



YCH080 语音芯片使用资料 V1.4

目 录

1、产品特点.....	3
2、芯片选型.....	3
3、应用范围.....	3
4、管脚图.....	3
4.1、YCH080-8S/P.....	3
5、电气参数.....	4
6、控制模式.....	4
6.1、按键控制.....	4
6.2、一线串口控制.....	6
6.3、两线串口控制.....	7
6.3.1、发送地址命令.....	7
6.3.2、读取芯片工作状态.....	7
6.3.3、控制命令.....	7
6.3.4、语音地址.....	8
7、复位及输出状态.....	8
7.1、复位.....	8
7.2、输出状态.....	8
8、应用电路.....	9
8.1、按键控制模式应用电路.....	9
8.1.1、YCH080-8S/P 按键控制（DC3V 供电，PWM 输出）.....	9
8.1.2、YCH080-8S/P 按键模式（DC3V 供电，DAC 输出外接三极管）.....	9
8.1.3、YCH080-8S/P 按键模式（DC3V 供电，DAC 输出外接功放）.....	9
8.1.4、YCH080-8S/P 按键模式（DC5V 供电，PWM 输出）.....	9
8.1.5、YCH080-8S/P 按键模式（DC5V 供电，DAC 输出外接三极管）.....	10
8.1.6、YCH080-8S/P 按键模式（DC5V 供电，DAC 输出外接功放）.....	10
8.2、一线串口控制模式应用电路.....	10
8.2.1、YCH080-8S/P 一线串口模式（MCU 为 DC3V 供电，PWM 输出）.....	10
8.2.2、YCH080-8S/P 一线串口模式（MCU 为 DC3V 供电，DAC 输出外接三极管）.....	11
8.2.3、YCH080-8S/P 一线串口模式（MCU 为 DC3V 供电，DAC 输出外接功放）.....	11
8.2.4、YCH080-8S/P 一线串口模式（MCU 为 DC5V 供电，PWM 输出）.....	11
8.2.5、YCH080-8S/P 一线串口模式（MCU 为 DC5V 供电，DAC 输出外接三极管）.....	11
8.2.6、YCH080-8S/P 一线串口模式（MCU 为 DC5V 供电，DAC 输出外接功放）.....	12



8.3、两线串口控制模式应用电路.....	12
8.3.1、YCH080-8S/P 两线串口模式（MCU 为 DC3V 供电，PWM 输出）	12
8.3.2、YCH080-8S/P 两线串口模式（MCU 为 DC3V 供电，DAC 输出外接三极管）	12
8.3.3、YCH080-8S/P 两线串口模式（MCU 为 DC3V 供电，DAC 输出外接功放）	12
8.3.4、YCH080-8S/P 两线串口模式（MCU 为 DC5V 供电，PWM 输出）	13
8.3.5、YCH080-8S/P 两线串口模式（MCU 为 DC5V 供电，DAC 输出外接三极管）	13
8.3.6、YCH080-8S/P 两线串口模式（MCU 为 DC5V 供电，DAC 输出外接功放）	13
9、程序范例.....	15
9.1、一线串口控制程序范例.....	15
9.2、两线串口控制程序范例.....	17
10、封装尺寸图.....	23
10.1、YCH080-8S 封装尺寸图.....	23
11、历史版本记录.....	24



1、产品特点

- 可编程一次性烧录（OTP）语音芯片；
- 语音长度可达80秒（6KHz采样率）；
- PWM和DAC两种音频输出方式；
- 内部集成时钟振荡器；
- 最大可存放700个语音文件；
- 最多可加载62段地址的语音；
- 具有按键控制模式、一线串口控制模式以及两线串口控制模式等；
- 触发防抖时间：50us（串口）和10ms（按键）；
- 7种按键触发方式；
- 支持播放不同采样率的语音文件；
- 支持BUSY状态输出功能；
- 支持USB端口下载；
- 工作电压：DC2.4~5.0V。

2、芯片选型

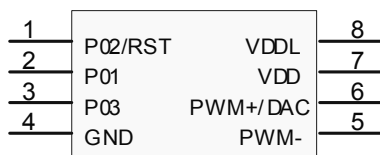
芯片型号	封装形式	语音长度	BUSY 输出	控制端口	输出端口
YCH080-8S	SOP8	80S	支持	3	0

3、应用范围

- 汽车（防盗报警器、倒车雷达、GPS 导航仪、电子狗、中控锁）；
- 智能家居系统；
- 家庭防盗报警器；
- 医疗器械人声提示；
- 家电（电磁炉、电饭煲、微波炉）；
- 娱乐设备（游戏机、游乐机）；
- 学习模型（早教机、儿童有声读物）；
- 智能交通设备（收费站、停车场）；
- 通信设备（电话交换机、电话机）；
- 工业控制领域（电梯、工业设备）；
- 高举玩具。

4、管脚图

4.1、YCH080-8S/P



封装引脚	引脚标号	简述	功能描述
1	P02/RST	KEY2/RST/SCK	按键 2/复位按键/两线串口时钟输入
2	P01	KEY1/DI/DATA	按键 1/两线串口 DATA IN/一线串口 DATA
3	P03	KEY3/DO	按键 3/两线串口 DATA OUT
4	GND	GND	地线
5	PWM-	PWM-	PWM-音频输出
6	PWM+/DAC	PWM+/DAC	PWM+音频输出/DAC 音频输出
7	VDD	VDD	电源输入端，供电电压 2.4~5V，需要接 104 电容到地线
8	VDDL	VDDL	电源输入端，供电电压 2.4~3.6V

5、电气参数

环境温度 25℃，工作电压 DC3V

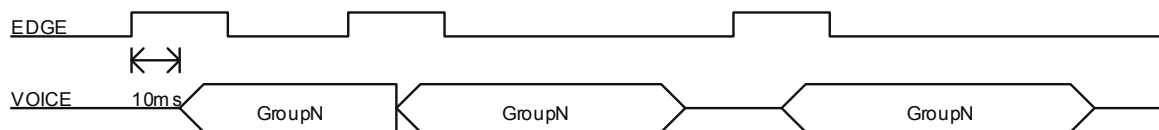
参数	标记	环境条件	最小值	典型值	最大值	单位
工作电压	V _{DD}	无	2.4	3	5.0	V
待机电流 (LDO ON)	ISB	VDD=3V，没负载	3	4	5	uA
待机电流 (LDO OFF)	ISB	VDD=3V，没负载	1	1.5	3	uA
工作电流	IOP	VDD=3V，没负载		600		uA
输出驱动电流	IOD	VDD=3V，VOUT=0.6V		10		mA
输出灌电流	IOS	VDD=3V，VOUT=2.4V		30		mA
PWM 驱动电流	IOD	VDD=3V，VOUT=1.5V		200		mA
PWM 灌电流	IOS	VDD=3V，VOUT=1.5V		200		mA

