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

Contents

Preface

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

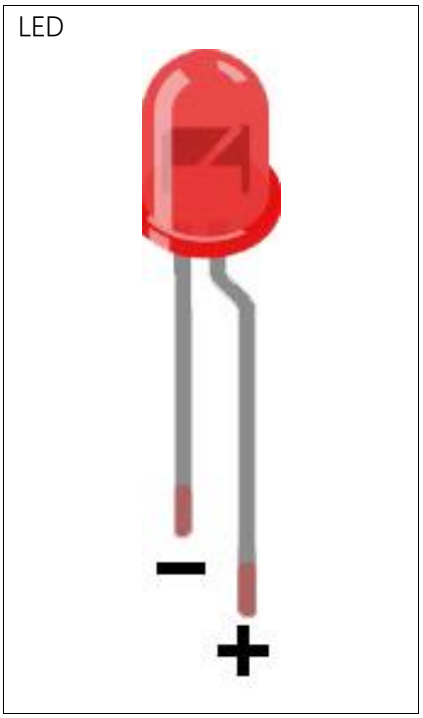
Chapter 1 GPIO

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

Preparation

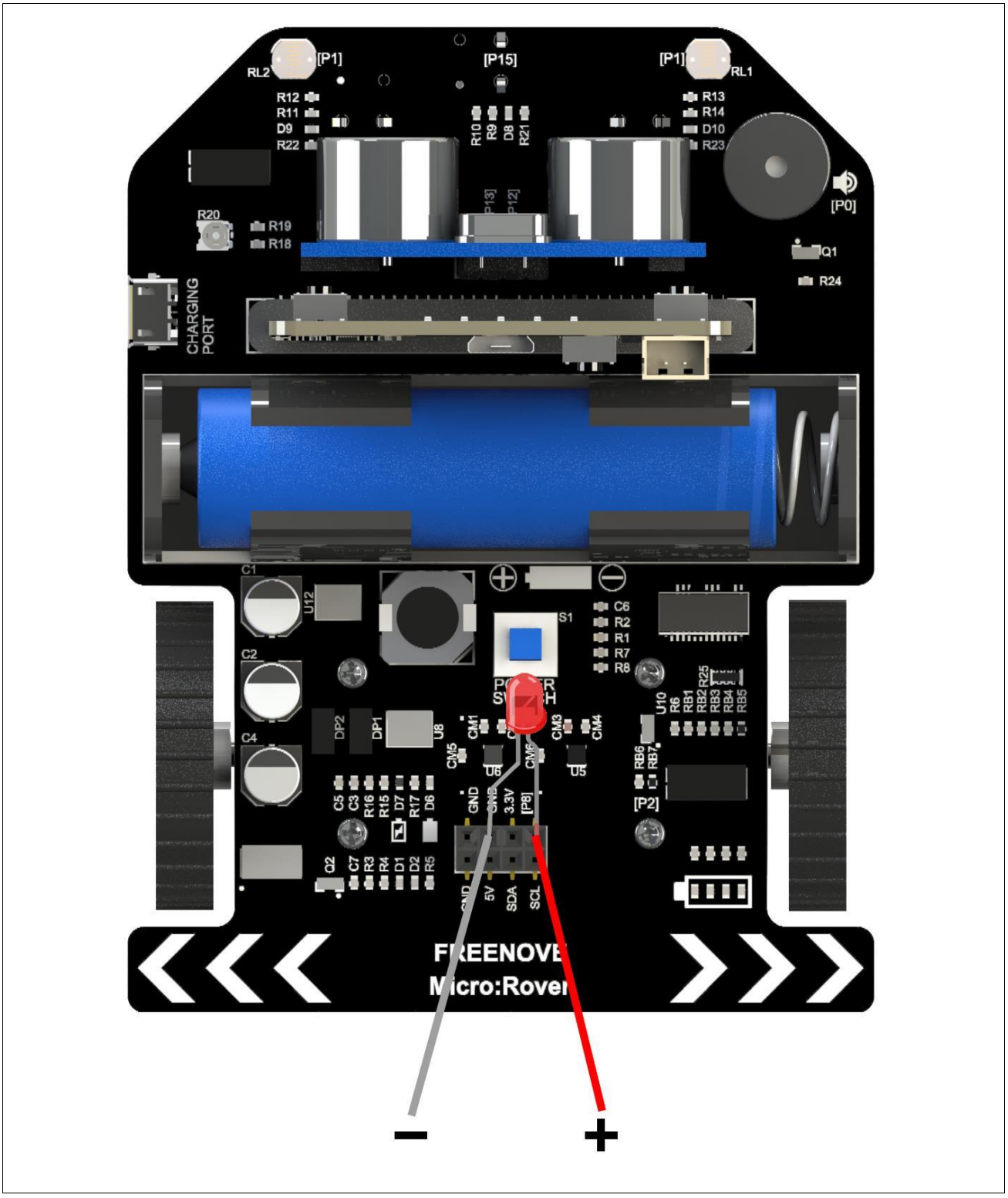
1. Insert micro:bit into Rover correctly.
2. Install battery into Rover.
3. Turn ON Rover power.
4. Connect micro:bit and computer through USB cable.

