

广州唯创电子有限公司

Shenzhen Waytronic Electronic Co., Ltd

# WT588F02BP-14S

## 大功率语音芯片说明书

版本号: V1.02

**免责声明:**

广州唯创电子有限公司申明: 说明书以官网资料为准, 如若资料内容有更新, 不会一一进行通知。如若使用 IC 时导致侵犯到第三方专利或其他权利, 不承担任何责任。如若使用我司 IC, 在航空卫星军事设备, 人身安全等领域, 造成了重大财产损失或生命伤害, 甚至生命死亡, 我司不承担任何责任。



## 目 录

1. 概述.....	3
2. 功能简述.....	3
3. 管脚描述.....	4
3.1. 脚位定义如下图: .....	4
3.2. 管脚分布图.....	4
4. 极限参数.....	5
5. 电气特性.....	5
6. 一线串口通讯.....	5
6.1. 管脚分配.....	5
6.2. 语音地址对应关系.....	6
6.3. 语音及命令码对应表.....	6
6.4. 一线串口时序图.....	7
7. 两线串口通讯.....	8
7.1. 管脚分配.....	8
7.2. 语音地址对应关系.....	8
7.3. 语音及命令码对应表.....	9
7.4. 两线串口时序图.....	9
8. 程序范例.....	11
8.1. 一线串口单字节控制程序.....	11
8.2. 一线串口双字节控制程序.....	12
8.3. 一线串口单字节连码示例(F3 + 01 + F3 + 02).....	13
8.4. 一线串口双字节连码示例(FFF3 + 0001 + FFF3 + 0002).....	13
8.5. 二线串口单字节控制程序.....	14
8.6. 二线串口双字节控制程序.....	15
8.7. 二线串口单字节连码示例(F3 + 01 + F3 + 02).....	16
8.8. 二线串口双字节连码示例(FFF3 + 0001 + FFF3 + 0002).....	16
9. 芯片应用原理图.....	18
9.1. 一线应用原理图.....	18
9.2. 两线应用原理图.....	18
9.3. 相关参数参考.....	18
9.4. 相关规范说明.....	19
10. 封装管脚图.....	21
11. 客户在线下载器.....	22
11.1. 下载原理图.....	23
11.2. 下载器使用说明.....	23
12. 关于芯片的静电保护.....	24
12.1. 静电防护策略分析.....	24
12.2. 可以尝试使用以下防静电电路.....	25

说明书以官网和业务员提供为准



## 版本记录

版本号	修改说明	修改日期
V1.00	原始版本	2023-09-11
V1.01	增加待机电流（ISB）参数	2023-10-07
V1.02	修改原理图供电范围，增加供电范围描述，去除不必要说明	2023-10-08





## 1. 概述

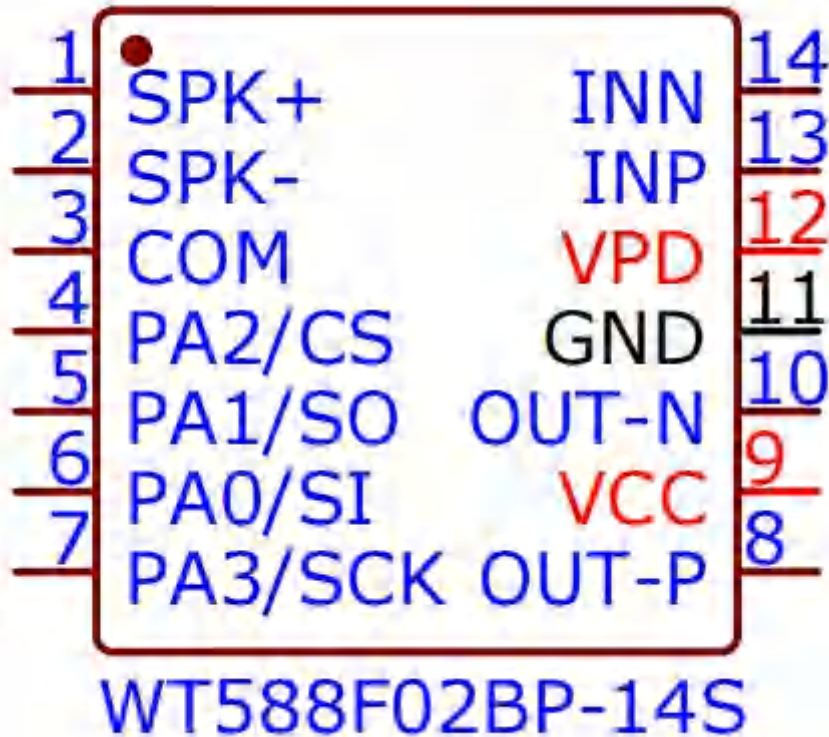
WT588F02BP-14S 是广州唯创电子有限公司研发的一款大功率语音芯片，其音频输出类型为 PWM，且内置 D 类音频功率放电路，可直接推 4 欧 3W 喇叭。其输出无需 LC 滤波也可正常工作。在全带宽范围内降低了 EMI 干扰，最大限度地减少对其他部件的影响。目前 WT588F02BP-14S，较高音质最大可以支持存放 170 秒语音内容（若客户对音质没有要求，最大可以存放 320 秒的语音内容）。WT588F02BP-14S 区别于传统语音芯片，它可以通过配套下载器实现在线更换芯片内部语音内容；芯片支持一线、两线串口通讯。

## 2. 功能简述

1. 工作电压 2.5V~5.5V（为了使功放能正常工作建议最低工作电压不低于 3.0V）；
2. 音频输出方式：16bit PWM；
3. 可直接驱动 4R 3W@5V 喇叭；
4. 支持 6K~32KHz 采样率的音频播放；
5. 客户可以通过配套下载器更换芯片内部语音内容；
6. 支持一线串口、两线串口；
7. 最大支持 1000 段地址语音；
8. 芯片内置 220KB 的 Flash，最多可放 170 秒高品质语音。
9. Flash 型语音芯片，芯片主控程序和 Flash 数据均可重复烧写。
10. 芯片上电初始化时间大概为 200ms
11. 芯片播放结束后，且 IO 口（DATA 和 CLK）保持电平稳定（高低电平皆可）1S 后，芯片进入休眠
12. 芯片带 D 类音频功率放电路，效率高达 88%，无需滤波器；
13. 内置短路保护、过流保护、过热保护；
14. 优异的全带宽 EMI 抑制能力
15. 优异的“上电/掉电”噪声抑制
16. 注意：芯片 PA2 脚平时为低，播放为高

### 3. 管脚描述

3.1. 脚位定义如下图：



### 3.2. 管脚分布图

Pad Name	Pad No.	ATTR.	Description 描述
SPK+	1	OUT	外接喇叭
SPK-	2	OUT	外接喇叭
COM	3	Bypass	可以接电容到地（具体参考后续原理图说明），减少高频噪音
PA2	4	I/O	忙信号输出/下载器烧写口
PA1	5	I/O	两线串口时钟信号输入端/一线串口数据信号输入端/下载器烧写口
PA0	6	I/O	两线串口数据信号输入端/下载器烧写口
PA3	7	I/O	下载器烧写口
OUT-P	8	OUT	音频信号输出
VCC	9	Power	电源正极
OUT-N	10	OUT	音频信号输出
GND	11	Power	电源负极
VPD	12	Power	内部功放电源
INP	13	IN	音频信号输入
INN	14	IN	音频信号输入

注意：请勿给 PA2 添加上下拉，会有几率导致 IC 进入非工作模式。



## 4. 极限参数

标识	范围值	单位
VCC~GND 电源电压	-0.5 ~ +5.5	V
Vin 输入电压	GND-0.3 < Vin < VDD+0.5	V
Vout 输出电压	GND < 0.3V ~ VDD+0.3	V
Top 工作温度	-20 ~ +85	°C
储存温度	-50 ~ +100	°C

备注：样品在实验室测试的结果，芯片在-40°C~+85°C下，能够正常工作。

## 5. 电气特性

VCC=3.0V, Ta=25°C

参数	象征	最低限度	典型	最大限度	测试条件
工作电压	VCC	2.5V		5.5V	
振荡频率	Fbank0	4.096MHz±3%		8.192MHz±3%	
RC 振荡器频率	Frc 1		65.536MHz±3%		
工作电流	IOP		10 mA		空载
IO 口逻辑电平 (H)	VIH	0.8 VCC			
IO 口逻辑电平 (L)	VIL			0.2 VCC	
IO 口输出电平 (H)	VOH	0.95 VCC			空载
IO 口输出电平 (L)	VOL			0.02 VCC	空载
IO 口驱动电流	IOH		16 mA		IO 为强驱动时, Vout=VCC-0.4V
待机电流	ISB		5uA		

