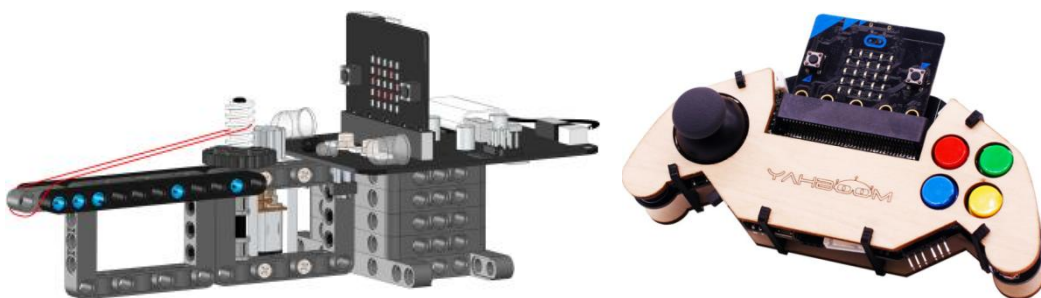


狙击手 micro:bit 手柄遥控



1. 实验目的

下载好程序后，打开狙击手的电源开关和 micro:bit 手柄的电源开关，两者会自动配对。按下 micro:bit 手柄遥控的按键，狙击手就会有相应的动作。遥控器右边的按键是开灯按键，往下按摇杆是关灯。摇杆往前后推是控制狙击手的夹子发射和回转。

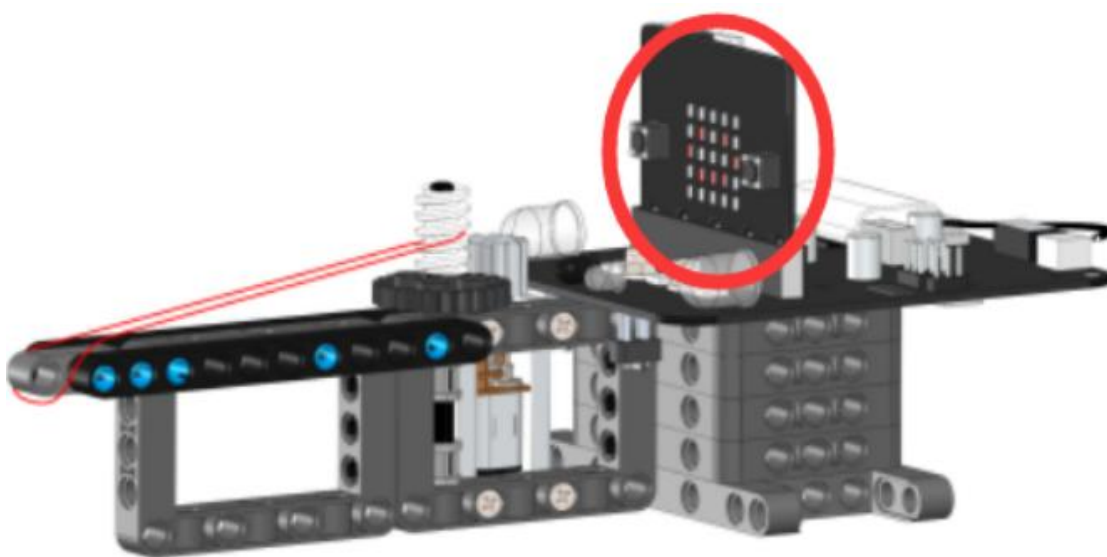
2. 入手准备

积木狙击手*1
micro:bit 手柄*1
USB 数据线*1

3. 编写程序

狙击手的程序：

接下来我们开始编写积木狙击手的程序，编写完成之后，需要将程序下载到 micro:bit 主板上。



```

1  # -*- coding: utf-8 -*-# Encoding cookie added by Mu Editor
2  from microbit import display, Image
3  import buildingbit
4  import radio
5
6  display.show(Image.HEART)
7  radio.on()
8  radio.config(group=1)

```

首先导入需要使用到的库 `display`、`Image`、`buildingbit` 和 `radio`;

`display.show(Image.HEART)`: 让 `microbit` 点阵上显示一个爱心图标;

`radio.on()`: 打开无线功能, 因为无线功能比较耗电和占内存, 所以默认关闭的, 也可以使用 `radio.off()` 关闭无线功能;

`radio.config(group=1)`: 配置无线组=1, 这样做其他设置了无线组=1 的 `microbit` 设备就可以互相通讯了, 默认是 0, 可以选择的组为 0~255, 设置的组值需要与手柄设置的一致, 否则无法正常通讯;

```

11 while True:
12     incoming = radio.receive()
13     if incoming == 'up':
14         buildingbit.car_back(100, 100, 0)
15     elif incoming == 'down':
16         buildingbit.car_run(100, 100, 0)
17     elif incoming == 'stop':
18         buildingbit.car_stop()

```

控制狙击手发射和回转功能:

`incoming = radio.receive()`: 接收无线传输过来的数据, 并且保存到 `incoming` 变量中; 如果 `incoming` 是 'up' 则让狙击手发射, 'down' 则让狙击手回转, 'stop' 则让狙击手停止;

```

20     if incoming == 'R':
21         buildingbit.car_HeadRGB(255, 0, 0)
22     elif incoming == 'G':
23         buildingbit.car_HeadRGB(0, 255, 0)
24     elif incoming == 'B':
25         buildingbit.car_HeadRGB(0, 0, 255)
26     elif incoming == 'Y':
27         buildingbit.car_HeadRGB(255, 255, 0)
28     elif incoming == 'turn_off':
29         buildingbit.car_HeadRGB(0, 0, 0)

```

如果 `incoming` 是 'R' 则车头大灯亮红灯, 'G' 则让车头大灯亮绿灯, 'B' 让车头大灯亮蓝灯, 'Y' 则让车头大灯亮黄灯。

incoming 的值需要与手柄发送的值对应，只有相同的值才可以接收并执行命令。

手柄摇杆控制的程序：

接下来我们开始编写手柄摇杆控制程序，编写完成之后需要将程序下载到手柄的 micro:bit 主板上。

```

1  # -*- coding: utf-8 -*-# Encoding cookie added by Mu Editor
2  from microbit import display, sleep, Image
3  import ghandle
4  import radio
5
6  display.show(0)
7  radio.on()
8  radio.config(group=1)

```

首先导入需要使用到的库 display、sleep、Image、ghandle 和 radio；

radio.on():打开无线功能；

radio.config(group=1): 设置无线组=1，与小车的组保持一致；

```

10 while True:
11
12     if ghandle.rocker(ghandle.up):
13         radio.send('up')
14         display.show(Image.ARROW_N)
15     elif ghandle.rocker(ghandle.down):
16         radio.send('down')
17         display.show(Image.ARROW_S)
18     elif ghandle.rocker(ghandle.left):
19         radio.send('left')
20         display.show(Image.ARROW_W)
21     elif ghandle.rocker(ghandle.right):
22         radio.send('right')
23         display.show(Image.ARROW_E)
24     elif ghandle.rocker(ghandle.pressed):
25         radio.send('turn_off')
26         display.show(Image.NO)
27     else:
28         radio.send('stop')
29         display.clear()

```

如果检测 ghandle.rocker(ghandle.up)为 True，则表示手柄的摇杆往上推，让无线发送 ‘up’ 命令，并且显示一个向上的图标；

如果检测 ghandle.rocker(ghandle.down)为 True，则表示手柄的摇杆往下推，

让无线发送 ‘down’ 命令，并且显示一个向下的图标；

如果检测 `ghandle.rocker(ghandle.left)` 为 `True`，则表示手柄的摇杆往左推，让无线发送 ‘left’ 命令，并且显示一个向左的图标；

如果检测 `ghandle.rocker(ghandle.right)` 为 `True`，则表示手柄的摇杆往右推，让无线发送 ‘right’ 命令，并且显示一个向右的图标；

如果检测 `ghandle.rocker(ghandle.pressed)` 为 `True`，则表示手柄的摇杆被按下，让无线发送 ‘pressed’ 命令，并且显示 ‘X’ 图标；

遥控没有操作发送 ‘stop’ 并清除显示；

```
31     if ghandle.B3_is_pressed():
32         radio.send('B')
33         display.show("B")
34     if ghandle.B4_is_pressed():
35         radio.send('Y')
36         display.show("Y")
37     if ghandle.B1_is_pressed():
38         radio.send('R')
39         display.show("R")
40     if ghandle.B2_is_pressed():
41         radio.send('G')
42         display.show("G")
```

接下来检测按键，对应 B1（红色），B2（绿色），B3（蓝色），B4（黄色）发送 ‘R’，‘G’，‘B’，‘Y’ 命令。



下载完成之后，看一下具体的效果和我们实验目的是不是一样的呢？

狙击手程序：

请用 MU 软件打开本课程的手柄控制_狙击手程序.py 文件查看。

手柄 Ghandle 程序：

请用 MU 软件打开本课程的手柄程序.py 文件查看。

注意：

- 1.下载程序到狙击手的 **micro:bit** 主板上，需要用 **MU** 软件打开该课程目录下面的手柄控制_狙击手程序.py 文件，然后点击 **Flash** 下载。
- 2.下载程序到遥控手柄的 **microbit** 主板上，需要用 **MU** 软件打开该课程目录下面的 **ghandle** 手柄程序.py 文件，然后点击 **Flash** 下载。
- 3.不管是下载小车还是手柄的程序，都需要先导入 **buildingbit** 的库文件到 **microbit** 里，如果不知道如何导入，请查看课前准备的导入 **buildingbit** 库的课程。
- 4.程序下载完成之后，给手柄和小车正常上电，它们会自动配对，在信号传输的范围内就可以操控了。