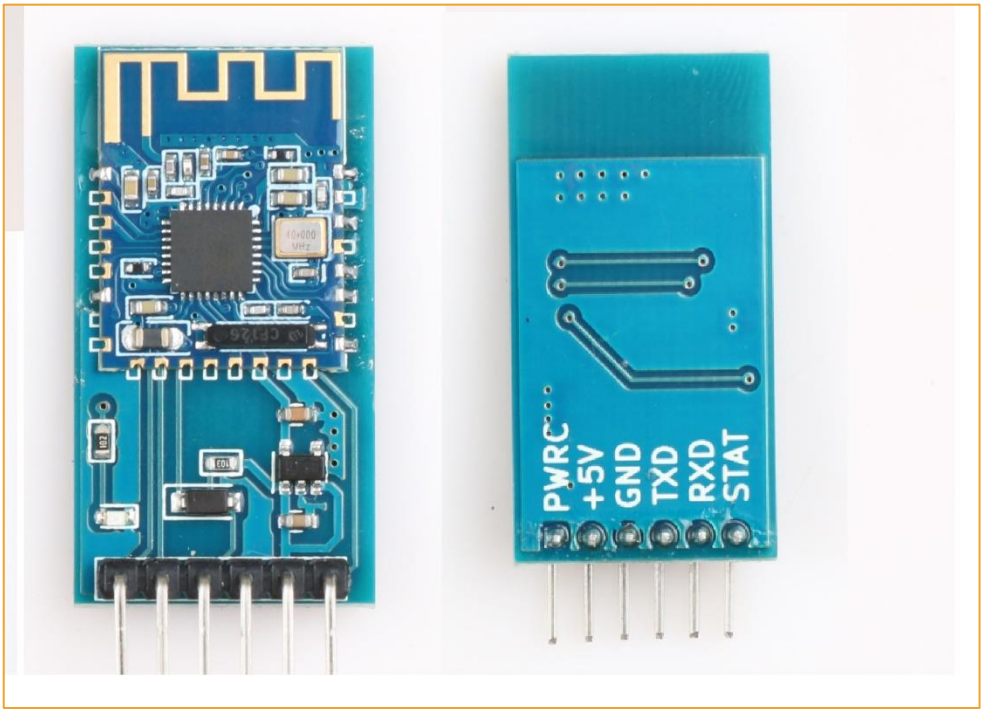
Mobile phone bluetooth piano experiment

Introduction to 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.



Introduction to buzzer

Some appliances often buzz when in an electric state , this is actually from a buzzer, and the annoying bell at school is but a larger buzzer. There are two kinds of buzzers. One is active buzzer, the other is passive buzzer. “active” and “passive” don’t mean the common power source, but a buzzer with or without internal oscillators. Active buzzer will buzz as long as you electricity it, but the frequency is fixed. Passive buzzer, buzzer without internal oscillators, will not buzz when electrified internal oscillators, it requires 2~5 kHz square wave to actuate, then wave forms in different frequency can buzz with corresponding sound.

Active buzzer Passive buzzer

The working principle of buzzer

The passive buzzer generates music mainly through the output of pulse signals of different levels at the I/O port of the single-chip microcomputer to control the buzzer's pronunciation. To generate audio pulse signals, it is necessary to calculate the period of certain audio (1/ frequency), and then divide this period by 2, that is, the time of half a period.SCM timer is used to time the time of this half cycle, every time the timing to the output pulse I/O port phase reversal, so in this I/O port to get the pulse frequency.

For example, if the Arduino USES 12MHzs crystalline oscillator to produce the sound with middle tone Re, the audio pulse frequency of 587Hzs should be output. The pulse period of the audio signal T=1/587=1703.5775us, and the half-cycle time is 852us. Therefore, the timer must be set to count =852us/1us=852.

In addition, the principle of passive buzzer sound is the current through the electromagnetic coil, the electromagnetic coil to generate a magnetic field to drive the vibration of membrane sound.So it takes a certain amount of current to drive it.

Experimental purpose

- Through the bluetooth APP of mobile phone, the buzzer is controlled to emit the corresponding sound of piano keys

Experimental principle

Connect the Keywish Arduino main control board with the bluetooth module, and use serial port to communicate. After power on, open the bluetooth APP KeywishBot. After the APP connects the bluetooth module, control the color of RGB through the APP.

