FREENOVE

Free your innovation

Freenove is an open-source electronics platform.
www.freenove.com

Warning

When you purchase or use Freenove Micro Rover, please note the following:

- This product contains small parts. Swallowing or improper operation can cause serious infections and death. Seek immediate medical attention when the accident happened.
- Do not allow children under 3 years old to play with or near this product. Please place this product in where children under 3 years of age cannot reach.
- Do not allow children lack of ability of safe to use this product alone without parental care.
- Never use this product and its parts near any AC electrical outlet or other circuits to avoid the potential risk of electric shock.
- Never use this product near any liquid and fire.
- Keep conductive materials away from this product.
- Never store or use this product in any extreme environments such as extreme hot or cold, high humidity and etc.
- Remember to turn off circuits when not in use this product or when left.
- Do not touch any moving and rotating parts of this product while they are operating.
- Some parts of this product may become warm to touch when used in certain circuit designs. This is normal. Improper operation may cause excessively overheating.
- Using this product not in accordance with the specification may cause damage to the product.

About

Freenove is an open-source electronics platform. Freenove is committed to helping customer quickly realize the creative idea and product prototypes, making it easy to get started for enthusiasts of programing and electronics and launching innovative open source products. Our services include:

- Electronic components and modules
- Learning kits for Arduino
- Learning kits for Raspberry Pi
- Learning kits for micro:bit
- Learning kits for Technology
- Product customization service
- Robot kits
- Auxiliary tools for creations

Our code and circuit are open source. You can obtain the details and the latest information through visiting the following web sites:
http://www.freenove.com
https://github.com/freenove
Your comments and suggestions are warmly welcomed, and please send them to the following email address:
support@freenove.com
If you have any business matters, please feel free to contact us:
sale@freenove.com

References

This product is named Freenove Micro Rover.

You can download the sketches and references used in this product in the following websites:

http://www.freenove.com

https://github.com/Freenove/Freenove_Micro_Rover

If you have any difficulties, you can send email to technical support for help.

Support

Freenove provides free and quick technical support, including but not limited to:

- Quality problems of products
- Problems in using products
- Questions for learning and technology
- Opinions and suggestions
- Ideas and thoughts

Please send email to:

support@freenove.com

On working day, we usually reply to you within 24 hours.

Copyright

Freenove reserves all rights to this book. No copies or plagiarizations are allowed for the purpose of commercial use.

The code and circuit involved in this product are released as Creative Commons Attribution ShareAlike 3.0. This means you can use them on your own derived works, in part or completely, as long as you also adopt the same license. Freenove brand and Freenove logo are copyright of Freenove Creative Technology Co., Ltd and cannot be used without formal permission.

Contents

Preface (Important)

Here are some additional projects for Rover. **You need to prepare some additional components or devices that are not included in the Rover kit.**

If you do not have these components, you will not be able to complete the next projects.

If you're not interested in additional projects. Never mind. Just leave them alone.

Enjoy your Rover.

Chapter 1 GPIO

Rover expands GPIO8 of micro:bit. You can use it to control some components, such as LED, neopixel, Buzzer, Servo and so on.

Preparation

1. Insert micro:bit into Rover correctly.
2. Install battery into Rover.
3. Turn on Rover power.
4. Connect micro:bit and computer through USB cable.