6、控制模式

6.1、按键控制

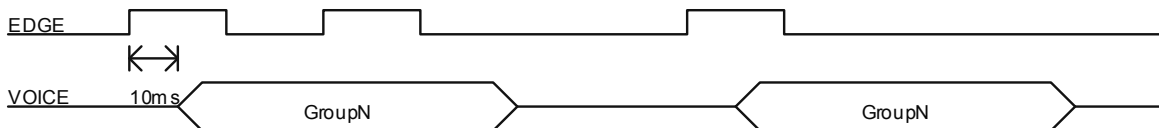
在按键控制模式下，任意控制端均可设置为脉冲可重复、脉冲不可重复、脉冲保持、脉冲不保持、电平保持可循环、电平保持不可循环、下一曲可循环等 7 种触发方式。

脉冲可重复



收到正脉冲信号后开始播放语音，在语音结束前如果还收到第二次正脉冲信号，则重新开始播放语音，在语音播放的过程中无收到正脉冲信号则播放完整段语音。

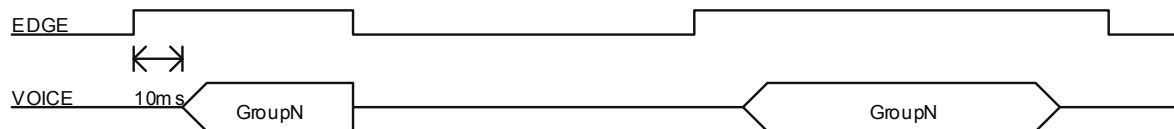
脉冲不可重复



收到正脉冲信号后开始播放语音，在语音结束前如果再次收到正脉冲信号，则没有动作产生，在语音播放结束后再次收到正脉冲

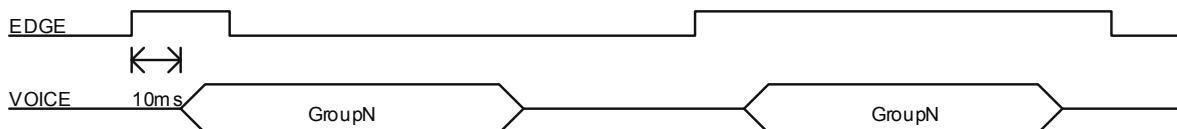
信号才会重新播放语音。

脉冲保持



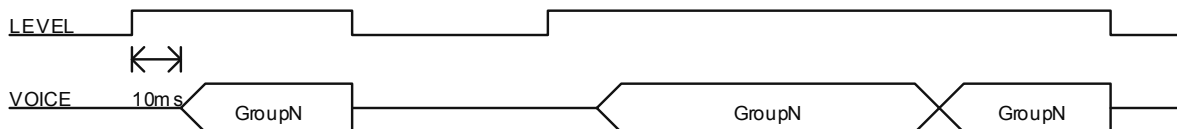
收到正脉冲信号后开始播放语音，正脉冲信号停止时，语音也停止播放。如果正脉冲信号一直保持，并且持续的时间超出语音播放的长度，则播放完语音即停止，不再继续播放语音。

脉冲不保持



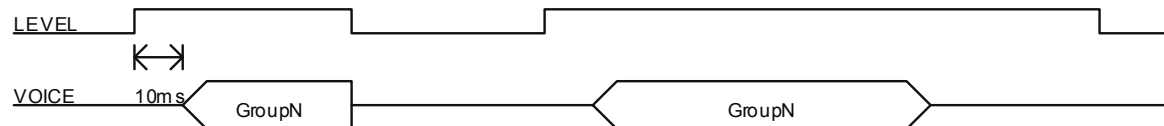
收到正脉冲信号后开始播放语音，正脉冲信号停止后语音继续播放完毕，如果正脉冲信号一直保持，并且持续的时间超出语音播放的长度，则播放完语音即停止，不再继续播放语音。

电平保持可循环



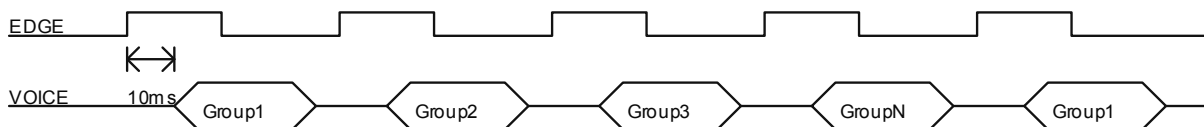
收到高电平后播放语音，高电平信号停止时，语音也停止播放。如果一直保持高电平信号，则会一直循环播放当前段语音语音。

电平保持不可循环



收到高电平后播放语音，高电平信号停止时，语音也停止播放。即时一直保持高电平信号，在语音播放完毕后也不再有任何动作。

下一曲可循环

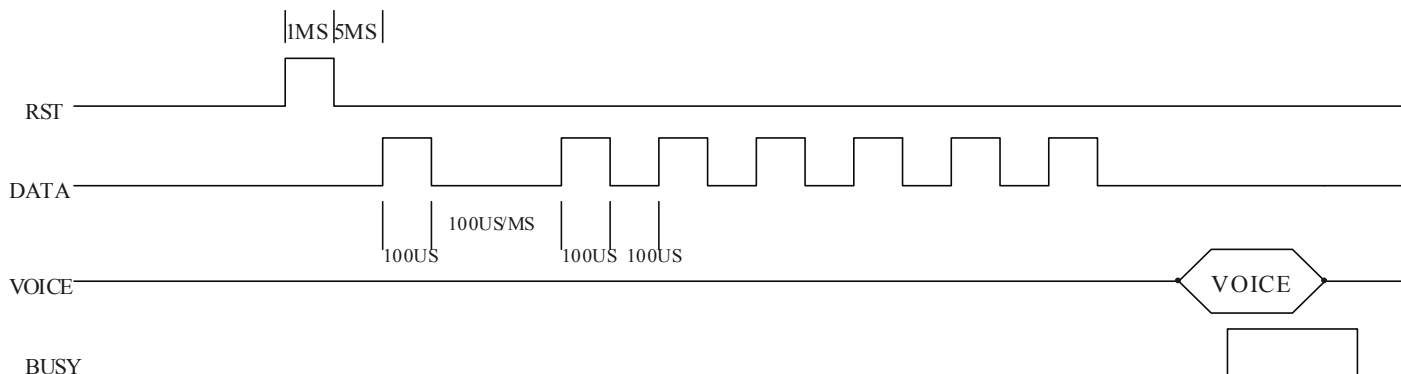


此触发方式仅限于在 P00 控制端口。收到正脉冲信号后开始播放第一段地址语音，再次收到正脉冲信号则播放第二段地址语音，当触发到最后一段后，再次触发则重新开始播放第一段语音，如此循环。