Additional Components



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.

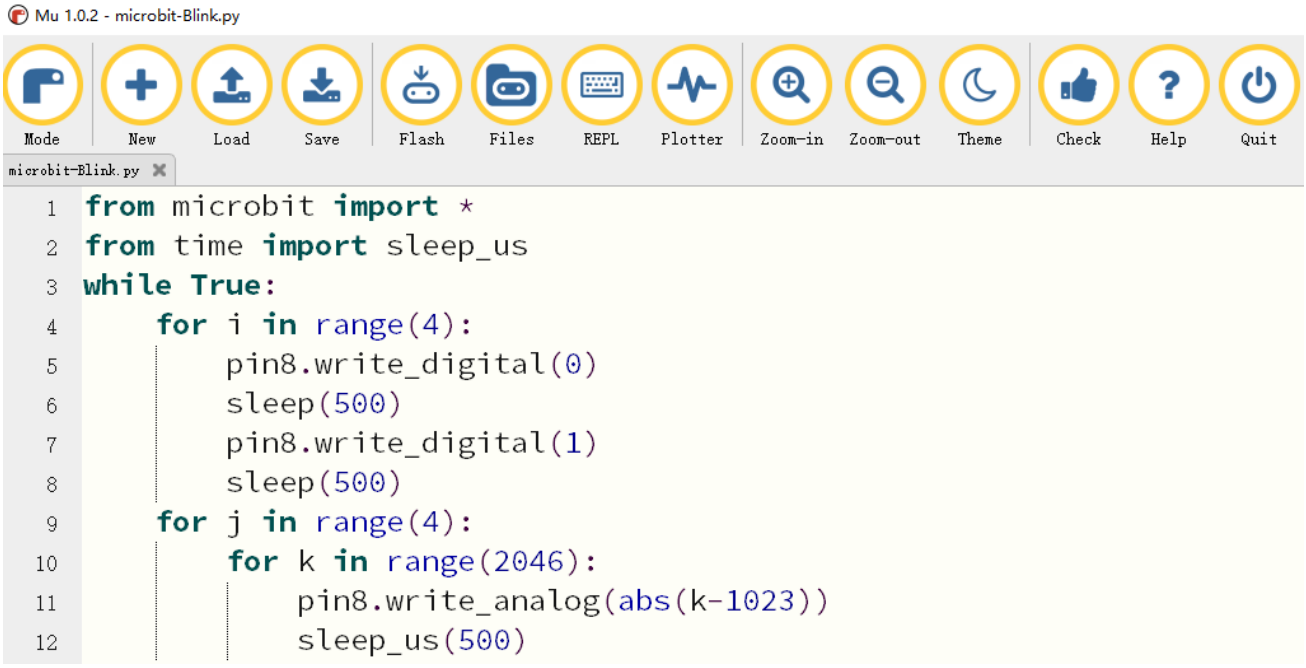


Code

Open the "microbit-Blink.py" with the Mu software. The path to the code is as follows:

File type	Path	File name
Python file	.. /Extended Python Projects/ 01.1_Blink	microbit-Blink.py

After the loading completes, the interface of Mu is as shown below:



Download the code into micro:bit. The LED will flash four times and then breathe four times, which repeats endlessly.

The following is the program code:

```
from microbit import *
from time import sleep_us
while True:
    for i in range(4):
        pin8.write_digital(0)
        sleep(500)
        pin8.write_digital(1)
        sleep(500)
    for j in range(4):
        for k in range(2046):
            pin8.write_analog(abs(k-1023))
            sleep_us(500)
```

Chapter 2 I2C-LCD1602

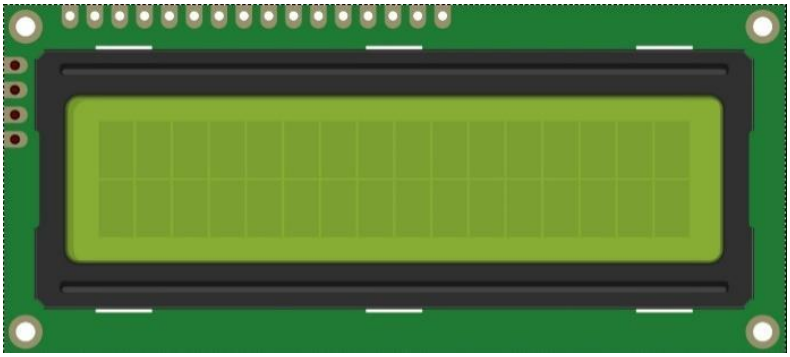
Rover expands I2C interface of micro:bit. You can use it to control any device with I2C interface, such as I2CLCD1602, BMP180 (barometer) and so on

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.