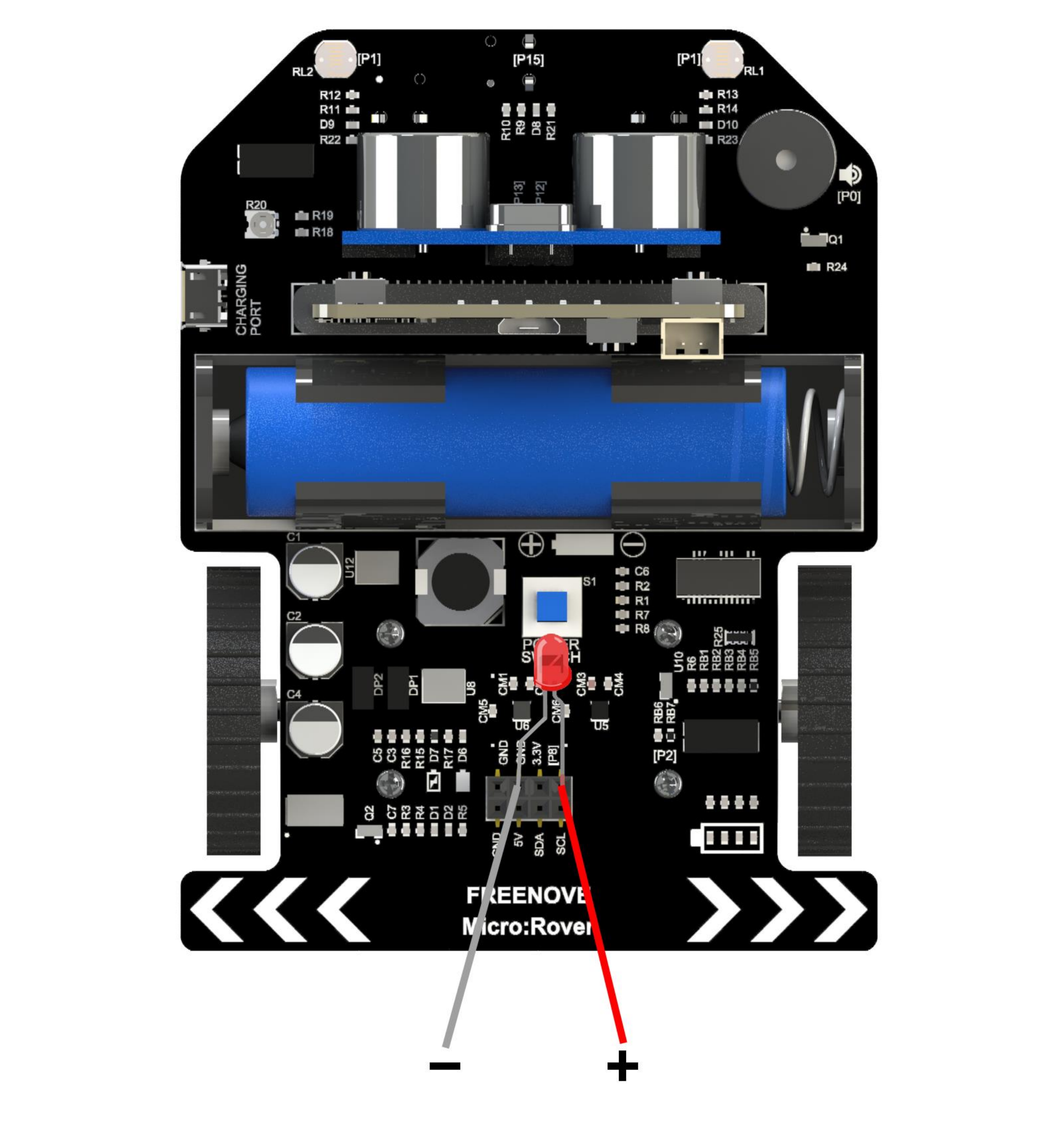
Open web version of makecode or windows 10 app version of makecode.

Additional Components

LED

LED Blink & Breathing

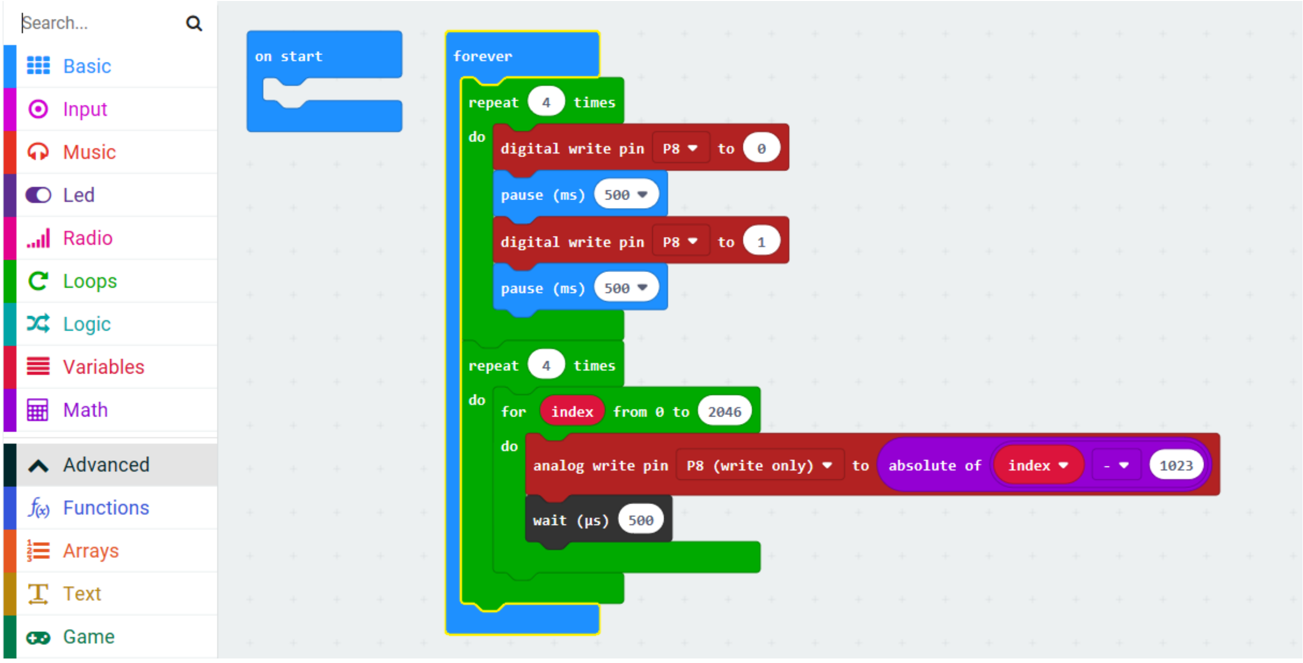
For direct-insert packaged LED, the longer pin is positive and the shorter pin is negative. Connect the positive pole of LED to P8 and the negative pole to GND.



