



深圳唯创知音电子有限公司

Shenzhen Waytronic Electronic Co., Ltd

WT588F02B-8S C001_01 说明书



V1.00

免责声明:

深圳唯创知音电子有限公司申明：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

版本记录:	2
1. 产品简介.....	3
2. 概述:.....	4
3. 功能简述:.....	4
4. 管脚描述:	4
5. 极限参数:	5
6. 电气特性:	5
7. 串口通信说明:	6
8. WT588F02B 时钟功能的应用电路图:	10
9. 程序范例.....	11
10. 客户在线下载器.....	16
11. 封装管脚图.....	17

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	2021-8-20

拓展代码	拓展部分 (可多选)															
	时钟接口	AD/RC 按键	矩阵按键 (>12 键)	电位器检测	红外接收	红外发射	光波通讯	温度检测	湿度检测	压力检测	气压芯片	超声波测距	靠近检测 (红外)	高精度温度 (一线)	数码管显示 (4 位)	LCD 驱动
C001	√															
C002	√		√												√	
C003	√						√									
C004	√		√				√								√	

基础部分 (有且只选一个)				
标准内置 FLASH 语音	标准外置 FLASH 语音	多通道 midi 播放	多通道混音	标准录音
01	02	03	04	05
√				
√				
√				
√				

WT588FXX 封装			组合功能代码
SOP8	SOP16	SSOP24	
√	√	√	C001_01
		√	C002_01
	√	√	C003_01
		√	C004_01

其他型号芯片拓展请与业务员说明, 可根据以上方案进行选型。

1. 产品简介

语音更新

1. 语音下载：将 bin 文件存入在线下载器，通过烧写测试板连接模块即可更换语音，操作方便快捷，可用于模块 IC 的语音更换

存储方式

1. 芯片内部有 183K byte 的存储空间（不包括主控程序）
2. 芯片主控程序与内部存储空间均可反复擦写

播放方式

1. PWM 输出：16 位 PWM 纯音频输出，可直接驱动 8Ω /0.5W 喇叭和蜂鸣器，
2. DAC 输出：14 位 DAC 音频输出，可外接功放
3. 支持无缝衔接功能（音源需做处理，请与业务员联系）（PWM 和 DAC 输出可由 bin 文件设置）

音频采样率

1. 语音采样率：支持 6~32Khz；

控制方式

1. 一线串口控制
2. 按键功能控制（可根据实际情况定制功能）（串口控制和按键控制的主控程序不同，下单时请联系业务员）

操作指令

1. 音量调节指令
2. 循环播放指令
3. 播放固定语音指令
4. 播放暂停指令
5. 时钟功能
6. 闹钟定时功能

技术规格

1. 供电电压：DC2.0~5.5V
2. 工作温度：-40~75 度
3. 湿度：5%~95%

应用场景

1. 需频繁更换语音的设备
2. 能适应实时时钟芯片的大部分场合
3. 需更换语音内容的户外设备

时钟功能

1. 有设置当前时间的功能；
2. 有闹钟定时的功能；
3. 有设置倒计时的功能；
4. 自动计算星期，检测闰年；

2. 概述:

WT588F02B-8S 是深圳唯创知音电子有限公司最新研发的一款 **16 位 DSP 语音芯片、内部振荡 32Mhz，16 位的 PWM 解码**。强大功能让 WT588F02B-8S 成为语音芯片行业中的佼佼者。目前 WT588F02B-8S，**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）。WT588F02B-8S 区别于传统 OTP 芯片，最大的突破是**客户可以自己通过配套下载器实现在线更换芯片内部语音内容**；并且芯片**内置硬件 SPI、UART、基础时钟**等各类资源，可以为客户定制各种不同的个性化功能产品。

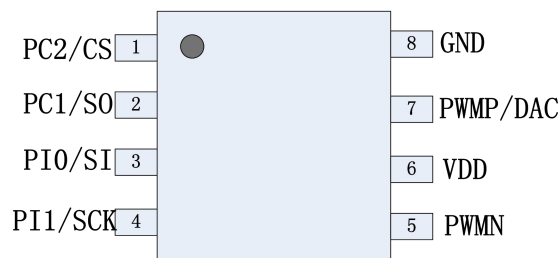
3. 功能简述:

1. 16 位 DSP 语音芯片、32Mhz 内部振荡；
2. 工作电压 2.0~5.5V；**建议电压：2.2~4.0V**（由于串口使用 TTL 电平通信）
3. 16bit 的 PWM/DAC 输出、可直接驱动 8R 0.5W 喇叭；
4. 支持 6K~32Khz 的 WAV 文件；
5. 客户可以通过配套下载测试板在线更换芯片内部语音内容；
6. 带语音提示功能（完整的 WT588F 标准播放功能）；
7. 支持 32768 段地址，有更多需求可以扩展；
8. 芯片内部有 183K byte 的存储空间（不包括主控程序）。
9. 芯片主控程序和内置存储数据均可擦除再烧写。
10. 读取当前时钟信息：年、月、日、时、分、秒、周
11. 设置当前时间信息：年、月、日、时、分、秒
12. 设置报警信息：列表 20 组报警
13. 放置用户数据（读写）：256 个字节 放在 FLASH 里面
14. 低电压检测功能，当电压低于 3.3V 时，不播放语音，并将低电压标志位置 1

选型注意:

1. 芯片的控制方式与输出方式可由 bin 文件设置，定制芯片时需要和业务员说明应用要求。
2. 如果需要更低待机功耗芯片，请联系我司业务员。

4. 管脚描述:



WT588F02B-8S

4.1. 管脚分布图

Pad Name	Pad No.	ATTR.	Description 描述
PC2	1	I/O	Uart-TX/闹钟信号输出口 2 (低电平)
PC1	2	I/O	WT5T01 秒跳信号接收口
PI0	3	I/O	Uart-RX
PI1	4	I/O	INT 闹钟信号输出口 1 (高电平)
PWMN	5	out	PWM 输出脚
VDD	6	Power	电源正极
PWMP	7	I/O	PWM 输出脚/DAC
GND	8	Power	电源负极

5. 极限参数:

标识	范围值	单位
VDD~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。

6. 电气特性:

参数	象征	最低限度	典型	最大限度	测试条件
工作电压	VCC	2.0V		5.5V	
振荡频率	Fbank0	4.096MHz±3%		8.192MHz±3%	
振荡频率(BANK 7)	Fbank7		32.768MHz±3%		
RC 振荡器频率	Frc 1		65.536 MHz±3%		
工作电流	IOP		5mA		空载
IO 口逻辑电平 (H)	VIH	0.8 VCC			
IO 口逻辑电平 (L)	VIL			0.2VCC	
	ILK			0.1 UA	
IO 口输出电平 (H)	VOH	0.95VCC			空载
IO 口输出电平 (L)	VOL			0.05V	空载
IO 口驱动电流	IOH		16mA		VOUT=VCC-0.4 V, PA 选择强度驱动选项
IO 口漏电流	IOL		-16 mA		Vout=0.4V PA

					选择了强度驱动选项
IO 口下拉电阻	RPD		50K/220 K/1M 无穷大 可以进行选择配置 默认 1M 内部下拉		引脚拉下, PA

7. 串口通信说明:

7.1. 串口控制说明:

7.1.1. 协议命令格式

WT588F02B-8S 内置标准 UART 异步串口接口, 属于 3.3V TTL 电平接口。通讯数据格式是: 起始位: 1 位; 数据位: 8 位; 奇偶位: 无; 停止位: 1 位。使用电脑串口调试助手, 需要正确设置串口的参数, 设置如图:



指令格式

唤醒码	起始码	长度	命令码	参数	累加和校验	结束码
0X00	0X7E	见下文	见下文	见下文	见下文	0XEF

返回格式

起始码	长度	命令码	参数	累加和校验	结束码
0X7E	04	见下文	见下文	见下文	0XEF

语音类串口指令: 只有指令长度和校验是正确的, 才会判断命令; 命令不存在时, 没有返回值, 参数过多也没有返回值, 参数错误返回 01

时钟类串口指令: 按照其表描述的返回一致就行, 只有长度和校验是正确的, 才会判断命令; 命令不存在, 没有返回值, 参数过多也没有返回值, 参数错误返回 01;

注意: “长度”是指长度 (1 字节)+命令码 (若干字节)+参数 (若干字节)+校验和 (1 个字节) 的长度 (字节数); “累加和校验”是指长度+命令码+参数的累加和的低字节;

返回参数(XX)	含义
00	代表成功
01	代表失败

7.1.2. 管脚说明:

封装形式	管脚	
	PC2	PI0
SOP8/DIP8	TXD	RXD

7.2. 语音播放指令：

命令码	功能	备注
A0	播放对应地址语音	
AE	音量调节	00 为静音，1F 为最大音量，共 32 级音量调节
AF	循环播放当前语音	循环播放可被 AB、B0 指令打断并失效
AB	停止播放当前语音	
B0	连码播放	

指令格式

唤醒码	起始码	长度	命令码	参数 1	参数 2	...	参数 end	校验码	结束码
00 00	7E	XX	0xA $\color{red}{X}$	Adr 1	Adr 2	..	Adr End	XX	0xEF

