



WTH040 语音芯片使用资料 V1.6

目 录

1、产品特点.....	3
2、芯片选型.....	3
3、应用范围.....	3
4、管脚图.....	4
4.1、WTH040-8S/WTH040-8P	4
4.2、WTH040-16S	4
5、电气参数.....	5
6、控制模式.....	5
6.1、按键控制	5
6.2、一线串口控制	7
6.3、两线串口控制	8
6.3.1、发送地址命令	8
6.3.2、读取芯片工作状态.....	8
6.3.3、控制命令	9
6.3.4、语音地址	9
7、复位及输出状态	9
7.1、硬件复位	9
7.2、输出状态	10
8、应用电路.....	10
8.1、按键控制模式应用电路	10
8.1.1、WTH040-8S/WTH040-8P按键控制（DC3V供电，PWM输出）	10
8.1.2、WTH040-8S/WTH040-8P按键模式（DC3V供电，DAC输出外接三极管）	10
8.1.3、WTH040-8S/WTH040-8P按键模式（DC3V供电，DAC输出外接功放）	11
8.1.4、WTH040-8S/WTH040-8P按键模式（DC5V供电，PWM输出）	11
8.1.5、WTH040-8S/WTH040-8P按键模式（DC5V供电，DAC输出外接三极管）	11
8.1.6、WTH040-8S/WTH040-8P按键模式（DC5V供电，DAC输出外接功放）	11
8.1.7、WTH040-16S按键模式（DC3V供电，PWM输出）	12
8.1.8、WTH040-16S按键模式（DC3V供电，DAC输出外接三极管）	12
8.1.9、WTH040-16S按键模式（DC3V供电，DAC输出外接功放）	12
8.1.10、WTH040-16S按键模式（DC5V供电，PWM输出）	12
8.1.11、WTH040-16S按键模式（DC5V供电，DAC输出外接三极管）	13
8.1.12、WTH040-16S按键模式（DC5V供电，DAC输出外接功放）	13
8.2、一线串口控制模式应用电路	13
8.2.1、WTH040-8S/WTH040-8P一线串口模式（MCU为DC3V供电，PWM输出）	13
8.2.2、WTH040-8S/WTH040-8P一线串口模式（MCU为DC3V供电，DAC输出外接三极管）	14
8.2.3、WTH040-8S/WTH040-8P一线串口模式（MCU为DC3V供电，DAC输出外接功放）	15
8.2.4、WTH040-8S/WTH040-8P一线串口模式（MCU为DC5V供电，PWM输出）	15
8.2.5、WTH040-8S/WTH040-8P一线串口模式（MCU为DC5V供电，DAC输出外接三极管）	16
8.2.6、WTH040-8S/WTH040-8P一线串口模式（MCU为DC5V供电，DAC输出外接功放）	16
8.2.7、WTH040-16S一线串口模式（MCU为DC3V供电，PWM输出）	17



8.2.8、WTH040-16S一线串口模式（MCU为DC3V供电，DAC输出外接三极管）	17
8.2.9、WTH040-16S一线串口模式（MCU为DC3V供电，DAC输出外接功放）	18
8.2.10、WTH040-16S一线串口模式（MCU为DC5V供电，PWM输出）	18
8.2.11、WTH040-16S一线串口模式（MCU为DC5V供电，DAC输出外接三极管）	19
8.2.12、WTH040-16S一线串口模式（MCU为DC5V供电，DAC输出外接功放）	19
8.3、两线串口控制模式应用电路	20
8.3.1、WTH040-8S/WTH040-8P两线串口模式（MCU为DC3V供电，PWM输出）	20
8.3.2、WTH040-8S/WTH040-8P两线串口模式（MCU为DC3V供电，DAC输出外接三极管）	20
8.3.3、WTH040-8S/WTH040-8P两线串口模式（MCU为DC3V供电，DAC输出外接功放）	21
8.3.4、WTH040-8S/WTH040-8P两线串口模式（MCU为DC5V供电，PWM输出）	21
8.3.5、WTH040-8S/WTH040-8P两线串口模式（MCU为DC5V供电，DAC输出外接三极管）	21
8.3.6、WTH040-8S/WTH040-8P两线串口模式（MCU为DC5V供电，DAC输出外接功放）	21
8.3.7、WTH040-16S两线串口模式（MCU为DC3V供电，PWM输出）	22
8.3.8、WTH040-16S两线串口模式（MCU为DC3V供电，DAC输出外接三极管）	22
8.3.9、WTH040-16S两线串口模式（MCU为DC3V供电，DAC输出外接功放）	22
8.3.10、WTH040-16S两线串口模式（MCU为DC5V供电，PWM输出）	22
8.3.11、WTH040-16S两线串口模式（MCU为DC5V供电，DAC输出外接三极管）	23
8.3.12、WTH040-16S两线串口模式（MCU为DC5V供电，DAC输出外接功放）	23
8.4、WTH系列语音芯片供电电压DC3V，MCU供电电压DC5V供电时通讯线的接法	23
9、程序范例	24
9.1、一线串口控制程序范例(适应于PWM输出方式)	24
9.2、一线串口控制程序范例(适应于DAC输出方式)	26
9.3、两线串口控制程序范例	28
10、封装尺寸图	34
10.1、WTH040-8S封装尺寸图	34
10.2、WTH040-8P封装尺寸图	35
10.3、WTH040-16S封装尺寸图	36
11、历史版本记录	37



1、产品特点

- 可编程一次性烧录（OTP）语音芯片；
- 语音长度可达40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；
- 一线最多可加载63段地址的语音，二线最多可以加载255段语音；
- 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；
- 工作电压：DC2.4 ~ 5.0V。

2、芯片选型

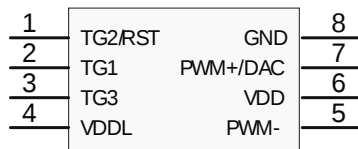
芯片型号	封装形式	语音长度	BUSY 输出	控制端口	输出端口
WTH040-8S	SOP8	40S	支持	3	0
WTH040-8P	DIP8	40S	支持	3	0
WTH040-16S	SOP16	40S	支持	4	4

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）；
- 高级玩具。

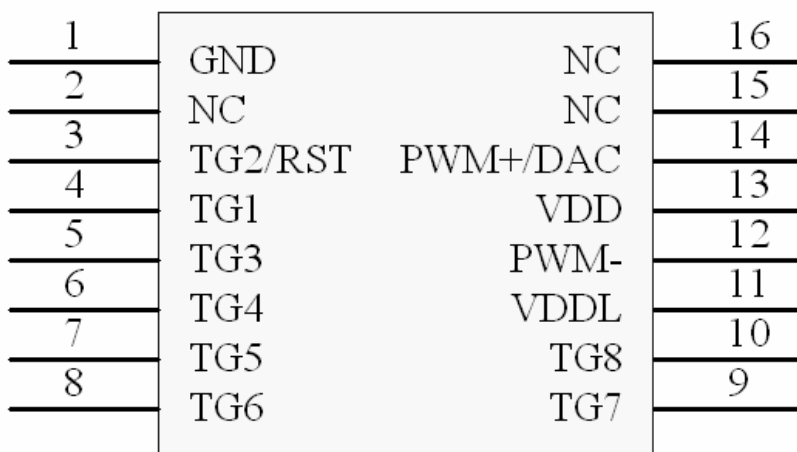
4、管脚图

4.1、WTH040-8S/WTH040-8P



封装引脚	引脚标号	简述	功能描述
1	TG2/RST	KEY2/RST/SCK	按键 2/复位按键/两线串口时钟输入
2	TG1	KEY1/DI/DATA	按键 1/两线串口 DATA IN/一线串口 DATA
3	TG3	KEY3/DO	按键 3/两线串口 DATA OUT
4	VDDL	VDDL	电源输入端，供电电压 2.4 ~ 3.6V，需要接 104 电容到地线
5	PWM-	PWM-	PWM-音频输出
6	VDD	VDD	电源输入端，供电电压 2.4 ~ 5V，需要接 104 电容到地线
7	PWM+/DAC	PWM+/DAC	PWM+音频输出/DAC 音频输出
8	GND	GND	地线

4.2、WTH040-16S



封装引脚	引脚标号	简述	功能描述
1	GND	GND	地线
2	NC	NC	空
3	TG2/RST	KEY2/RST/SCK	按键 2/复位按键/两线串口时钟输入
4	TG1	KEY1/DI/DATA	按键 1/两线串口 DATA IN/一线串口 DATA
5	TG3	KEY3/DO	按键 3/两线串口 DATA OUT
6	TG4	KEY4	按键 4



7	TG5	KEY5/OUT1	按键 5/输出 BUSY 信号或者闪光信号
8	TG6	KEY6/OUT2	按键 6/输出 BUSY 信号或者闪光信号
9	TG7	KEY7/OUT3	按键 7/输出 BUSY 信号或者闪光信号
10	TG8	KEY8/OUT4	按键 8/输出 BUSY 信号或者闪光信号
11	VDDL	VDDL	电源输入端，供电电压 2.4 ~ 3.6V
12	PWM-	PWM-	PWM-音频输出
13	VDD	VDD	电源输入端，供电电压 2.4 ~ 5V，需要接 104 电容到地线
14	PWM+/DAC	PWM+/DAC	PWM+ 音频输出/DAC 音频输出
15	NC	NC	空
16	NC	NC	空

5、电气参数

环境温度 25℃，工作电压 DC3V

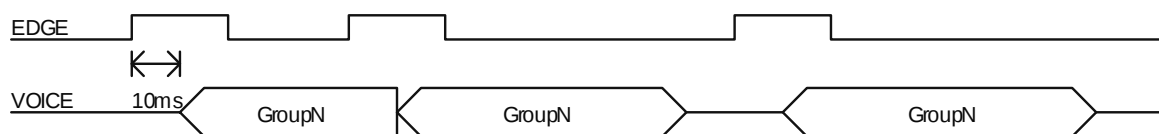
参数	标记	环境条件	最小值	典型值	最大值	单位
工作电压	V _{DD}	无	2.4	3	5.0	V
待机电流 (LDO ON)	ISB	VDD=3V，没负载	3	4	5	uA
待机电流 (LDO OFF)	ISB	VDD=3V，没负载	1	1.5	3	uA
工作电流	IOP	VDD=3V，没负载		600		uA
输出驱动电流	IOD	VDD=3V，VOUT=0.6V		10		mA
输出灌电流	IOS	VDD=3V，VOUT=2.4V		30		mA
PWM 驱动电流	IOD	VDD=3V，VOUT=1.5V		200		mA
PWM 灌电流	IOS	VDD=3V，VOUT=1.5V	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

6、控制模式

6.1、按键控制

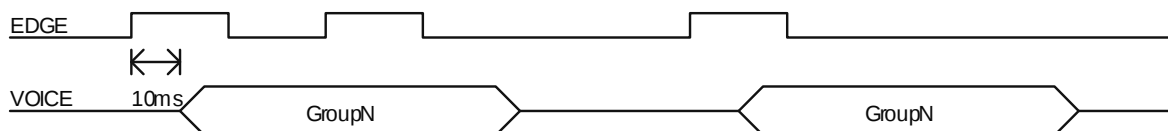
在按键控制模式下，任意控制端均可设置为脉冲可重复、脉冲不可重复、脉冲保持、脉冲不保持、电平保持可循环、电平保持不可循环、下一曲可循环等 7 种触发方式。

脉冲可重复



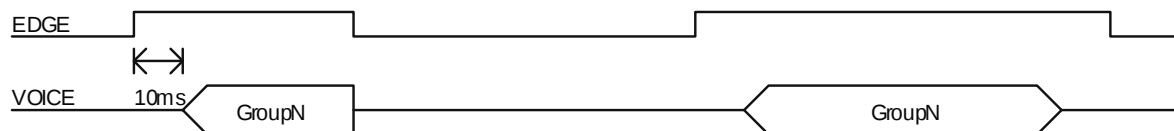
收到正脉冲信号后开始播放语音，在语音结束前如果还收到第二次正脉冲信号，则重新开始播放语音，在语音播放的过程中无收到正脉冲信号则播放完整段语音。