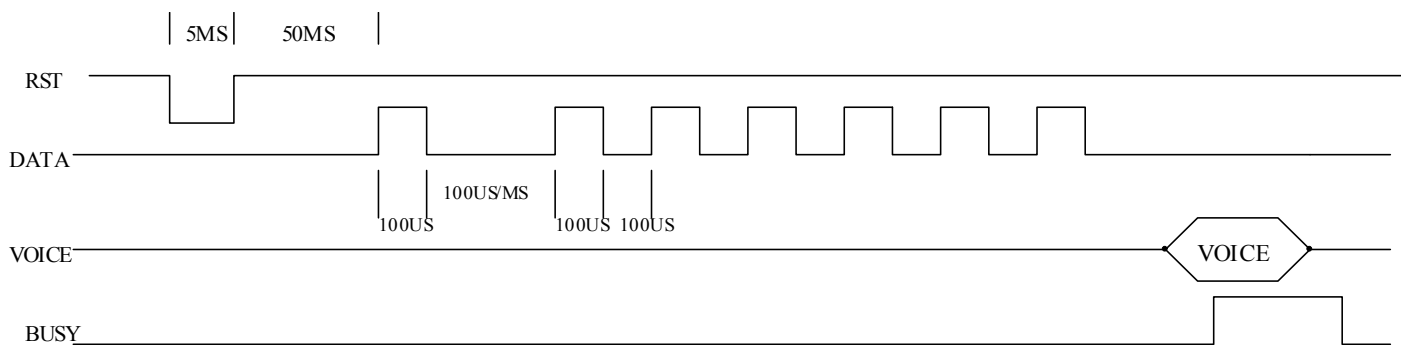
6.2、一线串口控制

一线串口控制模式是通过在 DATA 线上发送不同的脉冲数量以达到控制语音地址的目的，该控制模式具有控制端口少，可控制语音地址数量多等优点。常用于 MCU 控制端口紧缺的场合。但一线串口控制模式也容易受到外界脉冲信号的干扰，故需要谨慎使用。

一线串口控制时序中，有分硬件复位和软件复位两种，其中软件复位是先发送 200us 的 RESET 信号，等待 5ms 后发送 DATA，DATA 中每个脉冲保持在 1MS 的高电平，两个脉冲之间的间隔时间需要 100us。发送脉冲后等待 200us，开始播放地址语音，再过 200us 后 BUSY 信号发生变化。其时序如下图所示。**注：下面的第一个脉冲的低电平有 100US 和 100MS 之分，是 PWM 输出时选用 100US，DAC 输出时，选用 100MS。**



下图为硬件复位时的时序图，其中硬件复位是先发送 5MS 的/RESET 信号，等待 50ms 后发送 DATA，DATA 中每个脉冲保持在 100us 的高电平，两个脉冲之间的间隔时间需要 100us。发送脉冲后等待 200us，开始播放地址语音，再过 200us 后 BUSY 信号发生变化。其时序如下图所示。**注：下面的第一个脉冲的低电平有 100US 和 100MS 之分，是 PWM 输出时选用 100US，DAC 输出时，选用 100MS。**



脉冲数量以及所触发语音地址的对应关系如下表所示

序号	脉冲数	语音地址
1	1	0
2	2	1
3	3	2
4
5	62	61

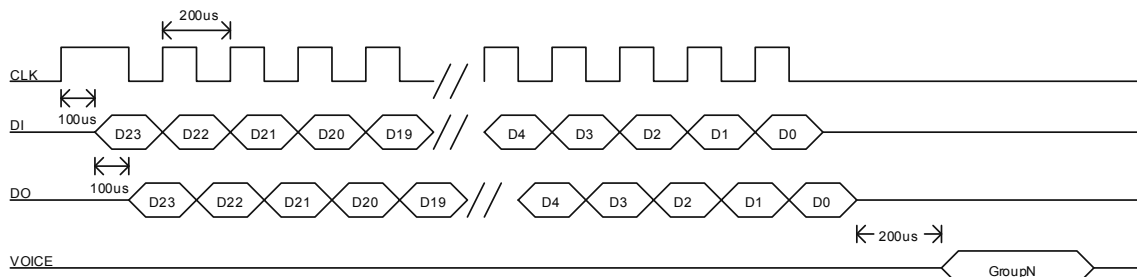


6.3、两线串口控制

6.3.1、发送地址命令

两线串口控制模式由 DI 及 CLK 发送数据信号控制语音地址，同时由 DO 返回相关操作数据。在发送 DI 前先将 CLK 拉高 100us 以唤醒芯片工作，继而发送周期为 200us 的 CLK 信号跟 DI，在接收到 DI 数据 100us 后 DO 返回 DI 所发送的数据。整个数据需要发送 24bit，数据发送完成后 200us，开始播放地址语音。

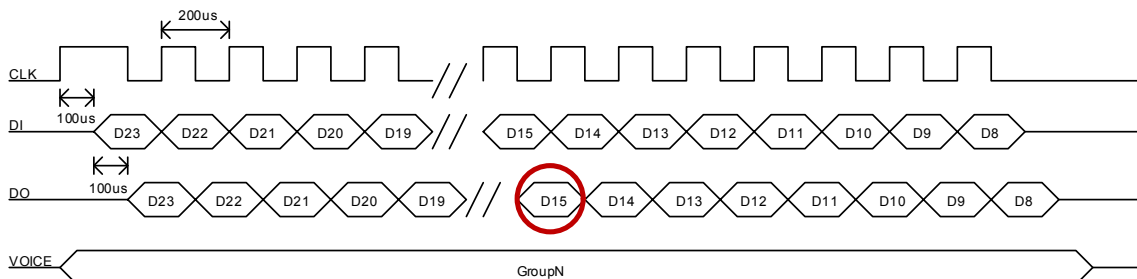
时钟周期的工作范围为 50~4000us。



/*注：二线时，若 CLK 脚上在上电时有高脉冲产生，会进入烧写模式，因此注意管脚电平，若是 51 内核单片机时，最好在 CLK 脚上并 1K-10K 电阻，在并 1 个 103 电容。*/

6.3.2、读取芯片工作状态

发送 16bit 的数据 0x2200，可以通过读取 DO 返回的数据，知道芯片的当前状态。由 DO 返回的 D15 数据，0 代表语音停止，1 代表芯片正在播放语音。





6.3.3、控制命令

命令	管脚	数据	时钟位	说明
初始化	DI	0×0A0140	24	每次上电后需要发送此命令，以唤醒芯片。
	DO			
播放	DI	0×18+语音地址	24	播放地址语音
	DO			
暂停	DI	0×0A0148	24	可以暂停正在播放的语音
	DO			
从暂停处播放	DI	0×0A0140	24	从暂停处恢复播放语音
	DO			
读取芯片状态	DI	0×2200	16	检验芯片是否处于停止状态，0 为停止，1 为播放
	DO	检验 D15		
休眠指令	DI	0×0A0100	24	直接进入休眠，可以打断正在播放的语音
	DO			

6.3.4、语音地址

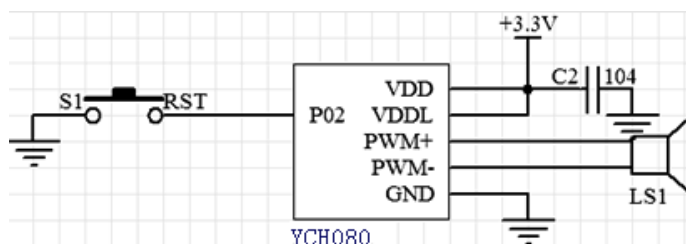
YCH080 在两线串口模式下最多可加载 256 段语音，其控制命令及语音地址对应关系如下。

