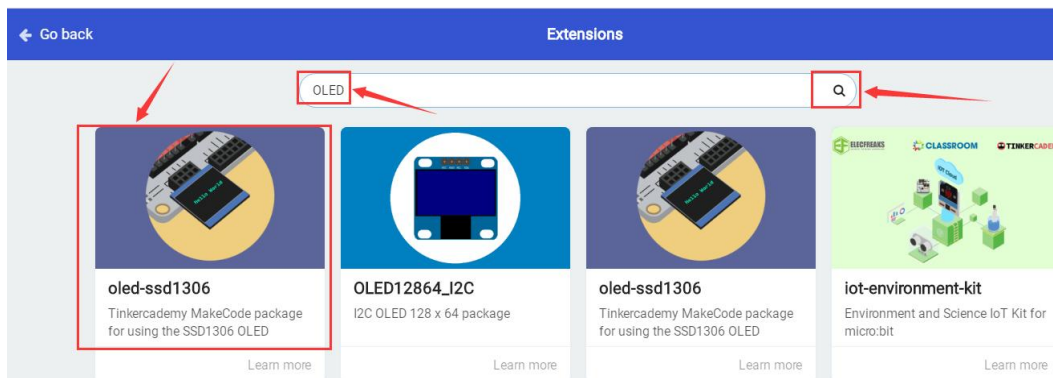
You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.

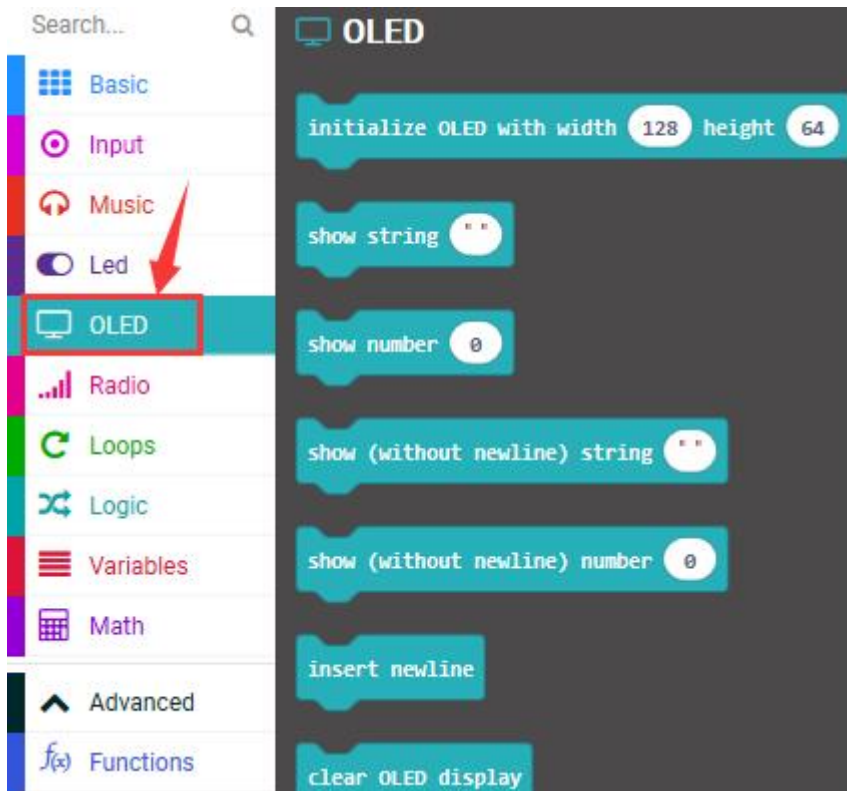
Next, we need to add the library file of OLED module:



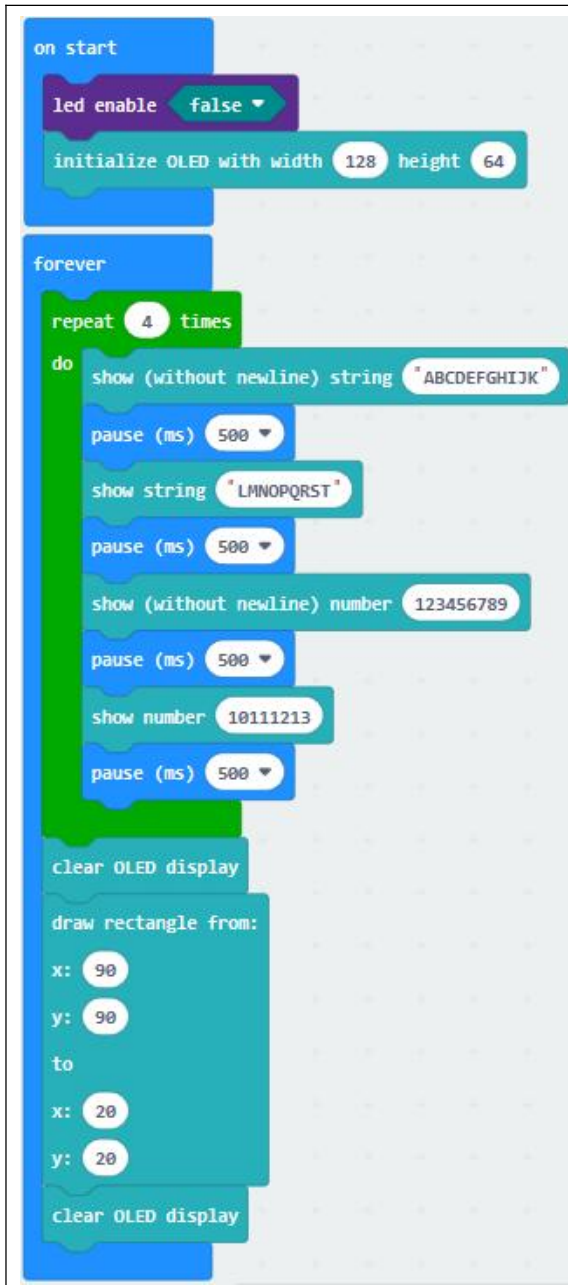
Search "OLED", as shown below, select the first "oled-ssd1306" library file, and install library.



After the installation, the library of OLED module will be displayed in the listed blocks on Makecode editor.



Test Code



"on start" : command block only runs once to start program.

Turn off dot matrix on micro:bit

OLED initializes, 128 in width, 64 in height

The program under the block "forever" runs cyclically.

Repeat 4 times

OLED shows character string "ABCDEFGHIJK"

Delay in 500ms

OLED shows character string "LMNOPQRST"

Delay in 500ms

OLED displays "123456789"

Delay in 500ms

OLED shows "10111213"

Delay in 500ms

Clear OLED display

Draw rectangular from x:90 y:90 to x:20 y:20

Clear OLED display

## 6.Test Result:

Wiring up, dial Voltmeter\_Switch to 5V end, plug in external power and dial Power\_Switch to ON end and upload code to micro:bit.

OLED display will show the corresponding characters and pictures.

## Project 28: Automatic Water Your Plant

### 1. Description

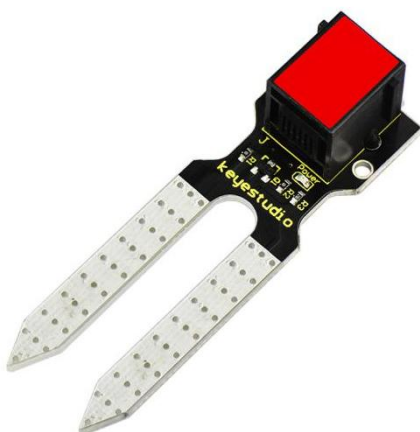
This is a simple soil humidity sensor aims to detect the soil humidity.

If the soil is in lack of water, the analog value output by the sensor will decrease; otherwise, it will increase. If you use this sensor to make an automatic watering device, it can detect whether your botany is thirsty to prevent it from withering when you go out.

### 2.What You Need:

- Micro:bit Board\*1
- EASY Plug Shield for micro bit V1.1\*1
- Micro USB Cable\*1
- EASY Plug Soil Humidity Sensor\*1
- EASY Plug OLED Module\*1
- RJ11 Cable\*2
- Slot AA Battery Holder\*1 1.5V AA Battery\*6

### EASY Plug Soil Humidity Sensor



The EASY Plug soil moisture sensor can

read the amount of moisture present in the soil surrounding it. It's an ideal for monitoring an urban garden, or your pet plant's water level. This soil moisture sensor uses the two probes to pass current through the soil, and then it reads that resistance to get the moisture level. More water makes the soil conduct electricity more easily (less resistance), while dry soil conducts electricity poorly (more resistance). If you use this sensor to make an automatic watering device, it will be helpful to remind you to water your indoor plants or to monitor the soil moisture in your garden.

### **3. Specification:**

- Working voltage: 3.3V-5V
- Working current:  $\leq 20\text{mA}$
- Output voltage: 0-2.3V (The greater the humidity, the higher the output voltage.) Sensor type: Analog output

Automotive Watering System:

We connect a soil humidity sensor and a relay module to MCU.

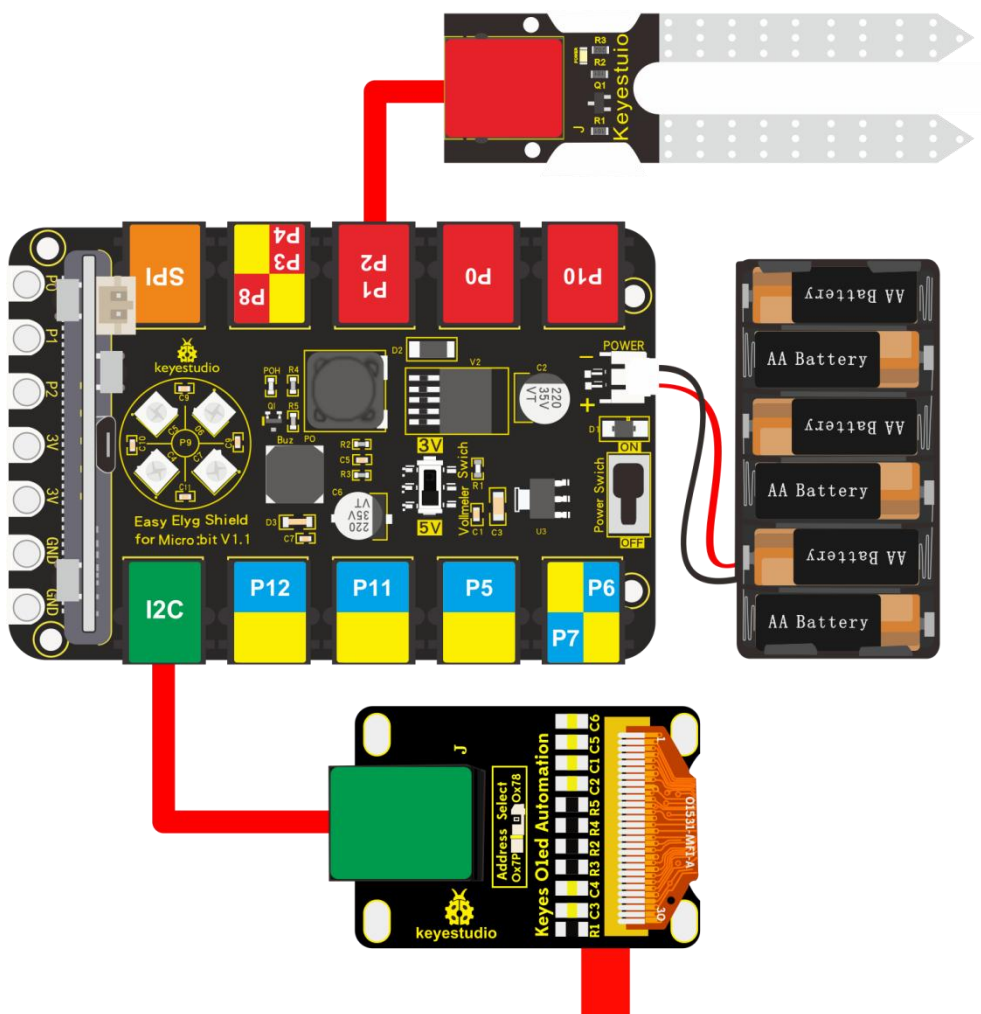
Separately link water pump and power on NO end of relay module.

