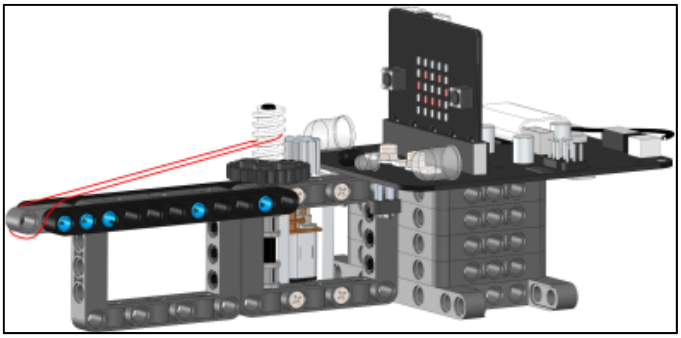
Python course Sniper---“Infrared remote control”

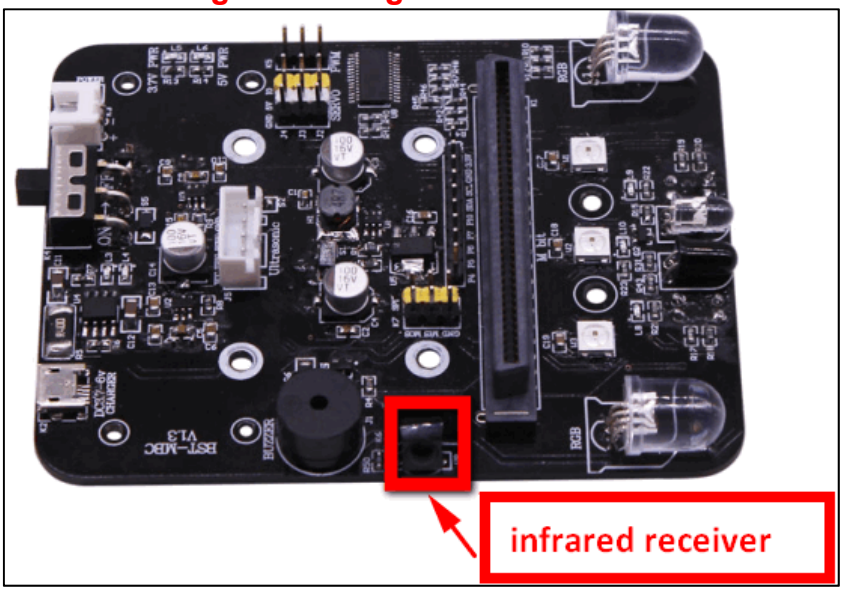


Note：

1. When performing infrared remote control, the remote controller should face the infrared receiver on the expansion board.

2. There is a plastic piece on the bottom of the infrared remote controller that needs to be taken down for normal use.

3. The infrared light emitted by the infrared remote controller and the infrared receiver is invisible to the human eye. It can be seen under the camera without filtering infrared light.



1. Learning goals

After downloading the program, open the power switch of the Sniper and press the button of the infrared remote controller. The Sniper will have corresponding action. On the Infrared remote controller, small light button, add button, subtract button to control the color of the colorful lights, the red power button to turn off light, 1~7 button represents the music do, re, mi, fa, sol, la, si. The front, back buttons control the direction of motor rotation to eject rubber band. 0 and 8, 9 are used to control the image of the dot matrix screen.

2. Preparation before class

We needs to be ready:
Building Block Sniper *1
Infrared remote controller *1
USB data cable *1

3. Programming

Program analysis：

1) Import library file we need. As shown below:

```
# -*- coding: utf-8-*-# Encoding cookie added by Mu Editor
# from microbit import *
from microbit import display, sleep, pin8, pin16
import buildingbit
import music
import neopixel
```

2) **display.off():** Turn off LED dot matrix. When using infrared avoid sensor, we must turn off the dot-matrix display, otherwise pin reuse will cause conflicts.

np = neopixel.NeoPixel(pin16, 3): Initialize the body programmable RGB lights. The first parameter is the RGB light pins, and the second parameter is the number of RGB lights.

np.clear(): Clear RGB lights

buildingbit.init_IR(pin8): Initialize the infrared receiver, the parameters are the pins of the infrared receiver;

buildingbit.car_run(0, 0, 0): Make motor stop

speed: Speed of car

a: Record data

As shown below:

```
display.off()
np = neopixel.NeoPixel(pin16, 3)
np.clear()
buildingbit.init_IR(pin8)
buildingbit.car_run(0, 0, 0)

speed = 100
a = 0
```

3) Read the data received by the infrared receiver, and only take the upper 8 bits.

```
while True:
    # print(hex(buildingbit.get_IR(pin8)))
    value = buildingbit.get_IR(pin8)
    value = value >> 8
```

The key code value of the remote control is shown in the following figure:

4) First,we need to process two special values. By default, you will always receive the value-0x01. If you press and hold a key, you will receive 0xff. The variable a is to solve the problem that you may receive -0x01 when you press and hold the key. The code value of each key.

```
23      # default
24      if value == -0x01:
25          a = a + 1
26          if (a > 3):
27              buildingbit.car_stop()
28              a = 0
29      # long pressed
30      elif value == 0xff:
31          a = 0
```

5) The code value of the red power button is 0x00, we set **buildingbit.car_HeadRGB (0, 0, 0)** is to turn off the headlights of Sniper.

```
33      # off
34      if value == 0x00:
35          buildingbit.car_HeadRGB(0, 0, 0)
```

6) Press Up button, control sniper launch

```
36      # up
37      elif value == 0x80:
38          buildingbit.car_run(speed, speed, 0)
```

7) Press horn button, car will whistle.

```
45      # buzzer
46      elif value == 0xa0:
47          music.pitch(698)
48          sleep(400)
49          music.pitch(0)
```

The other keys are written in the same way, each button corresponds to a function.

