Lesson 4 Play Music

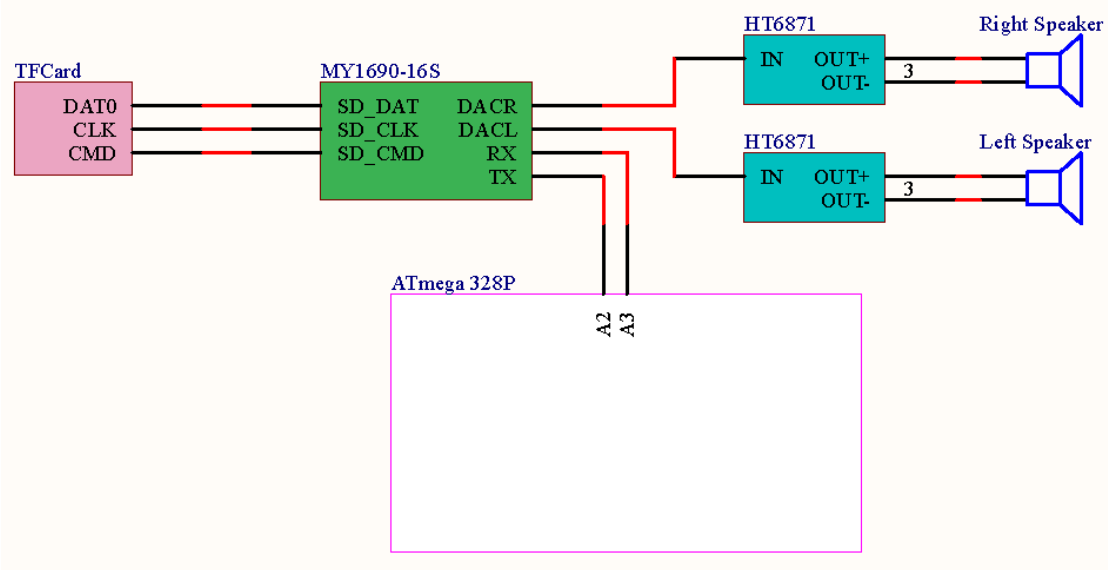
CONTENT

Ⅰ. Brief Introduction

In this chapter you will learn the principle of penguin bot playing music and how to program it to switch music.

Ⅱ. Principle of Penguin Bot Playing Music

Schematic Diagram



Components needed to play music

- **TF Card**: save music files
- **MY1690-16S audio decoding chip**: audio processing center, receive control signal of MCU and decode the audio files in SD card then output audio signal to the operational amplifier
- **Atmega328P MCU**: play music like switching music and control the volume
- **Ht6871 power amplifier chip**: receive signal of decoding chip then amplify and output it to the speaker
- **Speaker**: play the audio signal

Now you have a basic understanding of each component's function, what you need to do is to program Atmega328P MCU to control music playing. Atmega328P MCU connects to MY1690-16S Serial Port via Pin A2 and A3, which means you need to set A2 and A3 as software serial port as to communicate with MY1690-16S. Our program used NeoSWSerial libraries as software communication serial port to control music playing and since we've already written the underlying program for music playing so you only need to call in the function of MP3 class to play music.

```
class MY1690_16S {
public:
  int volume;
  String playStatus[5] = {"0", "1", "2", "3", "4"}; // STOP PLAYING
PAUSE FF FR
  void playSong(unsigned char num, unsigned char vol) {
    setVolume(vol);
    setPlayMode(4);
    CMD_SongSelet[4] = num;
    checkCode(CMD_SongSelet);
    mp3Serial.write(CMD_SongSelet, 7);
    delay(10);
  };
  String getPlayStatus() {
    mp3Serial.write(CMD_getPlayStatus, 5);
    delay(10);
    return getStatus();
  }
  String getStatus() {
    String statusMp3 = "";
    while (mp3Serial.available()) {
      statusMp3 += (char)mp3Serial.read();
```