序号	命令	语音地址
1	0×1800C8	0
2	0×1800E8	1
3	0×180108	2
4	0×180128	3
5	0×180148	4
6	0×180168	5
7
8	0×182BE8	251
9	0×182C08	252
10	0×182C28	253
11	0×182C48	254
12	0×182C68	255

7、复位及输出状态

7.1、复位

可通过 PC 软件设置 P02 为复位控制端口，负脉冲触发，保持 5ms 以上有效。





7.2、输出状态

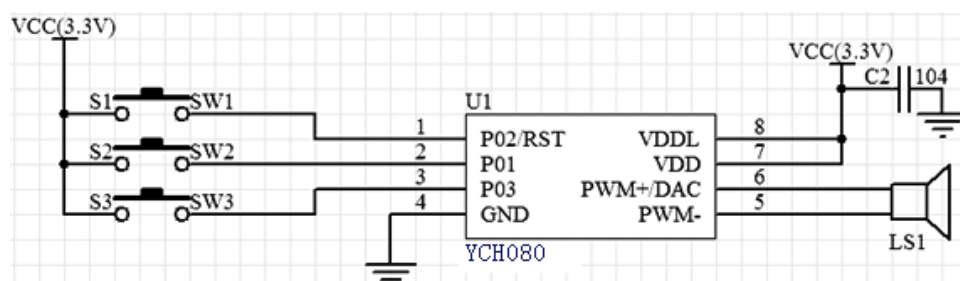
P05、P06、P07、P08、P10、P11、P12 均可以设置为输出端口，其中可设置的状态如下

- 待机状态高电平/低电平
- 播放语音时输出低电平
- 播放语音时输出高电平
- LED 闪光频率 6Hz
- LED 闪光频率 3Hz
- LED 闪光频率 1.5Hz
- LED 闪光频率 0.75Hz

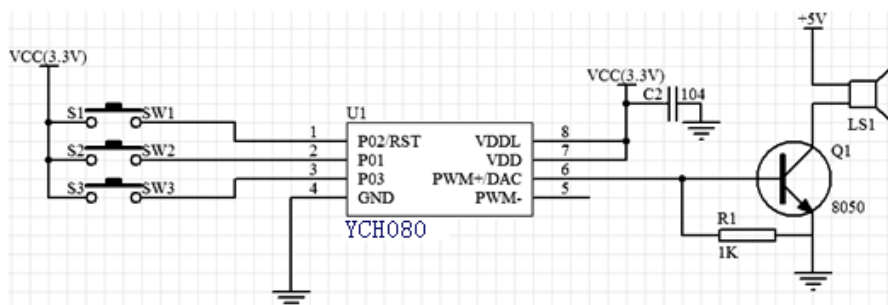
8、应用电路

8.1、按键控制模式应用电路

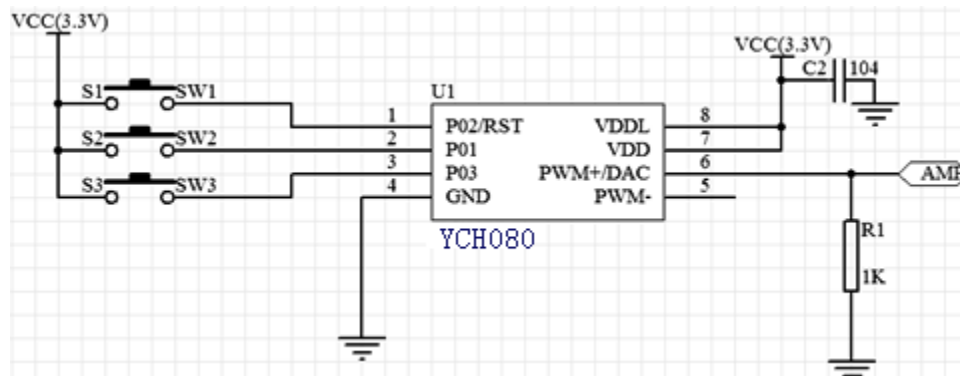
8.1.1、YCH080-8S/P 按键控制（DC3V 供电，PWM 输出）



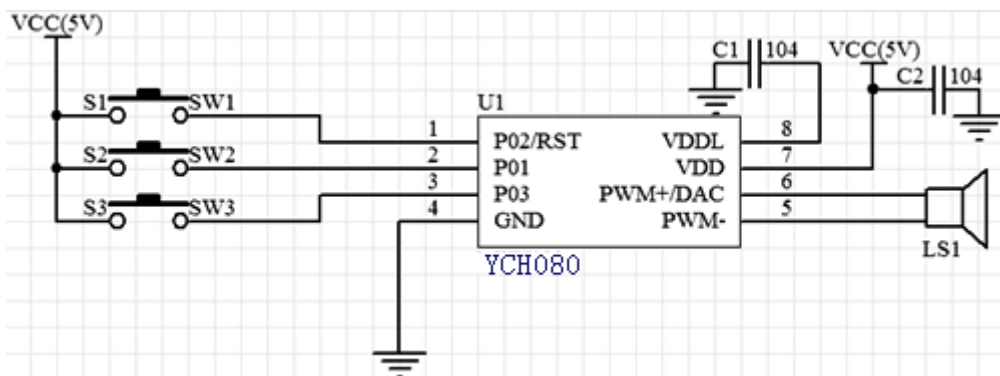
8.1.2、YCH080-8S/P 按键模式（DC3V 供电，DAC 输出外接三极管）



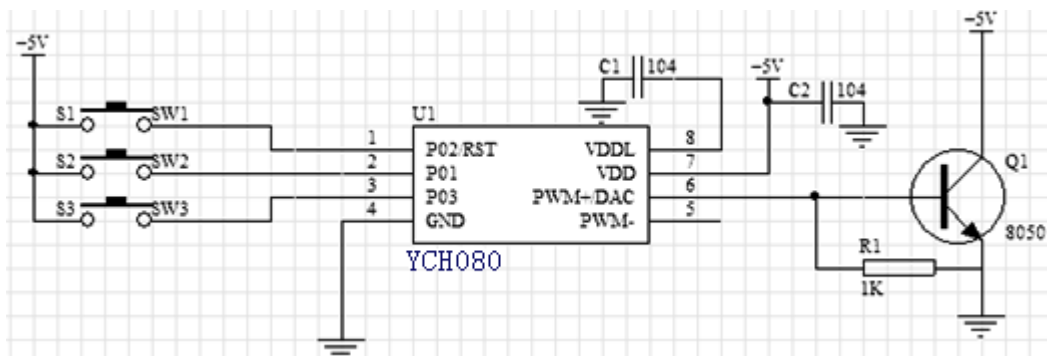
8.1.3、YCH080-8S/P 按键模式（DC3V 供电，DAC 输出外接功放）