When the soil lacks the water, MCU controls relay module(NO is

connected) to drive water pump to water plant; if the soil is wet enough, NO is cut off, water pump stops watering plant.

#### 4.Wiring Up:

Insert micro:bit onto EASY Plug shield, connect soil humidity sensor and OLED module to P1 and I2C port of shield.

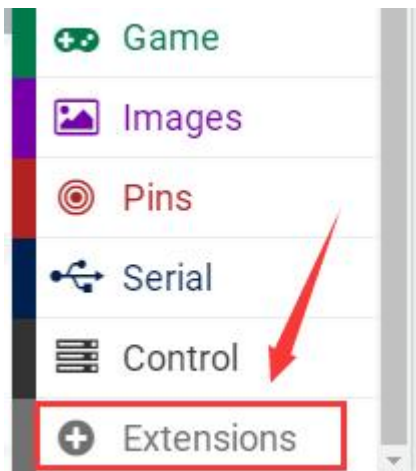


Note: Dial Voltmeter\_Switch to 5V end.

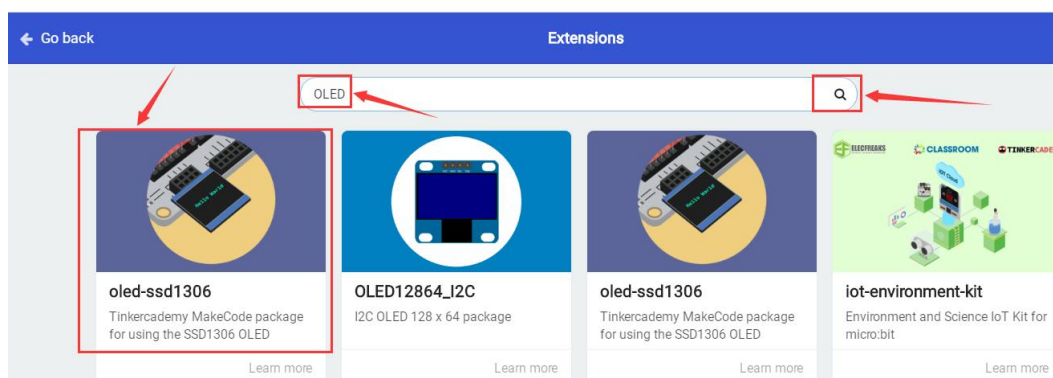
## 5. Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

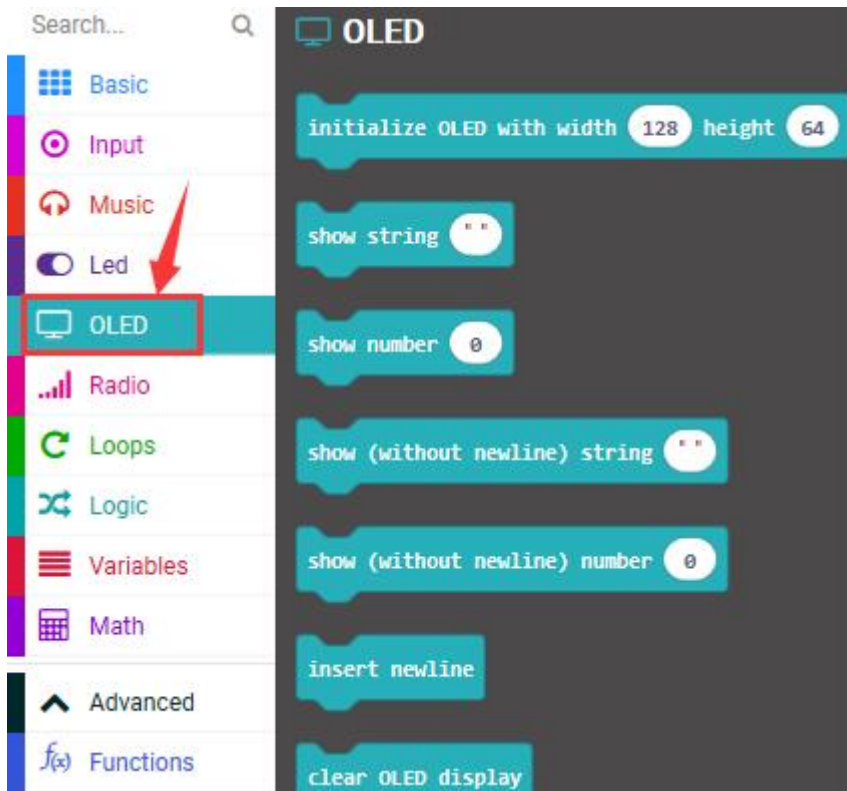
Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.



Search "OLED" , as shown below, select "oled -ssd1306" library and install it.



After the installation, you will view the OLED in the listed blocks.



Test Code:



"on start" : command block only runs once to start program.

Turn off dot matrix on micro:bit

OLED initializes, 128 in width, 64 in height

The program under the block "forever" runs cyclically.

Clear OLED display

Serial writes the soil humidity analog signals

OLED shows "Soil moisture value"

OLED shows the soil humidity analog signals

Delay in 300ms

## 6.Test Result:

Wiring up, dial Voltmeter\_Switch to 5V end, plug in external power and dial Power\_Switch to ON end and upload code to micro:bit.

Open CoolTerm, click Options and select SerialPort, set COM port and baud rate, set baud rate to 115200. Tap OK and Connect.

Insert the soil humidity sensor into soil, CoolTerm monitor and OLED module will display the analog value of soil humidity. As the humidity rises up, so does the analog value; otherwise, the analog soil humidity value will gradually decreases.

Untitled\_0 \*

File Edit Connection View Window Help

New Open Save Connect Disconnect Clear Data Options View Hex Help

Soil moisture value:329  
Soil moisture value:345  
Soil moisture value:416  
Soil moisture value:414  
Soil moisture value:406  
Soil moisture value:441  
Soil moisture value:450  
Soil moisture value:464  
Soil moisture value:483  
Soil moisture value:517  
Soil moisture value:531  
Soil moisture value:535  
Soil moisture value:527  
Soil moisture value:537  
Soil moisture value:537  
Soil moisture value:539  
Soil moisture value:544  
Soil moisture value:586  
Soil moisture value:633  
Soil moisture value:608  
Soil moisture value:584  
Soil moisture value:589  
Soil moisture value:609  
Soil moisture value:646  
Soil moisture value:644  
Soil moisture value:650

COM16 / 115200 8-N-1  
Connected 00:00:49

TX RX RTS CTS DTR DSR DCD RI



## Project 29: Ultrasonic Ranging

### 1. Description

Ultrasonic sensor can detect the distance. In this program, we will conduct you how to detect the distance with it.

### 2. What You Need:

- Micro:bit Board\*1
- EASY Plug Shield for micro bit V1.1\*1
- MicroUSB Cable\*1
- EASY Plug Ultrasonic Sensor\*1
- RJ11 Cable\*1
- 6-Slot AA Battery Holder\*1
- 1.5V AA Battery\*6

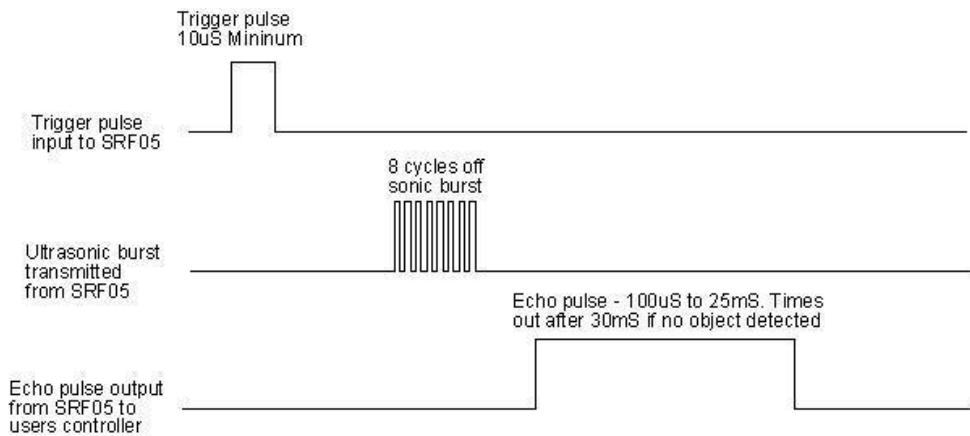
### EASY Plug Ultrasonic Sensor:



The ultrasonic module will emit the ultrasonic waves after trigger signal. When the ultrasonic waves encounter the object and are reflected back, the module outputs an echo signal, so it can



determine the distance of object from the time difference between trigger signal and echo signal.



1. Use method and timing chart of ultrasonic module:

Setting the delay time of Trig pin of SR04 to 10µs at least, which can trigger it to detect distance.

2. After triggering, the module will automatically send eight 40KHz ultrasonic pulses and detect whether there is a signal return. This step will be completed automatically by the module.

