



单相电能计量 SoC 芯片 FAQ-V98xx 系列

应用笔记

版本历史

时间	版本	修改内容
2016-1-13	V1.0	V9801S/V9811S/V9811A/V9821 FAQ 文档正式发布
2016-1-21	V1.1	修改 2.12、3.12 描述，增加 4.4：JTAG 第 10 脚应用说明
2016-2-2	V1.2	增加 2.17：晶体停振检测判断
2016-3-7	V1.3	增加 2.18：PLL 切换流程
2016-3-21	V1.4	增加 2.19：如何设置 RTC 多秒唤醒 增加 2.20：程序无法执行到 main 函数
2016-4-11	V1.5	增加 1.14：V9821 的 M2 引脚测量电池电压漏电问题
2016-7-29	V1.6	增加无功计量时，采样信号大小计算
2016-11-21	V1.7	增加电流不平衡应用，采样电路设计
2017-6-8	V2.0	增加若干技术问题
2017-8-1	V2.1	增加液晶关闭时 GPIO 推荐配置，增加电池电压测量 BAT 串联二极管

目录

版本历史	1
1. 硬件	4
1.1. 采样电路参考设计	4
1.1.1. 电流 IA 通道锰铜采样：	4
1.1.2. 电流 IB 通道 CT 采样：	5
1.1.3. 电压 U 通道电阻分压采样：	6
1.2. 5V/3.3V 电源供电区别？	6
1.3. 外部 32768 晶振是否需要接负载电容？	7
1.4. BAT 引脚是否参与芯片供电？	7
1.5. BAT 如何测量 6V 电池电压？	7
1.6. BAT 引脚测量电池电压偏高？	7
1.7. 休眠功耗偏大，且随着 VDD5 输入电压升高，休眠功耗越大？	7
1.8. GPIO 默认是否接上拉电阻？	7
1.9. GPIO 驱动能力？	8
1.10. 液晶管脚需要并联 220pF 电容？	8
1.11. 液晶显示异常问题？	8
1.11.1. 液晶上电显示正常，下电显示花屏？	8
1.11.2. 液晶显示部分段码显示异常？	8
1.11.3. 带电池的表上电液晶显示浓淡不均，3~5s 后消失/上电时，某些段码提前闪烁？	8
1.11.4. 液晶显示浓淡不均，且持续不变？	8
1.12. 没有用到的芯片引脚如何处理？	8
1.13. JTAG 端口复用存在无法烧写程序现象？	8
1.14. 远红外通讯时系统复位？	8
1.15. V9821 的 M2 引脚测量电池电压漏电问题？	9
1.16. 电流不平衡应用时，采样电路设计	9
2. 软件	9
2.1. 休眠功耗异常，概率性出现 100uA~300uA 的跳动？	9
2.2. 休眠功耗异常，带 RTC 脱机下载后，功耗有概率出现 80uA？	9
2.3. 芯片无法进入休眠状态，电池功耗到达 mA 级？	9
2.4. V9821 用上全部 SEG 时，发现 SEG34/SEG35/SEG39 无法点亮？	9
2.5. 如何正确执行喂狗操作？	10
2.6. 如何正确处理中断标志位？	10
2.7. M 通道测电池时，电池功耗会增大？	11
2.8. M 通道测温度时，发现温度测量不准？	11
2.9. 3.3V 电源系统发现电池漏电问题？	11
2.10. 3.3V 电源系统 LDO33 输出电压调节失效？	11
2.11. UART2 模块扩展标志位与内核标志位相矛盾？	12
2.12. 如何使用掉电中断？	12
2.13. 读取 RTC 时间数据异常？	12
2.14. 如何实现全温度范围 RTC 误差小于 5ppm（日计时误差小于 0.5s/day）？	13
2.15. 偶尔存在 IAP 无法写入的状态？	14
2.16. IAP 操作完成后程序异常？	14
2.17. 软件处理增加晶体停振检测判断？	14

2.18.	如何正确切换 PLL ?	14
2.19.	如何设置 RTC 多秒唤醒 ?	14
2.20.	程序无法执行到 main 函数 ?	15
2.21.	带电池 5V 供电系统应用, PDDET 配置注意事项 ?	15
3.	计量	16
3.1.	如何计算功率/有效值寄存器理论值 ?	16
3.1.1.	电压/电流有效值	16
3.1.2.	有功/无功功率	16
3.2.	如何将功率/有效值寄存器数值转换成实际值 ?	17
3.3.	正常计量时电流有效值为降频计量 (800K) 时的 3 倍 ?	17
3.4.	电压/电流有效值均为正值, 有功功率却为负值 ?	18
3.5.	V9811A 样机上电时, 发现反向灯点亮 ?	18
3.6.	有无功计量时, 无功功率反向问题 ?	18
3.7.	无功计量时, I _{max} 点误差异常 ?	18
3.8.	频率影响量实验时, 无功计量误差偏大 ?	18
3.9.	根据公式计算比值出错 ?	18
3.10.	V9821 样机自动校表时, 随机出现角差校正异常 ?	18
3.11.	V98XX 计量误差无法校正, 程序无法烧录 ?	19
3.12.	V9811 方案芯片升级为 V9811S 时, 角差无法校正 ?	19
3.13.	常数计量应用, IA 通道功率正向时, 计量正确; 功率反向时, 计量异常 ?	19
3.14.	50Hz/60Hz 电网应用时的区别 ?	20
3.15.	如何调整电表的高低温误差 ?	20
3.16.	调整 BGP 控制寄存器后, 影响温度测量 ?	20
4.	仿真	20
4.1.	在线仿真时, 系统状态寄存器 PWRDN 与 PWRUP 矛盾 ?	20
4.2.	仿真器供电时程序正常执行, 220V 供电时, 系统没有反应 ?	20
4.3.	仿真器能否调试休眠模式下代码运行 ?	21
4.4.	仿真器 JTAG 第 10 脚的用法 ?	21
4.5.	SP402 开启 RTC 校正功能时, 脱机下载失败 ?	21
4.6.	SD502 的 USB 驱动安装失败 (unknow device) ?	21
4.7.	报错: Fail to open #include file 'iar_check_symbols.h' ?	21
4.8.	报错: Page2 must be empty?	21
4.9.	工程建完后, 编译通过, 代码执行异常 ?	21
4.10.	KEIL 代码移植到 IAR 时数据处理相关问题 ?	22
5.	附录	22
5.1.	SetPLL 标准流程	22

1. 硬件

1.1. 采样电路参考设计

设计准则一： 芯片引脚接收到的最大采样信号值（峰值），单端信号不应超过 200mV，差分信号不应超过 400mV；

设计准则二： 芯片引脚接收到的最大采样信号值（峰值），乘上增益后不应超过基准电压值 1.1V；

另外，采样电路的 RC 网络推荐 360Ω+33nF，涉及到降频计量（800K）时，电流采样 RC 网络推荐 100Ω+100nF。

采样电路参考设计如下：

1.1.1. 电流 IA 通道锰铜采样：

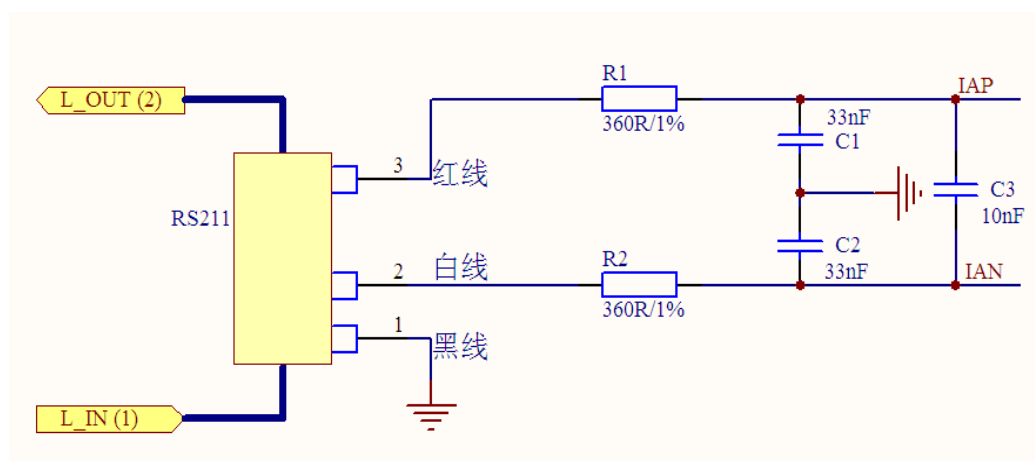


图 1.1-1 锰铜采样推荐电路

假定表计规格为 5 (60) A，220V，锰铜片 RS211 为 300μΩ，A 通道一般设置模拟增益为 32 倍，数字增益为默认 1 倍，考虑最大信号为 1.2Imax，计算如下：