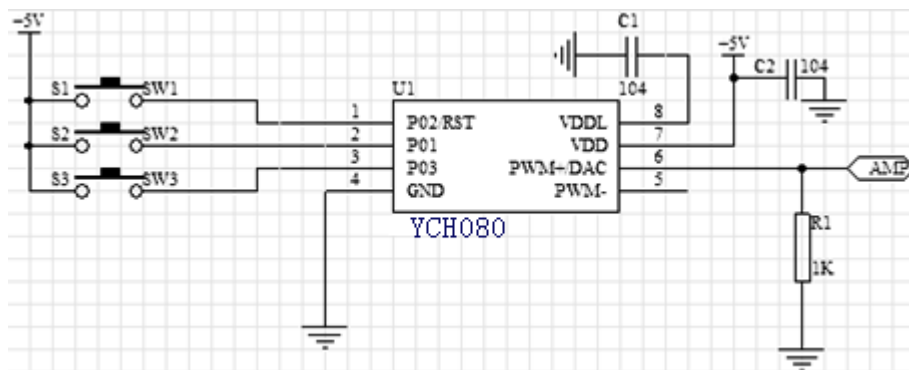
8.1.4、YCH080-8S/P 按键模式（DC5V 供电，PWM 输出）



8.1.5、YCH080-8S/P 按键模式（DC5V 供电，DAC 输出外接三极管）

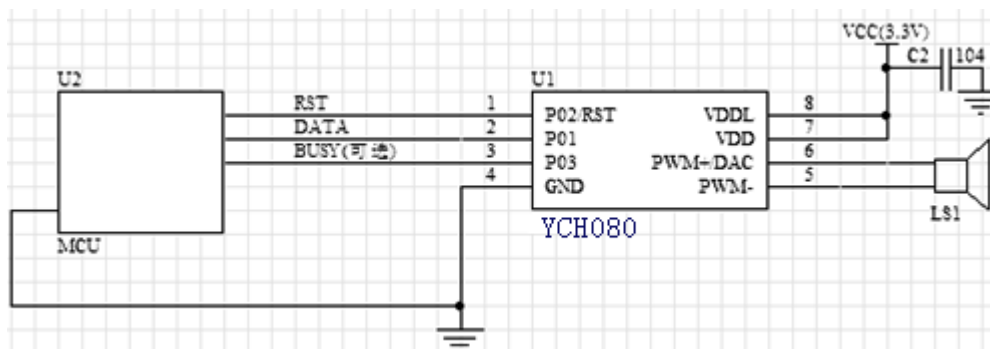


8.1.6、YCH080-8S/P 按键模式（DC5V 供电，DAC 输出外接功放）

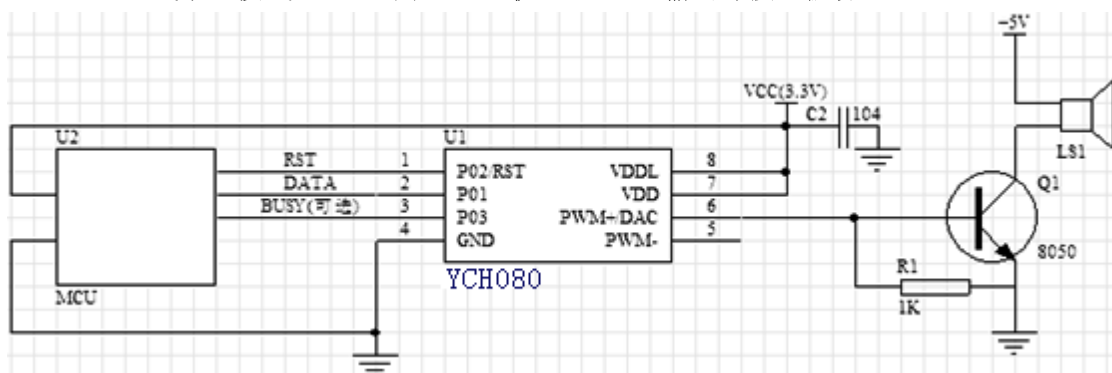


8.2、一线串口控制模式应用电路

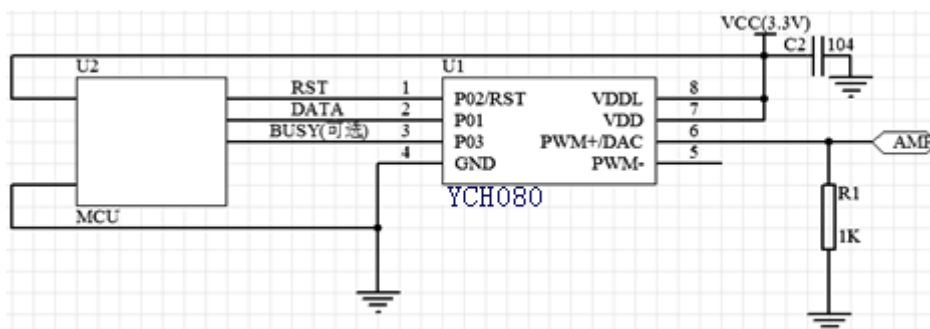
8.2.1、YCH080-8S/P 一线串口模式（MCU 为 DC3V 供电，PWM 输出）



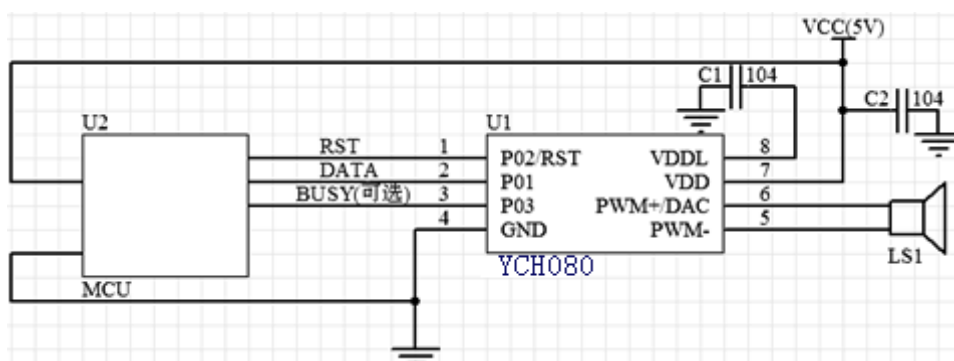
8.2.2、YCH080-8S/P 一线串口模式（MCU 为 DC3V 供电，DAC 输出外接三极管）