注意：芯片在正常播放以及初始化时，电源不要产生超过 1.3V 的电源波动。

## 6. 一线串口通讯

一线串口模式可以利用 MCU 通过 GPIO 作为 DATA 线给 WT588F 系列语音芯片发送数据以达到控制的目的。可以实现控制语音播放、停止、循环等。一线串口控制只需要占用一个 IO 口，在 DATA 线电平本身为高的时候，单字节通讯时，一条指令时间为  $10.2ms \{ (5ms + (0.1ms + 0.3ms) * 8) + 2ms = 10.2ms \}$ ；双字节通讯时，一条指令时间为单字节发码的两倍 20.4ms ( $10.2ms * 2 = 20.4ms$ )。详细可见下面介绍。

注意：芯片 PA2 脚平时为低，播放为高

### 6.1. 管脚分配

封装形式	管脚	
	PA1	PA2
SOP14	DATA	BUSY



## 6.2. 语音地址对应关系

注：地址小于 224 段默认为单字节发码，大于等于 224 段为双字节发码。

数据（十六进制）	功能
00H（单字节）/0000H（双字节）	播放第 0 段语音
01H（单字节）/0001H（双字节）	播放第 1 段语音
02H（单字节）/0002H（双字节）	播放第 2 段语音
.....	
DFH（单字节）/00DFH（双字节）	播放第 223 段语音
00E0H（双字节）	播放第 224 段语音
.....	
03E5H（双字节）	播放第 997 段语音
03E6H（双字节）	播放第 998 段语音
03E7H（双字节）	播放第 999 段语音

注意：如要播放该地址语音，只要发送该地址就能自动播放该地址语音，一条地址指令字节间间隔时间小于 10ms，建议用 2ms；在双字节控制时，两条地址指令时间间隔需大于 10ms。

## 6.3. 语音及命令码对应表

命令码	功能	描述
单字节： (E0H...EFH) 双字节： (FFE0H...FFEFH)	E0(FFE0) 音量最小，EF(FFEF)音量最大，共 16 级音量调节	在语音播放结束、播放过程中或者待机状态发此命令调节音量。
单字节：F1H 双字节：FFF1H	无缝衔接循环播放指令 (音源本身需做相应调整，详情请联系我司业务员)	执行此命令可循环播放当前段语音，可在语音播放时发送。F1(FFF1)循环指令执行过程中，可被 FE(FFFE)命令、普通地址指令、F3(FFF3)组合指令打断，并失效；需先发播放指令，再发循环播放指令。 再次发送 F1(FFF1)，停掉当前循环功能
单字节：F2H 双字节：FFF2H	循环播放当前语音	执行此命令可循环播放当前段语音，可在语音播放/语音停止时发送。F2(FFF2)循环指令执行过程中，可被 FE(FFFE)指令、普通地址指令、F3(FFF3)组合指令打断，并失效；需先发播放指令，再发循环播放指令。 再次发送 FFF2，停掉当前循环功能
单字节：F3H 双字节：FFF3H	连码播放	F3(FFF3H)+语音地址 A, F3(FFF3H)+语音地址 B, F3(FFF3H)+语音地址 C, ... 在播放地址 A 的时候，收到后面的码不中断，播放完 A，就播放 B，然后播放 C...。FFF3 与地址之间间隔需小于 10ms，大于 2ms，建议采用 5ms；而一组连码地址与下一组地址需大于 5ms 延时，建议采用 10ms（“F3(FFF3H)+语音地址”为一组连码地址），最多支持 40 段连码
单字节：FEH 双字节：FFFEH	停止播放当前语音	执行此命令可停止播放当前段语音。

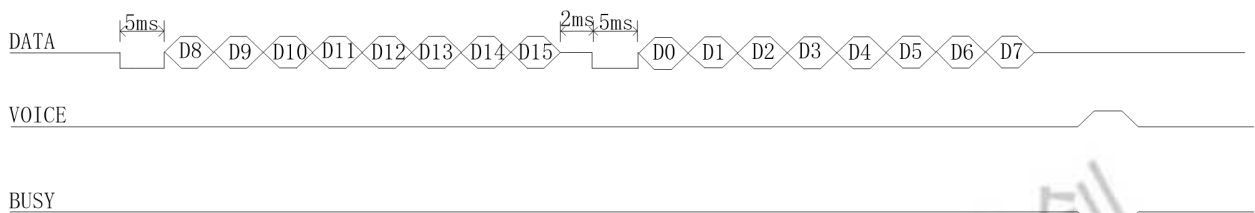
**注意：**

1、在未停止播放的情况下，如果没有命令码 FFF3H，只有语音地址，就会打断正在播放的语音

2、连码播放时，在语音还在播放期间，再次发送连码指令，不会打断当前连码播放，等待当前连码播放完成后，继续播放刚刚发送的连码语音

3、连码指令必须配合地址使用（例如：FFF3H+0000H+FFF3H+0001H）。FFF3H 可以方便的组合不同语音，FFF3H+地址 A+FFF3H+地址 B，最大可组合 40 组内容;也可以通过判断语音播放时的 BUSY 电平和播放结束时的 BUSY 电平的变化，完成组合播放。单字节发码同理；

4、因为 WT588F 语音初始化时间需要较长时间，而且初始化期间无法响应指令，因此，建议用户使用连码功能时一组连码地址发送之后延时 10ms 以上再发送下一组连码地址。

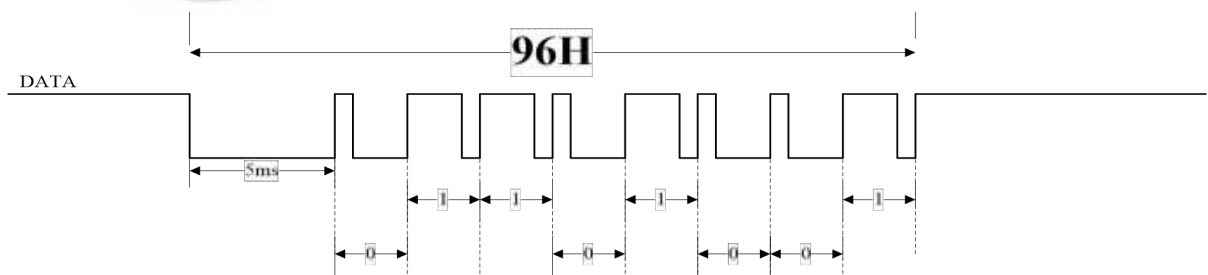
**6.4. 一线串口时序图**

先把数据线拉低 5ms（时间范围为 5ms-20ms）后，发送 16 位数据，先发高字节，再发低字节，先发送低位，再发送高位，使用高电平和低电平比例来表示每个数据位的值。

**注意：必须高电平在前，低电平在后。**

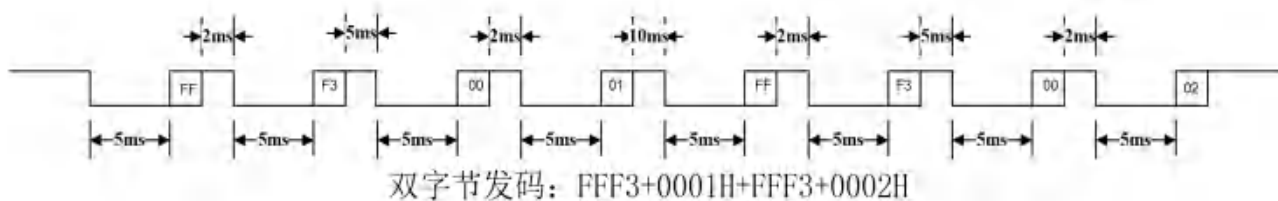
推荐使用 200us: 600us。取值范围：100us:300us ~ 400us:1200us。推荐使用 3:1 和 1:3 电平比例（电平比例范围为 3:1~5:1、1:3~1:5）以保障通讯稳定。

假如我们要发送 96H，那么他对应的时序图，如下所示：



假如我们要让芯片依次播放 01/02 地址的语音内容。即连码指令播放、FFF3+0001+FFF3+0002.对应时序可以如下图所示：





**注意:**

- 1、因为 WT588F 语音初始化时间需要较长时间，而且初始化期间无法响应指令；
- 2、连码播放时字节与字节之间需要留 2ms 的间隔时间，指令与地址之间需留 5ms 的间隔时间，而一组连码地址与下一组地址需大于 10ms 延时，现采用 10ms 的延时，可以参考上图；
- 3、芯片输入口默认内部 1M 下拉。因此客户在做低功耗休眠时，播放结束后可以将 DATA 拉低，防止倒灌电流；若 DATA 拉低，发指令前需将 DATA 拉高大于 5ms 后再发送指令。
- 4、发码后，BUSY 会在 20~50ms 内发生变化，由于内部功放使能受 BUSY 脚控制，且打开时间较慢，所以 BUSY 变化后最少有 70ms 静音，上位机已添加静音的操作。