play music

obtain playing status

```
  }
  return statusMp3;
};
void stopPlay() {
  setPlayMode(4);
  mp3Serial.write(CMD_MusicStop, 5);
  delay(10);
};
void setVolume(unsigned char vol) {
  CMD_VolumeSet[3] = vol;
  checkCode(CMD_VolumeSet);
  mp3Serial.write(CMD_VolumeSet, 6);
  delay(10);
};
void volumePlus(){
  mp3Serial.write(CMD_VolumePlus, 5);
  delay(10);
};
void volumeDown(){
  mp3Serial.write(CMD_VolumeDown, 5);
  delay(10);
};
void setPlayMode(unsigned char mode) {
  CMD_PlayMode[3] = mode;
  checkCode(CMD_PlayMode);
  mp3Serial.write(CMD_PlayMode, 6);
  delay(10);
};
void checkCode(unsigned char *vs) {
  int val = vs[1];
  int i;
  for (i = 2; i < vs[1]; i++) {
    val = val ^ vs[i];
  }
  vs[i] = val;
};
void ampMode(int p, bool m) {
  pinMode(p, OUTPUT);
  if (m) {
    digitalWrite(p, HIGH);
  } else {
    digitalWrite(p, LOW);
  }
};
```

stop playing

set up volume

turn up the volume

turn down the volume

set up playing mode

```
  void init() {
    ampMode(HT6871_PIN, HIGH);
    stopPlay();
    volume = 15;
  }
private:
  byte CMD_MusicPlay[5] = {0x7E, 0x03, 0x11, 0x12, 0xEF};
  byte CMD_MusicStop[5] = {0x7E, 0x03, 0x1E, 0x1D, 0xEF};
  byte CMD_MusicNext[5] = {0x7E, 0x03, 0x13, 0x10, 0xEF};
  byte CMD_MusicPrev[5] = {0x7E, 0x03, 0x14, 0x17, 0xEF};
  byte CMD_VolumePlus[5] = {0x7E, 0x03, 0x15, 0x16, 0xEF};
  byte CMD_VolumeDown[5] = {0x7E, 0x03, 0x16, 0x15, 0xEF};
  byte CMD_VolumeSet[6] = {0x7E, 0x04, 0x31, 0x00, 0x00, 0xEF};
  byte CMD_PlayMode[6] = {0x7E, 0x04, 0x33, 0x00, 0x00, 0xEF};
  byte CMD_SongSelet[7] = {0x7E, 0x05, 0x41, 0x00, 0x00, 0x00, 0xEF};
  byte CMD_getPlayStatus[5] = {0x7E, 0x03, 0x20, 0x23, 0xEF};
} MP3;
```

Music playing initialization

MY1690-16S communication order

Ⅲ. Write Program of Playing Music

The sketch used in this chapter is saved in below path and please refer to Upload Penguin Bot program and upload the codes then copy the music provided into the TF card.

\ELEGOO Penguin Bot V2.0\Penguin Bot Function Introduction\ Lesson 4 Play Music\PlayMusic\PlayMusic.ino

Code reviews:

```
#include "NeoSWSerial.h"
#define HT6871_PIN 7
#define SOFTWARE_RXD A2
#define SOFTWARE_TXD A3

NeoSWSerial mp3Serial(SOFTWARE_RXD,SOFTWARE_TXD);

class MY1690_16S {
public:
  int volume;
  String playStatus[5] = {"0", "1", "2", "3", "4"}; // STOP PLAYING
PAUSE FF FR
  void playSong(unsigned char num, unsigned char vol) {
```