脉冲不可重复



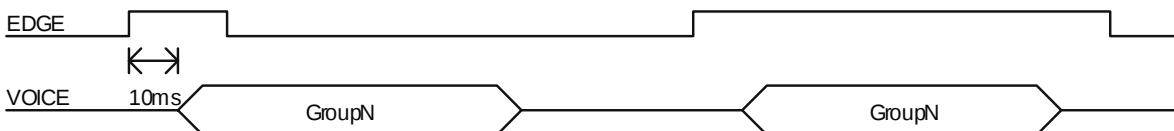
收到正脉冲信号后开始播放语音，在语音结束前如果再次收到正脉冲信号，则没有动作产生，在语音播放结束后再次收到正脉冲信号才会重新播放语音。

脉冲保持



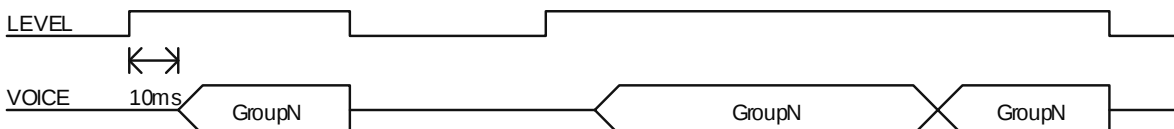
收到正脉冲信号后开始播放语音，正脉冲信号停止时，语音也停止播放。如果正脉冲信号一直保持，并且持续的时间超出语音播放的长度，则播放完语音即停止，不再继续播放语音。

脉冲不保持



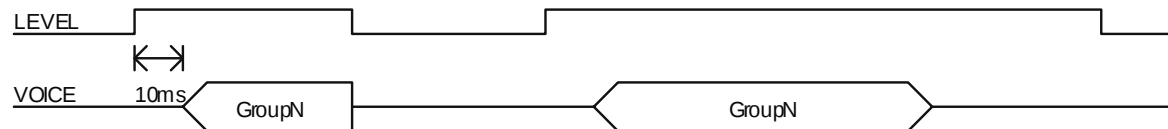
收到正脉冲信号后开始播放语音，正脉冲信号停止后语音继续播放完毕，如果正脉冲信号一直保持，并且持续的时间超出语音播放的长度，则播放完语音即停止，不再继续播放语音。

电平保持可循环



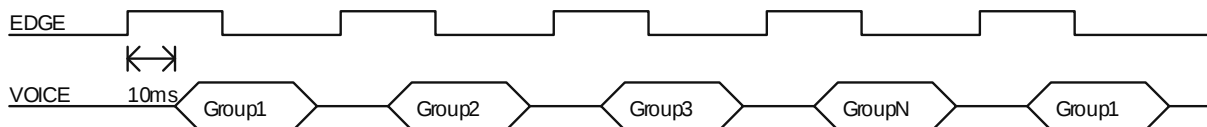
收到高电平后播放语音，高电平信号停止时，语音也停止播放。如果一直保持高电平信号，则会一直循环播放当前段语音语音。

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



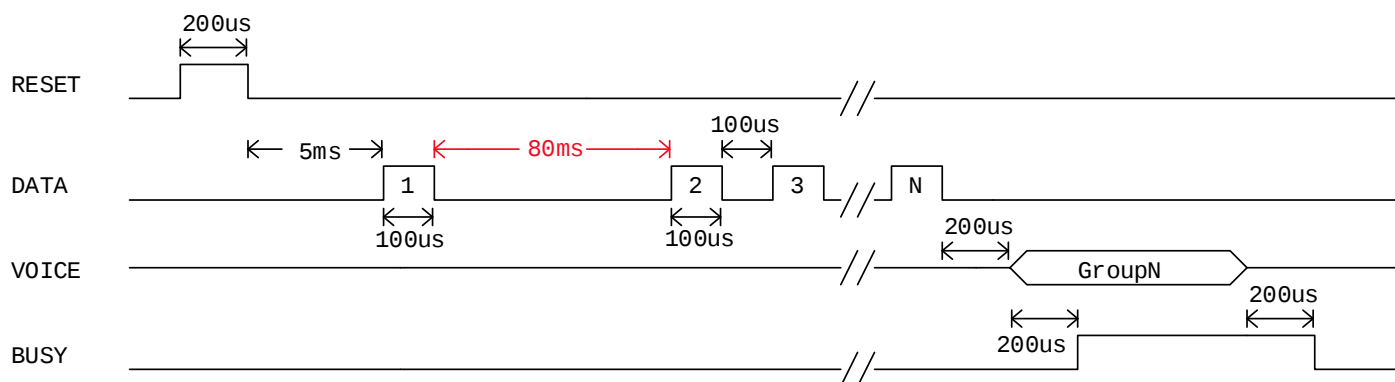
此触发方式仅限于在 P00 控制端口。收到正脉冲信号后开始播放第一段地址语音，再次收到正脉冲信号则播放第二段地址语音，当触发到最后一段后，再次触发则重新开始播放第一段语音，如此循环。

6.2、一线串口控制

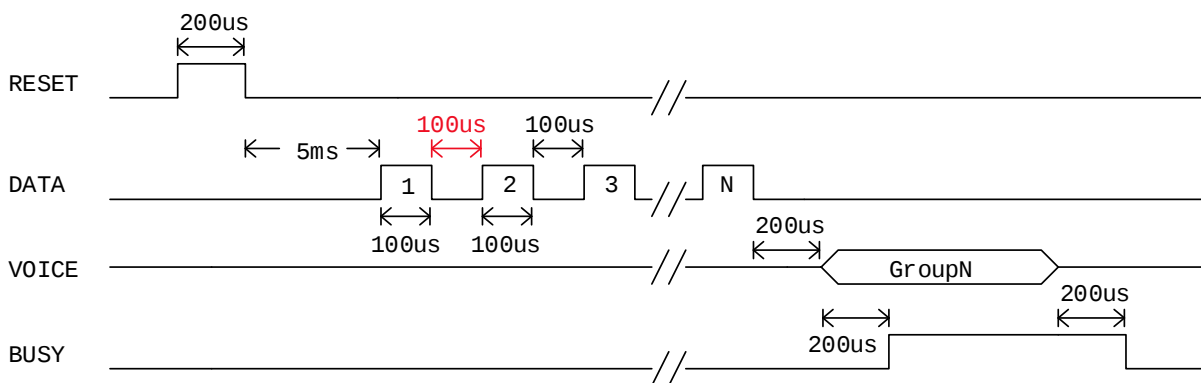
一线串口控制模式是通过在 DATA 线上发送不同的脉冲数量以达到控制语音地址的目的，该控制模式具有控制端口少，可控制语音地址数量多等优点。常用于 MCU 控制端口紧缺的场合。但一线串口控制模式也容易受到外界脉冲信号的干扰，故需要谨慎使用。

一线串口控制时序中，先发送 200us 的 RESET 信号（注：此时复位是软件复位，不是硬件复位），等待 5ms 后发送 DATA，DATA 中的第一个脉冲需要保持 100us 的高电平，等待 50ms（DAC 输出方式则等待 50ms，PWM 输出方式则等待 100us）后继续发送第二个脉冲，自第二个脉冲起，每个脉冲的高电平保持时间为 100us，且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。注意不同输出方式，其第一个脉冲的低电平宽度有所不同（时序图中红色部分）。其时序如下图所示。

DAC 输出方式时的一线串口控制时序:



PWM 输出方式时的一线串口控制时序:



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	61	60

6	62	61
7	63	62

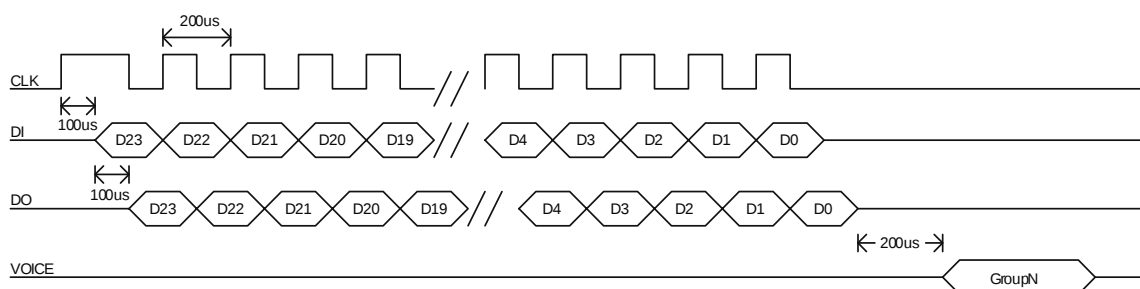
注意：为了防止误触发，在一线串口控制模式中，第一个地址的语音必须为静音。

6.3、两线串口控制

6.3.1、发送地址命令

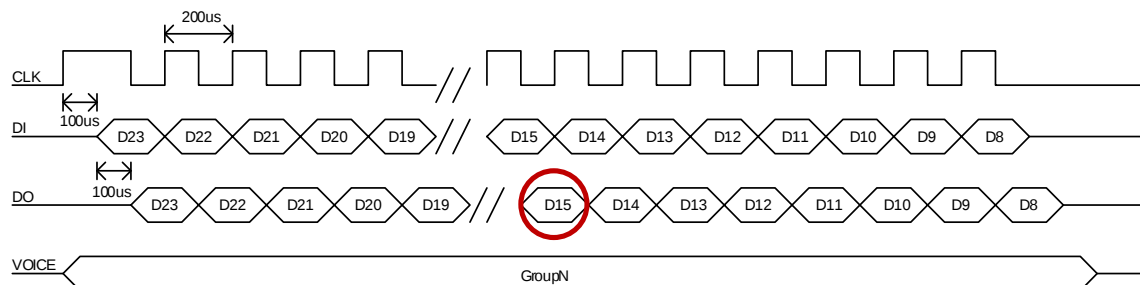
两线串口控制模式由 DI 及 CLK 发送数据信号控制语音地址，同时由 DO 返回相关操作数据。在发送 DI 前先将 CLK 拉高 100us 以唤醒芯片工作，继而发送周期为 200us 的 CLK 信号跟 DI，在接收到 DI 数据 100us 后 DO 返回 DI 所发送的数据。整个数据需要发送 24bit，数据发送完成后 200us，开始播放地址语音。

时钟周期的工作范围为 50us ~ 4000us。



6.3.2、读取芯片工作状态

发送 16bit 的数据 0x0022，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。



*注：在上电时，若 CLK 管脚有高脉冲，则进入烧写模式，芯片无法正常工作，须重新上电。使用 51 系列 MCU 的客户需注意！

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	0x0A0100	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

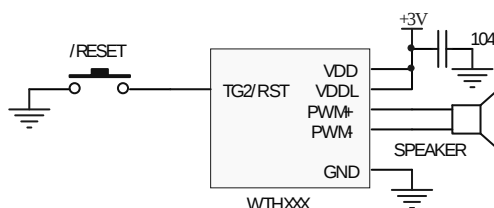
WTH040 在两线串口模式下最多可加载 256 段语音，其控制命令及语音地址对应关系如下。

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	0×182008	251
9	0×182028	252
10	0×182048	253
11	0×182068	254
12	0×182088	255

7、复位及输出状态

7.1、硬件复位

可通过 PC 软件设置 TG2 为硬件复位控制端口，负脉冲触发，保持 5ms 以上有效，此时复位为硬件复位，TG2 为低电平时，芯片停止工作；也可以通过 PC 软件设置 TG2 为软件复位控制端口，正脉冲触发，保持 100us 以上有效。此时复位为软件复位，仅清除寄存器的值，TG2 保持为高，芯片也正常工作。