## 7. 两线串口通讯

两线串口模式可以利用 MCU 通过 GPIO 作为 DATA 线和 CLK 线给 WT588F 系列语音芯片发送数据以达到控制的目的。可以实现控制语音播放、停止、循环等。两线串口控制方式具有较好的抗干扰能力，单字节时，一条指令最短时间为 8.6ms ((5ms+0.2ms\*8)+2=8.6ms)。详细说明可见下面介绍。

**注意:** 芯片 PA2 脚平时为低，播放为高

### 7.1. 管脚分配

封装形式	管脚		
	PA2	PA1	PA0
SOP14	默认为 BUSY 信号输出端	CLK	DATA

### 7.2. 语音地址对应关系

**注:** 地址小于 224 段默认为单字节发码，大于等于 224 段为双字节发码。

数据 (十六进制)	功能
00H (单字节) /0000H (双字节)	播放第 0 段语音
01H (单字节) /0001H (双字节)	播放第 1 段语音
02H (单字节) /0002H (双字节)	播放第 2 段语音
.....	
DFH (单字节) /00DFH (双字节)	播放第 223 段语音
00E0H (双字节)	播放第 224 段语音
.....	
03E6H (双字节)	播放第 998 段语音
03E7H (双字节)	播放第 999 段语音

**注意:**

- 1、如要播放该地址语音，只要发送该地址就能自动播放该地址语音，双字节指令时，一条地址指令字节间间隔时间小于 10ms，建议用 2ms；两条地址指令时间间隔需大于 10ms。
- 2、地址段数小于 224 段，默认为单字节发码（参考单字节说明书），大于等于 224 段，默认双字节发码

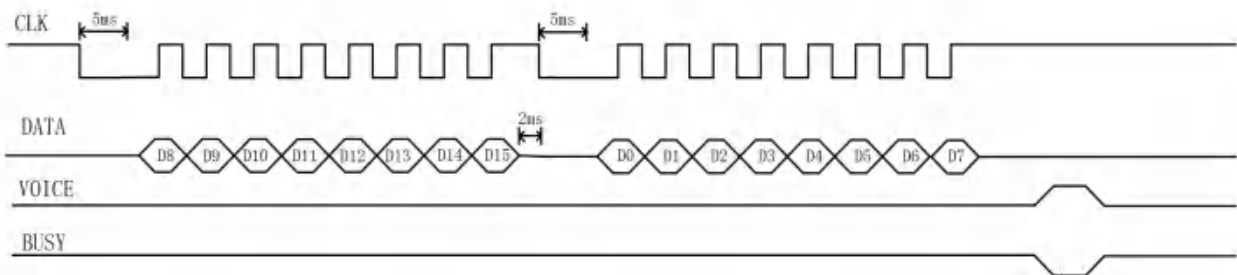
### 7.3. 语音及命令码对应表

命令码	功能	描述
单字节: (E0H...EFH) 双字节: (FFE0H...FFE7H)	E0(FFE0) 音量最小, EF(FFE7)音量最大,共 16 级音量调节	在语音播放结束、播放过程中或者待机状态发此命令调节音量。
单字节: F2H 双字节: FFF2H	循环播放当前语音	执行此命令可循环播放当前段语音, 可在语音播放/语音停止时发送。F2(FFF2)循环指令执行过程中, 可被 FE(FFFE)命令、普通地址指令、F3(FFF3)组合指令打断, 并失效; 需先发播放指令, 再发循环播放指令。 再次发送 F2(FFF2), 停掉当前循环功能
单字节: F3H 双字节: FFF3H	连码播放	F3(FFF3H)+语音地址 A, F3(FFF3H)+语音地址 B, F3(FFF3H)+语音地址 C, ... 在播放地址 A 的时候, 收到后面的码不中断, 播放完 A, 就播放 B, 然后播放 C...。FFF3 与地址之间间隔需小于 10ms, 大于 2ms, 建议采用 5ms; 而一组连码地址与下一组地址需大于 5ms 延时, 建议采用 10ms (“F3(FFF3H)+语音地址”为一组连码地址), 最多支持 40 段连码
单字节: FEH 双字节: FFEH	停止播放当前语音	执行此命令可停止播放当前段语音。

**注意:**

- 1、在未停止播放的情况下, 如果没有命令码 FFF3H, 只有语音地址, 就会打断正在播放的语音
- 2、连码播放时, 在语音还在播放期间, 再次发送连码指令, 不会打断当前连码播放, 等待当前连码播放完成后, 继续播放刚刚发送的连码语音
- 3、连码指令必须配合地址使用 (例如: FFF3H+0000H+FFF3H+0001H)。FFF3H 可以方便的组合不同语音, FFF3H+地址 A+FFF3H+地址 B, 最大可组合 40 组内容;也可以通过判断语音播放时的 BUSY 电平和播放结束时的 BUSY 电平的变化, 完成组合播放。单字节发码同理;
- 4、因为 WT588F 语音初始化时间需要较长时间, 而且初始化期间无法响应指令, 因此, 建议用户使用连码功能时一组连码地址发送之后延时 10ms 以上再发送下一组连码地址。

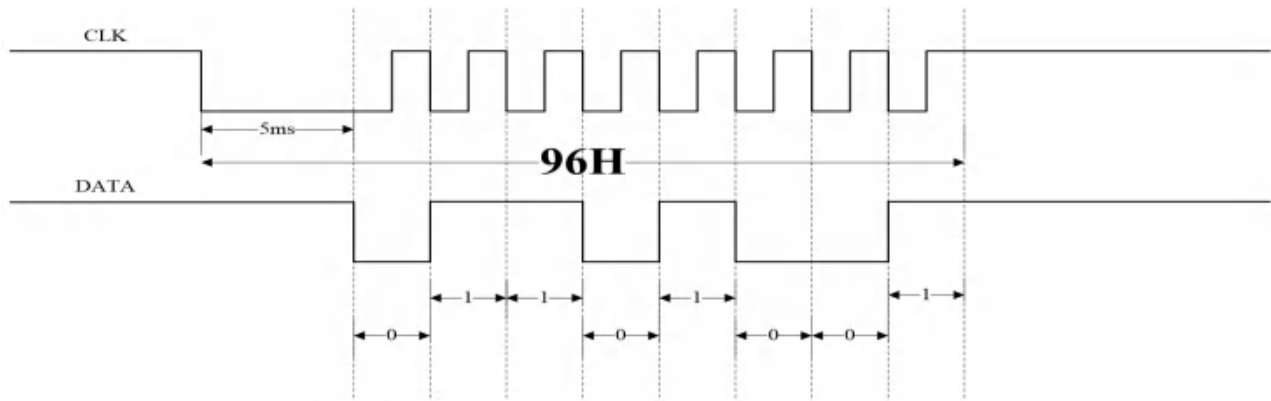
### 7.4. 两线串口时序图



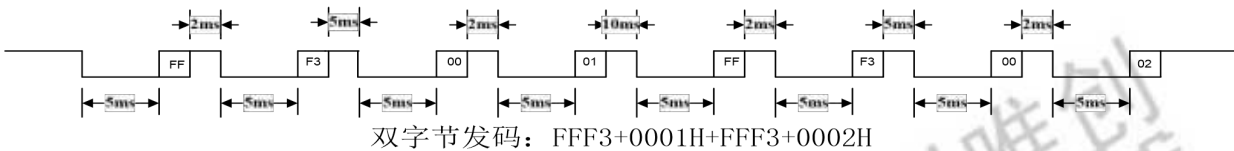
两线串口控制模式由芯片片时钟 CLK 和数据 DATA 进行控制操作, 每发一个字节数据前, 时钟信号 CLK 拉低 5ms 至 20ms, 推荐使用 5ms。接收数据低位在先, 在时钟的上升沿接收数据。时钟周期介于 200us~3.2ms 之间, 推荐高低电平持续时间各使用 350us (即一个 bit 发送的时钟周期为 700us)。发数据时先发高字节, 再发低字节, 先发低位, 再发高位。数据中的 00H~DFH 为语音地址指令, E0H~EFH 为音量调节命令, F2H 为循环播放命令, FEH

为停止播放命令。

假如我们要发送 96H，那么他对应的时序图，如下所示：



假如我们要让芯片依次播放 01/02/ 地址的语音内容。即连码指令播放 FFF3+0001+FFF3+0002（双字节发码，单字节为 F3+01+F3+02）对应时序可以如下图所示：



