按键控制 MP3 模块随机播放音乐

设计者：STCode （公众号同名）

1）功能描述

这个设计主要是通过按键来控制播放音乐，主要涉及到的内容有按键和 YX5300 MP3 音乐模块的使用，通过按压按键来达到随机播放音乐曲目的目的。

2）使用主要器件

1、Arduino Uno 控制板

2、YX5300 MP3 音乐模块

3、SD 卡

4、读卡器

5、按键

6、面包板

7、导线

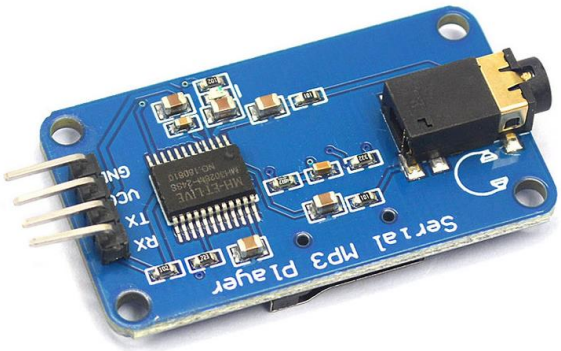
8、扬声器

3）元件介绍

1、YX5300 MP3 音乐模块

该模块是一个基于高音质 MP3 音乐芯片的 MP3 音乐播放器模块。支持采样率是 8KHz ~ 48KHz 的 MP3、WAV 格式文件。

板载 Micro SD 卡座，可插上存了音乐文件的 micro SD 卡。单片机可以通过串口发送命令进行切换音乐、调节音量、播放模式等操作。用户也可以通过 USB 转串口模块对该模块进行调试。模块与 UNO/AVR/ARM/PIC 等单片机系统兼容。UART TTL 串口控制播放模式，串口波特率为 9600bps，供电电源可为 3.2 ~ 5.2V DC.



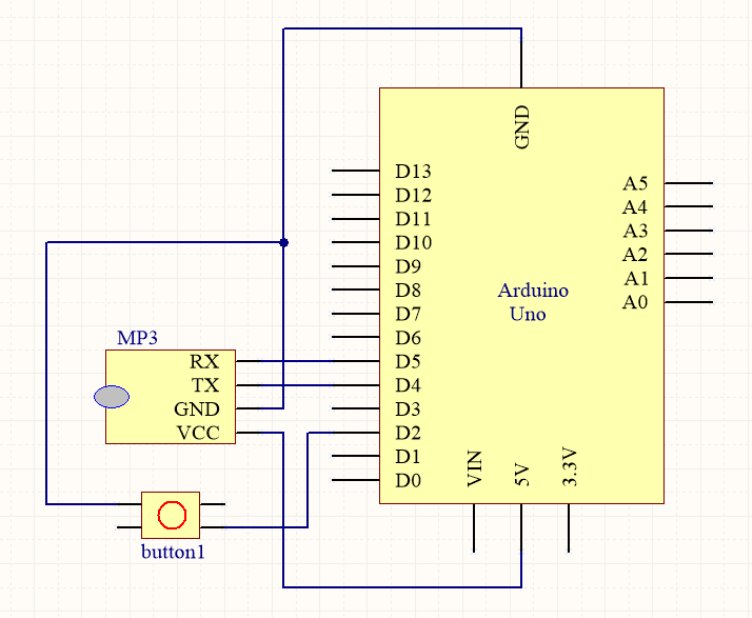
4）电路设计

Arduino----------YX5300 MP3

D4----------TX

D5----------RX

按键一端接 D2，另一端接到 GND，整体简易接线图如下所示：



5）关键代码讲解

```
#include <SoftwareSerial.h>  //添加软串口头文件
SoftwareSerial mySerial(4,5); // 音乐模块 TX 接 4，音乐模块 RX 接 5
```

首先引入软串口头文件，定义音乐模块 TX\RX 端口的接线

```
unsigned char play_song1[8] = {0x7E,0xFF,0x06,0x08,0x00,0x00,0x01,0xEF}
;  //1
unsigned char play_song2[8] = {0x7E,0xFF,0x06,0x03,0x00,0x00,0x02,0xEF}
;  //2
```