Code

Load code according to the table below:

File type	Path	File name
Hex file	../Expanding Projects/01.1_Blink	microbit-Blink.hex



```
on start

forever
    repeat 4 times
    do
        digital write pin P8 to 0
        pause (ms) 500
        digital write pin P8 to 1
        pause (ms) 500
    repeat 4 times
    do
        for index from 0 to 2046
        do
            analog write pin P8 (write only) to absolute of index - 1023
            wait (µs) 500
```

Download the code to micro:bit. The LED connected to the Rover expansion starts to blink four times, then breathes four times, and then circulates in turn.

Chapter 2 I2C-LCD1602

Rover expands I2C interface of micro:bit. You can use it to control any device with I2C interface, such as I2CLCD1602, BMP180 (barometer) and so on.

Preparation

1. Insert micro:bit into Rover correctly.
2. Install battery into Rover.
3. Turn on Rover power.
4. Connect micro:bit and computer through USB cable.

Open web version of makecode or windows 10 app version of makecode.

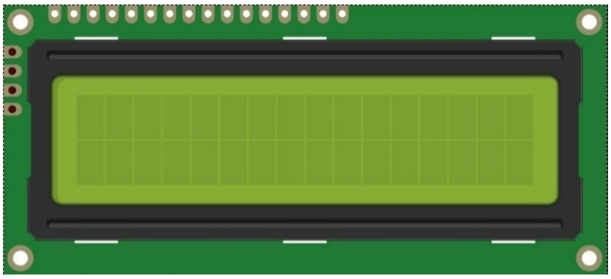
If you choose to load the project by importing Hex file, there is no need to add the Rover extension manually.
(How to import?)

If you choose to drag code manually, you first need to add I2C_LCD160 extension library.
Its GitHub address is below:
https://github.com/makecode-extensions/i2cLCD1602

(How to add extension?)

