

炬泉 HT5X2X FAQ

V1.3

炬泉光电科技（上海）股份有限公司

Tel: 021-51035886

Fax: 021-50277833

Email: sales@hitrendtech.com

Web: <http://www.hitrendtech.com>

修改记录

版本号	修改时间	创建\修改	修改内容
V1.0	2016-08-23	郭宇清	初版创建
V1.1	2017-04-22	徐晨曦	PF 口设计区别 (2.12) 增加关于外部中断口的应用 (2.29)
V1.2	2018-01-08	chyang	1. 删除 2.22 仿真接口 TDO 的配置相关说明; 2. 修改 3.4 VDD1P5 管脚外接电阻的上限为 100 欧姆; 3. 新增 2.28 低温-40 度, 系统时钟切换导致进入 hardfault 的规避方式;
V1.3	2018-01-30	chyang	1. 修改 2.28 的说明; 2. 删除 3.8 ECC 应用。

注: 本文档适用于 HT5023/HT5025/HT5027

目 录

1	HT5X2X VS HT5X1X.....	4
2	软件.....	4
2.1	RTC 定时器定时.....	4
2.1	RTC 补偿参数.....	4
2.2	PE7/LVDIN0, PE9/LVDIN1 管脚使用.....	7
2.3	PLL 锁定判断用户应用.....	7
2.4	SLEEP/HOLD 模式.....	8
2.5	OSC 起振时间.....	8
2.6	CLKCTRL0 配置.....	8
2.7	RTC 时间修改.....	8
2.8	RTC 定时器定时.....	8
2.9	EMU 初始化.....	8
2.10	无功相位校正寄存器的配置.....	8
2.11	脉冲口区别.....	9
2.12	Uart 使用.....	9
2.13	TPS 低功耗模式处理.....	9
2.14	startup.s 文件中 Flash 地址 FC0 处推荐配置.....	9
2.15	ADC 检测功能.....	9
2.16	在 HardFault 中不能加入软复位或者其他的强制复位的语句.....	9
2.17	建议不使用 PLL 停振检测功能.....	10
2.18	EMU 常数计量能否使用清零型能量累加寄存器.....	10
2.19	使用 INT 中断和 RX 中断的注意事项.....	10
2.20	系统时钟配置.....	10
2.21	OSC 停振之后 NMI 中断处理推荐.....	10
2.22	软复位说明.....	11
2.23	ADCIN2 复位问题.....	11
2.24	UART\7816 SREL 不可设置为 0.....	11
2.25	扫描键盘应用.....	11
2.26	ADC 同步缓存 buffer 应用.....	11
2.27	关于外部中断口的应用.....	12
2.28	低温-40 度下, 系统时钟由低频切换到高频 19.66M / 39.32M, 出现 hardfault 的规避方法.....	12
3	硬件.....	13
3.1	外部 32K 晶振应用说明.....	13
3.2	VDD 管脚.....	13
3.3	VBAT 管脚.....	13
3.4	VDD1P5 管脚.....	14
3.5	TEST 管脚.....	14
3.6	电源框图 VDD5V 说明.....	14
3.7	LCD 大小电流的电阻分压结构控制位 FCSET2 详解.....	14
4	可靠性问题.....	15
4.1	仿真接口设计.....	15
4.2	仿真接口增加 RST 建议.....	16

1 HT5X2X VS HT5X1X

详见: HT502X VS HT501X V1.1.pdf

2 软件

2.1 RTC 定时器定时

RTC 定时器使用时, 不推荐频繁打开关闭 RTC 定时器。

RTC 内部定时器的计数器是在每次 RTC 秒进位的时间才会清零, 因此用户在使用 RTC 定时器时如果打开定时器功能, 定时一段时间关闭定时器然后再次打开时, 需要注意计数器的清零是在秒进位时发生。因此如果经常不断的开关定时器时, 可能会导致定时不准确。

2.1 RTC 补偿参数

为了保证芯片在运行过程中能够根据环境温度进行稳定补偿, 需要减小 TPS 温度数值的跳动, 建议 RTC 补偿代码配置如下:

```
void RTC_compensate_initial(void)
{
uint8_t i;
uint32_t ichecksum;

//TPS config-----
HT_TBS->TBSCON=0x0301; //Chop 都打开 8次平均输出 osr=64
HT_TBS->TBSIE=0x00;
HT_TBS->TBSPRD=0x00; // 1s 打开 8次

//RTC configure-----
HT_RTC->RTCIE=0x00;
HT_RTC->RTCCON=0x00;
HT_RTC->RTCCON|=0x06; //高频补偿 128Hz 之后输出 1Hz
HT_RTC->DFIH=((uint32_t)0>>16);
HT_RTC->DFIL=(0);

/*****读取 Info 保存的 RTC 参数, 并计算校验和, 检查校验和是否正确 *****/
/*****如果校验和正确, 加载 Info 参数到 RTC 相应 registers, 否则写入默认值参数*****/
}
```

