# Cell phone bluetooth dimming experiment

Introduction of JDY-16 bluetooth module

Used in this experiment is JDY - 16 bluetooth module, details see "JDY - 16 bluetooth module test data/JDY - 16 bluetooth 4.2 module (JDY - 16 - V1.9)", in order to guarantee confirmation before using bluetooth communication module is normal, in "JDY - 16 bluetooth module test data" folder, use the JDY - 16 bluetooth module test APP and JDY - 16 bluetooth module test procedure, test the bluetooth communication module.The jdy-16 bluetooth module is a serial port bluetooth, which must be connected through the corresponding APP when using. It cannot be directly connected through the bluetooth that comes with the phone.
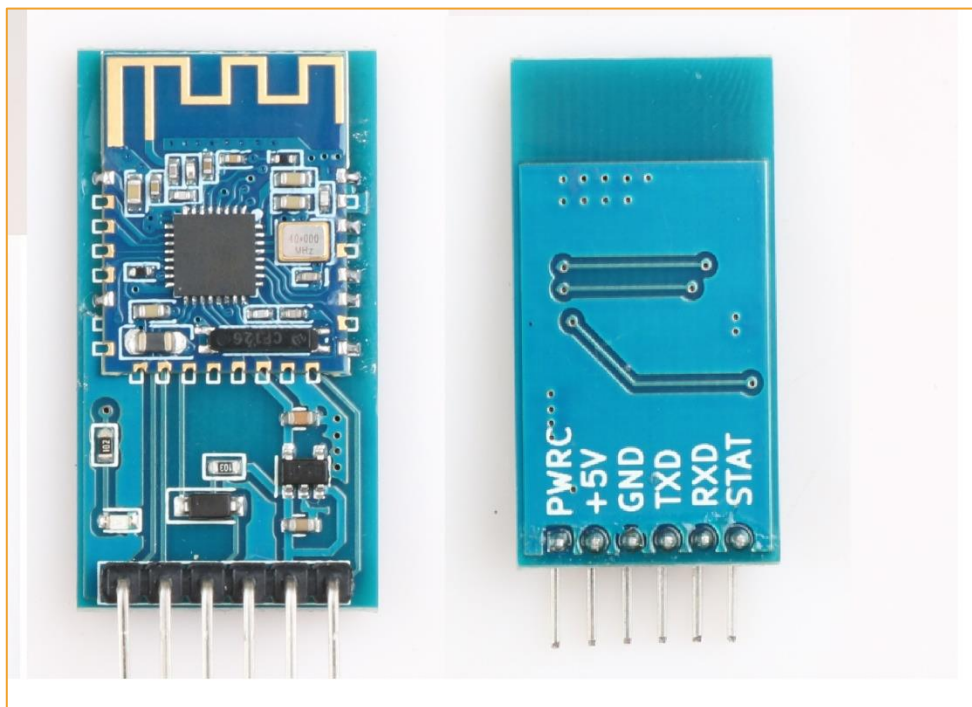
JDY-16 bluetooth module

Experimental purpose

● Connect the bluetooth module through the mobile phone bluetooth APP and adjust the color of RGB LED lamp.

The component list

◆ Arduinos Uno motherboard
◆ Breadboard
◆ USB cable
◆ RGB LED light module
◆ Bluetooth JDY16 bluetooth module

