SYN6658中文语音合成芯片
附加手册（SPI通讯）

北京宇音天下科技有限公司

010-62986600 010-62969399 www.voicetx.com

宇音天下官方订阅号

宇音天下售前咨询

重要声明

版权声明

商标声明

北京宇音天下科技有限公司的产品是北京宇音天下科技有限公司专有。在提及其他公司及其产品时将使用各自公司所拥有的商标，这种使用的目的仅限于引用。本文档可能涉及北京宇音天下科技有限公司的专利（或正在申请的专利）、商标、版权或其他知识产权，除非得到北京宇音天下科技有限公司的明确书面许可协议，本文档不授予使用这些专利（或正在申请的专利）、商标、版权或其他知识产权的任何许可协议。

不作保证声明

北京宇音天下科技有限公司不对此文档中的任何内容作任何明示或暗示的陈述或保证，而且不对特定目的的适销性及适用性或者任何间接、特殊或连带的损失承担任何责任。本手册内容若有变动，恕不另行通知。本手册例子中所用的公司、人名和数据若非特别声明，均属虚构。未得到北京宇音天下科技有限公司明确的书面许可，不得为任何目的、以任何形式或手段（电子的或机械的）复制或传播手册的任何部分。

保密声明

本文档（包括任何附件）包含的信息是保密信息。接收人了解其获得的本文档是保密的，除用于规定的目的外不得用于任何目的，也不得将本文档泄露给任何第三方。

本软件产品受最终用户许可协议（EULA）中所述条款和条件的约束，该协议位于产品文档和/或软件产品的联机文档中，使用本产品，表明您已阅读并接受了EULA的条款。

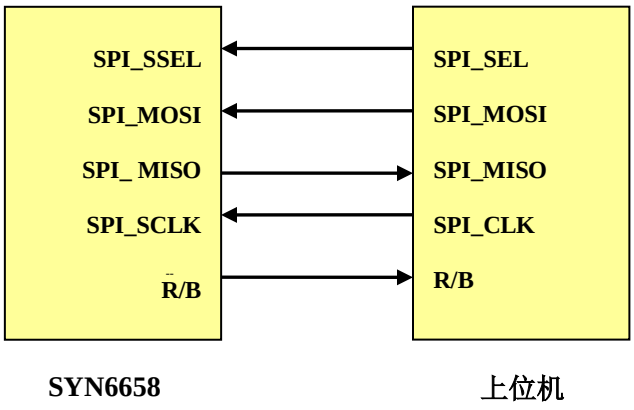
1 SPI通讯概述

- 《SYN6658中文语音合成芯片用户手册.pdf》为主手册，此附加手册仅针对SYN6658芯片的SPI通讯部分进行详尽描述。
- 主手册《SYN6658中文语音合成芯片用户手册.pdf》中SPI通讯部分的描述与此附加手册有不同的，以此手册为准。

2 SPI通讯模式

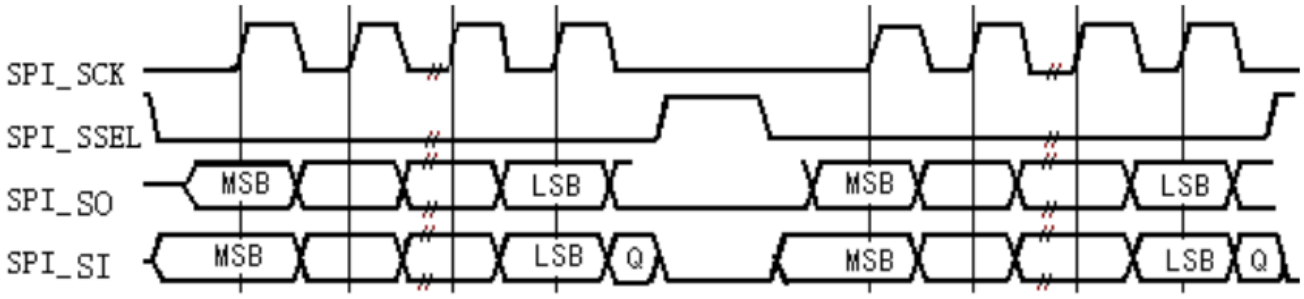
2.1 硬件连接

SYN6658 芯片的 SPI 接口是 4 线全双工同步串行通讯接口，上位机作为 SPI 通讯中的 Master 身份，SYN6658 芯片在 SPI 通讯中设为 Slave 身份， SPI 通讯所需的时钟信号由上位机提供。



2.2 通讯传输字节格式

SPI 通讯属于同步串行通讯，上位机在和 SYN6658 的通讯中，由上位机提供同步时钟信号，在同步时钟的上升沿 SYN6658 锁存 1bit 数据，每传输 8bits 数据完成一个字节数据的传输。



