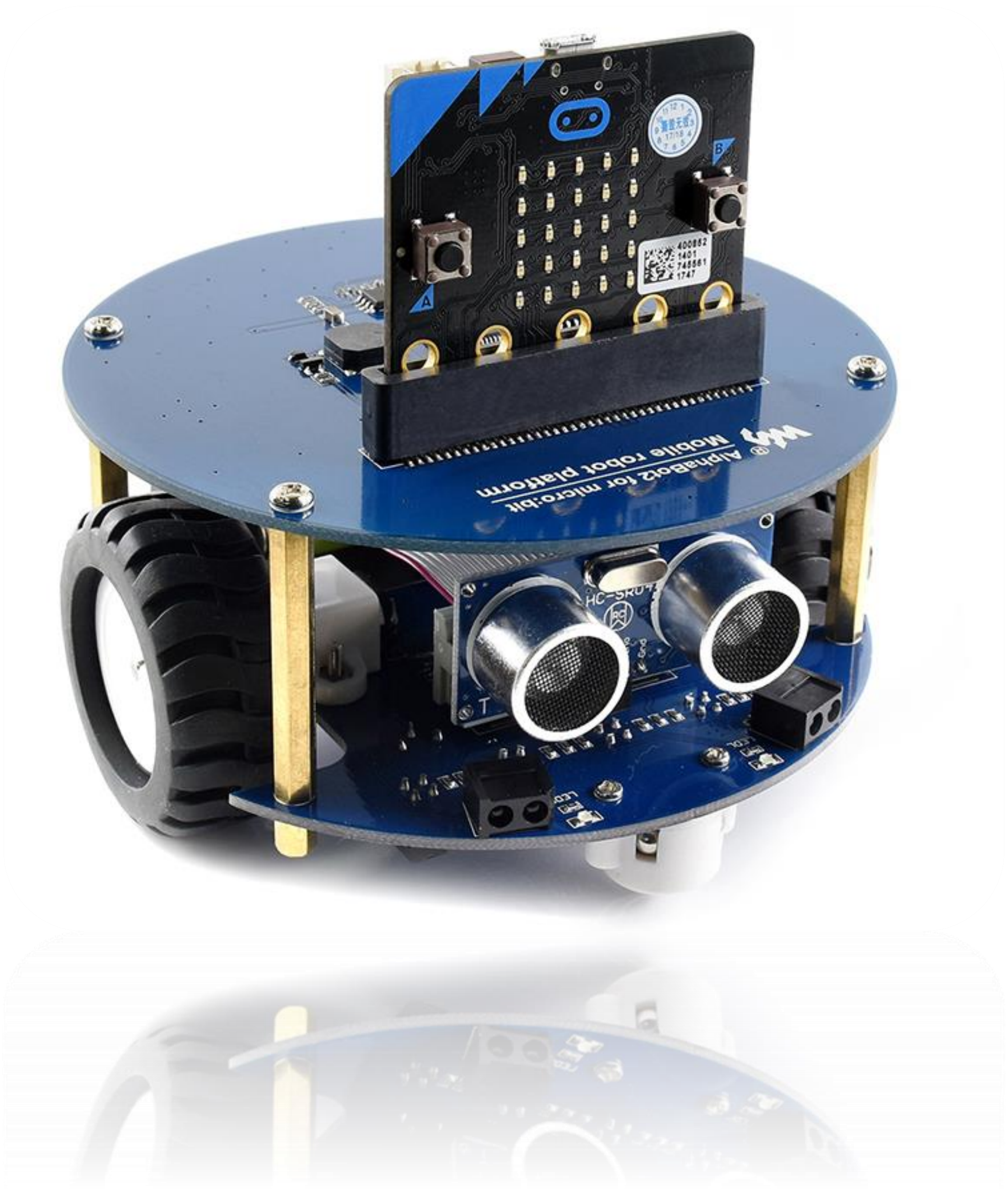
Alphabot2 for micro:bit User Manual



CONTENT

PREFACE

This Alphabot2 robot kit uses the BBC micro: bit as the host controller, combined with several functional modules, it is easy for the kids to experience robotic tricks such as line tracking, obstacle avoiding, ultrasonic ranging, servo operation, Bluetooth remote control, etc.

MICRO:BIT

The BBC micro: bit is a pocket-sized computer to learn programming for kids and beginners

- **Nordic nRF51822**
    - 16 MHz 32-bit ARM Cortex-M0 microcontroller
    - Bluetooth® 4.0 low energy/2.4GHz RF SoC
    - 16kB RAM
    - 256kB Flash
- **Freescale KL26Z** – 48 MHz ARM Cortex-M0+ MCU
- **Compass** – Freescale MAG3110 3-axis magnetometer, I2C interface
- **Accelerometer** – Freescale MMA8652 3-axis accelerometer, I2C interface
- **Power connector** – 3V power supply
- **Reset button** – reset the system
- **Micro USB connector** – for connecting PC, download code, serial communication, etc.

- **5x5 LED display** – 5x5 LED grid
- **Button A/B** – programmable buttons
- **I/O connection pins** – 5 I/O rings and 20-pin edge connector, including SPI, UART, I2C, Analog, PWM, etc.
- **Dimensions** – 5cm x 4cm

NOTES

1. Components in different package will be different. In this guide, we describe all components, and some of them may not be included in the package you buy.
2. The use manual, schematic, demo code and datasheet, all these resources can be downloaded in Wiki.
   - https://www.waveshare.com/wiki/AlphaBot2_for_micro:bit
3. There may be mistakes in this guide due to the limit of time. If you find them, please kindly contact Waveshare Team.[1]
4. To avoid of destroying the Alphabot2, we recommend you read this manual and following it if you are the first time receive your parcel.
5. Please check your parcel when you receive to check if all the components your bought are included. We recommend you to first learn how to use the micro: bit by following Chapter 1 to Chapter 3 if you are the beginner.

[1] Email: service@waveshare.com

6. You must turn on the power switch and make sure the battery supplies power normally before you connect USB cable to micro:bit for programming. You cannot power Alphabot2 by micro:bit, otherwise the power LED's light is very slight and Alphabot2 will work improperly.
7. You must turn off power before you plug/pull micro： bit, to avoid of destroying micro:bit. And please keep the LED matrix of micro:bit towards to outside.
8. All the demo codes mentioned here are provided in Wiki. Demo codes are based on make code programming environment of Microsoft. The makecode tool supports Graphic programming and JavaScript. The demo codes are all hex files, can be copied to micro:bit directly and run. (The micro:bit is recognized as a potable drive when connected to PC. Flashing demo code to micro:bit by copied hex file to this drive). You can also import demo code to miscode website for modifying and flashing.
9. All specifications and demo codes supplied herein are subject to change without notice at any time.

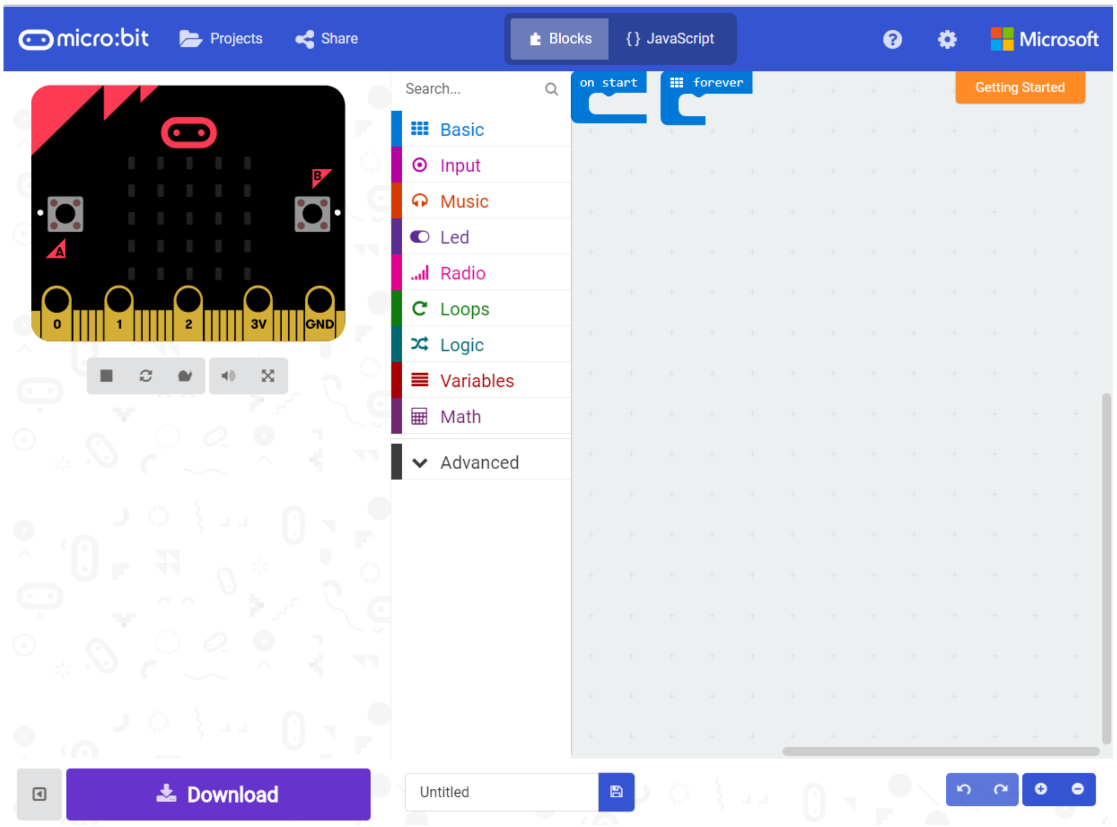
CHAPTER 1 LED LIGHTS

The micro:bit has 25 (5x5) individually-programmable LEDs, allowing you to display text, numbers and images, here we will learn about coding these LEDs.

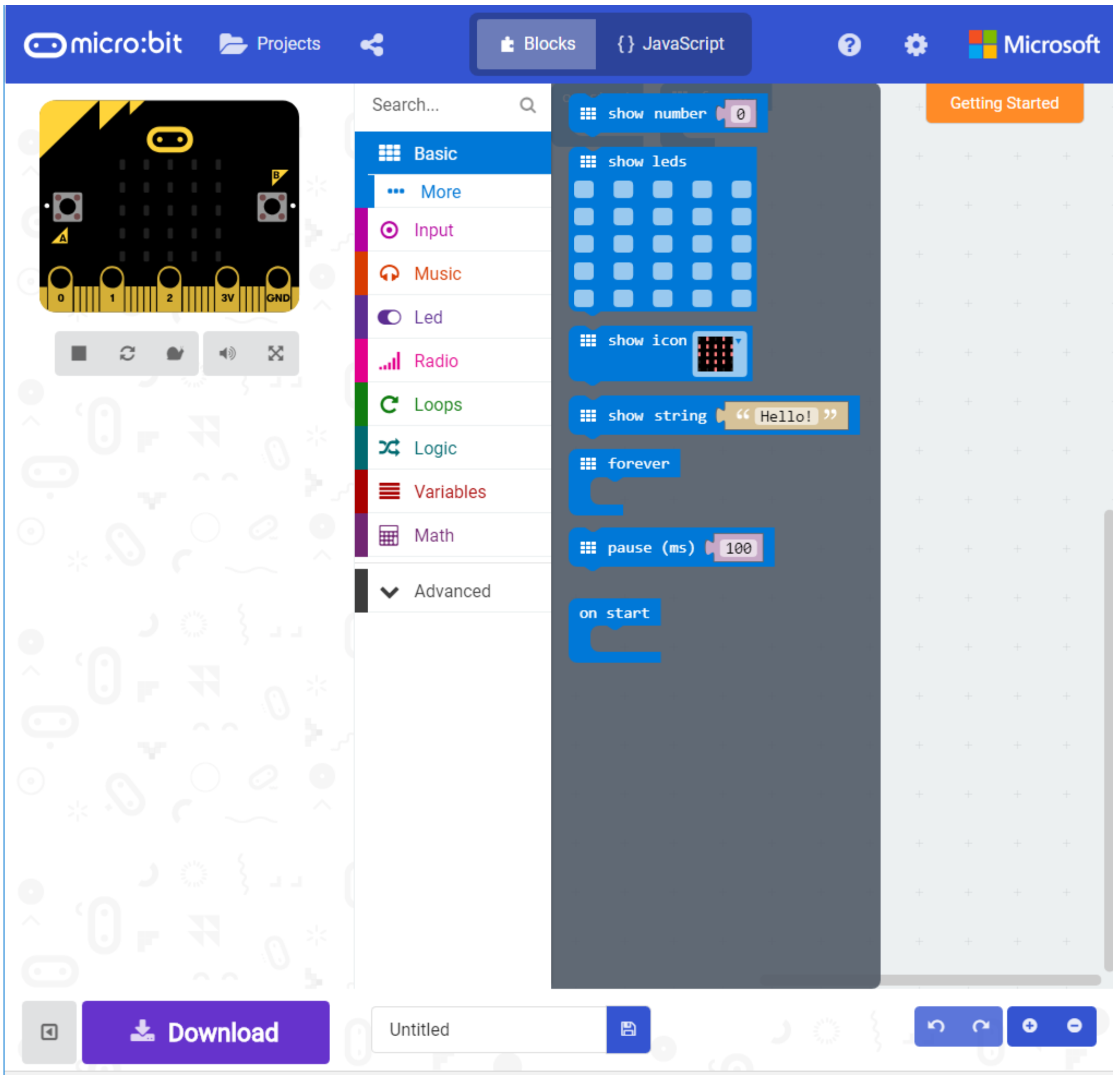
JAVASCRIPT BLOCKS EDITOR

The link of online JavaScript Block Editor of BBC: https://makecode.microbit.org/#

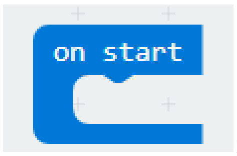
Accessing the editor. There is divided to four area, the upper area is menu bar, center area is Blocks, left is simulating demonstration area and the right is Blocks and JavaScript editing area.



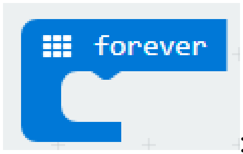
BLOCKS



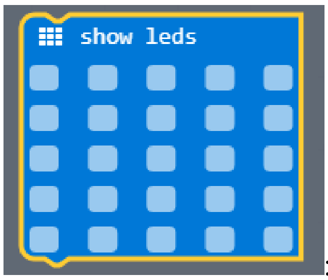
Blocks in **Basic** are mainly used for basic operation like LEDs controlling.



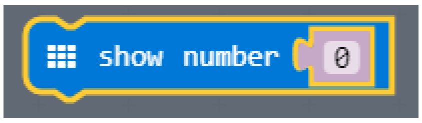
: Run the code at the beginning



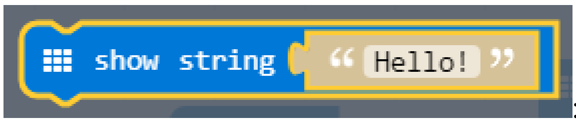
: Repeat the code forever at the background



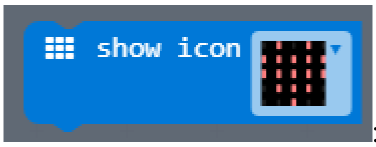
: Draw an image on the LED screen (LEDs)



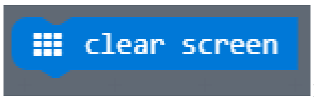
: Scroll a number on the LEDs



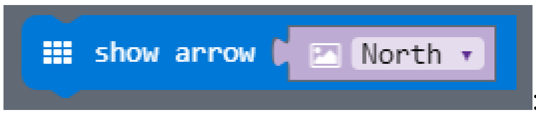
: Display text on the LEDs, display one character at a time.



: Draw selected icon on the LEDs



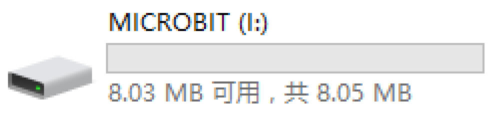
: Turn off all LEDs



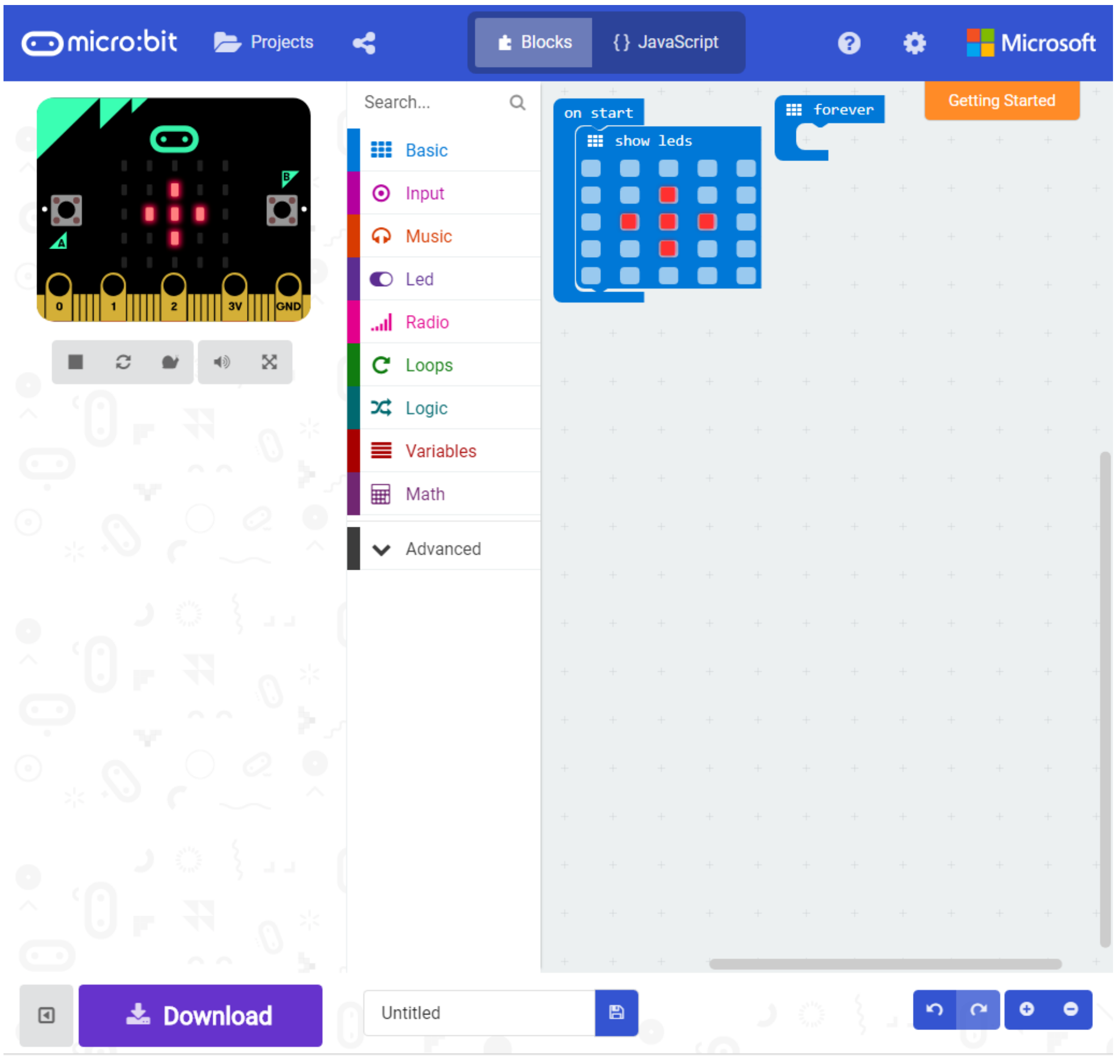
: Draw an arrow on the LEDs

LIGHTING LEDS

Connecting the micro:bit to computer via a micro USB cable, the micro:bit will show up on your computer as a drive called "MICROBIT".

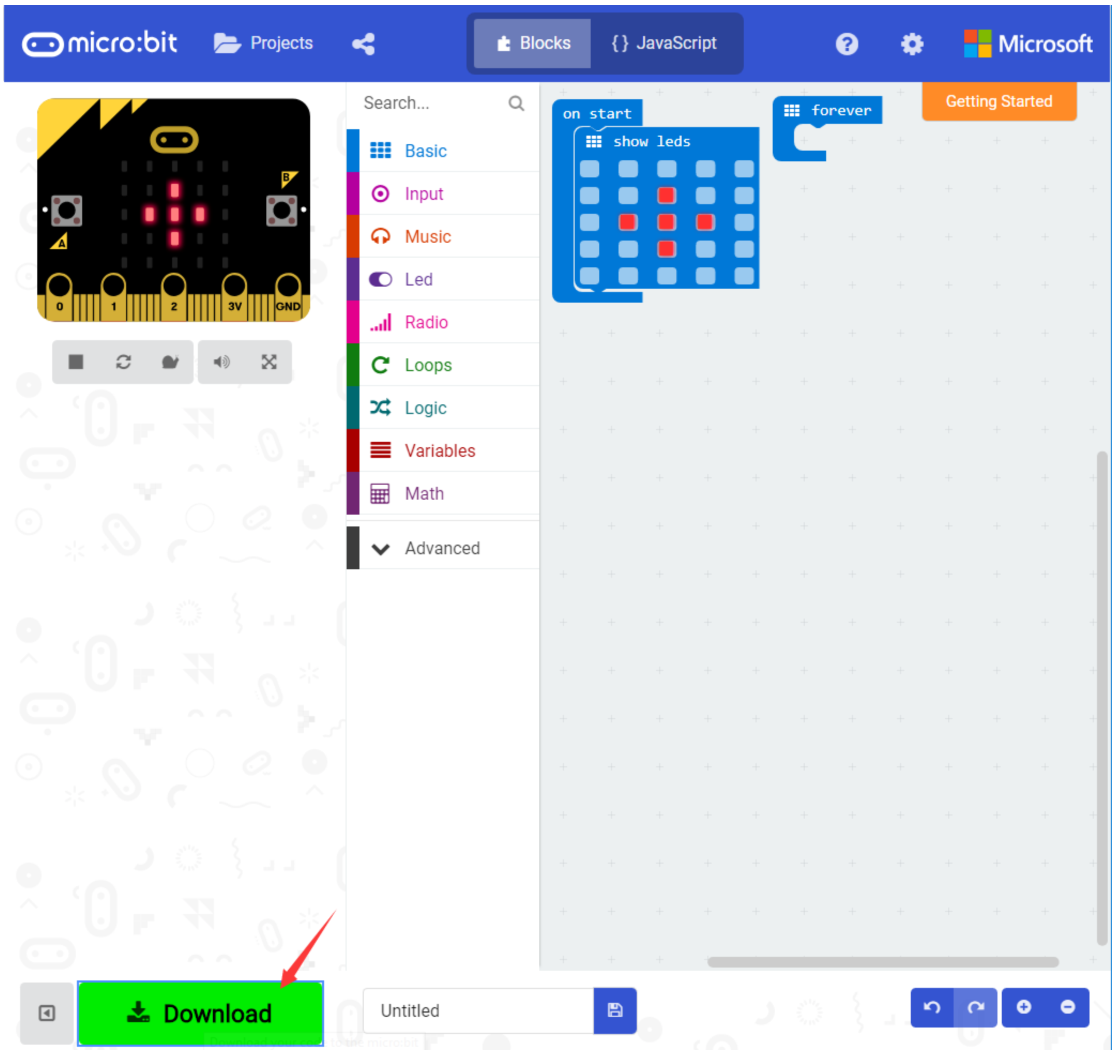


Now let' s start our first code that to light the LEDs with the blocks we describe above. Just as below:



Click the LED to make them light on, you can see that the LEDs of simulated micro:bit light on as the codes.