◆ Jumper wires

Wiring
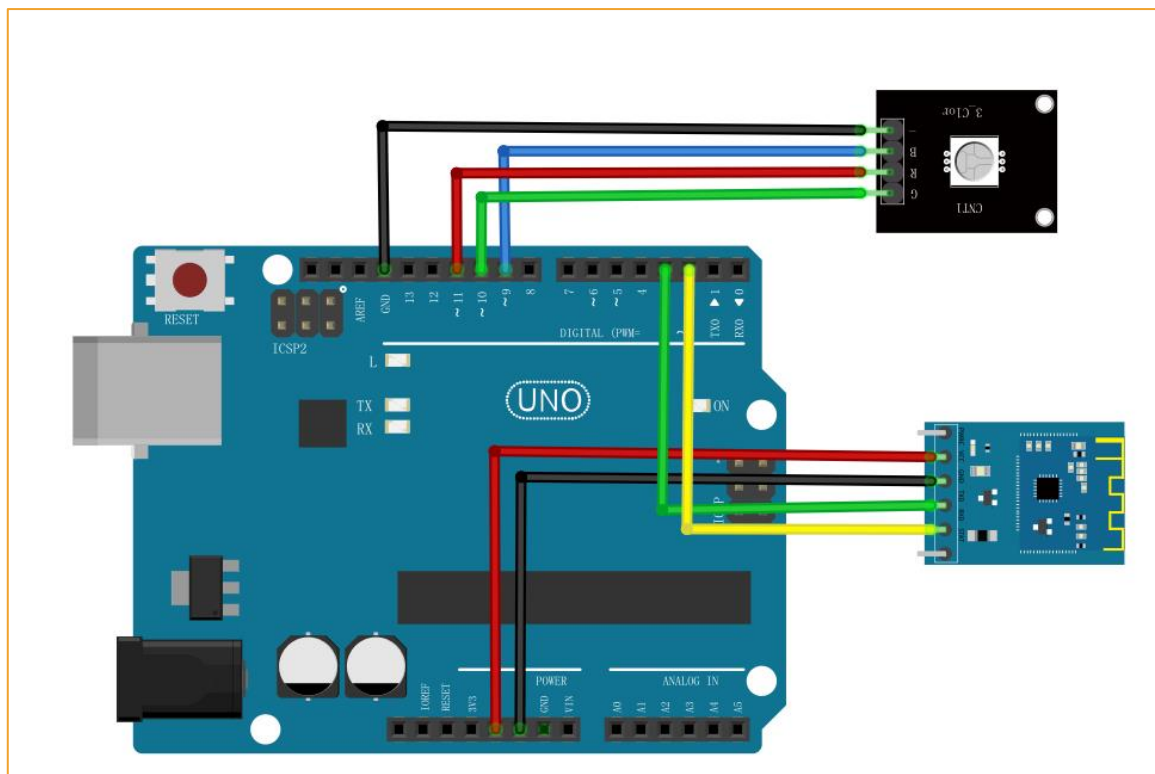
Jdy-16 bluetooth module connection

Bluetooth JDY16 bluetooth module	Arduino
GND	GND
VCC	5V
TXD	3
RXD	2

Common cathode RGB LED lamp connection

| Common anode RGB LED lamp	Arduino	break-over voltage
GND	GND	
Red	10	1.8~2.6V
Green	11	2.8~3.6V
Blue	9	2.8~3.6V

The experimental principle

The Keywish Arduino main control board is connected to the bluetooth module, and serial communication is used. After power is turned on, the bluetooth APP KeywishBot is opened. After the APP is connected to the bluetooth module, the color of RGB is controlled through the APP.
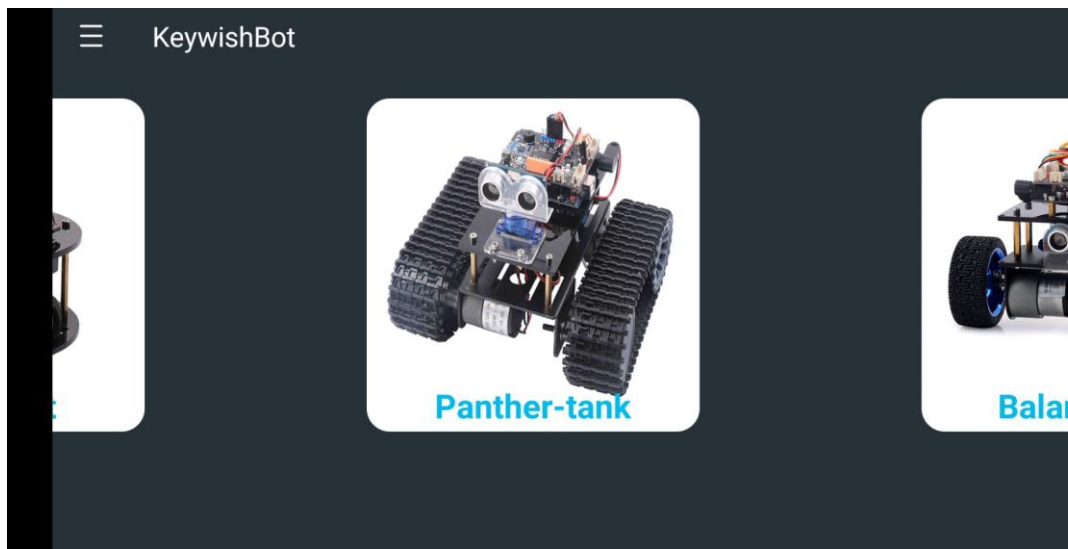
Experimental steps