7.2.1. 播放对应地址语音（A0）

唤醒码	起始码	长度	命令码	地址(高两位)	地址(低两位)	校验码	结束码
00	7E	05	A0	$\color{red}{XX}$	$\color{red}{XX}$	XX	EF

数据（十六进制）（XX XX）	功能
00 00H	播放第 0 段语音
00 01H	播放第 1 段语音
00 02H	播放第 2 段语音
.....
7F FDH	播放第 32,765 段语音
7F FEH	播放第 32,766 段语音
7F FFH	播放第 32,767 段语音

注意： 1.可以播放 32,767 段语音；

7.2.2. 音量调节指令（AE）

唤醒码	起始码	长度	命令码	音量调节	校验码	结束码
00	7E	04	AE	$\color{red}{XX}$	XX	EF

注意： 1.音量调节共有 32 级(00~1F)，其中 00 音量最小，1F 音量最大；
2.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；

7.2.3. 循环播放当前语音（AF）

唤醒码	起始码	长度	命令码	地址	校验码	结束码
00	7E	04	AF	$\color{red}{XX}$	XX	EF

注意： 1.地址----表示 01 或者 00；01：单曲循环；00:取消单曲循环；
2.执行此命令可循环播放当前段语音，可在语音播放时发送。循环指令执行过程中，可被 AB 命令、普通地址指令、B0 组合指令打断，并失效，需先发播放指令，再发循环播放指令；

7.2.4. 停止当前语音播放（AB）

唤醒码	起始码	长度	命令码	校验码	结束码
00	7E	03	AB	AE	EF

7.2.5. 连码播放 (B0)

唤醒码	起始码	长度	命令码	固定码	固定码	地址 1	地址 1	...	地址 n	地址 n	校验码	结束码
00	7E	XX	B0	FF	NN	X1_H	X1_L	Xn_H	Xn_L	XX	EF

注意: 1.FF/NN----固定码, 预留, 暂时没有作用, 可以直接用 00 00 替代(不可缺);
 2.示例: 假如我们要让芯片依次播放 01/01/02 地址的语音内容, 即连码指令播放;
 指令是: 00 7E 0B **B0** 00 00 **00 01** **00 01** **00 02** BF EF.

7.3. 时钟指令:

命令码	功能	命令码详解	备注
FF 01 01	获取当前时钟	FF 01 01 +R1 R2 R3 R4 R5 R6 R7 R8 R1:年 千位和百位; R2:年 十位和个位; R3:月; R4:日; R5:时; R6:分; R7:秒; R8:星期; (数据都为 16 进制)	00: 星期日 01: 星期一 02: 星期二 03: 星期三 04: 星期四 05: 星期五 06: 星期六
FF 01 02 R1 R2 R3 R4 R5 R6 R7	设置当前时钟信息	R1:年 千位和百位; R2:年 十位和个位; R3:月; R4:日; R5:时; R6:分; R7:秒; (数据都为 16 进制)	
FF 01 03 X1 X2 X3 X4 X5 X6	设置报警/闹钟信息	命令后接 6 个数据, 分别代表: X1:报警组, X2:报警状态, bit7=1, 该报警组使能 X3:星期, bit7=1, 星期使能 X4:日期, bit7=1, 日期使能 X5:时, bit7=1, 小时使能 X6:分, bit7=1, 分钟使能 若设置报警组 1 每天 7:30 报警则发: 7E 0B FF 01 03 01 80 00 00 87 9E B4 EF	星期和日期二选一使能且只能设置每周某一个星期使能 20 组报警组 X1 从 0x01~0x14 注意: X2 的 bit0~bit6 用户可自定义状态, 系统不会改变它的值;
FF 01 04	获取当前报警信息	有报警, 则反馈: FF 01 04 +如上 6 个信息	查询正在发生的报警。
FF 01 05 XX	清除对应定时	XX: 报警组	FF 代表全部
FF 01 06	查询多少组定时	反馈: FF 01 06 +报警组数	
FF 01 07 XX	查询 XX 组定时信息	反馈: FF 01 07 X1 X2 X3 X4 X5 X6	
FF 01 08	清除报警跟倒计时中断信号		成功返回 00, 失败返回 01