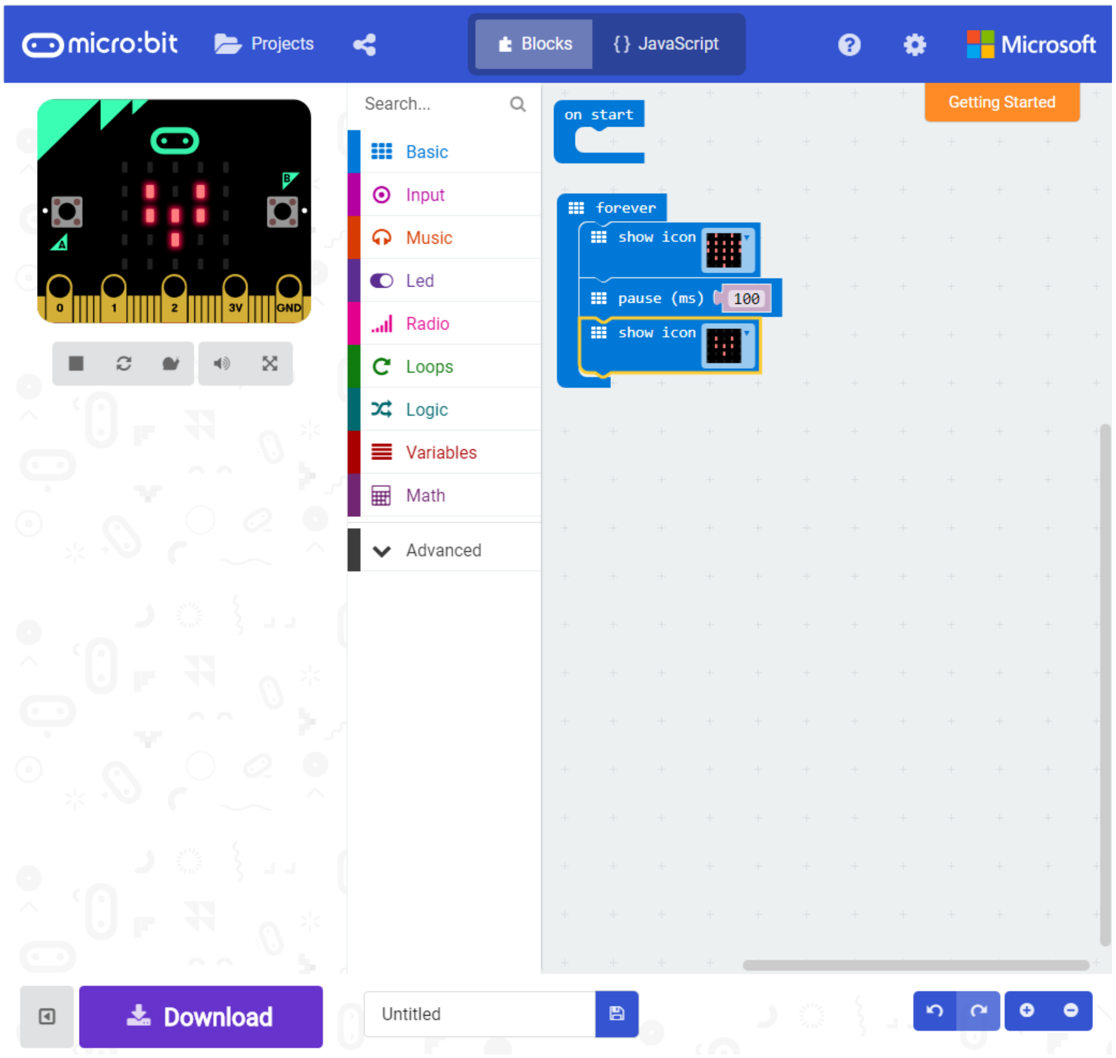
Click Download to download the code to the drive of micro:bit. You can also save the HEX file to another place, and then copy to micro:bit instead of directly download to micro:bit.


micro:bit
Projects
Blocks
{ } JavaScript
Microsoft
Search...
Basic
Input
Music
Led
Radio
Loops
Logic
Variables
Math
Advanced
on start
show leds
forever
Getting Started
Download
Untitled

HEART BEATING

This code, we use block show icon, first display big heart icon then the small heart.

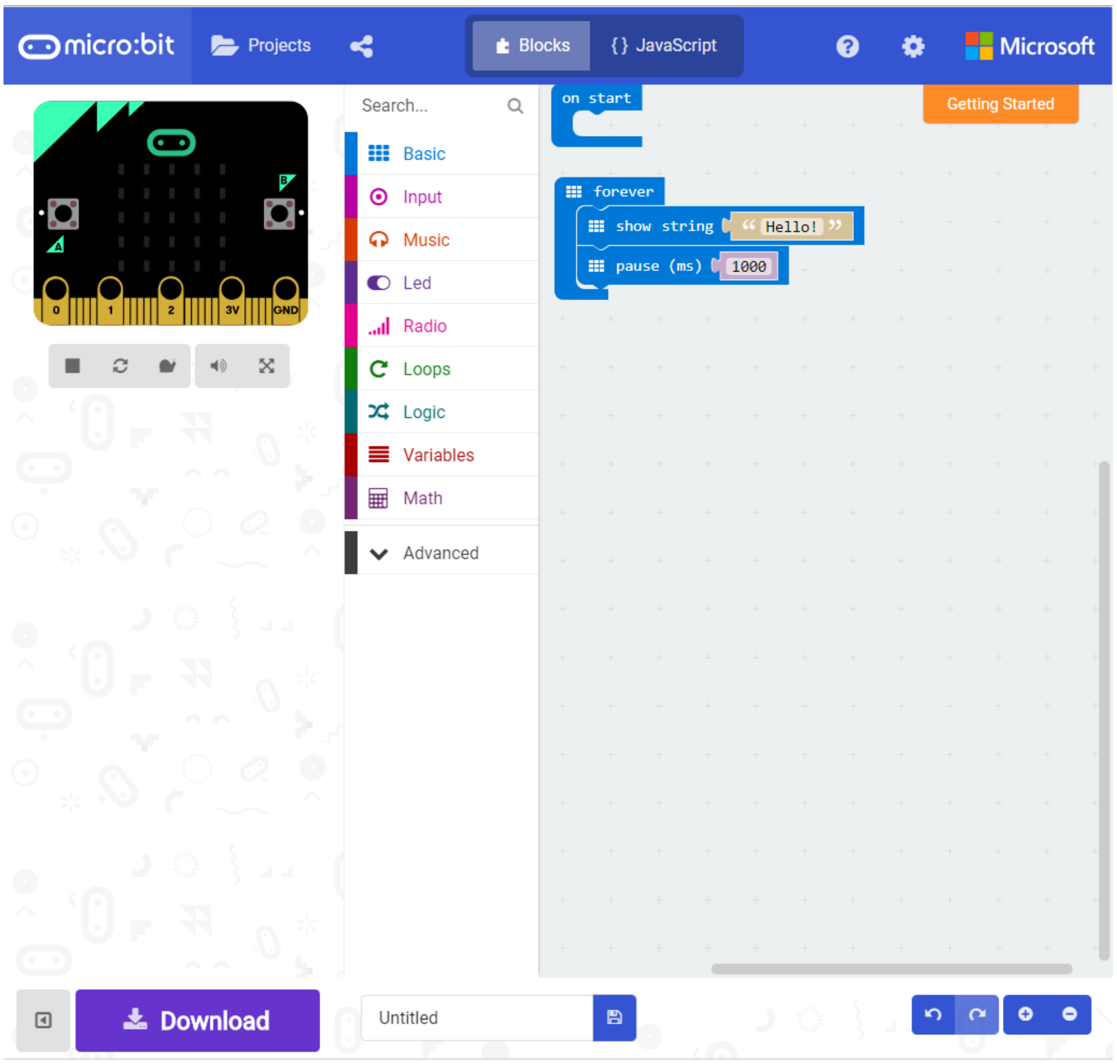
Repeat the code and it looks like heart beating. Let’ s do it.



```
on start

forever
    show icon
    pause (ms) 100
    show icon
```

DISPLAY TEXT

Here we use show string block to display text "Hello!" on LEDs. The text will scroll on LEDs character by character.



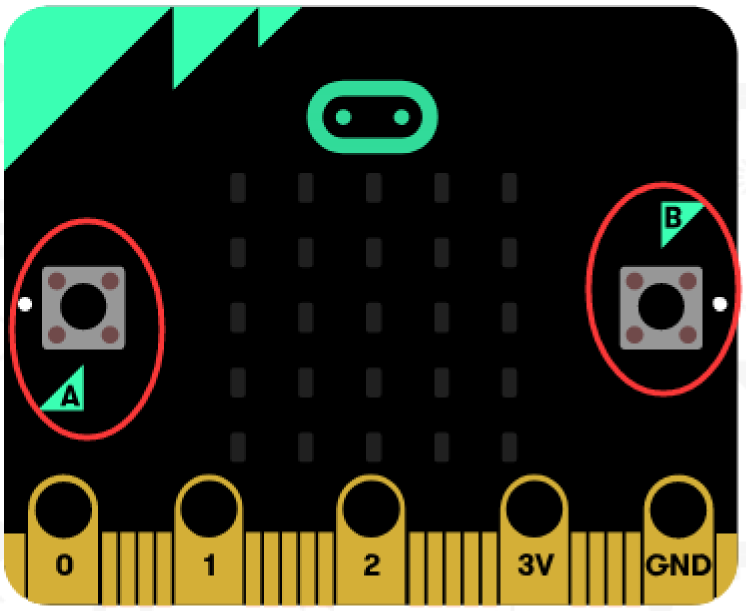
```
on start

forever
    show string "Hello!"
    pause (ms) 1000
```

【Tips】

To delete the blocks, you can right click and delete it or just drag it to the left side.

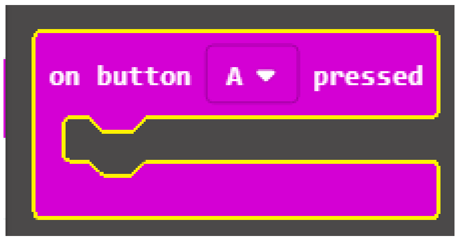
CHAPTER 2 BUTTONS

There are two programmable buttons on the front of the micro:bit, which are labelled A and B. In this chapter, we will code with these buttons.



PRESSING

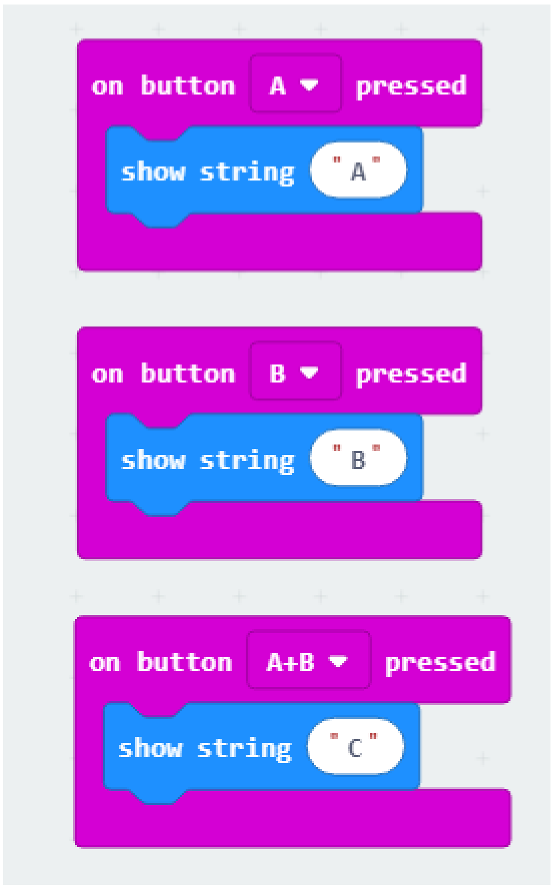
Blocks we use in this section are included in **Input** package. With this block, if A or B buttons on your micro:bit are pressed, codes are executed.



Coding the buttons to display characters, A when button A is pressed, B when button B is pressed, and C when both buttons are pressed.

1. We create a new project, place on button pressed block to run codes when button A, button B or buttons A+B are pressed separately.

2. Place show string block in on button pressed block to display string on LEDs display. Duplicate and complete the script to as below:



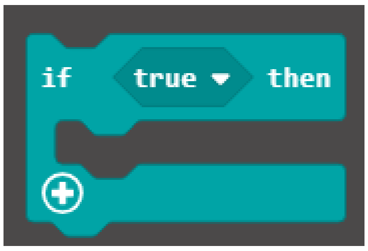
```
on button A ▾ pressed
    show string "A"

on button B ▾ pressed
    show string "B"

on button A+B ▾ pressed
    show string "C"
```

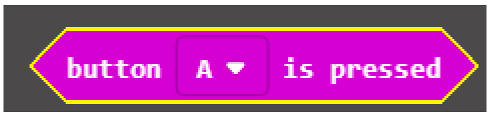
3. Connect your micro:bit to PC, click Download to flash the code to your board. Press buttons on micro:bit to test if expected string is displayed.

PRESSING 2

There is another way to detect buttons. In last section, codes are run when button is pressed, here we use if then and button is pressed blocks to do the same thing. Instead of running code directly, a Boolean value[2] "true" is returned if button is pressed, "false" if button is not be pressed.

[2] Boolean value: True or False

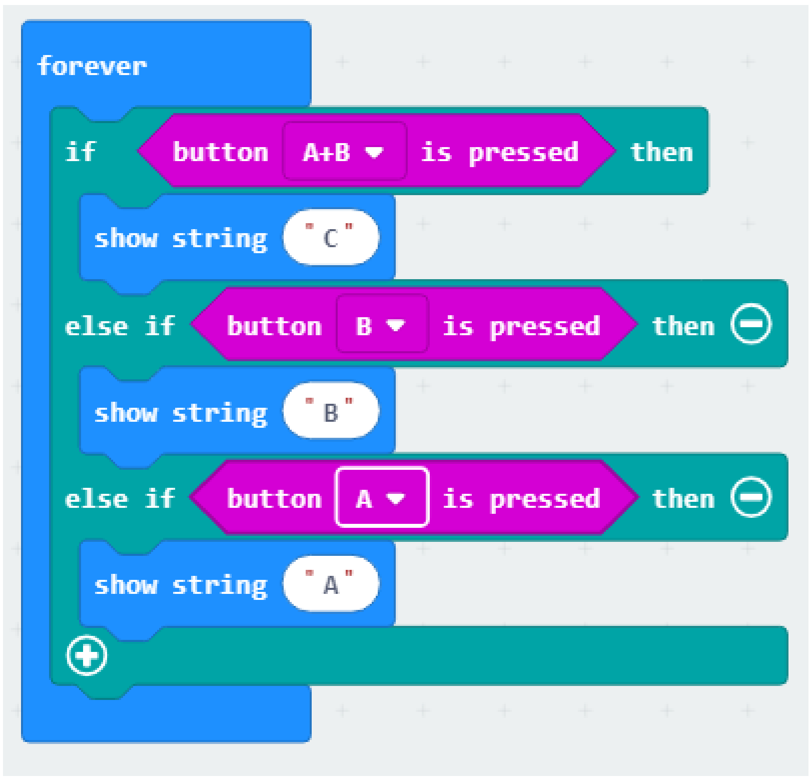




if then block is in **Logic** package, button is pressed block is in **Input** package.

1. Place if then block on forever, then put button is pressed block on it.

2. Place show string block, then clock ⊕ icon to complete the codes as below:



```
forever
    if button A+B is pressed then
        show string "C"
    else if button B is pressed then
        show string "B"
    else if button A is pressed then
        show string "A"
```

3. Connect your micro:bit to PC, click Download to flash the code to your board.

   Press buttons on micro:bit to test if expected string is displayed.

【Tips】

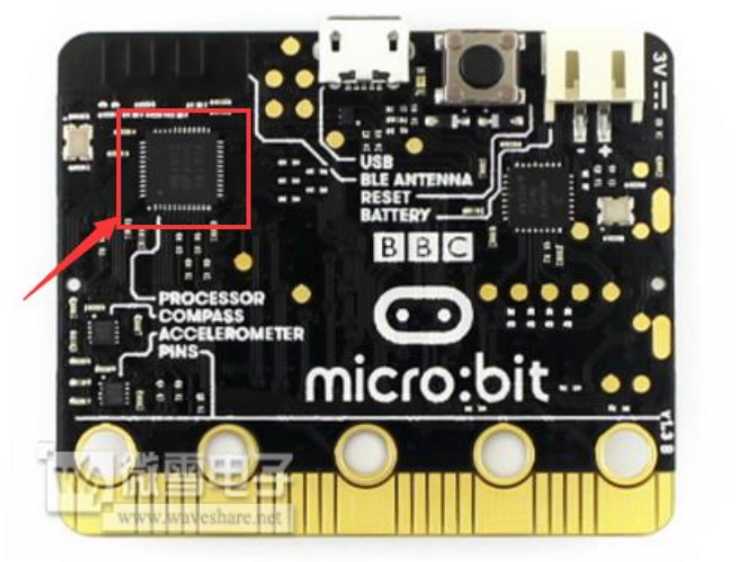
Some users may notice that in this script, A+B are detected first instead of A or B. You can try to change the detect order to see what will happen.

CHAPTER 3 SENSORS

Micro:bit integrates Temperature sensor, Accelerometer, Compass and so on. In this chapter, we learn to use these sensors.

TEMPERATURE SENSOR

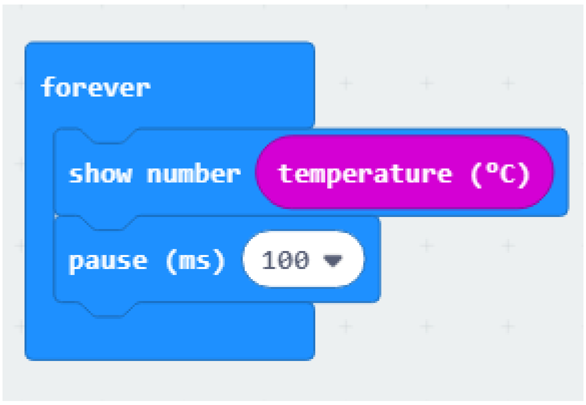
This sensor allows the micro:bit to detect the current ambient temperature in degree Celsius. It is embedded in the main chip nrf51822, which is used to detect the temperature of chip in fact. Even temperature of chip is a little different with ambient temperature, but we can use it to get an approximate value.



In **Input** package, temperature (°C) block is there. Combine with show number block, we can show the ambient temperature in your room on LEDs display.

1. Place show number block on forever.
2. Place the temperature (°C) blocks

3. Add one pause (ms) for delay.

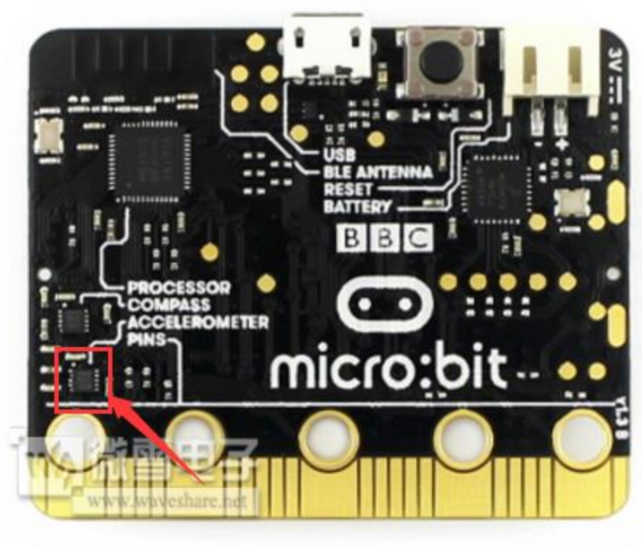


```
forever
    show number  temperature (°C)
    pause (ms)  100
```

4. Download the code to your micro:bit, touch the CPU (the main chip) for a moment, you will find the temperature became higher.

ACCELEROMETER

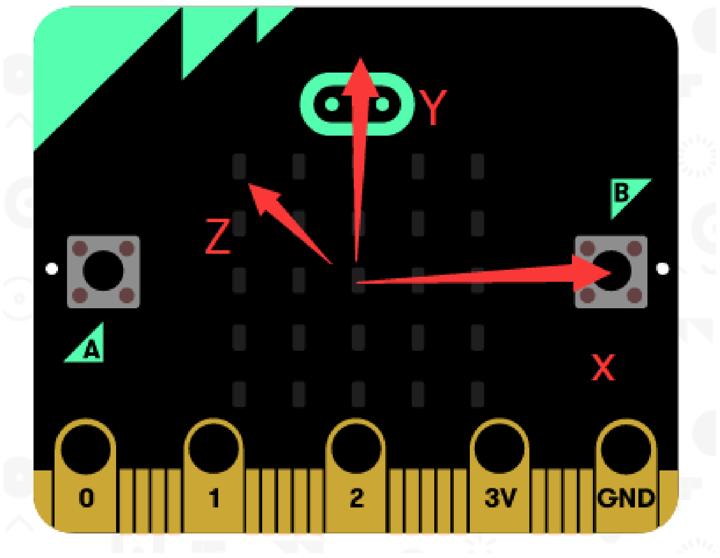
The accelerometer MMA8652 is integrated on the left bottom of micro:bit



.

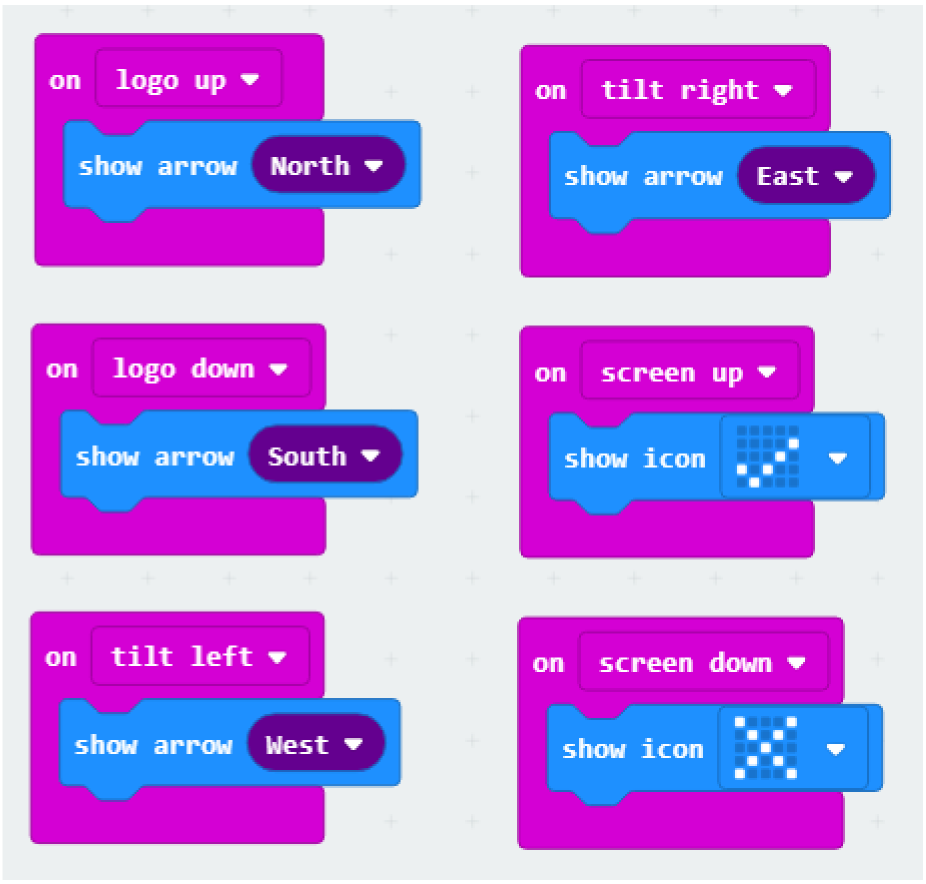
Acceleration is a trend to move. If we apply a force to an object in one direction, then the object will trend to move in the direction, we call this trend the acceleration. For example, we are being applied gravity all the time, if there is not any holder under our feet, we will free-fall. It is that we have a trend to falling, which is called the acceleration of gravity written as g in formulas.

We measure coordinate of x, y and z directions. With these three values, we could get the direction of accelerations. If there is only acceleration of gravity, the direction is downward. We keep measuring values of x, y and z coordinates to get the acceleration of micro:bit, and detect actions like shake, tilt and so on



1. Place on blocks, choose the action
2. Place show arrow block choose the arrow

3. Duplicate and complete the script

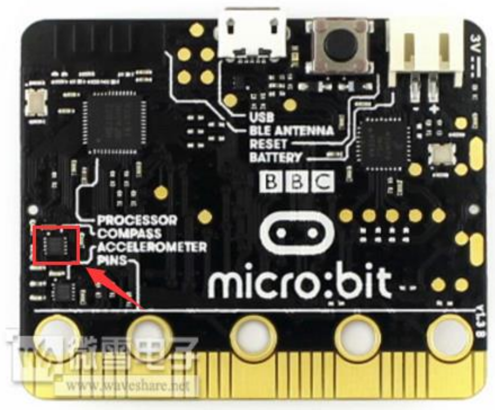


```
on logo up
    show arrow North

on tilt right
    show arrow East

on logo down
    show arrow South

on screen up
    show icon

on tilt left
    show arrow West

on screen down
    show icon
```

4. Download the code to your micro:bit, shake your micro:bit to see if arrows change.

MAGNETOMETER

The compass chip is MAG3110, marked by red frame. The compass can detect the earth’ s magnetic field by measuring x, y and z coordinates as well.