满足设计准则一：

$$1.2 \times 60A \times 300\mu\Omega \times 1.414 = 30.5424mV < 200mV$$

满足设计准则二：

$$30.5424mV \times 32 \times 1 = 997.3568mV < 1.1V$$

若为贴片锰铜或者 2 线锰铜时，PCB 走线需特别注意。锰铜一端（2 线锰铜时，其中一个焊盘）接地，但仍需单独接线至芯片 IAN 引脚。如下图所示：高亮走线必须保留，采样电阻一端不能直接接地，否则会影响计量。

注意：无功计量时，由于在无功运算中，信号会放大 1.56 倍。因此在计算输入信号是，要将上面推荐值缩小 1.56 倍。

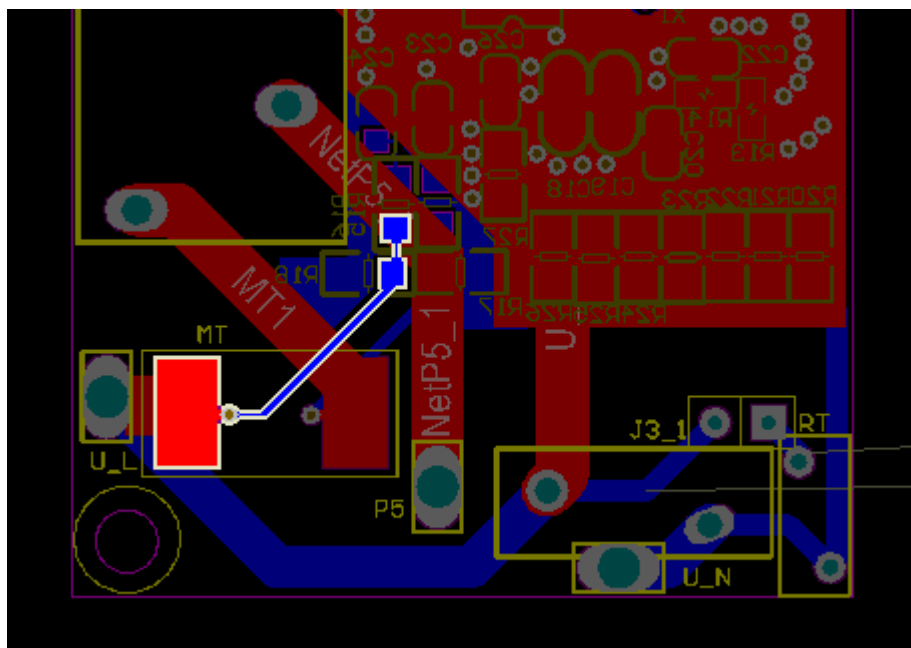


图 1.1-2 贴片锰铜走线方式

1.1.2. 电流 IB 通道 CT 采样：

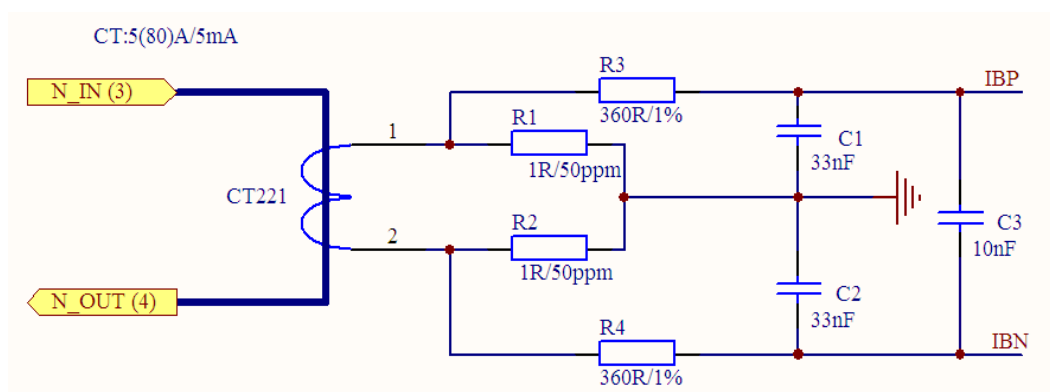


图 1.1-3 CT 采样推荐电路

假定表计规格为 5 (60) A, 220V, CT 规格为 5 (80) A/5mA, 差分采样, B 通道一般设置模拟增益为 4 倍, 数字增益为默认 1 倍, 考虑最大信号为 1.2Imax, 计算如下:

满足设计准则一:

$$1.2 \times \frac{60A}{5A} \times 5mA \times (1\Omega + 1\Omega) \times 1.414 = 203.616mV < 400mV$$

满足设计准则二:

$$203.616mV \times 4 \times 1 = 814.464mV < 1.1V$$

1.1.3. 电压 U 通道电阻分压采样：

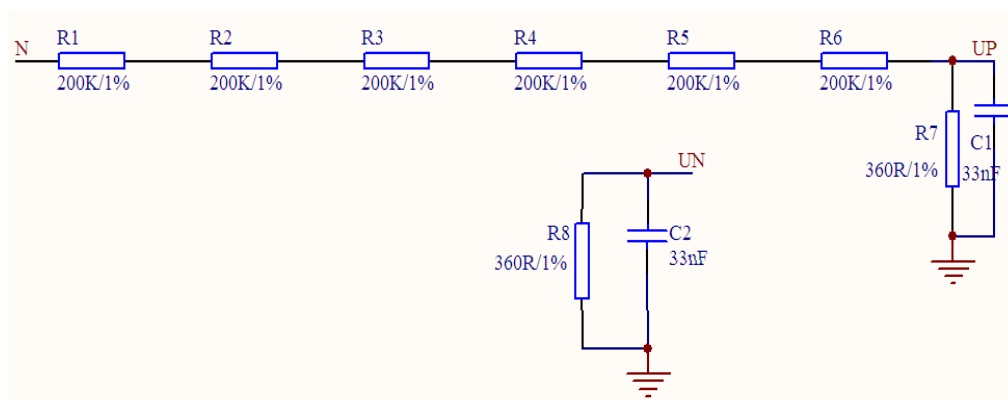


图 1.1-4 电压采样推荐电路

假定表计规格为 5 (60) A , 220V , 6 个 200K Ω 电阻分压 , 360 Ω 电阻采样 , U 通道一般设置模拟增益为 1 倍 , 数字增益为默认 1 倍 , 考虑最大信号为 1.2Un , 计算如下：

满足设计准则一：

$$1.2 \times \frac{220V}{6 \times 200K\Omega + 360\Omega} \times 360\Omega \times 1.414 = 111.955mV < 200mV$$

