Appendix A

Source code

This is the code for the first uno board (master):

- 1. // Using the SV5W module, this uno acts as a host and only sends data to the slave and does not receive information from the slave.
- 2. // And through the creation of soft serial port and SV5W module serial communication.
- 3. #include<SoftwareSerial.h>//Use a soft serial port
- 4. //Reason: uno only 0 (RX) and 1 (TX) a set of hardware serial port, this group of serial port is often used to communicate with the computer,
- 5. //if you need to communicate with SV5W module serial port need to use software simulation serial port (soft serial port
- 6. #include <Wire.h>//Use iic communication
- 7. bool mp3_control = false;//bool variable, the state quantity of whether the voice module is in playback state
- 8. SoftwareSerial softSerialsv(10,11);//Soft serial port is defined, 10 is RX and 11 is TX
- 9. unsigned char data1=0x00;//Use an unsigned number instead of 0x00.Writing 0X00 directly would cause the write () method to be overloaded,
- 10. //requiring type determination
- 11. String recipt="";//Used to accept the return value of the SV5W module
- 12. int num=0;
- 13. void setup()
- **14**. ⊀
- 15. Wire.begin();//Initialize iic communication with no written address value in parentheses, that is, access as a host
- 16. Serial.begin(9600);//Serial communication with the computer
- 17. softSerialsv.begin(9600);//Soft serial communication with sv5w
- 18. softSerialsv.listen();//Listen softserial port softSerialsv
- 19. //Module state control sensor pin (button)
- 20. pinMode(2, INPUT);//Control play and pause, the next 2,3,4,5,6,7 functions will be implemented with flag==true
- 21. pinMode(3, INPUT);//Switch to previous song
- 22. pinMode(4, INPUT);//Switch to the next song
- 23. pinMode(5, INPUT);//Control volume plus
- 24. pinMode(6, INPUT);//Control volume reduction

