

语音播报小助手

1. 学习目标

本次课程我们主要学习使用 **arduino** 和语音识别模块 **V3** 版、语音合成播报模块制作语音播报小助手。

2. 课前准备

语音识别模块、语音合成播报模块采用的是 **I2C** 通讯，将模块的 **SDA**, **SCL** 分别连接 **arduino UNO** 板子的 **SDA** 和 **SCL** 引脚。**VCC** 和 **GND** 分别连接 **arduino UNO** 的 **5V** 和 **GND**。

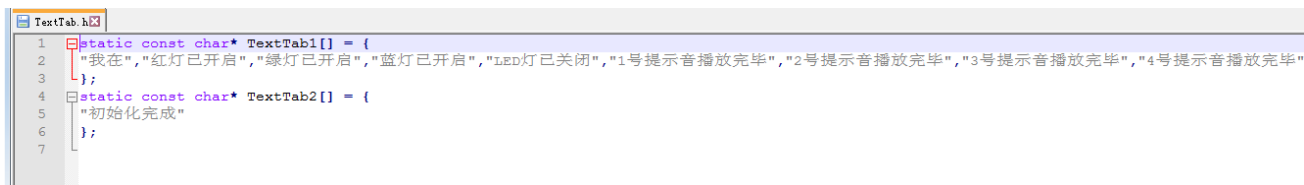
3. 程序

将语音识别模块的相关配置函数定义在 **ASR.cpp** 和 **ASR.h** 文件下，将语音合成播报模块的相关函数定义 **XFS.cpp** 和 **XFS.h**

```
Voice_broadcast_assistant ASR.cpp ASR.h TextTab.h XFS.cpp XFS.h
#include "ASR.h"

/*****
寄存器设置函数
*****/
bool I2CWrite(unsigned char reg_addr, unsigned char date)
{
    Wire.beginTransmission(ASR_I2C_ADDR); //发送Device地址
    Wire.write(reg_addr);                //发送要操作的寄存器地址
    Wire.write(date);                    //发送要设置的值
    if(Wire.endTransmission() != 0)      //发送结束信号
    {
        delay(10);
        return false;
    }
    delay(10);
}
```

由于 arduinoIDE 为 UTF-8 编码问题，将中文字符串定义在 TextTab.h 文件下，并使用文本软件保存为 GBK 编码方式。



```

1 static const char* TextTab1[] = {
2   "我在", "红灯已开启", "绿灯已开启", "蓝灯已开启", "LED灯已关闭", "1号提示音播放完毕", "2号提示音播放完毕", "3号提示音播放完毕", "4号提示音播放完毕"
3 };
4 static const char* TextTab2[] = {
5   "初始化完成"
6 };
7

```

初始化语音合成播报模块设置，注意这里的地址为 0x30，如果用的是旧版的语音合成播报模块那么为 0x50, 根据自己使用的播报模块版本设置相应的值。

```

static void XFS_Init()
{

    xfs.Begin(0x30); //设备i2c地址，地址为新版地址为0x30, 旧版地址为0x50
    delay(n);
    xfs.SetReader(XFS5152CE::Reader_XiaoYan); //设置发音人
    delay(n);
    xfs.SetEncodingFormat(XFS5152CE::GB2312); //文本的编码格式
    delay(n);
    xfs.SetLanguage(xfs.Language_Auto); //语种判断
    delay(n);
    xfs.SetStyle(XFS5152CE::Style_Continue); //合成风格设置
    delay(n);
    xfs.SetArticulation(XFS5152CE::Articulation_Letter); //设置单词的发音方式
    delay(n);
    xfs.SetSpeed(5); //设置语速1~10
    delay(n);
    xfs.SetIntonation(5); //设置语调1~10
    delay(n);
    xfs.SetVolume(10); //设置音量1~10
    delay(n);
}