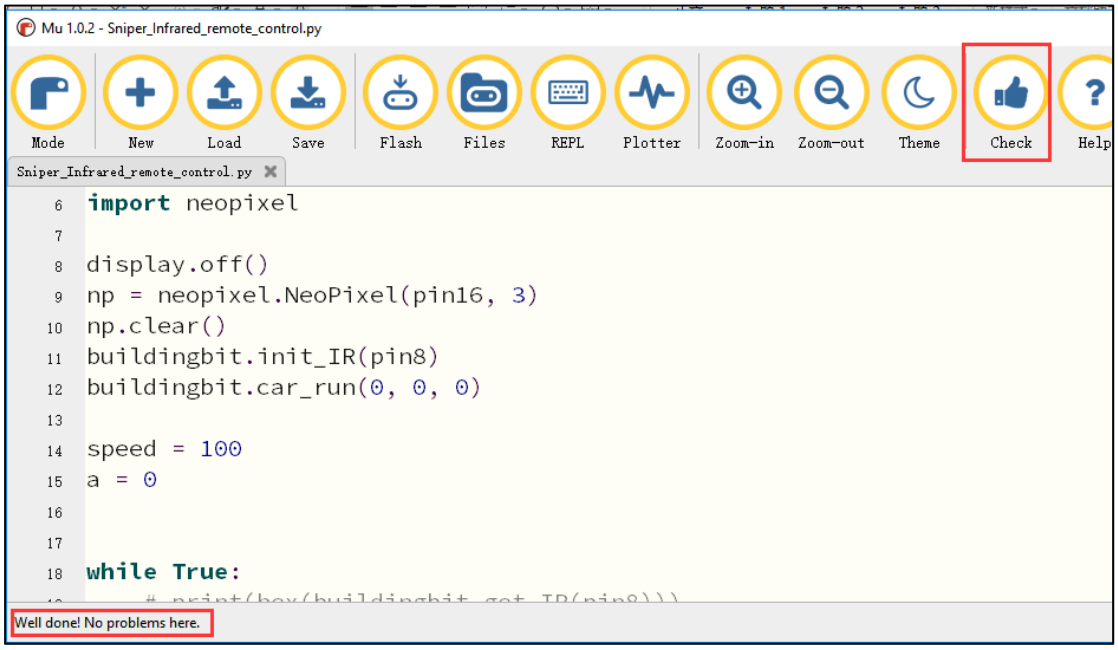
About Code :

Please use the MU software to open the Sniper_Infrared_remote_control.py file we provided.

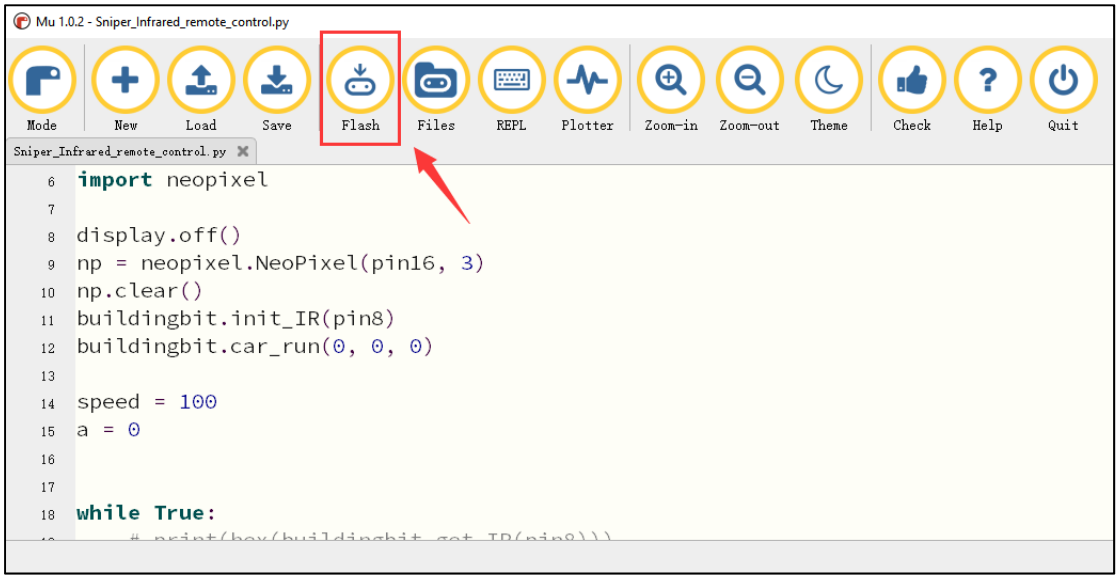
4. Download program

4.1 After programming is complete, plaese connect the computer and the micro:bit board with a Micro USB data cable.

4.2 You need to click the 【**Check**】 button to check if our code has an error. If a line appears with a cursor or an underscore, the program indicating this line is wrong. If there is no cursor or underline, it means that the code is correct, and the bottom left will prompt that the check is OK.



4.3 Click the【Flash】button to download the program to the micro: bit board of the building block sniper.



If the program is wrong or the experimental phenomenon is wrong after downloading, please confirm whether you have downloaded the Buildingbit library hex file we provided to the micro: bit board.

For the specific method of adding library files, please refer to 【1.Preparation before class】---【Python programming】