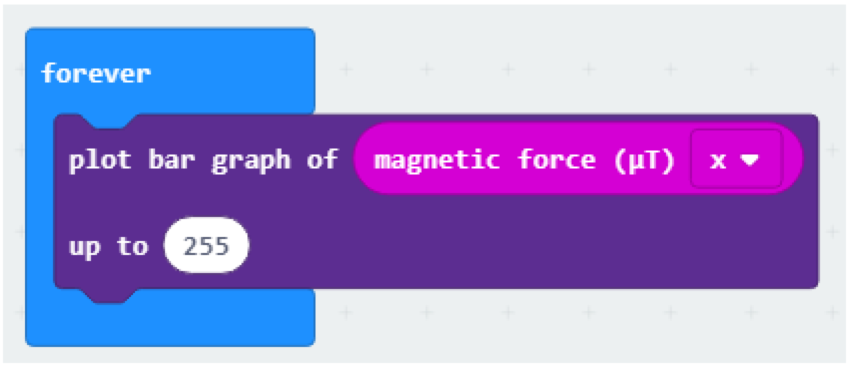


Magnetic Force:

We create a script to show the intensity of magnetic force by bar graph (LEDs)

1. Place plot bar graph of block to forever, which is in **Led** package
2. Place magnetic force (uT) block on the bar block, which can be found on

   Input->More, than set the max value to 255



```
forever
    plot bar graph of  magnetic force (µT)  x ▾
    up to  255
```

3. Download the script to your micro:bit
4. When you first time run the code on micro:bit, it will scroll text “to fill screen” to tell you that calibration step should be done first. Then you should to fill the LED screen by tilting the micro:bit. After lighting all the LEDs, a smiley is displayed to show you that calibration is finished.
5. The expected result is that a few of LEDs are lighted if there are not any magnetic things, and if you close a speaker[3] to the micro:bit, more LEDs will be lighted

Compass

With accelerometer, we can get the direction of gravity, which is downward.

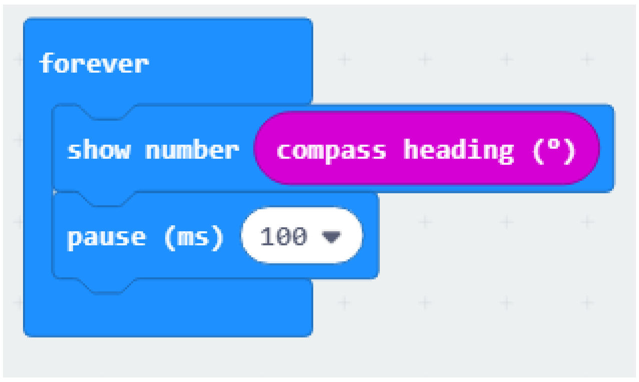
Magnetometer is influenced by earth’ s magnetic field, horizontal component of

[3] Most of the speaker have magnet, like speaker of mobile phone

geomagnetism always points to north of geomagnetism. Combine with accelerometer and magnetometer, we can get the direction of North.

【Note】 we you use compass, you should keep your micro:bit be static and cannot be influenced by other magnetic fields.

1. Place show number block to forever
2. Find block magnetic heading (°) to number block
3. Add pause block for delay



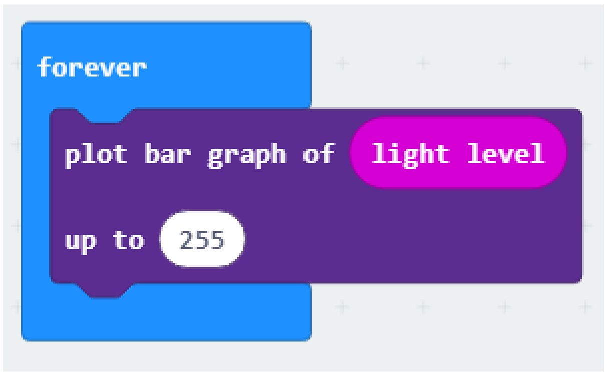
4. Download the script to your micro:bit and try it.
5. Calibration also need to do after downloading.

LIGHT SENSOR

Micro:bit has no actual light sensor, however, it could sense ambient light by reversing its LEDs matrix to become an input and sampling decay time of voltage

1. Place plot bar graph of block to forever

2. Drag light level block to bar block, set max value to 255



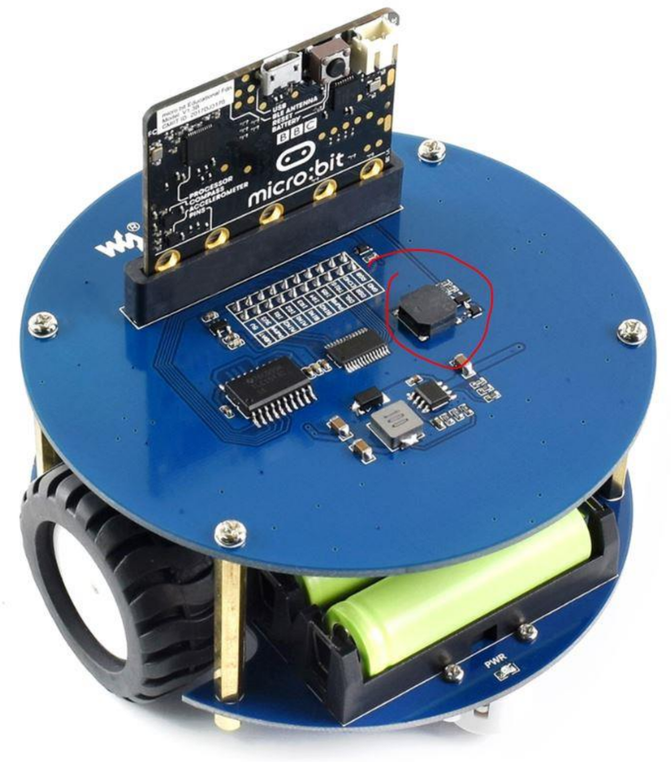
3. Download the script to your micro:bit and try

4. Open the flashlight of your mobile phone and try to light the LEDs with it.

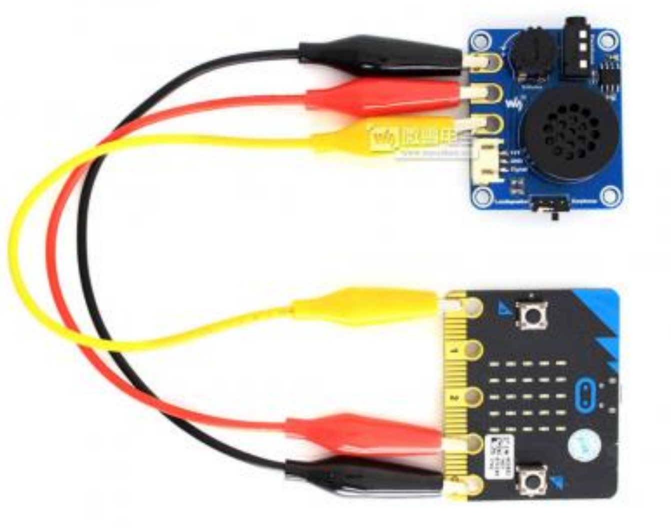
CHAPTER 4 MUSIC

Micro:bit doesn’ t have buzzer or speaker, so if you want to make your micro:bit to play sound or music, you need to connect external speaker. There is a buzzer on Alphabot2-for-micro:bit expansion board, you can insert the micro:bit to it.

【Note】 Before you read this chapter, you’ d better to assemble the Alphabot2 for mciro:bit.



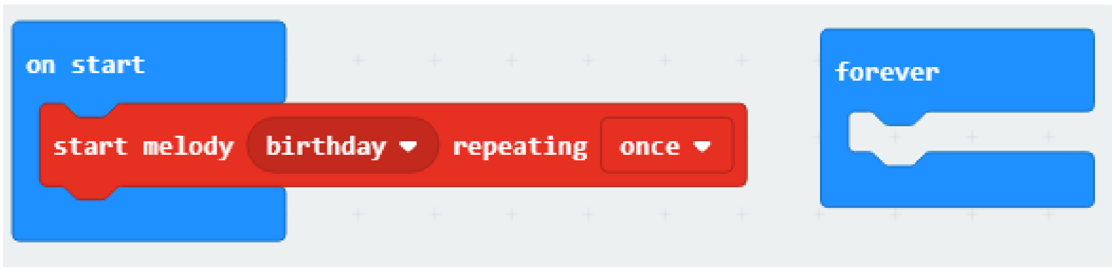
You can also connect a buzzer/speaker to micro:bit through P0 as below:



PLAY A MELODY

Blocks for music can be found on **Music** package.

1. Place start melody repeat block to on start or forever
2. Choose one melody like birthday and set it to repeat once.



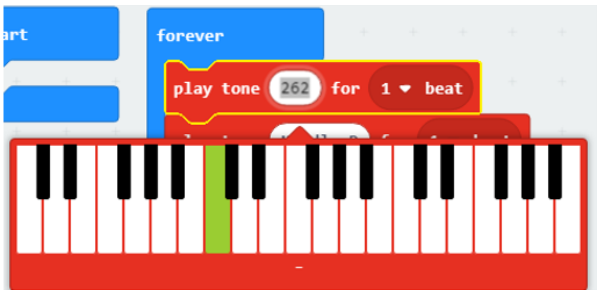
3. Download the script to your micro:bit and test if the melody is played.

【Tips】

sound is generated by shaking of object. People speak by shaking our vocal cord in different frequency. Buzzer works in the same way, it generates different tones when getting high/low level with different frequency from control board. You can change the tone for different sounds.

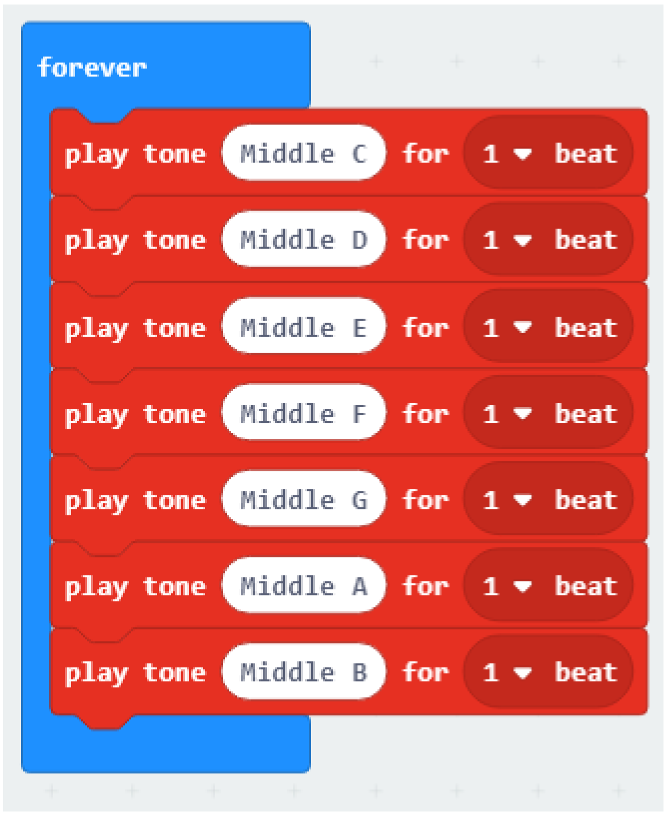
PLAY PIANO

Let’ s use blocks to simulate piano.



1. Place play tone for block to forever, set the tone to Middle C and 1 beat

2. Duplicate and complete the script as below



```
forever
    play tone Middle C for 1 beat
    play tone Middle D for 1 beat
    play tone Middle E for 1 beat
    play tone Middle F for 1 beat
    play tone Middle G for 1 beat
    play tone Middle A for 1 beat
    play tone Middle B for 1 beat
```

3. Download the script to your micro:bit, speaker connected will repeat tones

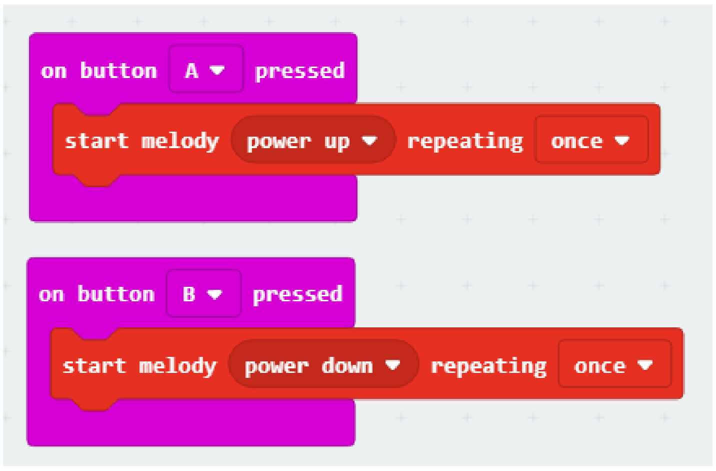
“do” , “re” , “mi” , “fa” , “sol” , “la” , “si” .

PLAY MELODY 2

We create a new project, to play different melody when buttons are pressed.

1. Place on button pressed block and set the start melody repeating on it

2. Duplicate and complete the script as below

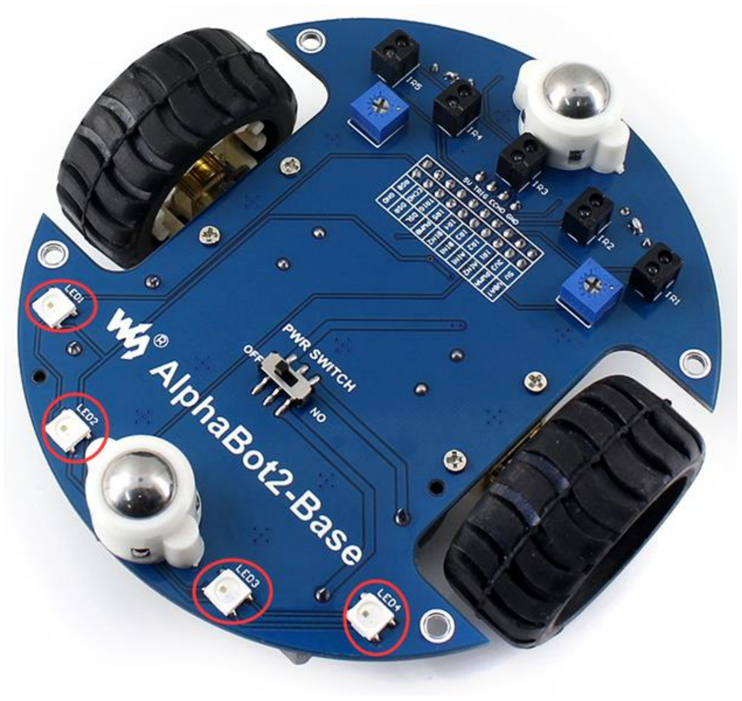


```
on button A ▾ pressed
    start melody power up ▾ repeating once ▾

on button B ▾ pressed
    start melody power down ▾ repeating once ▾
```

3. Download the script to your micro:bit, try to press the buttons

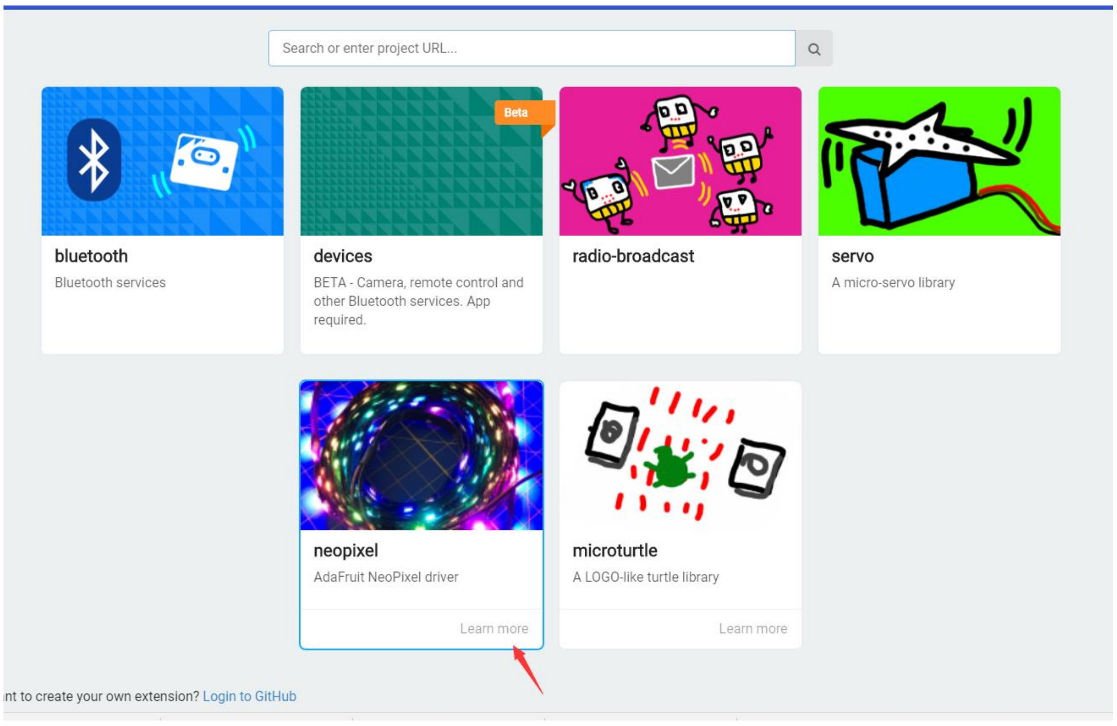
CHAPTER 5 RGB LED

Alphabot2 has four full color RGB LEDs, which are programmable for colorful effect. In this chapter, we code the RGB LED.

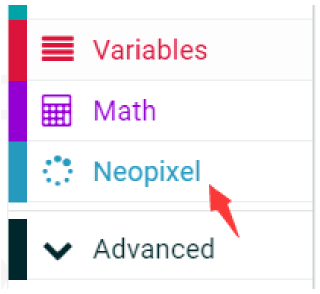


ADD PACKAGE

Before coding, we should add Neopixel package, click More... icon on top right, choose Add package (Extensions), then click neopixel.


Search or enter project URL...
bluetooth
Bluetooth services
devices
Beta
BETA - Camera, remote control and other Bluetooth services. App required.
radio-broadcast
servo
A micro-servo library
neopixel
AdaFruit NeoPixel driver
Learn more
microturtle
A LOGO-like turtle library
Learn more
nt to create your own extension? Login to GitHub

After adding, a blocks package Neopixel will be added



LIGHTING

We create a new project, and code to light all the RGB LED

1. Place Set item to NeoPixel at pin P8 with 4 leds as (GRB format) block to on start

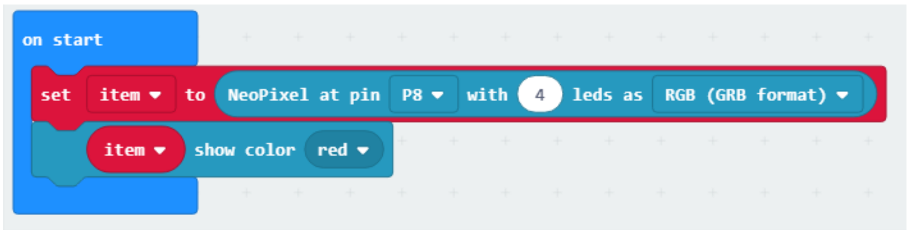
   -Variable item can be renamed

   -Pin P8 is selectable and P8 is the default pin of Alpahbot2' s RGB LED

2. Place item show color red

   -item should be same as the last block

   -color red is changeable



```
on start
    set item to NeoPixel at pin P8 with 4 leds as RGB (GRB format)
    item show color red
```

3. Download the script to micro:bit to test

DISPLAY DIFFERENT COLORS

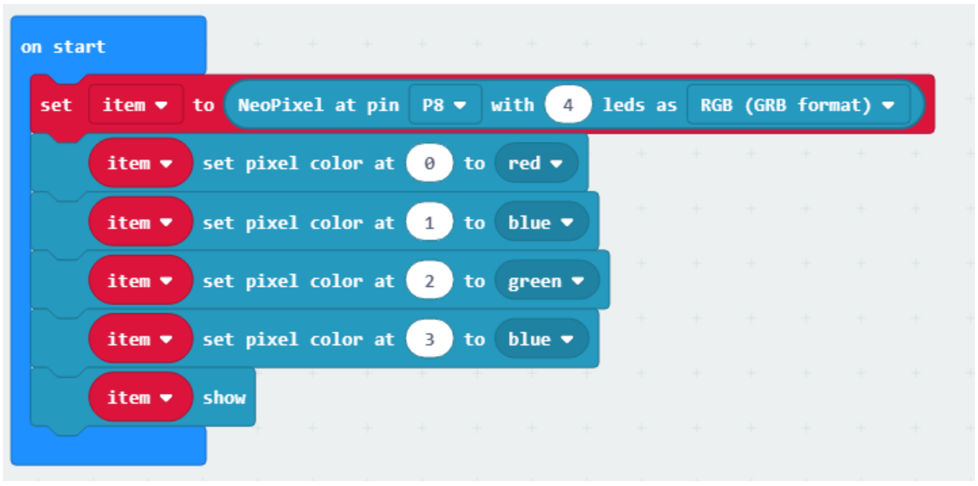
We modify the last script to make four RGB LEDs to display different colors, for example, red, yellow, green and blue

1. Delete block item show color red by dragging it to the left side
2. Place item set pixel color at 0 to red block

   -This block can be found in **Neopixel->More**

   -This block can set the color of every pixel from 0
3. Duplicate and complete four LEDs
4. Place item show block to the end

   - This block is very import, it finally shows the setting color to RGB LED, without it, the settings you done before are all useless.



```
on start
    set item to NeoPixel at pin P8 with 4 leds as RGB (GRB format)
    item set pixel color at 0 to red
    item set pixel color at 1 to blue
    item set pixel color at 2 to green
    item set pixel color at 3 to blue
    item show
```

5. Download the script to micro:bit for testing

BLINKY LEDS

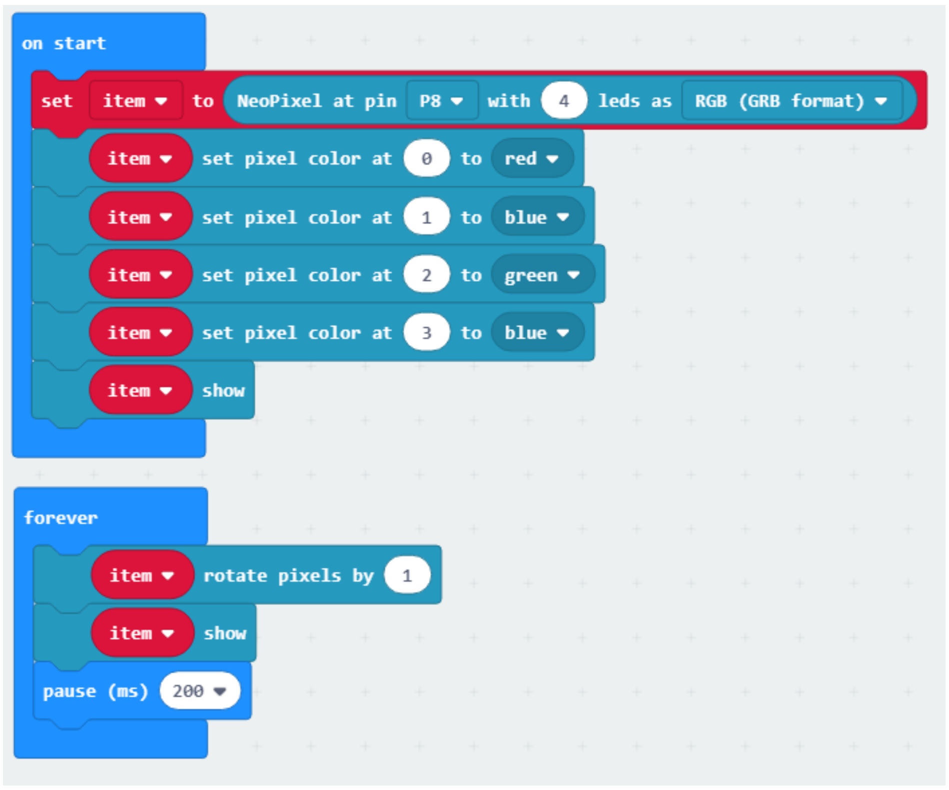
We modify the last script to cycling RGB LED display.

1. Place item rotate pixels by 1 block in forever

   -This block can transfer the color of current pixel to next one

2. Place item show and pause (ms) blocks

   - item show block is necessary



```
on start
    set item to NeoPixel at pin P8 with 4 leds as RGB (GRB format)
    item set pixel color at 0 to red
    item set pixel color at 1 to blue
    item set pixel color at 2 to green
    item set pixel color at 3 to blue
    item show

forever
    item rotate pixels by 1
    item show
    pause (ms) 200
```

3. Download the script to micro:bit and test

RAINBOW

We create a new project, coding four RGB LEDs for rainbow effect.

1. Place Set item to NeoPixel at pin P8 with 4 leds as (GRB format) block in on start
2. Create a variable hue, place set hue to 0

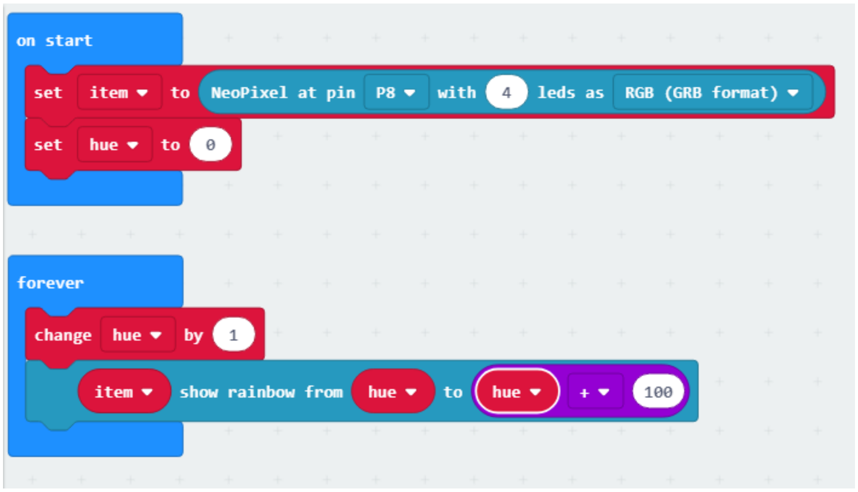
   - these blocks can be created and founded in **Variables** package

3. Place change hue to 1 block in forever
4. Place item show rainbow from 1 to 360

   - the range of hue is 1~360, different hue values stand for different colors, this block will show average hue (1~360) to 4 RGB LEDs. More the RGB LEDs, more colors are showed like rainbow.