SPI 通讯时序图

2.3 通讯要求

- 在发送一个字节之前，将 SPI_SSEL 置为低电平，发送一个字节后将 SPI_SSEL 置为高电平。
- 每两个字节之间必须插入一段等待,时间不少于 50us。
- 同一帧数据 2 个字节之间的延时不能超过 20ms，超过认为是下一帧数据的开始。
- 建议用户发完一个命令帧之后延时 200ms 以上再发下一个命令帧。
- 硬件电路必须将芯片波特率配置端口的两个管脚BAUD0、BAUD1，配置成低电平；（此配置与SPI实际通讯速率无关）；
- 上位机所采用的 CLK 频率不能超过 22000Hz。

2.4 芯片回传

SYN6658 芯片收到命令帧后会判断此命令帧正确与否。如果命令帧正确，则返回“接收成功”回传；如果命令帧错误，则返回“接收失败” 回传。

回传类型名称	回传数据	触发条件
初始化成功回传	无	SPI通讯不支持
收到正确的命令帧回传	0x41	接收成功，收到正确的命令帧
收到不能识别命令帧回传	0x45	接收失败，收到错误的命令帧
芯片播音状态回传	无	SPI通讯不支持“状态查询命令”
芯片空闲状态回传	无	SPI通讯不支持“状态查询命令”

2.5 命令支持

- 不支持：软件查询命令（若正在播音，强行使用会停掉播音），请使用硬件查询芯片状态（高电平：播音状态；低电平：空闲状态）；

- 不支持： 0x31文本缓存存储命令 和 0x32文本缓存播放命令，这两个命令为特殊应用；
- 在播音发声状态下，直接发送恢复命令或唤醒命令，会停止播音；

2.6 芯片各状态下的功耗参数

正常工作状态下		Standby
合成文本状态	空闲	
53mA	27mA	400μA

3 SPI通讯如何发送命令帧

说明：

- 为确保SPI通讯的稳定性,用户在发送命令帧时必须调用本手册提供的封装函数《SYN6658SpiSendFrame》，请用户将本封装函数《SYN6658SpiSendFrame》嵌入自己的程序中去。见本章第1节。
- 本封装函数《SYN6658SpiSendFrame》中调用的数据类型和3个子函数，客户根据自己的硬件情况自己配置和实现。见本章第2节。
- 本手册给出的用户发送命令帧的调用举例。见本章第3节。

3.1 发送命令帧的封装函数

```
/*********************************************************************
 * 功能   :  发送 1 个命令帧数据
 * 参数 1 :  命令帧的地址
 * 参数 2 :  命令帧的总长度（按字节计数）
 * 返回值:  0x41：表示下位机接收成功
 *          0x45：表示失败（通讯异常或准备命令帧有错）
 *********************************************************************/
T_UInt8   SYN6658SpiSendFrame(T_UInt8 *buffOfFrame, T_UInt16 nFrameLen)
{
    T_UInt8   nReceiveData;
    T_UInt8   nCircleTimes;
    T_UInt16 i;
    T_UInt8   nBuffStopAddEnd[5]={0xFD,0x00,0x01,0x02,0xFF};

    //循环确保 SYN6658 芯片为 Ready 状态（即 R/B 引脚为低电平）
    nCircleTimes = 0;
    while( SYN6658SpiCheckBusy() == 1 )
    {
        SYN6658SpiSendOneByte(0xA5);//发送暂停同步字节 0xA5，
        SYN6658SpiDelayMs(5);
        nCircleTimes++;
        if( nCircleTimes > 10 )              // 1 次成功率在 98%以上，多次确保成功
            return 0x45;                     //失败：通讯异常
    }

    //若为：合成命令或 Standby 命令
    if((*(buffOfFrame+3)==0x01 )||(*(buffOfFrame+3)==0x22 || *(buffOfFrame+3)==0x88))
    {
        //循环确保播音停止
        nCircleTimes = 0;
        do
        {
            if( nCircleTimes != 0)
                SYN6658SpiDelayMs(200);

            for(i=0;i<5;i++)                 //发送 Stop 命令(含结束字节)
                nReceiveData = SYN6658SpiSendOneByte(*(nBuffStopAddEnd+i));

            nCircleTimes++;
            if( nCircleTimes >= 4 )          // 1 次成功率在 99.5%以上，多次确保成功
                break;                       //退出
        }while(nReceiveData != 0x41) ;
        SYN6658SpiDelayMs(200);
    }
```

```
    //循环确保下位机成功接收命令帧
    nCircleTimes = 0;
    do
    {
        if( nCircleTimes != 0)
            SYN6658SpiDelayMs(200);

        //发送本命令帧
        for(i=0;i<nFrameLen;i++)
        {
            nReceiveData = SYN6658SpiSendOneByte(*(buffOfFrame+i));
            if( nReceiveData == 0x41 || nReceiveData == 0x45)                    //
                break;
        }

        //发送帧结束字节 0xFF，并得到下位机返回的值
        if (nReceiveData != 0x41 && nReceiveData != 0x45 )
            nReceiveData = SYN6658SpiSendOneByte(0xFF);

        nCircleTimes++;
        if( nCircleTimes >= 4 )                  // 1 次成功率在 99.5%以上，多次确保成功
            return 0x45;                         //失败：通讯异常或准备命令帧有错
    }while(nReceiveData != 0x41) ;

    return 0x41;                                 //接收成功
}
```

