

Infrared transmission experiment

Introduction the devices

This time, we will introduce the infrared transmitter and receiver modules, which actually play an important role in our daily life. Now such a device is widely used in many home appliances, such as air conditioning, television, DVD, etc. It is based on wireless remote sensing, but also a remote control, it is necessary to study its principle and how to use.

Infrared transmitting tube and infrared receiving tube are devices that convert electric energy into near-infrared light directly. Its structure and principle are similar to ordinary light-emitting diodes, but the semiconductor material is different.

The infrared receiver is a receiving, amplifying and demodulating device. The internal integrated circuit has been demodulated and the output is digital signal.



Infrared receiver

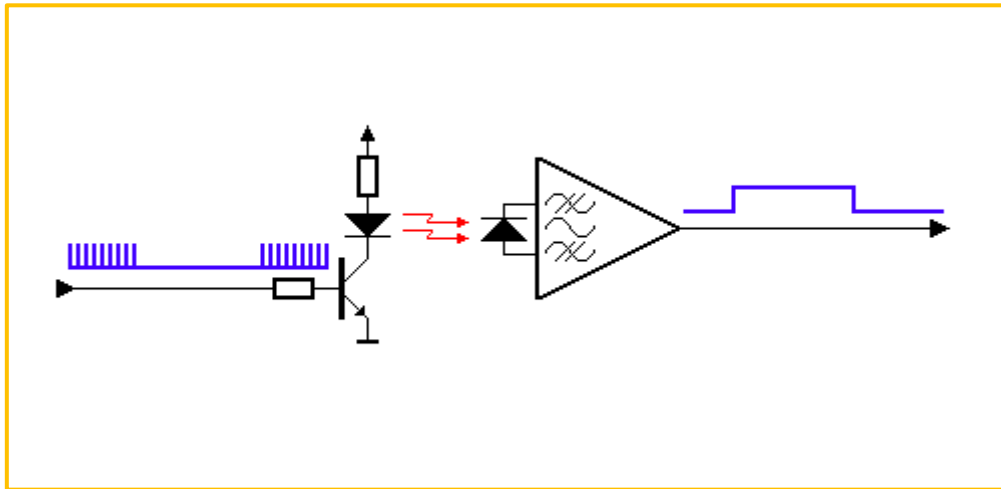


Infrared transmitter

Working Principle

To understand the structure of infrared receiver: infrared receiver is composed of IC and PD. IC is the processing element of the receiver, mainly composed of silicon crystal and circuit. It is a highly integrated device. PD is a photodiode whose main function is to receive optical signals. The infrared emitting diode sends out the modulation signal, and the infrared receiving head recovers the signal after receiving, decoding, filtering and a series of operations.

In this experiment, we wanted to use two Arduino motherboards, a transmitter (Master) and a receiver (Slave). After the circuit is connected, we can do the test.