```

设置语音识别的模式和词组

```

# if 1
I2CWrite(ASR_CLEAR_ADDR, 0x40); //清除掉电保存区, 录入前需要清除掉电保存区
BusyWait();
I2CWrite(ASR_MODE_ADDR, 1); //设置检测模式
BusyWait();
AsrAddWords(0, "xiao ya");
BusyWait();
AsrAddWords(1, "hong deng");
BusyWait();
AsrAddWords(2, "lv deng");
BusyWait();
AsrAddWords(3, "lan deng");
BusyWait();
AsrAddWords(4, "guan deng");
BusyWait();
AsrAddWords(5, "bo fang yi hao ti shi yin");
BusyWait();
AsrAddWords(6, "bo fang er hao ti shi yin");
BusyWait();
AsrAddWords(7, "bo fang san hao ti shi yin");
BusyWait();
AsrAddWords(8, "bo fang si hao ti shi yin");
BusyWait();
AsrAddWords(9, "bo fang wu hao ti shi yin");
BusyWait();
AsrAddWords(10, "qie huan bo yin ren xiao yan");
BusyWait();
AsrAddWords(11, "qie huan bo yin ren xu jiu");
BusyWait();
AsrAddWords(12, "qie huan bo yin ren xu duo");
BusyWait();
AsrAddWords(13, "qie huan bo yin ren xiao pin");
BusyWait();
AsrAddWords(14, "qie huan bo yin ren tan lao ya");