芯片的从 Flash 自动装载功能在不同的上电速度条件下可能失效, 凡是会被自动装载的寄存器, 需要软件人员手动在软件初始化的过程中重新写入一次。

所以建议用户代码里统一加上以下部分:

```
typedef union
```

```
{
```

```
struct
```

```
{
    _I uint32_t iDFAH; //偏移地址 0x104
    _I uint32_t iDFAL; //偏移地址 0x108
    _I uint32_t iDFBH; //偏移地址 0x10C
    _I uint32_t iDFBL; //偏移地址 0x110
    _I uint32_t iDFCH; //偏移地址 0x114
    _I uint32_t iDFCL; //偏移地址 0x118
    _I uint32_t iDFDH; //偏移地址 0x11C
    _I uint32_t iDFDL; //偏移地址 0x120
    _I uint32_t iDFEH; //偏移地址 0x124
    _I uint32_t iDFEL; //偏移地址 0x128
    _I uint32_t iTOff; //偏移地址 0x12C
    _I uint32_t iMCON01; //偏移地址 0x130
    _I uint32_t iMCON23; //偏移地址 0x134
    _I uint32_t iMCON45; //偏移地址 0x138
    _I uint32_t iChecksum; //偏移地址 0x13C

    // iChecksum = iDFAH + iDFAL + iDFBH + ... +iMCON01 + iMCON23 + iMCON45

}Muster;

_I uint32_t DataArry[15];
}HT_Info_Typedef;

#define HT_InfoData_Base (0x00040000+0x104)

#define HT_Info ((HT_Info_Typedef *) HT_InfoData_Base )

signed long Ax=0;
signed long Bx=-0;
signed long Cx=-0;
signed long Dx=0;
signed long Ex=0;
uint32_t i,checksum;
```

```
for(i=0;checksum=0;i<14;i++)
{
    checksum +=HT_Info->DataArray[i];
}

if(checksum!= HT_Info->DataArray[14])
{
    //用户报警代码，RTC 补偿系数可能存在问题，填写默认值：

        HT_RTC->MCON01=0;
        HT_RTC->MCON23=0;
        HT_RTC->MCON45=0x8600;

        HT_RTC->DFAH=(Ax>>16)&0xFFFF;
        HT_RTC->DFAL=(Ax>>0)&0xFFFF;

        HT_RTC->DFBH=(Bx>>16)&0xFFFF;
        HT_RTC->DFBL=(Bx>>0)&0xFFFF;

        HT_RTC->DFCH=(Cx>>16)&0xFFFF;
        HT_RTC->DFCL=(Cx>>0)&0xFFFF;

        HT_RTC->DFDH=(Dx>>16)&0xFFFF;
        HT_RTC->DFDL=(Dx>>0)&0xFFFF;

        HT_RTC->DFEH=(Ex>>16)&0xFFFF;
        HT_RTC->DFEL=(Ex>>0)&0xFFFF;
}
else
{
    //RTC 补偿系数正确
```

```
HT_RTC->DFAH = HT_INFO->Muster.iDFAH;
HT_RTC->DFAL = HT_INFO->Muster.iDFAL;
HT_RTC->DFBH = HT_INFO->Muster.iDFBH;
HT_RTC->DFBL = HT_INFO->Muster.iDFBL;
HT_RTC->DFCH = HT_INFO->Muster.iDFCH;
HT_RTC->DFCL = HT_INFO->Muster.iDFCL;
HT_RTC->DFDH = HT_INFO->Muster.iDFDH;
HT_RTC->DFDL = HT_INFO->Muster.iDFDL;
HT_RTC->DFEH = HT_INFO->Muster.iDFEH;
HT_RTC->DFEL = HT_INFO->Muster.iDFEL;
```

```
HT_RTC->Toff = HT_INFO->Muster.iToff;
```

```
HT_RTC->MCON01 = HT_INFO->Muster.iMCON01;
```

```
HT_RTC->MCON23 = HT_INFO->Muster.iMCON23;
```

```
HT_RTC->MCON45 = HT_INFO->Muster.iMCON45;
```