满足设计准则二：

$$111.955mV \times 1 \times 1 = 111.955mV < 1.1V$$

1.2. 5V/3.3V 电源供电区别？

V98XX 系列电源供电系统有以下特点：

- 1、支持 5V/3.3V 外电源供电；
- 2、芯片内部模拟电路和 GPIO 口等由 3.3V-LDO 供电；
- 3、芯片内部数字电路、PLL、RTC 和晶体起振电路等由 DVCC 供电；

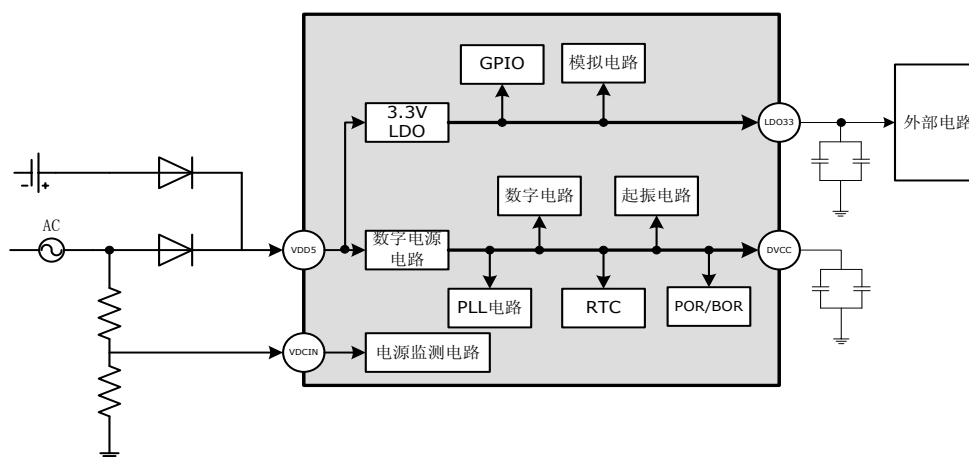


图 1.2-1 电源供电系统

5V 外电源供电时，经过内部 3.3V-LDO，因此 GPIO 口高电平为 3.3V，并非 5V，硬件设计时需要特别注意；3.3V 外电源供电时，建议将 VDD5 与 LDO33 引脚短接，旁路掉内部 LDO 模块（此时液晶驱动电压只支持 3V 和 3.3V）。

1.3. 外部 32768 晶振是否需要接负载电容？

芯片内置起振电路已有负载电容，容值为 12.5pF，同时用户可根据需要通过配置晶振控制寄存器 1（CtrlCry1，0x2860）的 Bit4，调节起振电路负载电容大小。

1.4. BAT 引脚是否参与芯片供电？

BAT 引脚仅用于测量电池电压或者其他外部直流电压信号，待测电压信号（V）的范围为-200mV~3.8V，不参与供电。V98XX 系列唯一供电引脚为 VDD5。

1.5. BAT 如何测量 6V 电池电压？

关闭芯片内部电阻分压及电容分压模式，采用外部电阻分压，建议 1M 和 200K 分压，分压后 BAT 脚电压必须小于 1.1V。

1.6. BAT 引脚测量电池电压偏高？

BAT 引脚测量电池电压，硬件设计建议电池正极与 BAT 引脚之间串接 1 个二极管，保证 BAT 引脚电压与 LDO33 电压差值不超过 0.3V，否则有可能会造成电池电压测量偏高。

1.7. 休眠功耗偏大，且随着 VDD5 输入电压升高，休眠功耗越大？

BAT 引脚直接接地导致系统漏电。若产品不需要用到电池，BAT 引脚必须悬空。

1.8. GPIO 默认是否接上拉电阻？

V98XX 系列 GPIO 默认为高阻态，并非默认接上拉电阻，可通过寄存器配置成输出高电平或者低电平，配置成输入状态时，需根据应用情况，外接上拉/下拉电阻。所有 I/O 端口均没有内置上拉/下拉电阻。

1.9. GPIO 驱动能力？

GPIO 端口由 3.3V-LDO 供电，GPIO 上消耗的总电流不能大于 3.3V-LDO 的最大驱动能力 30mA。单个 GPIO 端口长时间大于 10mA 的电流会对芯片造成损坏，12mA 电流短时间内不会对芯片造成损坏。

1.10. 液晶管脚需要并联 220pF 电容？

液晶管脚 (COM+SEG) 并联 220pF 电容，可改善 EMC 性能，建议添加，或至少预留焊接位置。

1.11. 液晶显示异常问题？

1.11.1. 液晶上电显示正常，下电显示花屏？

测量 LDO33 引脚电平，上下电电平是否一致。LCD 驱动电压来自内部 3.3V-LDO，LDO33 电平变化会影响 LCD 显示效果。

1.11.2. 液晶显示部分段码显示异常？

检查 COM，SEG 口连接是否正常，软件检查 COM，SEG 口是否配置正常，可能存在 SEG 配置成了 GPIO 功能的情况。

1.11.3. 带电池的表上电液晶显示浓淡不均，3~5s 后消失/上电时，某些段码提前闪烁？

建议下电休眠时，将液晶复用 GPIO 配置成输出低，这样做可以减少液晶的直流偏置，不影响液晶上电显示效果。

1.11.4. 液晶显示浓淡不均，且持续不变？

一般出现在液晶段码较粗的表型，与液晶负载过大，影响芯片液晶驱动波形，建议更换液晶。

1.12. 没有用到的芯片引脚如何处理？

数字部分引脚（如 SEG，GPIO）建议悬空（软件配置成输出 0 的状态），未用到的 ADC 通道（如 IBP/IBN）建议接地，BAT 建议悬空。没用到的 SEG 必须配置成 GPIO 口，并输出 0。

1.13. JTAG 端口复用存在无法烧写程序现象？

复用的 JTAG 口最好不要接电容，无法避免时，尽量选择容值较小的电容，建议 1nF 以下，否则会影响 JTAG 烧写 FLASH 的时序，造成无法下载问题。

1.14. 远红外通讯时系统复位？

红外发射电路电源选择了 LDO33 输出的 3.3V，芯片内部 LDO33 模块带载能力只有 30mA，全部外设所需驱动电流大于 30mA，导致系统复位。

1.15. V9821 的 M2 引脚测量电池电压漏电问题？

V9821 未提供 BAT 引脚，但提供了 M 通道信号输入引脚（M2，Pin42），M2 引脚输入的电压信号范围为-200mV~3.4V。因此 M2 引脚测量电池电压时，需要串联一个二极管进行降压，若直接与电池正极相连，会造成 30uA~120uA 的漏电。

1.16. 电流不平衡应用时，采样电路设计

基于 V98XX 设计的产品，若需要进行电流不平衡判断时，需注意 L 路、N 路电流采样电路硬件设计，确保两路信号乘以内部增益后基本一致。若 L 路/N 路信号相差过大，可能会超出比差校正范围（-50%~50%）。

以常见规格，5（60A），220V，1200imp/kwh 为例，推荐电路如下：

L 路锰铜采样，锰铜规格为 250uΩ，模拟增益开 32 倍；N 路 CT 采样，CT 规格为 5（60A）/2.5mA，采样电路为 2 个 2Ω电阻，差分采样，模拟增益开 4 倍；

2. 软件

2.1. 休眠功耗异常，概率性出现 100uA~300uA 的跳动？

与休眠状态下 GPIO 配置有关。某些 GPIO 口在休眠状态下配置成了输入状态，且引脚未连接任何器件，为悬空态。外部干扰信号抬高了休眠功耗，休眠状态下需将未用到的 GPIO 口配置成输出 0。