7.2、输出状态

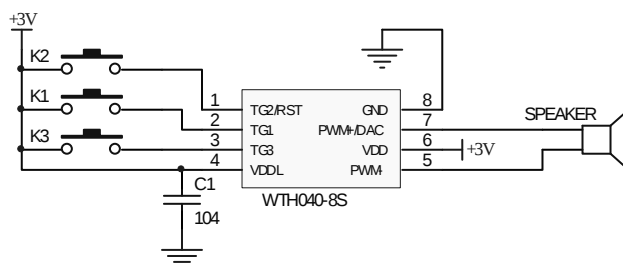
TG5、TG6、TG7、TG8 均可以设置为输出端口，其中可设置的状态如下

- 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

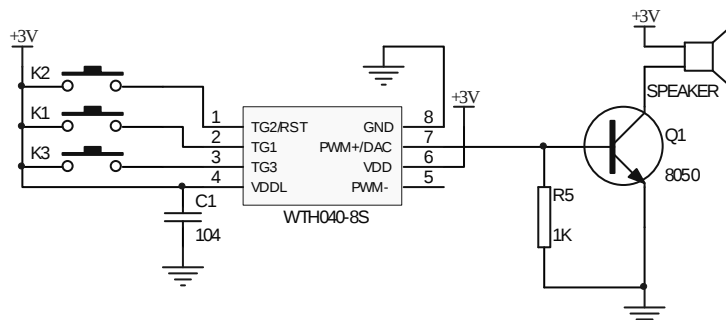
8、应用电路

8.1、按键控制模式应用电路

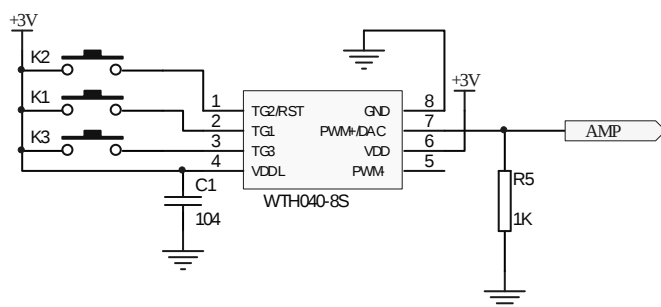
8.1.1、WTH040-8S/WTH040-8P按键控制（DC3V供电，PWM输出）



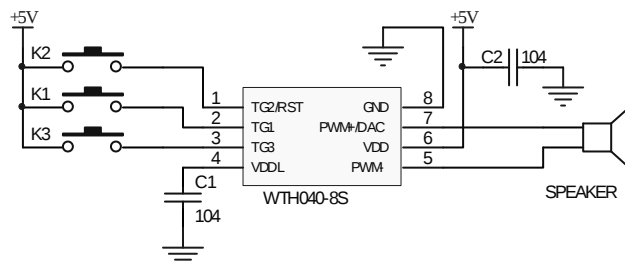
8.1.2、WTH040-8S/WTH040-8P按键模式（DC3V供电，DAC输出外接三极管）



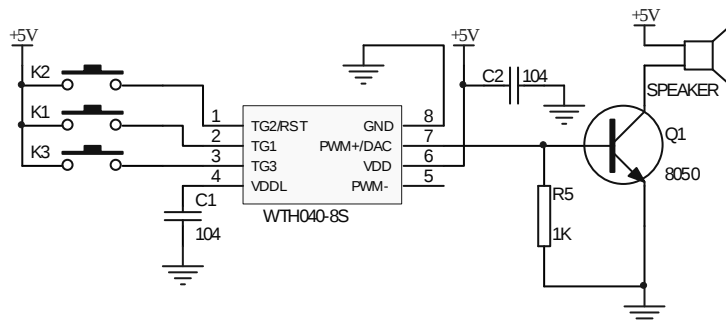
8.1.3、WTH040-8S/WTH040-8P按键模式 (DC3V供电 , DAC输出外接功放)



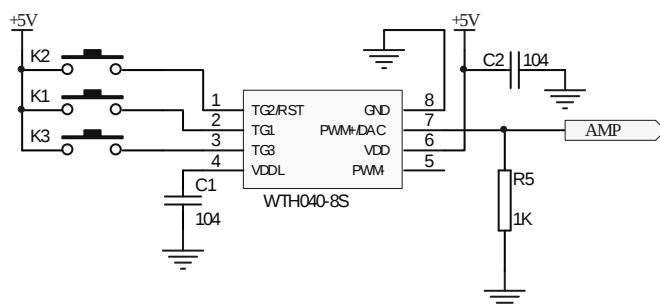
8.1.4、WTH040-8S/WTH040-8P按键模式 (DC5V供电 , PWM输出)



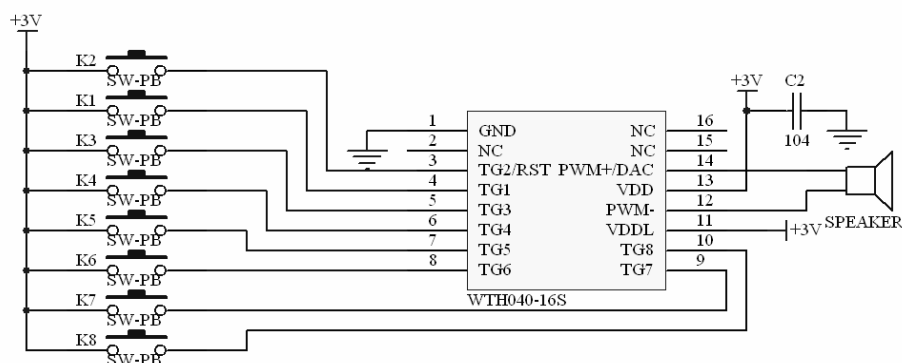
8.1.5、WTH040-8S/WTH040-8P按键模式 (DC5V供电 , DAC输出外接三极管)



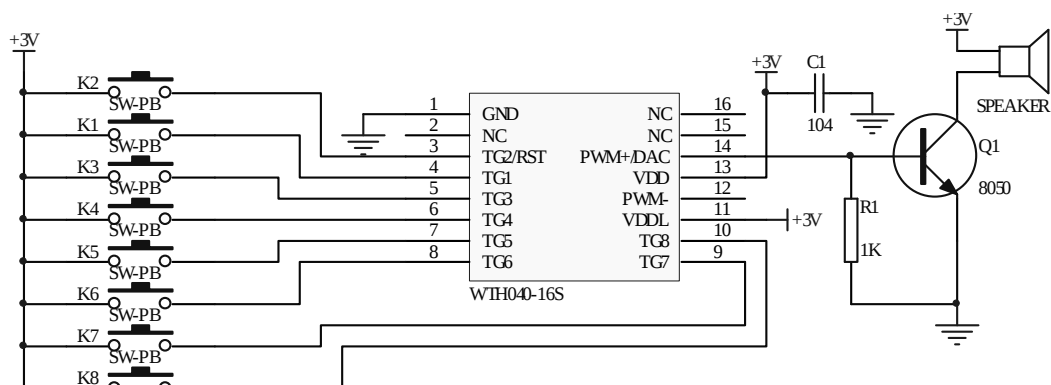
8.1.6、WTH040-8S/WTH040-8P按键模式 (DC5V供电 , DAC输出外接功放)



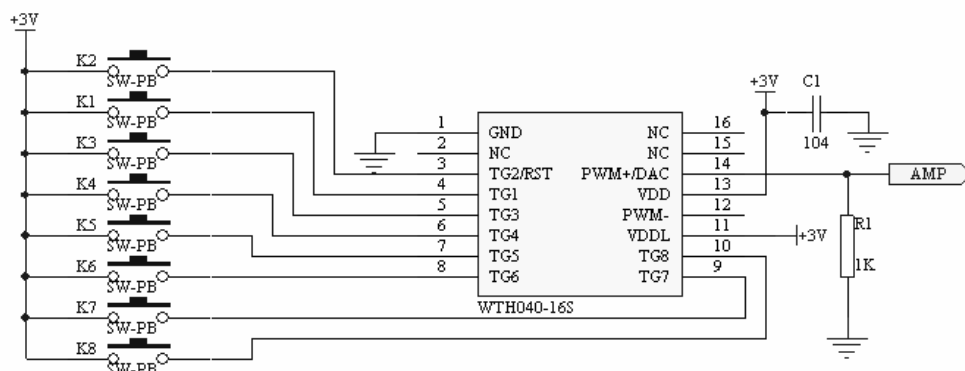
8.1.7、WTH040-16S按键模式 (DC3V供电, PWM输出)



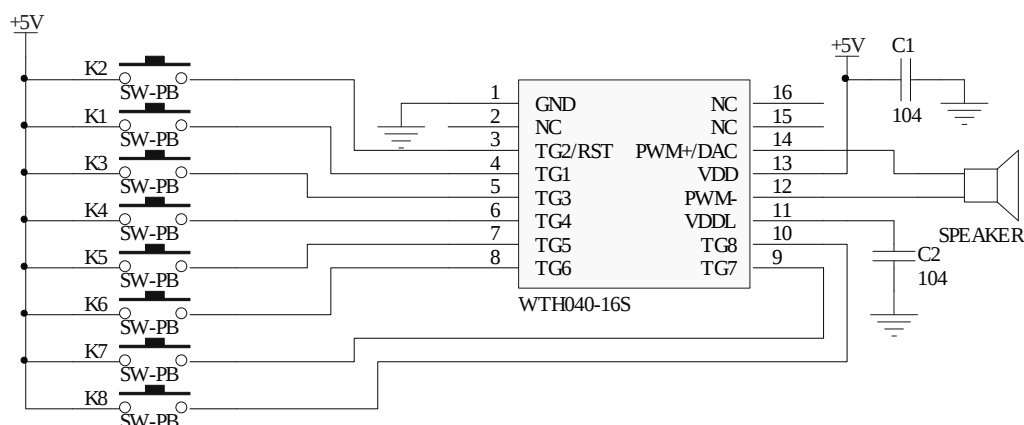
8.1.8、WTH040-16S按键模式 (DC3V供电, DAC输出外接三极管)



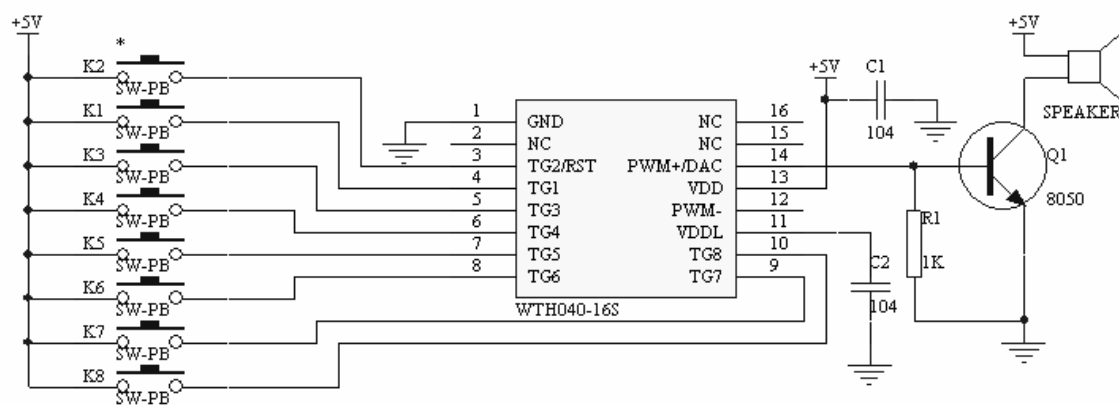
8.1.9、WTH040-16S按键模式 (DC3V供电, DAC输出外接功放)