FF 01 09 X1 X2 X3 X4	设置倒计时	X1:倒计时控制寄存器 bit7=1, 倒计时使能 bit6=1, 倒计时重载数据使能 X2:倒计时计数器 H X3:倒计时计数器 M X4:倒计时计数器 L	最大支持 4660 小时倒计时, 计时单位秒, 若想停止计时器, 则让 X1 的 bit7≠1。
FF 01 0A	查询倒计时	反馈: FF 01 0A	查询倒计时剩余秒数
FF 01 0B	查询状态寄存器	发送: 00 7E 05 FF 01 0B 10 EF 返回: 7E 05 FF 01 0B X1 X2 X1: bit7=1, 有报警信号; bit6=1, 有倒计时信号; bit5=1, 低电压信号; bit4=1, 正在播放; bit3~0, 预留 X2: 预留	

7.4. 存储指令:

FF 01 0C	读取存储数据	FF 01 0C adr0 adr1+adr19 adr0:表示要读取地址 (00~255) 最多一次读 20 个 byte 数据 返回: FF 01 0C data0 data1+data19 对应地址上放对应数据值 (data0)	例如: 读取位置 00/01/02/03/04 的值 00 7E 0A FF 01 0C 00 01 02 03 04 20 EF 返回: 7E 0A FF 01 0C A0 A1 A2 A3 A4 40 EF 注意: 掉电不记忆
FF 01 0D	写存储数据	FF 01 0D adr0 data0 adr1 data1+adr19 data19 adr0:要写入地址 (00~255); data0:前面地址要写入的数据; 地址+数据成组出现 注意: 最多一次可以写入 20 个 byte 数据	例如: 在位置 00/01/02/03/04 写入 A0/A1/A2/A3/A4 00 7E 0F FF 01 0D 00 A0 01 A1 02 A2 03 A3 04 A4 50 EF

指令格式

唤醒码	起始码	长度	命令码	参数	校验码	结束码
00	7E	XX	FF 01 XX	XX	XX	EF

返回格式

起始码	长度	命令码	参数	校验码	结束码
0x7E	04	XX	XX	XX	EF

注意:

返回参数(XX)	含义
00	代表成功
01	代表失败

7.5. 功能编号指令：

7.5.1. 查询功能编号指令（90）

唤醒码	起始码	长度	命令码	校验码	结束码
00	7E	03	90	93	EF

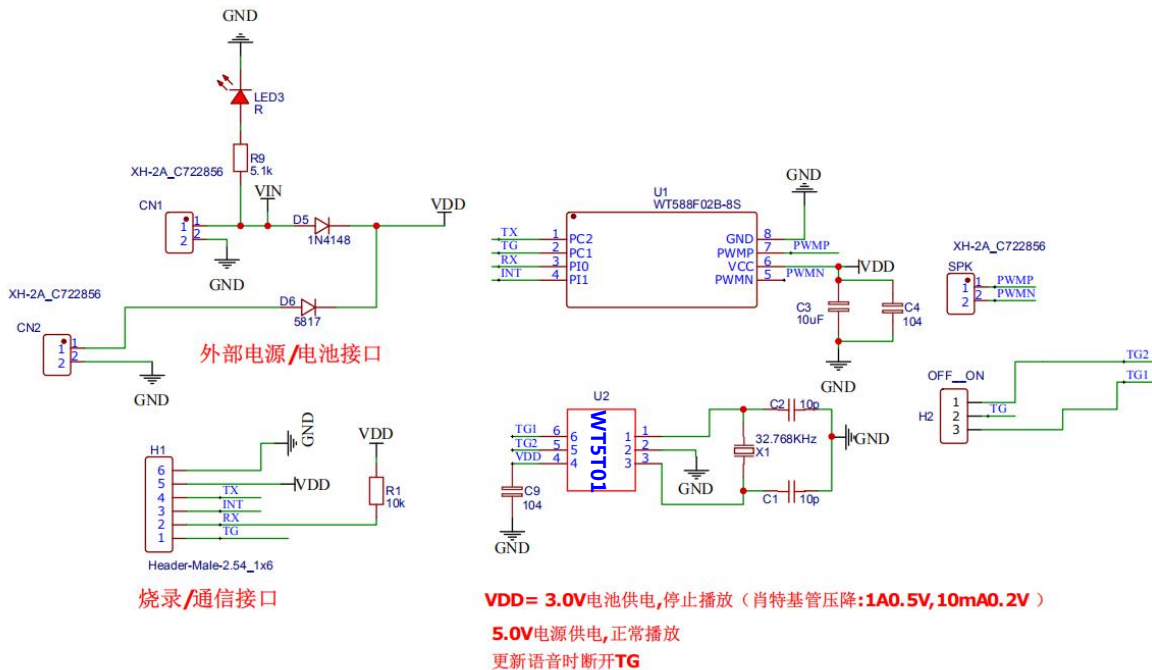
注意：1.返回格式：

起始码	长度	命令码	功能编号	校验码	结束码
7E	06	90	Cx xx xx	xx	EF

2.例如返回：7E 06 90 C0 01 01 58 EF

8. WT588F02B 时钟功能的应用电路图：

8.1. 时钟模块原理应用电路图：



注意：1. VDD 采用 3.0V 电池供电, 停止播放 (肖特基管压降: 1A0.5V, 10mA0.2V) 5.0V 电源供电, 正常播放;

