# IIS模块的使用

## 电路连接

VCC→3.3V GND→GND

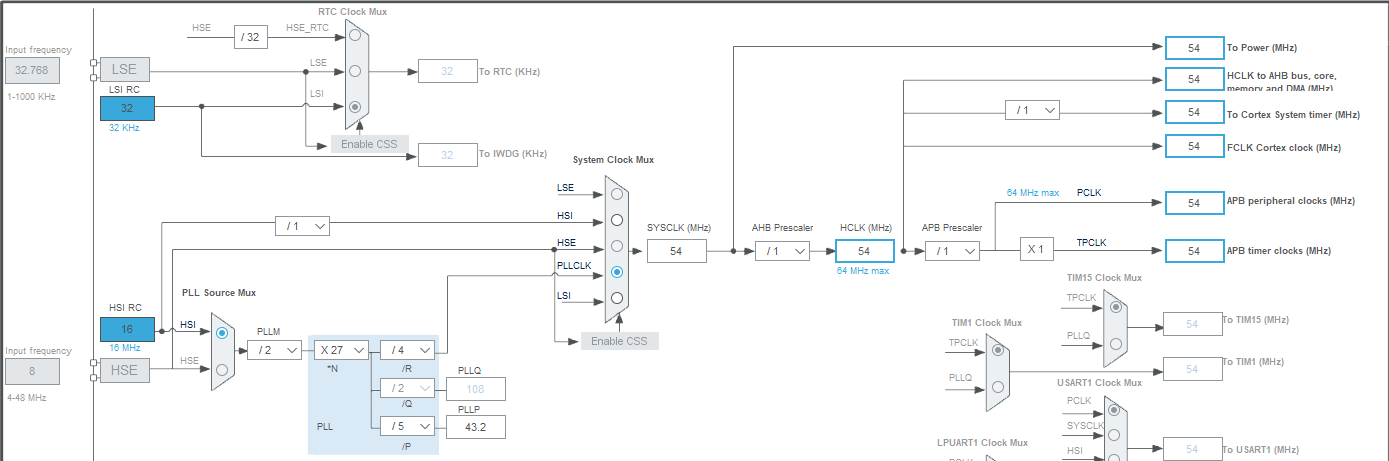
SDA→I2C1\_SDA (PB9) SCL→I2C1\_SCL (PB8)

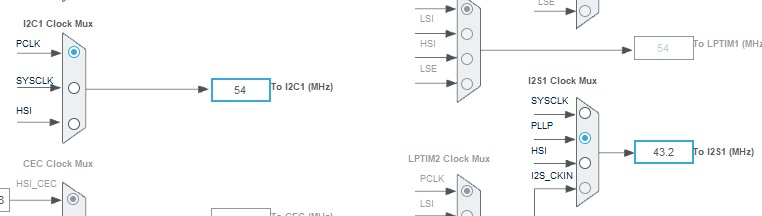
CLK→I2S1\_CK\_B (PA1) WS→I2S1\_WS\_B (PA4）

