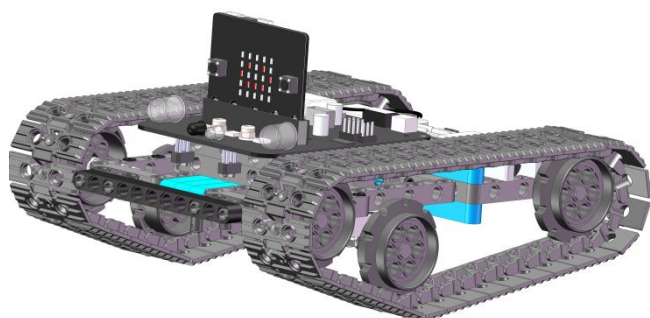
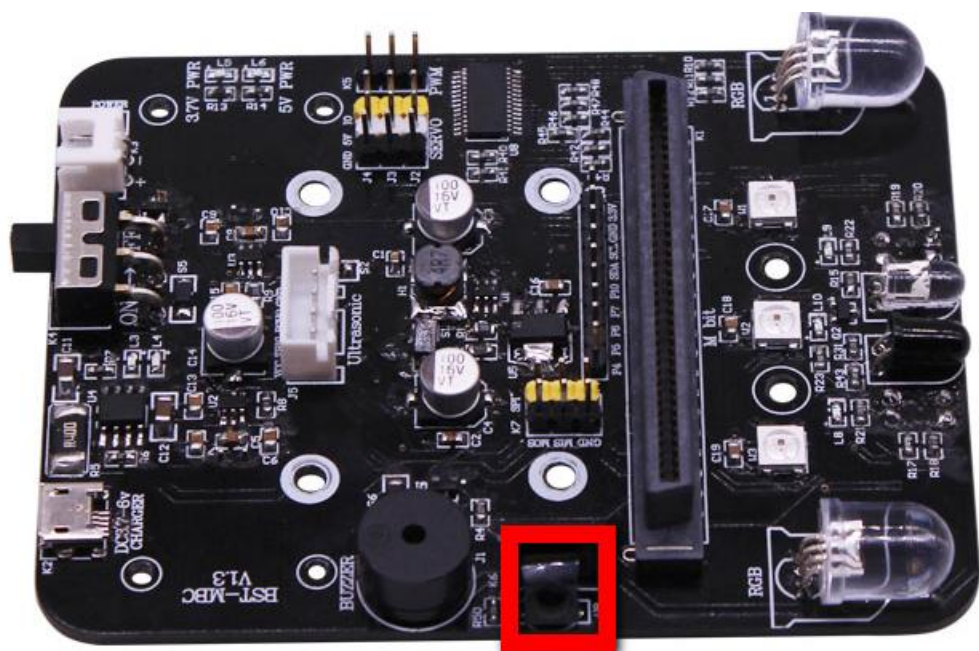


# 坦克车红外遥控



## 注意事项:

1. 在进行红外遥控的时候遥控器要对着扩展板上面的红外接收头。
2. 红外遥控器底部有一个塑料片需要取下来，才能正常使用。
3. 红外遥控器和红外接收头发出的红外光线人眼是无法看到的，可以在没有过滤红外光的摄像头下面看到。



红外接收头

## 1. 实验目的:

下载好程序后，打开坦克车的电源开关，按下红外遥控的按键，坦克车就会有相应的动作。红外遥控的小灯按键打开车头七彩灯亮白色，红色的电源键是关闭车头七彩灯按键；喇叭键是响一声鸣笛；前后左右按键控制小车的前后左右，左右旋按键控制小车速度的左右；+号表示控制小车速度增加，-号控制小车速度减少，数字 0 控制小车速度恢复默认；1~7 代表音乐中的 do、re、mi、fa、sol、la、

si; 8 打开车身可编程 RGB 灯, 9 关闭车身可编程 RGB 灯。

## 2. 入手准备:

积木坦克车\*1  
红外遥控器\*1  
USB 数据线\*1

## 3. 编写程序

接下来我们开始编写积木坦克车红外遥控的程序, 如下图所示:

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

首先导入需要使用到的库文件;

```
8 display.off()
9 np = neopixel.NeoPixel(pin16, 3)
10 np.clear()
11 buildingbit.init_IR(pin8)
12 buildingbit.car_run(0, 0, 0)
13 |
14 speed = 100
15 a = 0
```

display.off(): 关闭 microbitLED 点阵屏, 否则会 and 红外遥控接收器冲突而出错;

np = neopixel.NeoPixel(pin16, 3): 初始化车身可编程 RGB 灯, 第一个参数为 RGB 灯引脚, 第二个参数为 RGB 灯的数量;

np.clear(): 清除 RGB 灯显示;

buildingbit.init\_IR(pin8): 初始化红外接收器, 参数为红外接收器的引脚;

buildingbit.car\_run(0, 0, 0): 让小车停止;

speed: 小车速度;

a: 记录数据

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

读取红外接收器接收到的数据, 并且只取高 8 位, 遥控器的按键码值如下图所示:



```

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

```

先处理两个特殊的值，默认会一直收到-0x01 值，如果长按某个键，会收到 0xff，变量 a 是为了解决长按按键时有可能收到-0x01 的问题，其他就是对应每一个按键的码值；

```

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

```

红色电源按键的码值为 0x00，buildingbit.car\_HeadRGB(0, 0, 0)是关闭车头大灯；

```

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

```

向前按钮，控制小车前进；

```

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

```

蜂鸣器按钮，控制响一声鸣笛；

其他以此类推，每一个按钮里面写对应的功能。

以上就是本次坦克车的程序。编写完成之后，大家别忘了把它下载到 **micro:bit** 主板上哦~

如果出现下载失败的问题，请确认已经下载了 **buildingbit** 的库，如果没有请参考基础课程-课前准备-导入 **buildingbit** 库。

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？

### 程序代码：

请用 MU 软件打开本课程的[坦克车红外遥控.py](#) 文件查看。