3. If the signal returns, the Echo pin will output a high level, and the duration of the high level is the time from the transmission of the ultrasonic wave to the return.

the distance measured by ultrasonic wave = (speed \* time)/2



### 3. Specification:

Power Supply :+5V DC

Quiescent Current : <2mA

Working Current: 15mA

Effectual Angle: <15°

Ranging Distance : 2cm – 400 cm

Resolution : 0.3 cm

Measuring Angle: 30 degree

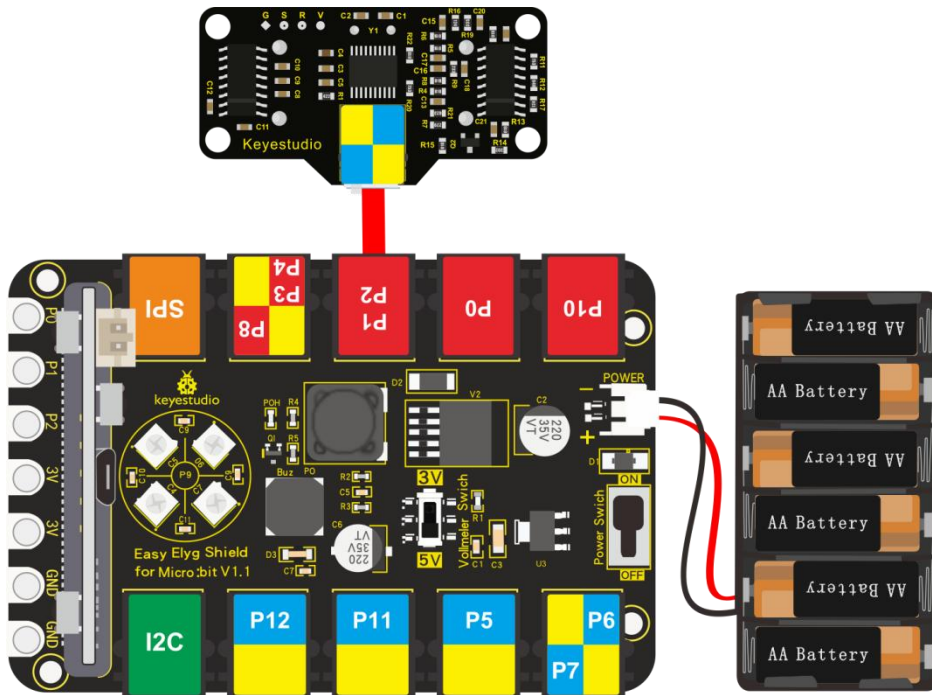
Trigger Input Pulse width: 10uS

Output echo signal: output TTL level(high), proportional to distance

Interface: dual digital I/O port

### 4.Wiring Up:

Insert micro:bit onto EASY Plug shield, connect an ultrasonic sensor to P1-P2 of shield.



Note: Dial Voltmeter\_Switch to 5V end

### 5.Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.

### Code 1:



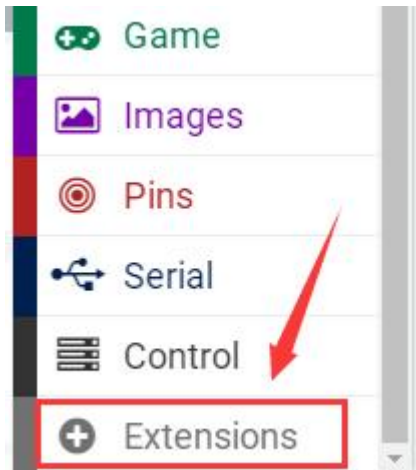
```
on start
  led enable false

forever
  digital write pin P1 to 0
  wait (µs) 2
  digital write pin P1 to 1
  wait (µs) 10
  digital write pin P1 to 0
  set distance to pulse in (µs) pin P2 pulsed high integer ÷ 58
  serial write value 'distance(cm)' = distance
  pause (ms) 50
```

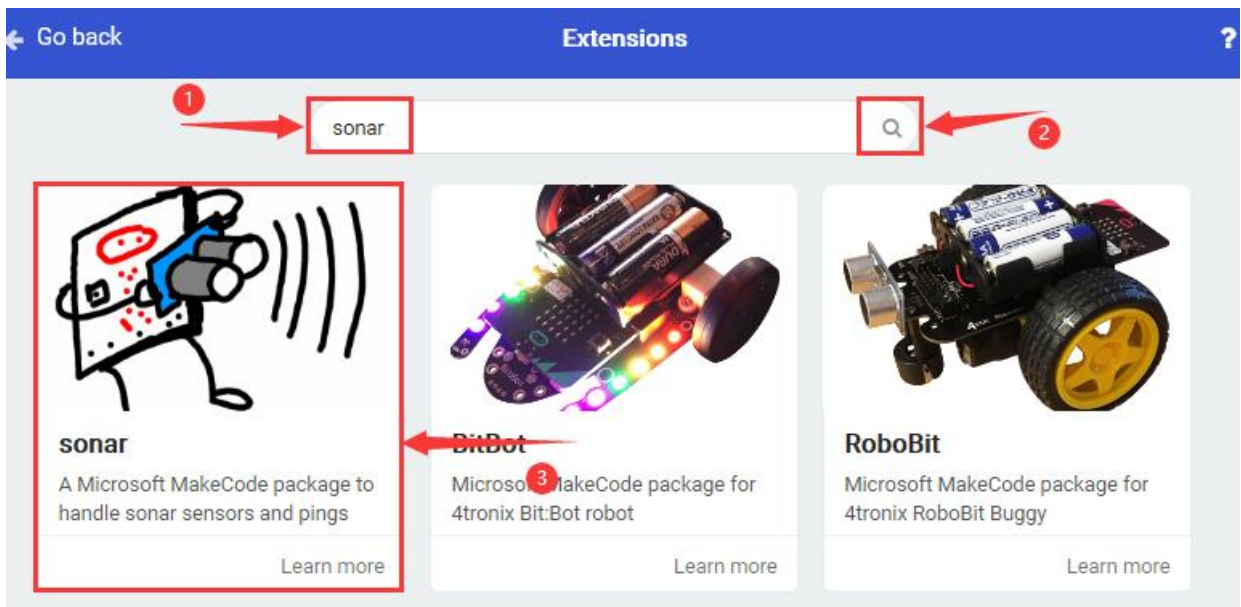
“on start” : command block only runs once to start program.  
Turn off dot matrix on micro:bit  
The program under the block “forever” runs cyclically.  
Set P1 to low level(0)  
Delay in 2ms  
Set P1 to high level(1)  
Delay in 10ms  
Set P1 to low level(0)  
Serial writes distance(cm)=distance  
Delay in 50ms

## Code 2:

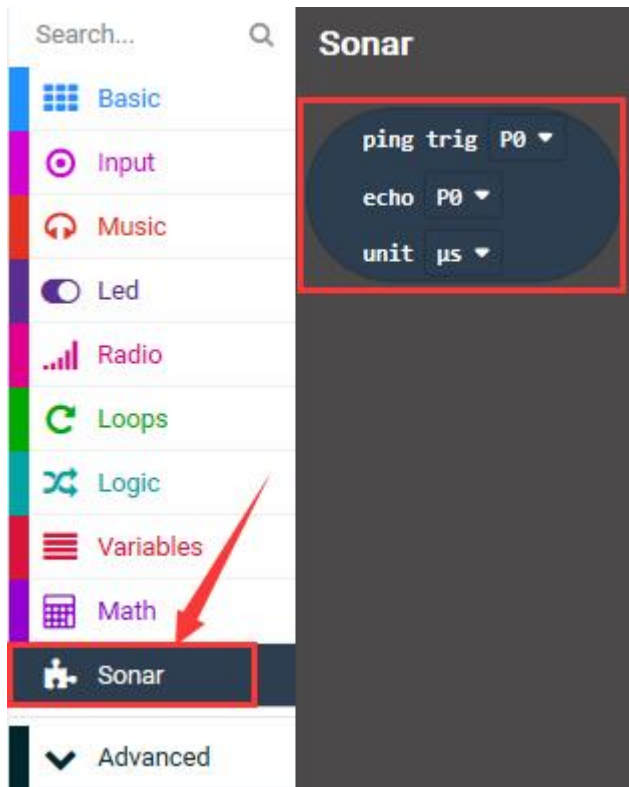
Next to add the library of ultrasonic sensor, click “Extensions”



Search "sonar" , and click sonar library to download it.



After the installation, you will view it appear in the listed blocks on Makecode editor.



Test Code:





“on start” : command block only runs once to start program.

Turn off LED matrix on micro:bit

The program under the block “forever” runs cyclically.

Set the distance read by ultrasonic sensor to variable distance

Delay in 50ms

Serial writes distance(cm)=the distance value read by ultrasonic sensor

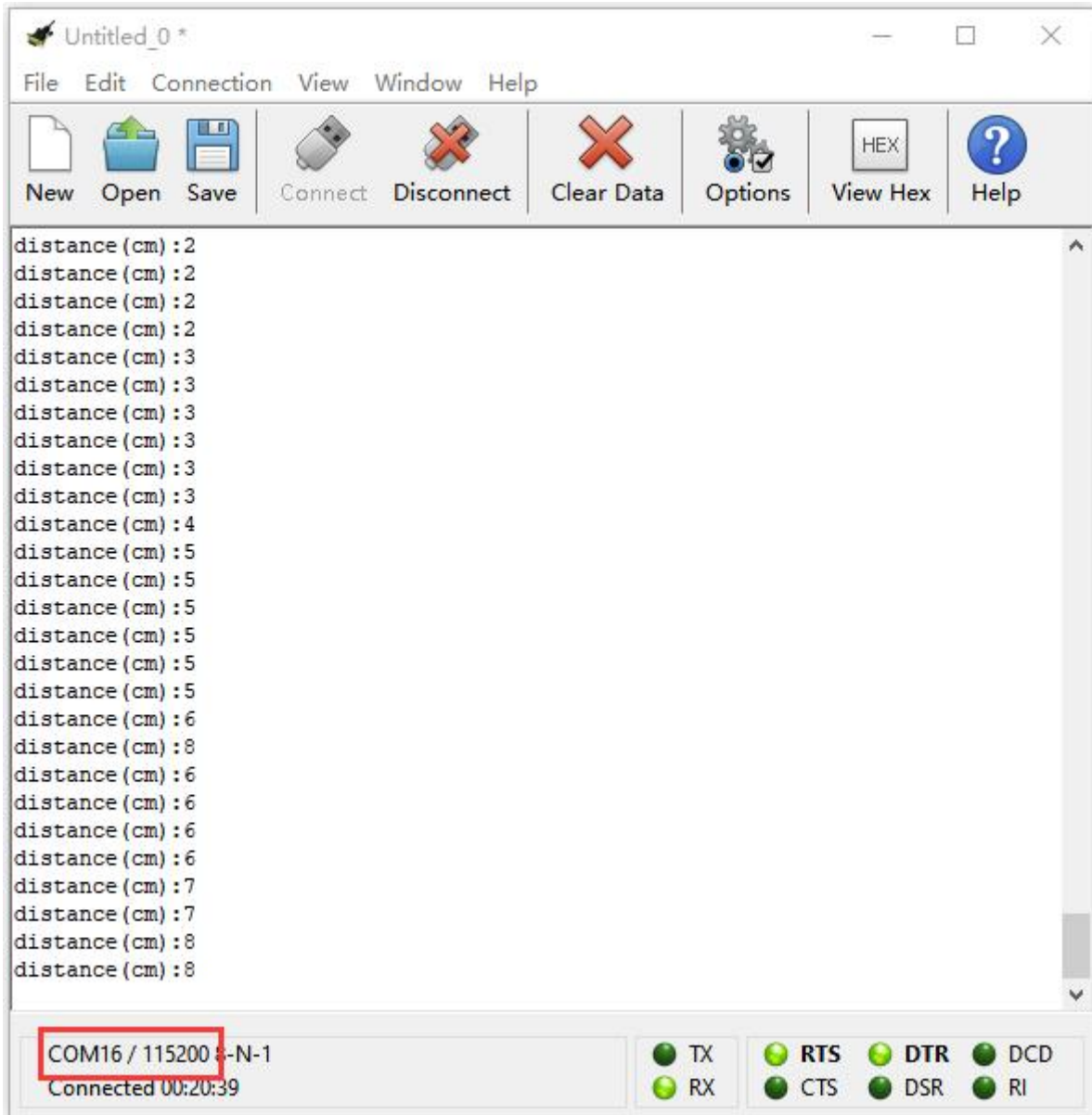
## 6.Test Result:

Wiring up, dial Voltmeter\_Switch to 3V end, plug in external power and dial Power\_Switch to ON end and upload code 1 and 2 to micro:bit.