RXSDA→I2S1\_SD(PA2) RXMCLK→同步时钟悬空

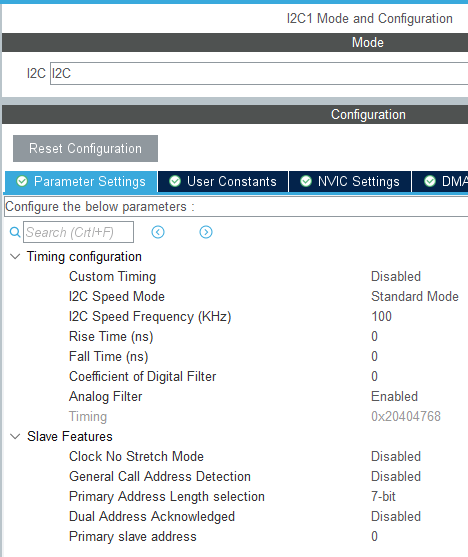
## 创建工程

按照下图所示设置系统时钟

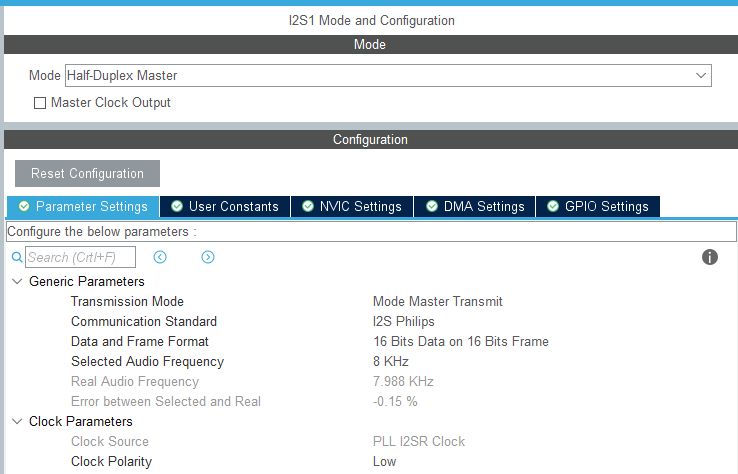




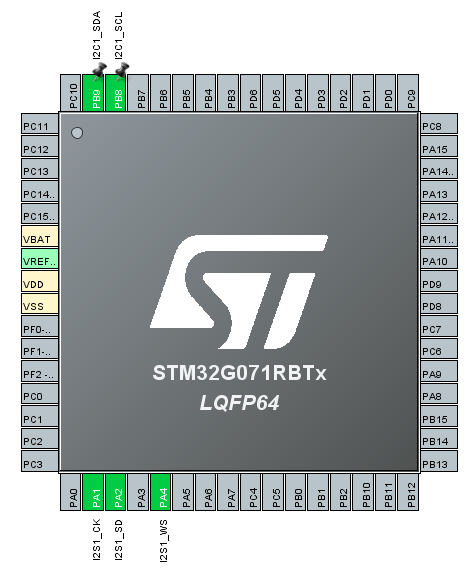
将I2C1配置为I2C模式，按如下配置I2C



按下图所示配置I2S1



进行管脚映射



## 编写设计代码

**添加函数声明变量定义以及主函数设计**

#include "wave\_data.h"

#define WM8960\_ADDRESS 0x1A

#define AUIDO\_START\_ADDRESS 58 /\* 相对于音频文件头大小的偏移量 \*/

static uint16\_t WM8960\_REG\_VAL[56] = //寄存器值

{

0x0097, 0x0097, 0x0000, 0x0000, 0x0000, 0x0008, 0x0000, 0x000A,

0x01C0, 0x0000, 0x00FF, 0x00FF, 0x0000, 0x0000, 0x0000, 0x0000,

0x0000, 0x007B, 0x0100, 0x0032, 0x0000, 0x00C3, 0x00C3, 0x01C0,

0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,

0x0100, 0x0100, 0x0050, 0x0050, 0x0050, 0x0050, 0x0000, 0x0000,

0x0000, 0x0000, 0x0040, 0x0000, 0x0000, 0x0050, 0x0050, 0x0000,

0x0000, 0x0037, 0x004D, 0x0080, 0x0008, 0x0031, 0x0026, 0x00ED

};

uint32\_t WaveDataLength=0;

uint8\_t WM8960\_Write\_Reg(uint8\_t reg, uint16\_t dat);

uint8\_t WM89060\_Init(void);

int main(void)

{

HAL\_Init();

SystemClock\_Config();

MX\_GPIO\_Init();

MX\_I2C1\_Init();

MX\_I2S1\_Init();

MX\_TIM2\_Init();

WM89060\_Init();

WaveDataLength = sizeof(WaveData) - AUIDO\_START\_ADDRESS; //计算音频数据长度 = 总长度 - 开头

while (1)

{

HAL\_I2S\_Transmit(&hi2s1,(uint16\_t\*)(WaveData + AUIDO\_START\_ADDRESS),WaveDataLength,1000);

HAL\_Delay(10);

}

}

**添加子函数**

uint8\_t WM8960\_Write\_Reg(uint8\_t reg, uint16\_t dat)

{

uint8\_t res,I2C\_Data[2];

I2C\_Data[0] = (reg<<1)|((uint8\_t)((dat>>8)&0x0001)); //寄存器地址

I2C\_Data[1] = (uint8\_t)(dat&0x00FF); //寄存器数据

res = HAL\_I2C\_Master\_Transmit(&hi2c1,(WM8960\_ADDRESS<<1),I2C\_Data,2,10);

if(res == HAL\_OK)

WM8960\_REG\_VAL[reg] = dat;

return res;

}

uint8\_t WM89060\_Init(void)

{

uint8\_t res;

//重置设备

res = WM8960\_Write\_Reg(0x0f, 0x0000);

if(res != 0)

return res;

else ; //WM8960复位成功

//设置电源

res = WM8960\_Write\_Reg(0x19, 1<<8 | 1<<7 | 1<<6);

res += WM8960\_Write\_Reg(0x1A, 1<<8 | 1<<7 | 1<<6 | 1<<5 | 1<<4 | 1<<3);

res += WM8960\_Write\_Reg(0x2F, 1<<3 | 1<<2);

if(res != 0) //电源设置失败

return res;