注意：

- 1、因为 WT588F 语音初始化时间需要较长时间，而且初始化期间无法响应指令；
- 2、连码播放时字节与字节之间需要留 2ms 的间隔时间，指令由于地址之间需留 5ms 的间隔时间，而一组连码地址与下一组地址需大于 10ms 延时，现采用 10ms 的延时，可以参考上图；
- 3、芯片输入口默认内部 1M 下拉。因此客户在做低功耗休眠时，播放结束后可以将 CLK 和 DATA 拉低，防止倒灌电流；若 CLK 拉低，发指令前需将 CLK 拉高大于 5ms 后再发送指令。
- 4、发码后，BUSY 会在 20~50ms 内发生变化，由于内部功放使能受 BUSY 脚控制，且打开时间较慢，所以 BUSY 变化后最少有 70ms 静音，上位机已添加静音的操作。



## 8. 程序范例

### 8.1. 一线串口单字节控制程序

```
/*-----  
;模块名称:Line_1A_WT588F_Single(UC8 DDATA)  
;功 能:实现一线串口通信函数  
;入 参: DDATA 为发送数据  
;出 参:  
;P_DATA 控制 IO  
;-----*/  
  
#define UC8    unsigned char  
#define P_DATA    P01  
  
Void Line_1A_WT588F(UC8 DDATA)  
{  
    UC8  S_DATA, j;  
    UC8  B_DATA;  
    S_DATA =  DDATA;  
    P_DATA = 0;  
    Delay_1ms(5);           //延时 5ms  
    B_DATA = S_DATA & 0X01;  
  
    for(j = 0; j < 8; j++)  
    {  
        if(B_DATA == 1)  
        {  
            P_DATA = 1;  
            Delay_N10us(60); //延时 600us  
            P_DATA = 0;  
            Delay_N10us(20); //延时 200us  
        }  
  
        else  
        {  
            P_DATA = 1;  
            Delay_N10us(20); //延时 200us  
            P_DATA = 0;  
            Delay_N10us(60); //延时 600us  
        }  
        S_DATA = S_DATA >> 1;  
        B_DATA = S_DATA & 0X01;  
    }  
}
```



```
P_DATA = 1;
```

```
}
```

## 8.2. 一线串口双字节控制程序

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

```
;模块名称:Line_1A_WT588F_Couple(UI16 USER_DATA)
```

```
;功 能:实现一线串口通信函数
```

```
;入 参:USER_DATA
```

```
;出 参:
```

```
;one_line_DATA 控制 IO
```

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

```
#define UC8    unsigned char
```

```
#define UI16   unsigned int
```

```
#define one_line_DATA P01
```

```
void Line_1A_WT588F_Couple(UI16 USER_DATA)
```

```
{
```

```
    UC8 i, num_temp;
```

```
    UC8 B_DATA;
```

```
    UI16  ddata_temp, pdata_temp;
```

```
    ddata_temp = USER_DATA;
```

```
    pdata_temp = ddata_temp & 0X00FF;    //移位前准备
```

```
    pdata_temp <<= 8;                    //低字节移位到高字节
```

```
    ddata_temp >>= 8;                    //高字节移位到低字节
```

```
    ddata_temp |= pdata_temp;           //移位后重新合并
```

```
    num_temp = 16;
```

```
    one_line_DATA = 0;                  //拉低数据线
```

```
    Delay_10us(500);                   //延时 5MS
```

```
    B_DATA = ddata_temp & 0X0001;
```

```
    for(i = 0; i < num_temp; i++)
```

```
    {
```

```
        if(i == 8)
```

```
        {
```

```
            one_line_DATA = 1;
```

```
            Delay_10us(200);           //延时 2MS
```

```
            one_line_DATA = 0;
```

```
            Delay_10us(500);           //延时 5MS
```

```
        }
```

```
        one_line_DATA = 1;              //拉高数据传输线 ,准备传输数据
```

```
        if(B_DATA == 0)
```

```
        {
```



```

        /*表示逻辑电平 0*/
        Delay_10us(20);           // 延时 200us
        one_line_DATA = 0;
        Delay_10us(60);           // 延时 600us
    }

    else
    {
        /*表示逻辑电平 1*/
        Delay_10us(60);           // 延时 600us
        one_line_DATA = 0;
        Delay_10us(20);           // 延时 200us
    }

    ddata_temp = ddata_temp >> 1;
    B_DATA = ddata_temp & 0x0001;
}

one_line_DATA = 1;
}

```

**8.3. 一线串口单字节连码示例(F3 + 01 + F3 + 02)**

```

/*-----
;模块名称:List_1A_Play_WT588F()
;功    能:实现一线串口连码发送函数
;入    参:
;出    参:
;-----*/

```

```

Void List_1A_Play_WT588F(void)
{
    Line_1A_WT588F(0xF3);
    Delay_1ms(2);           //延时 2ms
    Line_1A_WT588F(0x01);
    Delay_1ms(5);           //延时 5ms
    Line_1A_WT588F(0xF3);
    Delay_1ms(2);           //延时 2ms
    Line_1A_WT588F(0x02);
    Delay_1ms(5);           //延时 5ms
}

```

**8.4. 一线串口双字节连码示例(FFF3 + 0001 + FFF3 + 0002)**

```

/*-----
;模块名称:List_1A_Play_WT588F()
;功    能:实现一线串口连码发送函数
;入    参:

```



```
;出 参:
;-----*/
```

```
Void List_1A_Play_WT588F_Couple(void)
{
    Line_1A_WT588F_Couple(0xFFF3);
    Delay_1ms(5);           //延时 5ms
    Line_1A_WT588F_Couple(0x0001);
    Delay_1ms(10);        //延时 10ms
    Line_1A_WT588F_Couple(0xFFF3);
    Delay_1ms(5);         //延时 5ms
    Line_1A_WT588F_Couple(0x0001);
    Delay_1ms(10);        //延时 10ms
}
}
```

**8.5. 二线串口单字节控制程序**

```
/*-----
;模块名称:Line_2A_WT588F(UC8 DDATA)
;功 能:实现二线串口通信函数
;入 参:DDATA 为发送数据
;出 参:
;CLK_2A      时钟线
;P_DATA_2A   数据线
;-----*/
```

```
#define UC8    unsigned char
#define CLK_2A    P01
#define P_DATA_2A    P00

Void Line_2A_WT588F(UC8 DDATA)
{
    UC8 S_DATA, j;
    UC8 B_DATA;
    CLK_2A    = 1;           //拉高 CLK
    P_DATA_2A = 1;           //拉高 DATA
    S_DATA = DDATA;
    CLK_2A    = 0;           //拉低 CLK
    Delay_1ms(5);           //延时 5ms
    B_DATA = S_DATA & 0X01;

    for(j = 0; j < 8; j++)
    {
        CLK_2A = 0;           //拉低 CLK
        P_DATA_2A = B_DATA; //传输数据一位
        Delay_N10us(30);     //延时 300us
    }
}
```



```

CLK_2A = 1;           //拉高 CLK
Delay_N10us(30);    //延时 300us
S_DATA = S_DATA >> 1;
B_DATA = S_DATA & 0X01;
}

```

```

P_DATA_2A = 1;
CLK_2A    = 1;
}

```

## 8.6. 二线串口双字节控制程序

```

/*-----
;模块名称:Line_2A_WT588F(UI16 USER_DATA)
;功    能:实现二线串口通信函数
;入    参:USER_DATA 为发送数据
;出    参:
;CLK_2A      时钟线
;DATA_2A     数据线
;-----*/

```

```

#define UC8    unsigned char
#define UI16   unsigned int
#define CLK_2A    P01
#define DATA_2A  P02

```

```

void Line_2A_WT588F_Couple(UI16 USER_DATA)

```

```

{
    UC8 i, num_temp;
    UI16  ddata_temp, pdata_temp;
    ddata_temp = USER_DATA;
    pdata_temp = ddata_temp & 0X00FF;
    ddata_temp >>= 8;
    pdata_temp <<= 8;
    ddata_temp |= pdata_temp;
    num_temp = 16;
    CLK_2A = 0;           //拉低 CLK
    Delay_10us(500);     //延时 5ms

    for(j = 0; j < num_temp; j++)
    {
        if(j == 8)
        {
            CLK_2A = 1;
            DATA_2A = 1;
            Delay_N10us(200); //延时 2ms

```





```

        CLK_2A    = 0;
        Delay_N10us(500);    //延时 5ms
    }
    CLK_2A = 0;
    DATA_2A = ddata_temp & 0X0001;
    Delay_N10us(20);        //延时 200us
    CLK_2A    = 1;
    Delay_N10us(20);        //延时 200us
    ddata_temp = ddata_temp >> 1;
}

CLK_2A = 1;
DATA_2A = 1;
}