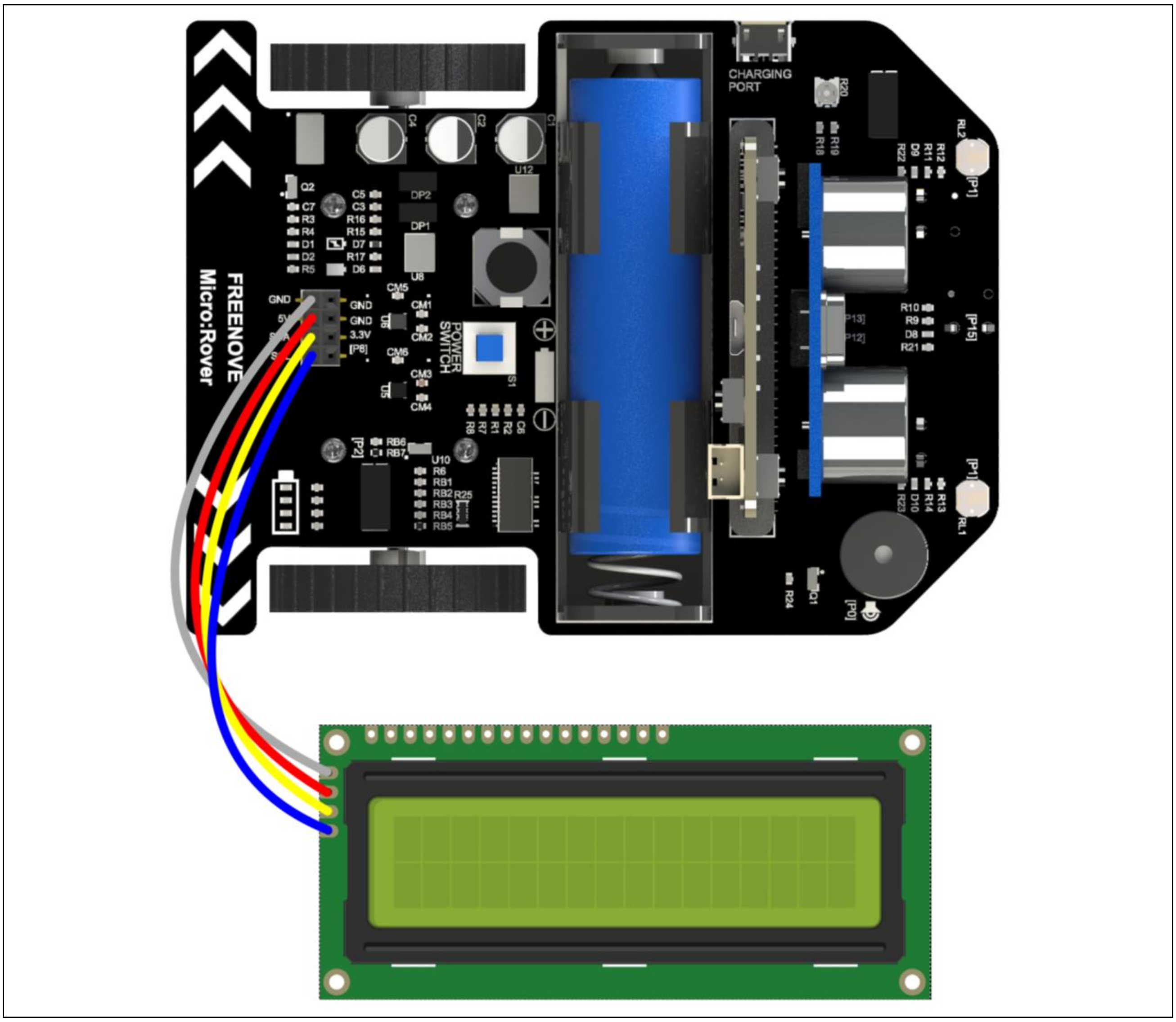
Additional Components

I2CLCD1602



Show Character

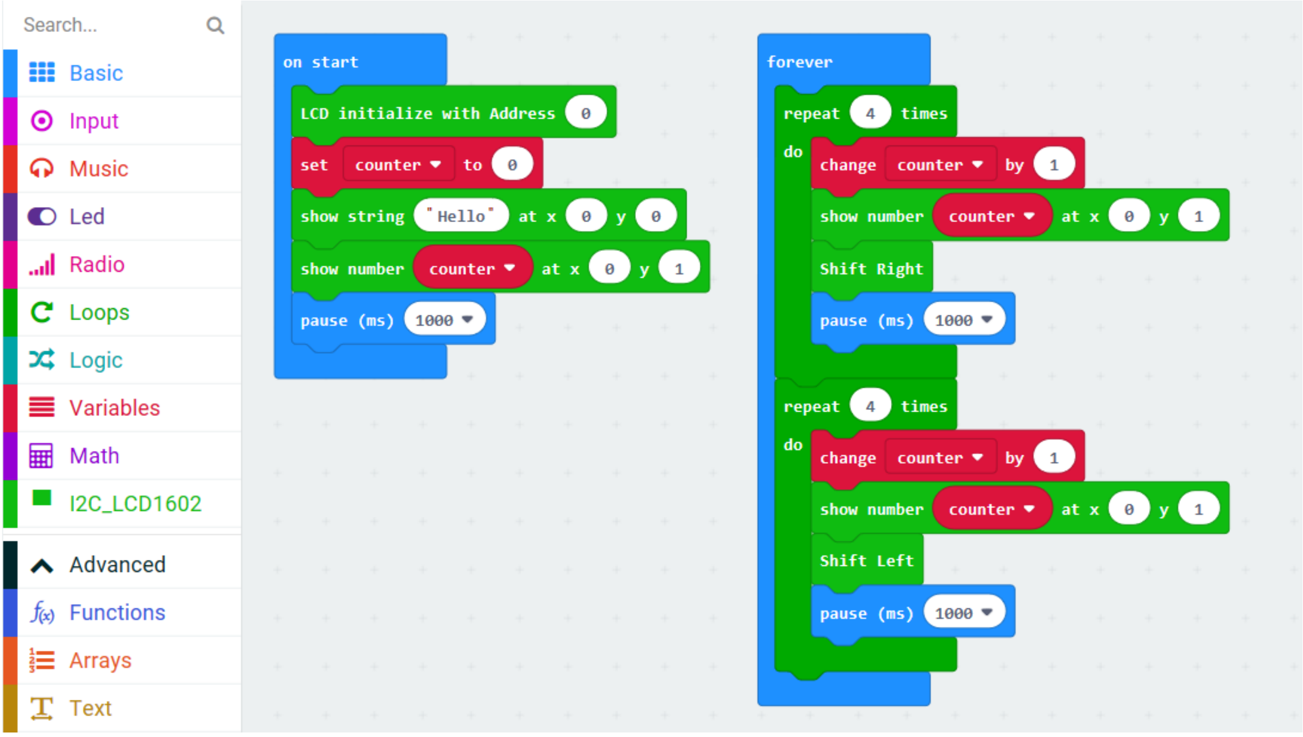
Connect I2CLCD1602 to I2C interface of Rover via jumper wires.



Code

Load code according to the table below:

File type	Path	File name
Hex file	../Expanding Projects/02.1_I2C-LCD1602	microbit-I2C-LCD1602.hex



```
on start
    LCD initialize with Address 0
    set counter to 0
    show string "Hello" at x 0 y 0
    show number counter at x 0 y 1
    pause (ms) 1000

forever
    repeat 4 times
    do
        change counter by 1
        show number counter at x 0 y 1
        Shift Right
        pause (ms) 1000
    repeat 4 times
    do
        change counter by 1
        show number counter at x 0 y 1
        Shift Left
        pause (ms) 1000
```

Download the code to micro:bit. Observe the contents showed on the LCD.

Chapter 3 Radio Control

Different BBC micro:bit can communicate with each other directly through Radio. In this chapter, an additional micro:bit will be used to control Rover.

Preparation

This project uses two micro:bit, one as remote control terminal and another on Rover. So there are two copies of the code for this project. We call remote end micro:bit(A) and the one on Rover micro:bit(B).