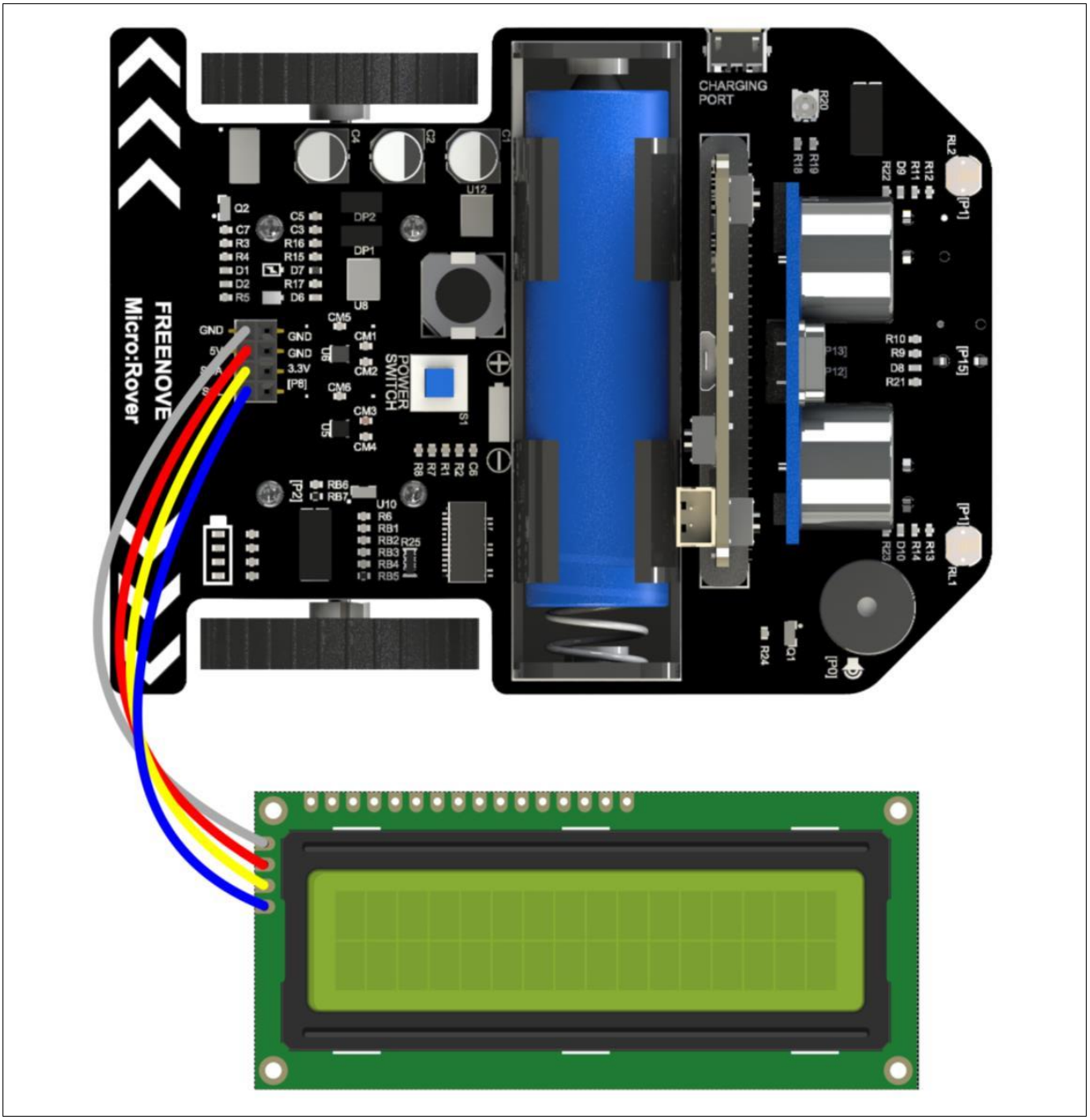
Additional Components

I2CLCD1602



Display Character

Connect I2CLCD1602 to I2C interface of Rover via jumper wires.