2.2. 休眠功耗异常，带 RTC 脱机下载后，功耗有概率出现 80uA？

带 RTC 脱机下载后，P9.3/P1.3/P1.4 会配置成输出状态，休眠前需根据实际应用配置这 3 个 GPIO 口。若为按键输入，则配置为输入使能，禁止输出；若为悬空，则配置为输出 0；

2.3. 芯片无法进入休眠状态，电池功耗到达 mA 级？

与 VDCIN 引脚悬空有关。系统若采用 GPIO 判断掉电，需将未用到的 VDCIN（掉电监测输入引脚）拉低。

2.4. V9821 用上全部 SEG 时，发现 SEG34/SEG35/SEG39 无法点亮？

采用 6COM，1 字节对应 1 个 SEG 的模式下，LCDM29，LCDM30 显存地址不连续。软件显示处理时，采用指针方式连续指向显存地址，刷新显存内容，SEG34/SEG35/SEG39 相对应的显存未正确写入数值，因此无法点亮；LCDM30 开始，需重新定义显存地址。

0x2C19	LCDM25	S25	-	-	SEG25
0x2C1A	LCDM26	S26	-	-	SEG26
0x2C1B	LCDM27	S27	-	-	SEG27
0x2C1C	LCDM28	S28	-	-	SEG28
0x2C1D	LCDM29	S29	-	-	SEG29
0x2C28	LCDM30	S30	-	-	SEG30
0x2C29	LCDM31	S31	-	-	SEG31
0x2C2A	LCDM32	S32	-	-	SEG32
0x2C2B	LCDM33	S33	-	-	SEG33
0x2C2C	LCDM34	S34	-	-	SEG34
0x2C2D	LCDM35	S35	-	-	SEG35

图 2.4-1 液晶显存地址

2.5. 如何正确执行喂狗操作？

不能在定时器中喂狗，喂狗操作采用连续喂 2 次狗的方式即可，不需要开关中断。

```
#define CLRWDT()    { WDTEN=0xA5;WDTCLR=0x5A; WDTEN=0xA5;WDTCLR=0x5A; }
```

如果程序运行超过 1.5s，未执行喂狗操作，系统会发生 WDT 溢出复位。由于 WDT 时钟来源于内部 RC 振荡器，存在一定误差，因此实际执行喂狗操作间隔推荐为 0.5s。

2.6. 如何正确处理中断标志位？

V98xx 系列中断机制为发生相应事件，即使在中断使能没有打开的情况下，也会将对应中断标志位置起，有利于采用查询方式处理中断标志位。建议将中断标志赋值给变量进行判断，并与中断使能做比较，之后进入中断处理，需保证首先清中断标志位。

```
__interrupt void UATRAndRTCInterrupt(void)
{
    u8 temp_ifg;

    if(EXIF&BIT5)
    {
        temp_ifg = ExInt2IE;          // 防止对未使能中断的位操作

        temp_ifg &= ExInt2IFG;

        if(temp_ifg&BIT7)             //脉冲中断
        {
            ExInt2IFG&=(~BIT7);

            //用户代码,CF2 中断处理函数

            CF2Event();
        }
    }
}
```

```

if(temp_ifg & BIT6)    //RTC 中断
{
    ExInt3IFG &= (~BIT6);

    //用户代码，RTC 中断处理函数

    RTCEvent();
}
}

```

2.7. M 通道测电池时，电池功耗会增大？

M 通道测量电池电压时，有 2 种分压方式。

表 7-11 ADC 控制寄存器 5 (CtrlADC5, 0x2863)			
0x2863, R/W, ADC 控制寄存器 5, CtrlADC5			
位		默认值	功能说明
bit7	保留	0	为保证系统正常工作，必须保证为 1。
Bit6	保留	0	为保证系统正常工作，必须保证为 0。
Bit5	GDE4	0	测量通道 (M) 的模拟增益控制。 0: $\times 1$; 1: $\times 1/4$ 。
Bit4	RESDIV	0	将该位置 1 使能 M 通道内部电阻分压电路，分压系数为 0.25。 该电路默认关闭。

图 2.7-1 M 通道分压配置

第一种为配置 RESDIV 为 1，开启内部电阻分压电路，此时测电池电压时，会额外产生 100uA 左右的功耗，若需要长时间判断电池电压，使用时建议每分钟测量一次。

第二种为配置 GDE4 为 1，开启 M 通道模拟增益 1/4，此时测量电池电压不会产生功耗，BAT 引脚悬空时，仍会测量到电池电压，不推荐使用。

2.8. M 通道测温度时，发现温度测量不准？

将 BGP 控制寄存器 (CtrlBGP, 0x2862) 的 Bit0 置 0，开启斩波去直流失调功能，有助于改善测温性能。

2.9. 3.3V 电源系统发现电池漏电问题？

BAT 引脚电压为 3.6V，高于 VDD5 电源输入电压 3.3V，发生漏电问题。系统由 3.3V 电源供电时必须将 CtrlLDO (0x2866) 的 Bit7 置 1。

2.10. 3.3V 电源系统 LDO33 输出电压调节失效？

3.3V 电源系统应用时，需将电源输入 VDD5 与 LDO33 短接，此时 LDO 控制寄存器 (CtrlLDO, 0x2866) 无法调节 LDO33

输出电压。LDO33 电平会影响外设工作，需要注意此时选用的液晶屏，标称工作电压必须为 3.0V 或者 3.3V，否则会影响显示效果。

2.11. UART2 模块扩展标志位与内核标志位相矛盾？

ExInt2IFG 发送标志位已经置位，但是相应的 TI 没有置位；或者 ExInt2IFG 接收标志位已经置位，但是 RI 标志位没有置位，需要将 ExInt2IFG 清除掉，并且在程序中初始化时，必须将 ExInt2IFG 接收发送标志及 RI, TI 都做清零处理。

```
if(ucTemp&BIT0)          //UART2 发送中断
{
    SCON2&=(~BIT1);    //清内核标志位
    ExInt2IFG&=(~BIT0); //清扩展标志位
    Uart2_Transmit();
}

if(ucTemp&BIT1)          //UART2 接收中断
{
    SCON2&=(~BIT0);    //清内核标志位
    ExInt2IFG&=(~BIT1); //清扩展标志位
    Uart2_Receive();
}
```

2.12. 如何使用掉电中断？

掉电中断判断是按照 mcu 当前的 2 个时钟周期来计算的，因此根据频率不同判断时间不同，且滤波不够充分可能存在误判，Systate 标志则是以 8 个 OSC 时钟滤波判断，更加准确。

掉电中断的优先级是最低，因此使用时需要将掉电中断的优先级设为 1，且保证使用的寄存器组与其他优先级 0 的中断不同，避免嵌套的发生。

由于 main () 函数所使用的寄存器组为 0，所以中断程序里如果明确使用哪个寄存器组（例如 #pragma register_bank=2），那么所使用的工作寄存器组不能为 0；如果是缺省，则无所谓，但是会浪费栈的资源和时间。

2.13. 读取 RTC 时间数据异常？

读取 RTC 时间数据时，MCU 应先读寄存器 RDRTC SFR (SFR 0xDA)，等待至少 5 个 OSC 时钟周期后，RTC 时间数据被锁定，MCU 再对所有计时寄存器（时间设置寄存器）进行读操作，从而得到当前时间数据。

```
void GetExtRTC(void)
{
    volatile uint8 RTCFLAG ;
    /*读寄存器每次读都必须要先读*/
    RTCFLAG=RTCLATCH;
    /* 此处必须等待至少 5 个 OSC 时钟,确保是 5 个 OSC 时钟,不是 5 个 NOP*/
    DelayOSC(5);
    gs_ClkTmp.ucSecond = RTCSC;
    gs_ClkTmp.ucMinute = RTCMiC;
    gs_ClkTmp.ucHour   = RTCHC;
```

```

gs_ClkTmp.ucWeek   = RTCWC;
gs_ClkTmp.ucDay     = RTCDC;
gs_ClkTmp.ucMonth   = RTCMoC;
gs_ClkTmp.ucYear    = RTCYC;
}

```

2.14. 如何实现全温度范围 RTC 误差小于 5ppm (日计时误差小于 0.5s/day) ?