5. Use 0 + 0 and hue block to complete the script

   - 0 + 0 block can be found on **Math** package

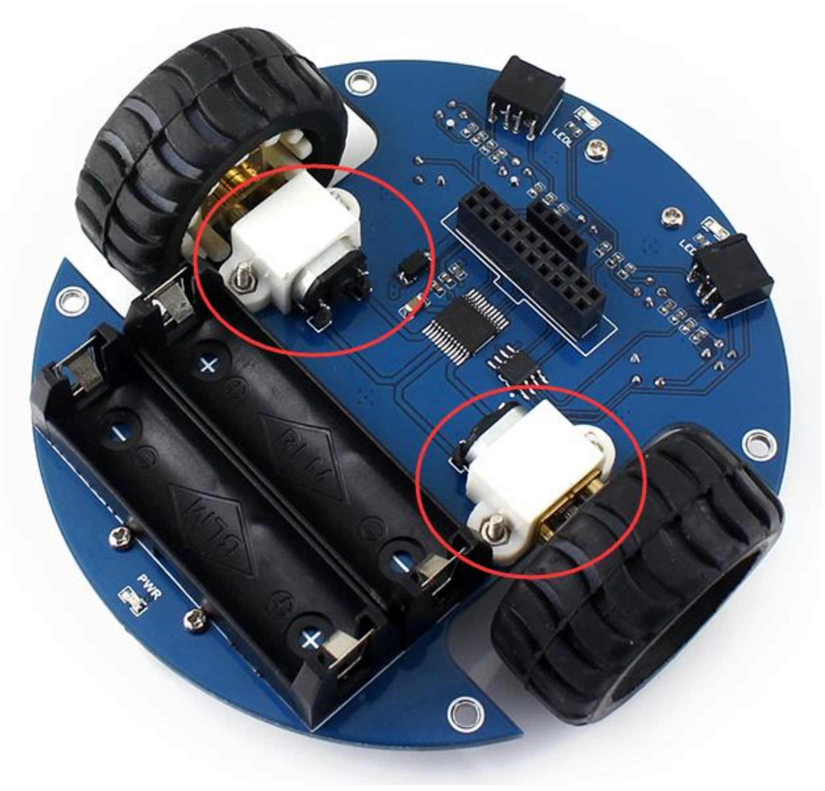


```
on start
    set item to NeoPixel at pin P8 with 4 leds as RGB (GRB format)
    set hue to 0

forever
    change hue by 1
    item show rainbow from hue to hue + 100
```

6. Download the script to micro:bit and test

CHAPTER 6 MOVING, ROBOT

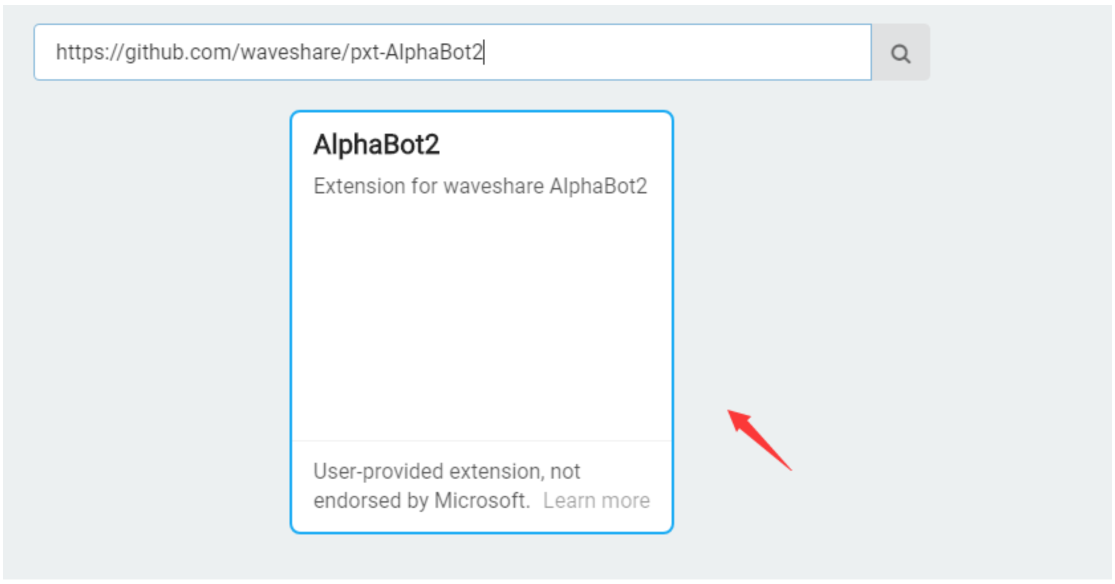
We have learned how to control LEDs and buzzer. In this chapter, let’ s control the Alphabot2 and let it move.



ADD PAKCAGE

Adding a package like what we do in last chapter. Input the address on box:

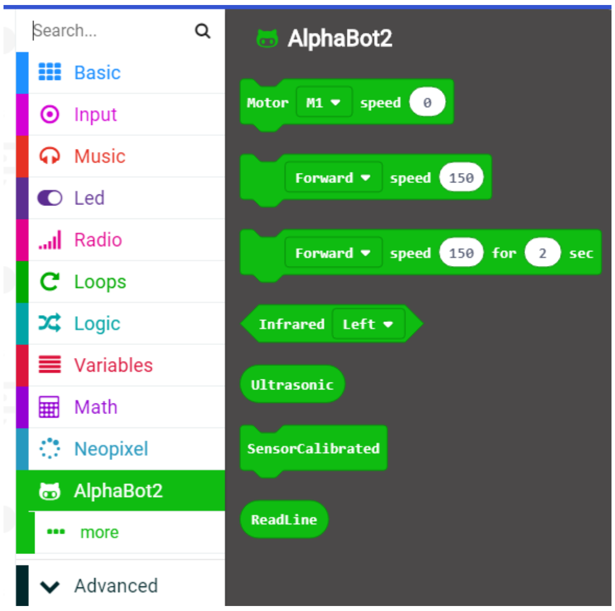
https://github.com/waveshare/pxt-AlphaBot2 , Click AlphaBot2.

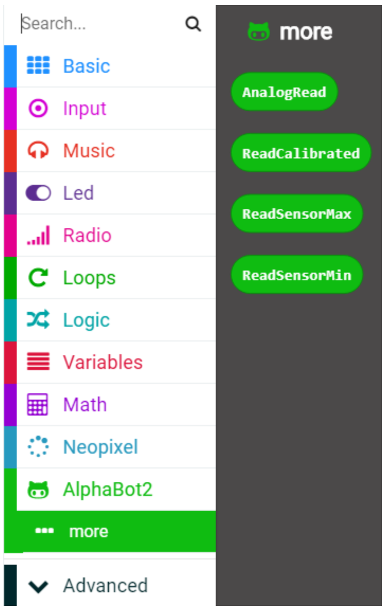

https://github.com/waveshare/pxt-AlphaBot2

AlphaBot2

Extension for waveshare AlphaBot2

User-provided extension, not endorsed by Microsoft. Learn more



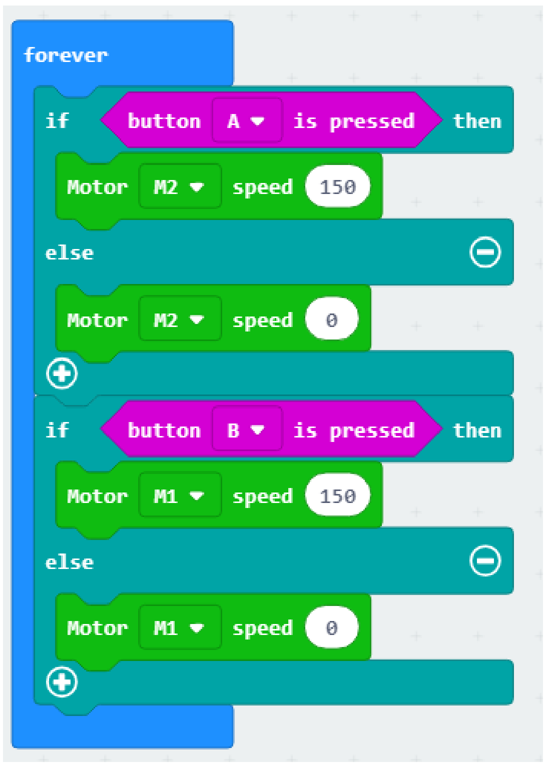


MOTORS

We create a new project to control the motors of Alphabot2 by buttons.

1. Place If then and button is pressed blocks in forever
2. Place Motor speed block after then

   - This block is used to control one motor, the range of speed is -255 to 255, bigger the number, fast the speed. Positive to forward, and negative to backward.
3. Duplicate and complete the script



```
forever
  if button A is pressed then
    Motor M2 speed 150
  else
    Motor M2 speed 0
  if button B is pressed then
    Motor M1 speed 150
  else
    Motor M1 speed 0
```

4. Download the script to micro:bit and test
5. Expected result is that: press A, motor M1 moving and stop when release, press B, motor M2 moving and stop when release

MOVING

We create a new project. Let Alphabot2 moves forward, backward, turns left and turns right by coding.

1. Place Motor speed block in on start

   - If both motors move forward, robot will move forward

   - If both motors move backward, robot will move backward

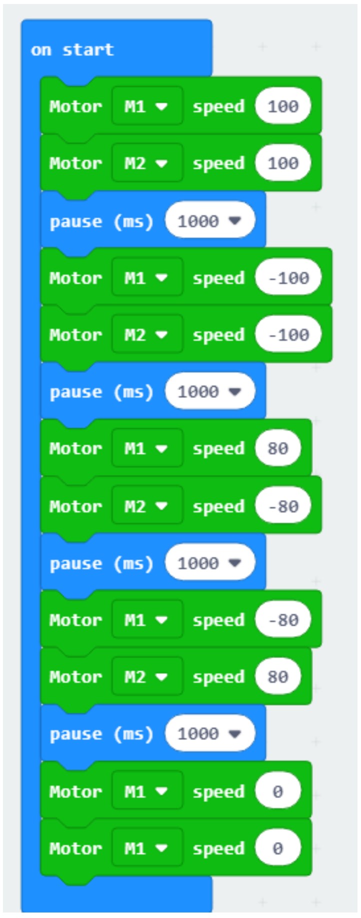
   - If left motor moves backward or stop, and right motor moves forward, robot will turn left (generally, if the speed of two motors are different, robot will make a turn)

   -If left motor moves forward, and right motor moves backward or stop, robot will turn right (generally, if the speed of two motors are different, robot will make a turn)

   -If both motors stop, robot stops.

2. Place pause (ms) for delays

3. Duplicate and complete the script as below



```
on start
  Motor M1 speed 100
  Motor M2 speed 100
  pause (ms) 1000
  Motor M1 speed -100
  Motor M2 speed -100
  pause (ms) 1000
  Motor M1 speed 80
  Motor M2 speed -80
  pause (ms) 1000
  Motor M1 speed -80
  Motor M2 speed 80
  pause (ms) 1000
  Motor M1 speed 0
  Motor M1 speed 0
```

4. Download the script to micro:bit and test

MOVING 2

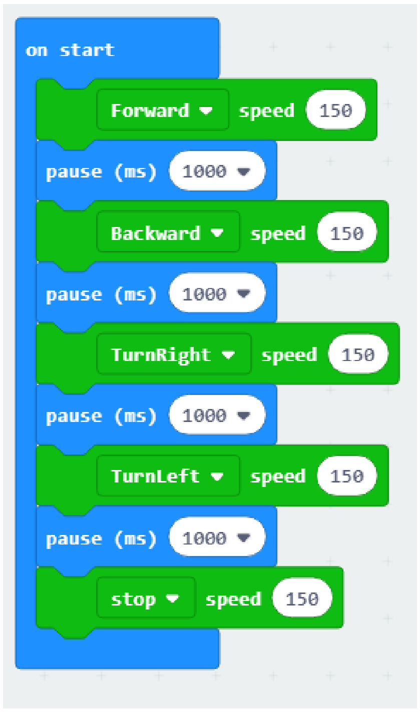
There is another block can also be used to control motor.

1. Place Forward speed 0 block in on start

   - This block can directly control the robot to move in all directions

2. Place pause (ms) block

3. Duplicate and complete the script as below



```
on start
  Forward ▾ speed 150
  pause (ms) 1000 ▾
  Backward ▾ speed 150
  pause (ms) 1000 ▾
  TurnRight ▾ speed 150
  pause (ms) 1000 ▾
  TurnLeft ▾ speed 150
  pause (ms) 1000 ▾
  stop ▾ speed 150
```

4. Download the script to micro:bit and test

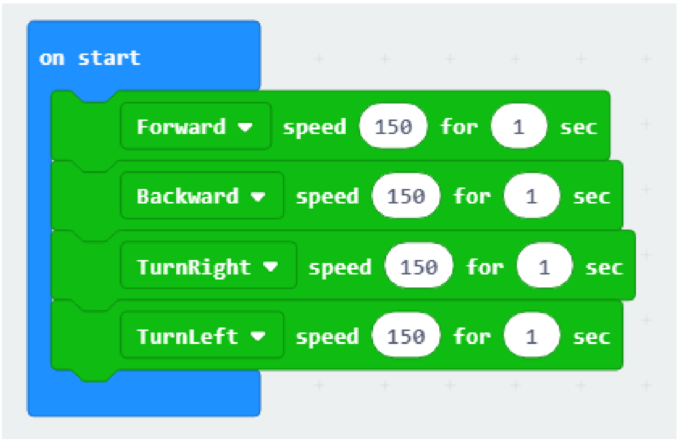
MOVING 3

There is another block which can control robot to move for a certain time.

1. Place Forward speed 150 for 2 sec block in on start

   - With this block, robot will move forward in speed 100 for 2 seconds then stop if there is not another setting

2. Duplicate and complete the script



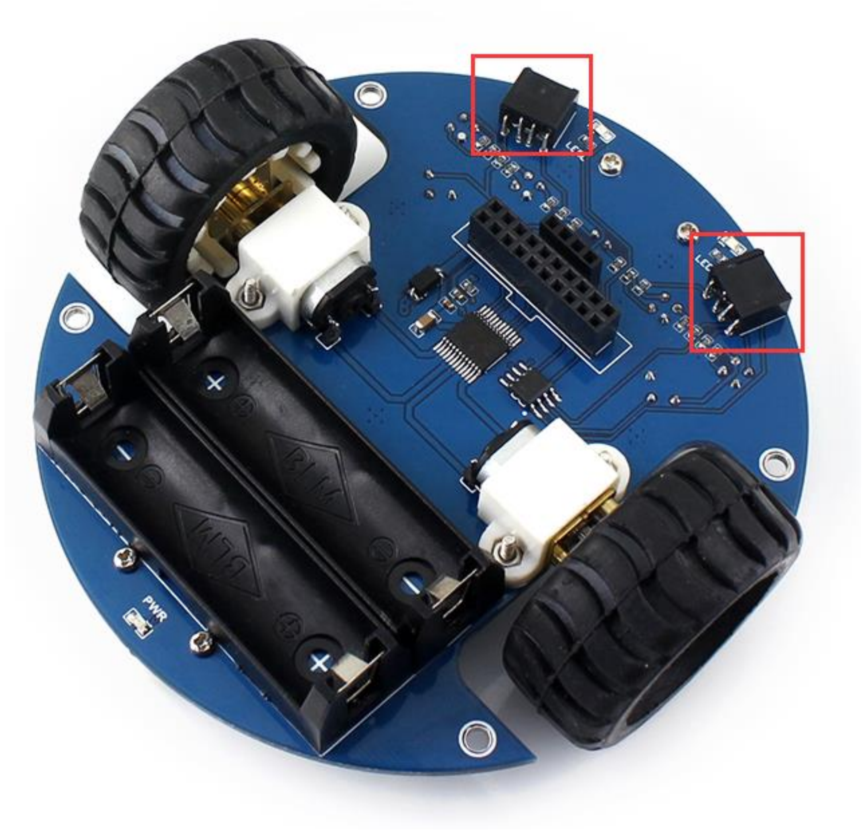
```
on start
    Forward ▾ speed 150 for 1 sec
    Backward ▾ speed 150 for 1 sec
    TurnRight ▾ speed 150 for 1 sec
    TurnLeft ▾ speed 150 for 1 sec
```

3. Download the script to micro:bit and test

4. After downloading, robot will move forward for 1 second, then backward for 1 second, and turn right, turn left, finally stop.

CHAPTER 7 INFRARED OBSTACLE AVOIDING

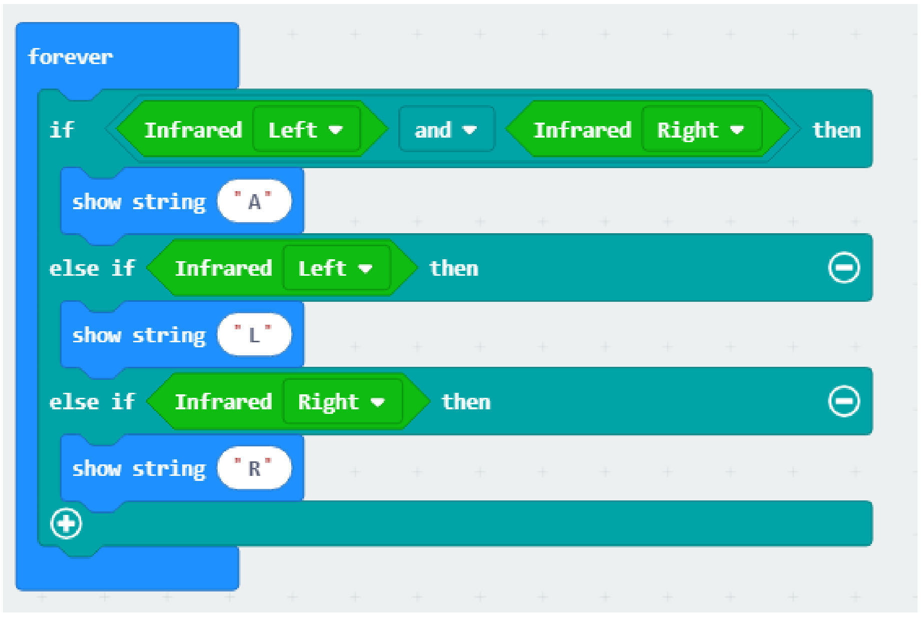
Two infrared sensors are integrated in the front of the Alphabot2-Base board, if you have assembled the Alphabot2, they are under the Ultrasonic sensor.



Every infrared sensor has one sender and one receiver, infrared lights are sent from the sender, if there is obstacle in the front, some of them will be reflected and received by the receiver. There are green LEDs beside infrared sensors to indicate obstacles.

INFARED SENSOR STATUS

1. Place if then block in forever
2. Replace true with and block
3. Put Infrared block to both sides of and
4. Place show string block
5. Duplicate and complete the scrip



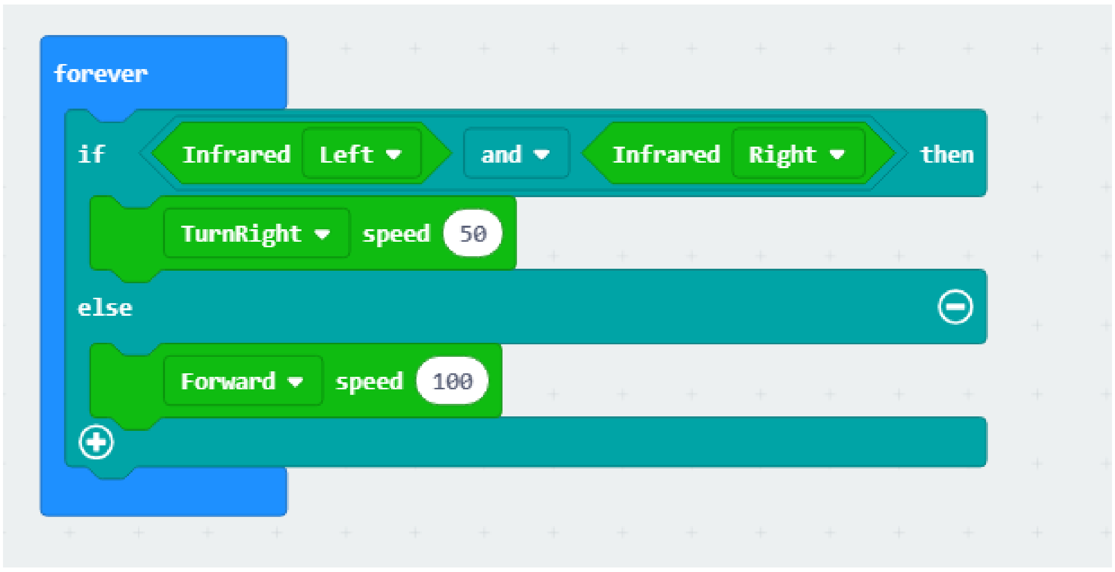
```
forever
  if Infrared Left and Infrared Right then
    show string "A"
  else if Infrared Left then
    show string "L"
  else if Infrared Right then
    show string "R"
```

6. Download the script to micro:bit and test it.
7. After downloading, LEDs of micro:bit will display "A" if obstacle is detected by both infrared sensors, "L" if obstacle is detected by the left infrared sensor and "R" if obstacle is detected by the right sensor.

OBSTACLE AVOIDING

We modify the last script to make robot to avoid obstacle.

1. Place show string block with speed block to make robot turn right when obstacle is detected.
2. Place pause (ms) block
3. Place Forward speed block after else to let robot move forward



```
forever
  if Infrared Left and Infrared Right then
    TurnRight speed 50
  else
    Forward speed 100
```

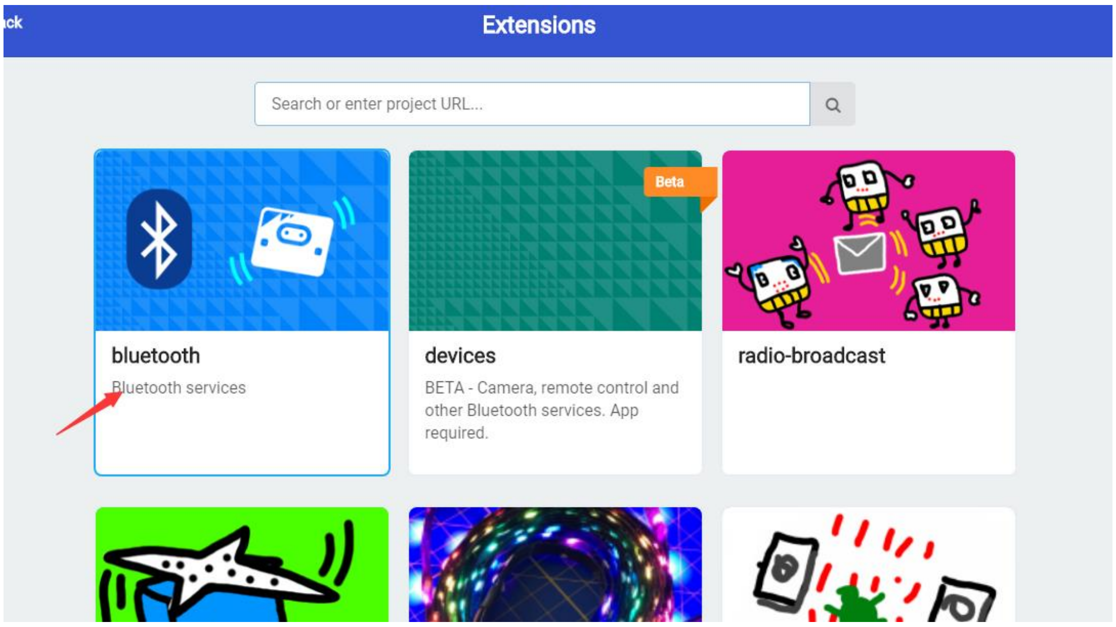
4. Download the script to micro:bit and test

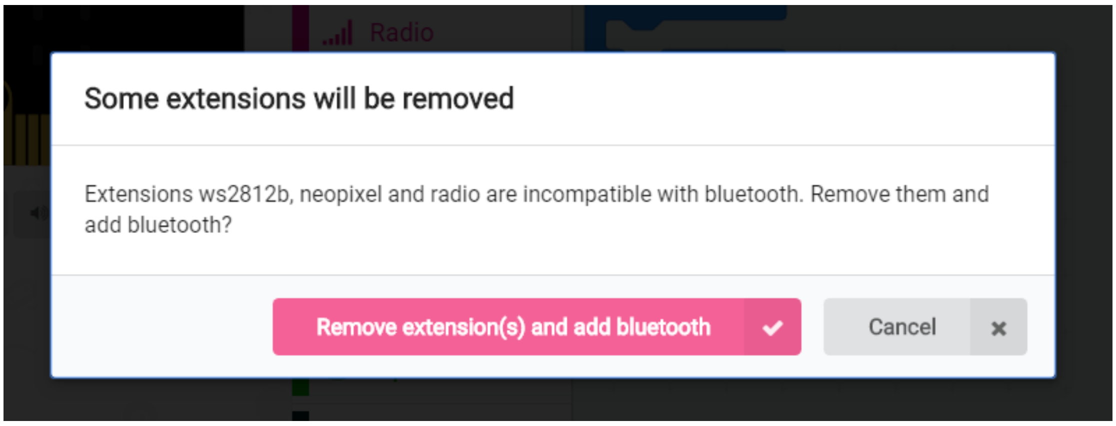
CHAPTER 8 BLUETOOTH

Nrf51822 is a BLE (Bluetooth Low Energy) module allows micro:bit to send and receive Bluetooth signals. With this feature, we can control robot by our mobile phones via Bluetooth.