Open CoolTerm, click Options and select SerialPort, set COM port and baud rate, set baud rate to 115200. Tap OK and Connect.

Micro:bit and CoolTerm monitor will display the current temperature, as shown below:

CoolTerm monitor will display the distance value



## Project 30: IR Remote Control Decoding

### 1. Description

Every mad scientist's lab, or teenager's secret room, needs advanced protection against intrusion by rogue agents or siblings. If you are one of

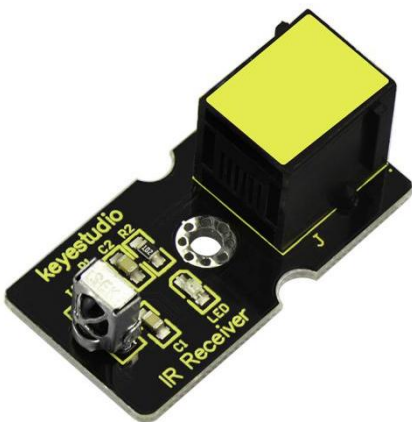


them, you should probably consider getting a Passive infrared (PIR) sensor for you. PIR sensors allow you to detect when someone is in your room when they shouldn't be.

## 2.What You Need:

- Micro:bit Board\*1
- EASY Plug Shield for micro bit V1.1\*1
- Micro USB Cable\*1
- EASY Plug IR Receiver Module\*1
- EASY Plug IR Remote Control\*1
- RJ11 Cable\*1
- 6-Slot AA Battery Holder\*1
- 1.5V AA Battery\*6

### EASY Plug IR Receiver Module



There is no doubt that infrared remote control is ubiquitous in daily life. It is used to control various household appliances, such as TVs, stereos, video recorders and satellite signal receivers. Infrared remote control is composed of infrared transmitting and infrared receiving systems, that is, an infrared



remote control and infrared receiving module and a single-chip microcomputer capable of decoding.



The 38K infrared carrier signal emitted by remote controller is encoded by the encoding chip in the remote controller. It is composed of a section of pilot code, user code, user inverse code, data code, and data inverse code. The time

interval of the pulse is used to distinguish whether it is a 0 or 1 signal and the encoding is made up of these 0, 1 signals.

The user code of the same remote control is unchanged. The data code can distinguish the key.

When the remote control button is pressed, the remote control sends out an infrared carrier signal. When the IR receiver receives the signal, the program will decode the carrier signal and determines which key is pressed. The MCU decodes the received 01 signal, thereby judging what key is pressed by the remote control.

Infrared receiver we use is an infrared receiver module. Mainly composed of an infrared receiver head, it is a device that integrates reception, amplification, and demodulation. Its internal IC has completed demodulation, and can achieve from infrared reception to output and be compatible with TTL signals. Additionally, it is suitable for infrared remote



control and infrared data transmission. The infrared receiving module made by the receiver has only three pins, signal line, VCC and GND. It is very convenient to communicate with arduino and other microcontrollers.

### **3. Specification:**

Interface: Easy plug

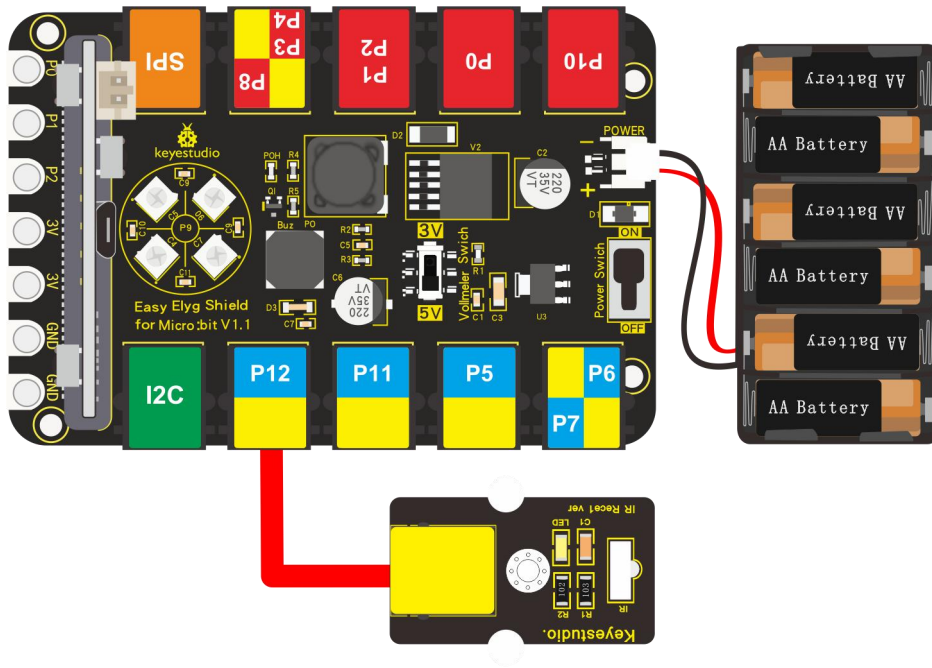
Working voltage: 5V Interface

Type: Digital

Modulation frequency: 38Khz

### **4.Wiring Up:**

Insert micro:bit onto EASY Plug shield, connect an IR receiver module to P12 port of shield.



Note: Dial Voltmeter\_Switch to 5V.

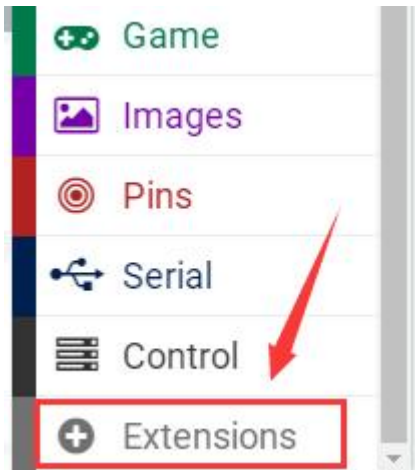
## 5.Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.

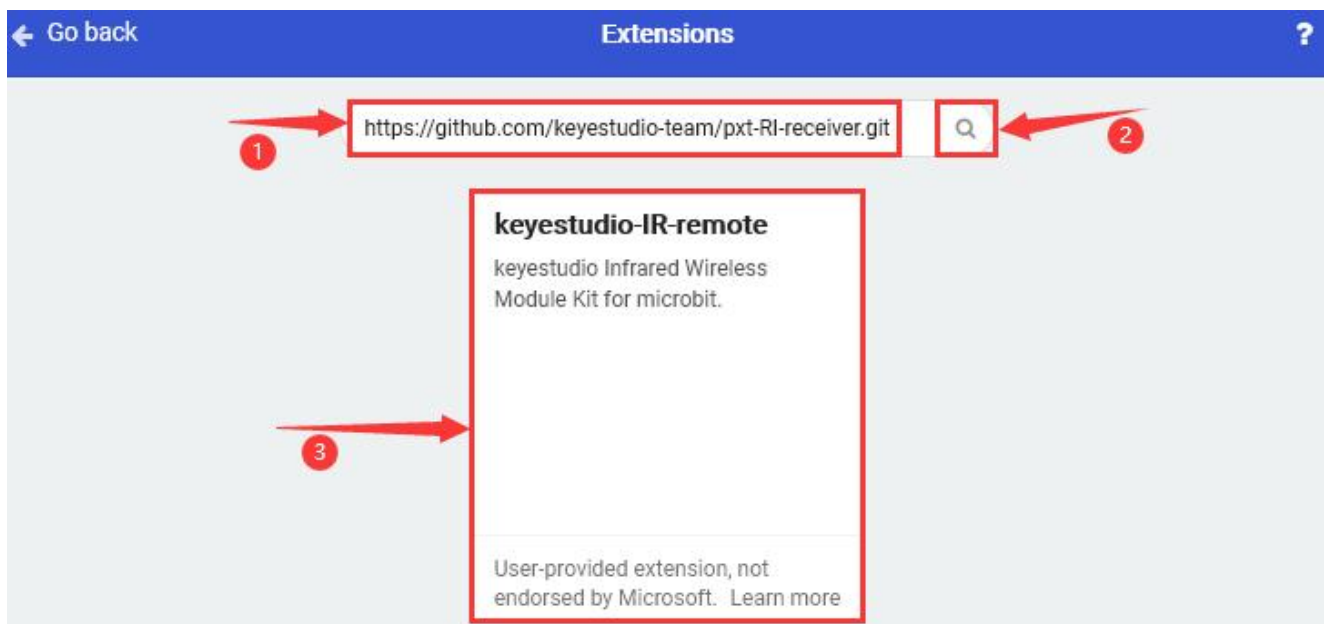
Next, we need to add the library of IR receiver module

Click "Extensions" on Makecode editor

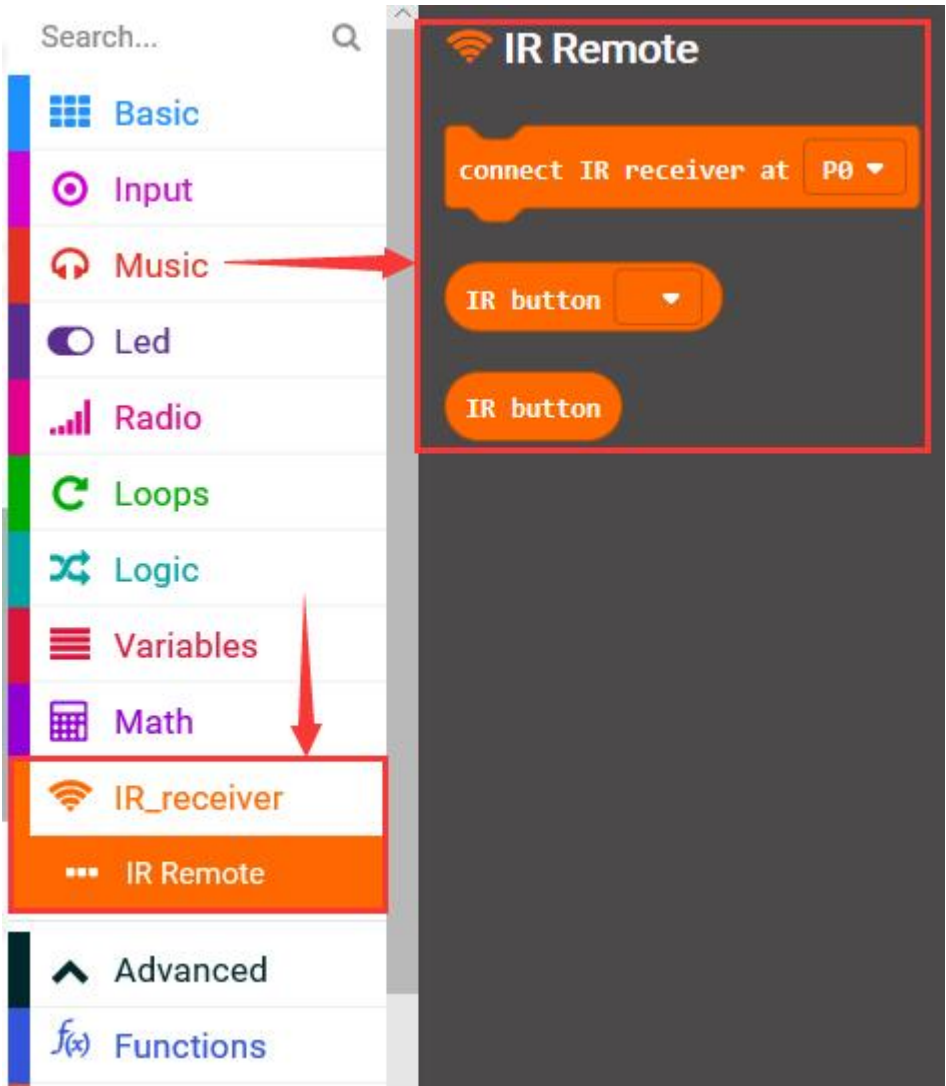


Enter the library link: Copy

<https://github.com/keyestudio-team/pxt-RI-receiver.git> and search, and download “keyestudio-IR-remote” file.