1. Insert micro:bit into Rover correctly.
2. Install battery into Rover.
3. Turn on Rover power.

Open web version of makecode or windows 10 app version of makecode.

If you choose to load the project by importing Hex file, there is no need to add the Rover extension manually.

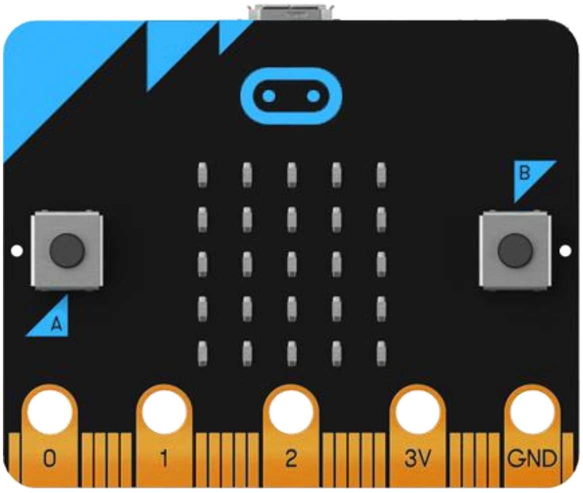
(How to import?)

If you choose to drag code manually, you first need to add Rover extensions.

(How to add Rover extension?)

Additional Components

BBC micro:bit.



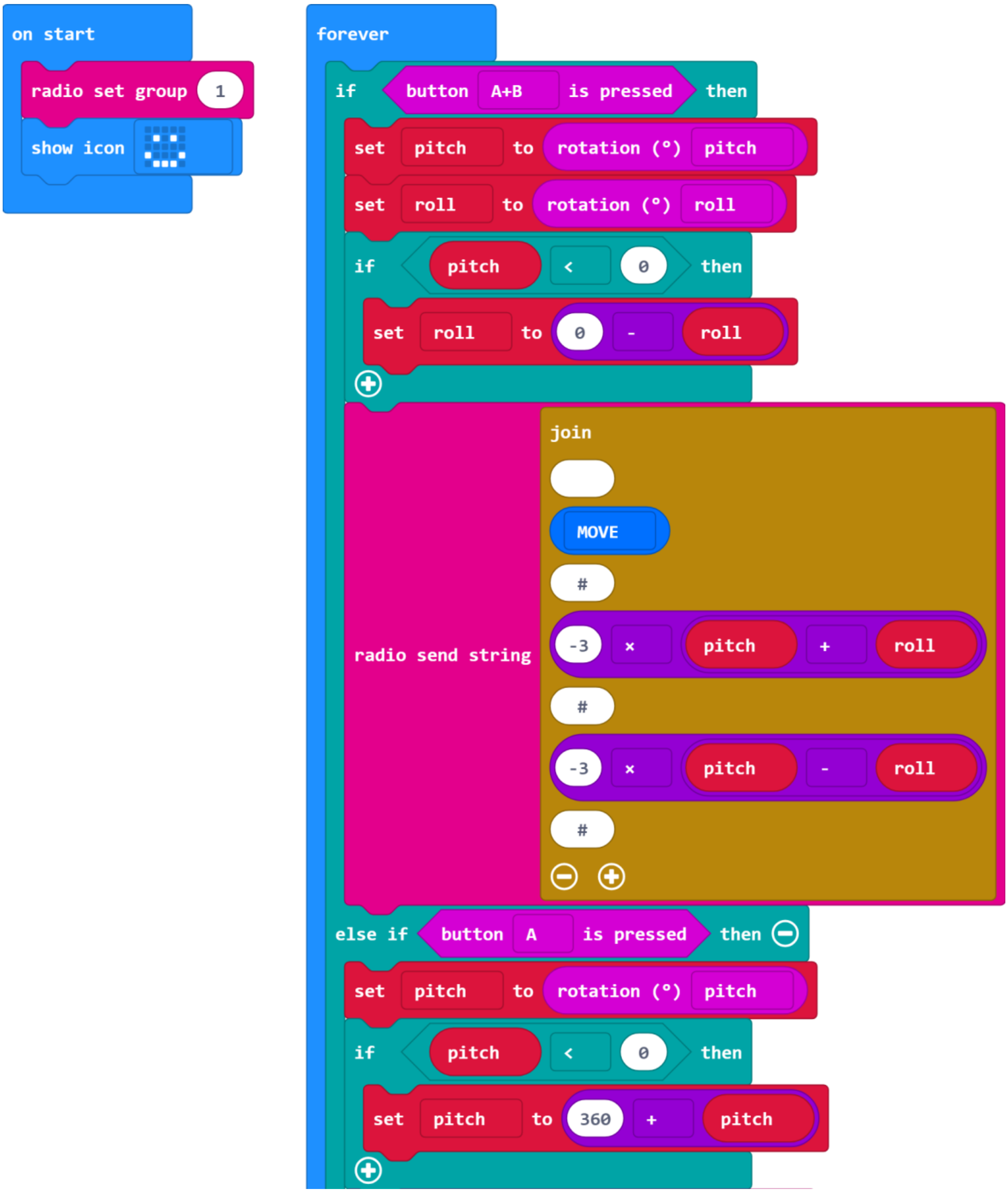
Radio Control

Code for remote

Connect micro:bit(A) with computer via USB.