The component list

- Arduinos Uno motherboard
- Bread plate

- USB cable
- Passive buzzer*1
- Bluetooth JDY16 bluetooth module*1
- jumpers

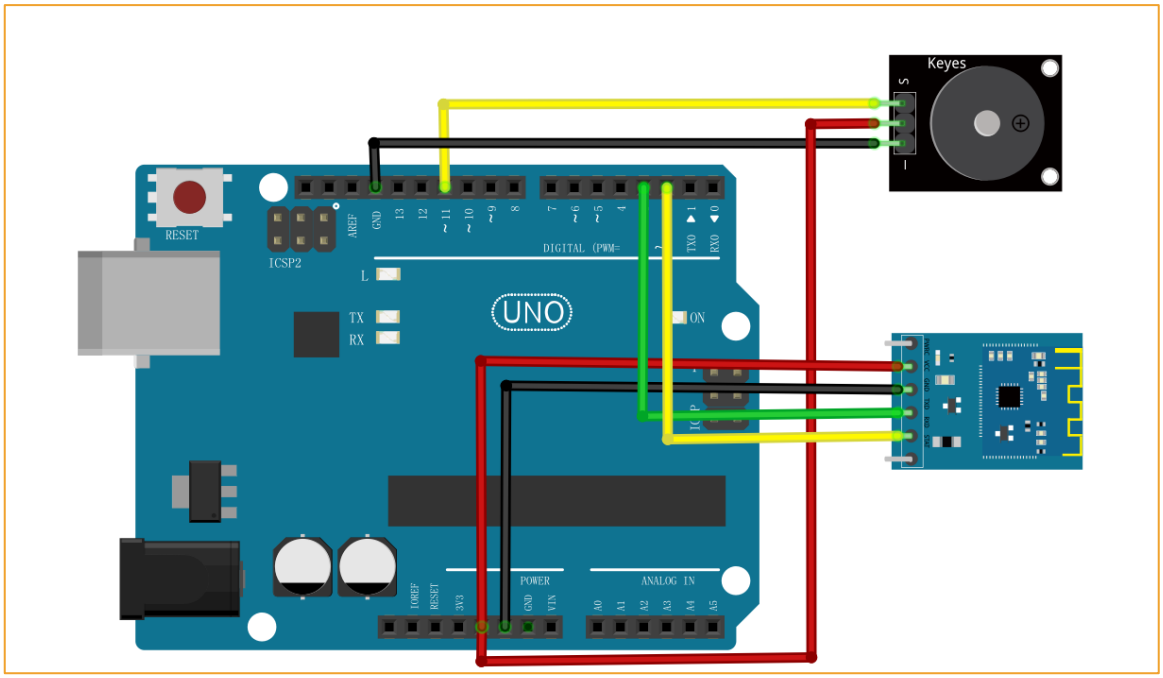
Wiring

Jdy-16 bluetooth module connection

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

The passive buzzer module signal control pin is connected to 11 pin, the positive pole is connected to 5V, and the negative pole is grounded.