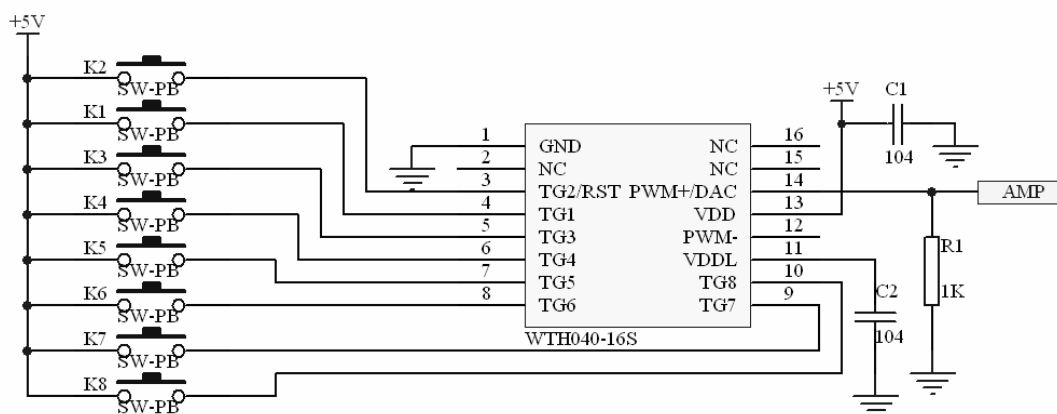
8.1.10、WTH040-16S按键模式 (DC5V供电, PWM输出)



8.1.11、WTH040-16S按键模式 (DC5V供电 , DAC输出外接三极管)



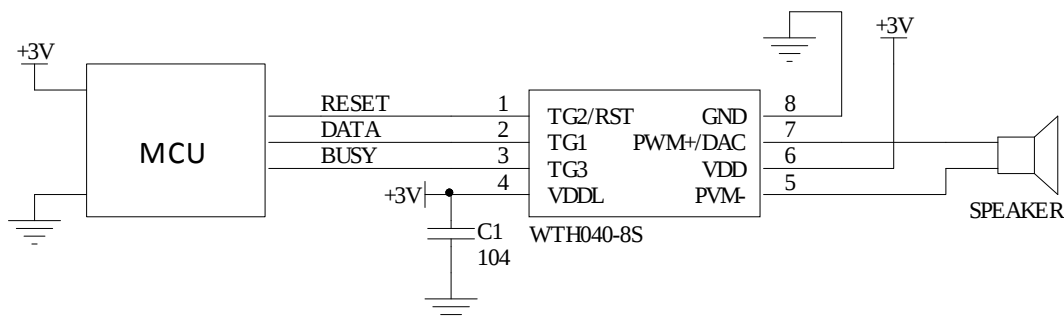
8.1.12、WTH040-16S按键模式 (DC5V供电 , DAC输出外接功放)



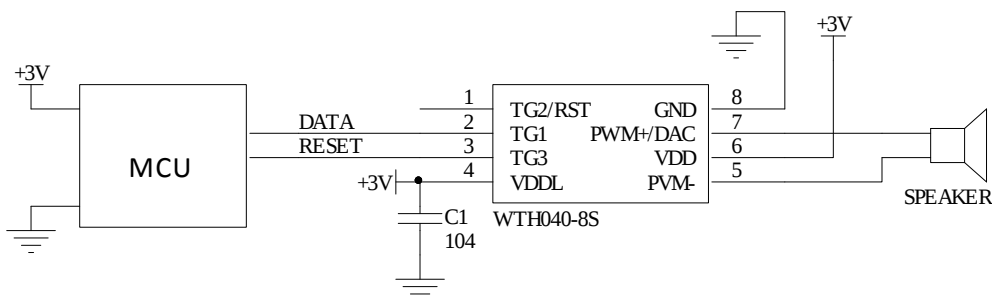
8.2、一线串口控制模式应用电路

8.2.1、WTH040-8S/WTH040-8P一线串口模式 (MCU为DC3V供电 , PWM输出)

