

CXG1 系列语音芯片使用资料



CXG1 系列（一次性烧写语音芯片）语音内容不可更换，在大多数应用中都作为从片使用，受 MCU 控制，不涉及其它外围电路，成本相对较低。适合应用在不需修改语音、语音长度短，音质较低场合。



目录

- 一、功能简述 3
- 二、用范围 3
- 三、电气参数 3
- 四、管脚图 4
 - 4.1、CXG1 系列-8S/8P4
- 五、控制模式 4
 - 5.1、按键控制 4
 - 5.2、一线串口控制 6
- 六、复位及输出状态 8
 - 6.1、硬件复位 8
 - 6.2、输出状态 8
- 七、应用电路 9
 - 7.1、按键控制模式应用电路 9
 - 7.1.1、CXG1 系列-8S/P 按键控制（PWM 输出） 9
 - 7.1.2、CXG1 系列-8S/P 按键模式（DAC 输出外接三极管） 9
 - 7.1.3、CXG1 系列-8S/P 按键模式（DAC 输出外接功放） 10
 - 7.2、一线串口控制模式应用电路 10
 - 7.2.1、CXG1 系列-8S/P 一线串口模式（PWM 输出） 10
 - 7.2.2、CXG1 系列-8S/P 一线串口模式（DAC 输出外接三极管） 10
 - 7.2.3、CXG1 系列-8S/P 一线串口模式（DAC 输出外接功放） 11
- 八、程序范例 11
 - 8.1、一线串口控制程序范例（PWM 输出方式） 11
 - 8.2 一线串口控制程序范例（DAC 输出方式） 13
 - 8.3 V4 版本一线串口控制 DAC 程序范例
- 九、封装尺寸图 19
 - 9.1、CXG1 系列-8S 封装尺寸图 19
 - 9.2、CXG1 系列-8P 封装尺寸图 20
- 版本记录 21



一、功能简述

1. 可编程一次性烧录 (OTP) 语音芯片;
2. 语音长度有 10、20、40、80 秒 (6KHz 采样率);
3. PWM 和 DAC 两种音频输出方式;
4. 内部集成时钟振荡器;
5. 最大可存放 700 个语音文件;
6. 一线串口最多可加载 63 段地址的语音;
7. 具有按键控制模式、一线串口控制模式;
8. 触发防抖时间: 50us (串口) 和 10ms (按键);
9. 7 种按键触发方式;
10. 支持播放不同采样率的语音文件;
11. 支持 BUSY 状态输出功能;
12. 工作电压: DC2.4~3.6, 最佳 3.3V。

二、用范围

1. 家电 (冰箱、电磁炉、电饭煲、微波炉);
2. 汽车 (防盗报警器、倒车雷达、GPS 导航仪、电子狗、中控锁);
3. 娱乐设备 (游戏机、游乐机);
4. 学习模型 (早教机、儿童有声读物);
5. 智能交通设备 (汽车站、收费站、停车场);
6. 通信设备 (电话交换机、电话机);
7. 工业控制领域 (电梯、工业设备);
8. 安防提示报警

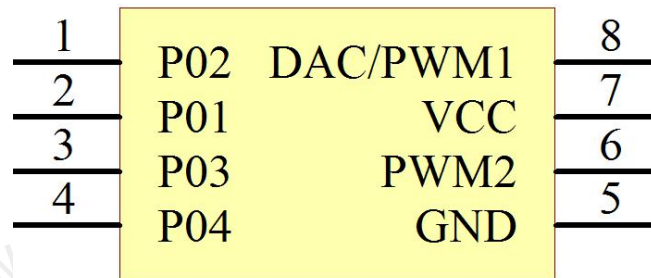
三、电气参数

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	3.6	V
待机电流	ISB	V _{DD} =3V, 没负载	1	1.5	3	uA
工作电流	IOP	V _{DD} =3V, 没负载		600		uA
输出驱动电流	IOD	V _{DD} =3V, V _{OUT} =0.6V		10		mA
输出灌电流	IOS	V _{DD} =3V, V _{OUT} =2.4V		30		mA
PWM 驱动电流	IOD	V _{DD} =3V, V _{OUT} =1.5V		200		mA
PWM 灌电流	IOS	V _{DD} =3V, V _{OUT} =1.5V		200		mA



四、管脚图

4.1、CXG1 系列-8S/8P



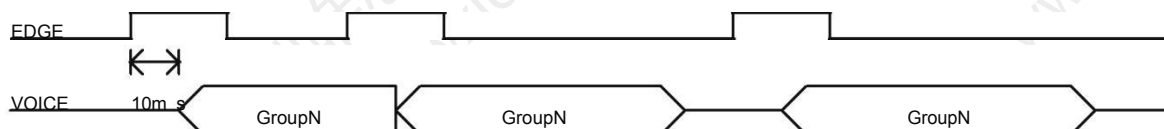
封装引脚	引脚标号	简述	功能描述
1	P02/RST	KEY2/RST	按键 2/复位引脚
2	P01	KEY1/DATA	按键 1/一线串口 DATA
3	P03	KEY3/BUSY	按键 3/BUSY(默认)
4	P04	KEY4	按键 4/BUSY (可选)
5	GND	GND	地线
6	PWM-	PWM-	PWM-音频输出
7	VDD	VDD	电源输入端, 供电电压 2.4~3.6V, 需要接 104 电容到地线
8	PWM+/DAC	PWM+/DAC	PWM+音频输出/DAC 音频输出