After the installation, you will see the “IR remote” in the listed blocks.



### Test Code:

```
on start
  led enable false
  connect IR receiver at P12

forever
  set val to IR button
  serial write value "IR" = val
  pause (ms) 1000
```

"on start" : command block only runs once to start program.  
Turn off LED matrix  
Connect IR receiver to P12  
The program under the block "forever" runs cyclically.  
Set val to IR button  
Serial writes IR=val  
Delay in 1000ms



**Code explanation:** serial monitor will refresh and show 0, when the key is not pressed; when it is pressed, CoolTerm will show the corresponding value.

**Note:**

1. Battery is not included, you need to make a purchase it(model:CR2025)
2. Make sure IR remote control is good, open camera on your cellphone, point IR remote control to camera and press button. It is good if you see purple light flashing.

**6.Test Result:**

Wiring up, dial Voltmeter\_Switch to 5V end, plug in external power and dial Power\_Switch to ON end and upload code to micro:bit.

Open CoolTerm, click Options and select SerialPort, set COM port and baud rate, set baud rate to 115200. Tap OK and Connect.

CoolTerm monitor will display the corresponding key values, as shown below:

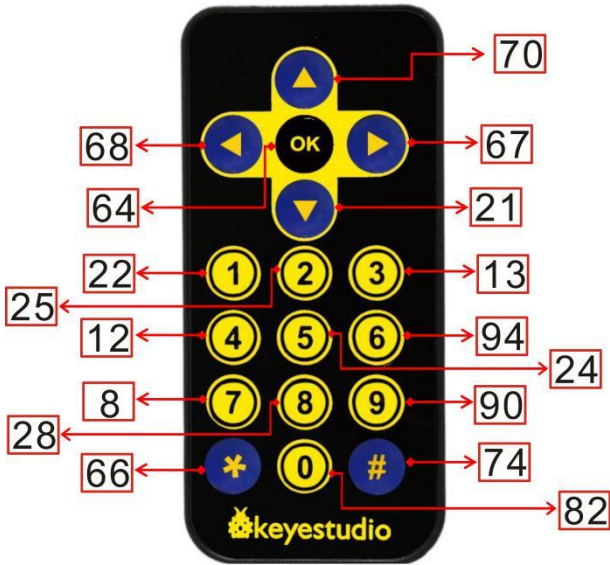


The screenshot shows the Keyestudio software interface. At the top, there is a title bar with the text "Untitled\_0\*" and standard window controls. Below the title bar is a menu bar with "File", "Edit", "Connection", "View", "Window", and "Help". A toolbar contains icons for "New", "Open", "Save", "Connect", "Disconnect", "Clear Data", "Options", "View Hex", and "Help". The main area displays a list of IR codes:

```
IR: 0
IR: 0
IR: 70
IR: 0
IR: 68
IR: 21
IR: 0
IR: 0
IR: 67
IR: 0
IR: 64
IR: 0
IR: 22
IR: 25
IR: 13
IR: 0
IR: 12
IR: 0
IR: 0
IR: 24
IR: 94
IR: 8
IR: 0
IR: 28
IR: 0
IR: 90
IR: 66
IR: 0
IR: 82
IR: 74
IR: 0
IR: 0
```

At the bottom, there is a connection status bar. On the left, it shows "COM16 / 115200 8-N-1" and "Connected 00:04:47". On the right, there are several status indicators represented by green circles: TX, RX, RTS, CTS, DTR, DSR, DCD, and RI.

Below we have listed out each button value of keyestudio remote control. So you can keep it for reference.



## Project 31: IR Remote Control RGB

### 1. Description

We decode the IR remote control with an IR receiver module and a micro:bit board before doing experiment. We could get key values which can control the external sensors/ modules after decoding IR remote control. In this experiment, we control 2812 2x2 full color RGB of shield with these key values.

### 2. What You Need:

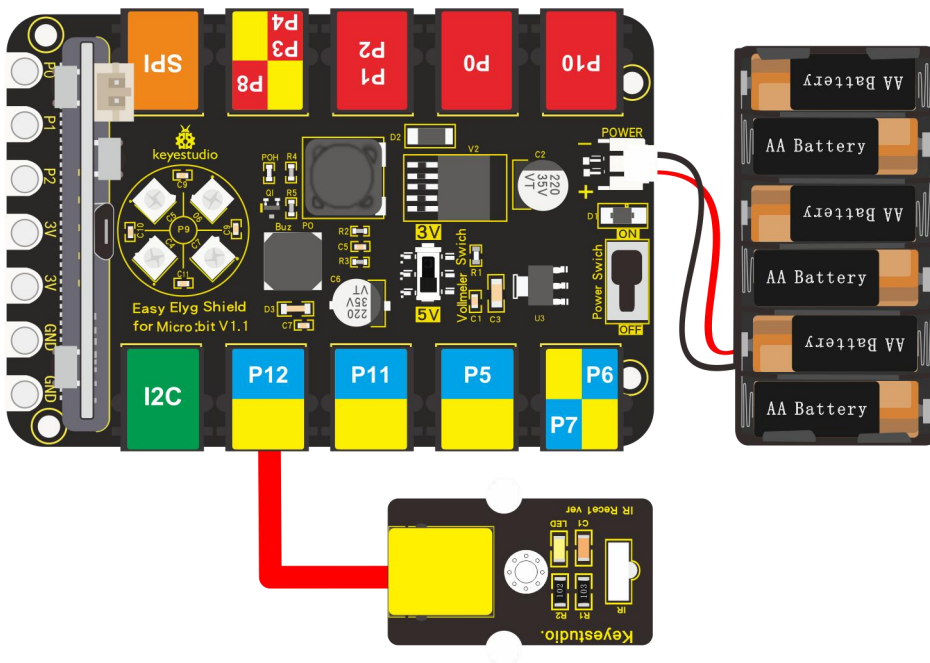
- Micro:bit Board\*1
- EASY Plug Shield for micro bit V1.1\*1
- Micro USB Cable\*1
- EASY Plug IR Receiver\*1



- EASY Plug IR Remote Control\*1
- RJ11 Cable\*1
- 6-Slot AA Battery Holder\*1
- 1.5V AA Battery\*6

### 3. Wiring Up:

Insert micro:bit onto EASY Plug shield, connect IR receiver to P12 port to shield with a RJ11 cable.



Note: Dial Voltmeter\_Switch to 5V end.

### 4.Test Code:

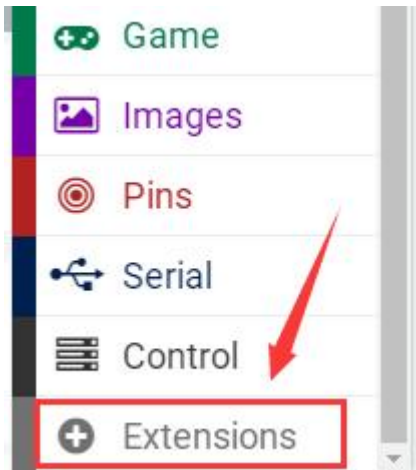
You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.



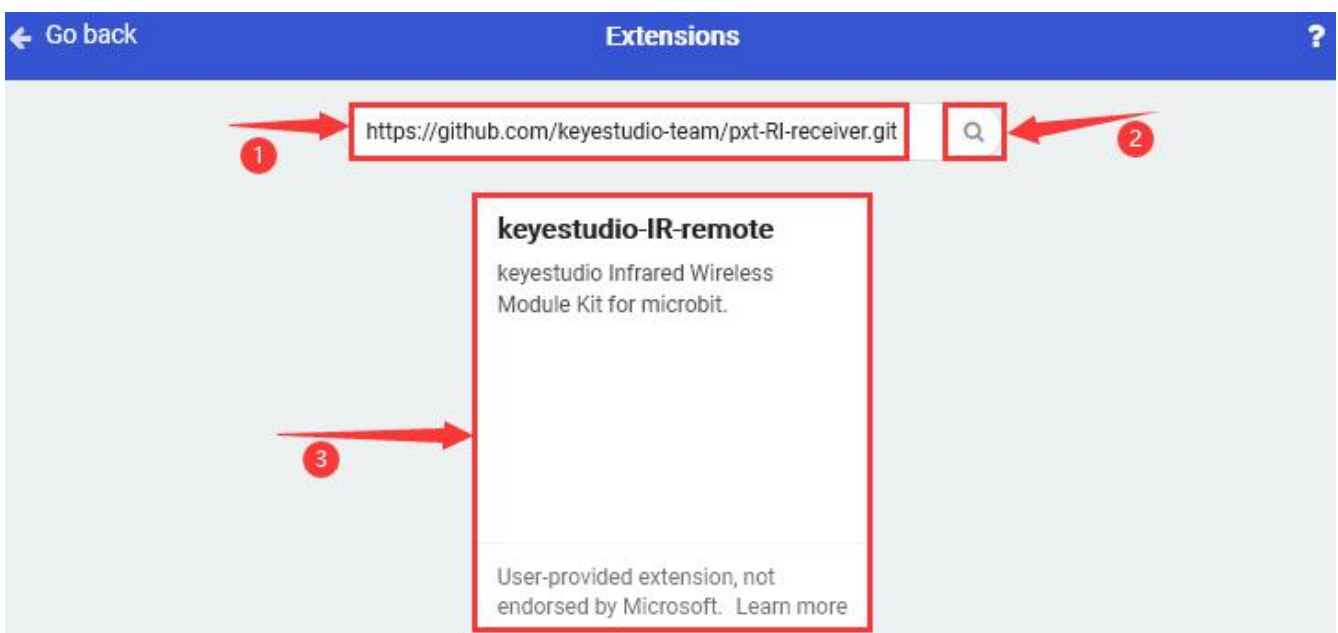
Next, we need to add the library file of IR receiver module.