- Connect the Arduino master controller to the computer through USB
- Open the ArduinoIDE programming example program \ BLE_CommonCathode_RGB \ BLE_CommonCathode_RGB Ino file
- Burnning the BLE_CommonAnode_RGB program
- Power the Arduino master board
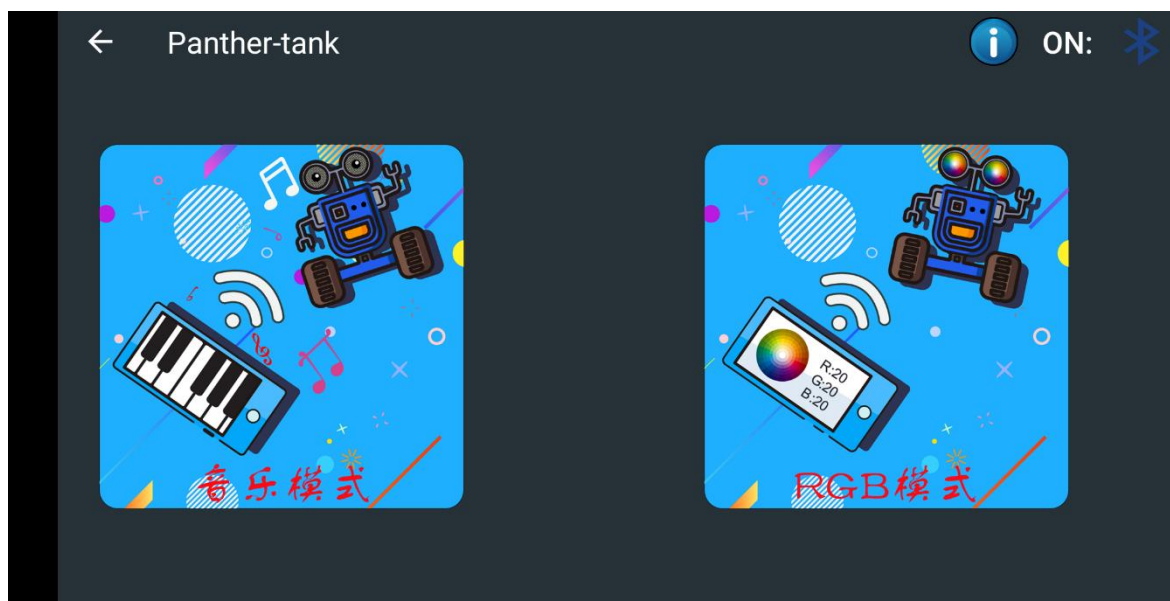- Open the KeywishRot APP to connect to the bluetooth module
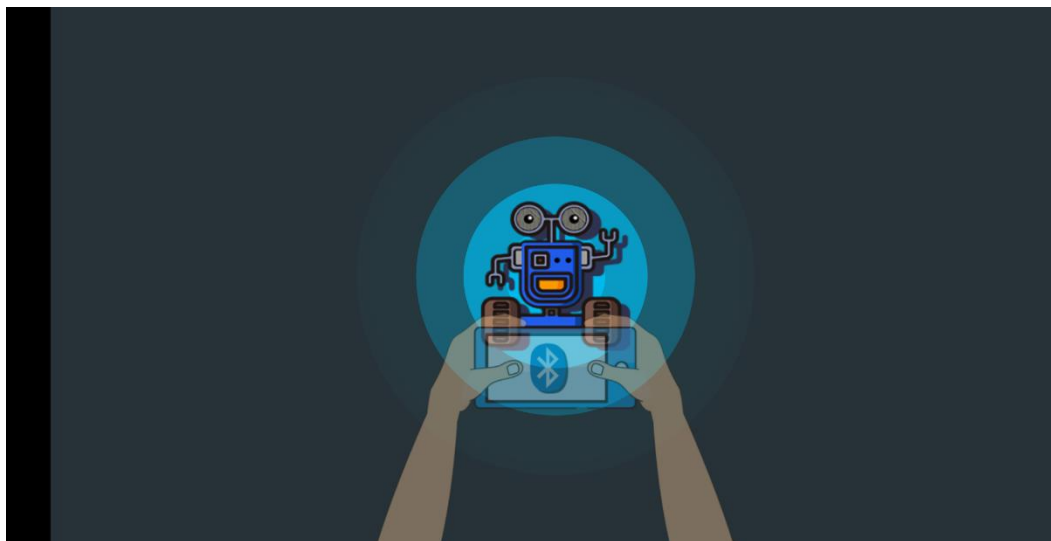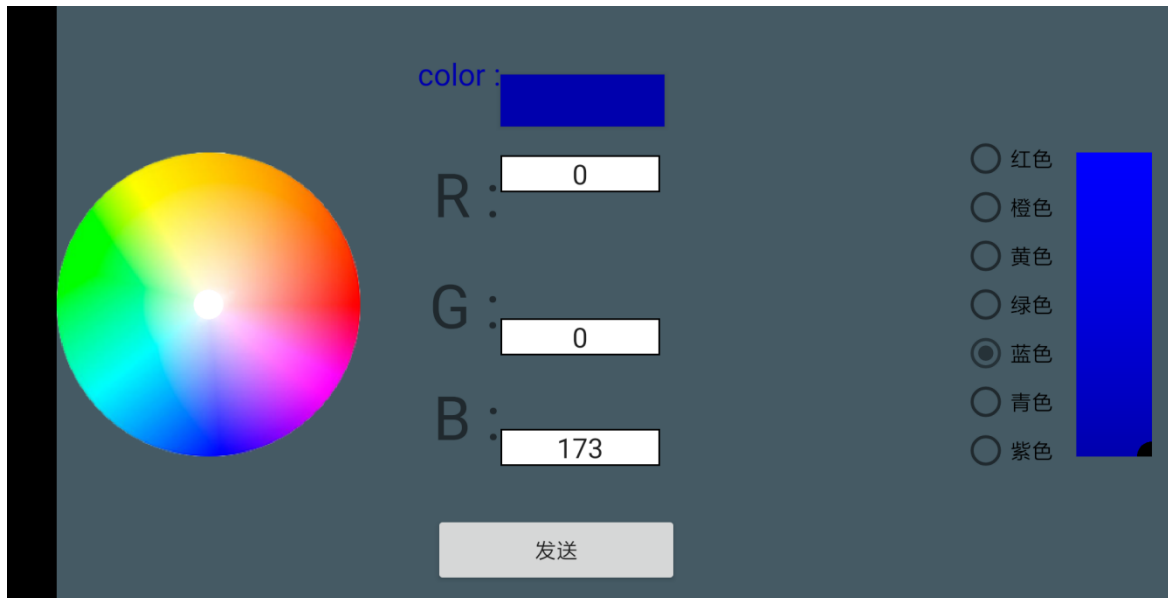