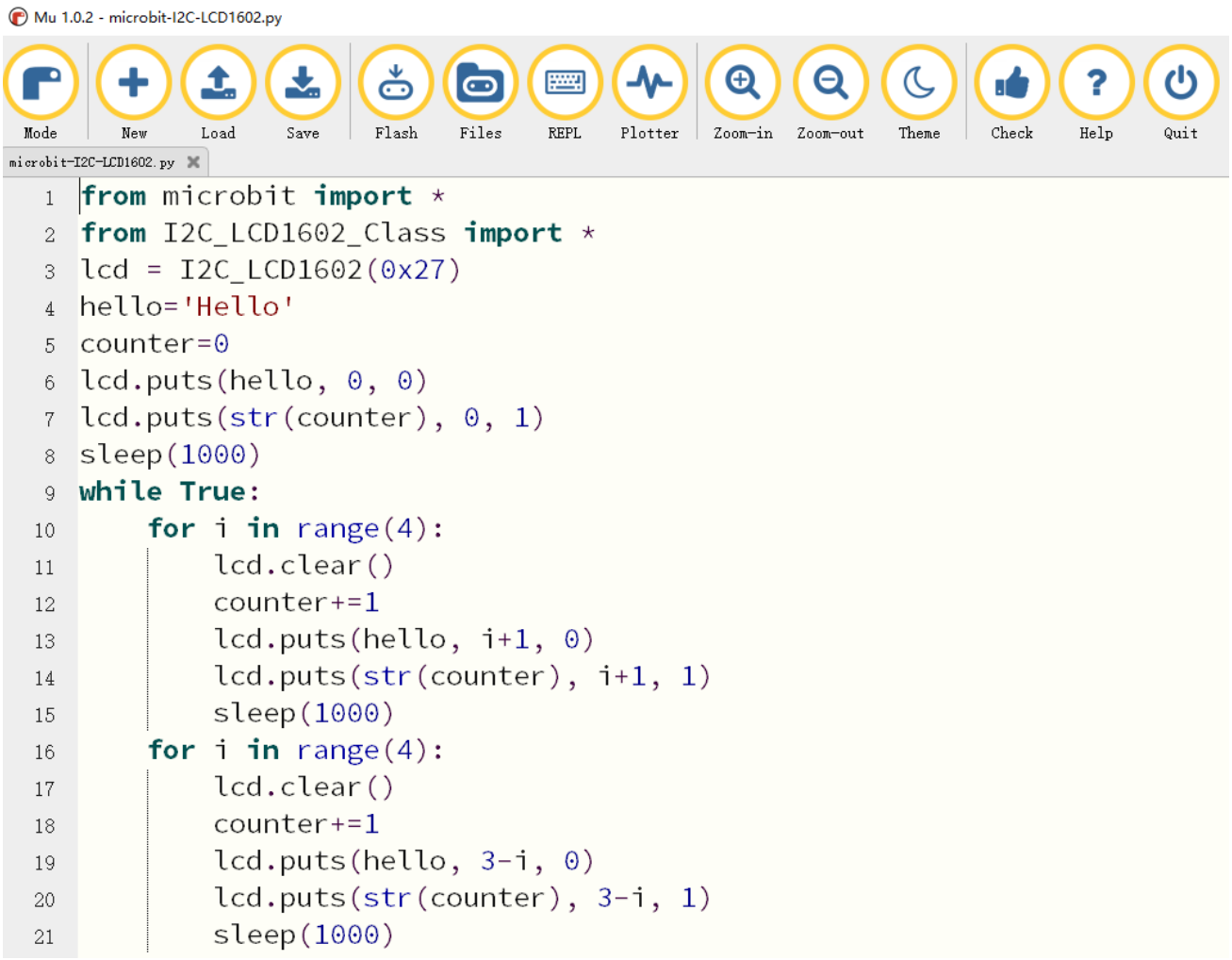


Code

Open the "microbit-I2C-LCD1602.py" with the Mu software. The path of the code is as follows:

File type	Path	File name
Python file	../Extended Python Projects/ 02.1_I2C-LCD1602	microbit-I2C-LCD1602.py

After the loading completes, the interface of Mu is as shown below:



Before downloading the program, import the "I2C_LCD1602_Class.py" file into micro:bit. The path of the code is as follows:

File type	Path	File name
Python file	.. / Extended Python Projects / Libraries	I2C_LCD1602_Class.py

After the import is completed, click the "Flash" button to download the code to micro:bit and observe content displayed on the LCD screen.

The following is the program code:

```
from microbit import *
from I2C_LCD1602_Class import *
lcd = I2C_LCD1602(0x27)
hello='Hello'
counter=0
lcd.puts(hello, 0, 0)
lcd.puts(str(counter), 0, 1)
sleep(1000)
while True:
    for i in range(4):
        lcd.clear()
        counter+=1
        lcd.puts(hello, i+1, 0)
        lcd.puts(str(counter), i+1, 1)
        sleep(1000)
    for i in range(4):
        lcd.clear()
        counter+=1
        lcd.puts(hello, 3-i, 0)
        lcd.puts(str(counter), 3-i, 1)
        sleep(1000)
```

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.