ADD PACKAGE

To use Bluetooth, we need to first add the supports package. Note that the Bluetooth package is incompatible with extensions ws2812b, neopixel and radio.


ck
Extensions
Search or enter project URL...
Beta
bluetooth
Bluetooth services
devices
BETA - Camera, remote control and other Bluetooth services. App required.
radio-broadcast



Some extensions will be removed

Extensions ws2812b, neopixel and radio are incompatible with bluetooth. Remove them and add bluetooth?

BLUETOOTH CONNETION

Before programming, you need to install the Bluetooth APP to your phone. (This app only support Android)

- Bluetooth App for Android

1. Place bluetooth *** service blocks in on start

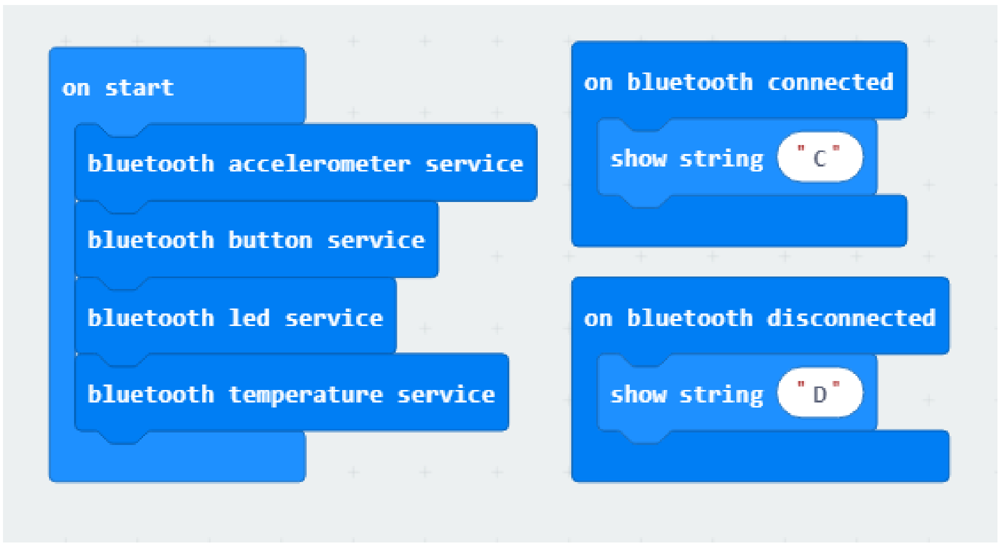
   - This there are several services can be used

2. Place on Bluetooth connected and show string "C" blocks

   - Make micro:bit show "C" if Bluetooth is connected

3. Place Bluetooth disconnected and show string "D" blocks

   - Make micro:bit show "D" if Bluetooth is disconnected

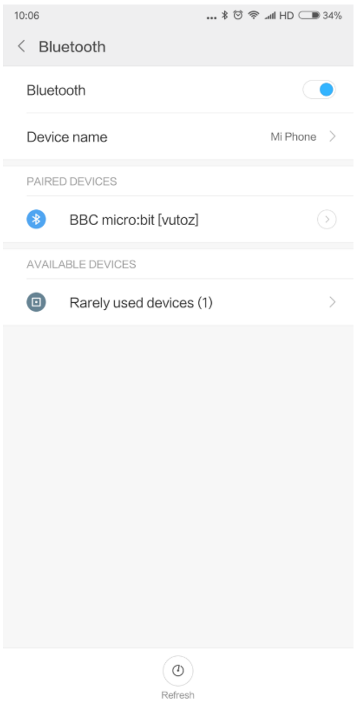


```
on start
    bluetooth accelerometer service
    bluetooth button service
    bluetooth led service
    bluetooth temperature service

on bluetooth connected
    show string "C"

on bluetooth disconnected
    show string "D"
```

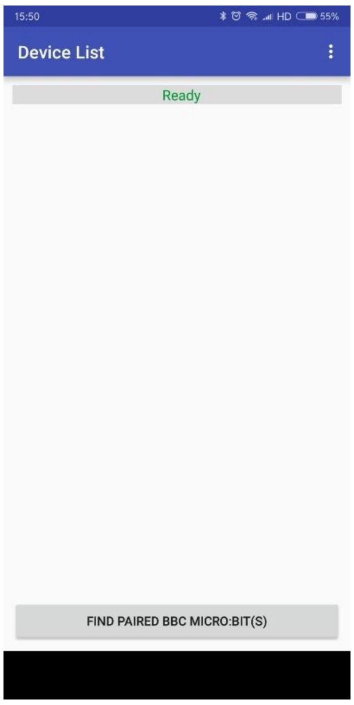
4. Download the script to micro:bit and test

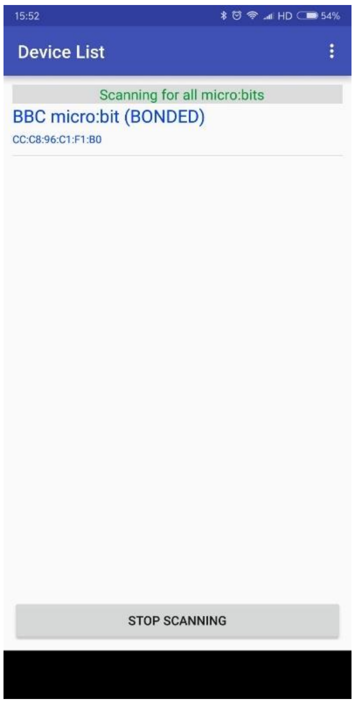
5. After downloading, you need to hold buttons A and B of micro:bit at the same time, and then press reset button, after Bluetooth icon is showed on LEDs matrix, release buttons. This step is to set the micro:bit to pairing mode.

6. Open Bluetooth service of your phone, scanning and find the BBC micro:bit device, pair with it and connect. After paring, icon “√” is displayed. Note that you should delete paired micro:bit device before a new pairing.



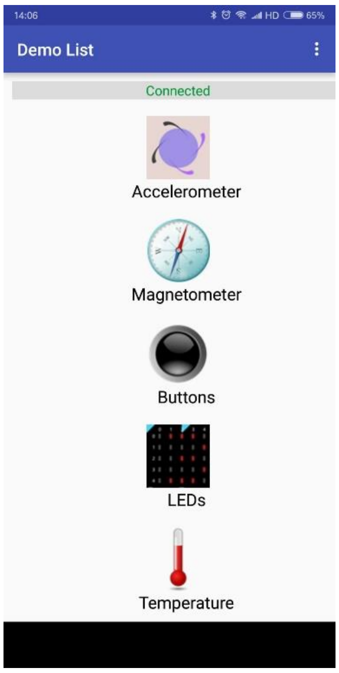
7. Open the Bluetooth App, click “FIND PAIRED BBC MICRO:BIT(S)”



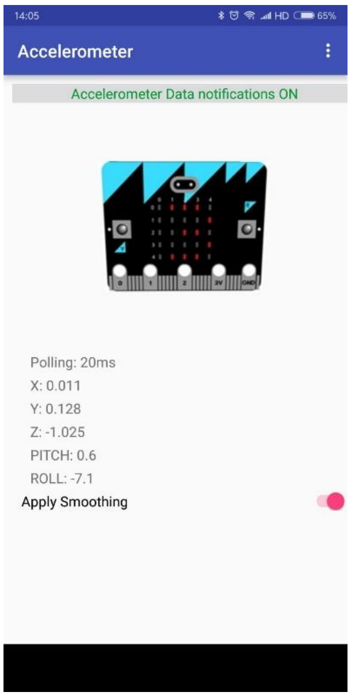


8. Click BBC micro:bit (BOUNDED) to connect. Note that only the device with (BOUNDED) suffix can be connected.

9. After connecting, control options are listed as below, and LEDs matrix shows "C"

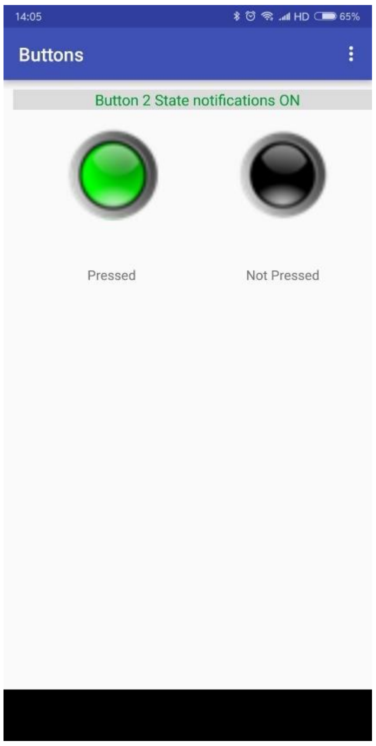


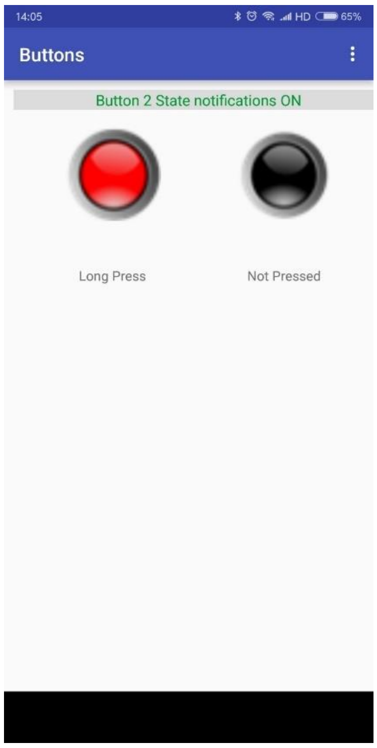
10. Then you can try to control micro:bit by Bluetooth. Click Accelerometer, tilt your micro:bit, and the micro:bit image in APP tilt as well.



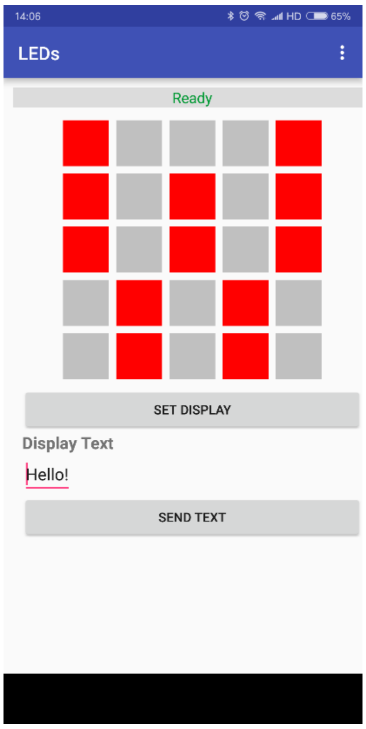
Polling: 20ms
X: 0.011
Y: 0.128
Z: -1.025
PITCH: 0.6
ROLL: -7.1

11. Click button option. Press buttons of your micro:bit. The buttons icon in APP react the button event.

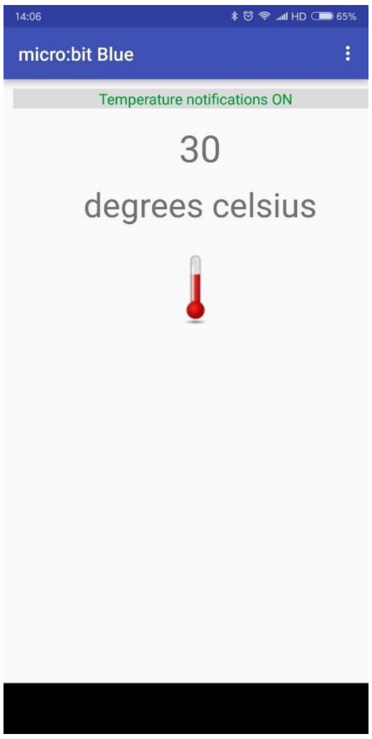




12. Click LEDs option, edit the LEDs matrix then clock SET DISPLAY, or send a text



13. Click Temperature option, the current temperature is showed in APP



Temperature notifications ON

30

degrees celsius

【Note】We only start four services in the code, so some of the services showed in APP are unavailable.

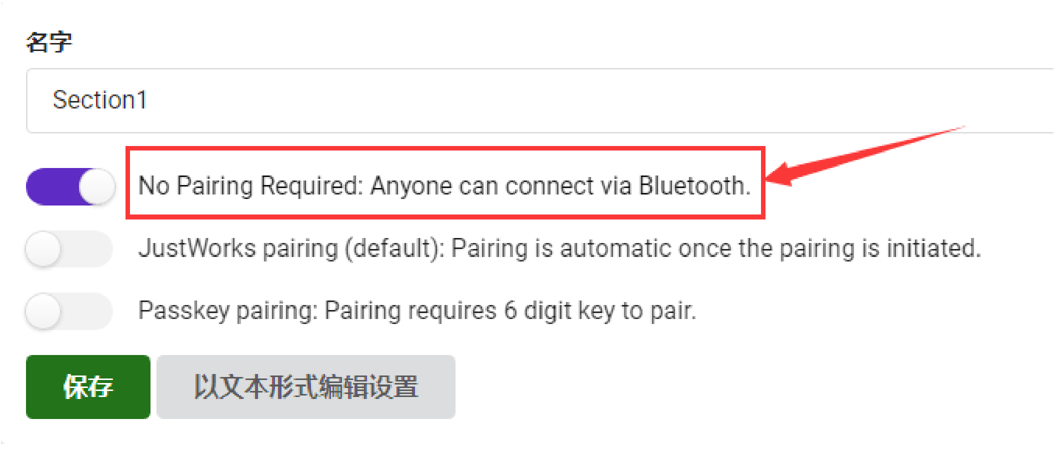
BLUETOOTH CONNECTING 2

To connect with micro:bit, we should first pair then connect, there are two much steps.

Here we modify the last script to make it simpler.

1. Click the icon on the top right: More... ->Project Setting, check the first option

   “No Pairing Required: Anyone can connect via Bluetooth”



名字

Section1

No Pairing Required: Anyone can connect via Bluetooth.

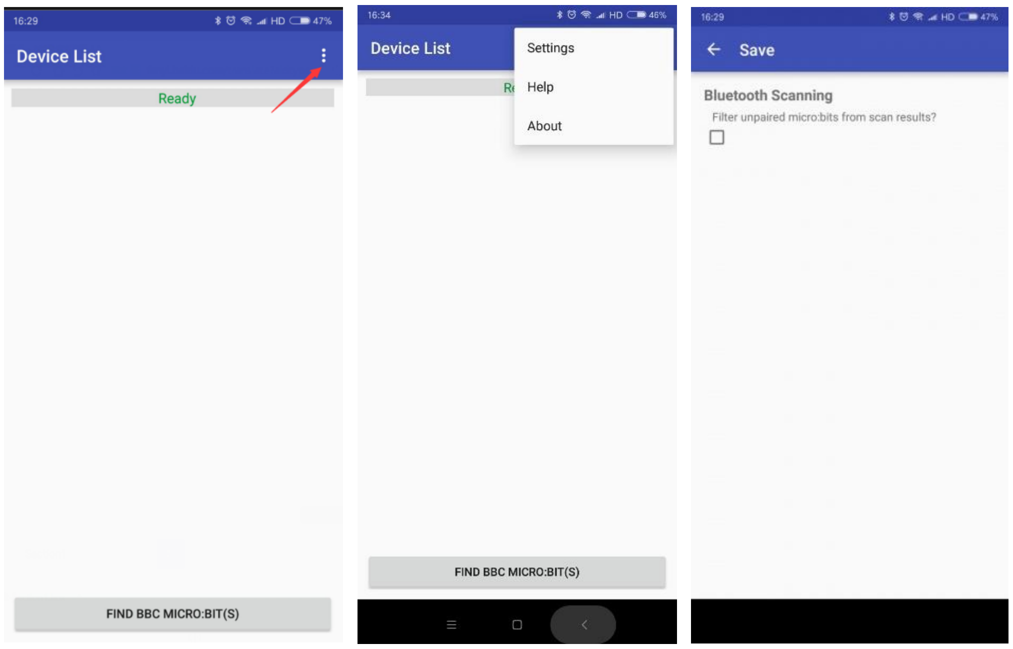
JustWorks pairing (default): Pairing is automatic once the pairing is initiated.

Passkey pairing: Pairing requires 6 digit key to pair.

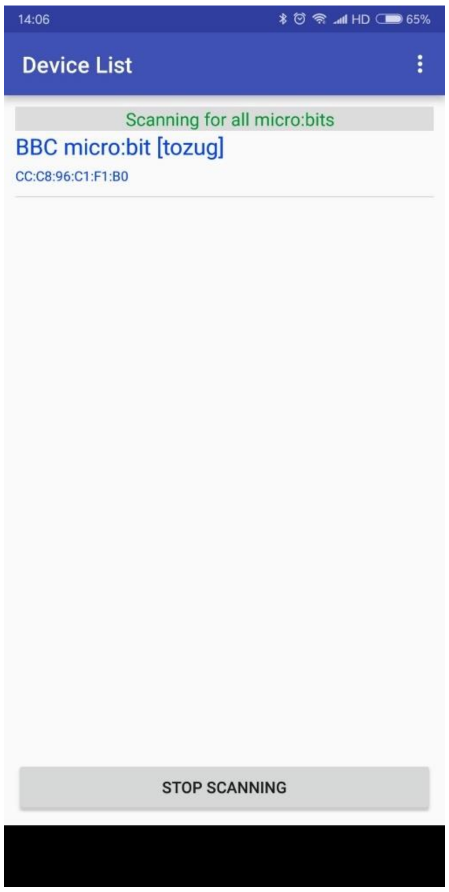
2. Download the script to micro:bit

3. Open App, and change its setting, clock the three-dots icon on the top right,

   choose settings, then uncheck option “Filter unpaired micro:bit from scan results



4. After setting, scanning devices again. BBC micro:bit[tozug][4] is listed, then you can clock it to connect with micro:bit. Don' t forget to delete the paired device (the device with (BOUNDED) suffix)


14:06 65%

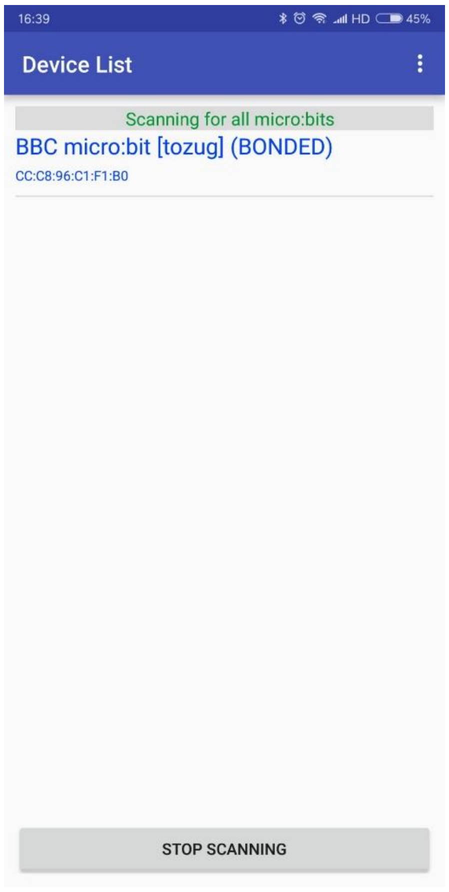
Device List

Scanning for all micro:bits

BBC micro:bit [tozug]

CC:C8:96:C1:F1:B0

STOP SCANNING


16:39 45%

Device List

Scanning for all micro:bits

BBC micro:bit [tozug] (BONDED)

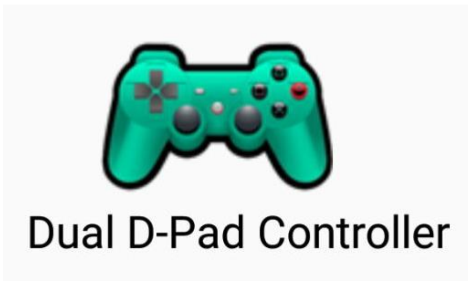
CC:C8:96:C1:F1:B0

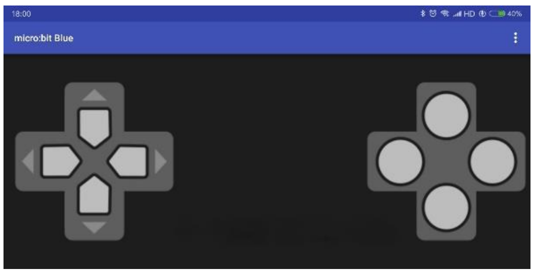
STOP SCANNING

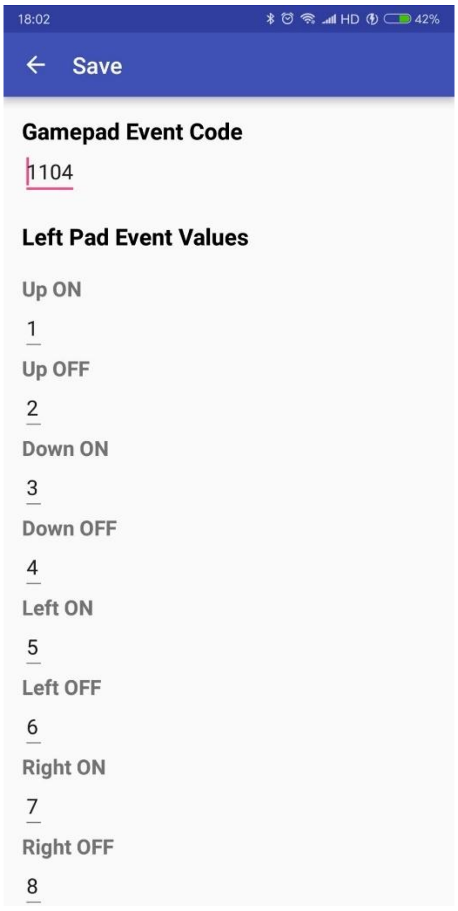
[4] [tozug] this information is different for every micro:bit

D-PAD CONTROLLER

In the App, you can find an option named Dual D-Pad Controller. Click it to enter a control page with dual D-pad (eight buttons). Click the top right three-dots icon you can change the settings. Every time virtual buttons are pressed, related signals are sent to micro:bit via Bluetooth. Gamepad events are triggered by these signals on micro:bit.