Click "Extensions" on Makecode



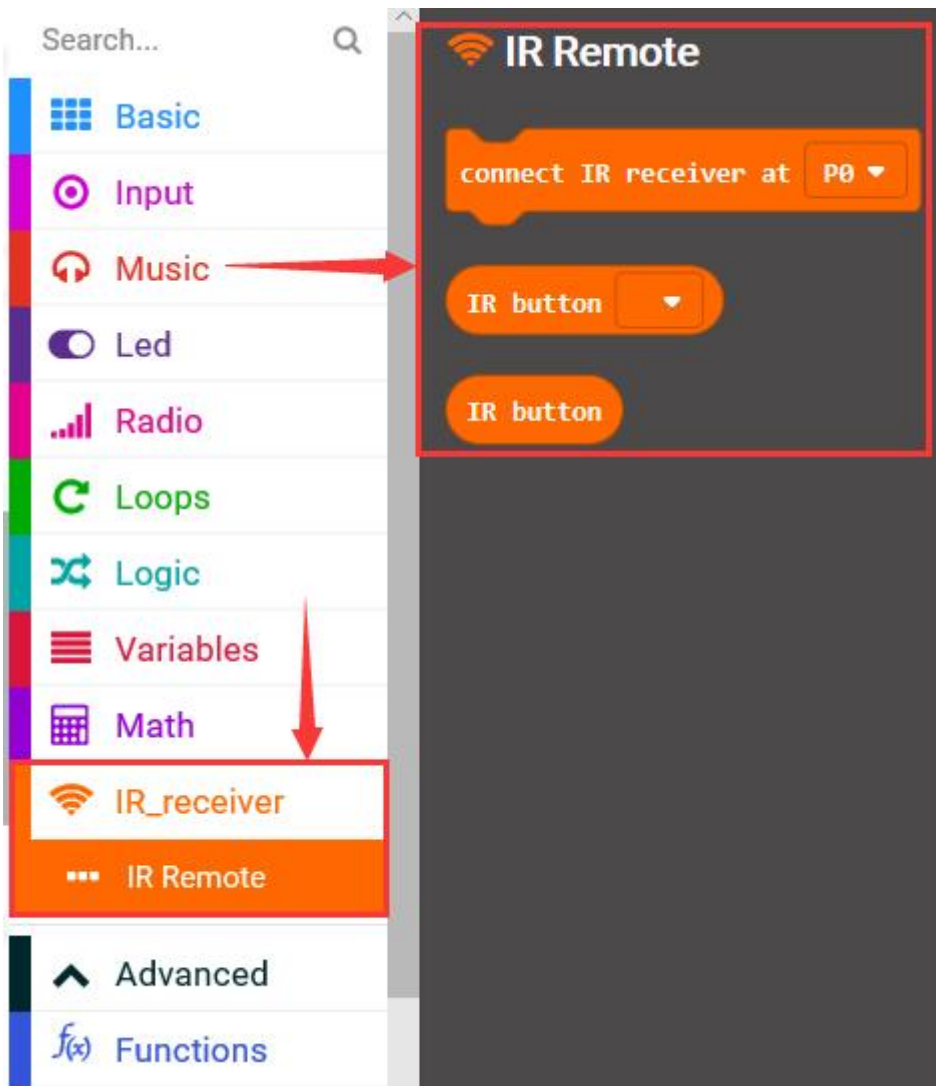
Enter the library link

<https://github.com/keyestudio-team/pxt-RI-receiver.git> and search, as shown below, download the file

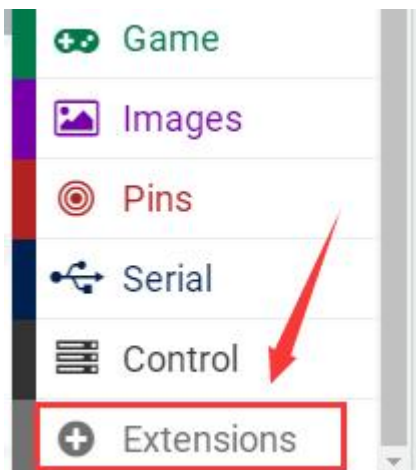




Then you could view the library of IR receiver module in the listed blocks.

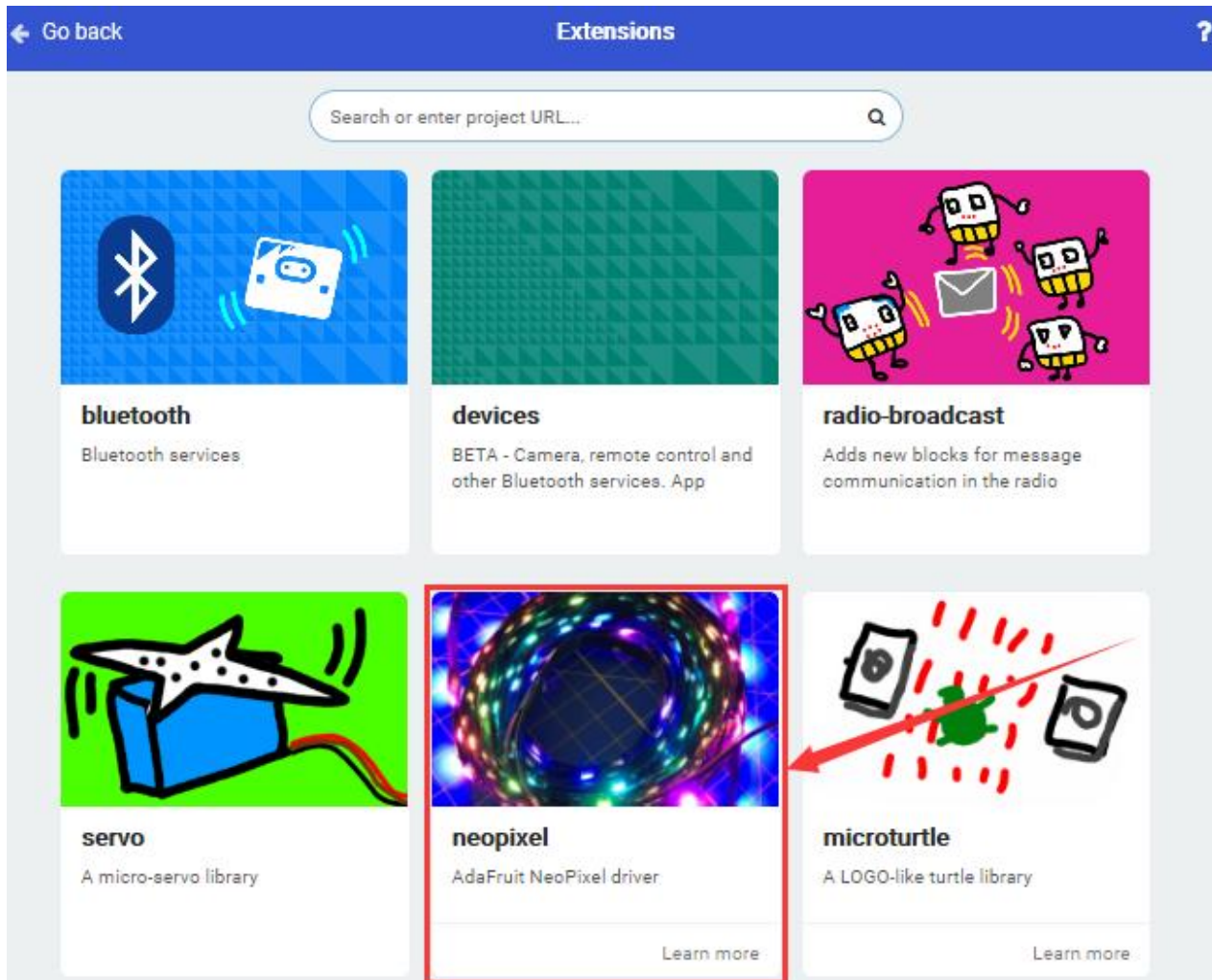


Repeat the above steps to add the library of "neopixel" :





Click "Extensions" → "neopixel" , after second, install the library of "neopixel" .





Search...

- Basic
- Input
- Music
- Led
- Radio
- Loops
- Logic
- Variables
- Math
- Neopixel**
- more

### Neopixel

- set strip to NeoPixel at pin P0 with 24 leds as RGB (GRB format)
- set range to strip range from 0 with 4 leds
- strip show rainbow from 1 to 360
- strip show color red
- strip show bar graph of 0 up to 255
- strip show
- strip clear
- hue 0 saturation 0 luminosity 0
- strip shift pixels by 1
- strip rotate pixels by 1

### Test Code:

```
on start
  led enable false
  connect IR receiver at P12
  set strip to NeoPixel at pin P9 with 4 leds as RGB (GRB format)
  strip clear
  set val to 0
  set val2 to 0
```



```
forever
  set val to IR button
  if val != 0 then
    set val2 to val
    if val2 = 70 then
      strip show color red
    else if val2 = 68 then
      strip show color orange
    else if val2 = 67 then
      strip show color yellow
    else if val2 = 64 then
      strip show color green
    else if val2 = 21 then
      strip show color blue
    else if val2 = 22 then
      strip show color indigo
    else if val2 = 25 then
      strip show color violet
    else if val2 = 13 then
      strip show color purple
    else if val2 = 12 then
      strip show color white
```

“on start” : command block only runs once to start program.

Turn off dot matrix on micro:bit

Connect IR receiver to P12

Set strip to Neopixel at pin p9 with 4 leds as RGB

Turn off 4pcs WS2812 RGB lights

Set val to 0

Set val2 to 0

The program under the block “forever” runs cyclically.

Set val to IR button

When val≠0, execute the program under then block

Set val2 to val

When val2=70, execute the program under then block

All RGB display red color

When val2=68, execute the program under then block

All RGB show orange color

When val2=67, execute the program under then block

All RGB display yellow color

When val2=64, execute the program under then block

All RGB display green color

When val2=21, execute the program under then block

All RGB display blue color

When val2=22, execute the program under then block

All RGB display indigo color

When val2=25, execute the program under then block

All RGB display violet color

When val2=13, execute the program under then block

All RGB display purple color

When val2=12, execute the program under then block

All RGB display white color



## 6.Test Result:

Wire up, plug in power, dial Voltmeter\_Switch to 5V end and upload code to micro:bit.

Point IR remote control to IR receiver and press keys.

Key	Status	2812 2x2 full-color RGB
	Press	Red Color
	Press	Orange Color
	Press	Yellow Color
	Press	Green Color
	Press	Blue Color
	Press	Indigo Color
	Press	Violet Color
	Press	Purple Color
	Press	White Color



## Project 32: Joystick

### 1. Description

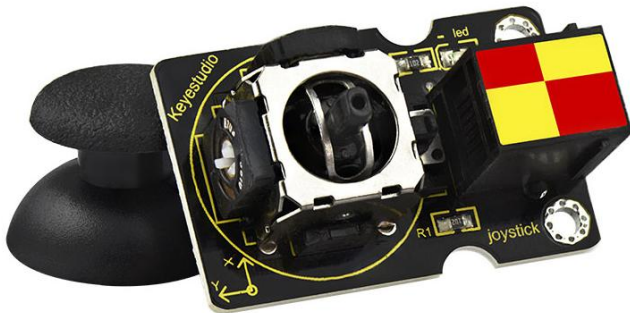
Lots of robot projects need joystick. This module provides an affordable solution.

By simply connecting to two analog inputs, the robot is at your commands with X, Y control. It also has a switch that is connected to a digital pin.