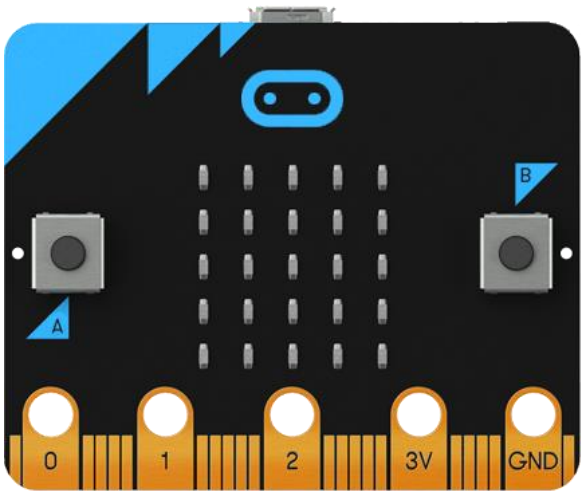
Preparation

This project uses two micro:bit, one as remote control terminal and the other inserted into 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.
4. Connect micro:bit and computer through USB cable.

Additional Components

BBC micro:bit



Radio Control

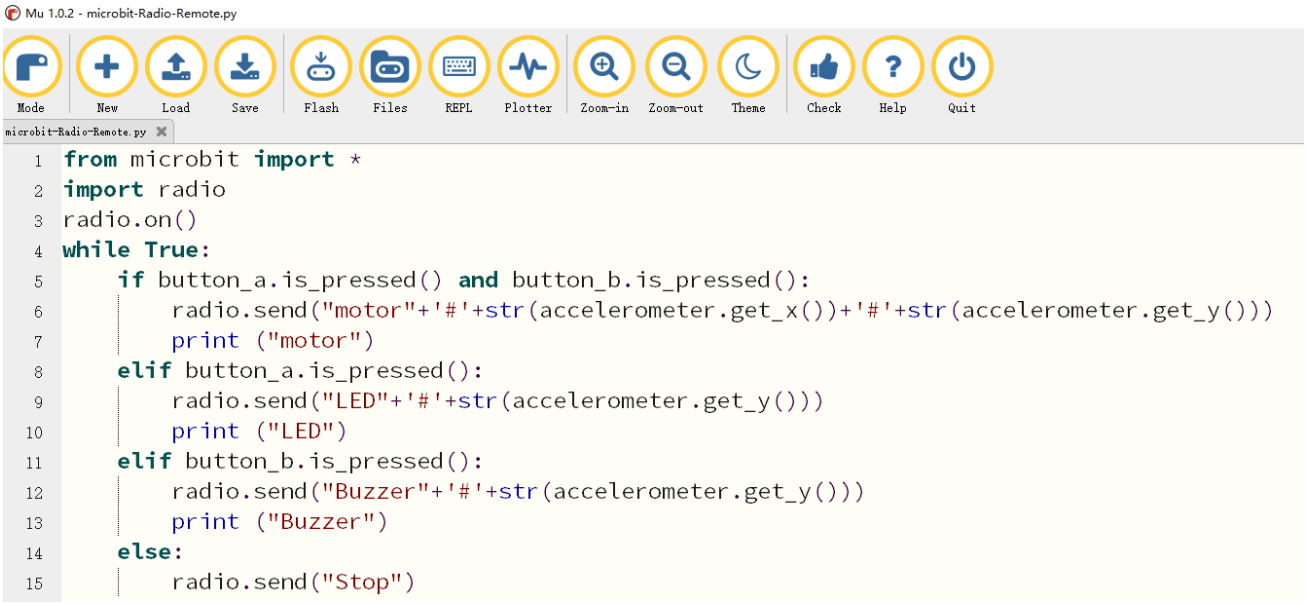
Code for remote

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

Open the "microbit-Radio-Remote.py" with the Mu software. The path of the code is as follows:

File type	Path	File name
Python file	../Extended Python Projects/ 03.1_RadioControl	microbit-Radio-Remote.py

After the loading completes, the interface of Mu is as shown below:




```
1  from microbit import *
2  import radio
3  radio.on()
4  while True:
5      if button_a.is_pressed() and button_b.is_pressed():
6          radio.send("motor"+'#'+str(accelerometer.get_x())+'#'+str(accelerometer.get_y()))
7          print ("motor")
8      elif button_a.is_pressed():
9          radio.send("LED"+'#'+str(accelerometer.get_y()))
10         print ("LED")
11     elif button_b.is_pressed():
12         radio.send("Buzzer"+'#'+str(accelerometer.get_y()))
13         print ("Buzzer")
14     else:
15         radio.send("Stop")
```

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.