实现全温度范围 RTC 误差小于 5ppm，根据实时温度对晶振进行补偿，需要满足以下 3 点：

1、 温度测量准确

芯片内置温度传感器，可根据数据手册“测量温度”章节，开启 M 通道测量温度功能，得到 M 通道直流数据值寄存器 (DATADM，0x10CF) 中温度值，再根据下面公式计算出实际温度大小 (T'，单位为℃)。

$$T' = \frac{B \times (D \times \frac{x_0}{2^{16}} + C)^{\frac{1}{2}} - A}{E}$$

其中， x_0 为寄存器 DATADM (0x10CF) 的读数 (十六进制)；A/B/C/D/E 均是温度曲线参数，用户可在 FLASH 地址为 0x420~0x430 的存储区读取 (以小端模式存储)。

T' 为芯片温度值，与实际环境温度存在一定偏差 (ΔT)，用户可在 FLASH 地址为 0x480 的存储区读取常温偏移值 ΔT (以大端模式存储)，是实际常温偏移值的 10 倍。

因此最终温度值应为：

$$T = T' + \frac{\Delta T}{10}$$

2、 晶体常温偏移

由于 32K 晶振常温下振荡频率并不是完全精确的 32768Hz (晶振频率容许偏差 $\pm 5\text{ppm}$)，因此需要对其做常温偏移补偿。用户可在 FLASH 地址为 0x48C 的存储区读取 RTC 常温偏移值 Delta。

3、 晶体二次校准系数 Bpara

不同型号的晶体，二次校准系数不同。万高推荐采用的晶体型号为精工 VT-200-F，其对应的二次校准系数 Bpara 值已经存放在 FLASH 地址为 0x498~0x4A8 的存储区内 (以大端模式存储)。若采用其他型号的晶体，需微调晶体二次校准系数。

4、 计算不同温度下 RTC 补偿值

用户从 FLASH 中取得相应变量后，代入下面公式，计算出不同温度下的 RTC 补偿值，写入 RTC 校正寄存器，实现全温度范围 RTC 误差小于 5ppm。

$$C = ((\text{float})Bpara) * (T - ((\text{float})Ti)) * (T - ((\text{float})Ti)) / 1000000 + Delta$$

其中：

Bpara：晶体二次校准系数；

T：当前温度值；

Ti：晶振顶点温度（一般为 24℃，用户可在 FLASH 地址为 0x4DA 的存储区读取，大端模式存储）

Delta：晶体常温偏移值

注意：公式中 C、Bpara、Delta 均扩大了 10 倍，RTC 校正分辨率在 0.1ppm，因此写入 RTC 校正寄存器的值应为 C/10，参考代码可向 FAE 索取。

2.15. 偶尔存在 IAP 无法写入的状态？

IAP 操作前，除开启 IAP 中断外，其他中断都必须关闭。IAP 完成后再开启。避免在 IAP 执行过程中被其他中断打断。

2.16. IAP 操作完成后程序异常？

代码的擦和写操作是在芯片挂起状态下运行，IAP 操作完成后需要靠中断退出挂起状态。在擦写操作前需要配置 IAP 中断，ExInt2IE=BIT6; EIE=BIT0; EA = 1;

2.17. 软件处理增加晶体停振检测判断？

当外部晶体停振时，RC 振荡电路自动将使用 OSC 时钟的电路（包括 PLL 电路）的时钟源切换到 RC 时钟，并激励 OSC 振荡电路恢复振荡。当晶体恢复振荡以后，使用 OSC 时钟的电路（包括 PLL 电路）的时钟源自动切换回到 OSC 时钟。用户可通过 OSC 位（bit7，ANState，0x286B）判断晶体是否停振。

表 8-16 OSC 时钟状态查询寄存器				
0x286B, R, 模拟电路状态寄存器, ANState				
位		R/W	默认值	功能说明
bit7	OSC	R	0	指示当前的 OSC 时钟状态。 0: 表示 OSC 时钟正常工作; 1: 表示晶体停振, 所有使用 OSC 时钟的电路 (包括 PLL 电路) 的时钟源已被切换为 RC 时钟。

图 2.17-1 OSC 时钟状态

用户在编写软件时，须在主循环中增加 OSC 时钟状态判断。若发现晶体停振，立即给出相应的异常指示，快速定位问题，方便在研发生产过程中及时挑选出晶体异常（受损、停振等）的表计，避免影响产品质量。

2.18. 如何正确切换 PLL？

在 V98XX 中，晶振电路产生的 OSC 时钟作为基准时钟输入 PLL 电路，PLL 电路将其倍频产生 PLL 时钟，为 MCU（MCU_PLL）、电能计量模块（MEA_PLL）和 ADC（ADC_PLL）提供输入时钟，SetPLL 标准流程见[附录 5.1](#)。

2.19. 如何设置 RTC 多秒唤醒？

系统进入浅睡眠状态时，RTC 依然可以正常工作，并可以提供 1 日/1 小时/1 分/1~64 秒/500ms/250ms/125ms/62.5ms 间隔的唤醒复位。其中天唤醒、小时唤醒以及分唤醒功能操作比较简单，不再赘述，重点说明多秒（2~64 秒）唤醒操作注意事项。

表 19-4 RTC 秒唤醒间隔寄存器 (SECINT, SFR 0xDF)

SFR 0xDF, R/W, RTC 秒唤醒间隔寄存器, SECINT				
位		R/W	默认值	功能说明
Bit7	-	R/W	0	保留
Bit6	-	R/W	0	必须先将寄存器 INTRTC (SFR 0x96) 设置为 0x07, 再将该位置 1, 才能通过该寄存器的 bit[5:0] 设置秒唤醒间隔时间。
Bit[5:0]	SEC	R/W	0	设置秒唤醒间隔时间, 实际秒唤醒间隔时间为 (bit[5:0]+1)s, 其中 bit[5:0] 为 1~63 (十进制)。 当 bit[5:0] 为 0 (十进制) 时, RTC 唤醒间隔为 62.5ms。

图 2.19-1 RTC 秒唤醒间隔寄存器

一、 SECINT 寄存器(SFR 0xDF)设置固定多秒唤醒时间：

设置固定的多秒唤醒时间,除了芯片上电第一次唤醒时间不确定外(与设置休眠唤醒之前的程序代码执行时间有关),之后的休眠唤醒时间是完全准确的。

二、 SECINT 寄存器(SFR 0xDF)设置不同多秒唤醒时间,进行 2~64 秒之间切换：

a、小值切换到大值：如果在休眠唤醒后的 1 秒之内设置新的唤醒阈值,则新的唤醒时间是大值与小值的差值；如果在休眠唤醒后的 1 秒之后再设置新的唤醒阈值,则新的唤醒时间是设置的大值时间,而不是大值与小值的差值。

例子:设置 3 秒唤醒,等待 3 秒休眠唤醒后,接着在唤醒后的 1 秒内继续设置 10 秒唤醒,则后面的唤醒时间为 7 秒(前一次唤醒到本次唤醒时间间隔);设置 3 秒唤醒,等待 3 秒休眠唤醒后,接着在唤醒后的 1 秒后 10 秒内继续设置 10 秒唤醒,则后面的唤醒时间为 10 秒(前一次唤醒到本次唤醒时间间隔)。

b、大值切换为小值：无论是在休眠唤醒后的 1 秒之内设置新的唤醒阈值,还是在休眠唤醒后的 1 秒之后,设置的时间值之内再设置新的唤醒阈值,唤醒时间都是该设置值。

例子:设置 10 秒唤醒,等待 10 秒唤醒后,无论是 1 秒内,还是 1 秒后 3 秒内设置 3 秒唤醒,唤醒时间均为 3 秒。

2.20. 程序无法执行到 main 函数？

如果代码中定义了过多的静态变量 (static) 或者常量 (const), 存储在 RAM 区, 可能会造成系统上电后, 对 RAM 初始化的时间过长 (系统上电后, 启动代码执行的时钟频率为 32.768KHz), 导致看门狗 (WDT) 复位, 程序不停地执行初始化 RAM 操作, 无法跑到 main 函数。