上面定义两个字符变量，用于存储指令，其中 play_song1[8]指令为播放歌曲 1，play_song2[8]指令播放歌曲 2，可以从倒数第二个 16 进制数据看出，播放歌曲 1 用 0x01，播放歌曲 2 用 0x02，以此类推；整数第四个 16 进制数据为设置循环播放还是播放一次，其中 0x08 指令为单曲循环，0x03 为只播放一次。

相关指令可以参考下图：

Command	Command bytes without checksum(HEX)	Remark
[Next Song]	7E FF 06 **01** 00 00 00 EF	
[Previous Song]	7E FF 06 **02** 00 00 00 EF	
[Play with index]	7E FF 06 **03** 00 **00 01** EF	Play the first song
	7E FF 06 **03** 00 **00 02** EF	Play the second song
[Volume up]	7E FF 06 **04** 00 00 00 EF	Volume increased one
[Volume down]	7E FF 06 **05** 00 00 00 EF	Volume decrease one
[Set volume]	7E FF 06 **06** 00 **00 1E** EF	Set the volume to 30 (0x1E is 30)
[Single cycle play]	7E FF 06 **08** 00 **00 01** EF	Single cycle play the first song
[Select device]	7E FF 06 **09** 00 **00 02** EF	Select storage device to TF card
[Sleep mode]	7E FF 06 **0A** 00 00 00 EF	Chip enters sleep mode
[Wake up]	7E FF 06 **0B** 00 00 00 EF	Chip wakes up
[Reset]	7E FF 06 **0C** 00 00 00 EF	Chip reset
[Play]	7E FF 06 **0D** 00 00 00 EF	Resume playback
[Pause]	7E FF 06 **0E** 00 00 00 EF	Playback is paused
[Play with folder and file name]	7E FF 06 **0F** 00 **01 01** EF	Play the song with the directory: /01/001xxx.mp3
	7E FF 06 **0F** 00 **01 02** EF	Play the song with the directory: /01/002xxx.mp3
[Stop play]	7E FF 06 **16** 00 00 00 EF	
[Cycle play with folder name]	7E FF 06 **17** 00 00 **01** EF	01 folder cycle play
[Shuffle Play]	7E FF 06 **18** 00 00 00 EF	
[Set single cycle play]	7E FF 06 **19** 00 00 **00** EF	Start up single cycle play

```
unsigned char pause[8]={0x7E,0xFF,0x06,0x0E,0x00,0x00,0x00,0xEF};
   //暂停播放
unsigned char play[8]={0x7E,0xFF,0x06,0x0D,0x00,0x00,0x00,0xEF};
   //恢复播放
unsigned char stop_play[8] = {0x7E,0xFF,0x06,0x16,0x00,0x00,0x00,0xEF};
   //停止播放指令
unsigned char top_volume[8]={0x7E,0xFF,0x06,0x06,0x00,0x00,0x1E,0xEF};
  //设置最大音量
```

此代码同时用到了上述指令

```
  mySerial.write(top_volume, 8); //设置音量为最大
  mySerial.write(stop_play, 8);  //开机静音
  randomSeed(analogRead(A0));    //随机数种子 A0
```

上述指令设置了播放的音量以及开机默认静音，还设置了随机数种子，用于按键按下后，随机播放一首歌曲。

随机播放歌曲可以用 switch 语句

```
switch (num)       //根据随机数随机播放曲目
  {
    case 1: mySerial.write(play_song1,8);
    break;
    case 2: mySerial.write(play_song2,8);
    break;
```

```
    case 3: mySerial.write(play_song3,8);
    break;
  default:
    break;
  }
```

下面是按键扫描子函数

```
void key_scan()    //按键扫描子函数
{
  if(digitalRead(keypin)==LOW)
  {
    delay(20);        //消抖
    if(digitalRead(keypin)==LOW)
    {
      num=random(1,11);  //随机数获取
      start_flag=1;      //播放标志位置 1
    }
  }
}
```

6）效果演示