//用户应继续判断寄存器中系数与 Info 中系数是否一致（防止自动装载失败），如果不一致则用户需再将 Info 中数据写一遍到对应寄存器

```
}
```

2.2 PE7/LVDIN0, PE9/LVDIN1 管脚使用

这两个 pin 脚靠近芯片 OSC 管脚，不建议用作 I/O 口模拟时钟高速翻转 GPIO 用。推荐用作 LVDIN 功能。

2.3 PLL 锁定判断用户应用

软件延时方式

normal 情况下，Pllen 打开后延时大于 4ms 即可

寄存器状态位等待方式

先将 PLL_LOCK_EN (CLKCTRL0 的 bit6) 置 0，再将 PLL_EN (CLKCTRL0 的 bit4) 置 1，然后判断 PLL_LOCK (CLKSTA 的 bit5) 状态是否为 1，若为 1 则判断为锁定（锁定后可将 PLL_LOCK_EN 置 1，将 PLL_LOCK 状态强制锁定为 1）。

2.4 SLEEP/HOLD 模式

1. 推荐客户在 VBAT 供电情况下进入 sleep/hold 模式，在 VSYS 供电下进入 sleep/hold 模式功耗会有抖动。
2. 用户在写软件时，切忌上电之后立马进入 sleep 模式，以免仿真器无法再次连接芯片并下载程序。

2.5 OSC 起振时间

外部晶振冷启动上电后 OSC 的稳定时间为 500ms，因为系统上电后默认跑 HRC 时钟，如果用户需要切换至 OSC 或 PLL 时钟时，需要延时等待 500ms 后，开启 LF 低频停振检测，若没有停振标志，再切换时钟。芯片在发生 POR 或 LBOR 复位时，由于是从 HRC 开始启动，如果软件人员需要切换到 OSC 时钟或者 PLL 时钟运行，需要等待 500ms 的时间，主要是由于 OSC 晶振起振到正常的频率值需要 500ms 的时间。

POR 复位：VCC \leq 0.3V 持续时间大于 200us 再恢复到 0.3V 以上时，系统触发 POR 复位；

LBOR 复位：VCC 在(0.3V, 1.9]范围内持续时间大于 200us 再恢复到 1.9V 以上时，系统触发 LBOR 复位。

2.6 CLKCTRL0 配置

CLKCTRL0 的 OSC_SLP 不要配置成大功耗模式，小功耗模式可以保证有效起振；CLKCTRL0 的 VDD1P5_LBOR 为内部电源控制位，保持默认配置为 1 即可，**建议不要修改**。

2.7 RTC 时间修改

建议客户修改 RTC 时间按照年、月、日、时、分、秒的顺序去修改寄存器的值。如果 2 月 1 日的时候，用户按照日月修改为 1 月 31 日时，会先判断 31 日对于 2 月是非法数据，就会导致修改时间不成功。

2.8 RTC 定时器定时

RTC 定时器使用时，如果关闭 RTC 定时器 (RTCCON.Bit6 (定时器 2) 或 RTCCON.Bit5 (定时器 1) 清零)，RTC 定时器 (1 或 2) 内部计数值不会立即清零，会等到发生秒进位的时刻才被清零。如果用户代码运行过程中会**改变定时周期**，应该按照如下流程进行：

1. 关闭 RTC 定时器(1/2, RTCCON.Bit6 和 RTCCON.Bit5 写 0)
2. 等待秒中断产生 (RTCIF.Bit0 置位) **注：如果没有这一步，RTC 内部定时器计数值不会从 0 开始计数，而是从上次计数值开始继续计数**
3. 设定 RTC 定时器新的周期 (RTCTMR1/2)
4. 使能 RTC 定时器 (RTCCON.Bit6 和 RTCCON.Bit5 写 1)

该问题 E 版之后已经修改过了

2.9 EMU 初始化

EMU 初始化需要把校表寄存器 98H 的 bit15=1 (打开 EMU 模拟的 LDO)，50H 的 bit13=1 (打开 VREF)，这两位配好后脉冲口才能出脉冲。

2.10 无功相位校正寄存器的配置

无功相位校正寄存器(2C) 在 819.2kHz/OSR=64 模式下推荐值为 0x0050。

无功相位校正寄存器(2C) 在 819.2kHz/OSR=128 模式下推荐值为 0x0028。

2.11 脉冲口区别

HT502X 与 HT501X 的 PF 口默认值不一样，HT501X 默认为 GPIO 口，输入上拉；HT502X 默认为 PF 口，高电平有效（上电默认是低电平），用户在硬件设计时注意区别。

2.12 Uart 使用

Uart 进行初始化时，必需先打开模块使能（CLKCTRL1 相应位），再进行串口寄存器的功能配置。否则，再没有使能模块的情况下，写相应寄存器无效。

客户使用 URAT 时，TXEN 必须一直打开。

当串口正在发送过程中，如果关闭 TXEN（UARTCON.Bit0）或者 CLKCTRL0 寄存器中对应的 UART 时钟使能位，则 TX 管脚将保持当前发送时的状态。