五、控制模式

5.1、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。

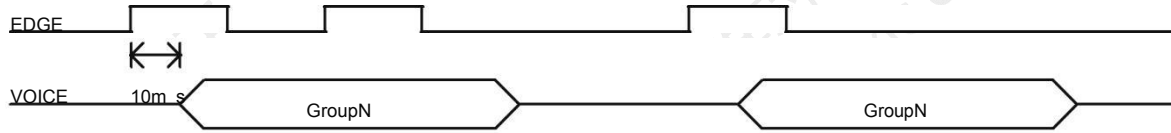
脉冲可重复





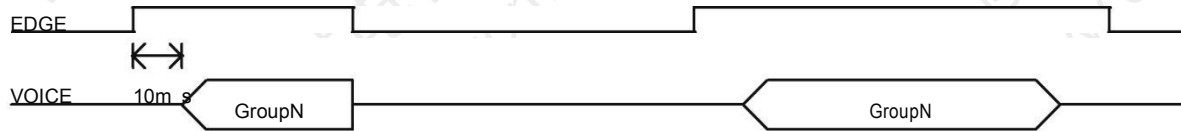
收到正脉冲信号后开始播放语音，在语音结束前如果还收到第二次正脉冲信号，则重新开始播放语音，在语音播放的过程中无收到正脉冲信号则播放完整段语音。

脉冲不可重复



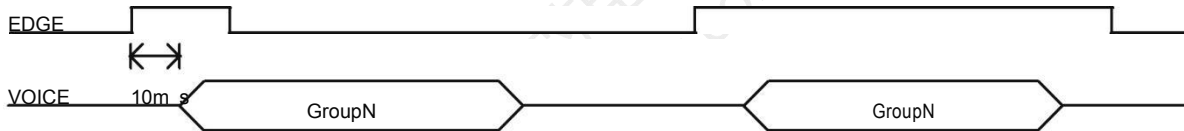
收到正脉冲信号后开始播放语音，在语音结束前如果再次收到正脉冲信号，则没有动作产生，在语音播放结束后再次收到正脉冲信号才会重新播放语音。

脉冲保持



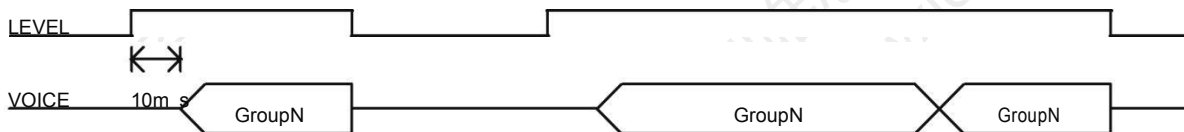
收到正脉冲信号后开始播放语音，正脉冲信号停止时，语音也停止播放。如果正脉冲信号一直保持，并且持续的时间超出语音播放的长度，则播放完语音即停止，不再继续播放语音。

脉冲不保持



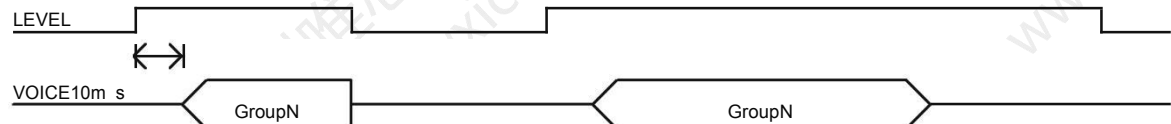
收到正脉冲信号后开始播放语音，正脉冲信号停止后语音继续播放完毕，如果正脉冲信号一直保持，并且持续的时间超出语音播放的长度，则播放完语音即停止，不再继续播放语音。

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。

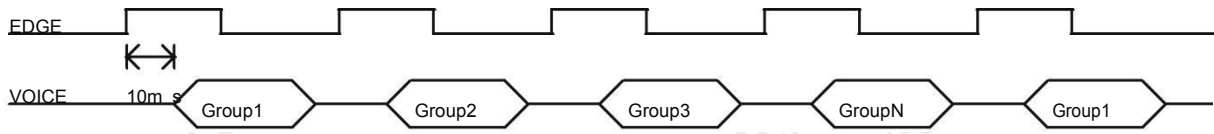
电平保持不可循环



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



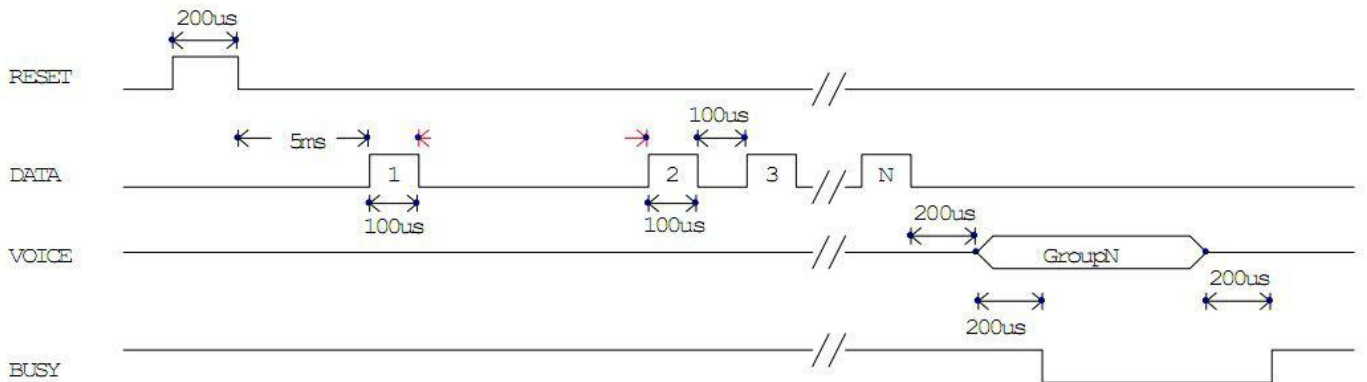
此触发方式仅限于在 P00 控制端口。收到正脉冲信号后开始播放第一段地址语音，再次收到正脉冲信号则播放第二段地址语音，当触发到最后一段后，再次触发则重新开始播放第一段语音，如此循环。