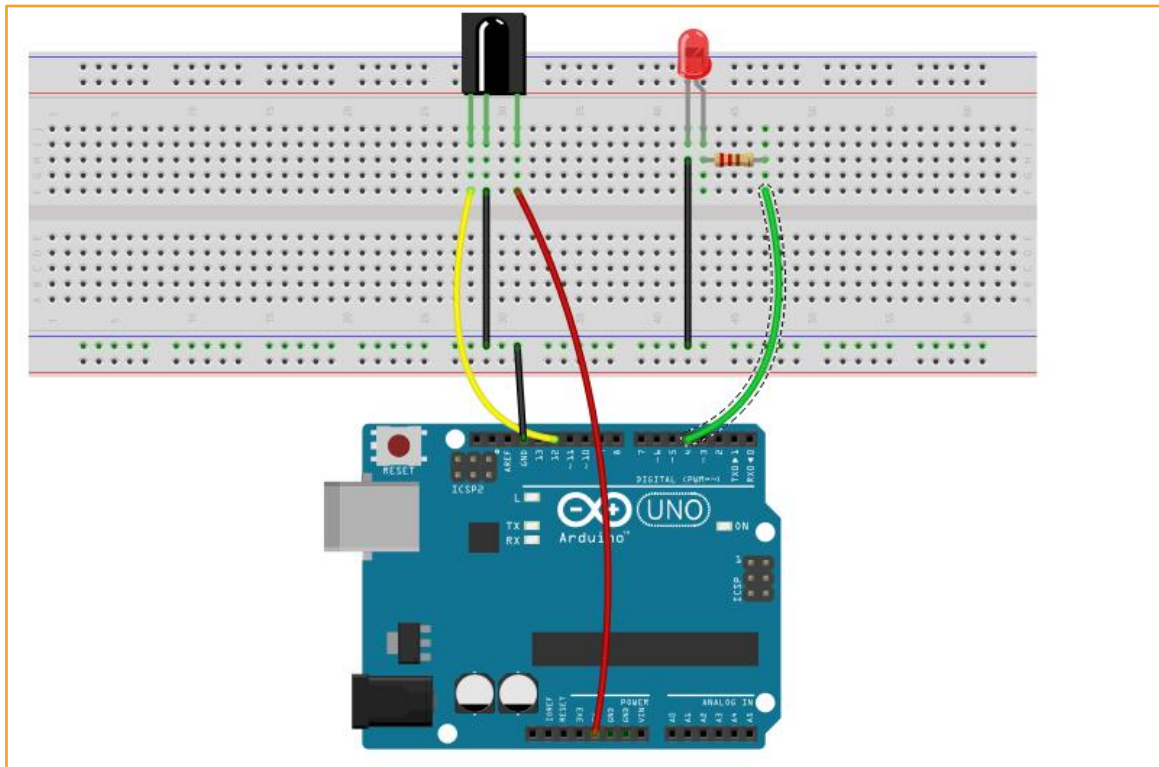
Component List

- ◆ Keywish Arduino Uno R3 mainboard*2
- ◆ USB cable*2
- ◆ Infrared transmitter module*1
- ◆ Infrared receiver module*1
- ◆ Jumper wires