2. 布线时电容 C3 和 C4 靠近 WT588F02B 芯片 VDD 脚 1cm 内, 以增强 WT588F02B 语音芯片的抗干扰能力。

3. 布线时电容 C9 靠近 WT5T01 芯片 VDD 脚 1cm 内, 以增强 WT5T01 芯片的抗干扰能力。

4. 此模块选用的晶振偏差为 ±20ppm。

9. 程序范例

以 STC15 单片机为例:

9.1. 串口/定时器初始化 Init()

```
void Init(void)
{
    /* *****串口 1 初始化***** */
    SCON = 0x50;          //8-bit variable UART
    TMOD = 0x20;          //Set Timer1 as 8-bit auto reload mode
    TH1 = TL1 = -(FOSC/12/32/BAUD); //Set auto-reload vaule
    TR1 = 1;              //Timer1 start run
    ES = 1;               //Enable UART interrupt
    /* *****IO 供电时钟芯片初始化***** */
    P2M0|=0X02;
    P21=1; //推挽输出高电平, 可提供十几毫安电流给时钟芯片
    /* *****定时器 0 初始化***** */
    AUXR &= 0x7F;        //定时器时钟 12T 模式
    TMOD &= 0xF0;        //设置定时器模式
    TL0 = 0x00;          //设置定时初始值
    TH0 = 0xA8;          //设置定时初始值
    TF0 = 0;             //清除 TF0 标志
    ET0 = 1;             //使能定时器中断
    TR0 = 1;             //定时器 0 开始计时
    /* *****看门启动***** */
    WDT_CONTR = 0x04;    // //18.432M : 0.68s
    WDT_CONTR|=0x20;
    EA = 1;              //Open master interrupt switch
}

```

9.2. 串口发送函数 SendData(byte dat)

```
/*-----
Send a byte data to UART
Input: dat (data to be sent)
Output:None
-----*/
void SendData(BYTE dat)
{
    //Wait for the completion of the previous data is sent
    while (busy1);
    busy1 = 1;
    SBUF = dat;          //Send data to UART buffer
}

```

9.3. 串口设置时钟函数 send_uart_rtc()

//input: cmd:命令字节 tempdata[]数据内容, 依命令不同而不同的内容; LEN 数据长度
 //output:依命令不同, 串口返回不同的内容, 如查询实时钟, 则返回 RTC 的年月日时分秒周

```

/*
*****
*****/
void send_uart_rtc(unsigned char cmd,unsigned char len)
{
unsigned char i,j,sum=0;
    j=len+5;
    send2_buf[0]=0x7e;//
    send2_buf[1]=len+5;
    send2_buf[2]=0xff;
    send2_buf[3]=0x01;//
    send2_buf[4]=cmd;

    for(i=0;i<len;i++)send2_buf[5+i]=tempdata[i];

    for(i=1;i<j;i++)sum+=send2_buf[i];
    send2_buf[len+5]=sum;
    send2_buf[len+6]=0xef;
    SendData(0x00);//低电平唤醒串口
    for(i=0;i<(len+7);i++)SendData(send2_buf[i]);
}
/*
*****
*****/

```

9.4. 时钟芯片语音播放函数 send_uart_rtc_play()

//input: LEN 数据长度，每段语音需二字节地址 (0x00 YY) ,可多段语音拼接在一起;

60:	年
61:	月
62:	日
63:	时
64:	分
65:	秒
66:	星期
67:	当前时间是
68:	查询
69:	设置
70:	时间
71:	报警
72:	倒计时
73:	定时

//YY:00-59 数字的语音; ，语音内容可更换

//output: 播放多段语音

```

/*
*****
*****/
void send_uart_rtc_play(unsigned char len) // 7E 05 A0 00 01 A6 EF
{
unsigned char i,j,sum=0;
    j=len+2;
    send2_buf[0]=0x7e;//

```

```

    send2_buf[1]=len+2;
    for(i=0;i<len;i++)send2_buf[2+i]=tempdata[i];
    for(i=1;i<j;i++)sum+=send2_buf[i];
    send2_buf[len+2]=sum;
    send2_buf[len+3]=0xEF;
    SendData(0x00);//低电平唤醒串口
    for(i=0;i<(len+4);i++)SendData(send2_buf[i]);
}
/* *****/