Buzzer	Arduino
-	GND
VCC	5V
S	11

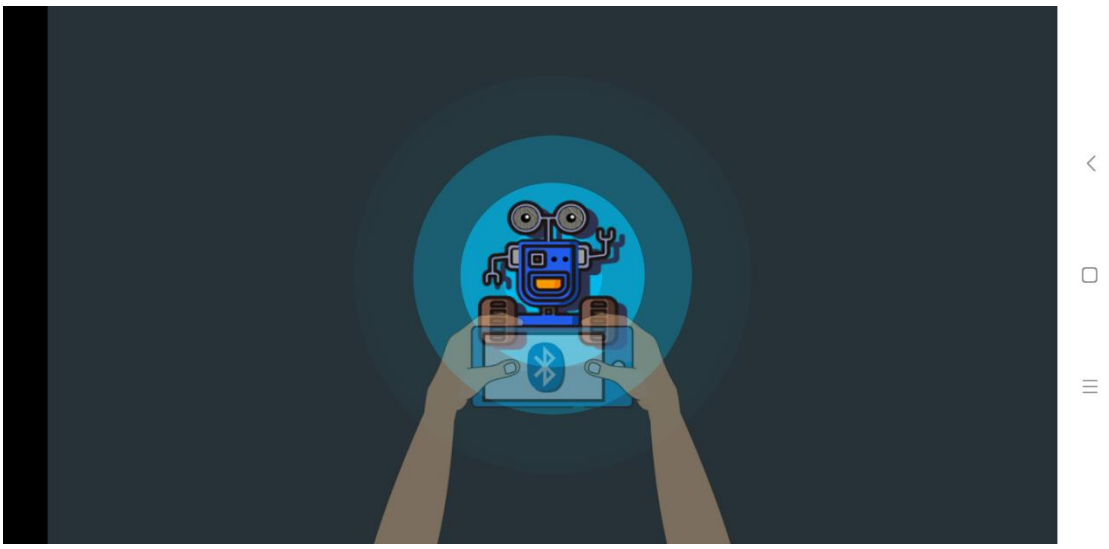


Experiment principle

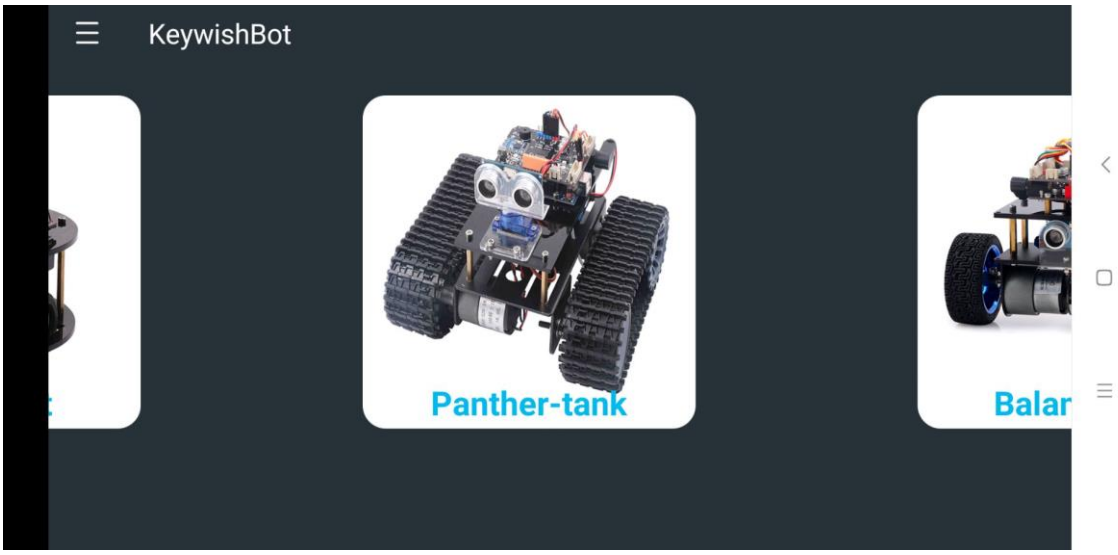
Connect the Keywish Arduino main control board with the bluetooth module and use serial port communication. After power on, open the bluetooth APP KeywishBot. After the APP connects the bluetooth module, control the buzzer through the APP.