Gamepad Event Code

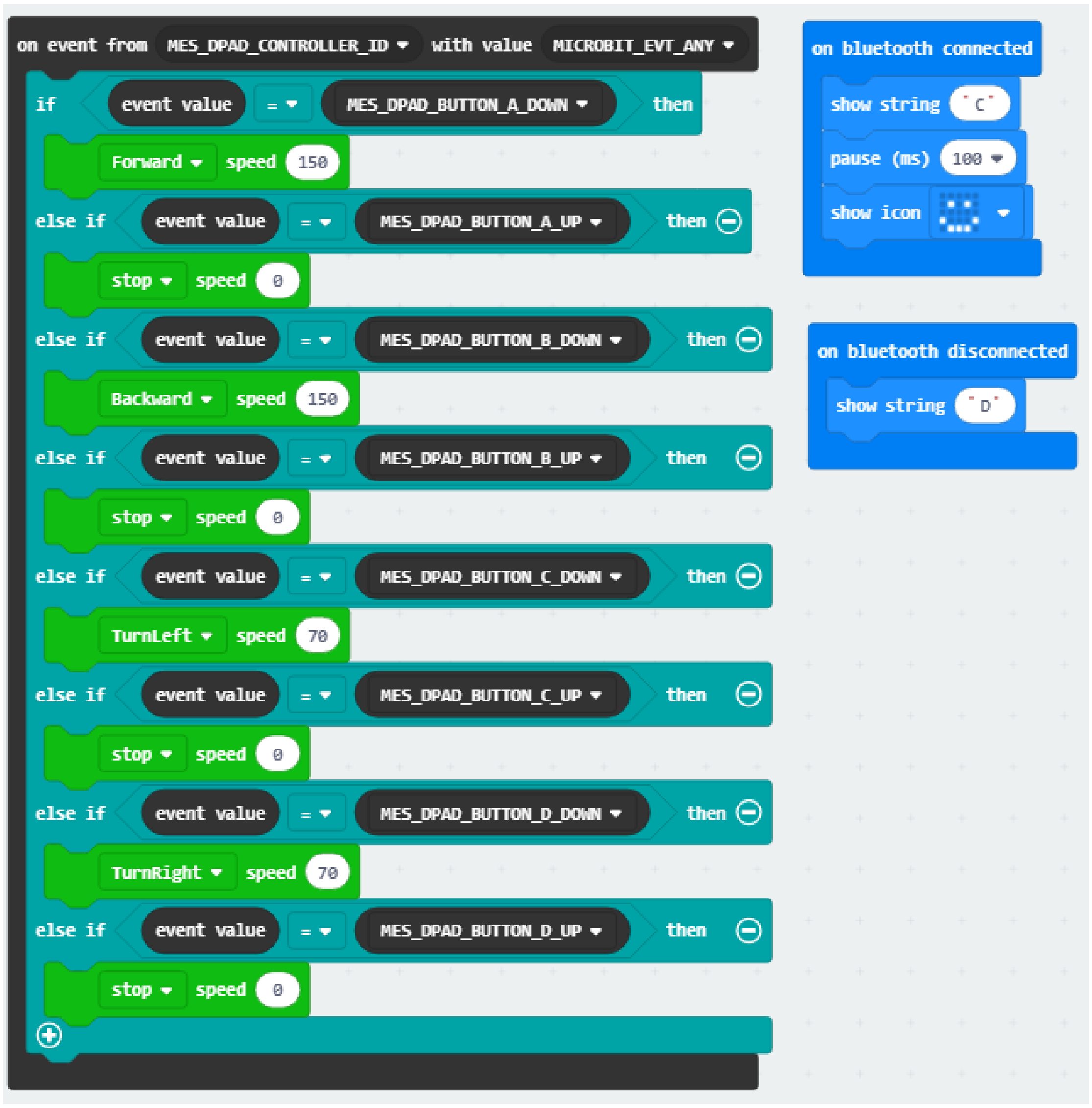
1104

Left Pad Event Values

Up ON	1
Up OFF	2
Down ON	3
Down OFF	4
Left ON	5
Left OFF	6
Right ON	7
Right OFF	8

Let’ s do the script.

1. The whole script is as blow:



```
on event from MES_DPAD_CONTROLLER_ID with value MICROBIT_EVT_ANY
  if event value = MES_DPAD_BUTTON_A_DOWN then
    Forward speed 150
  else if event value = MES_DPAD_BUTTON_A_UP then
    stop speed 0
  else if event value = MES_DPAD_BUTTON_B_DOWN then
    Backward speed 150
  else if event value = MES_DPAD_BUTTON_B_UP then
    stop speed 0
  else if event value = MES_DPAD_BUTTON_C_DOWN then
    TurnLeft speed 70
  else if event value = MES_DPAD_BUTTON_C_UP then
    stop speed 0
  else if event value = MES_DPAD_BUTTON_D_DOWN then
    TurnRight speed 70
  else if event value = MES_DPAD_BUTTON_D_UP then
    stop speed 0

on bluetooth connected
  show string "C"
  pause (ms) 100
  show icon

on bluetooth disconnected
  show string "D"
```

2. on event from ... with value block can be found in **Control** package

3. MES_DPAD_CONTROLLER_ID is signal (the event) from APP

4. MICROBIT_EVT_ANT is the value of the event

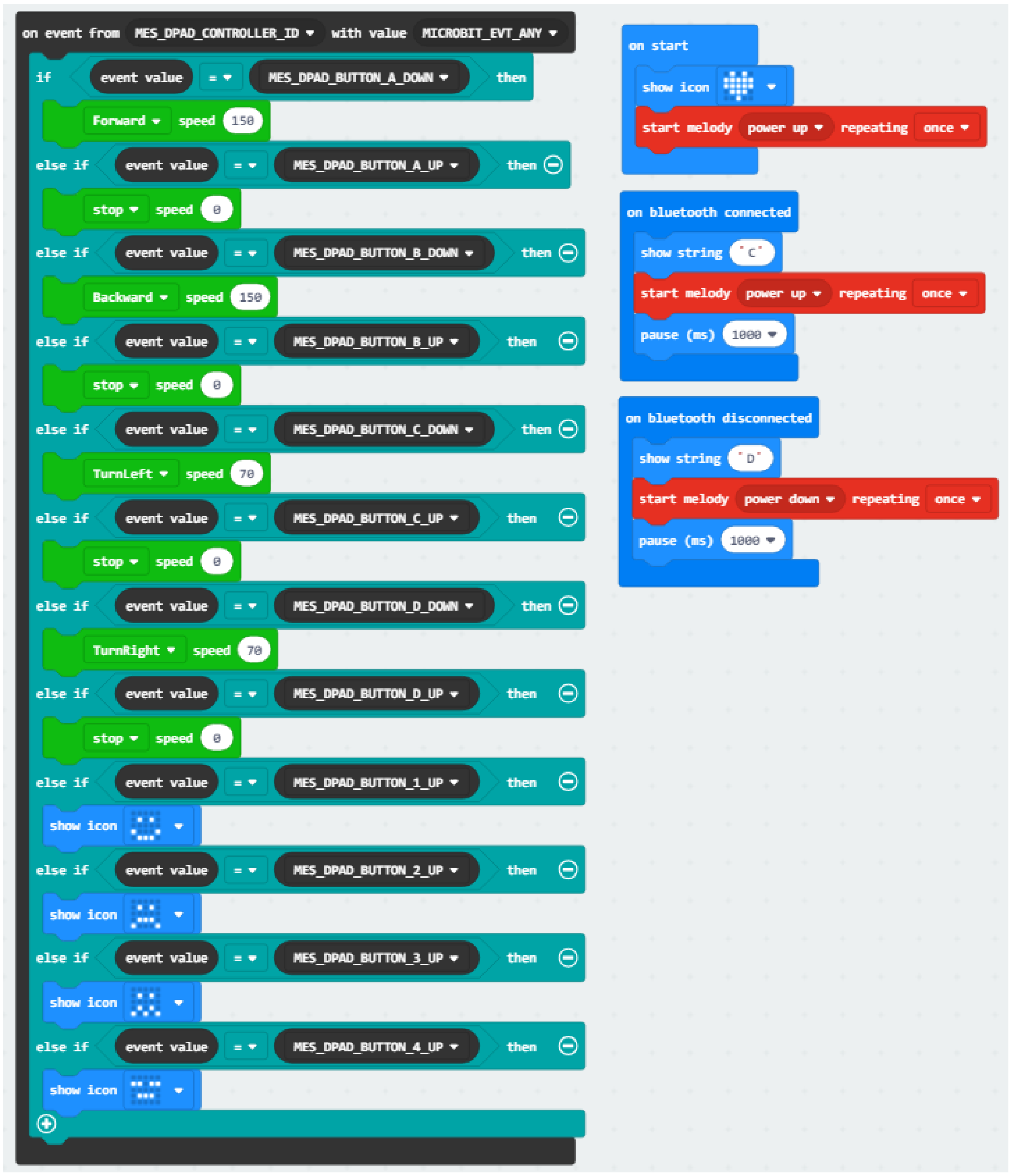
5. MES_DPAD_CONTROLLER_ID and MICROBIT_EVT_ANY blocks can be founded in

   Control->More

6. Download the script and test.

D-PAD CONTROLLER 2

In the Last script, we only use fours buttons, however, as we know, there is dual D-Pad controller in APP, which has eight buttons. We modify the last script, make micro:bit show different icon when the less four buttons are press.



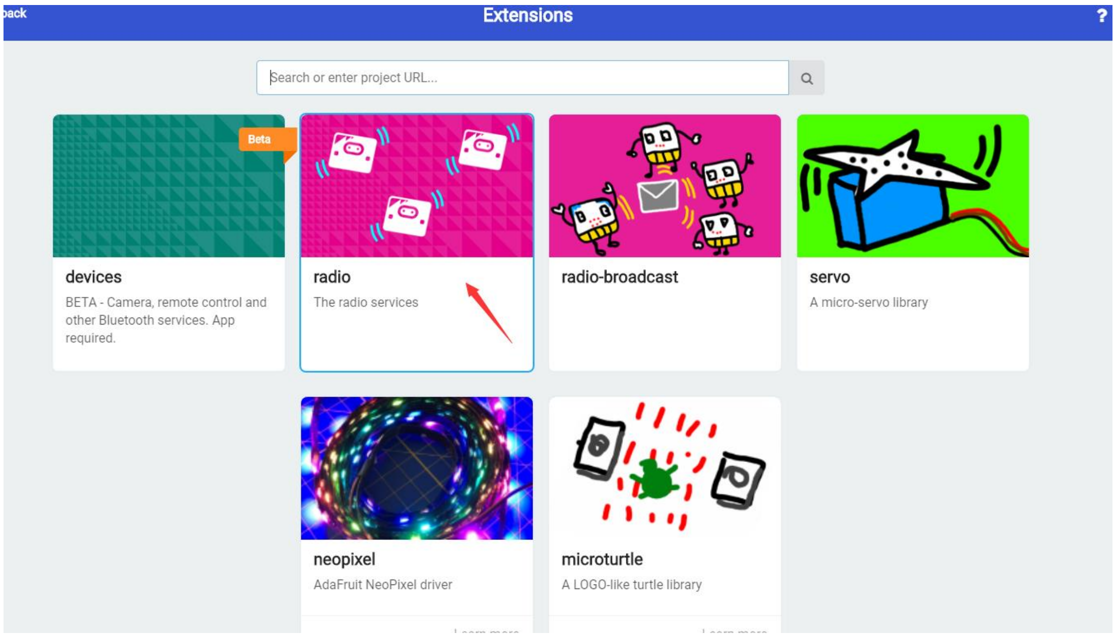
```
on event from MES_DPAD_CONTROLLER_ID with value MICROBIT_EVT_ANY
    if event value = MES_DPAD_BUTTON_A_DOWN then
        Forward speed 150
    else if event value = MES_DPAD_BUTTON_A_UP then
        stop speed 0
    else if event value = MES_DPAD_BUTTON_B_DOWN then
        Backward speed 150
    else if event value = MES_DPAD_BUTTON_B_UP then
        stop speed 0
    else if event value = MES_DPAD_BUTTON_C_DOWN then
        TurnLeft speed 70
    else if event value = MES_DPAD_BUTTON_C_UP then
        stop speed 0
    else if event value = MES_DPAD_BUTTON_D_DOWN then
        TurnRight speed 70
    else if event value = MES_DPAD_BUTTON_D_UP then
        stop speed 0
    else if event value = MES_DPAD_BUTTON_1_UP then
        show icon
    else if event value = MES_DPAD_BUTTON_2_UP then
        show icon
    else if event value = MES_DPAD_BUTTON_3_UP then
        show icon
    else if event value = MES_DPAD_BUTTON_4_UP then
        show icon

on start
    show icon
    start melody power up repeating once

on bluetooth connected
    show string "C"
    start melody power up repeating once
    pause (ms) 1000

on bluetooth disconnected
    show string "D"
    start melody power down repeating once
    pause (ms) 1000
```

CHAPTER 9 2.4G RADIO

2.GHz radio is a wireless communication, popularly used for remote communication like intercom. To do this project, you need at least two micro:bit and one Joystick for micro:bit.

ADD PACKAGE

Add radio package first.

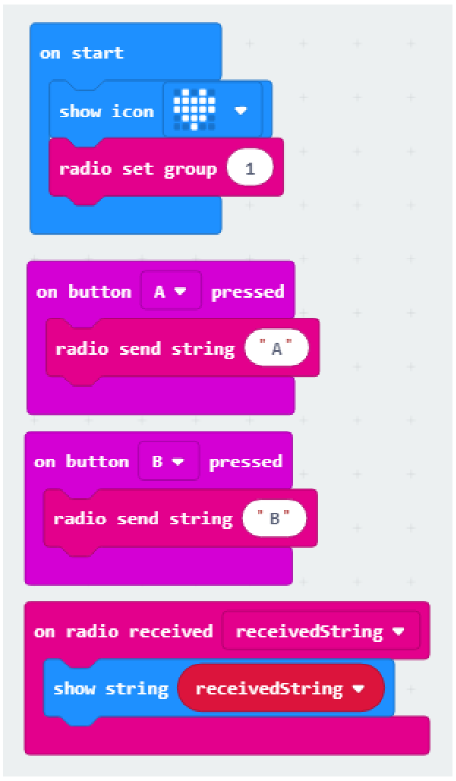


RADIO COMMUNICATING

Note that you should

1. You need to set the two micro:bit in one group using block radio set group 1
2. Place on button A pressed and radio send string A, let micro:bit to send string A when button A is pressed, same setting with the button B.

3. User radio received block to receive string.



```
on start
    show icon [heart] ▾
    radio set group 1

on button A ▾ pressed
    radio send string "A"

on button B ▾ pressed
    radio send string "B"

on radio received receivedString ▾
    show string receivedString ▾
```

4. Download the script to both micro:bit and test

5. Expected result it that, if the buttons of one micro:bit are pressed, A or B is showed on another micro:bit.

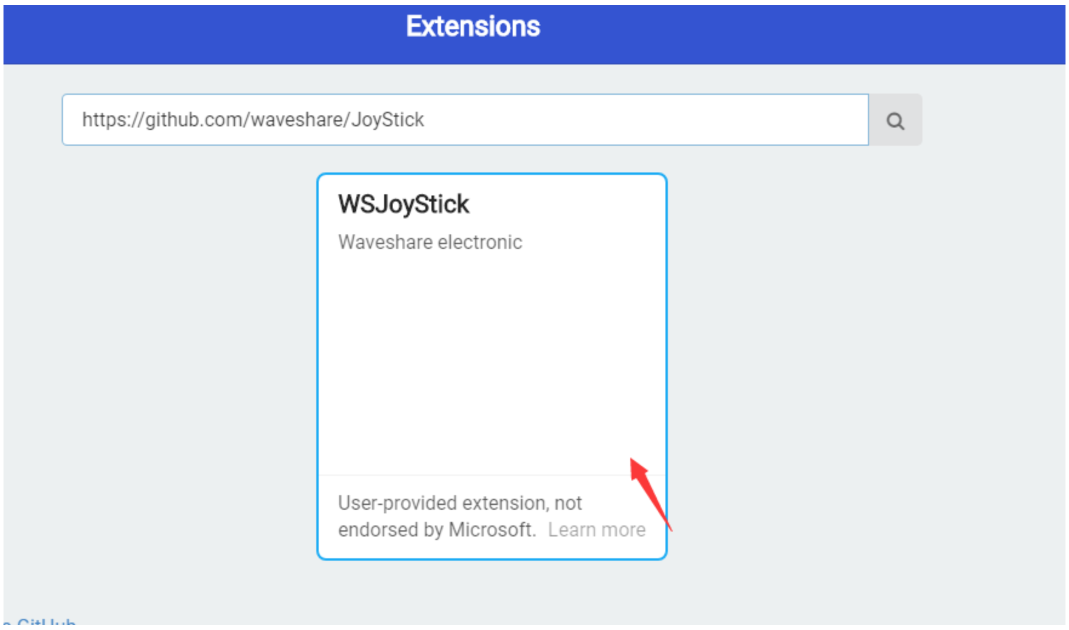
JOYSTICK FOR MICRO:BIT

Joystick for micro:bit is a little module with one joystick and four buttons. Here we use it to control Robot via Radio.

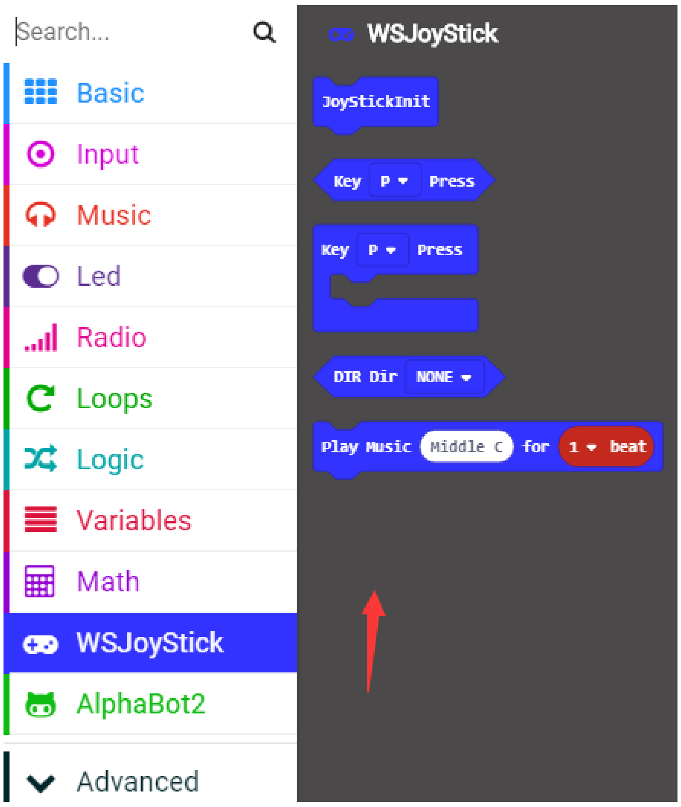
1. Add package

   We provide a package for joystick for micro:bit, you can add it with this address:

   https://github.com/waveshare/JoyStick

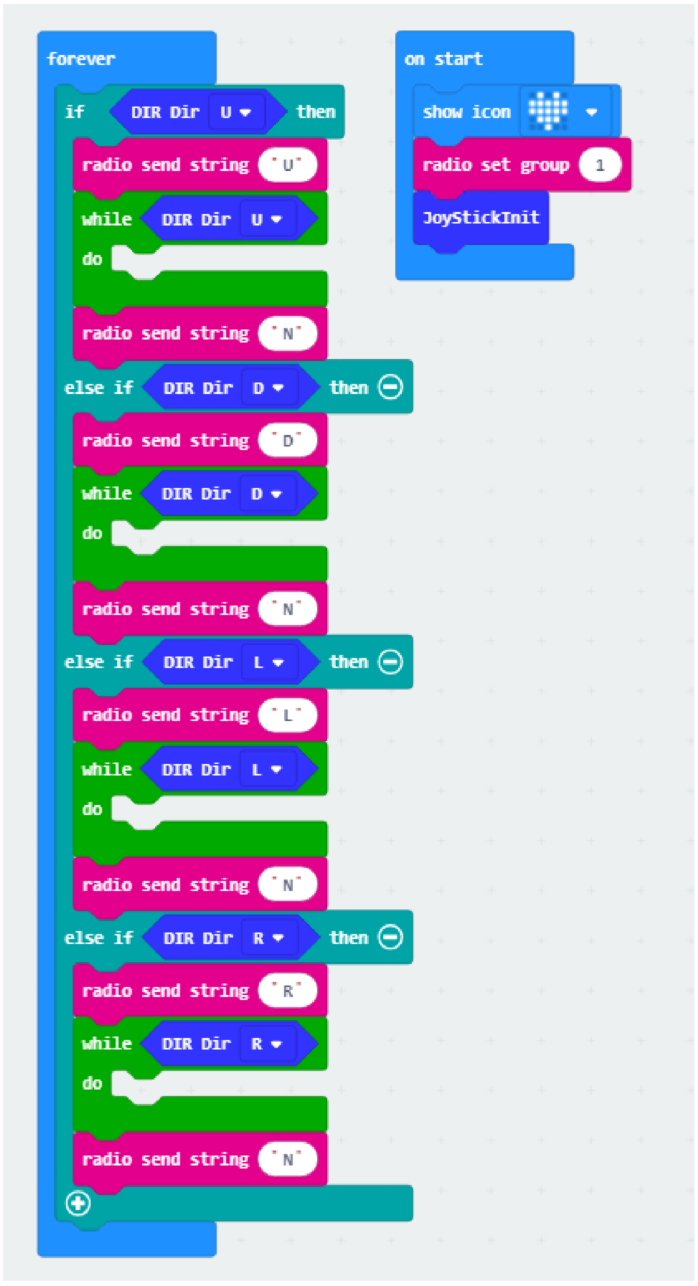

Extensions
https://github.com/waveshare/JoyStick
WSJoyStick
Waveshare electronic
User-provided extension, not endorsed by Microsoft. Learn more

There are five blocks. JoyStickInit block must be used on start to initialize joystick module.


Search...
Basic
Input
Music
Led
Radio
Loops
Logic
Variables
Math
WSJoyStick
AlphaBot2
Advanced
WSJoyStick
JoyStickInit
Key P Press
Key P Press
DIR Dir NONE
Play Music Middle C for 1 beat

2. The script for sender:

   - Download this script to one micro:bit as sender, which insert to joystick module



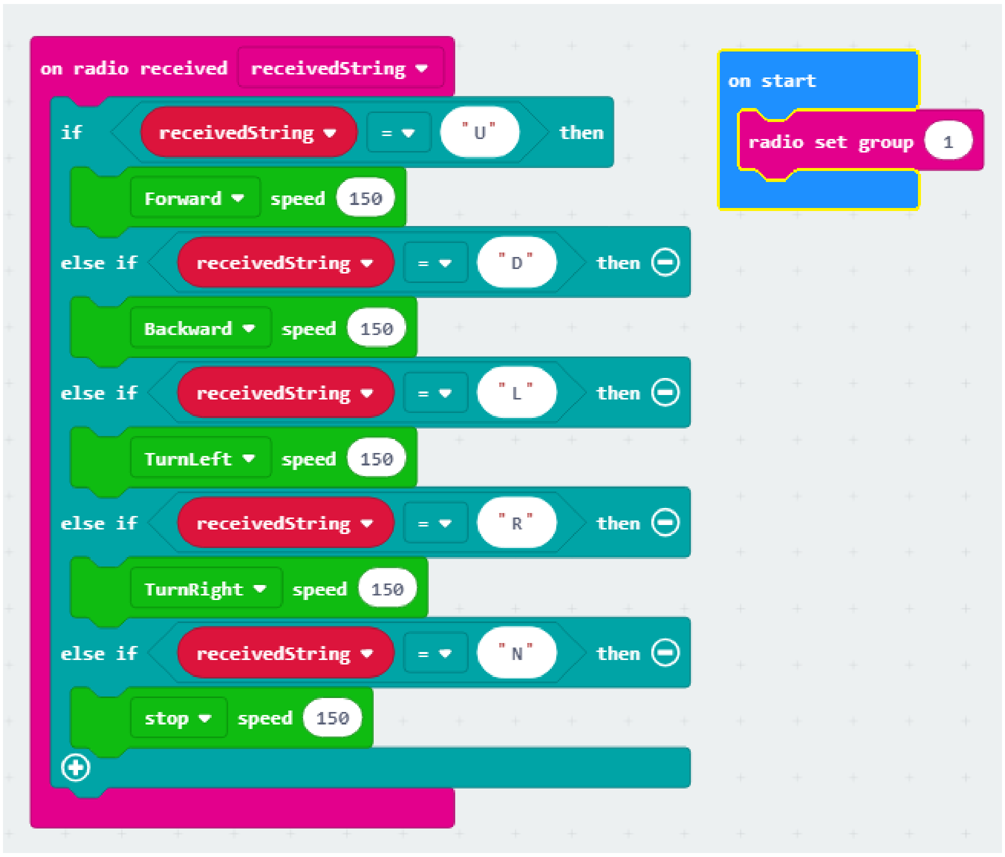
```
forever
  if DIR Dir U then
    radio send string "U"
    while DIR Dir U
    do
    radio send string "N"
  else if DIR Dir D then
    radio send string "D"
    while DIR Dir D
    do
    radio send string "N"
  else if DIR Dir L then
    radio send string "L"
    while DIR Dir L
    do
    radio send string "N"
  else if DIR Dir R then
    radio send string "R"
    while DIR Dir R
    do
    radio send string "N"

on start
  show icon
  radio set group 1
  JoyStickInit
```

3. The script for receiver

   -As we say, you must set the receiver and sender in the same group

   -Blocks "U", "D", "L", "R", "N" can be founded in **Advanced ->Text**

-Download this script to another micro:bit as receiver, which insert to Alphabot2