```

9.5. 串口中断接收返回帧

```
void Uart_Isr() interrupt 4
```

```

{
    if (RI)
    {
        RI = 0;                //Clear receive interrupt flag
        rxdata = SBUF;
        rxbuf[rx1cnt++]=rxdata;//接收到缓冲区数组
        rxcnt&=0x3f;           //防止缓冲区溢出 (0-63)
        rx_delayout=0;         //接收超时计数器清零 在定时器 0 中断超时断帧并
    }
}

```

设

```

    if (TI)
    {
        TI = 0;                //Clear transmit interrupt flag
        busy1 = 0;             //Clear transmit busy flag
    }
}

```

9.6. 串口返回帧解码 rtc_uart_decode()

```

/*
*****/
//[17:33:16.591]发→◇7E 05 FF 01 01 06 EF □ for(i=0;i<rx2cnt;i++)SendData(rxbuf[i]);
//[17:33:16.615]收←◆7E 0D FF 01 01 14 15 05 14 09 19 38
04 AE EF
// head len cmdquey 年 年 月 日 时 分 秒 周 校验码 End
// 0 1 2 3 4 5 6 7 8 9 10 11 12 13
void rtc_uart_decode(void)
{
    unsigned char i,j,sum=0,cmd_rtc;
    if(rxflag) //由定时器 0 中断的超时计数器大于超时参数时,标识串口返回帧
    {
        rxflag=0;
        j=rxcnt-2;
        if((rxbuf[0]==0x7e)&&(rxbuf[rxcnt-1]==0xef)) //检测帧头帧尾字节

```

```

{
for(i=1;i<j;i++)sum+=rxbuf[i];
if(sum==rxbuf[rxcnt-2]) //sum 检测校验和是否相等
{
cmd_rtc=rxbuf[4];
switch(cmd_rtc) //根据命令字节转到相应的功能，得到相应的反馈数据
{
case 1:query_rtc_reply();break;//查询时钟命令时，得到的串口返回帧数据
case 2:set_rtc_reply();break;//设置时钟命令时，得到的串口返回帧数据
case 3:set_alarm_reply();break;//设置报警定时器命令时，得到的串口返回帧数据
case 4:query_alarm_reply();break;
case 5:clear_alarm_reply();break;
case 6:query_alarm_groups_reply();break;
case 7:query_alarm_info_reply();break;
case 8:clear_alarmsign_reply();break;
case 9:set_clockdown_reply();break;
case 10:query_clockdown_reply();break;
case 11:clear_clockdownsign_reply();break;
default:break;
}
}
}
}
rx2cnt=0;
} //if(rx2flag)
}
}

```

仅以查询时钟的反馈帧为例，其它子功能类似

//返回帧数据：7E 0D FF 01 01 14 15 05 14 09 19 38 04 AE EF//

void query_rtc_reply(void)

```

{
unsigned char i;
for(i=0;i<8;i++)rtc_data[i]=rx2buf[i+5];//查询实时钟的串口返回帧数据，填入实时钟数据区
rtc_refresh=1;//实时钟数据区更新标志
}
}

```

/* *****/

9.7. 设置倒计时报警并 IO 口输出低电平 Start_Count_Down()

//input: unsigned long int colockdown_data(32 位, 4byte) 1byte 2byte 3byte 4byte(low byte)

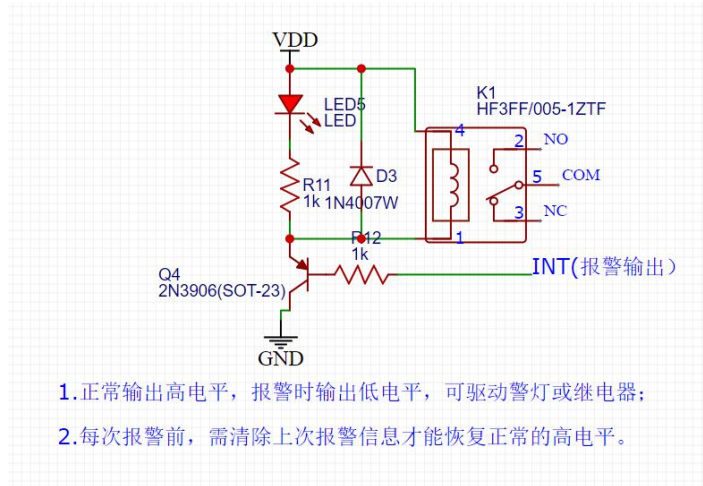
//

//↑
时秒数

↑ ↑
高字节第四位为倒计时启动位 低三字为倒计时秒数

//output: 倒计时秒数到, 输出低电平 (注意: 前提是要用指令消除上次报警输出低电平)