Experimental steps

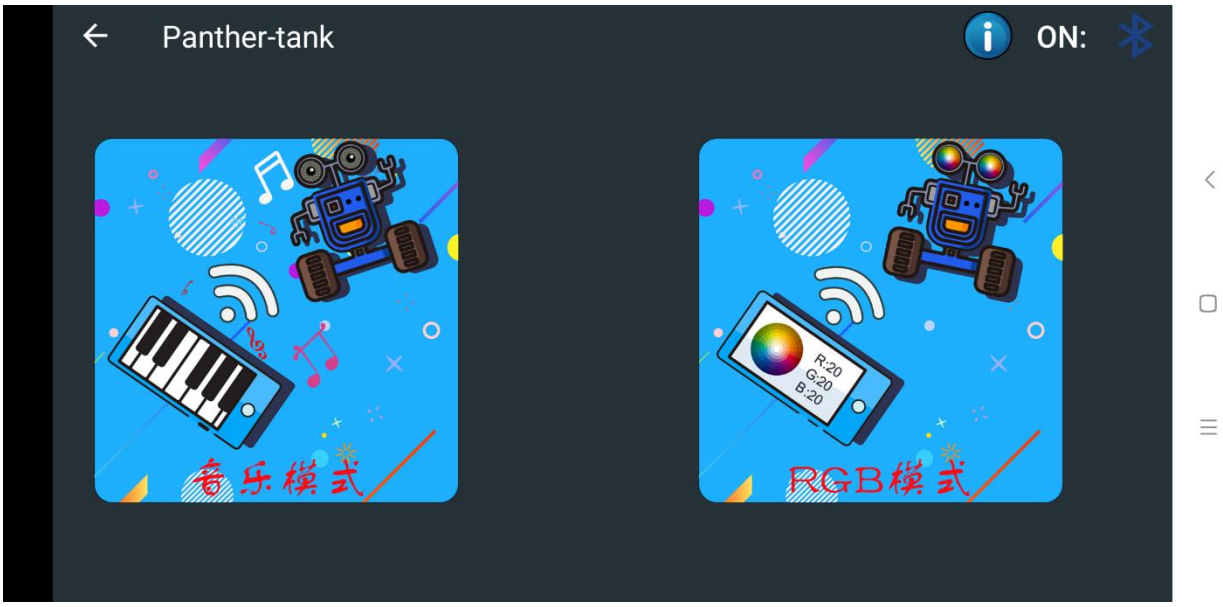
- Connect the Arduino master controller to the computer through USB
- Open mobile phone bluetooth piano experiment \ArduinoIDE programming sample program \BLE_Buzzer\ ble_buzzer.ino program
- Burn the ble_buzzer.ino program
- Power the Arduino master board
- Open the KeywishRot APP to connect to the bluetooth module