- 语音地址少于 32 段（包括 32 段）时的应用电路

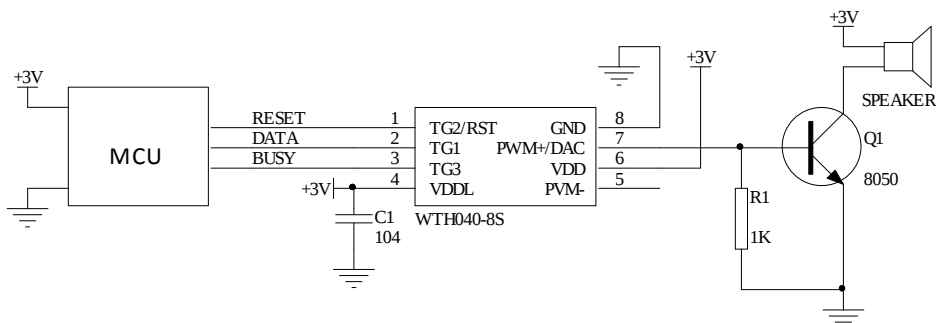


- 语音地址大于 32 段，少于 62 段（包括 62 段）时的应用电路

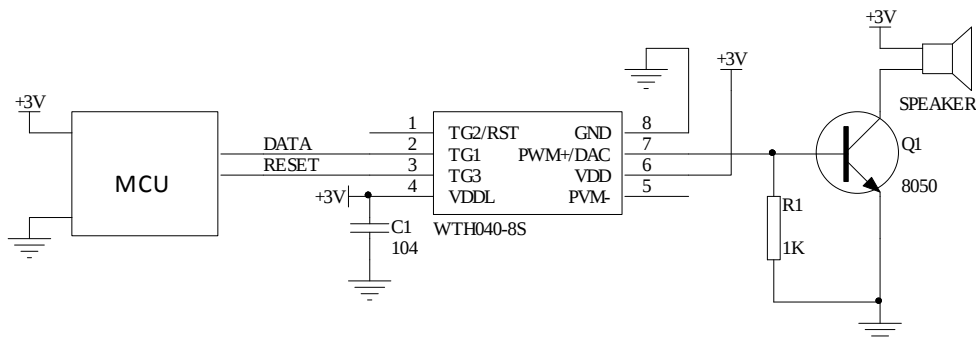


8.2.2、WTH040-8S/WTH040-8P一线串口模式（MCU为DC3V供电，DAC输出外接三极管）

- 语音地址少于 32 段（包括 32 段）时的应用电路

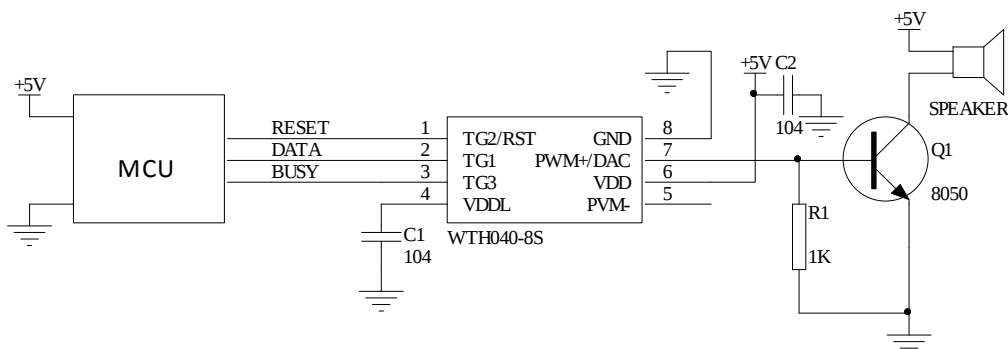


- 语音地址大于 32 段，少于 62 段（包括 62 段）时的应用电路

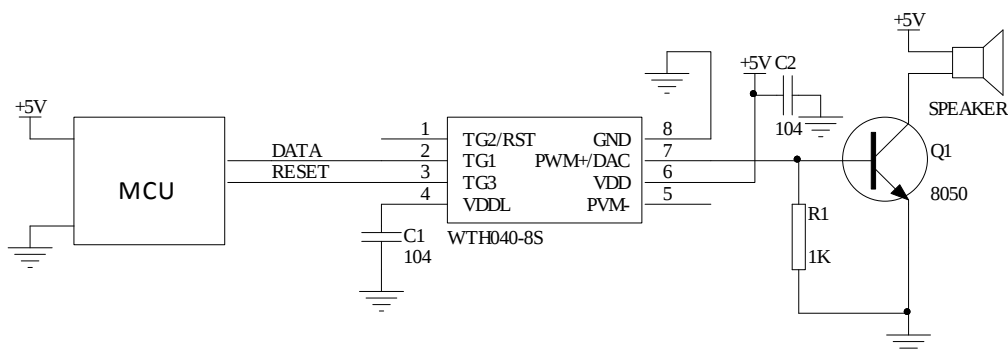


8.2.5、WTH040-8S/WTH040-8P一线串口模式（MCU为DC5V供电，DAC输出外接三极管）

- 语音地址少于 32 段（包括 32 段）时的应用电路

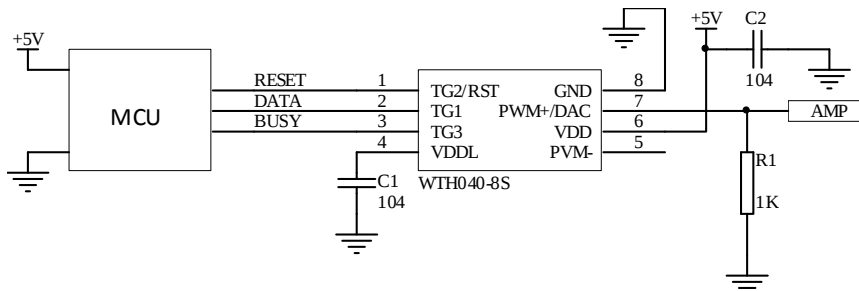


- 语音地址大于 32 段，少于 62 段（包括 62 段）时的应用电路

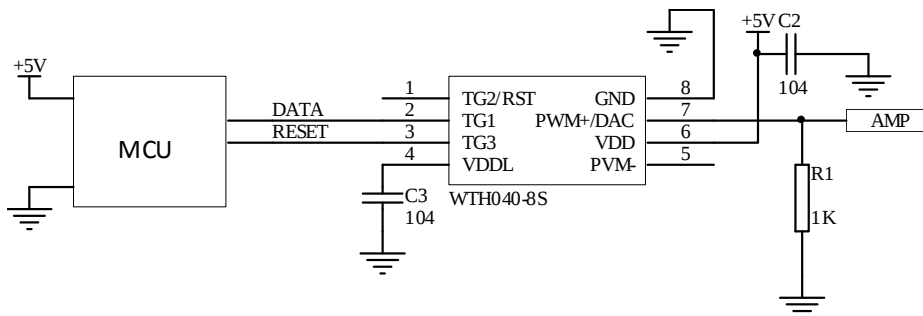


8.2.6、WTH040-8S/WTH040-8P一线串口模式（MCU为DC5V供电，DAC输出外接功放）

- 语音地址少于 32 段（包括 32 段）时的应用电路

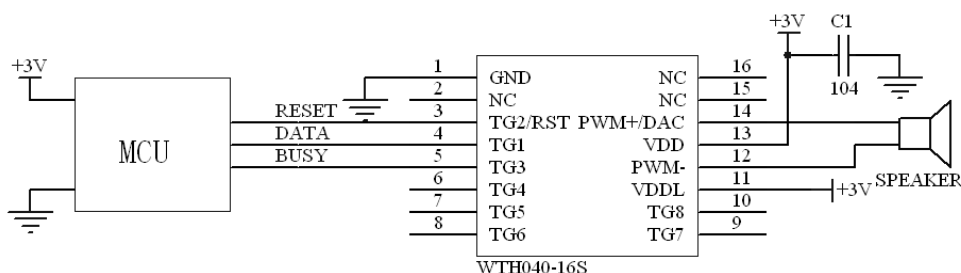


- 语音地址大于 32 段，少于 62 段（包括 62 段）时的应用电路

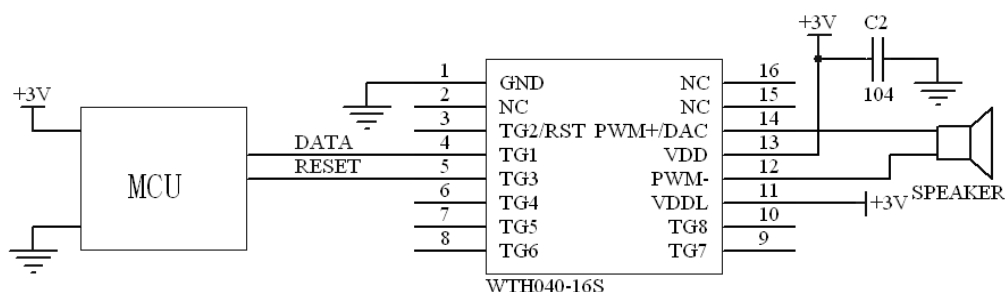


8.2.7、WTH040-16S一线串口模式（MCU为DC3V供电，PWM输出）

- 语音地址少于 32 段（包括 32 段）时的应用电路

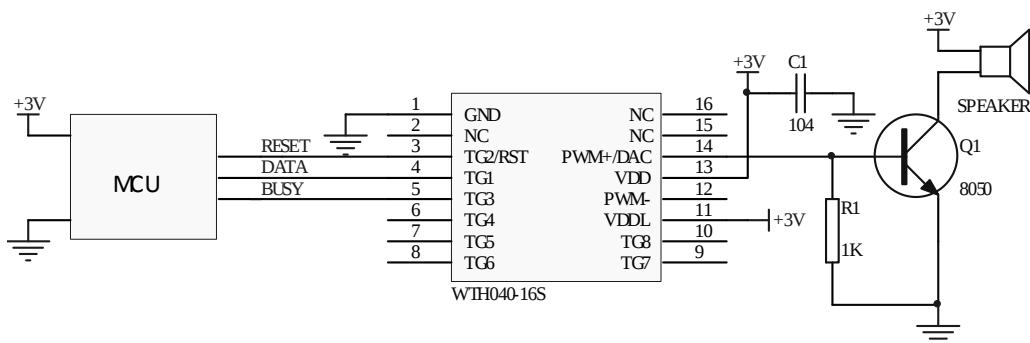


- 语音地址大于 32 段，少于 62 段（包括 62 段）时的应用电路

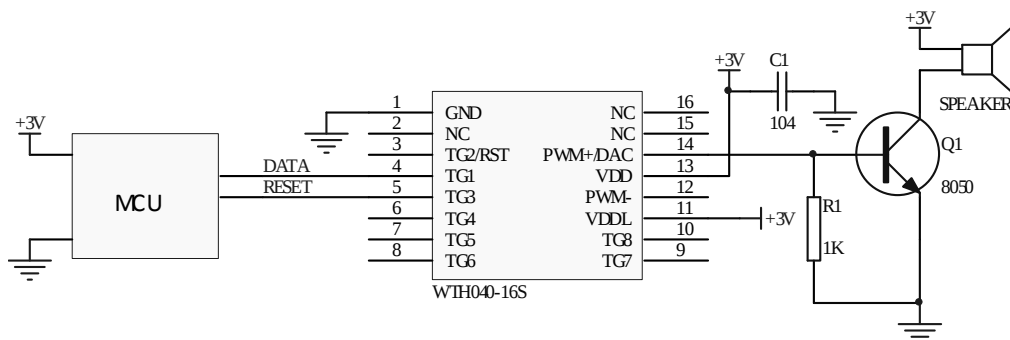


8.2.8、WTH040-16S一线串口模式（MCU为DC3V供电，DAC输出外接三极管）

- 语音地址少于 32 段（包括 32 段）时的应用电路

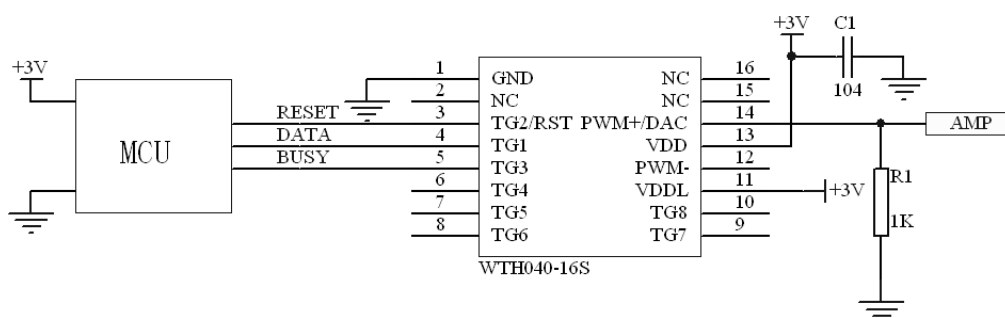


- 语音地址大于 32 段，少于 62 段（包括 62 段）时的应用电路

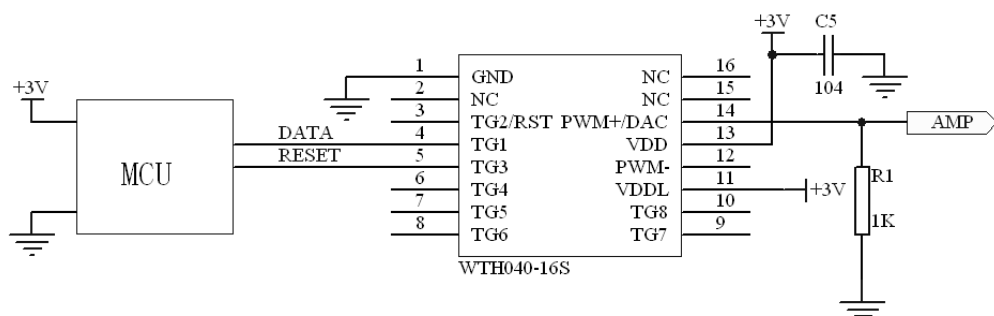


8.2.9、WTH040-16S一线串口模式（MCU为DC3V供电，DAC输出外接功放）

- 语音地址少于 32 段（包括 32 段）时的应用电路

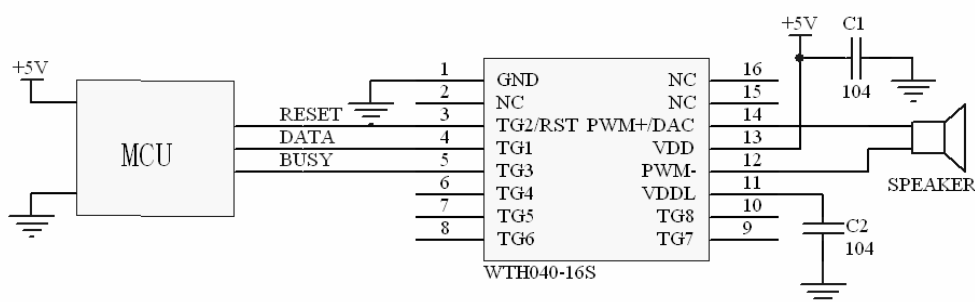


- 语音地址大于 32 段，少于 62 段（包括 62 段）时的应用电路

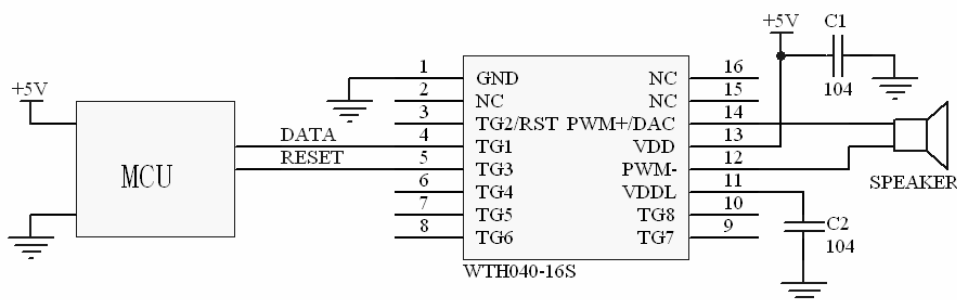