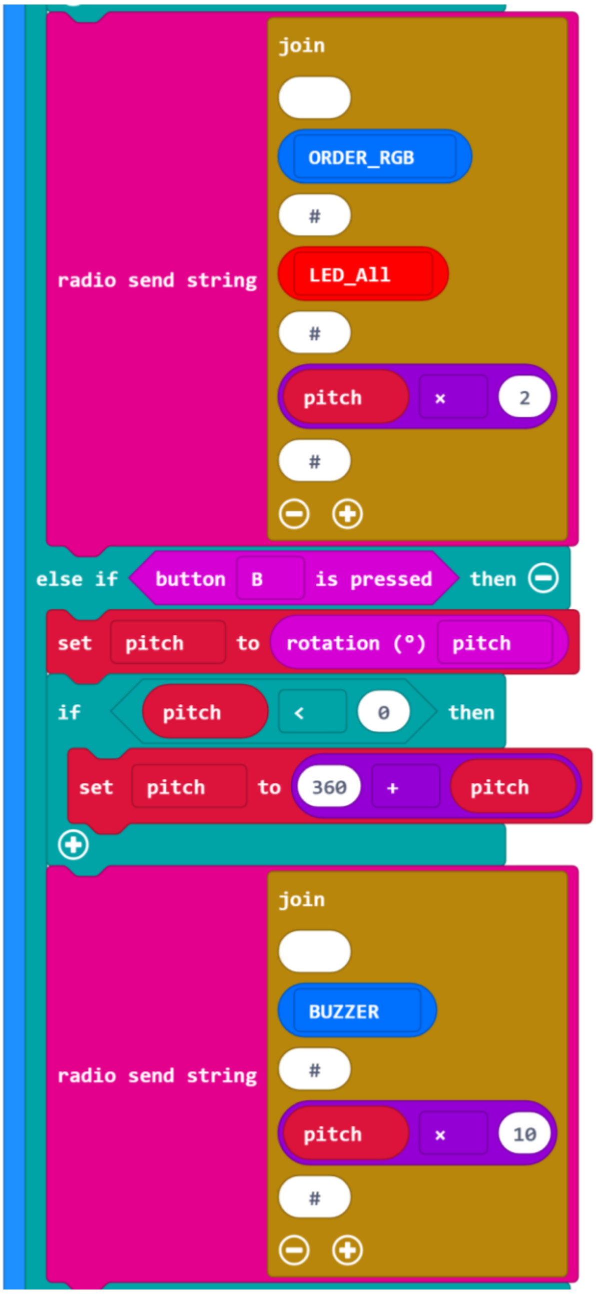
Load code according to the table below:

File type	Path	File name
Hex file	../Expanding Projects/03.1_RadioControl	microbit-Radio-Remote.hex

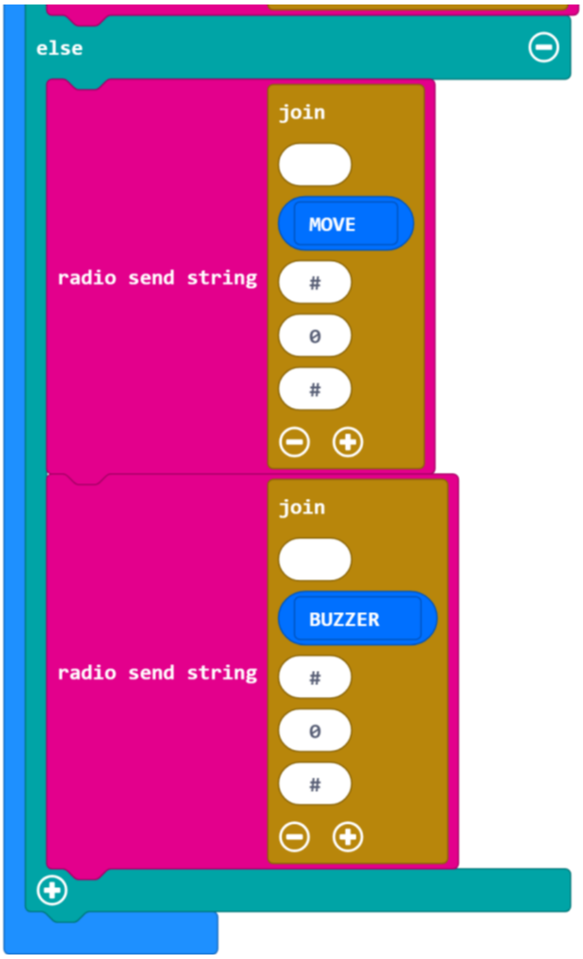


```
on start
    radio set group 1
    show icon

forever
    if button A+B is pressed then
        set pitch to rotation (°) pitch
        set roll to rotation (°) roll
        if pitch < 0 then
            set roll to 0 - roll
        radio send string join
                              ""
                              MOVE
                              #
                              -3 × pitch + roll
                              #
                              -3 × pitch - roll
                              #
    else if button A is pressed then
        set pitch to rotation (°) pitch
        if pitch < 0 then
            set pitch to 360 + pitch
```



```
radio send string join [ ] [ ORDER_RGB ] [ # ] [ LED_All ] [ # ] [ pitch × 2 ] [ # ]
else if button B is pressed then
    set pitch to rotation (°) pitch
    if pitch < 0 then
        set pitch to 360 + pitch
    radio send string join [ ] [ BUZZER ] [ # ] [ pitch × 10 ] [ # ]
```



```
else
    radio send string join [ ] [ MOVE ] [ # ] [ 0 ] [ # ]
    radio send string join [ ] [ BUZZER ] [ # ] [ 0 ] [ # ]
```