这种情况下，如果想 TX 管脚恢复到默认初始状态（正逻辑下为“高”），则可以采取以下两种措施：

- A) 发生复位（比如软复位/WDT 复位等）
 - B) 使能 TXEN 和 CLKCTRL0 寄存器中 UART 时钟使能位，让未发送完的数据发送完成即可
- 注：串口正在发送过程中表示：已经发送了起始位，但是还没有发完停止位

2.13 TPS 低功耗模式处理

为保证 RTC 正确补偿，建议始终保持 TPS 开启状态，在进入低功耗之前，为降低功耗，可配置 TPSPRD 分时开启 TPS 以降低功耗。可根据实际需求配置分时开启的时间间隔，若 1s 开启一次，平均功耗 < 0.7uA。推荐客户采用 1s 打开一次。

2.14 startup.s 文件中 Flash 地址 FC0 处推荐配置

```

CRP_Key          IF      :LNOT::DEF:NO_CRP
                  AREA    |.ARM.__at_0x0FC0|, CODE, READONLY
                  DCD     0xFFFFFFFF
                  ENDIF
    
```

关于.s 文件中 CRP_Key 配置推荐：

HT502X	推荐配置	0xXXXXFFX2	Flash 不加密，LRC 使能打开
注 1： X 表未此控制位不起任何作用，推荐用户统一用“F”填充“X” 注 2： 红色 FF 为 Flash 加密位，如果需要加密程序（无法从 SD 口读出），需将红色 FF 变成非 FF 即可。比如：FFFF00F2			

2.15 ADC 检测功能

先配 GPIO 为 ADC，再打开 ADC 使能。否则有可能导致第一次测试数据出错。

ADCINx 的测量输入范围为 800mVp，输入极限电压为 1.5V，否则内部 ADC 模块可能会被烧坏。

2.16 在 HardFault 中不能加入软复位或者其他的强制复位的语句

建议客户不写 hardfault 函数，如果写的话就再里面加一个死循环等待看门狗复位。

如果客户在 hardfault 中断函数内加入软复位或者其他的强制复位语句，用户代码进入了 hardfault 中断函数，

然后执行强制复位语句导致芯片复位，再次运行依然再进 **hardfault**，这样会进入一个死循环，导致芯片一直处于复位状态，不能恢复，仿真接口也无法连接芯片以及擦除下载程序。

2.17 建议不使用 PLL 停振检测功能

PLL 停振检测功能是由 LRC 去检测，LRC 有毛刺的情况，存在发生误判的机率，所以建议客户不要使用 PLL 停振检测功能。

OSC 停振检测是 LRC 经过了 128 分频后的时钟来检测，所以 LRC128 分频后毛刺会被滤掉，基本上 OSC 停振检测时非常可靠的，推荐使用 OSC 停振检测功能。

另外，当 OSC 冷启动时，需要大约 300ms 的稳定时间，在这段时间内，也容易导致误判 LF 停振，若 LF 停振检测打开，系统进入 NMI 中断（NMI 中断中不作处理等待 WDT 复位，默认 4s），造成系统启动慢的现象。

鉴于以上原因，建议在系统复位之后，首先关闭所有停振检测，即配置 CLKCTRL0 的 HRC_DET_EN=0，PLL_DET_EN=0，LF_DET_EN=0，再进行其他模块初始化；当系统启动时间超过 300ms 之后，再将 LF 停振检测打开，即配置 LF_DET_EN=1，以确保在 OSC 发生停振时，系统能够有效复位并切换到其它时钟源，但 PLL 停振检测建议始终关闭。

2.18 EMU 常数计量能否使用清零型能量累加寄存器

可以。常数计量模式下进入低功耗之后，能量的累加速度为 32KHz，而正常计量模式累加速度为 BS 位流。所以进入常数计量模式，只需将 CHNLCR 的 bit7 置 1，使 EMU 的 BS 时钟选择为 32K（LPMODE=1）即可读取清零型能量寄存器 ENERGYPC。

2.19 使用 INT 中断和 RX 中断的注意事项

使用 INT 中断引脚功能的时候，必须要将 PINFLT、PINFLT2 寄存器中相应的 INTx 引脚数字滤波功能打开。

使用 RX 中断引脚功能的时候，必须将 PINFLT 寄存器中对应 RXx 的引脚数字滤波功能打开。

F 版已经修改为：内部强制使能 RX 和 EX_INT 的外部引脚数字滤波功能，写 PINFLT 无效。

这个外部引脚数字功能仅对配置的 RX 或 INT 有效，对相应 PIN 脚的其它功能（GPIO 或其它复用功能）无效。

2.20 系统时钟配置

PLL 最高时钟频率为 39.32MHz，用户使用时不推荐使用最高频率时钟，有可能会造成 EMC 性能不太好，建议使用 PLL 倍频到 19.66M 时钟，Fcpu 可以进行二分频，或者更低频率。