// 下图为报警 INT 脚输出驱动报警灯或警号的电路原理图

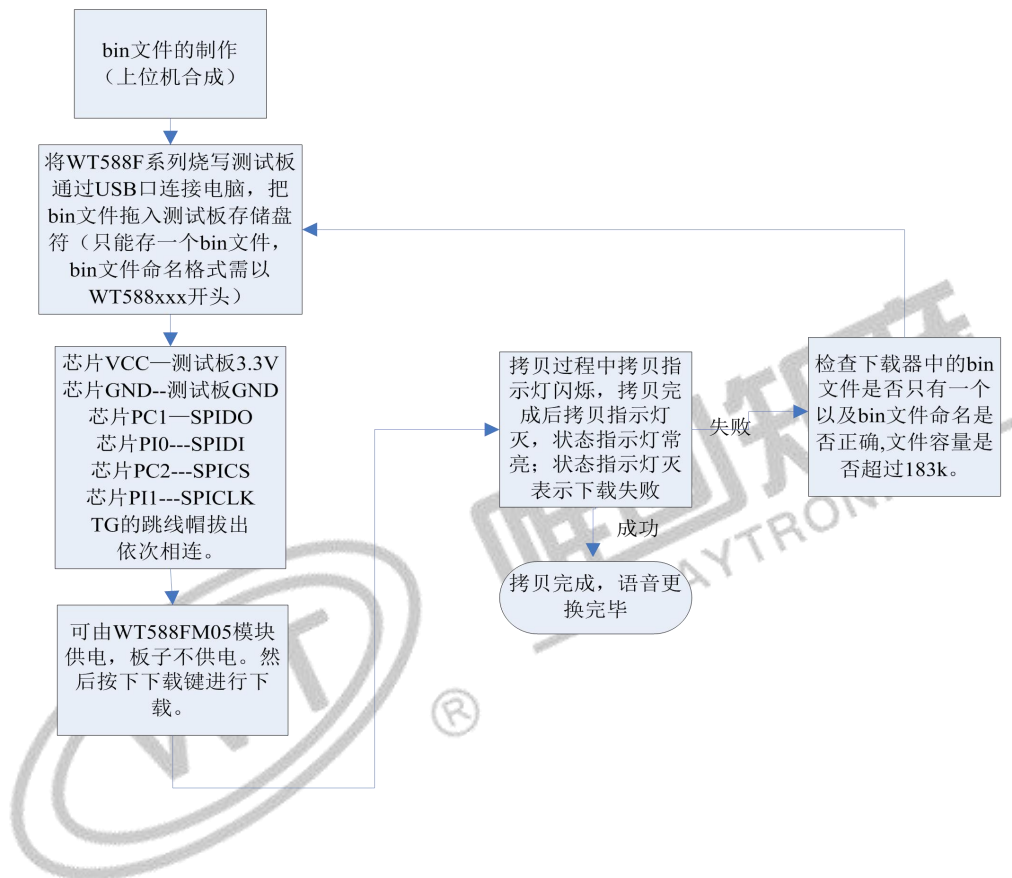


```
void Start_Count_Down(void)
{
    unsigned char i;
    send_uart_rtc(0x08,0); //清除上次定时报警命令
    Delay10ms();
    send_uart_rtc(0x0b,0); //清除上次倒计时报警命令
    Delay10ms();
    i=(unsigned char)(colockdown_data>>24);
    tempdata[0]=i|0x40;//启动倒计时
    for(i=1;i<4;i++)tempdata[i]=(unsigned char)(colockdown_data>>(8*(3-i)));//填充倒计时
秒数
    send_uart2_rtc(0x09,4);//发送启动倒计时命令
    Delay100ms();
}

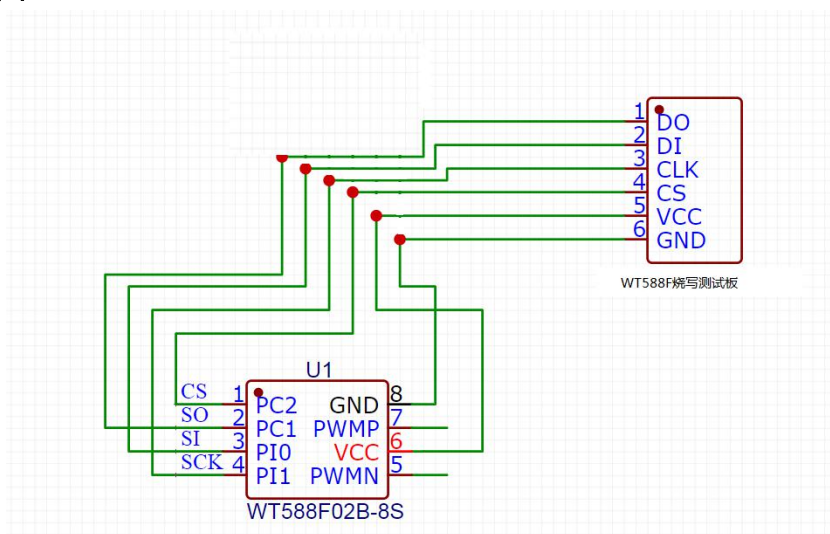
/* *****
/*-----
```