5.2、一线串口控制

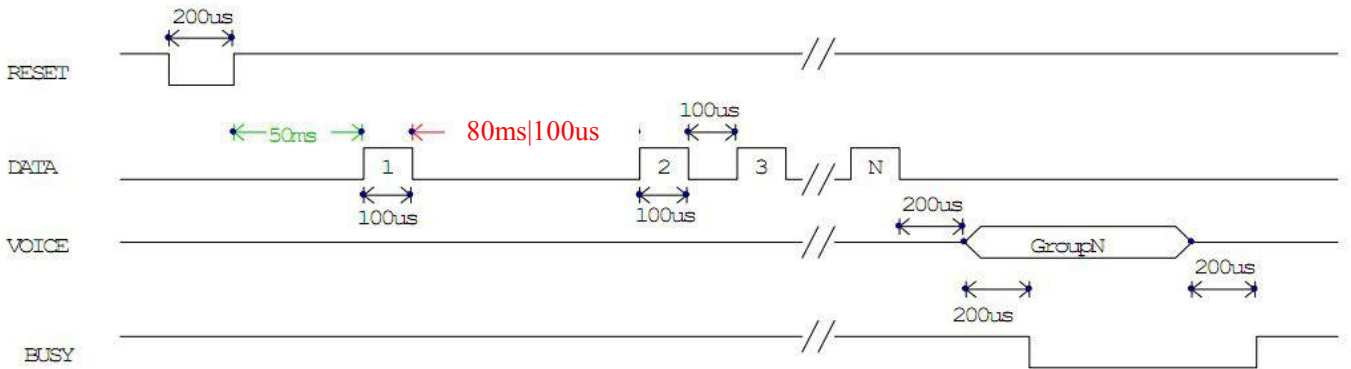
5.2.1 常规一线串口控制

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。

一线串口控制时序中，有分硬件复位和软件复位两种，默认采用软件复位（高电平有效）。其中软件复位是先发送 200us 的 RESET 信号，等待 5ms 后发送 DATA，DATA 中第一个脉冲保持的 100us 高电平，等待 80ms（DAC 输出方式则等待 80MS，PWM 输出方式则等待 100us）后发送第二个脉冲，自第二脉冲起，脉冲高电平保持时间为 100us 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 80MS 之分，是 PWM 输出时选用 100US，DAC 输出时，选用 80MS。**



下图位硬件复位时的时序图，其中硬件复位是先发送 50MS 的/RESET 信号，等待 5ms 后发送 DATA，DATA 中第一个脉冲保持在 100us 的高电平，等待 50ms（DAC 输出方式等待 50MS，PWM 输出方式则等待 100us）后发送第二个脉冲，每个脉冲高电平保持时间为 100us，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 80MS 之分，是 PWM 输出时选用 100US，DAC 输出时，选用 80MS。**



脉冲数量以及所触发语音地址的对应关系如下表所示

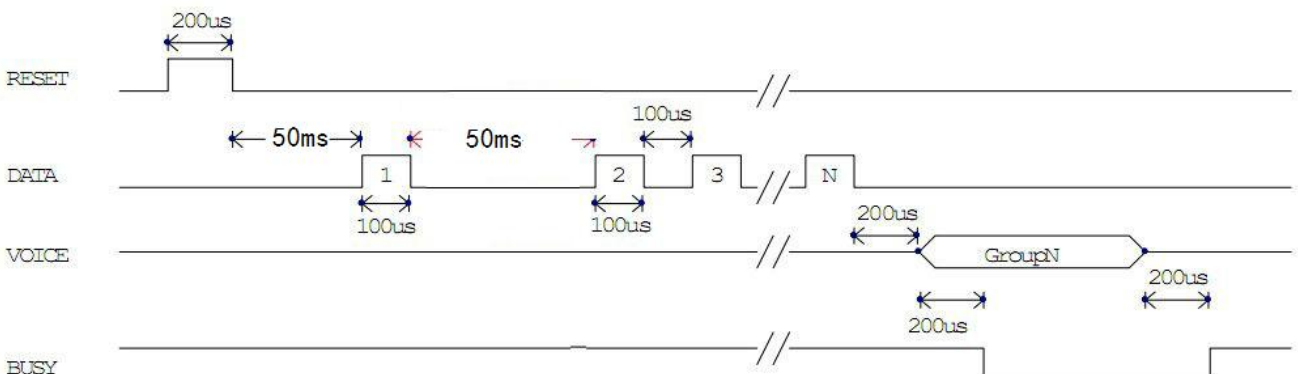
序号	脉冲数	语音地址
1	1	0
2	2	1
3	3	2
4
5	32	31

注意：为了防止误触发，在一线串口控制模式中，第一个地址语音必须为静音。

5.2.2 V4 版一线串口 DAC 控制

常规版本不能在每段中增加循环，设置每段音量，跳转，软件延时等复杂的方式，需要复杂的方式使用 V4 版本软件制作，控制时序与常规版本有一点差异。

软件复位是先发送 200us 的 RESET 信号，等待 50ms~100ms（不能少于 50ms）后发送 DATA，DATA 中第一个脉冲保持在 100us 高电平，与第二个脉冲之间的间隔时间需要 50ms。剩余脉冲间隔 100us，发送后等待 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。