8.2.10、WTH040-16S一线串口模式（MCU为DC5V供电，PWM输出）

- 语音地址少于 32 段（包括 32 段）时的应用电路

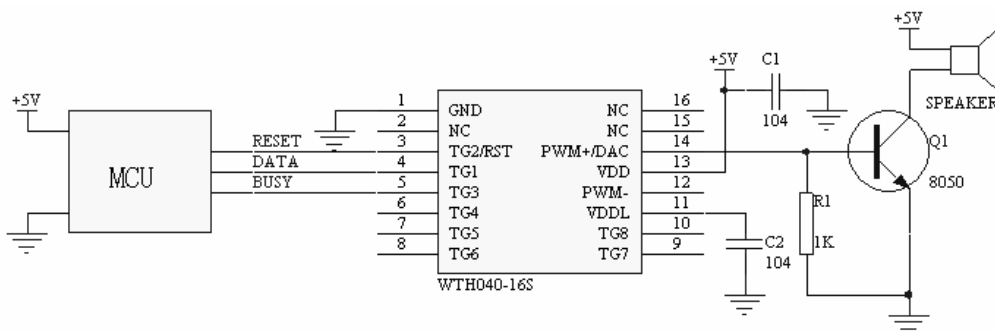


- 语音地址大于 32 段，少于 62 段（包括 62 段）时的应用电路

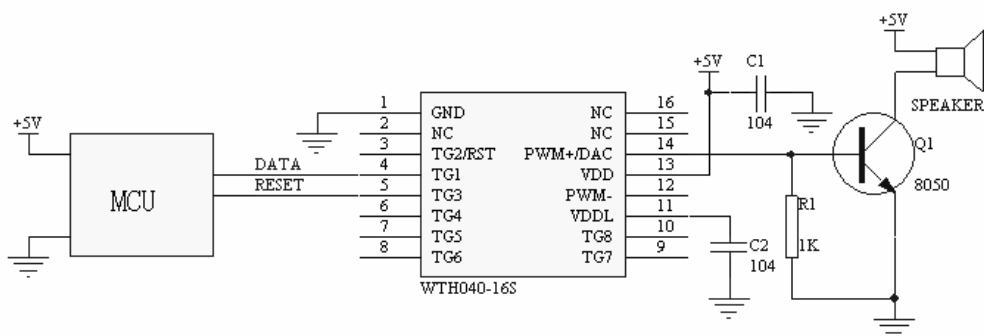


8.2.11、WTH040-16S一线串口模式（MCU为DC5V供电，DAC输出外接三极管）

- 语音地址少于 32 段（包括 32 段）时的应用电路

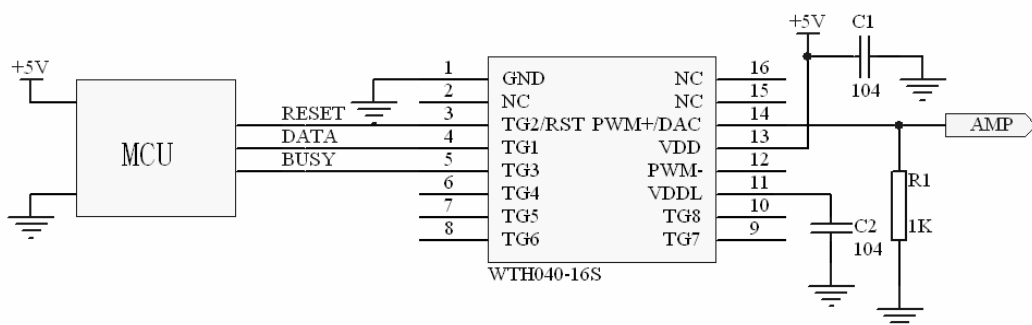


- 语音地址大于 32 段，少于 62 段（包括 62 段）时的应用电路

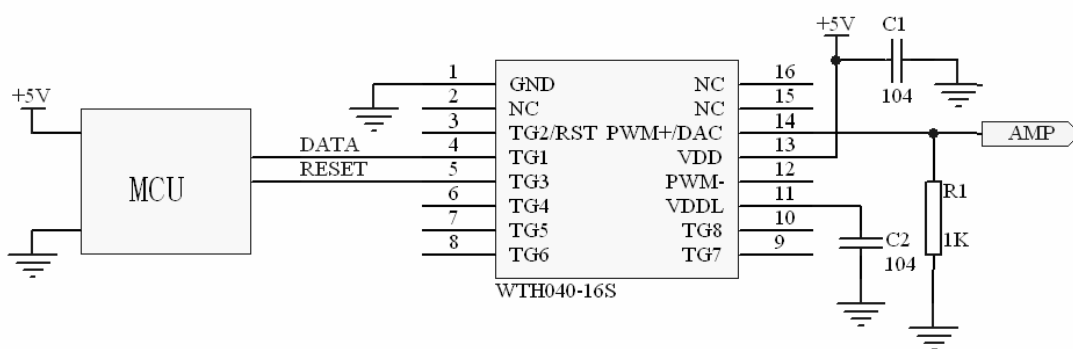


8.2.12、WTH040-16S一线串口模式（MCU为DC5V供电，DAC输出外接功放）

- 语音地址少于 32 段（包括 32 段）时的应用电路

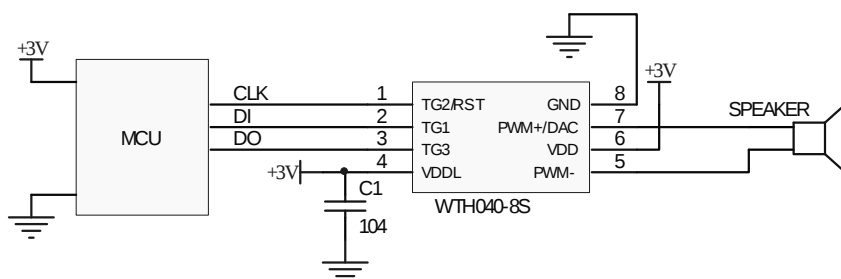


- 语音地址大于 32 段，少于 62 段（包括 62 段）时的应用电路

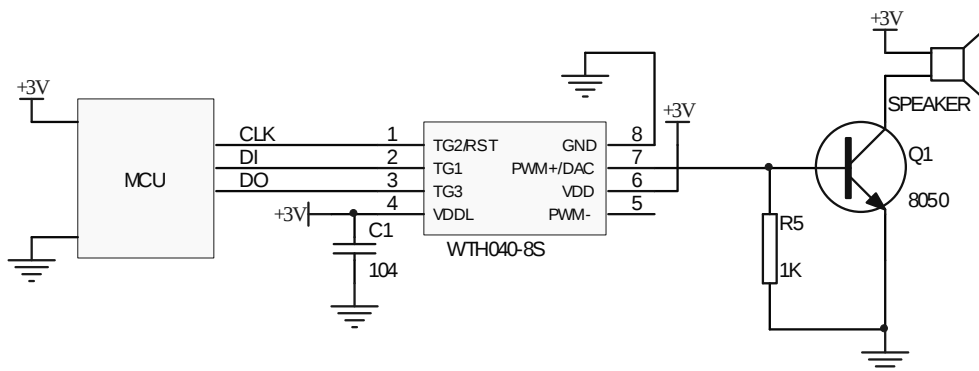


8.3、两线串口控制模式应用电路

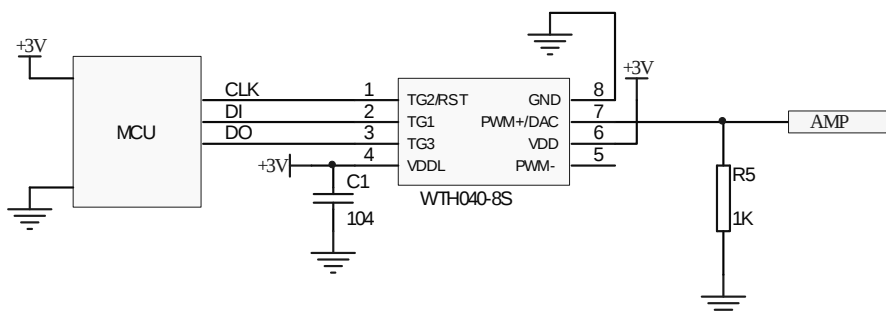
8.3.1、WTH040-8S/WTH040-8P两线串口模式 (MCU为DC3V供电，PWM输出)



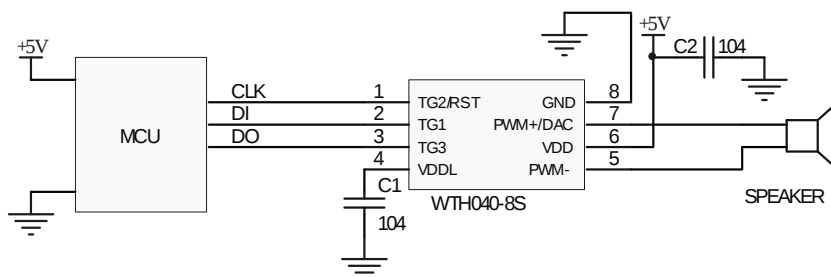
8.3.2、WTH040-8S/WTH040-8P两线串口模式 (MCU为DC3V供电，DAC输出外接三极管)



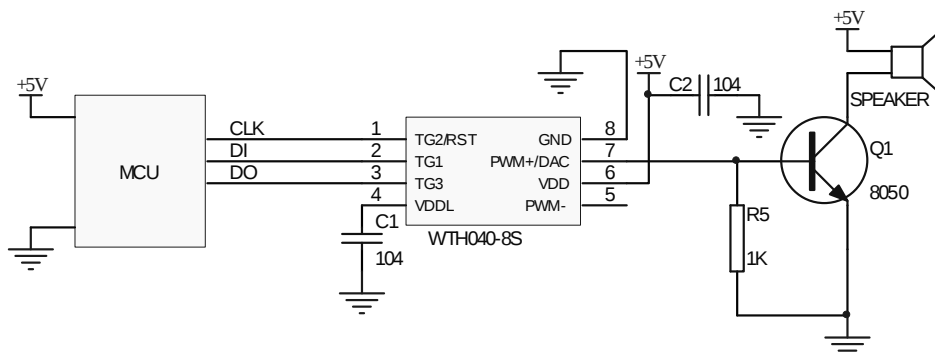
8.3.3、WTH040-8S/WTH040-8P两线串口模式（MCU为DC3V供电，DAC输出外接功放）



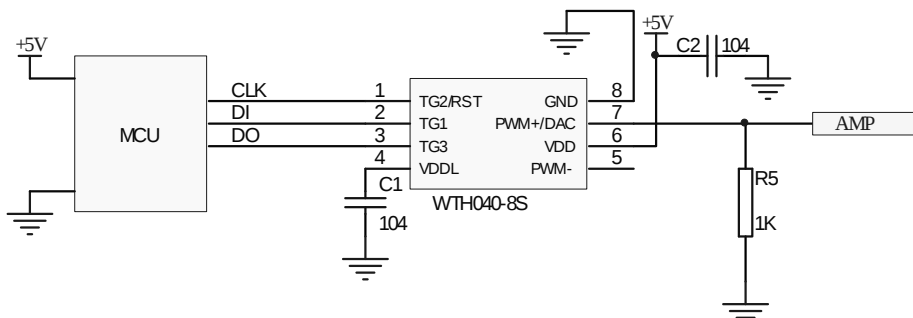
8.3.4、WTH040-8S/WTH040-8P两线串口模式（MCU为DC5V供电，PWM输出）



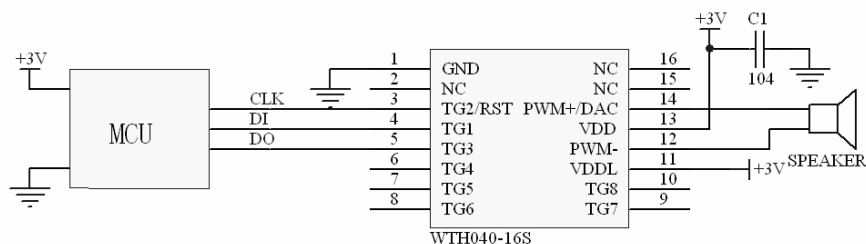
8.3.5、WTH040-8S/WTH040-8P两线串口模式（MCU为DC5V供电，DAC输出外接三极管）



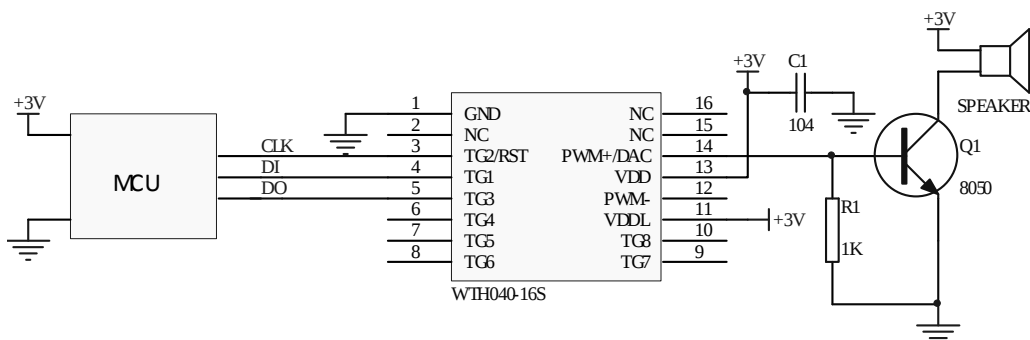
8.3.6、WTH040-8S/WTH040-8P两线串口模式（MCU为DC5V供电，DAC输出外接功放）



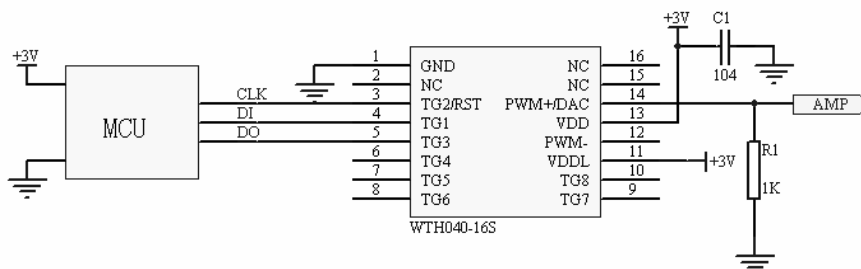
8.3.7、WTH040-16S两线串口模式（MCU为DC3V供电，PWM输出）



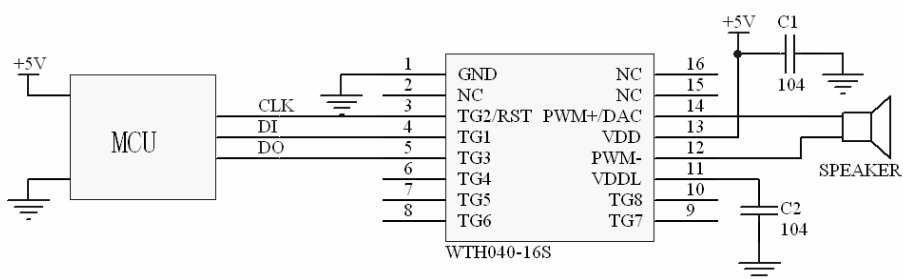
8.3.8、WTH040-16S两线串口模式（MCU为DC3V供电，DAC输出外接三极管）



