Appendix A

Source code

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

```
// 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.
// And through the creation of soft serial port and SV5W module serial communication.
#include<SoftwareSerial.h>//Use a soft serial port
//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,
//if you need to communicate with SV5W module serial port need to use software simulation serial port (soft serial port
#include <Wire.h>//Use iic communication
bool mp3_control = false;//bool variable, the state quantity of whether the voice module is in playback state
SoftwareSerial softSerialsv(10,11);//Soft serial port is defined, 10 is RX and 11 is TX
unsigned char data1=0x00;//Use an unsigned number instead of 0x00.Writing 0X00 directly would cause the write () method to be overloaded,
//requiring type determination
String recipt="";//Used to accept the return value of the SV5W module
int num=0;
void setup()
{
  Wire.begin();//Initialize iic communication with no written address value in parentheses, that is, access as a host
  Serial.begin(9600);//Serial communication with the computer
  softSerialsv.begin(9600);//Soft serial communication with sv5w
  softSerialsv.listen();//Listen softserial port softSerialsv
//Module state control sensor pin (button)
 pinMode(2, INPUT);//Control play and pause, the next 2,3,4,5,6,7 functions will be implemented with flag==true
  pinMode(3, INPUT);//Switch to previous song
  pinMode(4, INPUT);//Switch to the next song
  pinMode(5, INPUT);//Control volume plus
  pinMode(6, INPUT);//Control volume reduction
```

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

void loop()
{
  //Control module plays the song (key sensor high level trigger)
  if((digitalRead(2) == HIGH)&&(mp3_control == false))
  {

    delay(500);//Delay to shake


    mp3_control = true;//Change state mp3 control to true (playback state)

    //Send play command  AA 02 00 AC
    softSerialsv.write(0xAA);
    softSerialsv.write(0x02);
    softSerialsv.write(data1);
    softSerialsv.write(0xAC);
    delay(1000);

    //Send (current directory) sequential play instructions AA 18 01 07 CA
    softSerialsv.write(0xAA);
    softSerialsv.write(0x18);
    softSerialsv.write(0x01);
    softSerialsv.write(0x07);
    softSerialsv.write(0xCA);
    if(num==0){
      writer2_1();
      writer3_1();
      num++;
  }
  inspect_songQequence();
    digitalWrite(8,LOW);
  }
  //Control module pause play (button sensor high level trigger)
  if((digitalRead(2) == HIGH)&&(mp3_control == true))
```

```
    {

      delay(500);//Delay to shake
      mp3_control = false;//Change state mp3 control to false (paused state)
   //Send a pause command AA 03 00 AD
      softSerialsv.write(0xAA);
      softSerialsv.write(0x03);
      softSerialsv.write(data1);
      softSerialsv.write(0xAD);

       digitalWrite(8,HIGH);

    }
  //Control module plays the previous song (button sensor high level trigger)
    if((digitalRead(3) == HIGH)&&(mp3_control == true))
    {

      delay(500);//Delay to shake

      //Send the last command AA 05 00 AF
      softSerialsv.write(0xAA);
      softSerialsv.write(0x05);
      softSerialsv.write(data1);
      softSerialsv.write(0xAF);
      inspect_songQequence();

    }
  //Control module plays the next song (button sensor high level trigger)
     if((digitalRead(4) == HIGH)&&(mp3_control == true))
    {

      delay(500); //Delay to shake

      //Send the next command AA 06 00 B0
      softSerialsv.write(0xAA);
      softSerialsv.write(0x06);
      softSerialsv.write(data1);
      softSerialsv.write(0xB0);
      inspect_songQequence();

    }
```

```
//Control module volume plus (button sensor high level trigger)
 if((digitalRead(5) == HIGH)&&(mp3_control == true))
{

  delay(500); //Delay to shake

  //Send the volume plus command AA 14 00 BE
  softSerialsv.write(0xAA);
  softSerialsv.write(0x14);
  softSerialsv.write(data1);
  softSerialsv.write(0xBE);
}
 //Control module volume reduction (button sensor high level trigger)
if((digitalRead(6) == HIGH)&&(mp3_control == true))
{

  delay(500);//Delay to shake

 //Send the volume decrement command AA 15 00 BF
  softSerialsv.write(0xAA);
  softSerialsv.write(0x15);
  softSerialsv.write(data1);
  softSerialsv.write(0xBF);
}//Control module new and old song classification (button sensor high level trigger),
 // here uses the specified song to replace the classification function, the last three are to be changed
if((digitalRead(7) == HIGH))
{

  delay(500);//Delay to shake

  //Send the command to play the specified song AA 07 02 00 0B BE
  softSerialsv.write(0xAA);
  softSerialsv.write(0x07);
  softSerialsv.write(0x02);
  softSerialsv.write(data1);
  softSerialsv.write(0x0B);
  softSerialsv.write(0xBE);
  inspect_songQequence();
}

```

```
}
// Query the current track, suitable for 0-99 tracks
void inspect_songQequence(){
  softSerialsv.write(0xAA);
  softSerialsv.write(0x0D);
  softSerialsv.write(data1);
  softSerialsv.write(0xB7);
  //The serial port hexadecimal conversion decimal
  int i,j;
  String sq="";
  while (softSerialsv.available()) {
  int in = (char)softSerialsv.read();
  recipt+=in;
  recipt+=',';
  delay(2);
  }
  Serial.println(recipt);
  if(recipt.length()==17) sq=(String)recipt.charAt(11);// 1-9 songs,
  //these lines are converted with (String) because when you add two character variables, the result is an integer,
  //and when you add characters, you're actually adding their ASCII valu
  else sq=(String)recipt.charAt(11)+(String)recipt.charAt(12);//10 to 20 songs
  Serial.println(sq);
  writer2(sq);//Pass the song order to slave1
  writer3(sq);//Pass the song sequence to slave2
  recipt="";
}
// Send message to slave 2 (start playing)
void writer2_1(){
  Wire.beginTransmission(2);//Start transferring data

  Wire.write('1');//Send a signal to start displaying subtitles

  Wire.endTransmission(); //Ending the transfer
}
void writer3_1(){
  Wire.beginTransmission(3);//Start transferring data

  Wire.write('1');//Send a signal to start displaying subtitles

  Wire.endTransmission(); //Ending the transfer
}
```

```
// Send a message to Slave 2 (song sequence)
void writer2(String sq){
  Wire.beginTransmission(2);//Start transferring data
     if(sq.length()==1) Wire.write(sq.charAt(0));//Song number difference
  else {
    Wire.write(sq.charAt(0));
    Wire.write(sq.charAt(1));
  }

  Wire.endTransmission(); //Ending the transfer
 }
 //Send a message to Slave 3 (song sequence)
void writer3(String sq){
  Wire.beginTransmission(3); //Start transferring data
     if(sq.length()==1) Wire.write(sq.charAt(0));//Song number difference
  else {
    Wire.write(sq.charAt(0));
    Wire.write(sq.charAt(1));
  }
  Wire.endTransmission(); //Ending the transfer

 }
```