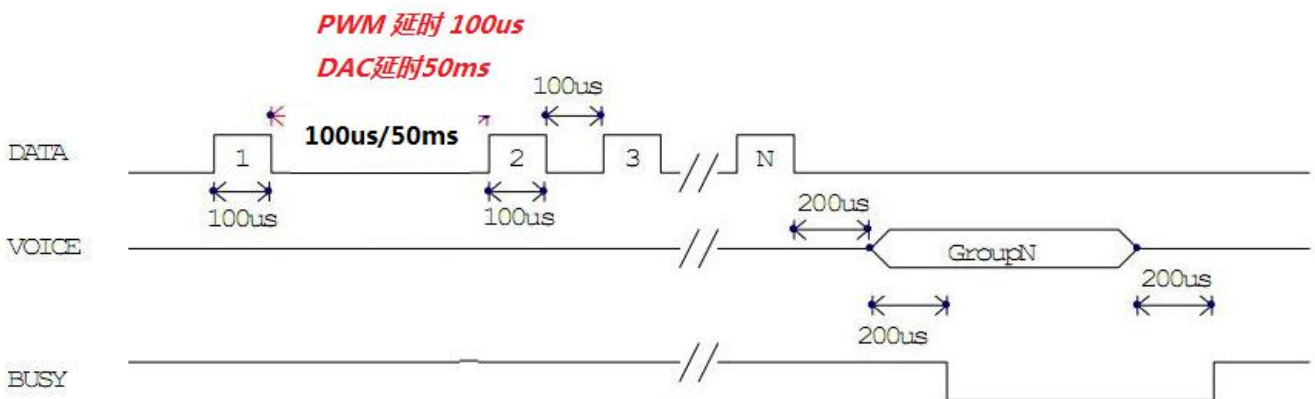




5.2.3 新版一线串口控制（限 CXG1080）

新版的一线串口是一个 I/O 控制无需复位引脚的控制方式，与常规控制相比，仅相差复位引脚不同，新版通信无需复位引脚，可配硬件复位（可不用）。

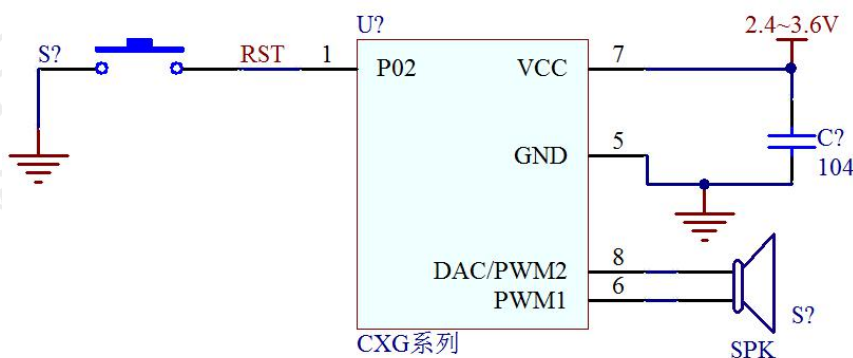
通信只需发 DATA，第一个脉冲保持在 100us 高电平，与第二个脉冲之间的间隔时间需要 100us (PWM 模式)/50ms (DAC 模式)。剩余脉冲间隔 100us，发送后等待 200us，开始播放地址语音，再过 200us 后 BUSY 信号发生变化。其时序如下图所示。