Wiring of Circuit

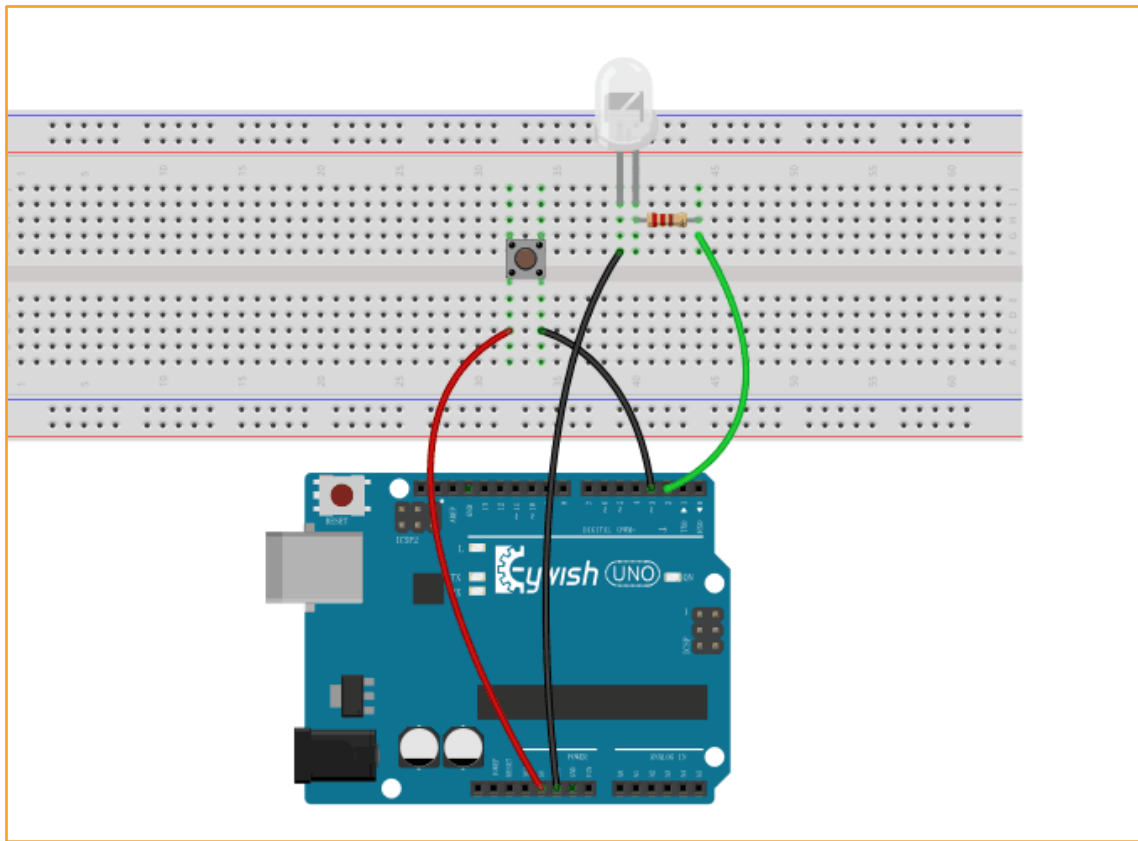
Infrared receiving connection

Arduino Uno	Infrared receiver
12	S
GND	-
+5V	+
Arduino Uno	LED
4	S
GND	GND



Infrared transmission experiment

Arduino Uno	receiver
3	+
GND	G
Arduino	button
4	Pin-1
+5V	Pin-2



Code

Transmitter:

```
#include "IRremote.h"           // 引用 IRRemote 函式库
const int buttonPin = 4;        // 按键(pushbutton)接在 pin 4
int buttonState = 0;            // 按键状态
IRsend irsend;                 // 定义 IRsend 物件来发射红外线讯号

void setup()
{
    Serial.begin(9600);
    pinMode(buttonPin, INPUT);  // 把 buttonPin 设置成 输入
}

void loop()
{
    // 读取按键状态
    buttonState = digitalRead(buttonPin);
    // 检查按键是否被按下
    // 有的话 buttonState 会是高电平输出
    if (buttonState == HIGH) {
        // 发射红外线讯号
        Serial.println("button");
        irsend.sendNEC(0x4CB3817E, 32); //
        这个编码即你按的键（接pin4）的编码，可以随意更改，如果要控制其他设备，只需将此编码改为相应编码就行
    }
    delay(200);
}
```