2.21 OSC 停振之后 NMI 中断处理推荐

如果用户代码检测到 OSC 停振而进入到 NMI 中断，推荐以下两种处理方式：

- A) 产生复位（WDT 复位或者内核软复位）

产生复位主要是基于以下理由：

NMI 一般很少产生，如果产生，有可能是芯片工作不正常，通过复位使芯片重新工作。（在目前的代码中，用户如果没有重写 NMI 中断函数而发生 NMI 中断，那么最终会导致 WDT 复位）

如果采用此种方案，那么芯片会有复位发生，导致运行不连续，从而产生一些比如丢失电量等问题

- B) 如果用户不想等待复位，查询停振检测标志恢复正常然后切换到 PLL 或者 OSC 时钟，然后退出 NMI 中断程序，使程序接着运行。

如果采用此种方案，那么 NMI 中断只像是普通的应用中断，对程序运行影响不会很大。

2.22 软复位说明

内核软复位 `System_reset`，不能复位 `HRC_EN` 和 `sysclk_sel`，即内核软复位系统时钟不会强制切换到 HRC，保留复位前的系统时钟配置；**与其他复位不一致**；

规避注意：若用户使用系统软复位，请在程序开始先配置 `HRC_EN` 使能，切换系统时钟为 HRC，再进行初始化。

2.23 ADCIN2 复位问题

TBS 模块 `TBSPRD` 寄存器的[bit11:bit10] (`ADC2PRD[1:0]`) **无复位值**，故复位后为随机值；设计已确认

规避注意：使用 ADCIN2 通道前，必须配置 `TBSPRD` 寄存器

2.24 UART\7816 SREL 不可设置为 0

UART\7816 模块，波特率设置寄存器 `SREL` 不建议设置为 0，当此寄存器设置为 0 时（即系统时钟的 2 分频），发送无输出信号；非 0 时正常。需要在应用中注意。

2.25 扫描键盘应用

按键扫描模块涉及的芯片引脚为固定的 4 个 `SCANIN[0...3]`，分别为 `PC10`, `PC9`, `PA5`, `PA6` 的复用功能，用户如果需要使用按键扫描功能，**可任意配置这 4 个引脚中的某些管脚为第二功能**。

这点设计上的改进，不同于 HT5X1X，但全兼容 HT5X1X 的应用设计。应用时请注意。

2.26 ADC 同步缓存 buffer 应用

波形数据缓存到 Buffer 需要时间，因此要求用户在启动了波形缓存功能后，必须等到 Buffer 存满之后才能进行读出操作（存满有中断标志，若使能后可进入 `EMU_IRQHandler`）。已验证如果在存入 Buffer 的过程中进行读操作，读出的数据会出错（表现为第 1 个点错）

2.27 关于外部中断口的应用

外部中断复用端口，配置为非中断功能时，内部切断与端口连接并锁定外部中断输入固定为 0（防止逻辑浮空），建议用户将 GPIO 口复用为外部中断口之后，如果需要再配回 GPIO 口的话，无论是否打开外部中断滤波功能都建议先关闭外部中断使能然后再配置为 GPIO 口，以免发生误触发。

2.28 低温-40 度下，系统时钟由低频切换到高频 19.66M / 39.32M，出现 hardfault 的规避方法

问题：低温-40 度下，系统时钟由 LRC 或 OSC 切换到高频 19.66M/39.32M 的情况下，芯片会进入到 hardfault。

问题原因：系统时钟由 LRC 或 OSC 切换到高频 19.66M/39.32M 下，功耗突然增加导致 VDD1P5 电压有毛刺，从而进入到 hardfault 状态。

规避方法：

- （1）硬件上，为兼顾改善 ESD 性能，VDD1P5 外部串接电阻的上限为 100 欧姆，详细可见 3.4 的说明；
- （2）软件上，建议系统时钟切换顺序，先由 LRC 或 OSC 切换到高频 9.83M，再切换到 19.66M/39.32M。

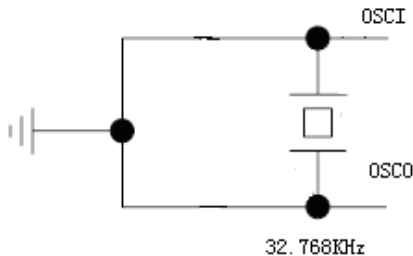
注：系统时钟选择 39.32M，且系统时钟选择不分频（`SYSCLK_DIV[2:0] = 000B`），必须提前使能指令延迟取指功能（`FLASHDLY.FLASH_DLY = 1`）。