六、复位及输出状态

6.1、硬件复位

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。



6.2、输出状态

P01、P03、P04 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

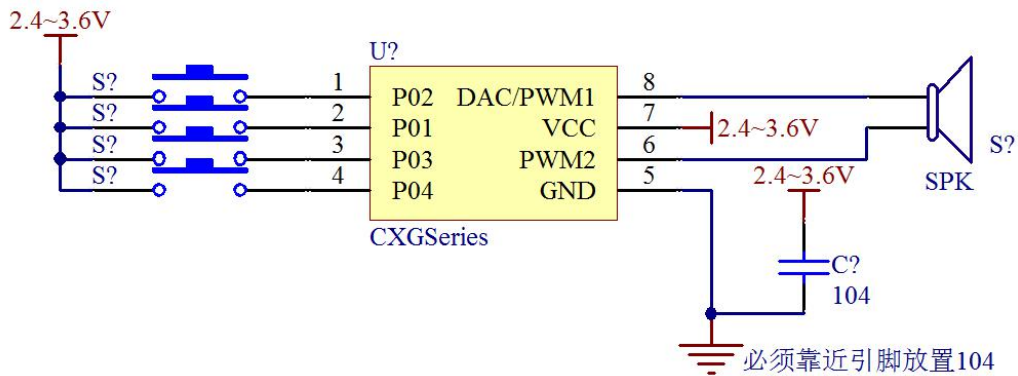


- 播放语音时输出高电平
- LED 闪光频率 6Hz
- LED 闪光频率 3Hz
- LED 闪光频率 1.5Hz
- LED 闪光频率 0.75Hz

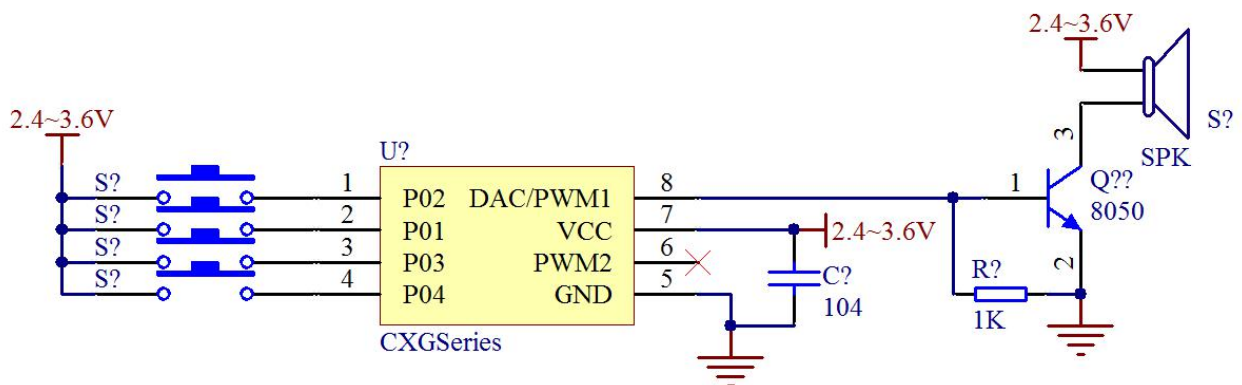
七、应用电路

7.1、按键控制模式应用电路

7.1.1、CXG1 系列-8S/P 按键控制（PWM 输出）

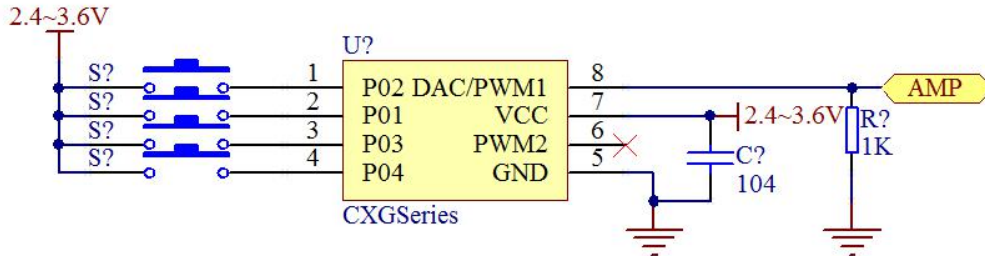


7.1.2、CXG1 系列-8S/P 按键模式（DAC 输出外接三极管）





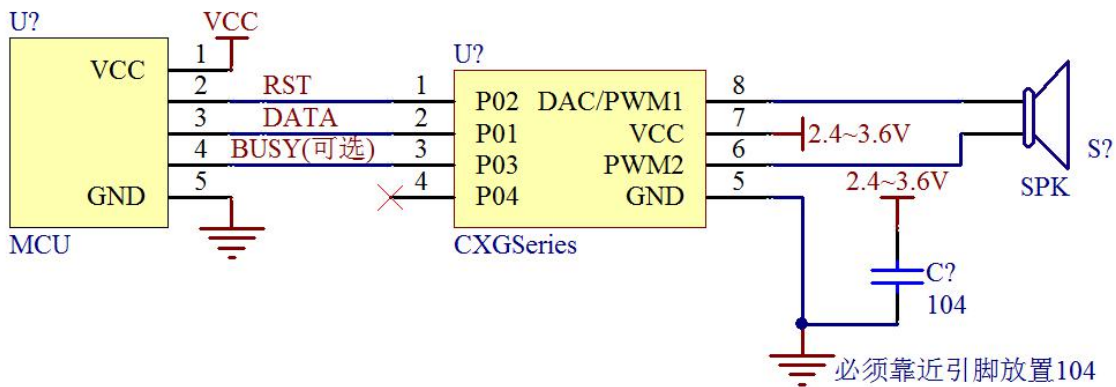
7.1.3、CXG1 系列-8S/P 按键模式 (DAC 输出外接功放)



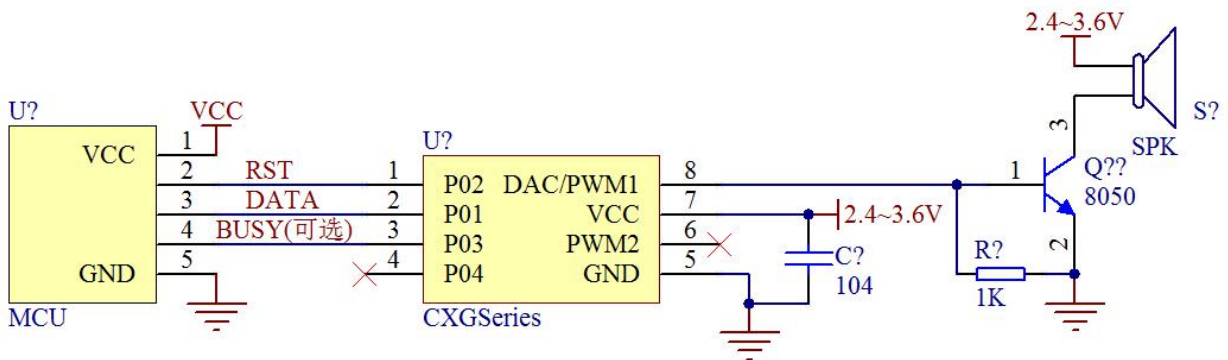
注意：DAC 输出端 1K 负载电阻不能去掉

7.2、一线串口控制模式应用电路

7.2.1、CXG1 系列-8S/P 一线串口模式 (PWM 输出)