6) select Panther - tank
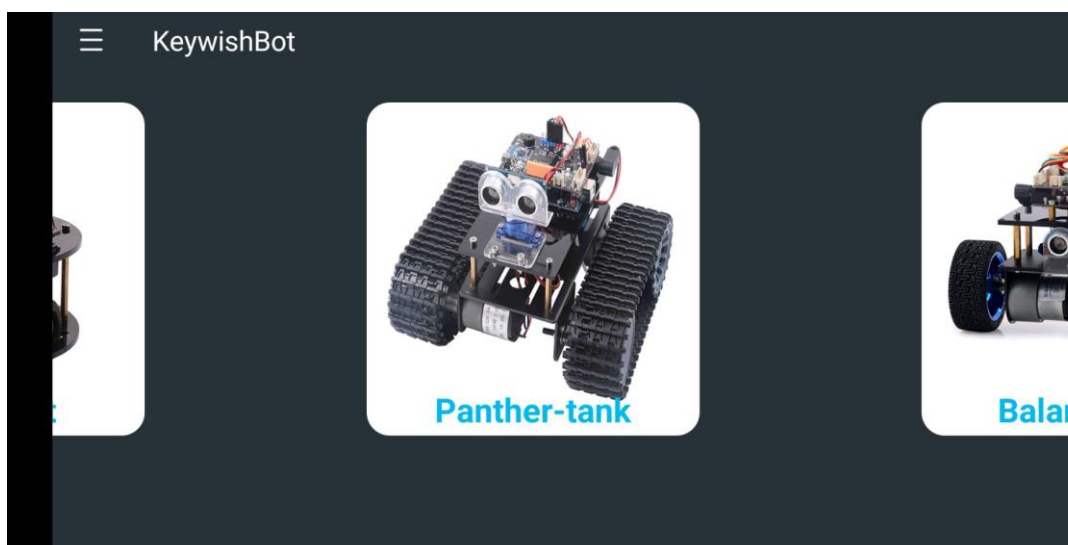
7) select RGB mode