建议用户以上两条规避方法同时用。

3 硬件

3.1 外部 32K 晶振应用说明

芯片已经内置电容，无需外接电容（内部集成了两个 24pf 电容）。如果外接电容，可能会影响芯片的 EMC 性能。



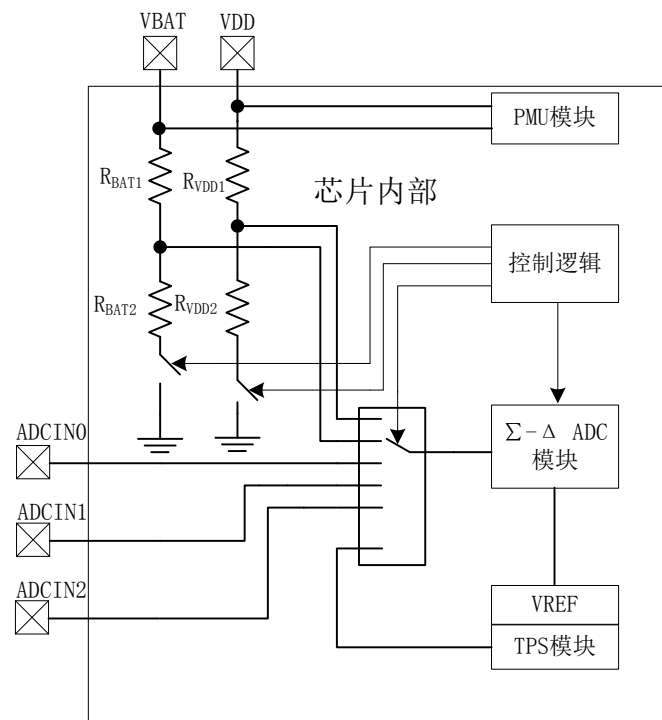
3.2 VDD 管脚

芯片两个 VDD 管脚必须外部连接在一起，否则会造成部分 GPIO 工作不正常。每个 VDD 外部挂一个 10uf 和一个 0.1uf 电容。

3.3 VBAT 管脚

VBAT 管脚输入电压不能超过 3.8V。

- TBS 模块框图：5路 ADC 电压（VDD, VBAT, ADCIN0, ADCIN1, ADCIN2）



VBAT 检测内阻为 30K (R_{BAT1} 为 24K, R_{BAT2} 为 6K), VDD 检测内阻为 42K (R_{VDD1} 为 36K, R_{VDD2} 为 6K)。ADCIN0, ADCIN1, ADCIN2 内部均无内阻。

3.4 VDD1P5 管脚

VDD1P5 管脚一般设计只需要外挂一个 0.1uF 的电容，用户可以根据 ESD 需求在 VDD1P5 和电容之间串接 0-100 欧姆电阻，串接电阻越大，可能对 ESD 改善效果越明显，但是不能超过 100 欧姆。

3.5 TEST 管脚

TEST 管脚接一个 1K 电阻上拉到 VDD 和一个 0.01uf 电容到地。如果不接上拉电阻，会导致快速上电的话，电容充电导致 IO 口上电时间延迟几个 ms。