8.2.3、YCH080-8S/P 一线串口模式（MCU 为 DC3V 供电，DAC 输出外接功放）

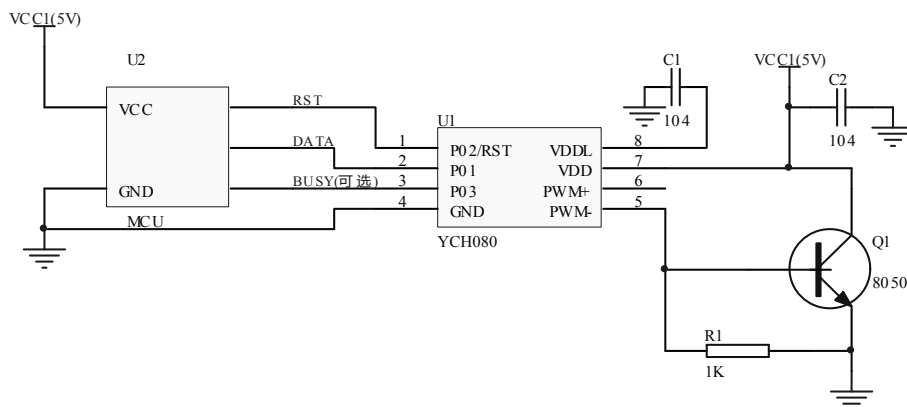


8.2.4、YCH080-8S/P 一线串口模式（MCU 为 DC5V 供电，PWM 输出）

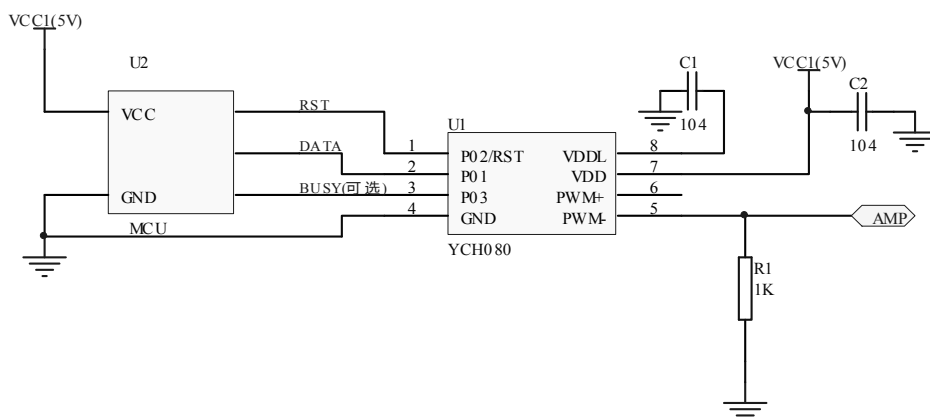




8.2.5、YCH080-8S/P 一线串口模式（MCU 为 DC5V 供电，DAC 输出外接三极管）

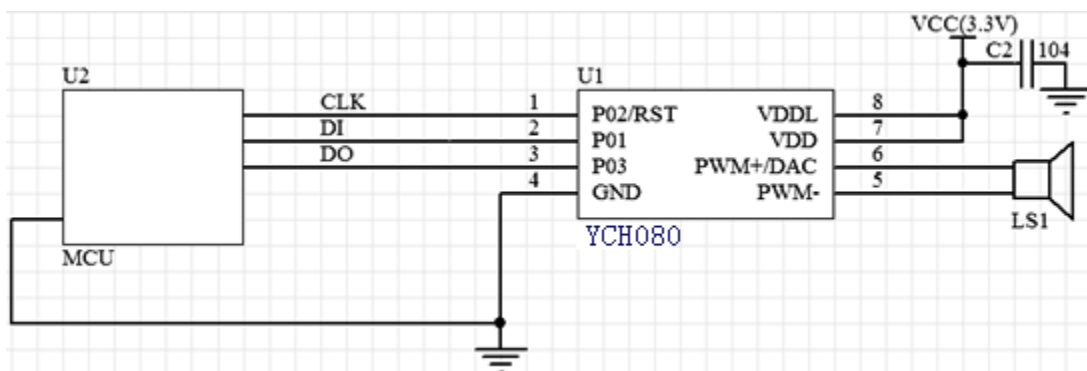


8.2.6、YCH080-8S/P 一线串口模式（MCU 为 DC5V 供电，DAC 输出外接功放）

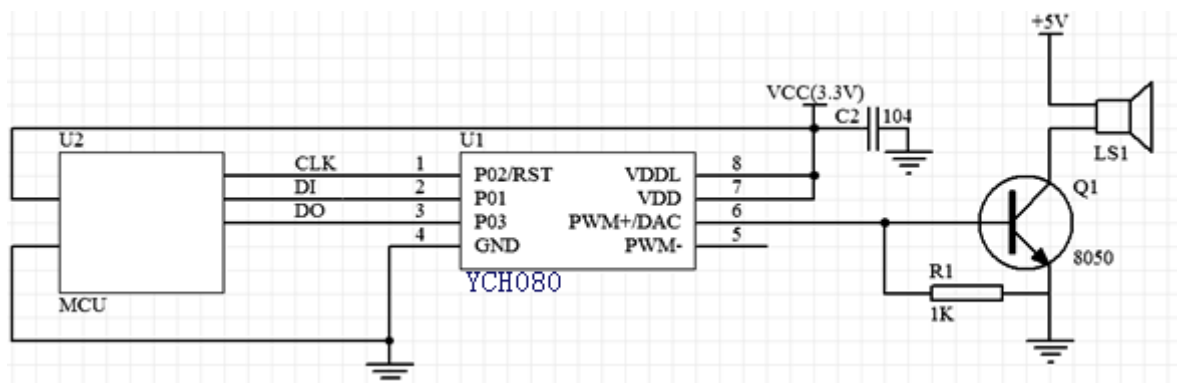


8.3、两线串口控制模式应用电路

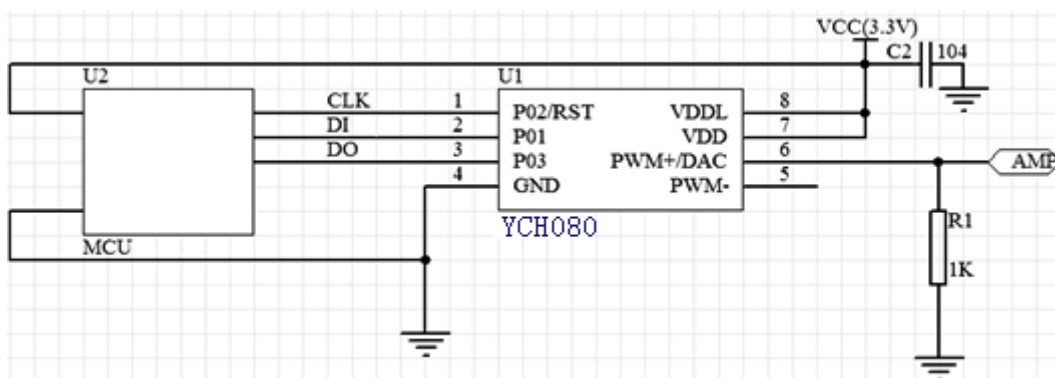
8.3.1、YCH080-8S/P 两线串口模式（MCU 为 DC3V 供电，PWM 输出）