Receiver:

```
#include "IRremote.h"

int RECV_PIN = 12; //定义红外接收器的引脚为12
int LED_PIN = 4;   //定义发光LED引脚数字4
int a = 0;
IRrecv irrecv(RECV_PIN);

decode_results results;

void setup()
{
    Serial.begin(9600);
    irrecv.enableIRIn(); // 初始化红外接收器
    pinMode(LED_PIN, OUTPUT); //设置发光LED引脚数字4
}

void loop() {
    if (irrecv.decode(&results)) {
        Serial.println(results.value, HEX);

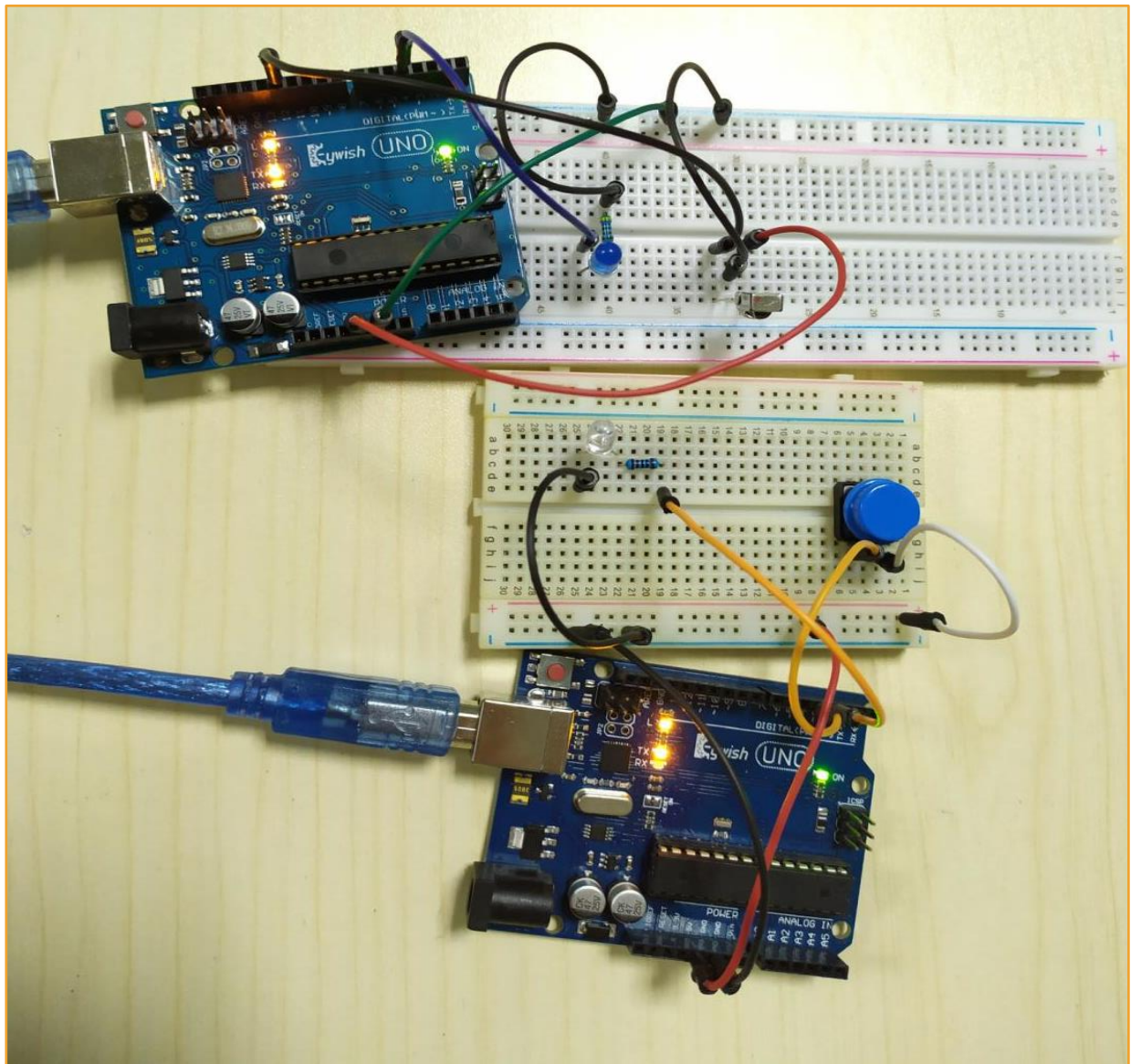
        if(results.value == 0x4CB3817E & a == 0) //此处的32位值与发射部分发射的值要保持一致
        {
            digitalWrite(LED_PIN, HIGH); //LED点亮
            a=1;
        }
        else if(results.value == 0x4CB3817E & a == 1)
        {
            digitalWrite(LED_PIN, LOW); //LED熄灭
            a=0;
        }

        irrecv.resume(); // 接收下一个值
    }

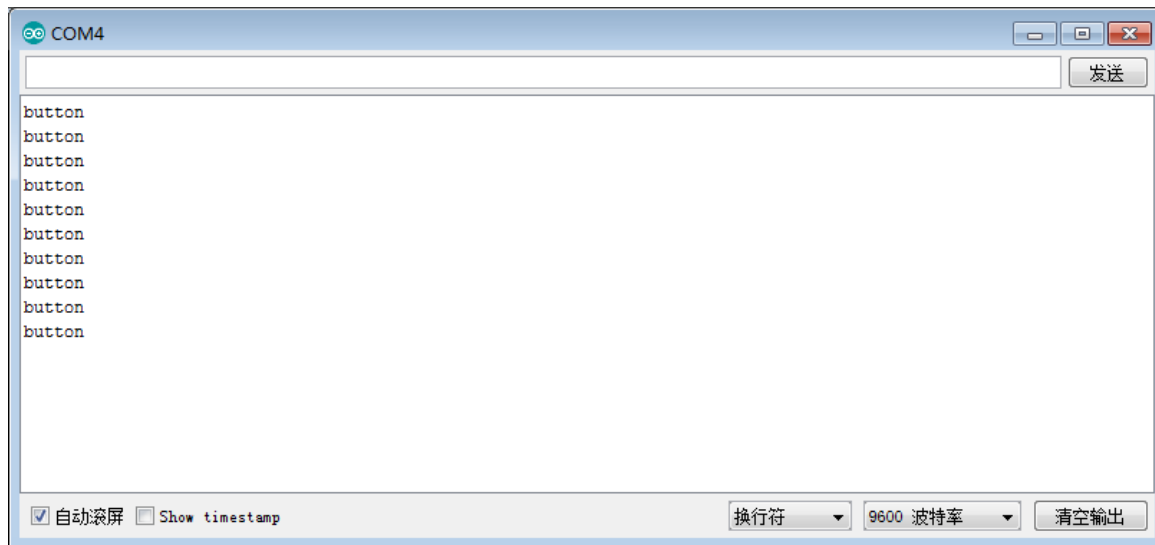
    delay(120);
}
```