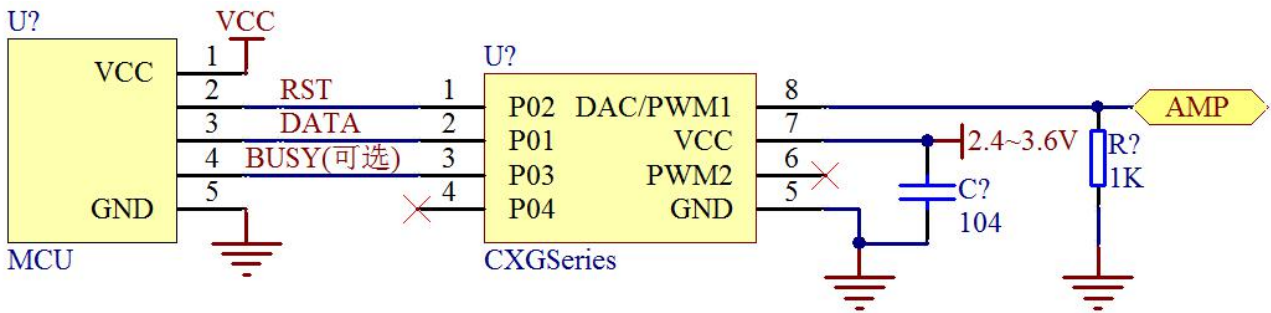


7.2.2、CXG1 系列-8S/P 一线串口模式 (DAC 输出外接三极管)





7.2.3、CXG1 系列-8S/P 一线串口模式（DAC 输出外接功放）



注意：DAC 输出端 1K 负载电阻不能去掉

八、程序范例

8.1、一线串口控制程序范例（PWM 输出方式）

```
//单片机: STC10F04
//晶振 :11.0592MHz
#include"reg51.h" /*reg51 头文件*/
sbit CXG_REST=P2^1;
sbit CXG_PLUSE=P2^0;
sbit KEY1=P3^7;

; 模块名称:Delay_10us
; 功能: 延时函数,STC10F04 10us @11.0592MHz
; 入参: unsigned int n 延时次数
; 出参: 无

*/

void Delay_10us(unsigned int n)
{
    unsigned char i;
    for(;n>0; n-- )
    {
        for(i=8;i>0;i-- )
        {
            nop_();
            nop_();
            nop_();
            nop_();
            nop_();
        }
    }
}
```



```
    nop_();  
    nop_();  
    }  
}  
}
```

：模块名称：Delay_10ms
：功能：延时函数,STC10F04 10 m s @11.0592MHz
：入参：unsigned int z 延时次数
：出参：无

```
-----  
void Delay_10ms(unsigned int  
z)  
{  
    unsigned int i,j;  
    for (i=z, i>0; i-- )  
    {  
        for (j=8450;j>0;j--  
        );  
    }  
}
```

：模块名称：CXG_1 Line PWM
：功能：CXG 系列芯片一线串口控制，PWM 输出方式适用
：入参：unsigned char cnt 脉冲数，必须大于 0
：出参：0 出错；1 成功。

```
-----  
unsigned char CXG_1LinePWM(unsigned char cut)  
{  
if(cut==0)  
    return 0;  
CXG_REST=1  
Delay_10us(20); //复位高电平时间 200us  
CXG-REST=0;  
Delay_1ms(5); //5ms  
While(cut--)  
{  
CXG_PLUSE=1;  
Delay_10us(10); // 100us 高电平  
CXG_PLUSE=0;  
Delay_10us(10); // 100us 低电平  
}  
CXG_PLUSE=0;  
return 1;
```



```

}

-----
; 模块名称: main
; 功能: main
; 入参: void
; 出参: void
:----- */

void main(void)
{
    CXG_REST=0;
    CXG_PLUSE=0;
    while(1)
    {
        if(KEY1==0)
        {
            CXG_1LinePWM(0x02);
        }
        Delay_10ms(30);
    }
}

```

8.2 一线串口控制程序范例（DAC 输出方式）

```

//单片机: STC10F04
//晶振 :11.0592MHz
#include: : "reg51.h" /*reg51 头文件*/
sbit CXG_REST=P2^1;
sbit CXG_PLUSE=P2^0;
sbit CEY1=P3^7;

-----
; 模块名称:Delay_10us
; 功能: 延时函数,STC10F04 10us @11.0592MHz
; 入参: unsigned int
n 延时次数
; 出参: 无

----- */

void Delay_10us(unsigned int n)
{
    unsigned char i;
    for(n=i;n>0; n--)
    {
        for(i=8;i>0;i--)
        {

```



```

nop_();
nop_();
nop_();
nop_();
nop_();
nop_();
nop_();
nop_();
}
}
}

```

: 模块名称: Delay_10ms
: 功能: 延时函数,STC10F04 10 ms @11.0592MHz
: 入参: unsigned int z 延时次数
: 出参: 无

```

----- */
void Delay_10ms(unsigned int z)
{
    unsigned int i,j;
    for (i=z, i>0; i-->0)
    {
        for (j=8450;j>0;j-->0);
    }
}
}

```

: 模块名称: CXG_1 Line DAC
: 功能: CXG 系列芯片一线串口控制, DAC 输出方式适用
: 入参: unsigned char cnt 脉冲数, 必须大于 0
: 出参: 0 出错; 1 成功。

```

----- */
unsigned char CXG_1LineDAC(unsigned char cnt)
{
    if(cnt==0)
        return 0;
    CXG_REST=1;
    Delay_10us(20); //复位高电平时间 200us
    CXG-REST=0;
    Delay_10us(500); //5ms
    /*******第一个脉冲******/
    CXG_PLUSE=1;
    Delay_10us(10); // 100us 高电平
    CXG_PLUSE=0;
    Delay_10ms(8); //80ms 低电平
}