The following is the program code:

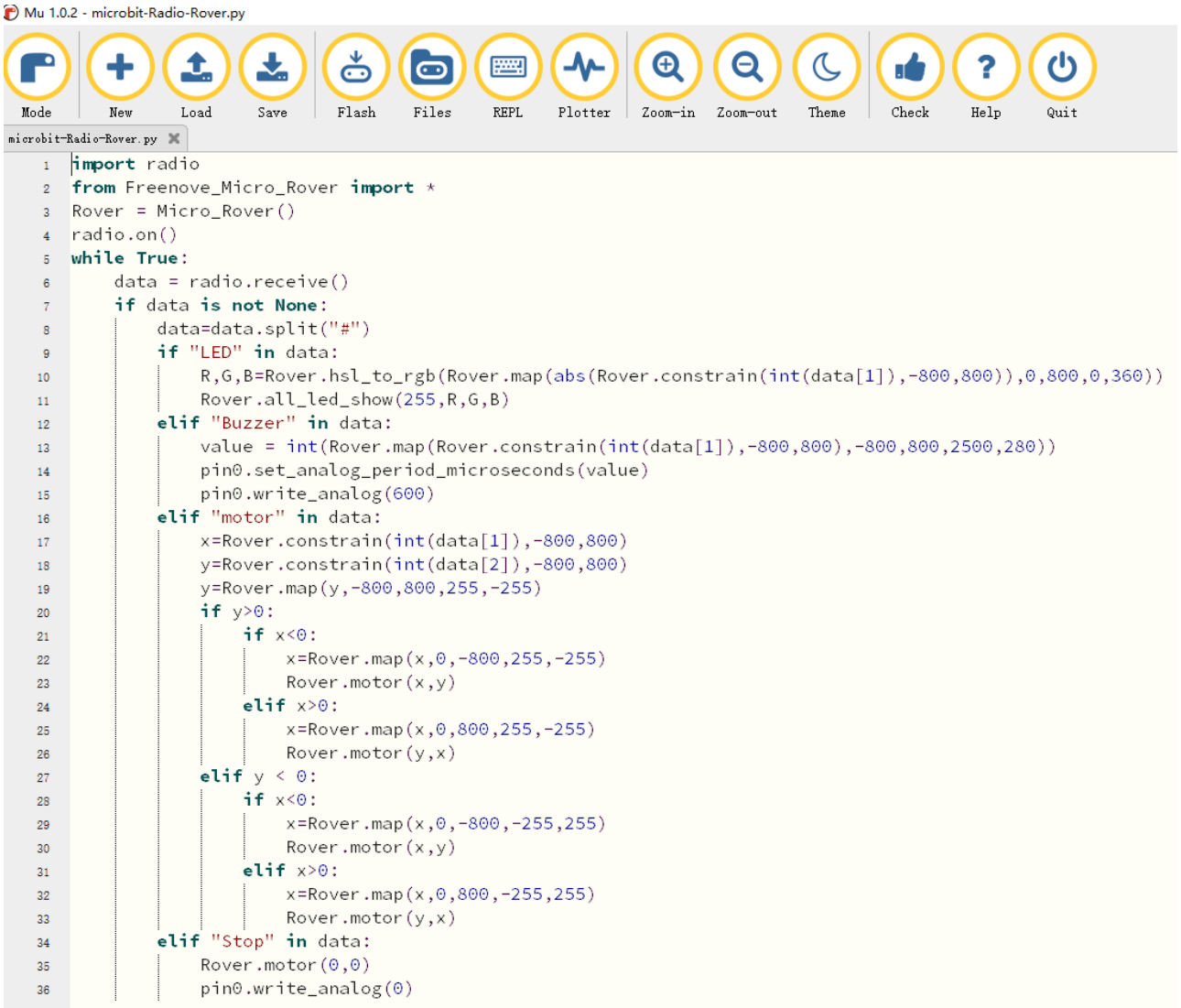
```
from microbit import *
import radio
radio.on()
while True:
    if button_a.is_pressed() and button_b.is_pressed():
        radio.send("motor"+'#'+str(accelerometer.get_x())+'#'+str(accelerometer.get_y()))
        print ("motor")
    elif button_a.is_pressed():
        radio.send("LED"+'#'+str(accelerometer.get_y()))
print ("LED")
    elif button_b.is_pressed():
        radio.send("Buzzer"+'#'+str(accelerometer.get_y()))
        print ("Buzzer")
    else:
        radio.send("Stop")
```