Experiment Result

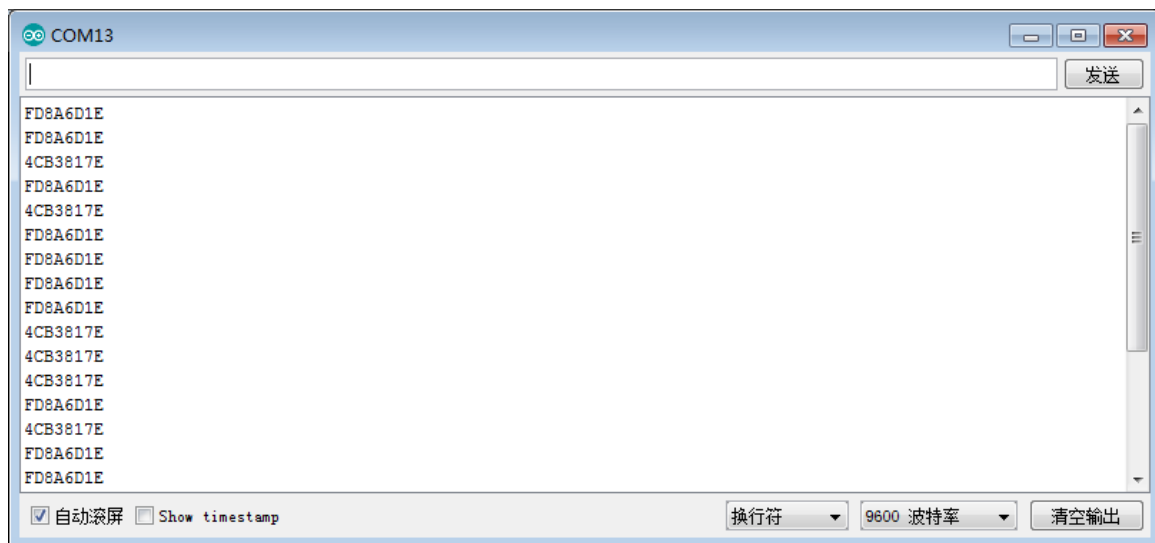
This time we use two Arduino mainboards, when downloading the codes, you have to know which is the transmitter code and which is receiver code. If the download is wrong, there will be no result. After downloading the codes, we open the Serial Monitor window, if you can see the following displayed data, it shows that you succeed.