```



```

/*****/
/*其他脉冲*/
While(--cut)
{
    CXG_PLUSE=1;
    Delay_10us(10); //100us 高电平
    CXG_PLUSE=0;
    Delay_10us(10); //100us 低电平

}

    CXG_PLUSE=0;

    return 1;
}
/*-----
; 模块名称: main
; 功能: main 函数
; 入参: void
; 出参: void
; -----*/
void main(void)
{
    CXG_REST=0;
    CXG_PLUSE=0;

    while(1)
    {
        if(KEY1==0)
        {
            CXG_1LinePWM(0x02);
        }
        Delay_10ms(30);
    }
}

```

8.3 V4 版本一线串口控制 DAC 程序范例

```

//单片机: STC10F04
//晶振 :11.0592MHz
#include : "reg51.h" /*reg51 头文件*/
sbit CXG_REST=P2^1;

```



```
sbit CXG_PLUSE=P2^0;
```

```
sbit CEY1=P3^7;
```

```
-----  
; 模块名称:Delay_10us
```

```
; 功能: 延时函数,STC10F04 10us @11.0592MHz
```

```
; 入参: unsigned int  
n 延时次数
```

```
; 出参: 无
```

```
-----  
*/
```

```
void Delay_10us(unsigned int n)
```

```
{  
    unsigned char i;  
    for(n=i;n>0; n--)  
    {  
        for(i=8;i>0;i--)  
        {  
            nop_();  
            nop_();  
            nop_();  
            nop_();  
            nop_();  
            nop_();  
            nop_();  
        }  
    }  
}
```

```
-----  
; 模块名称: Delay_10ms
```

```
; 功能: 延时函数,STC10F04 10 ms @11.0592MHz
```

```
; 入参: unsigned int z 延时次数
```

```
; 出参: 无
```

```
-----  
*/
```

```
void Delay_10ms(unsigned int z)
```

```
{  
    unsigned int i,j;  
    for (i=z, i>0; i--)  
    {  
        for (j=8450;j>0;j--  
            );  
    }  
}
```

```
-----  
; 模块名称: CXG_1 Line DAC
```




； 功能：CXG 系列芯片一线串口控制，DAC 输出方式适用
； 入参：unsigned char cnt 脉冲数，必须大于 0
； 出参：0 出错；1 成功。

```
----- */
unsigned char CXG_1LineDAC(unsigned char cut)
{
if(cnt==0)
    return 0;
CXG_REST=1
Delay_10us(20); //复位高电平时间 200us
CXG-REST=0;
Delay_10ms(50); //50ms

CXG_PLUSE=1;
Delay_10us(10);

CXG_PLUSE=0;
Delay_10ms(50);
cut--;
/*****/
/*其他脉冲*/
While(cut--)
{
CXG_PLUSE=1;
Delay_10us(10); //100us 高电平
CXG_PLUSE=0;
Delay_10us(10); //100us 低电平

}

CXG_PLUSE=0;

return 1;
}
/-----*/
； 模块名称： main
； 功能： main 函数
； 入参： void
； 出参： void
； -----*/

void main(void)
{
CXG_REST=0;

CXG_PLUSE=0;
```



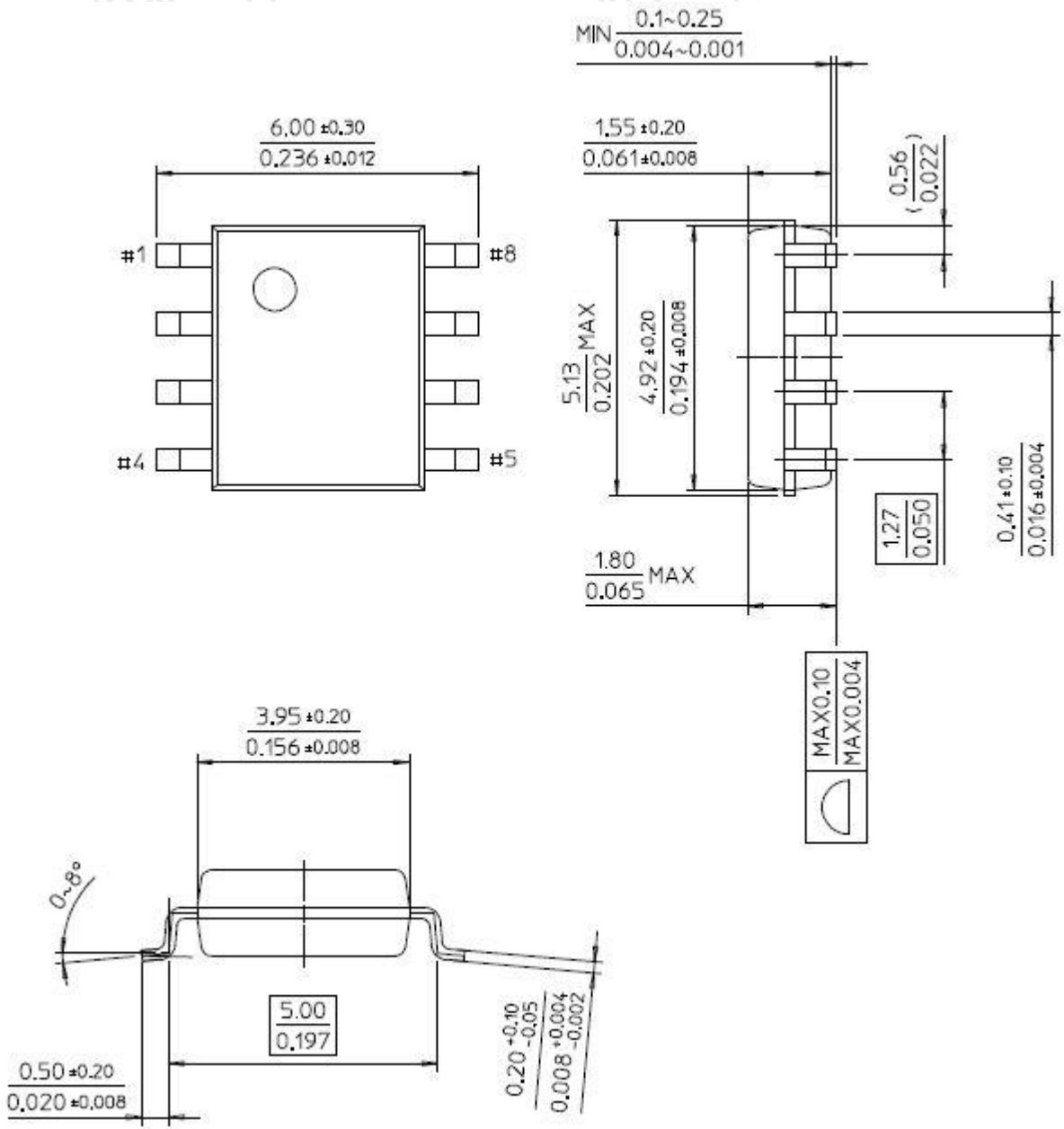
```
while(1)
{
  if(KEY1==0)
  {
    CXG_1LinePWM(0x02);
  }
  Delay_10ms(30);
}
}
```