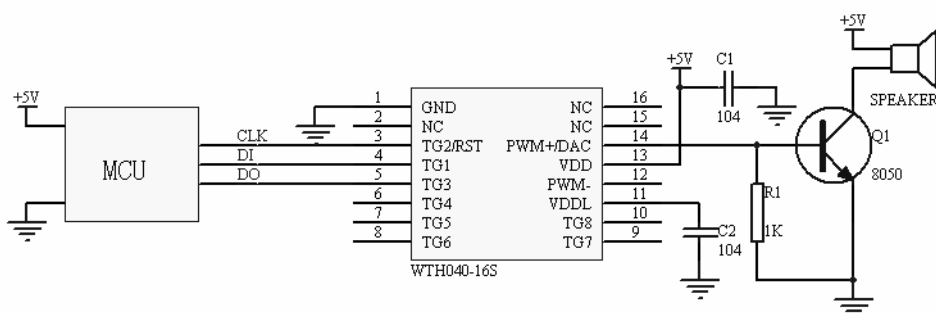
8.3.9、WTH040-16S两线串口模式（MCU为DC3V供电，DAC输出外接功放）



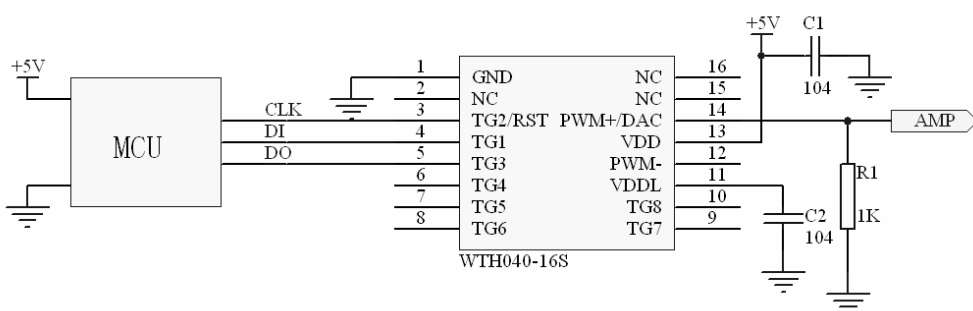
8.3.10、WTH040-16S两线串口模式（MCU为DC5V供电，PWM输出）



8.3.11、WTH040-16S两线串口模式（MCU为DC5V供电，DAC输出外接三极管）



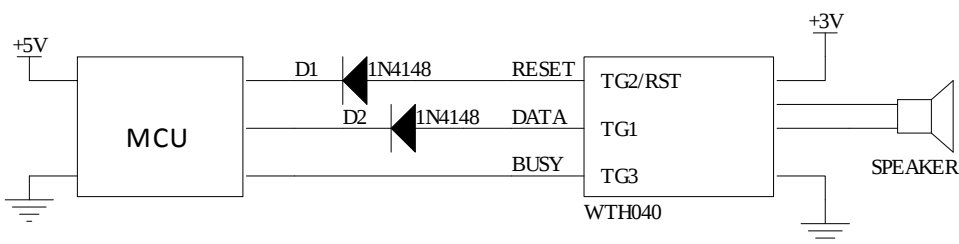
8.3.12、WTH040-16S两线串口模式（MCU为DC5V供电，DAC输出外接功放）



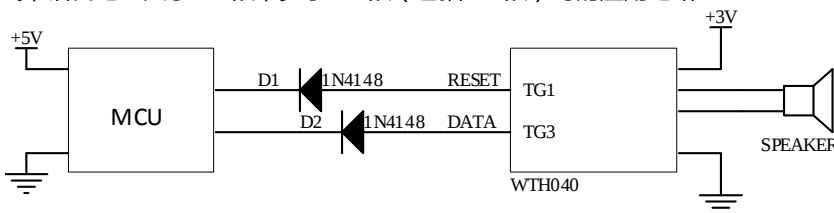
8.4、WTH系列语音芯片供电电压DC3V，MCU供电电压DC5V供电时通讯线的接法

以下的原理图仅提供 MCU 跟 WTH 系列语音芯片存在压差时的通讯线接法部分，具体的原理图还需参考前面的描述到的各种控制方式应用电路。

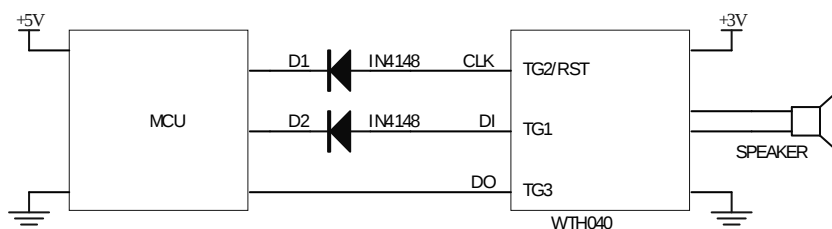
- 一线串口控制模式，语音地址少于 32 段（包括 32 段）时通讯线的接法



- 一线串口控制模式，语音地址大于 32 段，少于 62 段（包括 62 段）时的应用电路



➤ 两线串口控制模式时通讯线的接法



9、程序范例

9.1、一线串口控制程序范例(适应于PWM输出方式)

//MCU: STC10F04

//晶振: 11.0592MHz

#include "reg51.h" /* reg51 头文件 */

sbit WTH_REST = P2^1;

sbit WTH_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 10ms @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: WTH_1LinePWM
;功 能: WTH 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 WTH_1LinePWM(unsigned char cnt)
{
    if(cnt==0)
        return 0;
    WTH_REST = 1;
    Delay_10us(20); //复位高电平时间 200us
    WTH_REST = 0;
    Delay_1ms(5); //5ms

    while(cnt--)
    {
        WTH_PLUSE = 1;
        Delay_10us(10); //100us 高电平
        WTH_PLUSE = 0;
        Delay_10us(10); //100us 低电平
    }

    WTH_PLUSE = 0;
    return 1;
}
```

```
/*-----  
;模块名称:   Main  
;功    能:   Main 函数  
;入    参:   void  
;出    参:   void  
;-----*/  
void Main(void)  
{  
    WTH_REST = 0;  
    WTH_PLUSE = 0;  
  
    while(1)  
    {  
        if(KEY1 == 0)  
        {  
            WTH_1LinePWM(0x02);  
        }  
        Delay_10ms(30);  
    }  
}
```

9.2、一线串口控制程序范例(适应于DAC输出方式)

```
//MCU: STC10F04  
//晶振: 11.0592MHz  
  
#include "reg51.h"    /* reg51 头文件 */  
  
sbit WTH_REST = P2^1;  
sbit WTH_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 10ms @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: WTH_1LineDAC
;功    能: WTH 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 WTH_1LineDAC(unsigned char cnt)
{
    if(cnt==0)
        return 0;
    WTH_REST = 1;
    Delay_10us(20);    //复位高电平时间 200us
    WTH_REST = 0;
    Delay_1ms(5);      //5ms

    /***** 第一个脉冲 *****/
    WTH_PLUSE = 1;
```

```
Delay_10us(10);      //100us 高电平
WTH_PLUSE = 0;
Delay_10ms(8);        //80ms 低电平
/*****/

/*其他脉冲*/
while(--cnt)
{
    WTH_PLUSE = 1;
    Delay_10us(10); //100us 高电平
    WTH_PLUSE = 0;
    Delay_10us(10); //100us 低电平
}

WTH_PLUSE = 0;
return 1;
}

/*-----
;模块名称: Main
;功    能: Main 函数
;入    参: void
;出    参: void
;-----*/
void Main(void)
{
    WTH_REST = 0;
    WTH_PLUSE = 0;

    while(1)
    {
        if(KEY1 == 0)
        {
            WTH_1LineDAC(0x02);
        }

        Delay_10ms(30);
    }
}
```

9.3、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;
    TH1=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))
    {
```

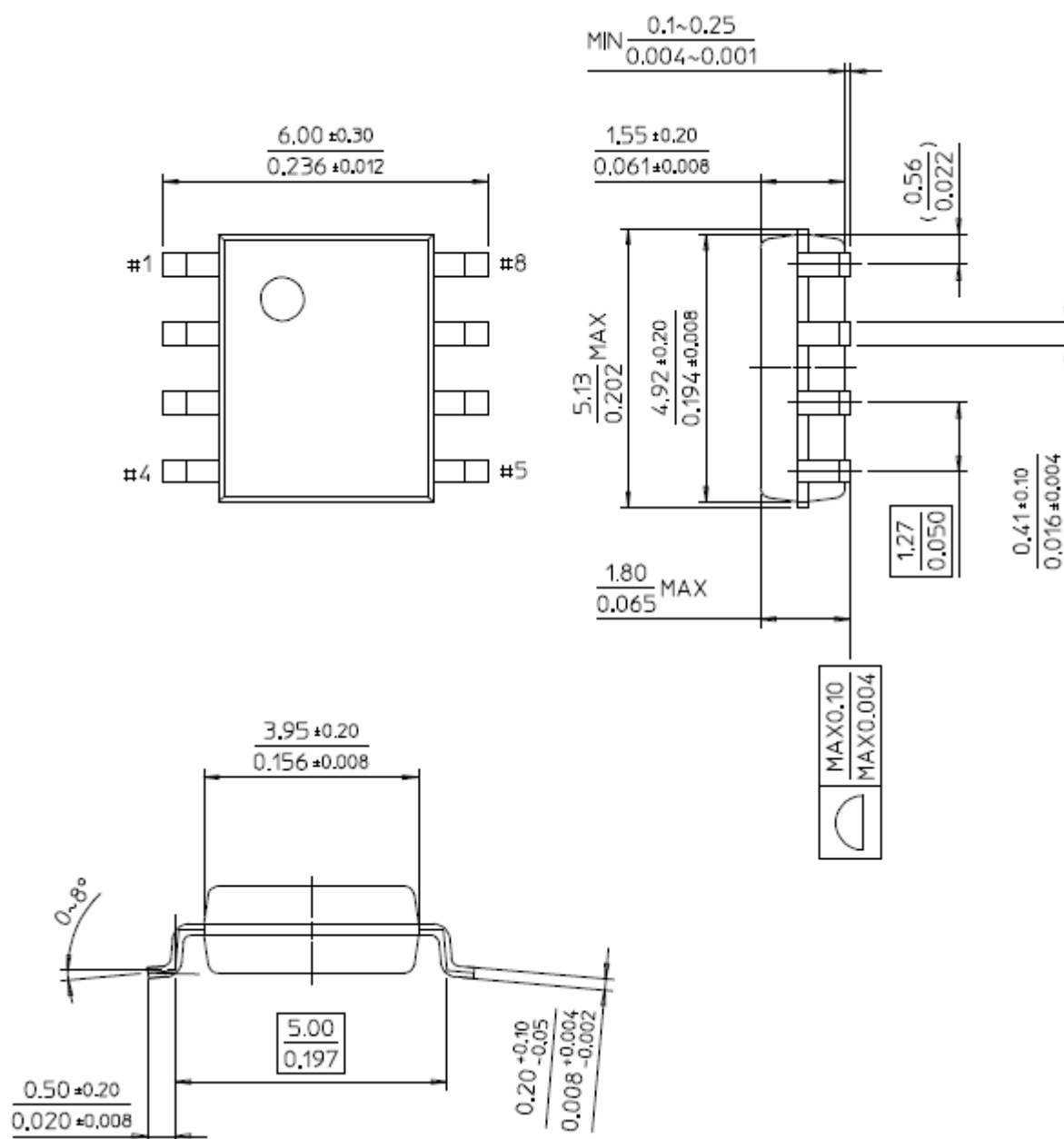


```
        table[n++] = SBUF;
        m++;
    }
    else if(a == 0Xed)
    {
        b = 0;
        n = 0;
        flag = 1;
    }
    RI = 0;
}
```