Transmitter Exeripment Result

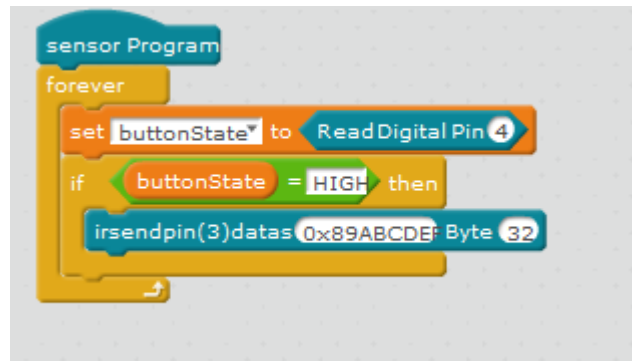


Receiver Exeripment Result



MBlock graphical programming program

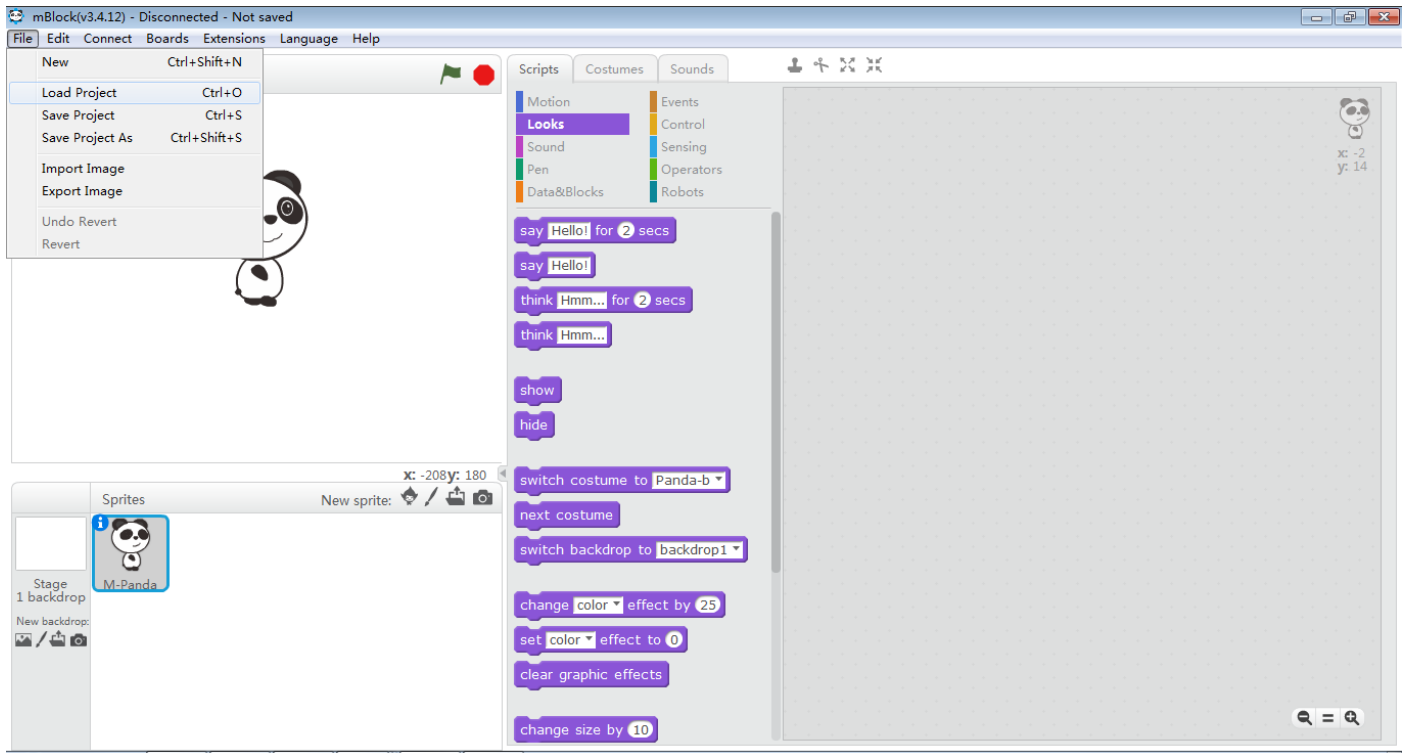
Transmitter:



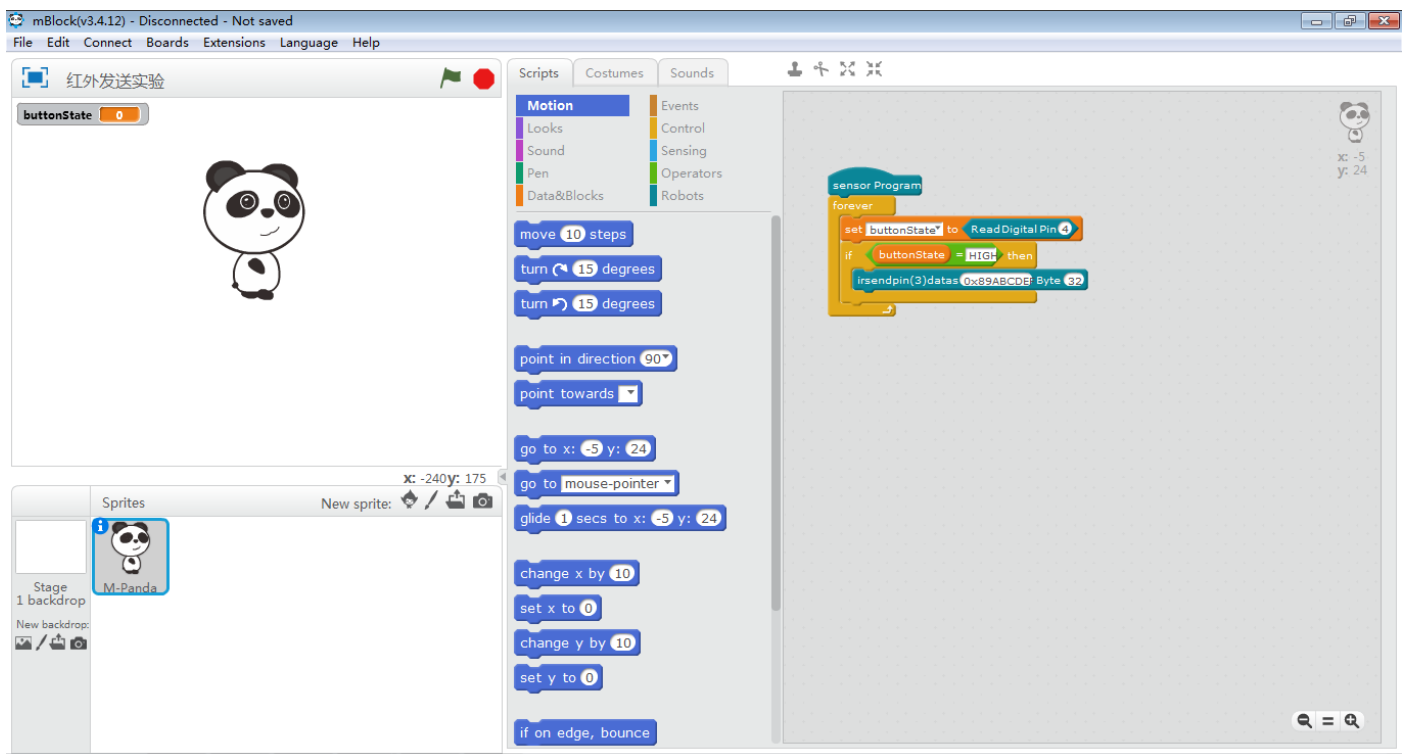
Receiver:



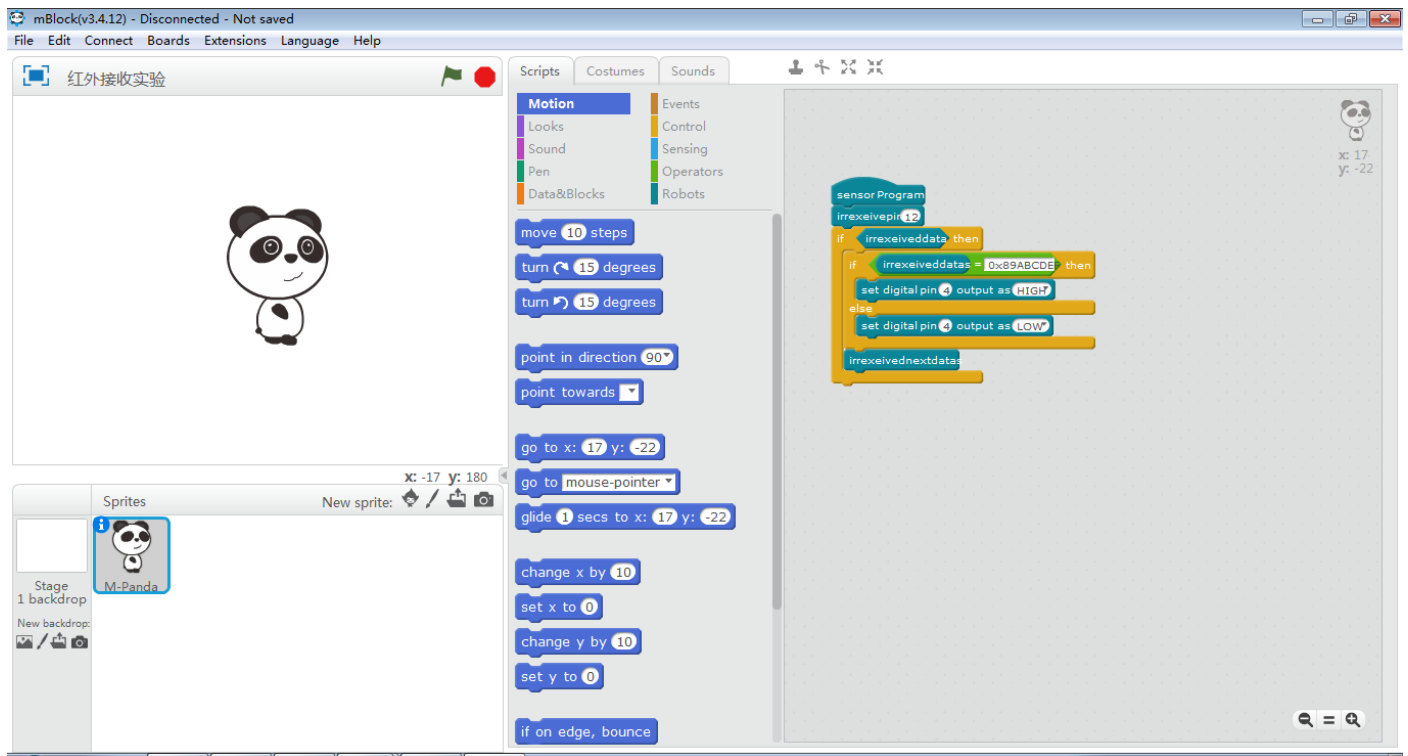
You can also use mblosk to open a finished program file, which is a . sb2 file. Here 's how to open it:



transmitter

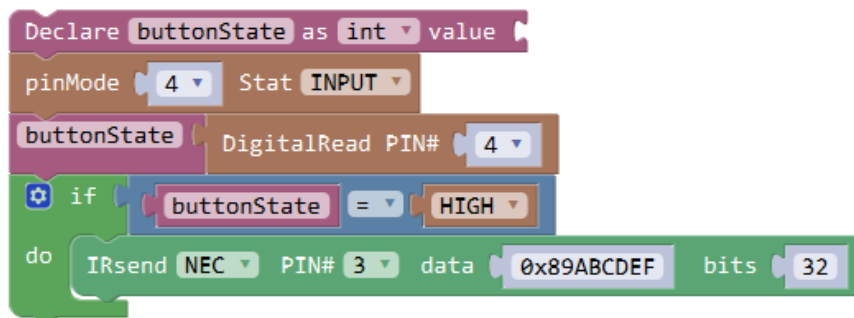


receiver

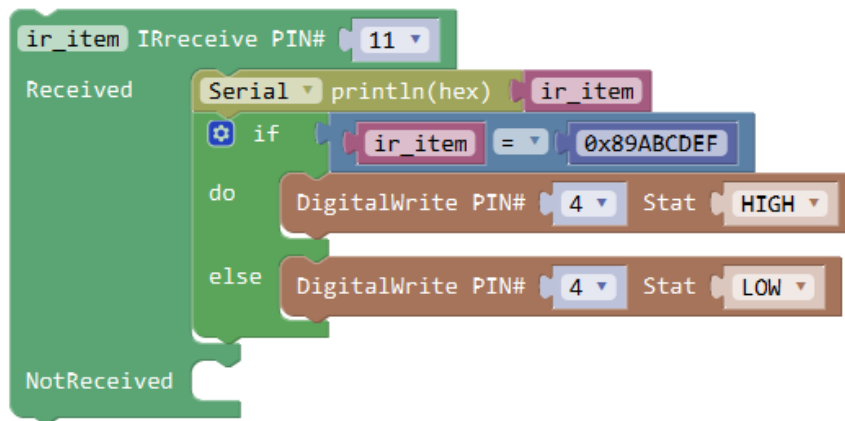


Mixly graphical programming program

Transmitter:



Receiver:



MagicBlock graphical programming program