3.6 电源框图 VDD5V 说明

-VDD5v 是 VSYS 或 VBAT 选择结果，是和外部 VDD 一致，注意区分 VDD 和 VDD1P5；

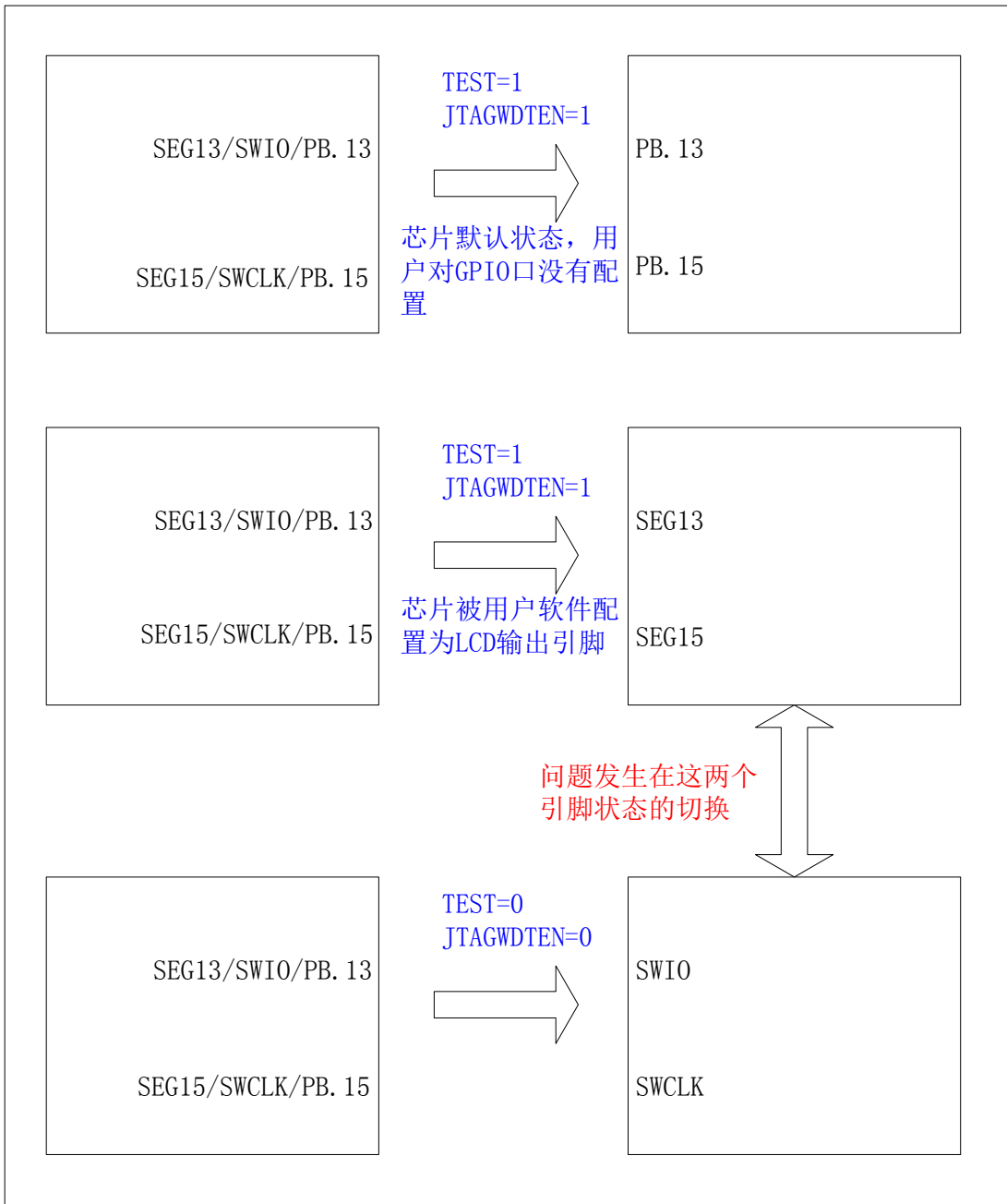
-系统切换为外部 VBAT 供电时，VDD5v 为 VBAT 电压减去 switch 导通阻抗的压降，略低于 VBAT 电压；

3.7 LCD 大小电流的电阻分压结构控制位 FCSET2 详解

FCSET2	0 (兼容 HT5X1X)	1 (大电流更大, 小电流更小)
大电流模式	68.5k/3=22.8k	34.7K/3=11.6K
小电流模式	416k/3=138.7k	2.08M/3=693K

4 可靠性问题

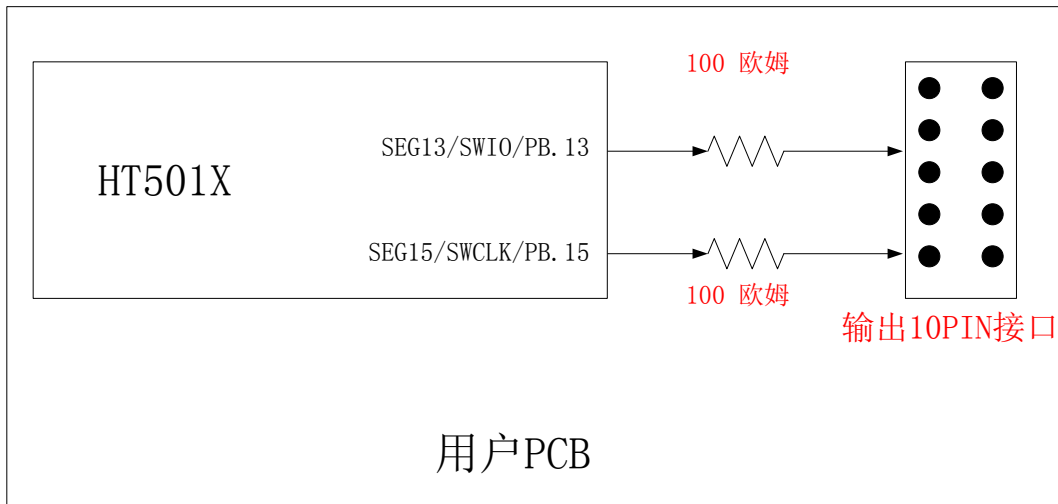
4.1 仿真接口设计



如果在 TEST, JTAGWDTEN 引脚拉高的时候，芯片的调试引脚被用户软件配置为 LCD 输出引脚，Ulink2 的仿真输出口与芯片的 LCD 引脚会直接接触，接口拔插的过程中如果产生静电或者则有导致 LCD 模块内部损坏的可能。规避方法如下：