8) adjust the color of RGB lamp on the APP

Arduino IDE Code

```
#include "ProtocolParser.h"
#include <SoftwareSerial.h>

#define Software_TX 2
#define Software_RX 3
SoftwareSerial BLE_JDY_16(Software_RX, Software_TX);
#define RGB_RED    10
#define RGB_GREEN  11
#define RGB_BLUE   9

long color;
ProtocolParser *mProtocol = new ProtocolParser();

void setup() {
    Serial.begin(9600);
    pinMode(RGB_RED,OUTPUT);
    pinMode(RGB_GREEN,OUTPUT);
    pinMode(RGB_BLUE,OUTPUT);
    delay(100);
}

void setColor(int red,int green,int blue)
{
    analogWrite(RGB_RED,red);
    analogWrite(RGB_GREEN,green);
    analogWrite(RGB_BLUE,blue);
```

```
}

void loop() {
    static bool recv_flag;
    mProtocol->RecevData();
    recv_flag = mProtocol->ParserPackage();
    if (recv_flag) {
      switch (mProtocol->GetRobotControlFun()) {
        case E_LED:
          color = mProtocol->GetRgbValue();
          setColor( color>>16, (color>>8)&0xFF, color&0xFF);
          break;
      }
    }
    return;
}
```

Mlock graphical programming program

MBlock writes the bluetooth dimming program as shown in the figure below:

```
define setColor red green blue
  Set pwm pin 10 output as red
  Set pwm pin 11 output as green
  Set pwm pin 9 output as blue


sensor Program
softSerialinit RX 3 TX 2
Set Baud Rate 9600 ▼
forever
    Receive Bluetoth Data
    if  Received Bluetooth Data  then
        if  Bluetooth Mode E_LED ▼  then
            Bluetooth Set RGB
```

Mixly graphical programming program

```
Declare color as long value

setColor with: red, green, blue
do  AnalogWrite pin 10  Assignment is  red
    AnalogWrite pin 11  Assignment is  green
    AnalogWrite pin 9   Assignment is  blue

setup
    setup SoftwareSerial RX# 3  TX# 2
    Serial baud rate 9600
    SoftwareSerial baud rate 9600

Receiving Bluetooth Data
    if  Received Bluetooth Data
    do  if  Bluetooth mode Dimming mode
        do  color  Bluetooth Gets RGB Value
            Bluetooth dimming
```