10. 客户在线下载器

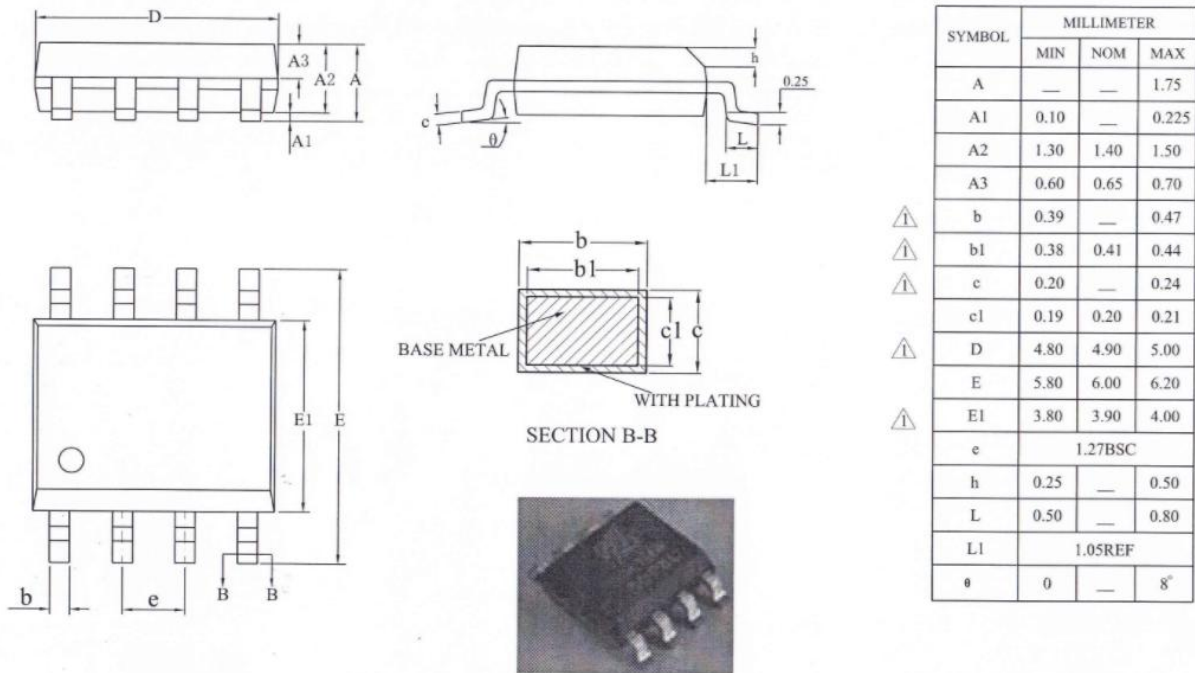
将 bin 文件存入在线下载器 WT588F 系列烧写测试板中，通过烧写测试板烧写测试可更换语音，可用于生产或客户端设备语音的更换，使用方便、快捷、操作简单。(WT588F 系列烧写测试板已有详细说明，可联系业务员提供。)



10.1. 下载原理图



11. 封装管脚图



深圳唯创知音电子有限公司（原名：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。

总公司名称：深圳唯创知音电子有限公司

电话：0755-29605099 0755-29606621 0755-29606993

传真：0755-29606626

全国统一服务热线：4008-122-919

E-mail：WT1999@waytronic.com

网址：<http://www.waytronic.com>

地址：广东省深圳市宝安区福永镇福安机器人产业园 11 栋 4 楼

分公司名称：广州唯创电子有限公司

电话：020-85638557

E-mail：864873804@qq.com

网址：www.w1999c.com

地址：广州市花都区天贵路 62 号 TGO 天贵科创 D 座 409 室

分公司名称：北京唯创虹泰科技有限公司

电话：010-89756745

传真：010-89750195

E-mail：BHL8664@163.com

网址：www.wcht1998.com.cn

地址：北京昌平区立汤路 186 号龙德紫金 3 号楼 902 室