BusyWait();
AsrAddWords(15, "qie huan bo yin ren xu xiao bao");
BusyWait();
while(cleck != 16)
{
    WireReadData(ASR_NUM_CLECK, &cleck, 1);
}

```

读取识别结果序号并作出相应的语音播报和设置。

```

WireReadData(ASR_RESULI, &result, 1); //读取识别序号值，并赋值给result，默认是0xff
delay(100);
Serial.println(result);

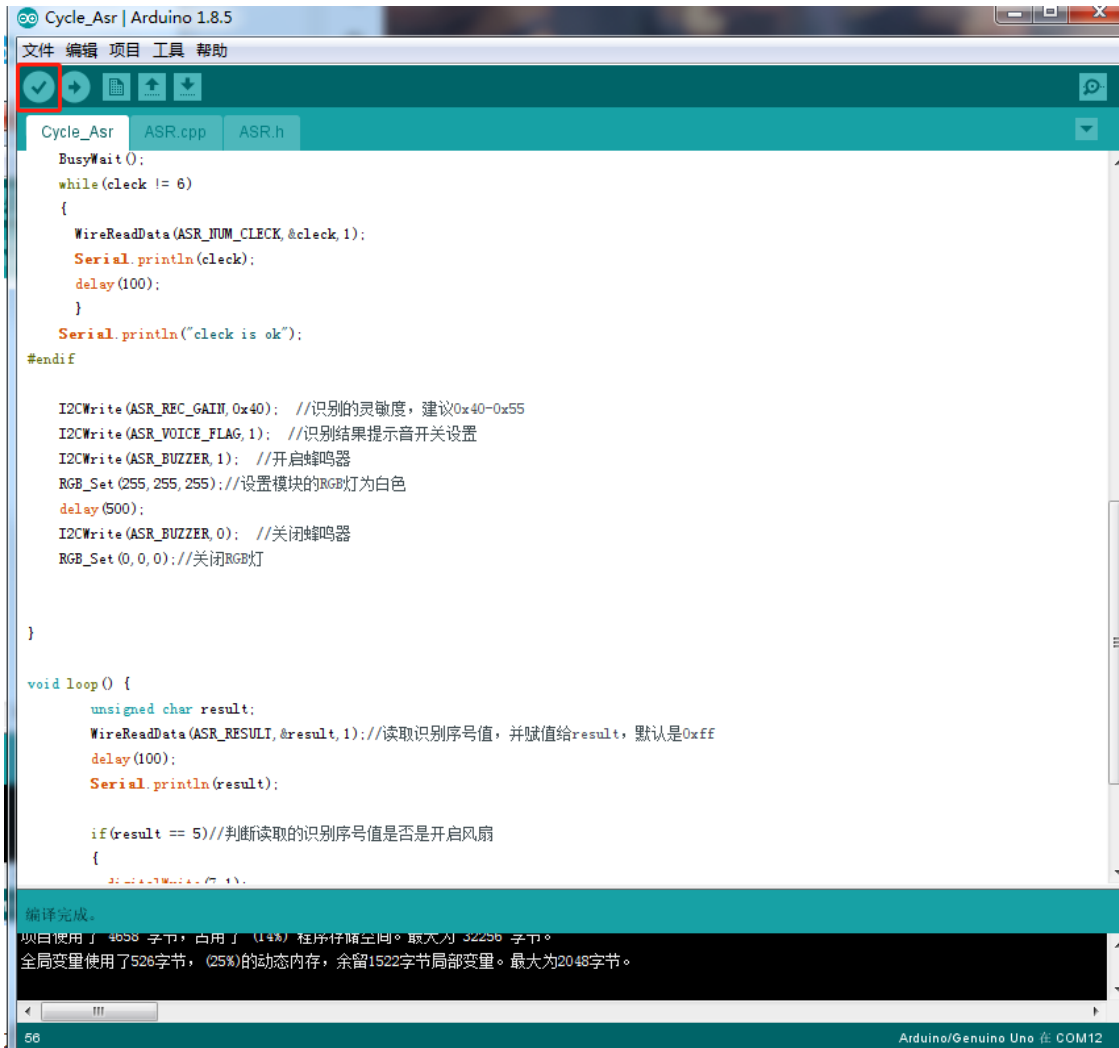
if(result == 0)//小亚
{
    xfs.StartSynthesis(TextTab1[0]);
    while(xfs.GetChipStatus() != xfs.ChipStatus_Idle)
    {

        delay(30);
    }
}
else if(result == 1)//红灯
{
    xfs.StartSynthesis(TextTab1[1]);
    while(xfs.GetChipStatus() != xfs.ChipStatus_Idle)
    {
        delay(30);
    }
}
else if(result == 2)//绿灯
{
    xfs.StartSynthesis(TextTab1[2]);
    while(xfs.GetChipStatus() != xfs.ChipStatus_Idle)
    {
        delay(30);
    }
}
else if(result == 3)//蓝灯
{
    xfs.StartSynthesis(TextTab1[3]);
    while(xfs.GetChipStatus() != xfs.ChipStatus_Idle)
    {