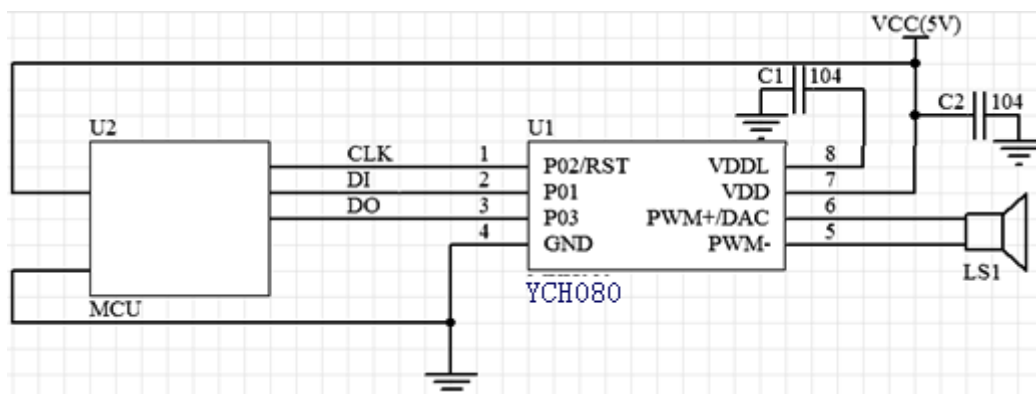
8.3.2、YCH080-8S/P 两线串口模式（MCU 为 DC3V 供电，DAC 输出外接三极管）



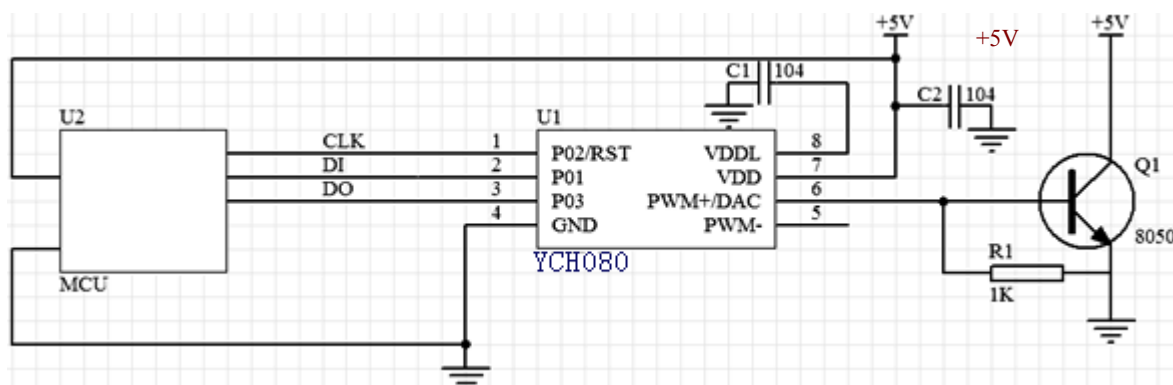
8.3.3、YCH080-8S/P 两线串口模式（MCU 为 DC3V 供电，DAC 输出外接功放）



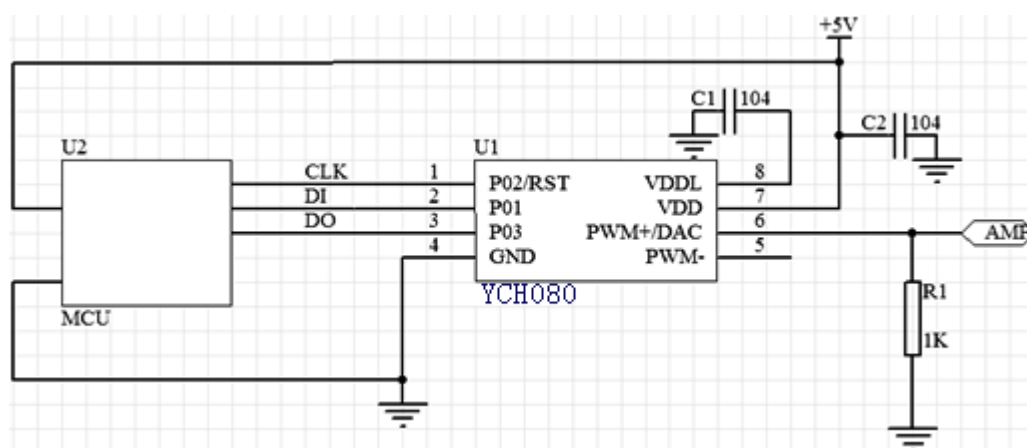
8.3.4、YCH080-8S/P 两线串口模式（MCU 为 DC5V 供电，PWM 输出）



8.3.5、YCH080-8S/P 两线串口模式（MCU 为 DC5V 供电，DAC 输出外接三极管）



8.3.6、YCH080-8S/P 两线串口模式（MCU 为 DC5V 供电，DAC 输出外接功放）





9、程序范例

9.1、一线串口控制程序范例

芯片型号: AT89C2051

晶振 11.0592M

```
#include<at89x51.h>
#include<intrins.h>
#define uchar unsigned char
#define uint unsigned int
sbit key0=P1^0;
sbit key1=P1^1;
sbit key2=P1^2;
sbit rst=P3^3;
sbit sda=P3^4;
uchar num;
void delayms(xms)
{
    uint i,j;
    for(i=xms;i>0;i--)
        for(j=110;j>0;j--);
}
void delay100us(xus)
{
    uint i,j;
    for(i=xus;i>0;i--)
        for(j=10;j>0;j--);
}
void one_line(uchar addr)
{
    uchar i;
    rst=1;
    delay100us(2);
    rst=0;
    delayms(5);
    for(i=0;i<addr;i++)
    {
        sda=1;
        delay100us(1);
        sda=0;
        delay100us(1);
    }
}
```



```
void key()
{
    if(key0==0)
    {
        delayms(10);
        if(key0==0)
        {
            while(!key0);
            num++;
            if(num==0xff)
                num=1;
        }
    }
    if(key1==0)
    {
        delayms(10);
        if(key1==0)
        {
            while(!key1);
            num--;
            if(num==0)
                num=0xff;
        }
    }
    if(key2==0)
    {
        delayms(10);
        if(key2==0)
        {
            while(!key2);
            one_line(num);
        }
    }
}

void init()
{
    rst=0;
    sda=0;
    P1=0xff;
    num=0;
}
```




```
void main()
{
    init();
    while(1)
    {
        key();
    }
}
```