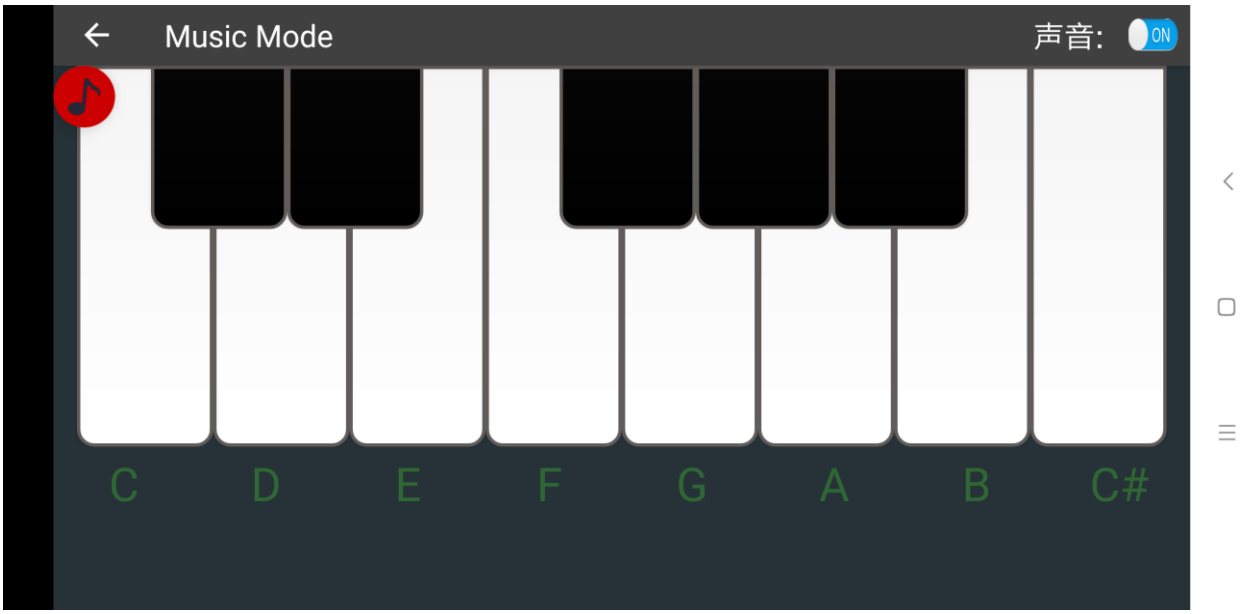
6) select Panther - tank



7) select music mode



8) control the sound of the buzzer on the APP



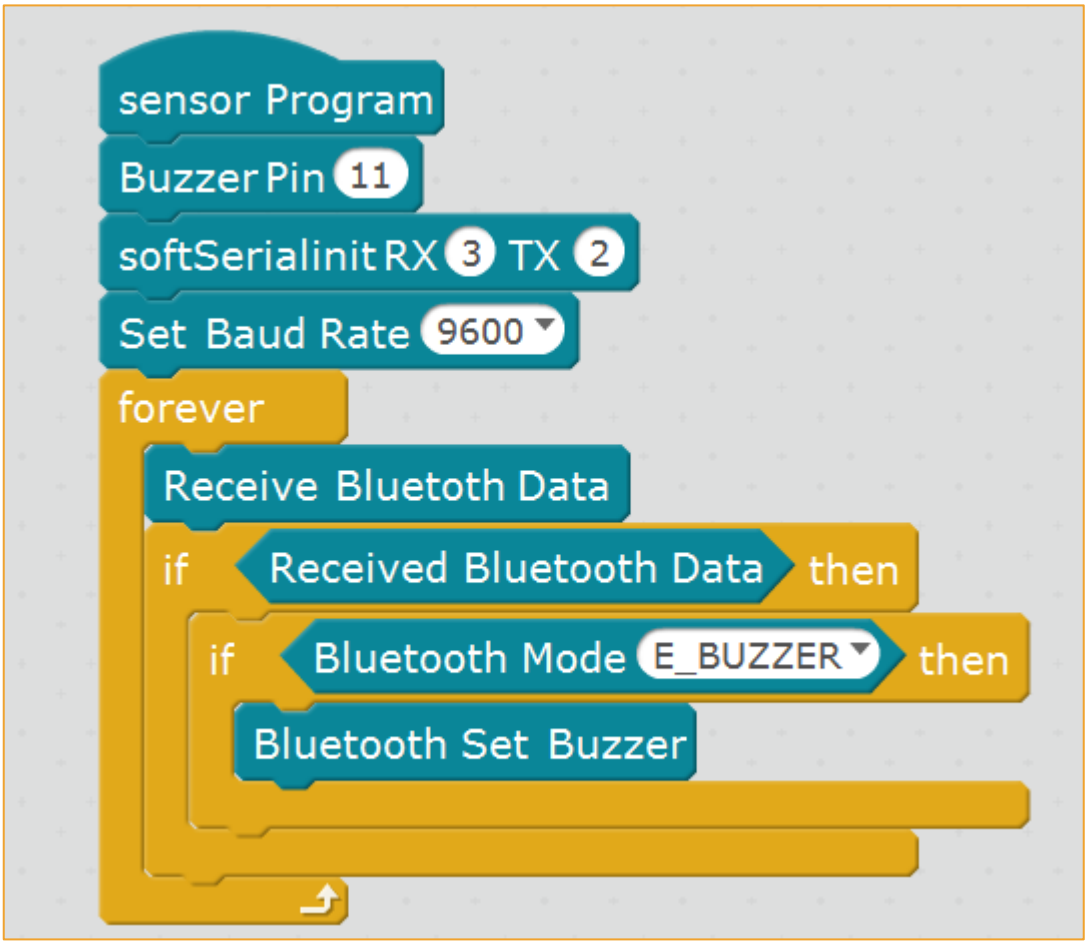
Code

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

#define Software_TX 2
#define Software_RX 3

#define BUZZER_PIN    11

ProtocolParser *mProtocol = new ProtocolParser();
Buzzer *mBuzzer = new Buzzer(BUZZER_PIN);
void setup() {
    Serial.begin(9600);
    pinMode(BUZZER_PIN, OUTPUT);
    delay(100);
}

void PianoSing(byte b[])
{
    union result
    {
      float d;
      unsigned char data[4];
    }r1,r2;
    r2.data[0]=b[0];
    r2.data[1]=b[1];
```

```
    r2.data[2]=b[2];
    r2.data[3]=b[3];
    //mBuzzer->noTone(9);
    mBuzzer->_tone(r2.d,120, 2);
}
void loop() {
    static bool recv_flag;
    mProtocol->RecevData();
    recv_flag = mProtocol->ParserPackage();
    if (recv_flag) {
        if( mProtocol->GetRobotControlFun() == E_BUZZER) {
          PianoSing((byte *)mProtocol->GetPianoSing());
        }
     }
     return;
}
```

MBlock programming program

MBlock writes the program as shown in the figure below:



Mixly programming program