libraries for serial port

define the pins of software serial port

define the objects of software serial

write a class including all the function of music playing

```
  setVolume(vol);
  setPlayMode(4);
  CMD_SongSelet[4] = num;
  checkCode(CMD_SongSelet);
  mp3Serial.write(CMD_SongSelet, 7);
  delay(10);
};
String getPlayStatus() {
  mp3Serial.write(CMD_getPlayStatus, 5);
  delay(10);
  return getStatus();
}
String getStatus() {
  String statusMp3 = "";
  while (mp3Serial.available()) {
    statusMp3 += (char)mp3Serial.read();
  }
  return statusMp3;
};
void stopPlay() {
  setPlayMode(4);
  mp3Serial.write(CMD_MusicStop, 5);
  delay(10);
};
void setVolume(unsigned char vol) {
  CMD_VolumeSet[3] = vol;
  checkCode(CMD_VolumeSet);
  mp3Serial.write(CMD_VolumeSet, 6);
  delay(10);
};
void volumePlus(){
  mp3Serial.write(CMD_VolumePlus, 5);
  delay(10);
};
void volumeDown(){
  mp3Serial.write(CMD_VolumeDown, 5);
  delay(10);
};
void setPlayMode(unsigned char mode) {
  CMD_PlayMode[3] = mode;
  checkCode(CMD_PlayMode);
  mp3Serial.write(CMD_PlayMode, 6);
  delay(10);
};
```

```
  void checkCode(unsigned char *vs) {
    int val = vs[1];
    int i;
    for (i = 2; i < vs[1]; i++) {
      val = val ^ vs[i];
    }
    vs[i] = val;
  };
  void ampMode(int p, bool m) {
    pinMode(p, OUTPUT);
    if (m) {
      digitalWrite(p, HIGH);
    } else {
      digitalWrite(p, LOW);
    }
  };
  void init() {
    ampMode(HT6871_PIN, HIGH);
    stopPlay();
    volume = 15;
  }
private:
  byte CMD_MusicPlay[5] = {0x7E, 0x03, 0x11, 0x12, 0xEF};
  byte CMD_MusicStop[5] = {0x7E, 0x03, 0x1E, 0x1D, 0xEF};
  byte CMD_MusicNext[5] = {0x7E, 0x03, 0x13, 0x10, 0xEF};
  byte CMD_MusicPrev[5] = {0x7E, 0x03, 0x14, 0x17, 0xEF};
  byte CMD_VolumePlus[5] = {0x7E, 0x03, 0x15, 0x16, 0xEF};
  byte CMD_VolumeDown[5] = {0x7E, 0x03, 0x16, 0x15, 0xEF};
  byte CMD_VolumeSet[6] = {0x7E, 0x04, 0x31, 0x00, 0x00, 0xEF};
  byte CMD_PlayMode[6] = {0x7E, 0x04, 0x33, 0x00, 0x00, 0xEF};
  byte CMD_SongSelet[7] = {0x7E, 0x05, 0x41, 0x00, 0x00, 0x00, 0xEF};
  byte CMD_getPlayStatus[5] = {0x7E, 0x03, 0x20, 0x23, 0xEF};
} MP3;

void setup() {
  mp3Serial.begin(9600);
  MP3.init();
  MP3.playSong(1, 20);// play song 0001, set volume 20
}
void loop() {}
```

define a class named MP3

start software serial port

MP3 initialization and start playing the first music at volume of 20

Call in MP3.init() in setup() to proceed initialization then you play music with MP3.playSong()

Ⅳ. Switch Music

Since the underlying program has been written, the only program you need to write is”MP3.playSong(1, 20);”.

There are 2 parameters of Play Song. One is to select the serial number of music file from 1-1000,which needs to be preset in TF card like 0001 represents the first song and 0002 represents the second song and so on. The other parameter is volume which can be set from 0 to 30.

So if you want to play the second song at the highest volume, what should you would you need to do? Well, the answer is very simple.

1. Copy a music file into TF card
2. Name it as 0002.mp3
3. then find ”MP3.playSong(1, 20); in above codes
4. modify it as ”MP3.playSong(2, 30);