```

4. 编译下载程序

4.1 我们需要通用 Arduino IDE 软件打开文件，然后点击菜单栏中的“√”编译程序，并且等待左下角出现“编译成功”的字样。



```
Cycle_Asr | Arduino 1.8.5
文件 编辑 项目 工具 帮助
Cycle_Asr ASR.cpp ASR.h

BusyWait();
while(cleack != 6)
{
    WireReadData(ASR_I2C_ADDR, &cleack, 1);
    Serial.println(cleack);
    delay(100);
}
Serial.println("cleack is ok");
#endif

I2CWrite(ASR_REC_GAIN, 0x40); //识别的灵敏度, 建议0x40-0x55
I2CWrite(ASR_VOICE_FLAG, 1); //识别结果提示音开关设置
I2CWrite(ASR_BUZZER, 1); //开启蜂鸣器
RGB_Set(255, 255, 255); //设置模块的RGB灯为白色
delay(500);
I2CWrite(ASR_BUZZER, 0); //关闭蜂鸣器
RGB_Set(0, 0, 0); //关闭RGB灯

}

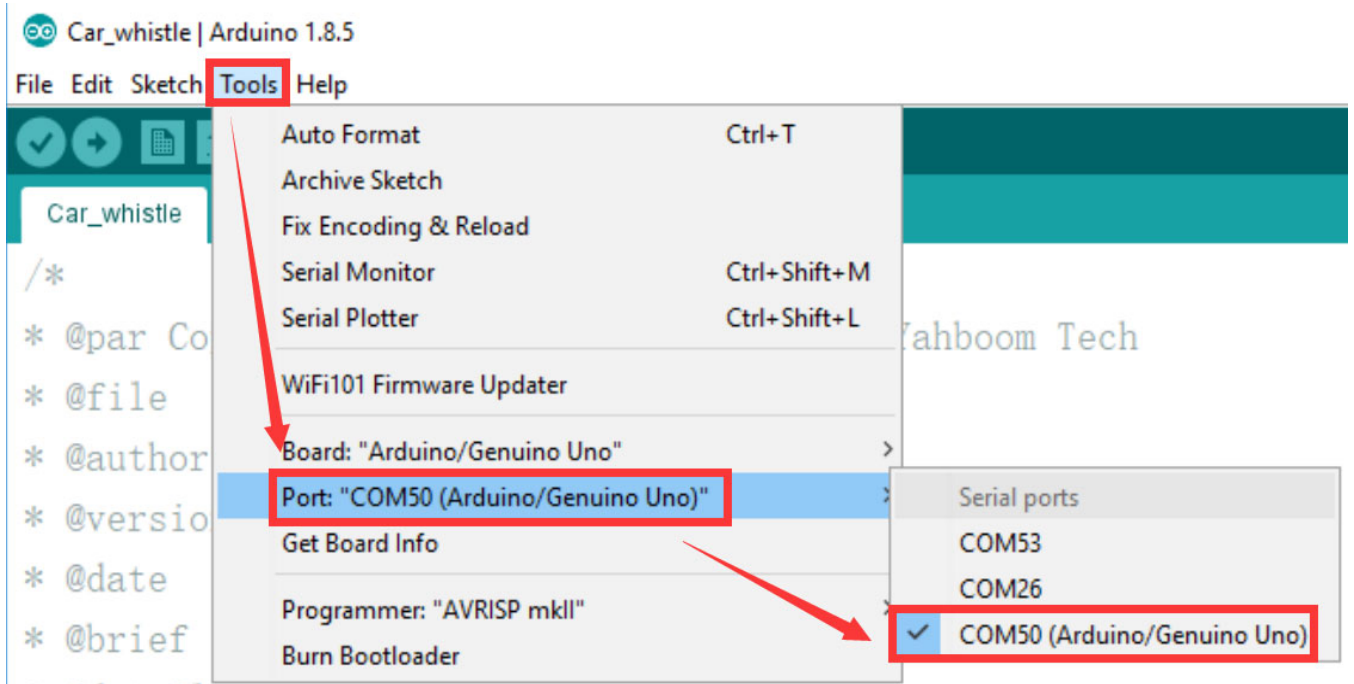
void loop() {
    unsigned char result;
    WireReadData(ASR_RESULT, &result, 1); //读取识别序号值, 并赋值给result, 默认是0xFF
    delay(100);