Code for Rover

Open the "microbit-Radio-Rover.py" with the Mu software. The path of the code is as follows:

File type	Path	File name
Python file	../Extended Python Projects/ 03.1_RadioControl	microbit-Radio-Rover.py

After the loading completes, the interface of Mu is as shown below:



Import “Freenove_Micro_Rover_2.py” into micro:bit. The path of the code is as follows:

File type	Path	File name
Python file	.. / Extended Python Projects / Libraries	Freenove_Micro_Rover_2.py

After the import is completed, click the "Flash" button to download the code to micro:bit, and you can use micro:bit(A) to control Rover.

Note: The Freenove_Micro_Rover_2.py file does not contain the get_distance() function associated with the ultrasonic module. To use the ultrasonic module, import the Freenove_Micro_Rover.py file

The following is the program code:

```
import radio
from Freenove_Micro_Rover import *
Rover = Micro_Rover()
radio.on()
while True:
    data = radio.receive()
    if data is not None:
        data=data.split("#")
        if "LED" in data:
            R,G,B=Rover.hsl_to_rgb(Rover.map(abs(Rover.constrain(int(data[1]),-
800,800)),0,800,0,360))
            Rover.all_led_show(255,R,G,B)
        elif "Buzzer" in data:
            value = int(Rover.map(Rover.constrain(int(data[1]),-800,800),-800,800,2500,280))
            pin0.set_analog_period_microseconds(value)
            pin0.write_analog(600)
        elif "motor" in data:
            x=Rover.constrain(int(data[1]),-800,800)
            y=Rover.constrain(int(data[2]),-800,800)
            y=Rover.map(y,-800,800,255,-255)
            if y>0:
                if x<0:
                    x=Rover.map(x,0,-800,255,-255)
                    Rover.motor(x,y)
                elif x>0:
                    x=Rover.map(x,0,800,255,-255)
                    Rover.motor(y,x)
            elif y < 0:
                if x<0:
                    x=Rover.map(x,0,-800,-255,255)
                    Rover.motor(x,y)
                elif x>0:
                    x=Rover.map(x,0,800,-255,255)
                    Rover.motor(y,x)
        elif "Stop" in data:
            Rover.motor(0,0)
            pin0.write_analog(0)
```

What's Next?

THANK YOU for participating in this learning experience!

We have reached the end of this Tutorial. If you find errors, omissions or you have suggestions and/or questions about the Tutorial or component contents of this Kit, please feel free to contact us: support@freenove.com
We will make every effort to make changes and correct errors as soon as feasibly possible and publish a revised version.

If you want to learn more about Arduino, Raspberry Pi, Smart Cars, Robotics and other interesting products in science and technology, please continue to visit our website. We will continue to launch fun, cost-effective, innovative and exciting products.
http://www.freenove.com/

Thank you again for choosing Freenove products.