解决办法为：1、减少静态变量的使用；2、常量增加关键字 code, 定义在 flash 中 (**const unsigned char code** InitPara[]);

另外, 不同编译器对于 const 与 code 的规定不同。例如：在 Keil 51 中, const 修饰的变量为只读, 放在 RAM 中, code 修饰的变量放在 flash 中；在 keil MDK 中, const 修饰的变量放在 flash 中, code 修饰符无法使用。移植代码时, 需要特别注意。

2.21. 带电池 5V 供电系统应用, PDDet 配置注意事项？

在 5V 供电且带电池的应用中, 因为整个电源会在 3.6V 和 5V 之间切换。当进入休眠时, PDDet 设置为 1, 当来电时系统电源变为 5V, 此时 VCDIN 上电阈值会从 1.2V 升高到 1.4V 左右。如果 VCDIN 是按照 1.2V 设计, 就会存在无法休眠唤醒的风险。

解决办法有两种：

第一种：将上电时 VCDIN 的分压提高到 1.4V 以上。为了安全起见，VCDIN 的检测最好统一放在稳压芯片前端，低于 7V 休眠（变压器输出 12V 时，5V 系统）。对于已经设计好成对 5V 进行检测的方案，需要调整分压电阻的参数，但是要注意，调整参数后有可能造成掉电检测电压过低（3.9V）数据来不及保存的情况。

第二种：对于无法修改硬件的客户，修改软件设置：5V 应用时：POWSWT_ENN 置 1（CtrlM(0x2865)的 bit7），同时 PDDET 一直保持默认状态 0--开启，但是这样会带来 2uA 的功耗。

综上所述，以后在给客户推荐时：

- 1、 尽量采用 3.3V 系统
- 2、 VCDIN 的检测点放在稳压电源之前。

3. 计量

3.1. 如何计算功率/有效值寄存器理论值？

3.1.1. 电压/电流有效值

电压/电流有效值寄存器的理论数值与输入信号间的关系如下：

$$RMS = V \times G \times K$$

其中：

V：输入信号有效值；

G：当前增益；（模拟增益为主，数字增益一般设置默认 1 倍）

K：常数， $K = 1.8117 \times 10^9$ ；

例 1：电流通道 IA 的采样信号为 2.35mV，增益为 16，则 RMSI1（I1 电流有效值秒平均值，0x10D9）应为：

$$RMS = 0.00235 \times 16 \times 1.8117 \times 10^9 = 0x40F6D70$$

例 2：电压通道 U 的采样信号为 18mV，增益为 1，则 RMSU（电压有效值秒平均值，0x10D8）应为：

$$RMS = 0.018 \times 1 \times 1.8117 \times 10^9 = 0x1F19927$$

3.1.2. 有功/无功功率

有功/无功功率寄存器的理论数值与输入信号间的关系如下：

$$P = Vi \times Gi \times Vv \times Gv \times K \times C$$

其中：

V_i 、 V_v ：分别为电流和电压通道输入信号有效值；

G_i 、 G_v ：分别为电流和电压通道的增益；(模拟增益为主，数字增益一般设置默认 1 倍)

K ：常数，当计算有功功率秒平均值时， $K = 1.5413 \times 10^9$ ；当计算无功功率秒平均值时， $K = 2.4167 \times 10^9$ ；

C ：当公式用于计算有功功率时， $C = \cos \theta$ ；当公式用于计算无功功率时， $C = \sin \theta$ ；其中 θ 为电流与电压信号之间的相角；

例 1：当电压通道信号 V_v 为 18mV， G_v 为 1，IA 电流通道信号 V_i 为 2.3mV， G_i 为 16， $\cos \theta$ 为 1，则 DATAP (E1 功率秒平均值) 应为：

$$P = 0.0023 \times 16 \times 0.018 \times 1 \times 1.5413 \times 10^9 \times 1 = 0xF941D$$

3.2. 如何将功率/有效值寄存器数值转换成实际值？

寄存器数值为 32 位补码形式，需要通过乘上比例系数，得到实际的有效值，比例系数计算公式为：

$$D = \frac{V_n}{\text{Value}}$$

Value：根据理论公式计算得到的电压/电流有效值/功率值

V_n ：实际电压/电流有效值/功率值

D ：比例系数

例 1：设实际电压有效值为 220V 时，RMSU (电压有效值秒平均值，0x10D8) 的值为 0x1F19927 (32610599)，则：

$$D = \frac{220}{32610599} = 6.746 \times 10^{-6}$$

当 RMSU (电压有效值秒平均值，0x10D8) 读到的值为 0x3000000 (50331648)，则实际的电压有效值为

$$V_n = 50331648 \times D = 339$$

功率/电流有效值与上述类似。

3.3. 正常计量时电流有效值为降频计量 (800K) 时的 3 倍？

由于使能了电压/电流有效值计算通道上的带通滤波器 (BPFEN, Bit6, PMCtrl, 0x287A)，而带通滤波器系数寄存器 (0x10EF) 没有根据不同计量模式配置不同数值。正常计量模式时，用户应将该寄存器配置为默认值 0x889374BC，降频计量模式时，用户应将其配置为 0x911D3C9C。

3.4. 电压/电流有效值均为正值，有功功率却为负值？

电压/电流有效值为无符号数，不管接线正反，寄存器读取值均为正值；有功功率为有符号数，与电压/电流接线方式有关。出现上述问题，为硬件接线接反所致。

3.5. V9811A 样机上电时，发现反向灯点亮？

上电初始化完成后，需等待功率寄存器收敛稳定，才可读取功率寄存器的数值，通过符号位（功率寄存器的最高位）来判断功率方向。E1/E2 功率瞬时值（DATAIP/DATAIQ，0x10D1/0x10D2）收敛时间为 250ms，E1/E2 功率秒平均值（DATAP/DATAQ，0x10D6/0x10D7）收敛时间为 3s。

3.6. 无功计量时，无功功率反向问题？

无功计量时，如果电流信号为正值，则 E2 功率寄存器（DATAIQ，0x10D2/DATAQ，0x10D7）的值为负，功率将累加到反向 E2 能量累加寄存器；如果电流信号为负值，则 E2 功率寄存器的值为正，功率将累加到正向 E2 能量累加寄存器。

3.7. 无功计量时，Imax 点误差异常？

选择 E1 和 E2 计量通道功能时（DBLEN，Bit4，PMCtrl3，0x287A），当 DBLEN=0 时，标记 E2 的寄存器用于存储根据 I1 电流计算得到的无功数据，此时 E2 计量通道上的积分器会引入额外的 1.568 倍增益。无功计量时，计算信号大小需额外乘以 1.568，若得到的信号值大于 1.1V，导致 ADC 量程溢出，误差异常。

3.8. 频率影响量实验时，无功计量误差偏大？

若表计涉及到无功计量，而且需要做频率影响量实验时，可根据频率偏差，修改功率比差值的方式，调整无功计量误差。相关例程代码可向 FAE 索取。

3.9. 根据公式计算比差值出错？

有效值/功率比差校正寄存器的值计算公式如下：

$$S = 2^{31} \left(\frac{1}{1+e} - 1 \right) + S_1 \left(\frac{1}{1+e} \right)$$

经过查程序，发现程序中 2 的 31 次方表示为 2E31，而程序会认为是一个科学表示法，会认为是 2×10 的 31 次方，所以计算错误，改为常数 0x80000000，计算便正确。