9.2、两线串口控制程序范例

//晶振 11.0592M AT89C2051 单片机，数码管动态显示，用 74LS164 控制

```
#include< at89x51.h >
#include< intrins.h >
#define uchar unsigned char
#define uint unsigned int
sbit cs = P3^2;
sbit clk = P3^3;
sbit sda = P3^4; //定义一线数据口
sbit slk = P3^5; //定义数码管时钟口
sbit dat = P3^7; //定义数码管数据口
uchar code table2[]=
{0xc0,0xf9,0xa4,0xb0,
 0x99,0x92,0x82,0xf8,
 0x80,0x90,0x88,0x83,
 0xc6,0xa1,0x86,0x8e,0xff}; //0-f,全灭
uchar code table1[]="1 music yan shi error 3 music 2 music ";
uchar table[16]="is";
uchar n,a,b,m,send0,send1;
bit flag,flg;
void delayms( uint xms )
{
    uint i,j;
    for( i=xms ; i>0 ; i-- )
        for( j=110 ; j>0 ; j-- );
}
void delay100us( uint xus )
{
    uint i,j;
    for( i=xus ; i>0 ; i-- )
        for( j=10 ; j>0 ; j-- );
}
```



```
void delayus( uint xus )
{
    while(xus--);
}
void sendone(uchar i)
{
    uchar j,c;
    c=table2[i];
    for(j=0;j<8;j++)
    {
        slk=0;
        dat=c&0x80;
        //  _nop_();
        slk=1;
        //  _nop_();
        c<<=1;
    }
    dat=1;
    slk=1;
}
void two_line(uchar addr)
{
    uchar i;
    if((addr==0x18)||((addr==0x0a)||((addr==0x22))
    {
        clk=1;
        delay100us(1);
    }
    send1=send0;
    for(i=0;i<8;i++)
    {
        clk=1;
        sda=addr&0x80;
        if(cs==1)
            send0=send0|0x01;
        else
            send0=send0&0xfe;
        delay100us(2);
        clk=0;
        delay100us(2);
        addr<<=1;
    }
}
```



```
        send0<<=1;
    }
    clk=0;
    sda=0;
}
void send()
{
    uchar d=0;
    if(table[0]==0x02)
    {
        ES=0;
        for(d=30;d<38;d++)
        {
            SBUF=table1[d];
            while(!TI);
            TI=0;
        }
        ES=1;
        for(d=1;d<m;d++)
        {
            two_line(table[d]);
        }
        sendone(table[1]%16);
        sendone(table[1]/16);
        sendone(table[0]%16);
        sendone(table[0]/16);
        if((table[1]==0x22)&&table[2]==0x00)
        {
            if(send0&0x80)
            {
                sendone(8);
                sendone(8);
                sendone(2);
                sendone(0);
            }
            else
            {
                sendone(7);
                sendone(7);
                sendone(2);
                sendone(0);
            }
        }
    }
}
```



```
        }  
    }  
}  
else  
{  
    ES=0;  
    for(d=16;d<21;d++)  
    {  
        SBUF=table1[d];  
        while(!TI);  
        TI=0;  
    }  
    ES=1;  
}  
send0=0;  
send1=0;  
for(d=0;d<m;d++)  
{  
    table[d]=0;  
    delayms(1);  
}  
}  
void init()  
{  
    TMOD=0x20;  
    MXH1=0xfd;  
    TL1=0xfd;  
    SCON=0x50;  
    EA=1;  
    ES=1;  
    TR1=1;  
    P3=0xff;  
    P1=0xff;  
    cs=1;  
    clk=0;  
    sda=0;  
    delayms(5);  
    delayus(1);  
    delay100us(1);  
    sendone(16);  
    sendone(16);
```

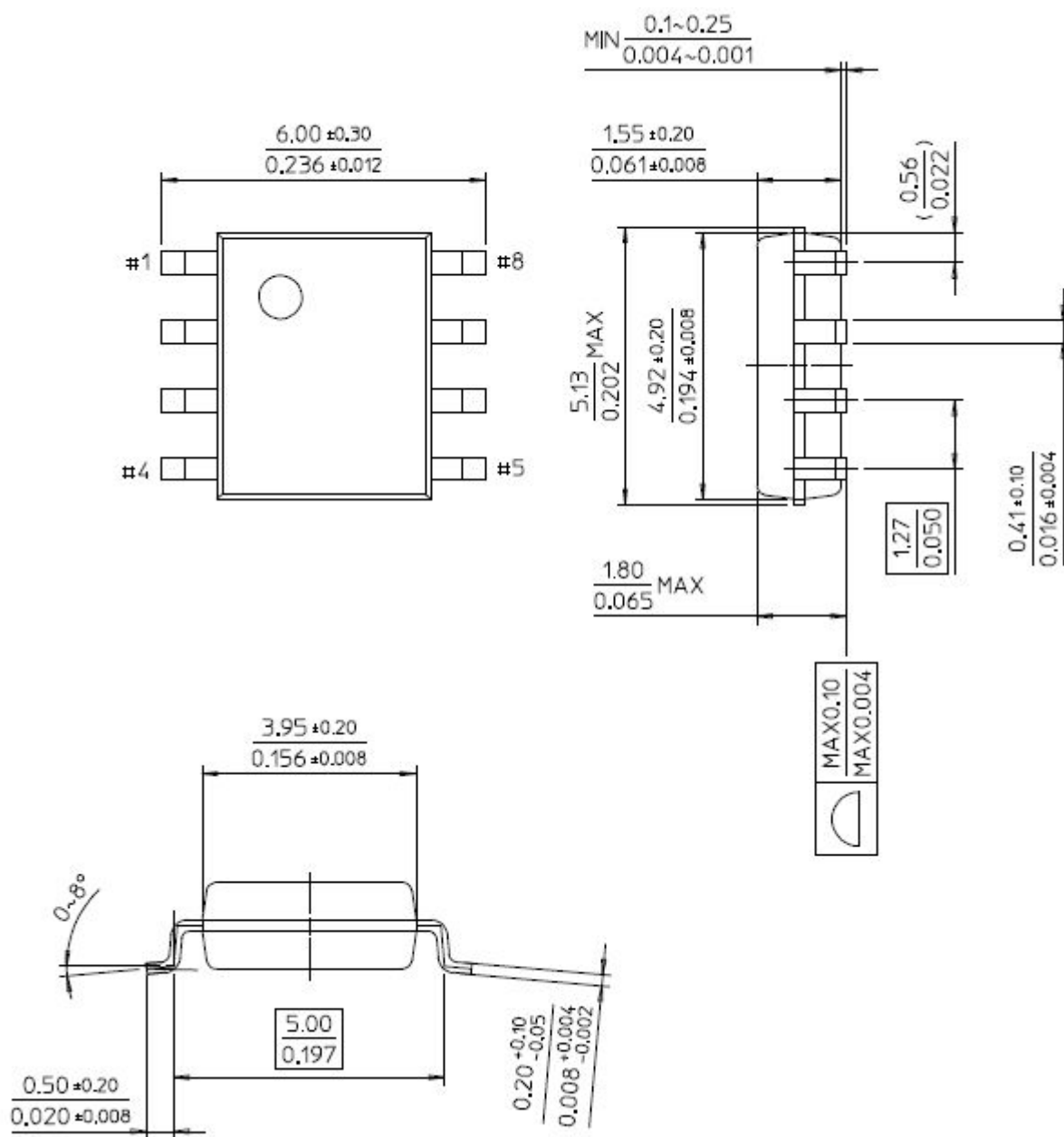


```
sendone(16);
sendone(16);
}
void main()
{
    init();
    while(1)
    {
        if(flag==1)
        {
            flag=0;
            send();
        }
    }
}
void ser() interrupt 4
{
    a=SBUF;
    if(a==0xEE)
    {
        b=1;
        m=0;
    }
    else if((b==1)&&(a!=0xEE)&&(a!=0xED))
    {
        table[n++]=SBUF;
        m++;
    }
    else if(a==0xED)
    {
        b=0;
        n=0;
        flag=1;
    }
    RI=0;
}
```

10、封装尺寸图

10.1、YCH080-8S 封装尺寸图

单位: mm



11、历史版本记录

版本号	日期	描述
V1.0	2012-05-30	原始版本
V1.4	2013-8-10	修改电路图