Download the code to micro:bit (A). It can be powered by USB or battery.

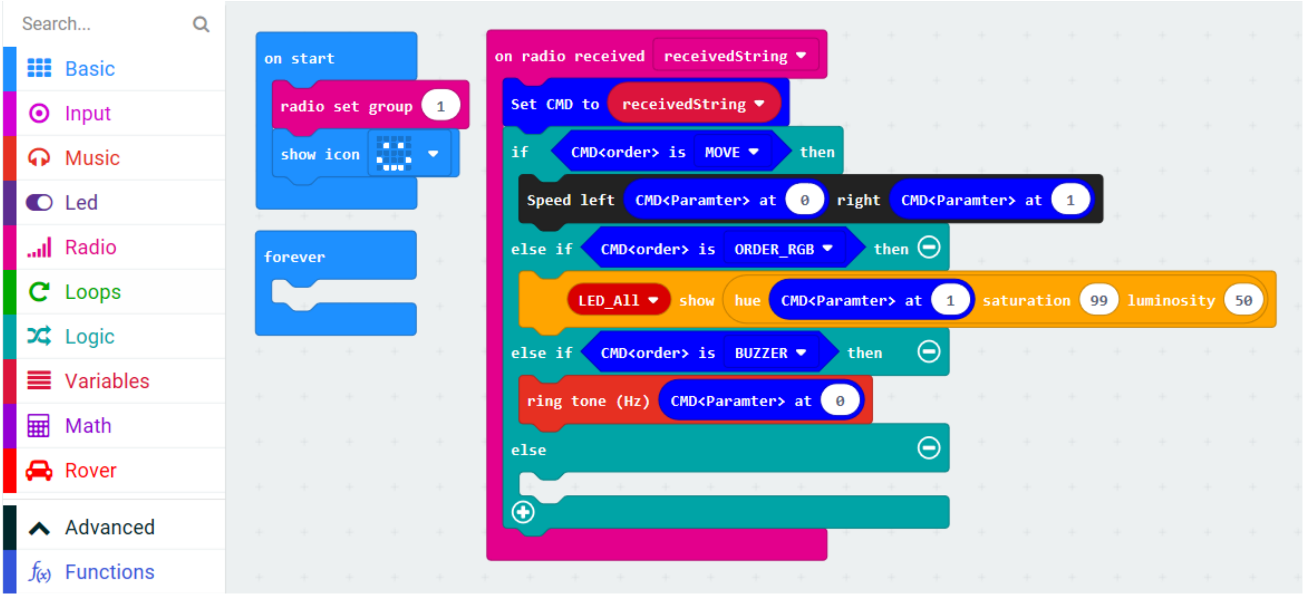
This code uses keys+gravity induction to control Rover.
Key A+gravity PITCH angle is used to set the color of RGB LED.
Key B+gravity PITCH angle is used to set the buzzer frequency.
Key A+B+gravity PITCH and ROLL direction angle is used to control movement of Rover.

Code for Rover

Connect micro:bit (B) to computer via USB cable.

Load code according to the table below:

File type	Path	File name
Hex file	../Expanding Projects/03.1_RadioControl	microbit-Radio-Rover.hex



```
on start
    radio set group 1
    show icon

forever

on radio received receivedString
    Set CMD to receivedString
    if CMD<order> is MOVE then
        Speed left CMD<Paramter> at 0 right CMD<Paramter> at 1
    else if CMD<order> is ORDER_RGB then
        LED_All show hue CMD<Paramter> at 1 saturation 99 luminosity 50
    else if CMD<order> is BUZZER then
        ring tone (Hz) CMD<Paramter> at 0
    else
```

Download the code to micro:bit(B).

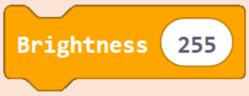
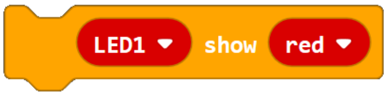
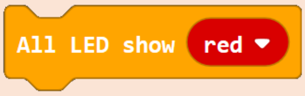
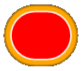
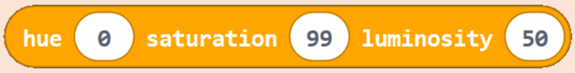
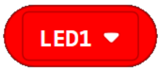
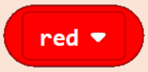
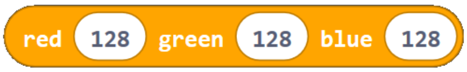
Then you can use micro:bit(A) to control Rover.