3.2 用户需配置和实现

```
//注意：以下数据类型和 3 个子函数，客户根据自己的硬件情况自己配置和实现

#define   T_UInt8       unsigned char              //无符号单字节
#define   T_UInt16      unsigned int               //无符号 2 字节整型
#define   T_VOID        void


/*****************************************************************
 * 功能   : 延时参数中给定的毫秒数
 * 参数 1 : 需延时的毫秒数
 *****************************************************************/
 T_VOID SYN6658SpiDelayMs(T_UInt16 nMsCount)
{
    //用户实现
}

 /*****************************************************************
 * 功能   : 检测状态 R/B 引脚的电平值确定芯片忙闲状态
 * 返回值: 1：R/B 引脚为高电平，表示 SYN6658 芯片为 Busy
 *         0：R/B 引脚为低电平，表示 SYN6658 芯片为 Ready
 *****************************************************************/
T_UInt8 SYN6658SpiCheckBusy(T_VOID)
{
    //用户实现
}

/*****************************************************************
 * 功能   : 传输 1 个字节
 * 参数 1 : 发送的本字节值
 * 返回值: 接收的字节值
 *****************************************************************/
T_UInt8 SYN6658SpiSendOneByte( T_UInt8 data1)
{
    //用户实现
}
```

3.3 封装函数调用举例

```
main()
{
    INT8U nReturnByte;

    //命令帧：合成“欢迎观看语音合成系统的演示”
    T_UInt8 buffSyn[31] = {0xFD,0x00,0x1C,0x01,0x01,0xBB,0xB6,0xD3,0xAD,0xB9,0xDB,0xBF,
                           0xB4,0xD3,0xEF,0xD2,0xF4,0xBA,0xCF,0xB3,0xC9,0xCF,0xB5,0xCD,
                           0xB3,0xB5,0xC4,0xD1,0xDD,0xCA,0xBE};
    T_UInt8 buffStop[4]      = {0xFD,0x00,0x01,0x02};          //命令帧：停止
    T_UInt8 buffPause[4]     = {0xFD,0x00,0x01,0x03};          //命令帧：暂停
    T_UInt8 buffResume[4]    = {0xFD,0x00,0x01,0x04};          //命令帧：恢复
    T_UInt8 buffStandby[4]   = {0xFD,0x00,0x01,0x22};          //命令帧：Standby
    T_UInt8 buffWake[4]      = {0xFD,0x00,0x01,0xFF};          //命令帧：唤醒

    ......

    //合成“欢迎观看语音合成系统的演示”，并等待播音完毕（即 R/B 引脚为低电平）
    nReturnByte = SYN6658SpiSendFrame(buffSyn, 31);
    SYN6658SpiDelayMs(10);
    while(true)
    {
        if( SYN6658SpiCheckBusy() == 0 )
            break;
    }

    //合成“欢迎观看语音合成系统的演示”，并在播音过程中，执行“暂停”“恢复”“停止”命令
    nReturnByte = SYN6658SpiSendFrame(buffSyn, 31);
    SYN6658SpiDelayMs(1500);
    nReturnByte = SYN6658SpiSendFrame(buffPause, 4);
    SYN6658SpiDelayMs(1500);
    nReturnByte = SYN6658SpiSendFrame(buffResume, 4);
    SYN6658SpiDelayMs(1500);
    nReturnByte = SYN6658SpiSendFrame(buffStop, 4);
    SYN6658SpiDelayMs(1500);

    //合成“欢迎观看语音合成系统的演示”，并在播音过程中，执行“Standy”命令；然后再“唤醒”；然后再合成
    nReturnByte = SYN6658SpiSendFrame(buffSyn, 31);
    SYN6658SpiDelayMs(1500);
    nReturnByte = SYN6658SpiSendFrame(buffStandby, 4);
    SYN6658SpiDelayMs(1500);
    nReturnByte = SYN6658SpiSendFrame(buffWake, 4);
    SYN6658SpiDelayMs(1500);
    nReturnByte = SYN6658SpiSendFrame(buffSyn, 4);
    SYN6658SpiDelayMs(1500);
    ......
}
```