将 ULINK2 的输出 10PIN 引脚，如果是通过长针与客户 PCB 相连，最好能将与客户 PCB 的 TEST 和 JTAGWDTEN 引脚相连的两个 GND 引脚变长，两个接头相连的 GND 引脚也变长，这种方式就是确保两个接口的 GND 先接触连接，TEST, JTAGWDTEN 再被 GND 拉低，此时是调试模式下，让 ULINK2 的 2 个 GPIO 接口与客户 PCB 的 SWIO, SWCLK 调试接口相连。

在客户的 PCB 板上，从芯片的引脚 SWIO, SWCLK 引出到调试的外部插针的位置，串入一个 100 欧姆的电阻。



4.2 仿真接口增加 RST 建议

在使用 IAR 的仿真环境中，因 JLINK 在 IAR 的仿真环境中和目标板建立连接的时间较 Keil MDK 使用 ULINK 的时间更长，如果芯片已烧录了上电之后立刻进入 sleep/hold 模式的应用代码，这时就会造成在 IAR 环境下，无法再次建立连接的现象。

为了解决这样的问题，同时增加仿真烧写的便利，**建议拉出 MCU 的复位脚连接到 ULINK or JLINK 的 PIN 15**（我们推荐的烧写口已经预留了一个口进行连接），利用 J-LINK 发出的 reset 信号复位芯片，在未进入 sleep/hold 模式前，完成通讯（hold 住 cpu），进行烧写。同时，应用软件要求**检测到芯片外部 RST 引脚产生的 reset 信号之后到进入 sleep/hold 前，有 >200ms 的时延**，保证进入 sleep/hold 前，J-LINK 发出的“hold 住 cpu”指令完成。

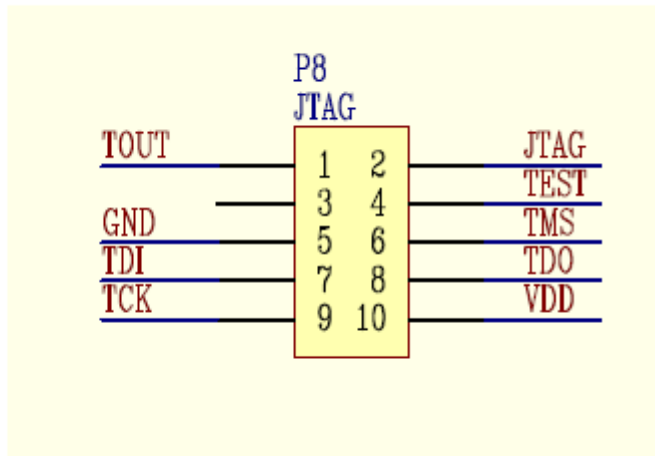
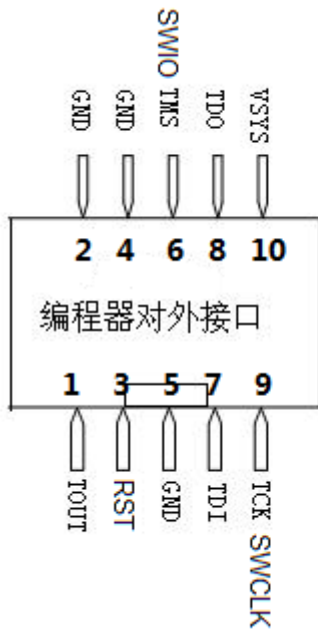
General Ulink Port

1	VCC	2	NC
3	TRST	4	GND
5	TDI	6	GND
7	TMS	8	GND
9	TCK	10	GND
11	RTCK	12	GND
13	TDO	14	GND
15	Reset	16	GND
17	NC	18	GND
19	NC	20	GND

General J-link Port

1	VCC	2	VCC (optional)
3	TRST	4	GND
5	TDI	6	GND
7	TMS	8	GND
9	TCK	10	GND
11	RTCK	12	GND
13	TDO	14	GND
15	Reset	16	GND
17	NC	18	GND
19	NC	20	GND

推荐以及钜泉工具的仿真接口:



注: RST 用于连接 MCU RST PIN 和仿真器的 PIN 15 Reset