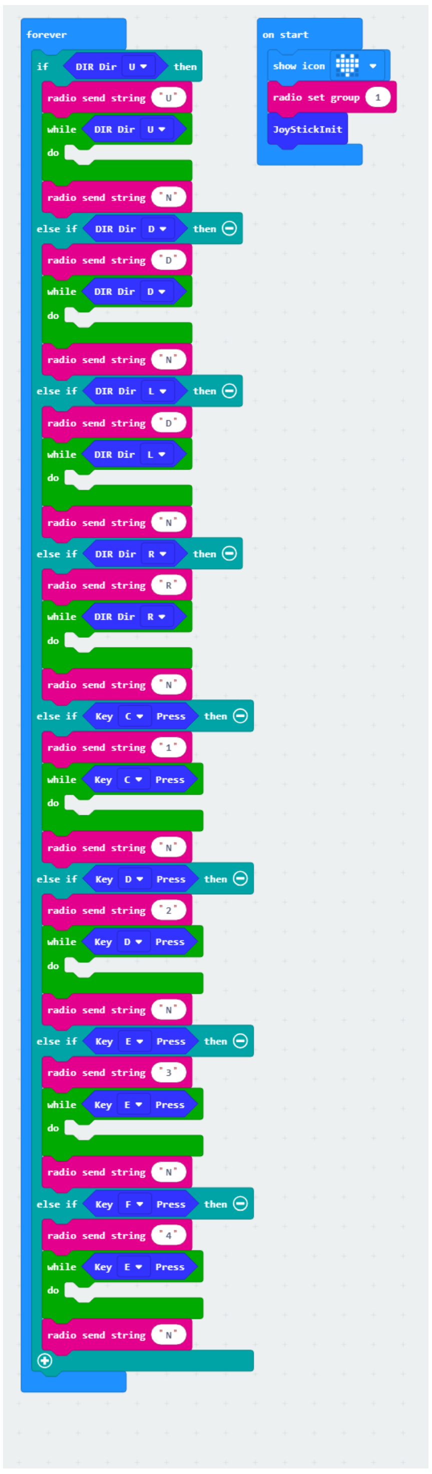


```
on radio received receivedString
    if receivedString = "U" then
        Forward speed 150
    else if receivedString = "D" then
        Backward speed 150
    else if receivedString = "L" then
        TurnLeft speed 150
    else if receivedString = "R" then
        TurnRight speed 150
    else if receivedString = "N" then
        stop speed 150

on start
    radio set group 1
```

JOYSTICK FOR MICRO:BIT 2

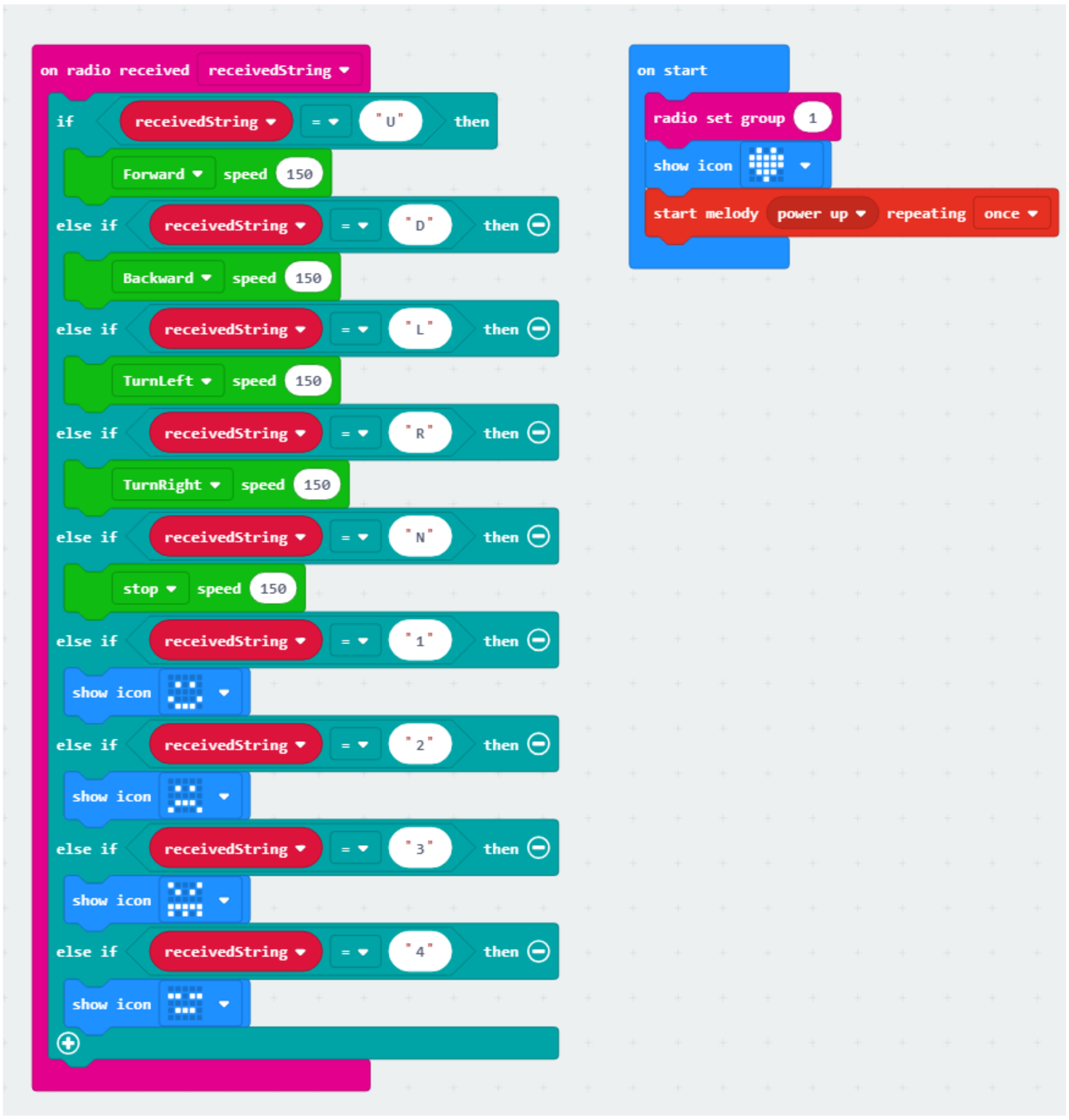
Modify the last two scripts, make LEDs matrix show emoticons when four color buttons of joystick module are pressed

1. The script for sender:



```
forever
    if DIR Dir U then
        radio send string "U"
        while DIR Dir U
        do
        radio send string "N"
    else if DIR Dir D then
        radio send string "D"
        while DIR Dir D
        do
        radio send string "N"
    else if DIR Dir L then
        radio send string "D"
        while DIR Dir L
        do
        radio send string "N"
    else if DIR Dir R then
        radio send string "R"
        while DIR Dir R
        do
        radio send string "N"
    else if Key C Press then
        radio send string "1"
        while Key C Press
        do
        radio send string "N"
    else if Key D Press then
        radio send string "2"
        while Key D Press
        do
        radio send string "N"
    else if Key E Press then
        radio send string "3"
        while Key E Press
        do
        radio send string "N"
    else if Key F Press then
        radio send string "4"
        while Key E Press
        do
        radio send string "N"

on start
    show icon
    radio set group 1
    JoyStickInit
```

2. Script for receiver:

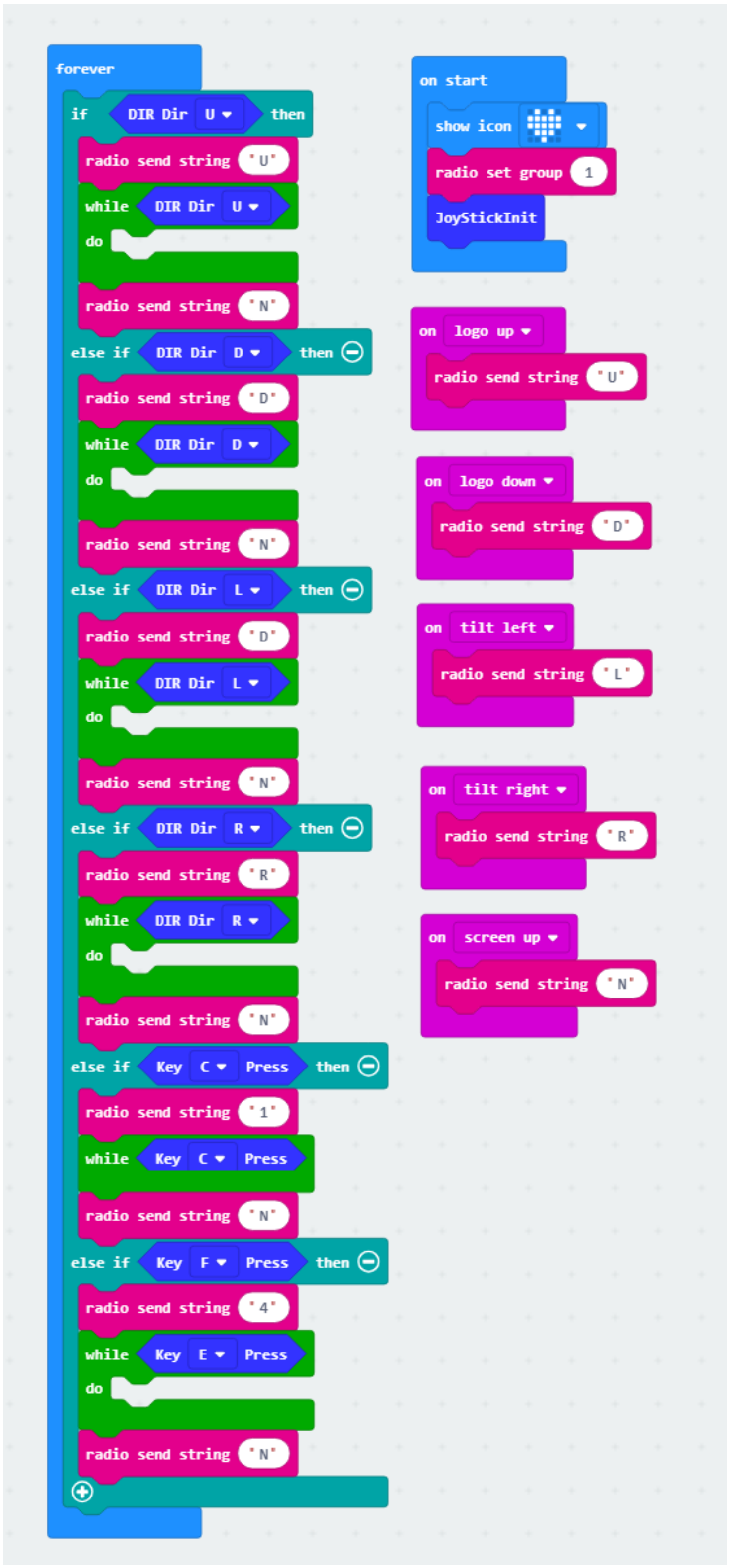


```
on radio received receivedString
    if receivedString = "U" then
        Forward speed 150
    else if receivedString = "D" then
        Backward speed 150
    else if receivedString = "L" then
        TurnLeft speed 150
    else if receivedString = "R" then
        TurnRight speed 150
    else if receivedString = "N" then
        stop speed 150
    else if receivedString = "1" then
        show icon
    else if receivedString = "2" then
        show icon
    else if receivedString = "3" then
        show icon
    else if receivedString = "4" then
        show icon

on start
    radio set group 1
    show icon
    start melody power up repeating once
```

POSTURE CONTROL

In chapter 3, we have described sensors, accelerometer can be used to detect status of micro:bit, so we add the posture control part to last scripts.

1. Script for sender:

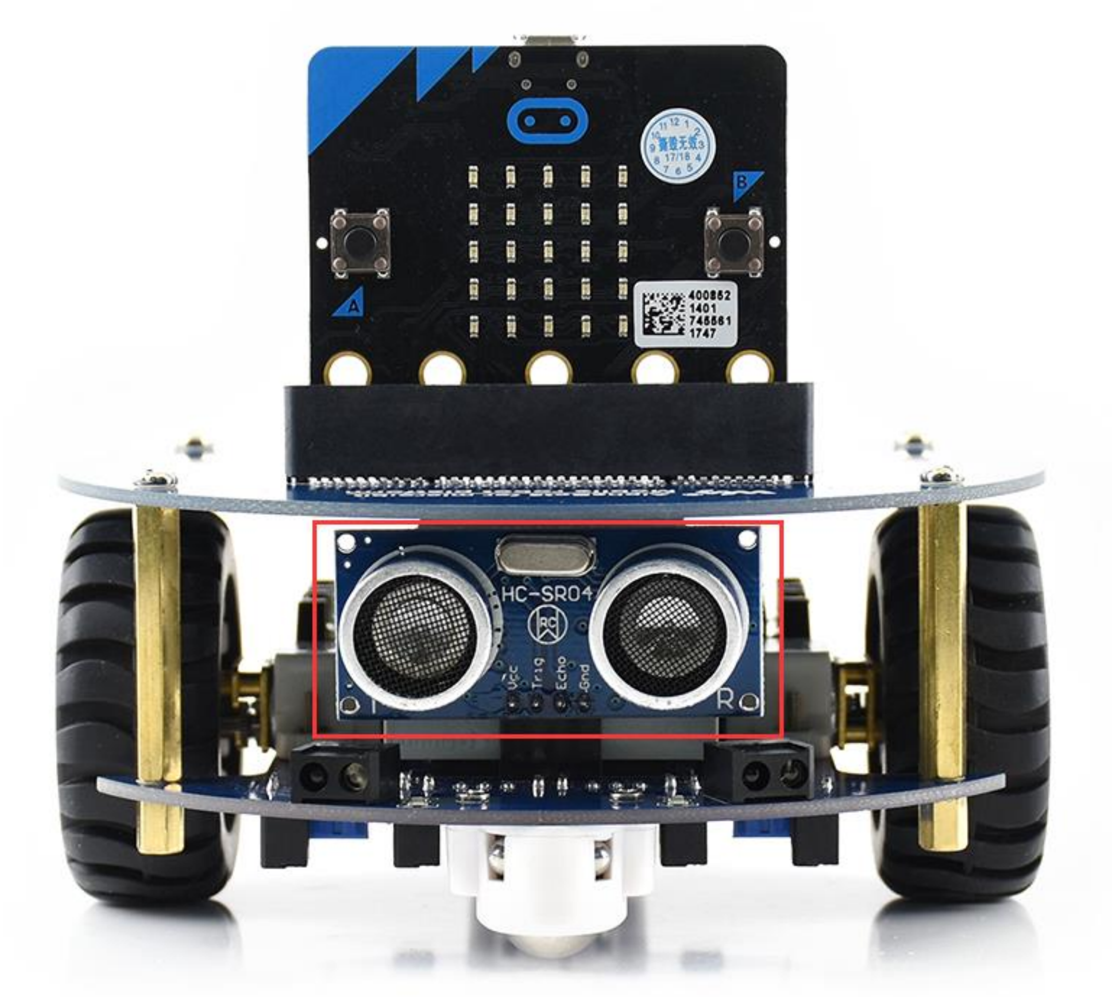


```
forever
  if  DIR Dir U ▾  then
    radio send string "U"
    while  DIR Dir U ▾
    do
    radio send string "N"
  else if  DIR Dir D ▾  then ⊖
    radio send string "D"
    while  DIR Dir D ▾
    do
    radio send string "N"
  else if  DIR Dir L ▾  then ⊖
    radio send string "D"
    while  DIR Dir L ▾
    do
    radio send string "N"
  else if  DIR Dir R ▾  then ⊖
    radio send string "R"
    while  DIR Dir R ▾
    do
    radio send string "N"
  else if  Key C ▾ Press  then ⊖
    radio send string "1"
    while  Key C ▾ Press
    radio send string "N"
  else if  Key F ▾ Press  then ⊖
    radio send string "4"
    while  Key E ▾ Press
    do
    radio send string "N"
  ⊕

on start
  show icon ▾
  radio set group 1
  JoyStickInit

on logo up ▾
  radio send string "U"

on logo down ▾
  radio send string "D"

on tilt left ▾
  radio send string "L"

on tilt right ▾
  radio send string "R"

on screen up ▾
  radio send string "N"
```

2. Download the script to sender and test. Robot will move when you tilt micro:bit.

CHAPTER 10 ULTRASONIC OBSTACLE AVOIDANCING

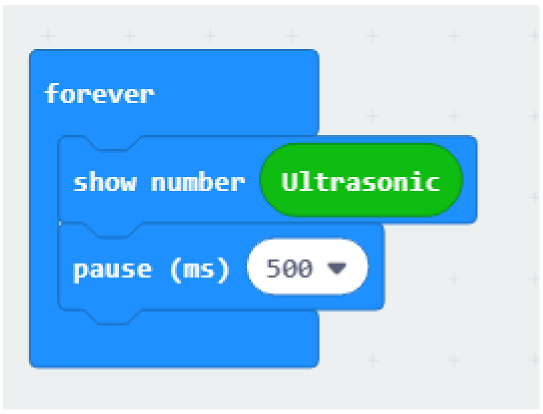
The eyes of the Alphabot2 is ultrasonic sensor, we can use it to do obstacle avoidance.



ULTRASONIC SENSOR

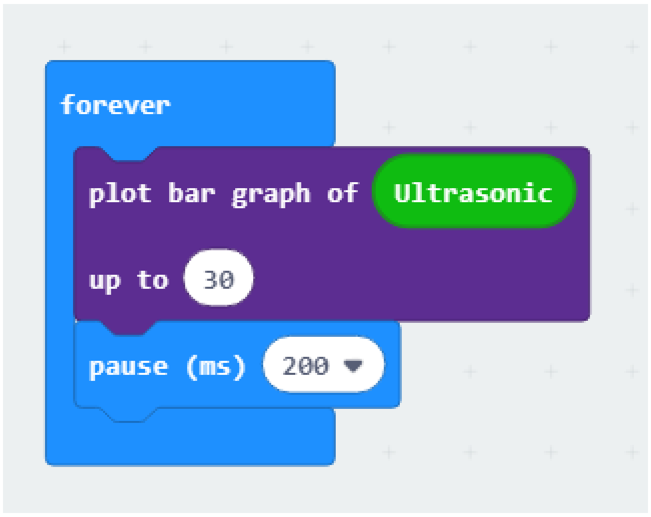
It is similar to the infrared sensor, ultrasonic sensor has one sender and one receiver as well. Ultrasonic sensor can detect the distances of front obstacles.

Let’ s create a new project to read value from ultrasonic sensor.


```
forever
    show number Ultrasonic
    pause (ms) 500
```

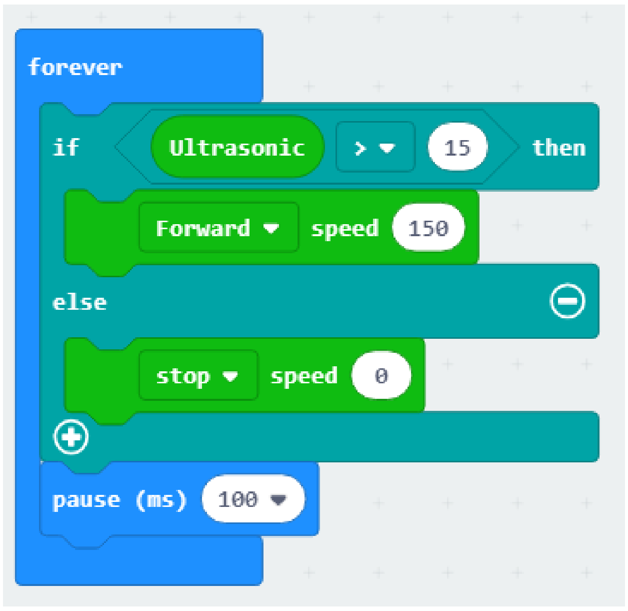
-Ultrasonic block can be founded in **AlphaBot2** package. This block is used to read the distance value from Ultrasonic sensor, which is integer

We can also plot bar graph to show the distance value


```
forever
    plot bar graph of Ultrasonic
    up to 30
    pause (ms) 200
```

OBSTACLE AVOIDING

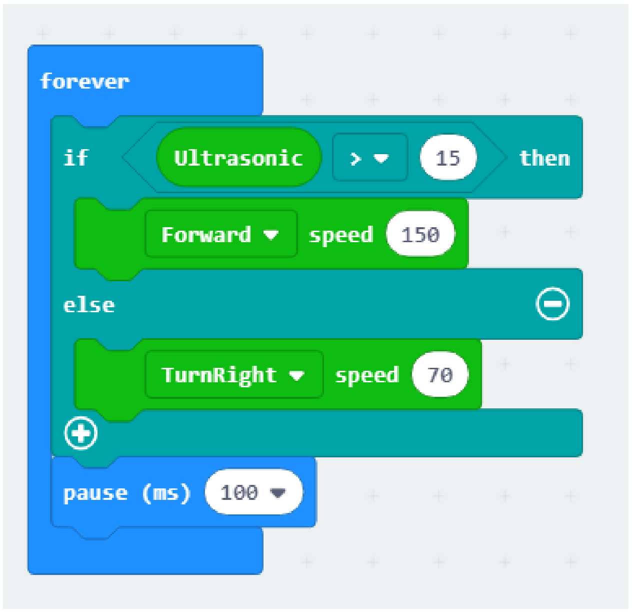
With the ultrasonic sensor, we can make Robot stop if there is obstacle in the front



```
forever
    if Ultrasonic > 15 then
        Forward speed 150
    else
        stop speed 0
    pause (ms) 100
```

Robot keeps forward if the distance of obstacle in the front is longer than 15cm at the beginning. and stop when the distance is less than 15cm.

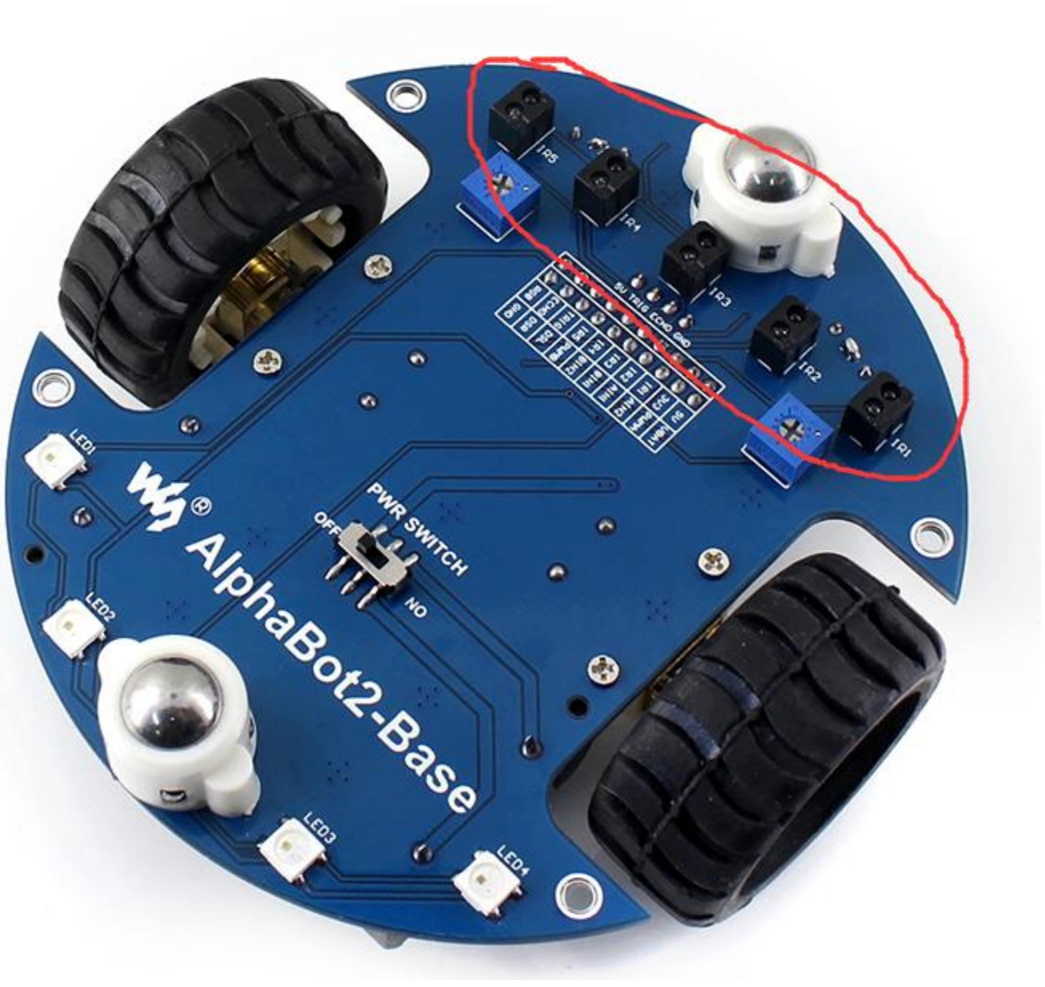
Modify the late script, we can let Robot turn right if obstacle is detected.



```
forever
    if Ultrasonic > 15 then
        Forward speed 150
    else
        TurnRight speed 70
    pause (ms) 100
```

CHAPTER 11 LINE TRACKING

In the bottom side of Alphabot2-base board, tracker sensor is integrated for tracking function.



TRACKER SENSOR

The tracker sensor helps the robot move along the black line on the ground. There are five detectors at the bottom of the sensor, and the infrared LED light is used to project infrared light to the ground to detect the deviation of the module relative to the black line.

In **Alphabot2** package:

SensorCalibrated block is used to calibrate the sensor for accurate Min and Max value.

AnalogRead block is used to read original data from infrared tacker sensor.