- 25. pinMode(7, INPUT);//New songs and old songs classification, here uses the specified song instead of classification function
- 26. pinMode(8,OUTPUT);//Initialize pin 8 for sending interrupt trigger mode to slave 1
- 27. digitalWrite(8,LOW);//The default is low, and when high, slave 1 executes the interrupt function

```
28.
      }
29.
30.
      void loop()
31.
32.
       //Control module plays the song (key sensor high level trigger)
33.
       if((digitalRead(2) == HIGH)\&\&(mp3 control == false))
34.
       {
35.
36.
        delay(500);//Delay to shake
37.
38.
         mp3_control = true;//Change state mp3 control to true (playback
39.
state)
40.
41.
        //Send play command AA 02 00 AC
42.
        softSerialsv.write(0xAA);
43.
        softSerialsv.write(0x02);
44.
        softSerialsv.write(data1):
45.
         softSerialsv.write(0xAC);
46.
        delay(1000);
47.
48.
         //Send (current directory) sequential play instructions AA 18 01
07 CA
49.
        softSerialsv.write(0xAA);
50.
        softSerialsv.write(0x18);
51.
         softSerialsv.write(0x01);
52.
         softSerialsv.write(0x07);
53.
         softSerialsv.write(0xCA);
54.
        if(num==0)
55.
          writer2 1();
56.
          writer3 1();
57.
          num++:
58.
       }
59.
       inspect songQequence();
60.
        digitalWrite(8,LOW);
61.
       }
62.
       //Control module pause play (button sensor high level trigger)
       if((digitalRead(2) == HIGH)&&(mp3 control == true))
63.
```

```
64.
       {
65.
66.
         delay(500);//Delay to shake
67.
         mp3 control = false;//Change state mp3 control to false (paused
state)
68.
       //Send a pause command AA 03 00 AD
69.
         softSerialsv.write(0xAA);
70.
         softSerialsv.write(0x03);
71.
         softSerialsv.write(data1);
72.
        softSerialsv.write(0xAD);
73.
74.
         digitalWrite(8,HIGH);
75.
76.
77.
      //Control module plays the previous song (button sensor high level
trigger)
78.
       if((digitalRead(3) == HIGH)&&(mp3 control == true))
79.
       {
80.
81.
         delay(500);//Delay to shake
82.
83.
        //Send the last command AA 05 00 AF
84.
        softSerialsv.write(0xAA);
85.
         softSerialsv.write(0x05):
86.
         softSerialsv.write(data1);
87.
        softSerialsv.write(0xAF);
88.
        inspect songQequence();
89.
90.
       }
91.
      //Control module plays the next song (button sensor high level
trigger)
92.
        if((digitalRead(4) == HIGH)\&\&(mp3 control == true))
93.
       {
94.
95.
        delay(500); //Delay to shake
96.
97.
        //Send the next command AA 06 00 B0
98.
         softSerialsv.write(0xAA):
99.
        softSerialsv.write(0x06);
100.
        softSerialsv.write(data1);
101.
         softSerialsv.write(0xB0);
102.
        inspect_songQequence();
103.
104.
       }
```

```
105.
       //Control module volume plus (button sensor high level trigger)
        if((digitalRead(5) == HIGH)&&(mp3 control == true))
106.
107.
       {
108.
109.
        delay(500); //Delay to shake
110.
111.
        //Send the volume plus command AA 14 00 BE
112.
        softSerialsv.write(0xAA);
113.
        softSerialsv.write(0x14);
114.
        softSerialsv.write(data1);
115.
        softSerialsv.write(0xBE);
116.
       }
        //Control module volume reduction (button sensor high level
117.
trigger)
118.
       if((digitalRead(6) == HIGH)\&\&(mp3 control == true))
119.
       {
120.
121.
        delay(500);//Delay to shake
122.
123.
        //Send the volume decrement command AA 15 00 BF
124.
        softSerialsv.write(0xAA);
125.
        softSerialsv.write(0x15);
126.
        softSerialsv.write(data1);
127.
        softSerialsv.write(0xBF);
128.
        }//Control module new and old song classification (button sensor
high level trigger),
129.
        // here uses the specified song to replace the classification
function, the last three are to be changed
130.
       if((digitalRead(7) == HIGH))
131.
       {
132.
133.
        delay(500);//Delay to shake
134.
135.
        //Send the command to play the specified song AA 07 02 00 0B
BE
136.
        softSerialsv.write(0xAA);
        softSerialsv.write(0x07);
137.
        softSerialsv.write(0x02);
138.
139.
        softSerialsv.write(data1);
140.
        softSerialsv.write(0x0B);
141.
        softSerialsv.write(0xBE);
142.
        inspect_songQequence();
143.
       }
144.
```

```
145. }
146. // Query the current track, suitable for 0-99 tracks
147. void inspect songQequence(){
148.
       softSerialsv.write(0xAA);
149.
       softSerialsv.write(0x0D);
150.
       softSerialsv.write(data1);
151.
       softSerialsv.write(0xB7);
152.
       //The serial port hexadecimal conversion decimal
153.
       int i,j;
154.
       String sq="";
155.
       while (softSerialsv.available()) {
       int in = (char)softSerialsv.read();
156.
157.
       recipt+=in;
158.
       recipt+=',';
159.
       delay(2);
160.
       }
161.
       Serial.println(recipt);
162.
       if(recipt.length()==17) sq=(String)recipt.charAt(11);// 1-9 songs,
163.
       //these lines are converted with (String) because when you add
two character variables, the result is an integer,
164.
       //and when you add characters, you're actually adding their ASCII
valu
165.
        else sq=(String)recipt.charAt(11)+(String)recipt.charAt(12);//10 to
20 songs
166.
       Serial.println(sq);
167.
       writer2(sq);//Pass the song order to slave1
168.
       writer3(sq);//Pass the song sequence to slave2
169.
       recipt="";
170. }
171. // Send message to slave 2 (start playing)
172. void writer2 1(){
173.
       Wire.beginTransmission(2);//Start transferring data
174.
175.
       Wire.write('1');//Send a signal to start displaying subtitles
176.
177.
       Wire.endTransmission(); //Ending the transfer
178.
      }
      void writer3 1(){
179.
180.
       Wire.beginTransmission(3);//Start transferring data
181.
182.
       Wire.write('1');//Send a signal to start displaying subtitles
183.
184.
       Wire.endTransmission(); //Ending the transfer
185. }
```

```
186. // Send a message to Slave 2 (song sequence)
187. void writer2(String sq){
188.
       Wire.beginTransmission(2);//Start transferring data
189.
          if(sq.length()==1) Wire.write(sq.charAt(0));//Song
                                                                 number
difference
190.
       else {
191.
        Wire.write(sq.charAt(0));
192.
        Wire.write(sq.charAt(1));
193.
       }
194.
195.
       Wire.endTransmission(); //Ending the transfer
196.
197.
      //Send a message to Slave 3 (song sequence)
198. void writer3(String sq){
199.
       Wire.beginTransmission(3); //Start transferring data
200.
          if(sq.length()==1) Wire.write(sq.charAt(0));//Song
                                                                 number
difference
201.
       else {
202.
        Wire.write(sq.charAt(0));
203.
        Wire.write(sq.charAt(1));
204.
205.
       Wire.endTransmission(); //Ending the transfer
206.
207. }
```