### 2.What You Need:

- Micro:bit Board\*1
- EASY Plug Shield for micro bit V1.1\*1
- Micro USB Cable\*1
- EASY Plug Joystick Module\*1
- RJ11 Cable\*1
- 6-Slot AA Battery Holder\*1
- 1.5V AA Battery\*6

### EASY Plug Joystick Module



This is a joystick very similar to the ' analog ' joysticks on PS2 (PlayStation 2) controllers. It is a self-centering spring loaded joystick,



meaning when you release the joystick it will center itself. It also contains a comfortable cup-type knob/cap which gives the feel of a thumb-stick.

The goal of the joystick is to communicate motion in 3D (3-axis) to an Arduino. This is achieved by housing two independent 10K potentiometers (one per axis). These potentiometers are used as dual adjustable voltage dividers, providing x and y axis analog input in a control stick form.

This joystick also contains a switch which activates when you push down on the cap. The switch is the small black box on the rear of the joystick.

The basic idea of a joystick is to translate the stick's position on three axes — the X-axis (left to right), the Y-axis(front and back) into electronic information an Arduino can process and the Z-axis(up and down)

### **3. Specification:**

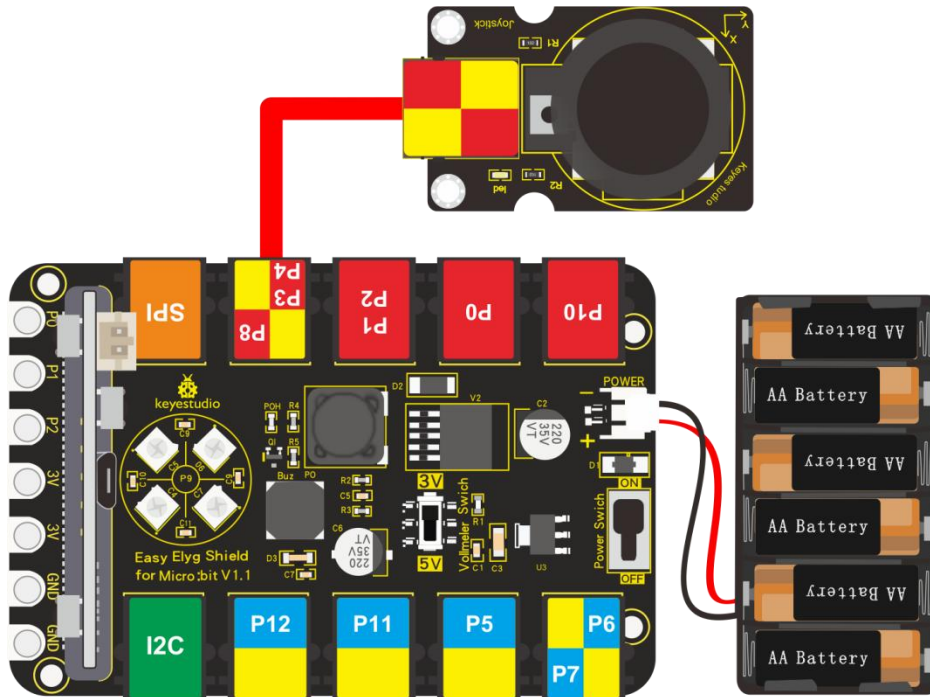
- Interface: Easy plug
- Working voltage: 3.3V to 5V
- Interface type: analog port and digital port

### **4.Wiring Up:**

Insert micro:bit onto EASY Plug shield and connect joystick module to



P3-P4-P8 port of shield with a RJ11 cable.




Note: Dial Voltmeter\_Switch to 3V end.

## 5. Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.



The image shows a Scratch script for a micro:bit. It starts with an "on start" block containing a "led enable" block set to "false". This is followed by a "forever" loop containing three "set" blocks: "set X to analog read pin P3", "set Y to analog read pin P4", and "set B to digital read pin P8". Below these are three "serial write value" blocks for "x", "y", and "B", each with a corresponding variable dropdown. The loop ends with a "pause (ms) 100" block.

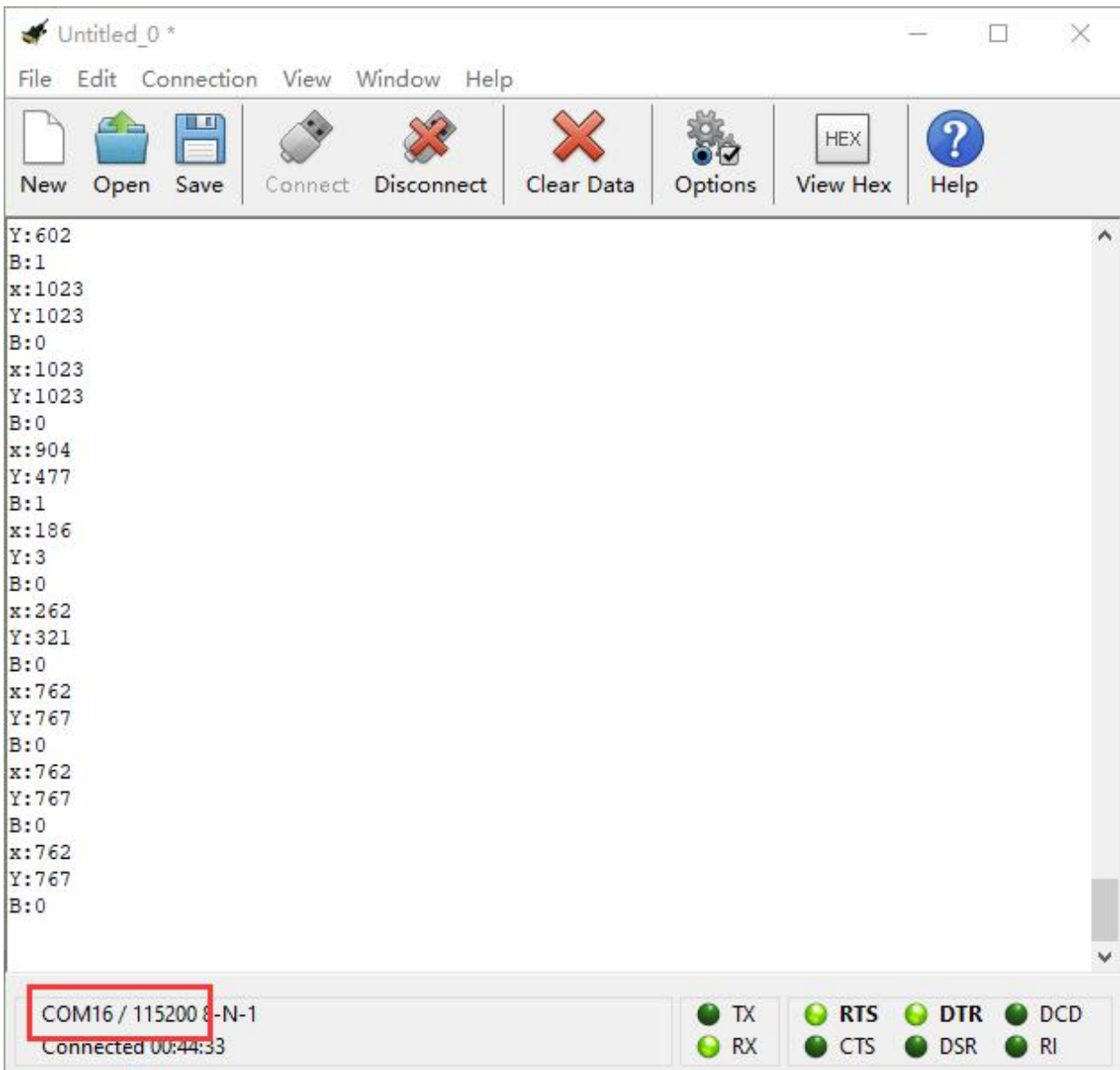
"on start": command block only runs once to start program.  
Turn off dot matrix on micro:bit  
The program under the block"forever"runs cyclically.  
Set variable X to digital signals on X axis  
Set variable Y to digital signals on Y axis  
Set variable B to digital signals on Z axis  
Serial writes X=P3 to read the analog signals on X axis  
Serial writes value X=P3 to read the analog signals on X axis  
Serial writes value Y=P4 to read the analog signals on Y axis  
Serial writes value B=P8 to read the digital signals on Z axis  
Delay in 100ms

### 6.Test Result:

Wiring up, dial Voltmeter\_Switch to 3V end, plug in external power and dial Power\_Switch to ON end and upload code to micro:bit.

Open CoolTerm, click Options and select SerialPort, set COM port and baud rate, set baud rate to 115200. Tap OK and Connect.

CoolTerm monitor shows the analog value on x, y axis and pin B



## Project 33: What Time Is It?

### 1. Description

In this program, we will demonstrate how to read time by combing



DS3231 clock module with micro:bit. You will view the time on CoolTerm monitor.

## 2. What You Need:

Micro:bit Board\*1

EASY Plug Shield for micro bit V1.1\*1

Micro USB Cable\*1

EASY PlugDS3231 Clock Module\*1

RJ11 Cable\*1

6-Slot AA Battery Holder\*1

1.5V AA Battery\*6

### **EASY Plug DS3231 Clock Module**

The DS3231 is a low-cost, extremely accurate I2C real-time clock (RTC) with an integrated temperature-compensated crystal oscillator (TCXO) and crystal. A real-time clock (RTC) is a computer clock (most often in the form of an integrated circuit) that keeps track of the current time.

At the heart of the module is a low-cost, extremely accurate RTC chip from Maxim – **DS3231**. It manages all timekeeping functions and features a simple two-wire I2C interface which can be easily interfaced with any microcontroller of your choice.

The chip maintains seconds, minutes, hours, day, date, month, and year



information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year (valid up to 2100).

The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator. It also provides two programmable time-of-day alarms and one programmable square wave output.

The DS3231 incorporates a battery input, and maintains accurate timekeeping when main power to the device is interrupted.

The built-in power-sense circuit continuously monitors the status of VCC to detect power failures and automatically switches to the backup supply. So, you need not worry about power outages, your MCU can still keep track of time.

EASY PlugDS3231 module integrates pin GND, VCC, SDA and SCL for easily wiring up and can communicate with MCUs in synchronic and serial way.

SCL is a serial clock pin for I2C interface.

SDA is a serial data pin for I2C interface.

VCC pin supplies power for the module.

GND is a ground pin.