Appendix

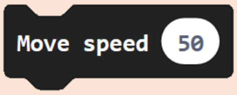
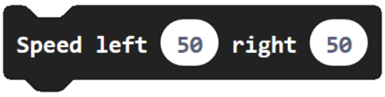
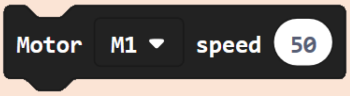
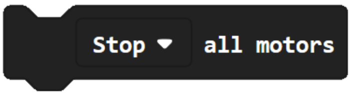
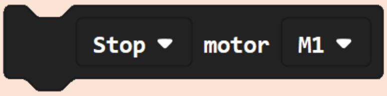
The appendix section introduces the role of each block in Rover Extension.

Rover Extension divides all blocks into four groups according to their functions: [LED, Motors, Sensors, Commands].

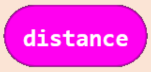
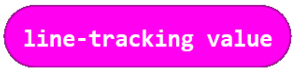
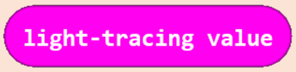
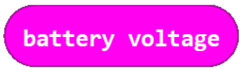
LEDs

Block	Function
 Brightness 255	Set the brightness of all RGB LEDs on the Rover.
 LED1 show red	Set specific RGB LED to a specific color.
 All LED show red	Set all RGB LED to a specific color.
	Color palette provides 16 kinds of colors.
 hue 0 saturation 99 luminosity 50	The HSL color picking model returns the RGB color value.
 LED1	RGB LED combination selection list.
 red	Common color list provides 10 kinds of colors
 red 128 green 128 blue 128	The RGB color picking model returns the RGB color value.

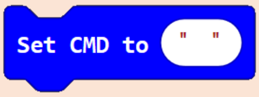
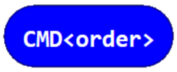
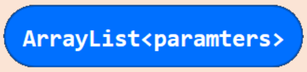
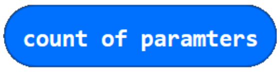
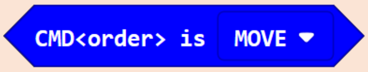
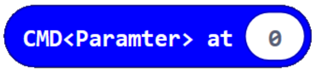
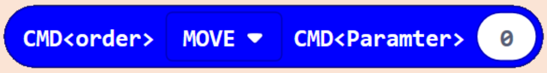
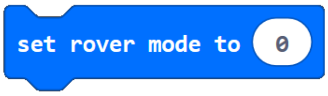
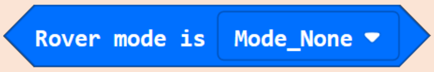
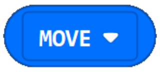
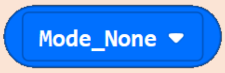
Motors

Block	Function
 Move speed 50	Set the two motors at the same speed to makes the Rover move forward or backward. The positive value is for forward and the negative value is for backward.
 Speed left 50 right 50	Set the speed of the left and right motors to make the Rover move or turn.
 Motor M1 ▾ speed 50	Set the speed of only one motor (M1 or M2).
 Stop ▾ all motors	Make two motors stop or brake at the same time.
 Stop ▾ motor M1 ▾	Make only one motor (M1 or M2) stop or brake.

Sensors

Block	Function
 distance	Start the ultrasonic ranging module and return the measured distance. This block is a time-consuming block. If you use this block multiple times in a short time, you need use a variable to save the returned distance value.
 line-tracking value	Returns the value of the line-tracking Sensor.
 light-tracing value	Returns the value of the light-tracing Sensor.
 battery voltage	Returns the battery voltage value.

Commands

Block	Function
 Set CMD to " "	Set CMD as a string and parse it into CMD<order> and CMD<parameters>. And store them in CMD<order> and ArrayList<paramters> accordingly.
 CMD<order>	Return the parsed command.
 ArrayList<paramters>	Return an array in which parsed parameters are stored.
 count of paramters	Returns the number of parsed parameters.
 CMD<order> is MOVE ▾	Determine whether CMD < order > is the currently specified command.
 CMD<Paramter> at 0	Gets the specified parameter in the parameter array.
 CMD<order> MOVE ▾ CMD<Paramter> 0	Combine a specified command with a specified parameter. It is used when sending a command.
 set rover mode to 0	Set a variable to represent the mode of Rover.
 Rover mode is Mode_None ▾	Judge if the Rover mode variable is the specified mode.
 MOVE ▾	Command/action list.
 Mode_None ▾	Rover mode list.

What's next?

Thanks for your reading.

This tutorial is all over here. If you find any mistakes, missions or you have other ideas and questions about contents of this tutorial or the kit and etc, please feel free to contact us, and we will check and correct it as soon as possible.

If you want to learn more about micro:bit, Arduino, Raspberry Pi, smart cars, robots and other interesting products in science and technology, please continue to focus on our website. We will continue to launch cost-effective, innovative and exciting products.

Thank you again for choosing Freenove products.