```

**8.7. 二线串口单字节连码示例(F3 + 01 + F3 + 02)**

```

/*-----
;模块名称:List_2A_Play_WT588F()
;功    能:实现二线串口单字节连码发送函数
;入    参:
;出    参:
;-----*/

```

```

Void List_2A_Play_WT588F(void)
{
    Line_2A_WT588F(0xF3);
    Delay_N10us(200);        //延时 2ms
    Line_2A_WT588F(0x01);
    Delay_N10us(500);        //延时 5ms
    Line_2A_WT588F(0xF3);
    Delay_N10us(200);        //延时 2ms
    Line_2A_WT588F(0x02);
    Delay_N10us(500);        //延时 5ms
}

```

**8.8. 二线串口双字节连码示例(FFF3 + 0001 + FFF3 + 0002)**

```

/*-----
;模块名称:List_2A_Play_WT588F()
;功    能:实现二线串口双字节连码发送函数
;入    参:
;出    参:
;-----*/

```

```

Void List_2A_Play_WT588F_Couple(void)
{
    Line_2A_WT588F_Couple(0xFFF3);
}

```

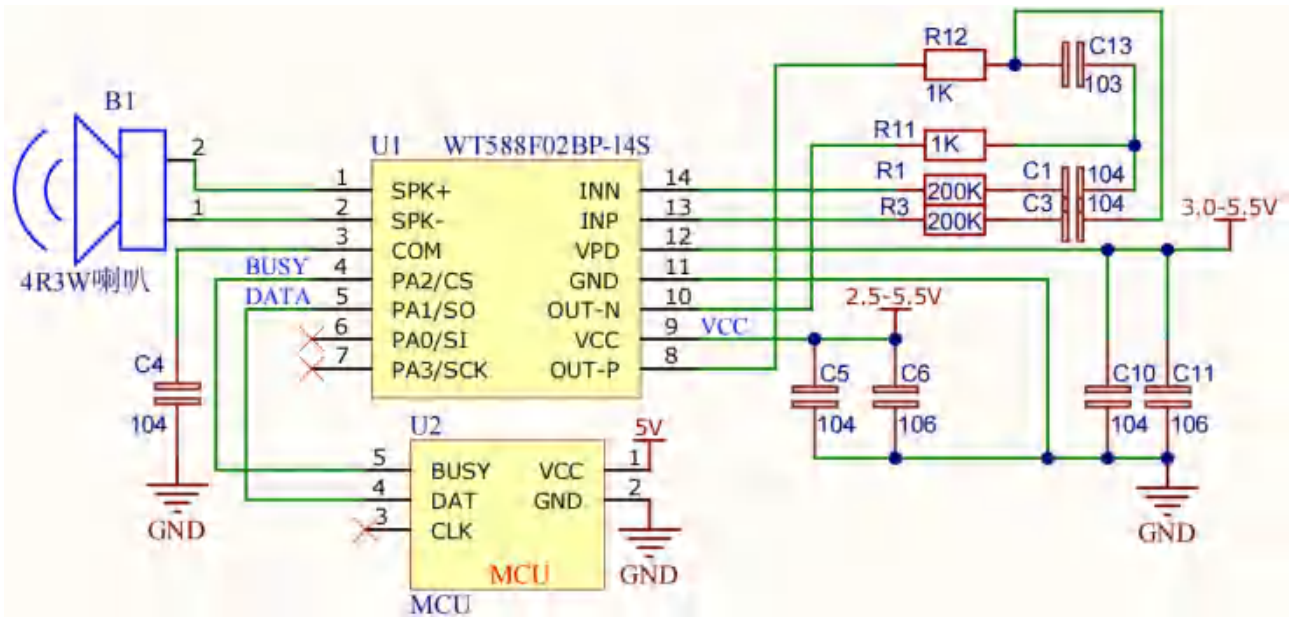


```
Delay_N10us(500);           //延时 5ms  
Line_2A_WT588F_Couple(0x0001);  
Delay_N10us(1000);         //延时 10ms  
Line_2A_WT588F_Couple(0xFFF3);  
Delay_N10us(500);          //延时 5ms  
Line_2A_WT588F_Couple(0x0002);  
Delay_N10us(1000);         //延时 10ms  
}
```

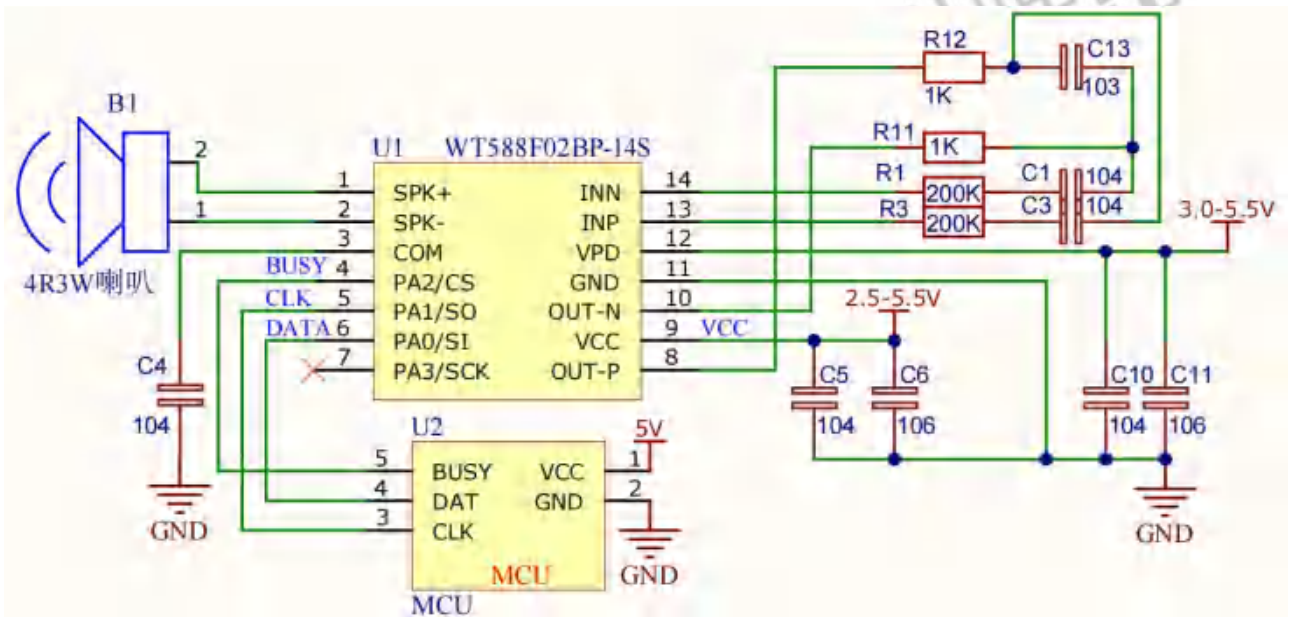


## 9. 芯片应用原理图

### 9.1. 一线应用原理图



### 9.2. 两线应用原理图



### 9.3. 相关参数参考

#### 1、WT588F02BP-14S 芯片供电说明：

1.1、VCC 脚为 IO 及语音音频输出部分供电，电压范围为 2.0-5.5V，建议供电范围 3.3-5V，当语音芯片采用 3.3V 供电而客户 MCU 采用 5V 供电或者语音芯片采用 5V 供电而客户 MCU 采用 3.3V 供电时，需要采用电平转换电路，可能会产生漏电流，请注意检查；

1.2、VPD 脚为内部功放供电，电压范围为 3.0-5.5V，建议供电范围为 3.3-5V，当客户需要声音较大时，建议采用 5V 供电；当客户更多的是考虑成本，不想采用多的降压芯片时，可以采用与客户 MCU 的 IO 口电平一致的供电电压供电（客户 MCU 的 IO 口可能是 3.3V 或者 5V），此时客户采用 3.3V 供电或者 5V 供电即可。

2、功率及喇叭参数（5V 供电）：

A、当需求输出 3W 时，R1、R3 采用 200K 电阻，R11、R12 采用 1K 电阻，C13 采用 103 电容，C1、C3 采用 104 电容，此时喇叭选用 4R3W；

B、当需求输出 2W 时，R1、R3 采用 200K 电阻，R11、R12 采用 1K 电阻，C13 不贴电容采用 3.9K 电阻，C1、C3 采用 104 电容，此时喇叭选用 4R2W；

C、当需求输出 1W 时，R1、R3 采用 300K 电阻，R11、R12 采用 6.8K 电阻，C13 采用 103 电容，C1、C3 采用 104 电容，此时喇叭选用 4R1W；

D、当需求输出 2W 时，R1、R3 采用 100K 电阻，R11、R12 采用 1K 电阻，C13 电容采用 3.3K 电阻，C1、C3 采用 104 电容，此时喇叭选用 8R2W；