### **3. Specification:**



Temperature range: -40°C to +85°C

Timing accuracy: about  $\pm 5$ ppm

Output: 1Hz and 32.768kHz

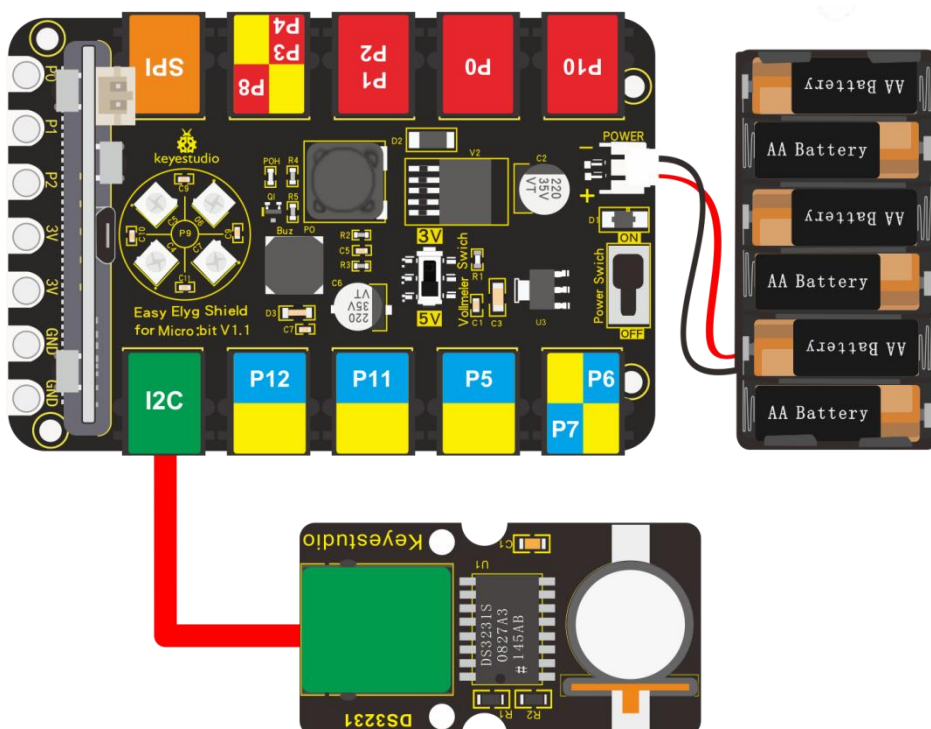
High speed (400kHz), I2C serial bus

Supply voltage: 3.3V to 5.5V

Output Level: TTL level

#### 4.Wiring Up:

Insert micro:bit onto EASY Plug shield, connect DS3231 clock module to I2C port of shield with a RJ11 cable.





Note: Dial Voltmeter\_Switch to 5V end

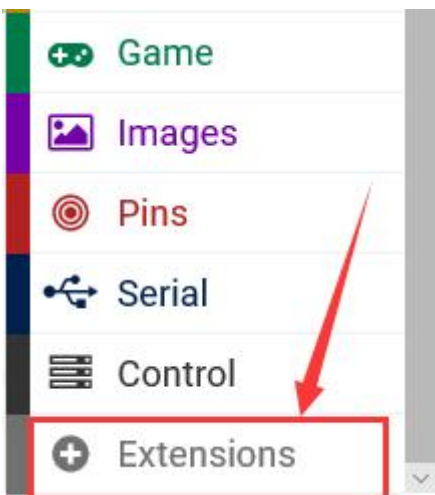
## 5. Test Code:

You could navigate <https://makecode.micro:bit.org/reference> to have access to more details.

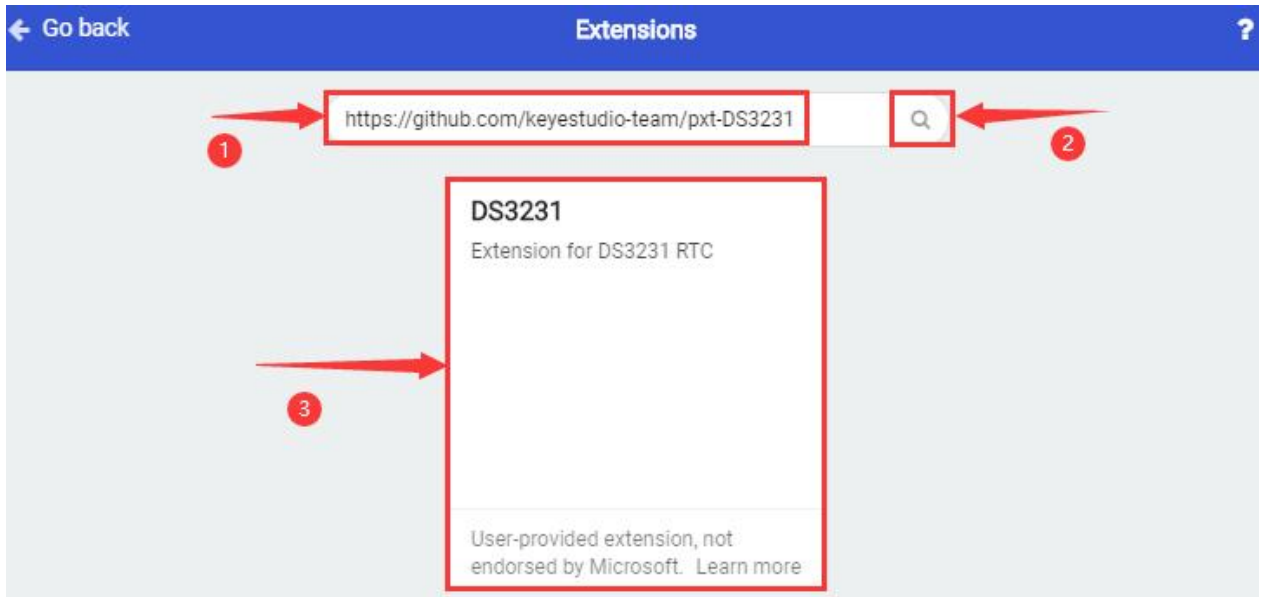
Browse link <https://makecode.micro:bit.org/> to edit your program. The following test code is as for your reference.

Need to set test code, therefore, let' s add the library of DS3231 clock module.

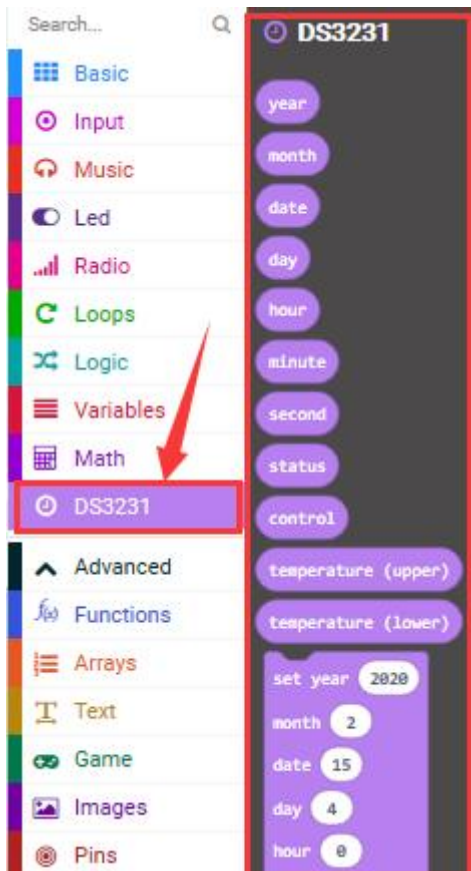
Set code with library file, click "Extensions"



Enter library link <https://github.com/keyestudio-team/pxt-DS3231> and search it, download it as follows:



After the installation, you could see it in the listed blocks on Makecode editor.





## Test Code:

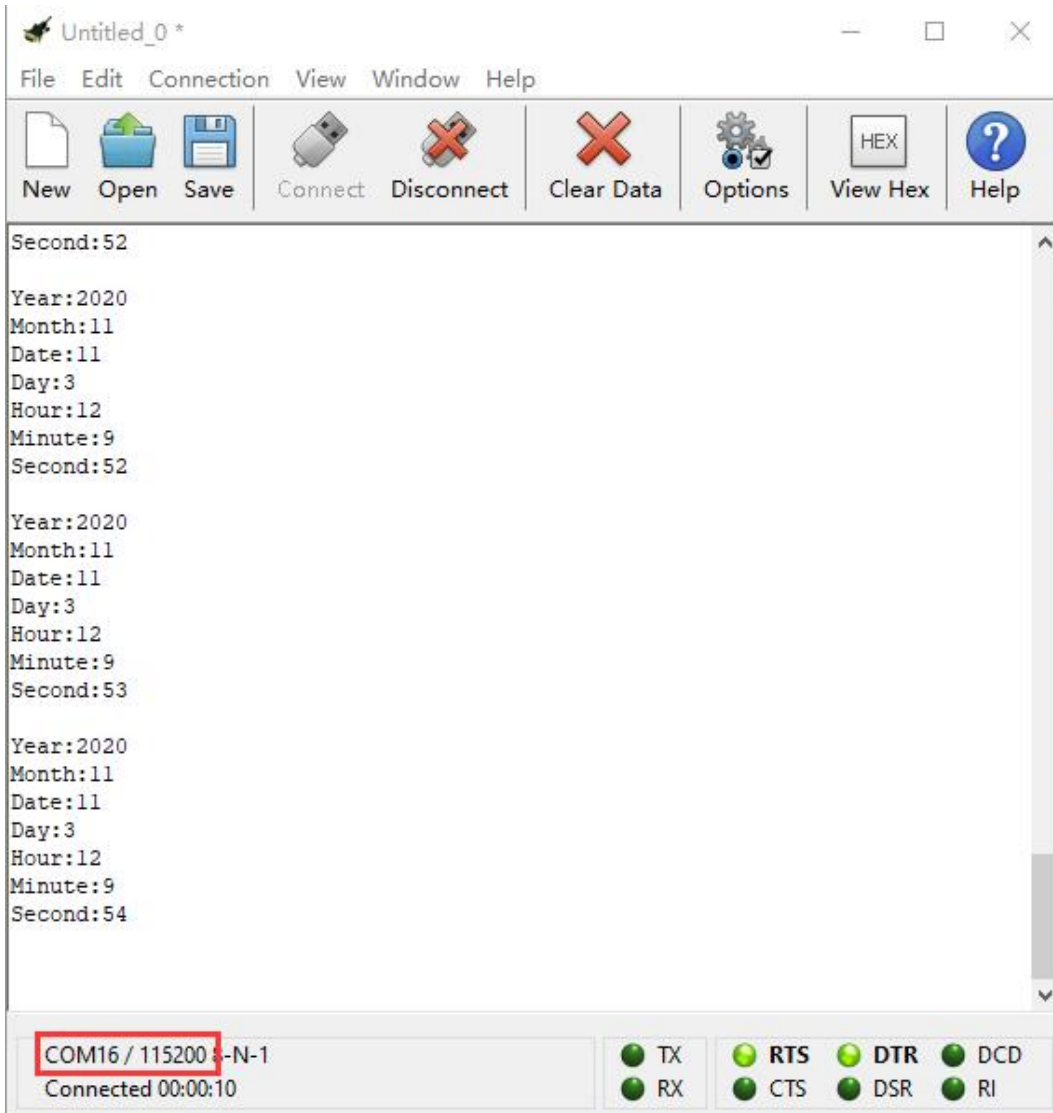
```
on start
  led enable false
forever
  serial write value "Year" = year
  serial write value "Month" = month
  serial write value "Date" = date
  serial write value "Day" = day
  serial write value "Hour" = hour
  serial write value "Minute" = minute
  serial write value "Second" = second
  serial write line ""
  pause (ms) 500
```

## 6.Test Result:

Wiring up, dial Voltmeter\_Switch to 5V end, plug in external power and dial Power\_Switch to ON end and upload code to micro:bit.

Open CoolTerm, click Options and select SerialPort, set COM port and baud rate, set baud rate to 115200. Tap OK and Connect.

CoolTerm monitor will display time(year/month/day/week/hour/minute/second), as shown below:



## 6. Resource:

<https://fs.keyestudio.com/KS4022-4023>