3.10. V9821 样机自动校表时，随机出现角差校正异常？

经过程序调试发现，自动校表角差校正有一个 BUG，当计算结果为负值时，处理有问题。例如，计算角差校正值是 -3，按照角差校正寄存器的定义，负数只要最高位置 1 即可，写入角差校正寄存器的值应为 0x83，但是，程序计算的负数确是 FD，这样就会补偿过头，造成出现 4% 的误差。

3.11. V98XX 计量误差无法校正，程序无法烧录？

与外部晶振停振，自动切换到内部 RC 时钟有关，更换晶振后解决。建议客户增加晶体停振检测功能，给出相关报警提示，方便定位问题原因。

3.12. V9811 方案芯片升级为 V9811S 时，角差无法校正？

方案涉及到电流通道切换问题，V9811 的角差校正寄存器在切换通道之前，而 V9811S 的角差寄存器在切换通道之后，如下图所示：发生通道切换时，V9811 芯片，IA 信号进入 E2 计量通道，对应的校表寄存器为：E2 功率比差校正，E2 功率二次补偿，角差校正控制寄存器 1，容易搞混，而 V9811S 芯片，IA 信号进入 E2 计量通道，对应的校表寄存器为：E2 功率比差校正，E2 功率二次补偿，角差校正控制寄存器 2，比较合理。芯片升级时，软件需要做相应处理。

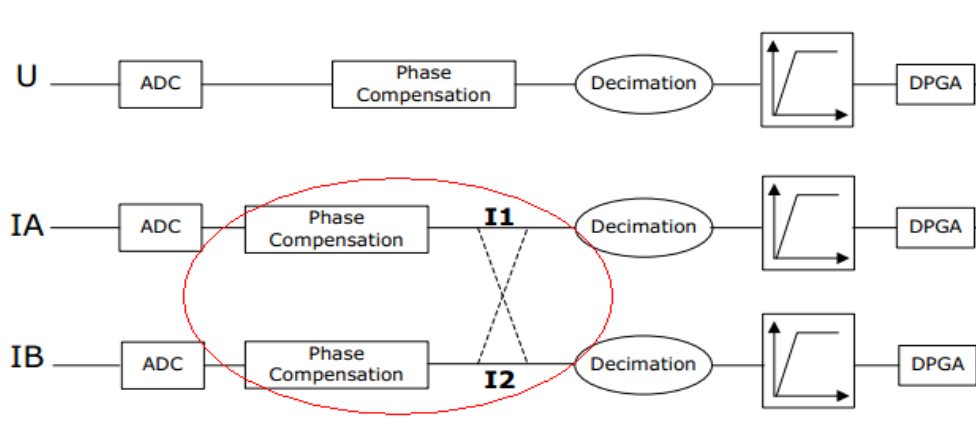


图 3.11-1 V9811 角差校正寄存器

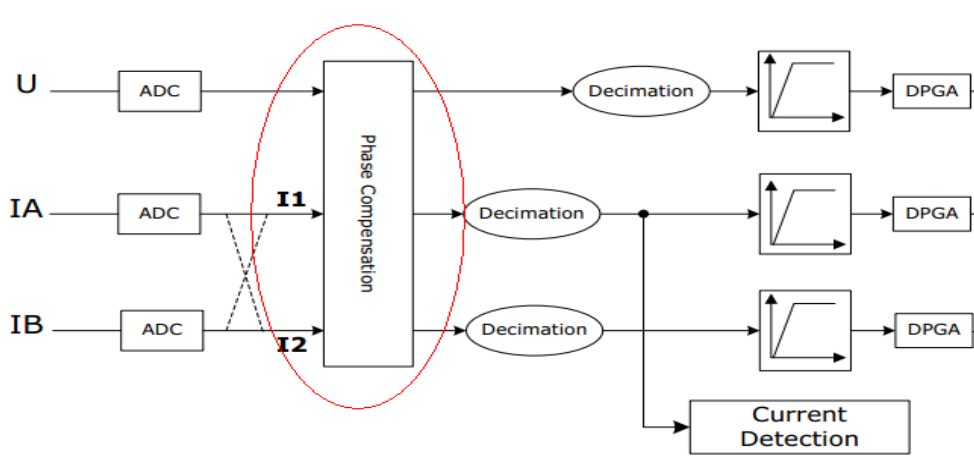


图 3.11-2 V9811S 角差校正寄存器

3.13. 常数计量应用，IA 通道功率正向时，计量正确；功率反向时，计量异常？

单相三线项目会采用 V98xx 常数计量模式（PSEL1/PSEL0，Bit[1:0]，PMCtrl4，0x287D）。常数计量模式下，常数功率值（DATA CP，0x10FC）仅累加到正向 E1 能量累加寄存器（PPCNT，0x10F0），此时用于输出 CF 的 E1 能量脉冲源（CFSEL1/CFSEL0，Bit[1:0]，CFCtrl，0x287E）必须选择正向 E1 能量。若 E1 能量脉冲源选择了正向+反向 E1 能量绝对值之和，计量因此出错。

3.14. 50Hz/60Hz 电网应用时的区别？

默认情况下，V98xx 支持 50Hz 电网应用。用户可通过配置 PLL 控制寄存器 (CtrlPLL, 0x2868) 的 Bit5，切换成 60Hz 电网应用。但是必须注意：选择 60Hz 时，MCU 时钟和电能计量时钟频率都相应的调整为 1.2 倍。因此，与时钟相关的设置，如：定时器、波特率、计量门限等，都需要重新配置。详见数据手册“50Hz/60Hz 电网应用对时钟频率的影响”这一章节。

3.15. 如何调整电表的高低温度误差？

V98xx 内置一个 BandGap 电路，通过调整 BandGap 温度系数 (RESTL<1:0>和 REST<2:0>, Bit[5:1], CtrlBGP, 0x2862)，可调整电表高低温度误差，具体步骤如下：

Step1: 先确定 BGP 控制寄存器的当前配置，假设细调寄存器 REST<2:0>='010'，粗调 RESTL<1:0>='00'，那么查表知道目前对应的温度系数调节幅度为+20ppm。

Step2: 实验确定电表的高低温度误差。例如 20 摄氏度误差为 0，高温 80 度为 0.6%，低温 -40 度为 -0.4%。那么需要调节的误差幅度是 $(0.6\% - (-0.4\%))/2$ ，为 0.5%，调节方向以高温为基准，应该将高温误差往负调。对应 ppm 是 $-0.5\%/(80-20) = -5000/60 = -83\text{ppm}$ ，四舍五入取 -80ppm。

Step3: 误差和 REF 的变化幅度是负两倍关系，因此对应要将 REF 往正方向调节 40ppm。考虑到现有配置已经是 +20ppm，那么最终要调至 +60ppm。查表知道当 RESTL<1:0>='11' 时，对应 +70ppm，REST<2:0>='111' 时，对应 -10ppm，组合起来等于 60ppm。

3.16. 调整 BGP 控制寄存器后，影响温度测量？

BGP 控制寄存器 (CtrlBGP, 0x2862) 推荐配置为 0x02，若为了调整电表高低温度误差，修改了 BGP 控制寄存器的值，M 通道测出的温度值会有一些影响。对于 RTC 精度要求较高的项目，需要根据不同的 BGP 控制寄存器配置，对 M 通道测出的温度值做补偿。补偿方法为测得温度值乘以系数 K，系数 K 与 BGP 控制寄存器的对应关系参考万高 RTC 校正例程或者向万高 FAE 索取。