//配置时钟

//MCLK->div1->SYSCLK->DAC/ADC sample Freq = 25MHz(MCLK)/2\*256 = 48.8kHz

WM8960\_Write\_Reg(0x04, 0x0000); //预设

//配置ADC和DAC

WM8960\_Write\_Reg(0x05, 0x0000); //预设

//配置音频接口

//I2S格式16位字长

WM8960\_Write\_Reg(0x07, 0x0002); //√

//Configure HP\_L and HP\_R OUTPUTS

WM8960\_Write\_Reg(0x02, 0x006F | 0x0100); //LOUT1 Volume Set 输出模拟电压6F最大

WM8960\_Write\_Reg(0x03, 0x006F | 0x0100); //ROUT1 Volume Set

//Configure SPK\_RP and SPK\_RN

WM8960\_Write\_Reg(0x28, 0x007F | 0x0100); //Left Speaker Volume 7F最大

WM8960\_Write\_Reg(0x29, 0x007F | 0x0100); //Right Speaker Volume 7F

//输出使能

WM8960\_Write\_Reg(0x31, 0x00F7); //左右声道都打开

//配置DAC的值

WM8960\_Write\_Reg(0x0a, 0x00FF | 0x0100); //FF无衰减

WM8960\_Write\_Reg(0x0b, 0x00FF | 0x0100);

//3D 音效控制

//WM8960\_Write\_Reg(0x10, 0x001F);

//配置 MIXER

WM8960\_Write\_Reg(0x22, 1<<8 | 1<<7);

WM8960\_Write\_Reg(0x25, 1<<8 | 1<<7);

//Jack Detect

WM8960\_Write\_Reg(0x18, 1<<6 | 0<<5);

WM8960\_Write\_Reg(0x17, 0x01C3);

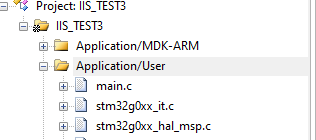
WM8960\_Write\_Reg(0x30, 0x0009);//0x000D,0x0005

return 0;

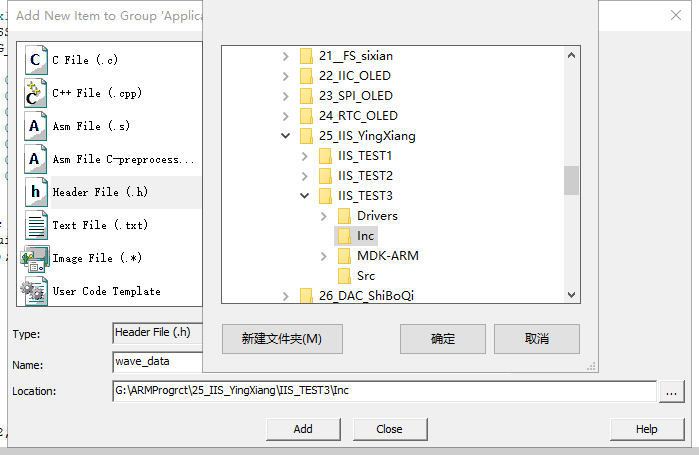
}

**添加数据文件**

鼠标右键点击下图灰色位置，选择添加新的项目

****

按下图所示创建新文件。注：文件保存位置必须是Inc文件夹。



添加成功后在此文件夹下写入数据，可用GoldWave（音频抽样）、UltraEdit（范围选择复制）与记事本（批量替换为0x格式）产生音频数据。

#ifndef \_\_WAVEDATE\_H

#define \_\_WAVEDATE\_H

#include "stm32g0xx\_hal.h"

const uint16\_t WaveData[] = {

0x99,0x02,0x68,0x05,0x07,0x05,0xCC,0x05,0x1D,0x03,0xEA,0x02,0xC6,0x02,0x50,0xFE

,0x95,0xFA,0x9B,0x03,0x3B,0x00,0x80,0xFA,0x01,0x01,0x62,0xFB,0x35,0xFB,0x98,0xF9

,0xBA,0xFE,0xA5,0xF8,0x92,0xF5,0x42,0xF9,0xF4,0xF8,0x58,0xFF,0x3D,0x01,0x89,0xFB

,0x10,0xFC,0x63,0xFA,0x52,0xFC,0xB6,0xF6,0xA1,0xFC,0x05,0xF8,0xB6,0xF8,0x0B,0xFA

,0xC0,0xFD,0xF9,0xF3,0xA8,0xF4……} //此处是大量音频数据

#endif

## 结果

编译下载成功后可听到声音。

## 总结

声音信号有待改善

1声音采样原因：GoldWave对音频信号进行截取与重采样，另存为一文件。UltraEdit打开新保存的文件，按照16进制复制，在UltraEdit创建新的文件并粘贴，然后选区域复制（不带其他没用字符）。最后打开记事本粘贴，用替换功能将所有“空格”替换成为“,0x”稍作修改就可导入音频数据矩阵。

2设置：语音模块设置、STM32的IIS模块设置以及用软件采样过程都得配合起来。采样频率、位数等等。