E、当需求输出 1W 时，R1、R3 采用 200K 电阻，R11、R12 采用 1K 电阻，C13 不贴电容采用 3.9K 电阻，C1、C3 采用 104 电容，此时喇叭选用 8R1W；

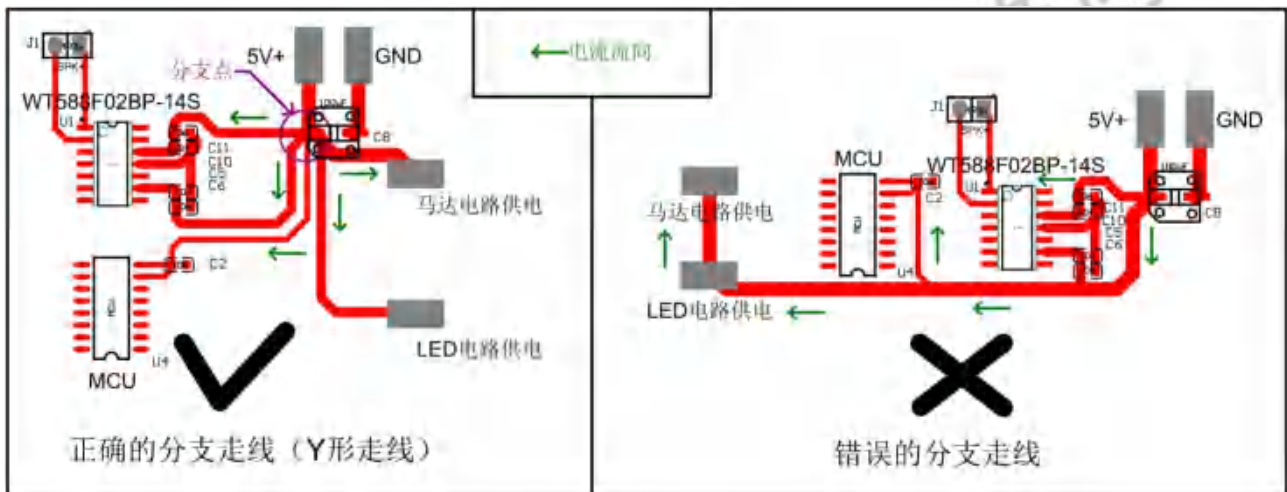
F、增益设置和输入电阻关系的公式： $A_{vd}=230K/R_{in}$ ； $R_{in}=R1+R11$  或  $R_{in}=R3+R12$

3、语音芯片内置功放启动时间设定参数：

A、当语音芯片第 3 脚（COM）的 C4 电容采用 104 电容且芯片 5V 供电时，启动时间为 61ms(典型)；

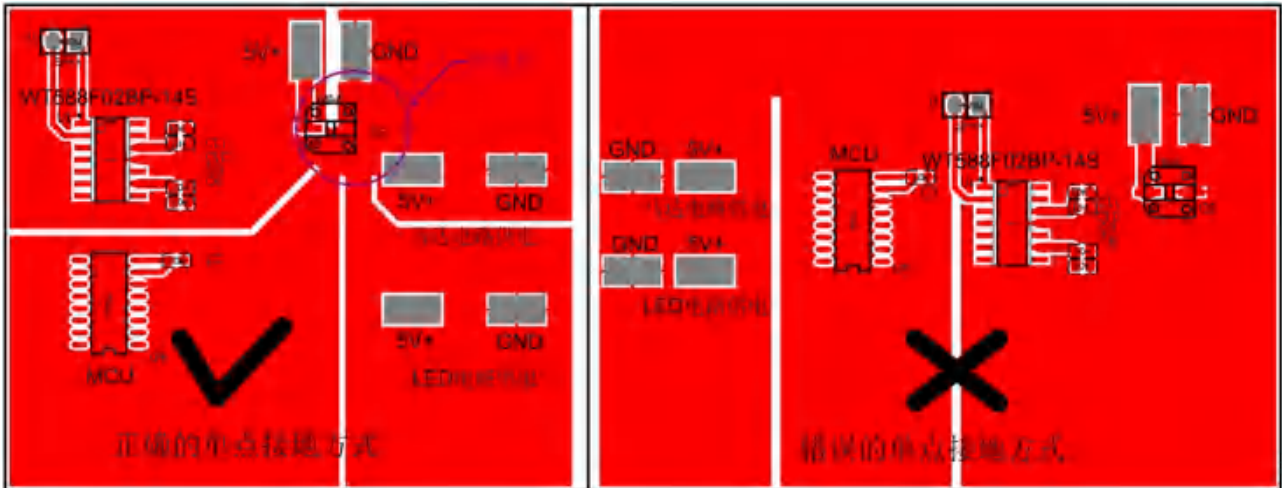
B、当语音芯片第 3 脚（COM）悬空时，启动时间为 43ms(典型)，不建议采用此方案，此时缺少了一个基本的抗干扰能力；

9.4. 相关规范说明



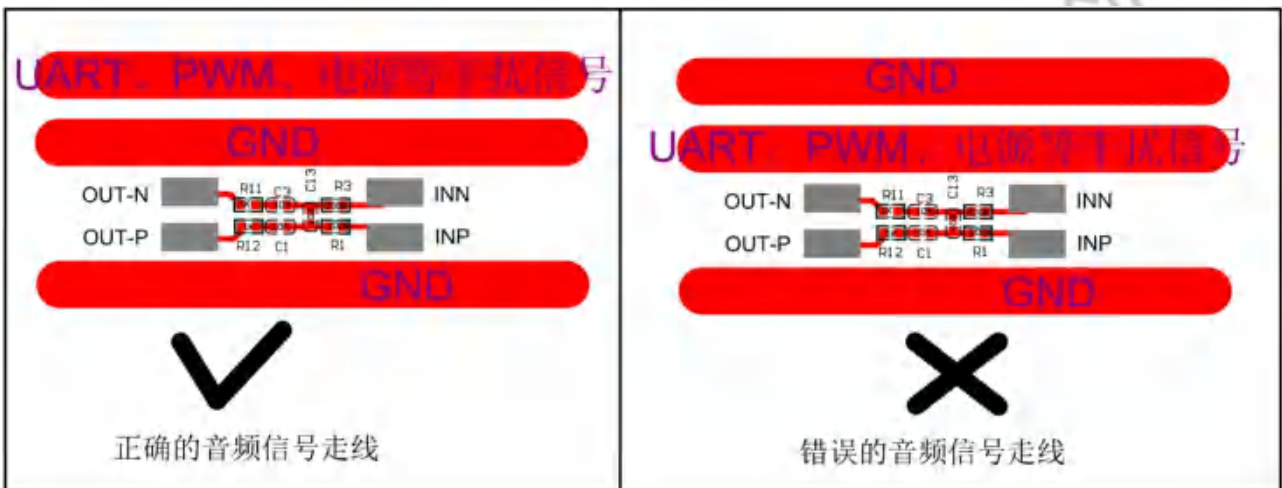
1、在 PCB 布局时，语音芯片电源脚需要接电容到地，电源的走线必须先经过电容再连接到芯片电源脚，电容 GND 网络到语音芯片 GND 脚与电容 5V 网络到语音芯片 VCC、VPD 脚间距离分别小于 5mm；如果有 ESD 元件时，电源走线先经过 ESD 元件，再经过电容，最后连接到语音芯片电源脚；

2、PCB 设计时，为了避免其它电路工作时影响电源系统的稳定性，建议电源走线为分支走线（Y 型走线），从电源输入端电容正极开始分支走线；



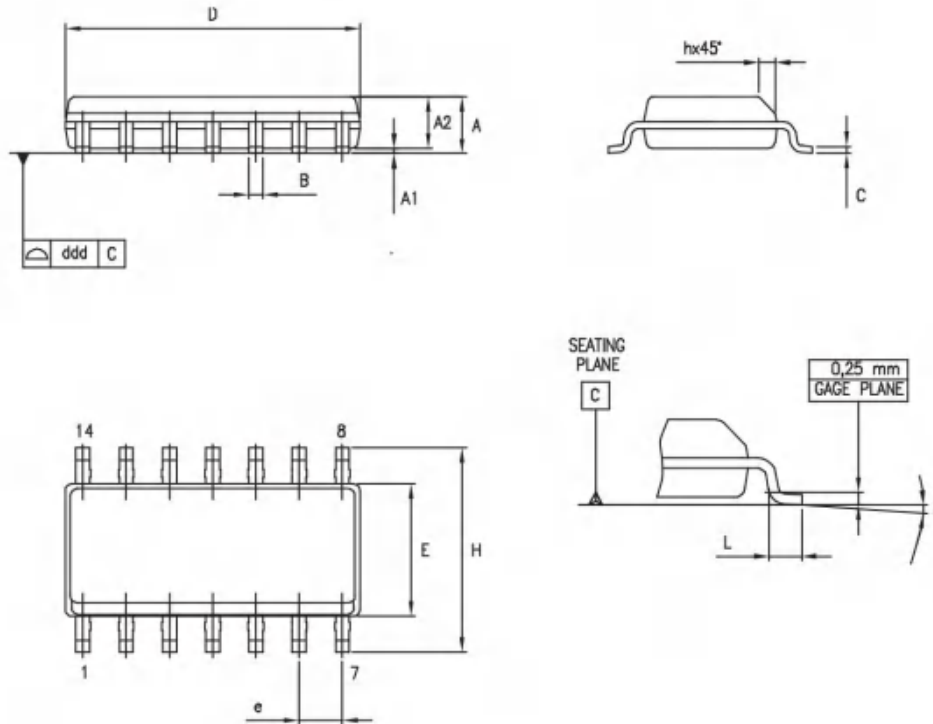
3、地分割走线：当系统电路有数字信号或无线信号，建议将语音芯片与其它有带干扰的电路的地线独立，单点连接；