    Serial.println(result);

    if(result == 5) //判断读取的识别序号值是否是开启风扇
    {
        // ...
    }
}

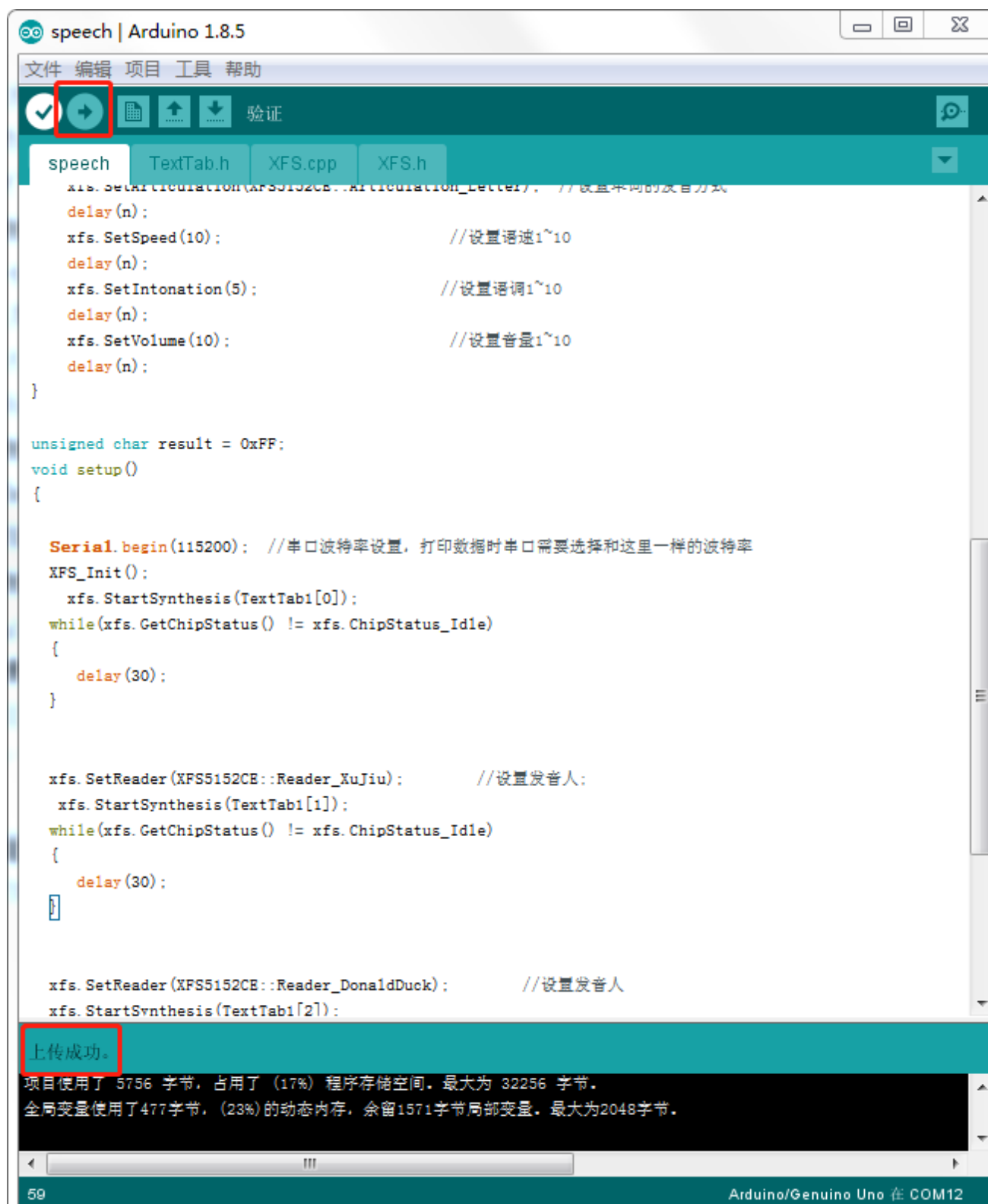
编译完成。
项目使用了 4658 字节, 占用了 (14%) 程序存储空间。最大为 32256 字节。
全局变量使用了 526 字节, (25%) 的动态内存, 余留 1522 字节局部变量。最大为 2048 字节。

56 Arduino/Genuino Uno 在 COM12
```

4.2 在 Arduino IDE 的菜单栏中, 我们需要选择【工具】---【端口】---选择设备管理器中刚刚显示端口号, 如下图所示。



4.3 选择完成后，点击菜单栏下的“→”将代码上传到 UNO 板。当左下角出现“上传完成”字样时，表示程序已成功上传到 UNO 板，如下图所示。



5. 实验现象

程序下载成功以后，如果设置为条件为 1，则需要等待清除缓存区、设置模式、录入识别词组等，并检测录入词组数是否与实际读取到模块已经录入的词组数对应并打印出录入的词组数目，正确则继续往下执行，错误则陷入循环无法退出重新按下 **arduino** 复位键重新启动程序，等待设置完毕后模块的 **RGB** 灯会亮起白色 **1s** 和鸣笛 **1s** 并播报“初始化完成”。之后可以开始进行语音识别之前录入的词组，可通过与语音识别设置和关闭 **LED** 灯，播放 **1-5** 号提示音，语音切换播音人。如果之前已经录入过了且无需修改可以把条件改为 **0**，可以跳过清除缓存和录入词条的时间。