ReadCalibrated block is used to read the calibrated data

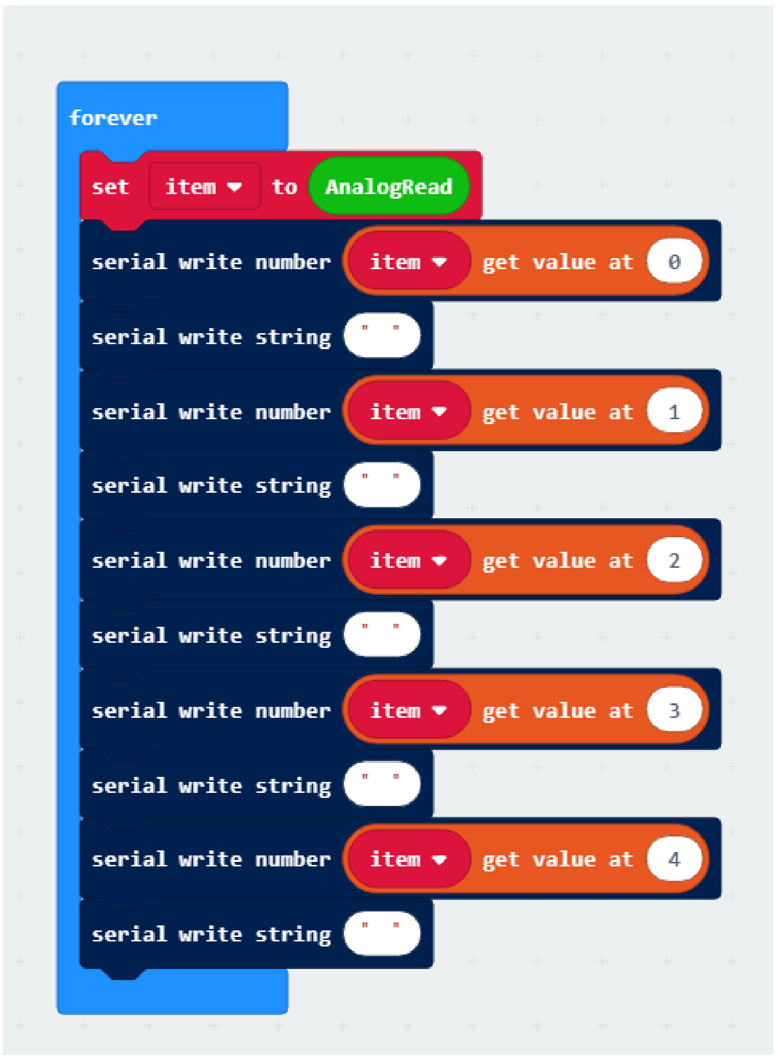
ReadSenorMax block is used to read the Max calibrated value

ReadSensorMin block is used to read the Min calibrated value

ReadLine block is used to get the position of black line in range 0~4000.   0: Robot is on the extreme left; 2000: Robot is on the center; 4000: Robot is on the extreme right.

ORIGINAL DATA

We create a new project, to read the five value (there are five detectors) and print it to



```
forever
  set item ▾ to AnalogRead
  serial write number item ▾ get value at 0
  serial write string " "
  serial write number item ▾ get value at 1
  serial write string " "
  serial write number item ▾ get value at 2
  serial write string " "
  serial write number item ▾ get value at 3
  serial write string " "
  serial write number item ▾ get value at 4
  serial write string " "
```

-set item to block can be founded in Variable

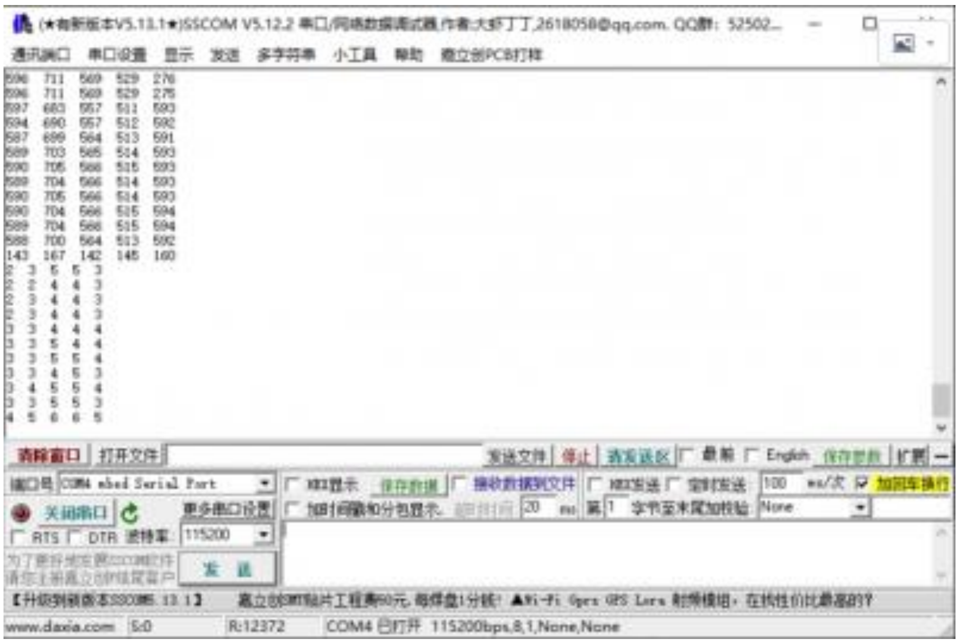
-serial write number and serial write string blocks can be founded in

Advanced->Serial

-item get value at block can be founded in **Advanced->Arrays**

1. Complete the script as above

2. Download the script to micro:bit and open serial software, set baud rate to 115200, 8N1. Data of sensors are printed to serial as below:



```
596  711  569  529  276
596  711  569  529  275
597  683  557  511  593
594  690  557  512  592
587  699  564  513  591
589  703  565  514  593
590  705  566  515  593
589  704  566  514  593
590  705  566  514  593
590  704  566  515  594
589  704  566  515  594
588  700  564  513  592
143  167  142  145  160
2  3  5  5  3
2  2  4  4  3
2  3  4  4  3
2  3  4  4  3
3  3  4  4  4
3  3  5  4  4
3  3  5  5  4
3  3  4  5  3
3  4  5  5  4
3  3  5  5  3
4  5  6  6  5
```

端口号 COM4 mbed Serial Port　波特率: 115200

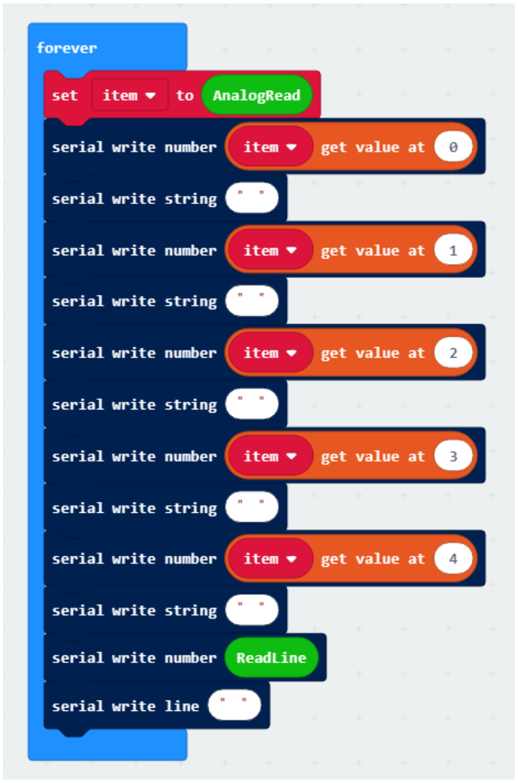
www.daxia.com　S:0　R:12372　COM4 已打开　115200bps,8,1,None,None

- If you put the Alphabot2 on the air, the output values are small, about 0~10

- If you put the robot to white ground, the data will rise to 500~700

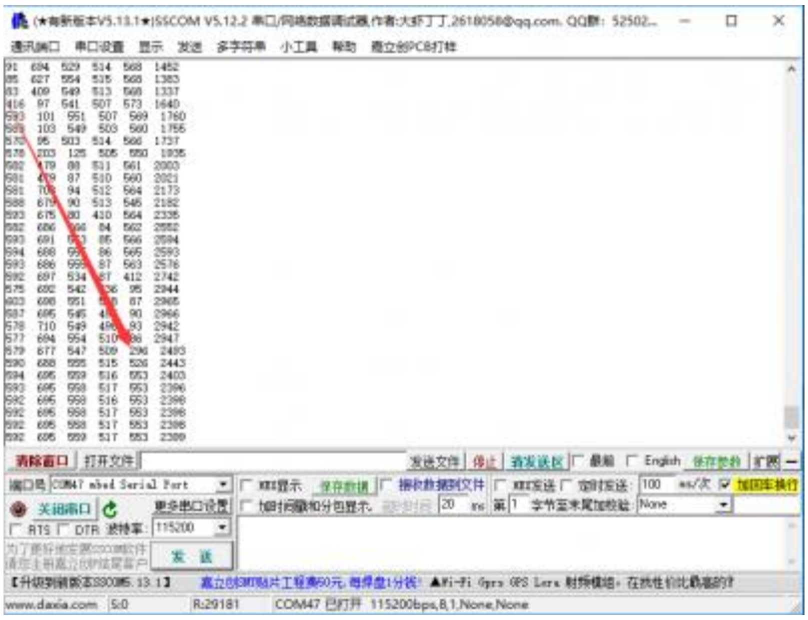
- If you put it on black line, the output are about 100.

3. We modify the script to output the position of black line



```
forever
    set item ▾ to AnalogRead
    serial write number  item ▾ get value at 0
    serial write string " "
    serial write number  item ▾ get value at 1
    serial write string " "
    serial write number  item ▾ get value at 2
    serial write string " "
    serial write number  item ▾ get value at 3
    serial write string " "
    serial write number  item ▾ get value at 4
    serial write string " "
    serial write number  ReadLine
    serial write line " "
```

-Download the code to micro:bit. When black line is under the tracker sensor, data are smaller to 80~100. The extreme right data is the position of black line. Note that these data are not be calibrated. (The black line moves from detector 1 to detector 5 according to the red arrow)

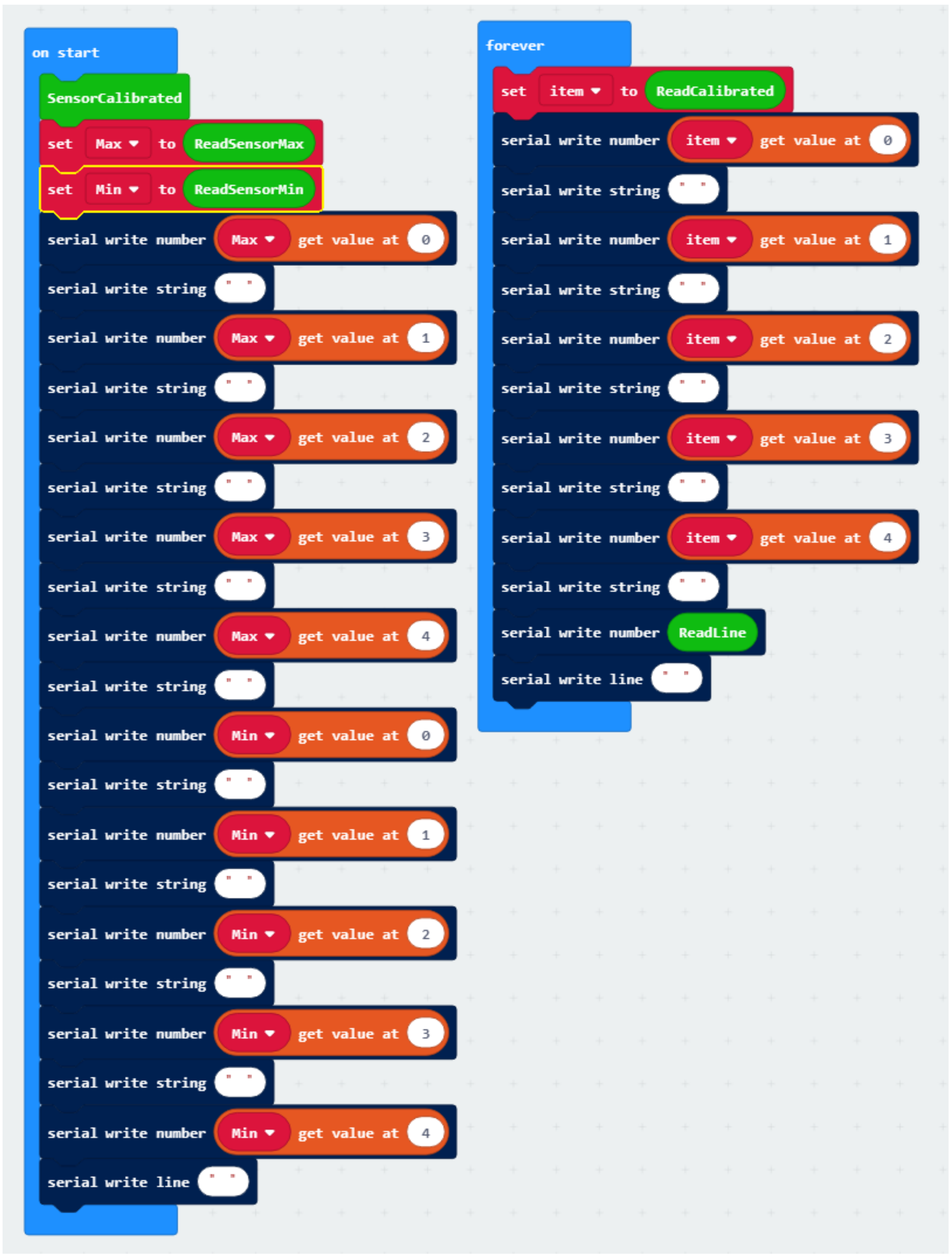


```
91   694  529  514  568  1452
95   627  554  515  568  1383
93   409  549  513  568  1337
416  97   541  507  573  1640
593  101  551  507  569  1760
589  103  549  503  560  1756
570  95   503  514  566  1737
578  203  125  505  550  1935
582  179  88   511  561  2003
581  479  87   510  560  2021
581  708  94   512  564  2173
588  679  90   513  546  2182
593  675  80   410  564  2335
592  686  [illegible]  84   562  2552
593  691  [illegible]  85   566  2594
594  688  [illegible]  86   565  2593
593  686  [illegible]  87   563  2576
592  697  534  87   412  2742
575  692  542  [illegible]  95   2944
603  698  551  [illegible]  87   2965
587  695  545  [illegible]  90   2966
578  710  549  [illegible]  93   2942
577  694  554  510  88   2947
579  677  547  509  296  2483
590  688  555  515  526  2443
594  695  559  516  553  2403
593  695  558  517  553  2396
592  695  558  516  553  2398
592  695  558  517  553  2398
592  695  558  517  553  2398
592  695  559  517  553  2399
```

(★最新版本V5.13.1★)SSCOM V5.12.2 串口/网络数据调试器,作者:大虾丁丁,2618058@qq.com. QQ群: 52502...
通讯端口 串口设置 显示 发送 多字符串 小工具 帮助 联系作者
清除窗口 打开文件
发送文件 停止 清发送区 最前 English 保存参数 扩展
端口号 COM47 mbed Serial Port
关闭串口 更多串口设置
RTS DTR 波特率: 115200
加时间戳和分包显示 超时时间: 20 ms 第1 字节 至 末尾 加校验 None
定时发送 100 ms/次 加回车换行
发 送
www.daxia.com S:0 R:29181 COM47 已打开 115200bps,8,1,None,None

CALIBRATED DATA

We modify the last script to get calibrated data.



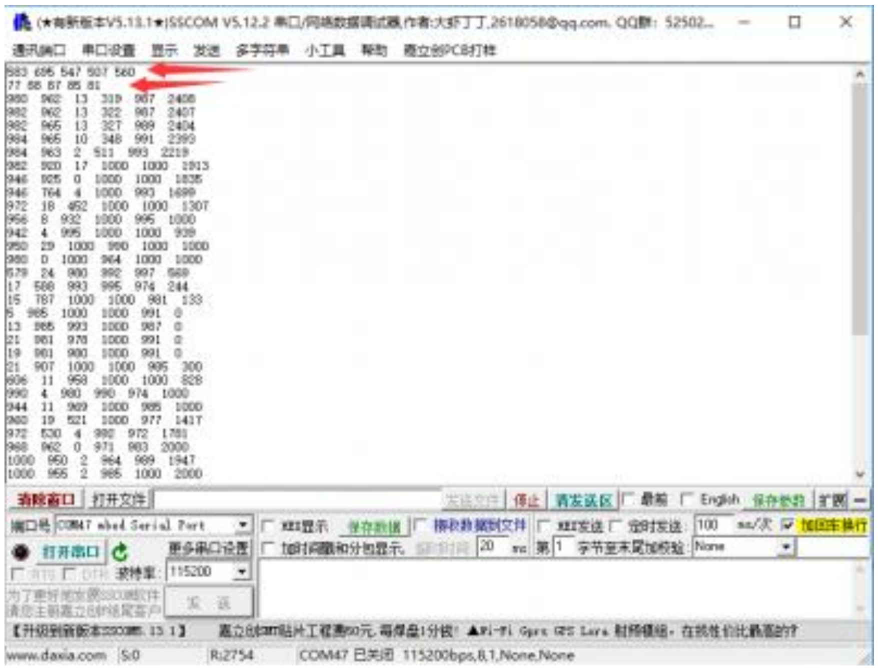
```
on start
    SensorCalibrated
    set Max ▾ to ReadSensorMax
    set Min ▾ to ReadSensorMin
    serial write number Max ▾ get value at 0
    serial write string " "
    serial write number Max ▾ get value at 1
    serial write string " "
    serial write number Max ▾ get value at 2
    serial write string " "
    serial write number Max ▾ get value at 3
    serial write string " "
    serial write number Max ▾ get value at 4
    serial write string " "
    serial write number Min ▾ get value at 0
    serial write string " "
    serial write number Min ▾ get value at 1
    serial write string " "
    serial write number Min ▾ get value at 2
    serial write string " "
    serial write number Min ▾ get value at 3
    serial write string " "
    serial write number Min ▾ get value at 4
    serial write line " "

forever
    set item ▾ to ReadCalibrated
    serial write number item ▾ get value at 0
    serial write string " "
    serial write number item ▾ get value at 1
    serial write string " "
    serial write number item ▾ get value at 2
    serial write string " "
    serial write number item ▾ get value at 3
    serial write string " "
    serial write number item ▾ get value at 4
    serial write string " "
    serial write number ReadLine
    serial write line " "
```

-Download the script to micro:bit

-Put the robot on ground, and make sure the black line is on the center of ground.

After 3s (run the code), robot begin to calibrate, turn left first, then turn right, finally turn back to original place. Calibration is done for accurate tacking data

The calibrated data is outputted as below:



```
583 695 547 507 560
77 88 87 85 81
980  962  13  319  987  2408
982  962  13  322  987  2407
982  965  13  327  989  2404
984  965  10  348  991  2393
984  963  2  511  993  2219
982  920  17  1000  1000  1913
946  925  0  1000  1000  1835
946  764  4  1000  993  1699
972  18  452  1000  1000  1307
956  8  932  1000  995  1000
942  4  995  1000  1000  939
950  29  1000  990  1000  1000
980  0  1000  964  1000  1000
579  24  980  992  997  569
17  588  993  995  974  244
15  787  1000  1000  981  133
5  985  1000  1000  991  0
13  985  993  1000  987  0
21  981  978  1000  991  0
19  981  980  1000  991  0
21  907  1000  1000  985  300
606  11  958  1000  1000  828
990  4  980  990  974  1000
944  11  989  1000  985  1000
980  19  521  1000  977  1417
972  530  4  980  972  1781
988  962  0  971  983  2000
1000  950  2  964  989  1947
1000  965  2  985  1000  2000
```

(★最新版本V5.13.1★)SSCOM V5.12.2 串口/网络数据调试器,作者:大虾丁丁,2618058@qq.com. QQ群: 52502...
通讯端口 串口设置 显示 发送 多字符串 小工具 帮助 嘉立创PCB打样
清除窗口 打开文件 发送文件 停止 清发送区 最前 English 保存参数 扩展
端口号 COM47 mbed Serial Port
打开串口 更多串口设置 波特率: 115200
RTS DTR
HEX显示 保存数据 接收数据到文件 HEX发送 定时发送: 100 ms/次 加回车换行
加时间戳和分包显示 超时时间: 20 ms 第 1 字节至末尾加校验: None
发 送
www.daxia.com S:0 R:2754 COM47 已关闭 115200bps,8,1,None,None

-First line shows the max values of every sensor

-Second line shows the min values of every sensor

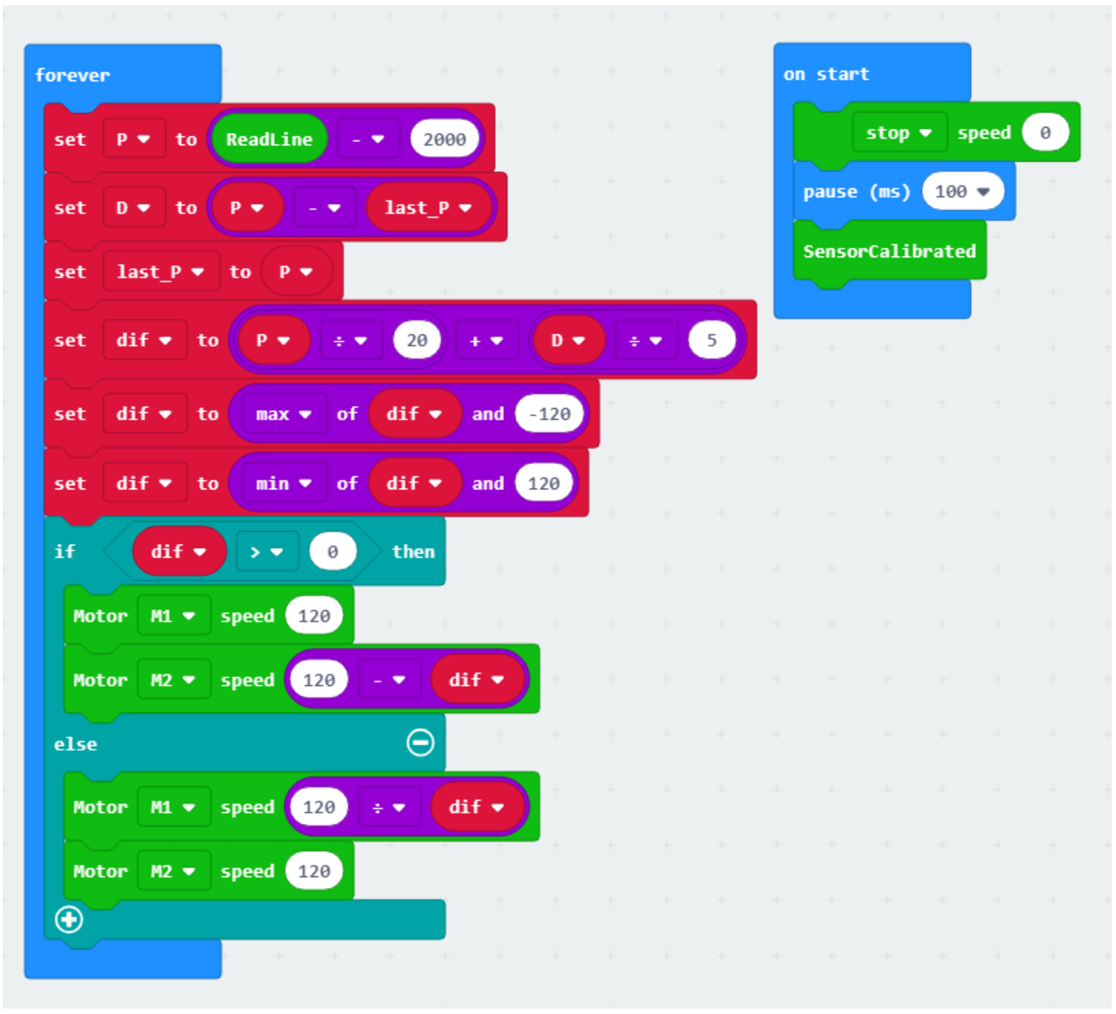
-Flowing the calibrated data. For every line, it shows the calibrated data of sensor 1 to sensor5 and finally the position of black line.

-The calibrated data of sensor is in range 0~1000. Smaller the value, closer the black line

-The calibrated position of black line is in range 0~4000. 0: on the extreme left, 2000: on the center; 4000: on the extreme right.

TRACKING

We have learned how the tracker sensor work, now let’ s create a new project to do the line tracking.



```
forever
    set P to ReadLine - 2000
    set D to P - last_P
    set last_P to P
    set dif to P ÷ 20 + D ÷ 5
    set dif to max of dif and -120
    set dif to min of dif and 120
    if dif > 0 then
        Motor M1 speed 120
        Motor M2 speed 120 - dif
    else
        Motor M1 speed 120 ÷ dif
        Motor M2 speed 120

on start
    stop speed 0
    pause (ms) 100
    SensorCalibrated
```

-P is the tolerance of position. positive: robot deviation to right; negative: robot deviation to left.

- D is the difference between current P and last P, bigger the D, faster the robot response.

-dif is the difference which motor need to change, this value is based on P and D