4、解决“POP”声：语音芯片标配的电路及程序控制就已经解决了“POP”声音问题；当客户 MCU 的 IO 口需要控制语音芯片 BUSY 脚来控制功放输出时，可能需要在程序控制、音频处理方面做相关参数的调整（具体请咨询我司 FAE）。



5、音频信号输出（OUT-N 与 OUT-P 脚），音频电路信号走线及途径元件尽可能用地包围走线，避开无线信号、数字信号走线，不要走平行线，减小干扰。

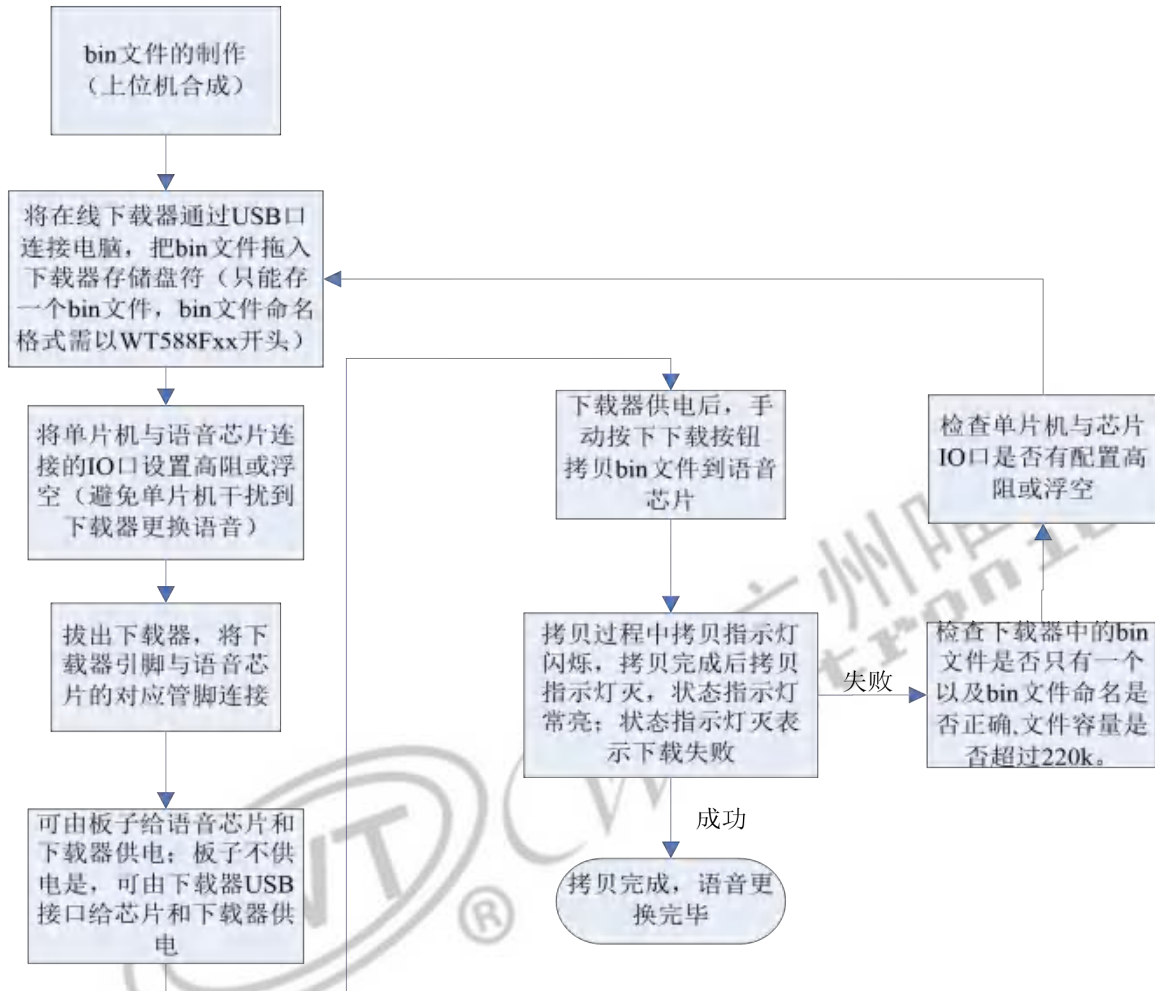
### 10. 封装管脚图



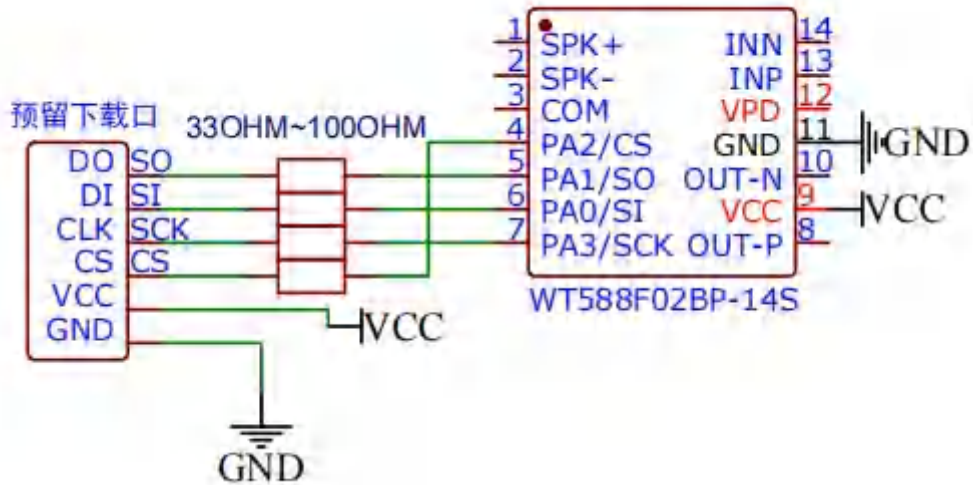
名称	尺寸					
	毫米			英寸		
	最小值	典型值	最大值	最小值	典型值	最大值
A	1.35	1.56	1.75	0.05	1.56	0.068
A1	0.10		0.25	0.004		0.009
A2	1.10		1.65	0.04		0.06
B	0.33		0.51	0.01		0.02
C	0.19		0.25	0.007		0.009
D	8.55		8.75	0.33		0.34
E	3.80		4.0	0.15		0.15
e		1.27			0.05	
H	5.80		6.20	0.22		0.24
h	0.25		0.50	0.009		0.02
L	0.40		1.27	0.015		0.05
k	8° (最大值)					
ddd			0.10			0.001

## 11. 客户在线下载器

将 bin 文件存入在线下载器，通过在线下载器连接设备即可更换语音，可用于生产或客户端设备语音的更换，使用方便、快捷、操作简单，适用于需频繁更换语音的设备。（在线下载器已有视频解说，可联系业务员提供。）



### 11.1. 下载原理图



注意：更换时，MCU 的 GPIO 跟语音芯片连接的 IO 口需要处于高阻或者悬空状态。

### 11.2. 下载器使用说明



(1) 制作 bin 文件：登陆网页上位机网址后，加载要播放的语音文件或使用文字转换音频功能加载语音后制作 bin 文件。WT588 网页制作客户登录网址：<http://WT588f.waytronic.com:8083>。bin 文件制作步骤可参考《WT588F 网页上位机操作说明 2020-8-31》或视频解说，可联系业务员提供。

(2) 存放 bin 文件：通过 USB 连接电脑，会出现一个盘符（第一次使用需格式化），将做好的 bin 文件存入盘符中，bin 文件名称格式需为 WT588XX(以 WT588 开头，后面可自行定义)。Bin 文件存放后，将下载器与电脑断开连接。