4. 仿真

4.1. 在线仿真时，系统状态寄存器 PWRDN 与 PWRUP 矛盾？

仿真器与芯片相连时，MODE 引脚会被拉低。PWRUP 在 MODE 引脚拉低的情况下会一直置 1 与 VDCIN 引脚电平无关，PWRDN 可以正常使用。

4.2. 仿真器供电时程序正常执行，220V 供电时，系统没有反应？

仿真器 JTAG 与芯片相连时，MODE 引脚拉低，芯片处于调试状态。此时 P0 端口复用为 JTAG 口，屏蔽 WDT 溢出复位（即看门狗失效），220V 供电时，由于喂狗不及时导致系统工作异常。

4.3. 仿真器能否调试休眠模式下代码运行？

系统进入休眠状态后，时钟 1 关闭，CPU 及其外设（如 GPIO 等）停止工作，因此，JTAG 端口既不能用于在线仿真调试，也不能用于程序下载。

4.4. 仿真器 JTAG 第 10 脚的用法？

如果项目为复费率表型，带 RTC，建议 JTAG 第 10 脚与芯片 P9.3/P1.3/P1.4 任一引脚相连。SD502/SP402 开启 RTC 校正功能后，会在脱机下载 HEX 之前，将 P9.3/P1.3/P1.4 配置成 1Hz 信号输出，仿真器会将芯片输出的 1Hz 信号与标准秒脉冲信号比较，从而得到 RTC 常温偏移值，并将其写入地址为 0x48C 的 FLASH 储存区中。

4.5. SP402 开启 RTC 校正功能时，脱机下载失败？

SP402 开启 RTC 校正功能时，会先进行温度校正和 RTC 常温补偿，温度校正时环境温度要求在 15°C~35°C 之间，超过此温度烧写会报错。

4.6. SD502 的 USB 驱动安装失败 (unknow device) ？

与 USB2 未停用有关，将 Intel(R) 82801G (ICH7 Family) USB2 Enhanced Host Controller 设置停用。

4.7. 报错：Fail to open #include file 'iar_check_symbols.h' ？

iar_check_symbols.h 这个头文件在 IAR 安装目录下，并非在所建工程目录下，因此需要重新定义路径。改为 #include "..\src\lib\iar_check_symbols.h" 即可。头文件 "iar_cfi.h" 与之类似。

4.8. 报错：Page2 must be empty?

FLASH 从 0x400 地址开始的 512 字节需空出来，SD502 会将 RTC 校正相关的数据拷贝到 0x400 的 FLASH 地址中。KEIL 开发环境下，已在启动文件 STARTUP.A51 做了配置，具体为 CSEG AT 0400H 等几句代码；IAR 开发环境下，需要在 main.c 函数中添加相关代码，具体如下：

```
__root const unsigned char __code codepsword[1024]@0x400=
{
    0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,
    0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,
    ...}
```

4.9. 工程建完后，编译通过，代码执行异常？

需要添加万高提供的相关配置文件，包括启动文件 cstartup.s51；iar_banked_code_support.s51（代码超过 64K，需要分 BANK 时添加）；IAR8.3 版本还需添加 workaround.s51。

4.10. KEIL 代码移植到 IAR 时数据处理相关问题？

KEIL 编译器为大端模式存储，即：高位存在低地址，低位存在高地址；IAR 编译器为小端模式存储，即：高位存在高地址，低位存在低地址；例如将四个字节变量 0x1234abcd 写入到以 0x0000 开始的内存中，对应的存储方式如下：

内存地址	大端模式	小端模式
0x0000	0x12	0xcd
0x0001	0x34	0xab
0x0002	0xab	0x34
0x0003	0xcd	0x12

5. 附录

5.1. SetPLL 标准流程

```
#define SETPLL_800K 0
#define SETPLL_1_6M 1
#define SETPLL_3_2M 2
#define SETPLL_6_4M 3
#define SETPLL_13M 4
#define SETPLL_26M 5
unsigned char SetPLL(unsigned char PLLClk)
{
    unsigned char i;
    i=0;
    FSC=0;
    FWC=0;
    MCUFRQ = 0;
    while((MCUFRQ))
    {
        i++;
        if(i>20)
        {
            return 0; //在一定时间内没有锁定
        }
    }
    MEAFRQ = 0;
    while((MEAFRQ))
    {
        i++;
        if(i>20)
```

```

    {
        return 0; //在一定时间内没有锁定
    }
}
for(i=0;i<5;i++);
CtrlCLK = 0xc0;
CtrlPLL = 0;          //800K pll 800k, DSP 800K,adc 200k,并切换到 800kPLL
while(!(PLLLCK&0x01))
{
    i++;
    if(i>50)
    {
        return 0; //在一定时间内没有锁定
    }
}
MCUFRQ=1;
i=0;
while(!(MCUFRQ))
{
    i++;
    if(i>20)
    {
        return 0; //在一定时间内没有锁定
    }
}
MEAFRQ=1;
i=0;
while(!(MEAFRQ))
{
    i++;
    if(i>20)
    {
        return 0; //在一定时间内没有锁定
    }
}
//for(i=0;i<5;i++);
PMG=0;      //开启计量时钟
switch(PLLCIk)
{
    case 0:      //pll 800k, DSP 800K,adc 200k,并切换到 800kPLL
    {
        CtrlCLK = 0xc0;//
        CtrlPLL = 0x00;//
    }
    break;
}

```

```

case 1:      //pll 1.6M, DSP 1.6M,adc 400k,并切换到 1.6MPLL
{
    CtrlCLK = 0xd5;//
    CtrlPLL = 0x00;//
}
break;
case 2:      //pll 3.2M, DSP 3.2M,adc 800k,并切换到 3.2MPLL
{
    CtrlCLK = 0xea;//
    CtrlPLL = 0x00;//
}
break;
case 3:      //pll 6.4M, DSP 3.2M,adc 800K,并切换到 6.4MPLL
{
    CtrlCLK = 0xeb;//
    CtrlPLL = 0x00;//
}
break;
case 4:      //pll 13M , DSP 3.2M,adc 800K,并切换到 13MPLL
{
    CtrlCLK = 0xeb;//
    CtrlPLL = 0x40;//
}
break;
case 5:      //pll 26M , DSP 3.2M,adc 800K,并切换到 26MPLL
{
    CtrlCLK = 0xeb;//
    CtrlPLL = 0xc0;//
}
break;
default:    //pll 3.2M, DSP 3.2M,adc 800k,并切换到 3.2MPLL
{
    CtrlCLK = 0xea;//
    CtrlPLL = 0x00;
}//
break;
}

//CtrlCLK = PLLCLK;//ADC 800K dsp 3.2M MCU 13M;
//CtrlPLL = CTRLPLL;// 13M
while(!(PLLLCK&0x01))
{
    i++;
    if(i>50)
    {
        return 0; //在一定时间内没有锁定
    }
}

```



```

    }
}
// for(i=0;i<5;i++);
MCUFRQ=1;
i=0;
while(!(MCUFRQ))
{
    i++;
    if(i>20)
    {
        return 0; //在一定时间内没有锁定
    }
}
MEAFRQ=1;
i=0;
while(!(MEAFRQ))
{
    i++;
    if(i>20)
    {
        return 0; //在一定时间内没有锁定
    }
}
//for(i=0;i<5;i++);
PMG=0;      //开启计量时钟
CtrlCry0=0;//调整启振波形
CtrlCry1=3;
CtrlCry2|=0x20; //使能起振电路停振复位
CtrlBGP=0x02;// 改善高低温误差+10PPM
CtrlIM = 0x01;      //禁止 M 通道去直
//CtrlADC1 = 0X30;
return 1;
}

```