10、封装尺寸图

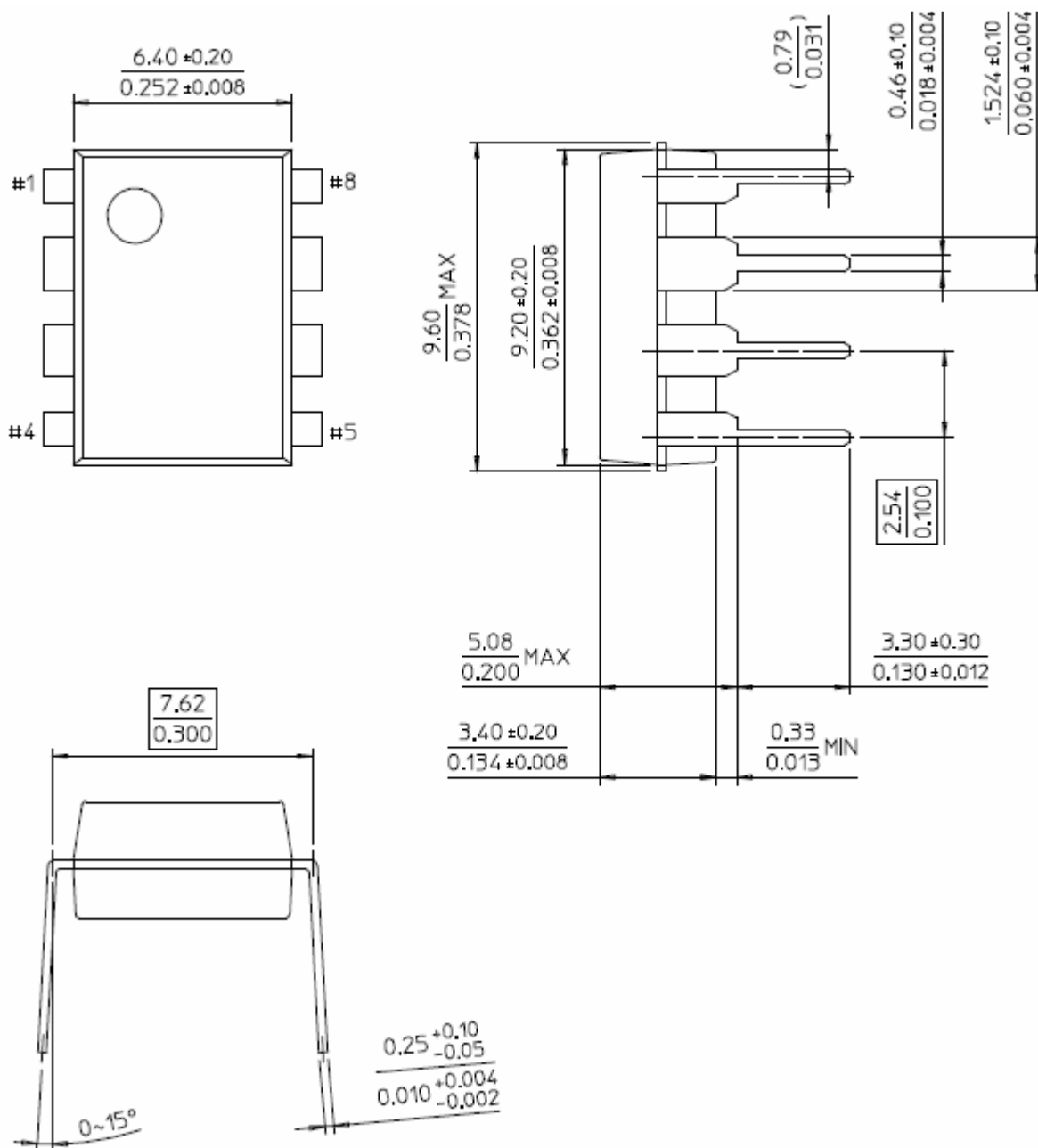
10.1、 WTH040-8S封装尺寸图

单位：mm



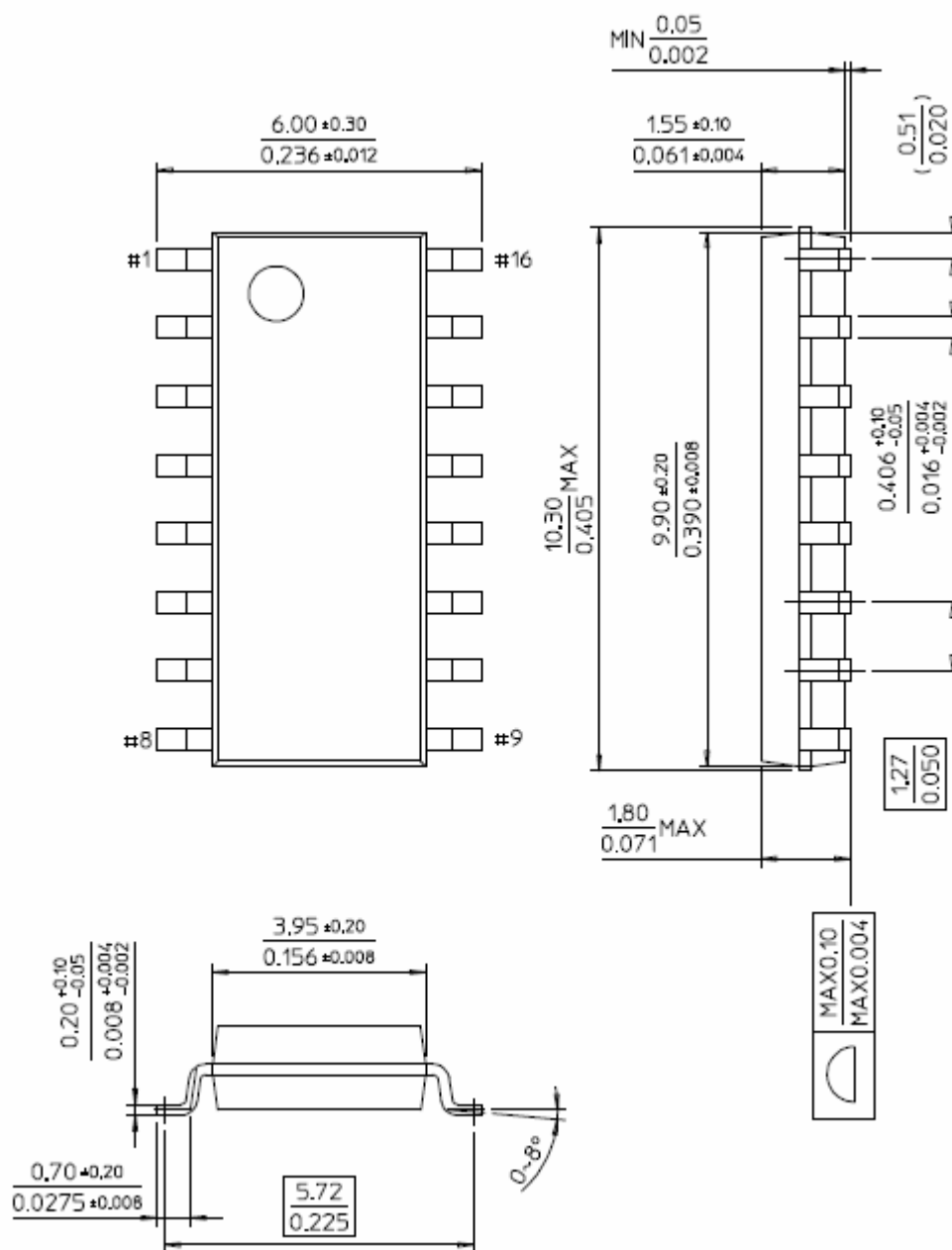
10.2、WTH040-8P封装尺寸图

单位：mm



10.3、WTH040-16S封装尺寸图

单位：mm





11、历史版本记录

版本号	日期	描述
V1.0	2010-12-07	原始版本
V1.1	2010-12-27	修改 WTH040-8S 管脚定义及其相关描述和原理图
V1.2	2011-04-01	增加部分应用电路及注意事项
V1.3	2011-6-14	增加 WTH040 上电注意事项和休眠指令
V1.4	2011-10-08	修改一线串口控制时序、参考例程和参考应用电路
V1.5	2011-10-19	修改应用电路图和参考例程
V1.6	2011-10-19	修正管脚图和管脚功能列表不对应的错误



广州唯创电子有限公司（原广州唯创科技有限公司）1999 年创立于广州市天河区，是一家集语音芯片研发、语音产品方案设计、语音产品生产、语音编辑上位机软件开发的高新技术公司。业务范围涉及汽车电子、多媒体、家居防盗、通信、家电、医疗器械、工业自动化控制、玩具及互动消费类产品等领域。团队有着卓越的 IC 软、硬件开发实力和设计经验，秉持着「积极创新、勇于开拓、满足顾客、团队合作」的理念，力争打造“语音业界”的领导品牌。

唯创主要生产 WTV 系列语音芯片、WT588D 系列语音芯片、WTH 系列语音芯片、WTB 系列语音芯片、WT5S 系列语音芯片、WTE 系列语音芯片、WTR 可录音系列语音芯片、WTM 系列高音质语音应用模块、WTF 系列的高性价比长时间播放模块，及特约代理的 APLUS 系列语音芯片、ISD 全系列可录放语音芯片等。率先提供最完备、多元化的客需解决方案，节约研发成本，缩短研发周期，使产品在最短的时间内成熟上市。在汽车电子及特种车领域，自主研发的公交车报站器在国内有着很好的市场口碑，为叉车使用安全而开发的叉车超速报警器是国内第一家研发此类产品并大量生产的企业。

唯创坚持“以人为本，不断进行核心技术创新，优良的售后技术跟踪服务”的经营策略，使得唯创能傲立于语音产品行业。WTV 系列语音芯片、WT588D 系列语音芯片、WTR 可录音系列语音芯片、WTM 系列高音质语音应用模块、WTF 系列的高性价比长时间播放模块等都是唯创的自主品牌，具有很强的市场竞争优势。产品、模块、编辑软件等的人性化设计，使得客户的使用更方便。于 2006 年新成立的北京唯创虹泰分公司主要以销售完整的方案及成熟产品为宗旨，以便于为国内北方客户提供更好的服务。

唯创持续在研发与技术升级领域大力投资，每年平均提拨超过 20% 的营业额作为研发经费，在我们的研发团队中，有超过 90% 员工钻研技术及产品发展。并与同行业大厂合作，勇于迈出下一个高峰。

总公司名称：广州市唯创电子有限公司

电话：020-85638557 85638660 38357061 38055581

技术支持 E-mail：sos30@1999c.com

地址：广东省广州市天河区棠东东路 55 号 3 楼

传真：020-85638637

网址：<http://www.w1999c.com>

分公司名称：北京唯创虹泰科技有限公司

电话：010-89756745

E-mail：BHL8664@163.com

地址：北京昌平区立汤路 186 号龙德紫金 3#902 室

传真：010-89750195

网址：www.wcht1998.com.cn

广州唯创电子有限公司深圳办事处

移动电话：0755-83555462 36956575 83044339

业务支持 E-mail：sos@1999c.com

地址：深圳市福田区福华路 29 号（京海花园）7 层 7E

传真：0755-83044339