九、封装尺寸图

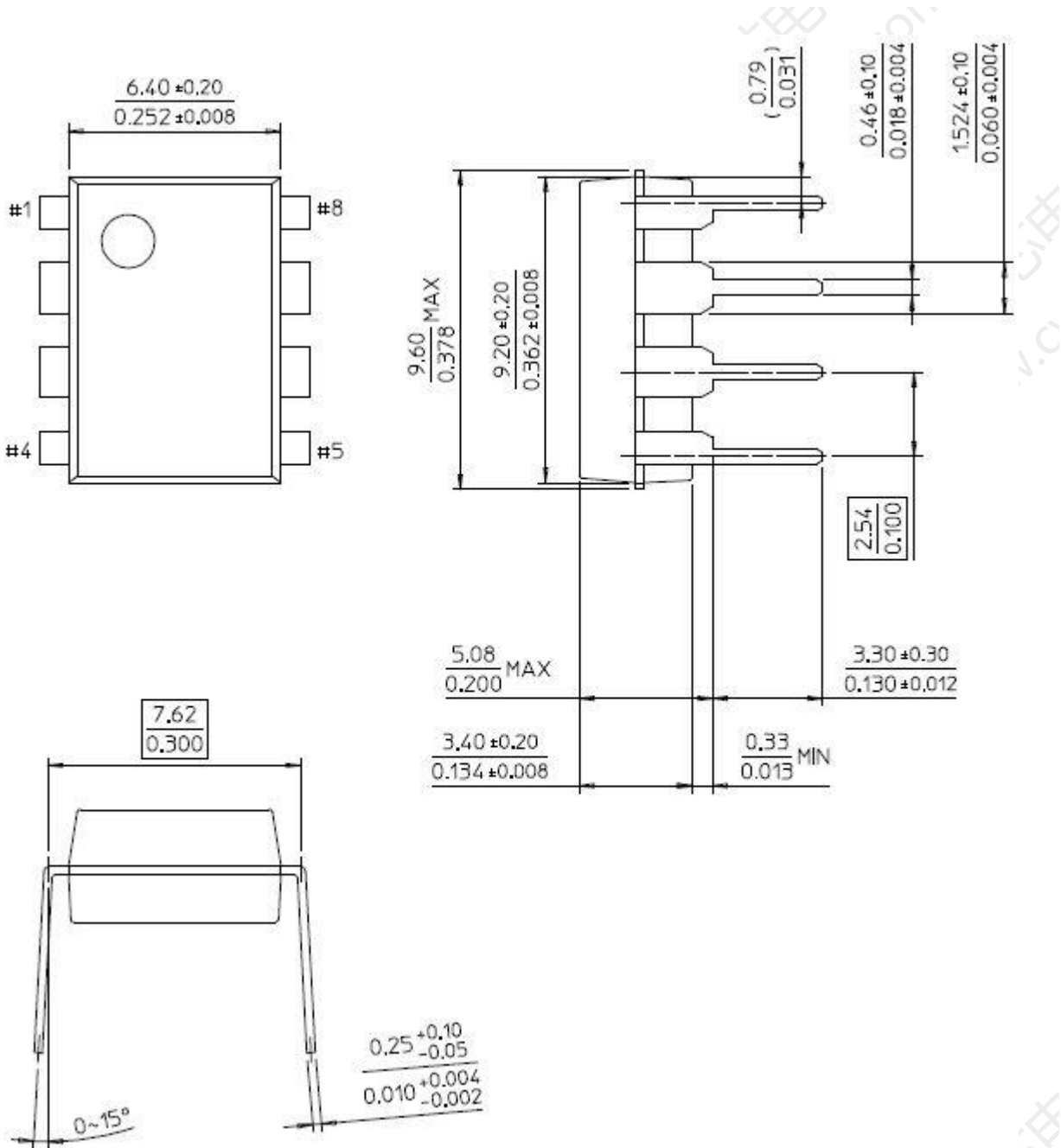
9.1、CXG1 系列-8S 封装尺寸图

单位: mm





9.2、CXG1 系列-8P 封装尺寸图





10、版本记录

版本号	修改说明	修改日期
V1.00	原始版本	2015-10-29
V1.01	增加参考代码	2016-8-15
V1.02	优化串口控制方式	2018.11.21
V1.03	增加 V4 版本控制时序说明	2018.12.06