(3) 连接芯片：让单片机释放语音芯片，可将单片机连接语音芯片的 IO 口设置为输入或者浮空；将下载器的烧写口连接到语音芯片对应的 IO 口（DO-PA1、DI-PA0、CLK-PA3、CS-PA2、VCC-VCC、GND-GND），客户画板时可在板子上留好下载口，方便下载更换语音。

(4) 给下载器供电：客户板子有供电时可通过 VCC 和 GND 脚给下载器供电；客户板子没供电的情况下，可以通过 USB（可用适配器或电脑 USB 供电（一般不建议用电脑 USB 供电））





给下载器和芯片供电；如板子有按上面原理图留下载口，可以直接将下载器插到下载口进行烧写。(使用“WT588FXXB 下载器”需要给语音芯片和下载器的电压控制在 3.3V~4.2V；使用“USB 下载器”，可以供电 3.3V~5.0V)

(5) 使用“USB 下载器”，需要手动按下载按钮。(使用 WT588FXXB 下载器，下载器供电 2s 后，下载器会自动将 bin 文件下载到语音芯片里面，**故需先连接下载口后再给下载器供电。**)

(6) 下载判断：POWER 电源指示灯、DOWN 拷贝指示灯、OK 状态指示灯、NG 状态指示灯。

电源指示灯常亮表示该产品已经处于工作状态。

拷贝指示灯：平时熄灭，拷贝过程中闪烁，拷贝结束后熄灭。

状态指示灯：拷贝成功 OK 等常亮，拷贝失败 NG 灯常亮。

(在线下载器已有视频解说，可联系业务员提供。)

## 12. 关于芯片的静电保护

通过测试，发现芯片如果受到 ESD 容易使芯片损坏且不可逆转，接触式 6KV 时，尤其是 4、7、10 脚，对静电的抗性较弱。我司对 ESD 设计研究得出以下的一些信息。

一般 ESD 设计一般有三个方向：方向一，降低减弱 ESD 放电对保护模板的冲击强度；方向二，增加目标 ESD 的免疫力(Robust)；方向三，软件上的看门狗/Reset。

这三个方向中，方式一是我们主要的研究方向，因为方向二受材质和本身晶圆的影响不容易处理，而方向三可靠性不高，因为该芯片已有看门狗但是是软件看门狗，可靠性不高，如果导致芯片损坏那么看门狗也没任何用处。

### 12.1. 静电防护策略分析

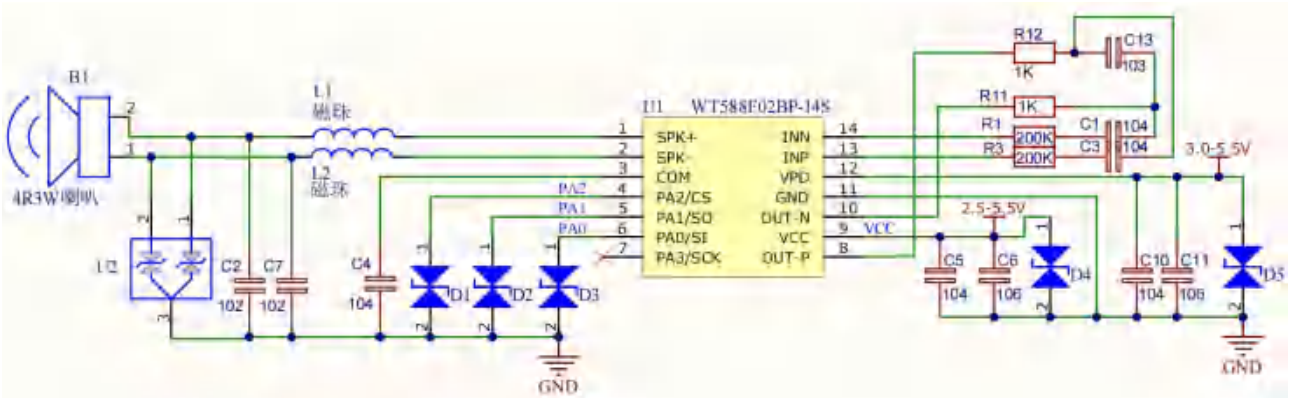
我们对芯片防静电的措施一方面是从我们自身生产上做工作，另一方面是客户在客户本身生产上做工作，并且在完成成品后在客户端使用也能有一定的静电保护措施。

从物理上，我司的生产随着生产引入作业前必须佩戴接线静电手环、机台接地后，由于静电引起的不良已有降低，是否就能完全避免静电的影响呢？答案是否的，普通衣服走动过程中会产生 1KV 的静电电压，在测试中可以得知 1KV 静电暂时不会对我们该 IC 引起损坏或不良，但不排除冬天等特殊干燥情况下，衣服静电比该值高，静电不由工人的手传导到机台芯片，而是衣服等接触传导到机台等地方引起不良，所以尽可能让我们的工作人员穿静电服和导电鞋，并且保证环境的湿度与温度，让静电尽可能少产生。此外，除了生产过程中需要防范，软件的设计上也尽可能做一些防 ESD 的措施，首先不用的 IO 口进行接地处理。

而客户生产过程中，大多数是外发 SMT，这一过程中客户的 SMT 厂是否做好静电防护需要客户跟踪，而且一家合格的 SMT 工厂应该会有相应的防静电措施。

除了这些因素外，最重要的是从 PCB 上芯片的外围电路做保护，这样才能保证产品在终端客户手上稳定可靠。

12.2. 可以尝试使用以下防静电电路



注：

1、静电管参数

当语音芯片在产品用应用，需要过相关静电认证时，需要加ESD元件，如TVS管；选型时需要根据芯片供电参数、供电环境等进行选配；

2、EMC 相关

当语音芯片内置功放输出推动喇叭发声音，有输出音频线过长或者器件布局靠近 EMI 敏感设备时，建议采用磁珠（L1、L2）和电容（C2、C7）组成的滤波网络，磁珠及电容要尽量靠近芯片放置。当客户应用场景不需要采用磁珠与电容组成的滤波网络时，语音芯片 SPK 输出直连喇叭既可。





广州唯创电子有限公司——于 1999 年创立于广州市天河区，是一家专注于语音技术研究、语音产品方案设计及控制等软、硬件设计的高新技术公司。业务范围涉及电话录音汽车电子、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、硬件开发能力和设计经验，秉持着「积极创新、勇于开拓、满足顾客、团队合作」的理念，为力争打造“语音业界”的领导品牌。

我公司是一家杰出的语音芯片厂家，从事语音芯片研究及外围电路开发；同时为有特别需求的客户制订语音产品开发方案，并且落实执行该方案，完成产品的研发、测试，声音处理，直至产品的实际应用指导等一系列服务。经过多年的发展，公司形成了一个完善的新品流程体系，能快速研发出新品以及完善产品。语音芯片系列包含:WT2000、WT2003、WT5001、WT588D、WTH、WTV、WTN 等，每一款语音芯片我们都追求精益求精、精雕细琢不断开发和完善，以求更佳的品质、为客户实现更多的价值。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。

不仅如此，还推出的多种语音模块，如 WT2000 录音模块，通过外围电路的扩展，更贴近广大用户的需求。

我们也是 MP3 芯片研发生产厂家。随着公司的外围技术扩展，在 2004 年开始生产 MP3 芯片，以及提供 MP3 方案。在同行里面有相当高的知名度，到现在为止更新换代一起出了 8 种 MP3 解决方案，并且得到市场的广泛认可。其中的 WT2000、WT2003 等芯片以音质表现极其优秀不断被客户所接受并使用。

在语音提示器方面，我们也从事于语音提示器生产厂家：经过多年的技术储备，开始向语音提示器领域拓展，并且得到了可喜的成果，成为语音提示器生产厂家里的一员。根据探头的类别：有超声波语音提示器，红外人体感应语音提示器，光感应语音提示器。同时也针对不同的领域开发了：自助银行语音提示器，欢迎光临迎宾器，语音广告机，语音门铃等等产品。可以肯定将来会有更多的新产品上市，来满足广大的用户的需求。让我们的生活更加智能化，人性化。

公司名称：广州唯创电子有限公司

电话：[020-85638557](tel:020-85638557)

E-mail：[864873804@qq.com](mailto:864873804@qq.com)

网址：[www.w1999c.com](http://www.w1999c.com)

地址：广州市花都区新华街道天贵大厦 A 座 706-708 室

公司名称：深圳唯创知音电子有限公司（研发中心）

网址：[www.waytronic.com](http://www.waytronic.com)

地址：深圳市宝安区福永街道中粮（福安）智汇创新园 6 栋 2 楼