

炬泉 MCU HT602X_FAQ

V1.8

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>

版本更新记录

版本号	修改时间	修改内容
V1.0	2014-4-29	初始版本
V1.1	2015-2-20	增加项目
V1.2	2015-7-6	增加项目
V1.3	2015-12-29	1. 增加 HT6023/6025/6027 芯片的问题 2. 删除 RTCIF 清 0 的 bug, 后续新版本已经修正该问题 3. 增加 VCC 与 VRTC 管脚设计问题 4. 增加 startup.s 的配置说明 5. 红外占空比问题后续版本已修改, 该项删除 6. 修改 RTC 定时器定时的描述
V1.4	2016-03-09	1, 修改 1.12 启动文件中 Flash 控制参数配置
V1.41	2016-04-06	1. INT 中断使用注意事项 2. 提升 HT602X 方案的 ESD 性能的方法
V1.5	2016-05-04	1. 增加 1.15 停振检测, 2.4VRTC 应用; 2. 修改 1.10uart 初始化顺序, 1.4OSC_SLPD 版修改说明
V1.6	2016-05-17	修改 1.2RTC 补偿中 MCONxx 默认值. 增加 2.6 芯片版本识别方法 增加 3.2 仿真接口增加 RST 建议 修改 3.2 仿真口串接 1K 电阻, 提升 ESD 优先
	2016-07-12	1.1RTC 补偿参数增加 TPS&RTC 配置 register 初始化, 以减小补偿后的输出跳动。
	2016-07-21	修改 3.2 连接仿真器 RST 描述
V1.7	2016-08-09	1, 修改 1.5 系统时钟切换 POR 和 LBOR 说明 2, 增加 1.15UART\7816 SERL 不可设置为 0; 3, 增加 1.16 HRCADJ 初始化 4, 增加 2.8 GPIO 带载能力描述 5, 修改 2.4 TEST 上拉到 VCC 6, 修改 1.2 外部数字滤波描述 7, 增加 1.18 Toff 对温度测量的误差影响 8, 1.14 增加停振检测的详细描述 9, 增加 1.19 关闭 LRC 10, 增加 1.20 睡眠模式下 RTC 自动补偿 11, 修改 2.5 VRTC 描述, 增加 RTC 模块工作电压描述 12, 增加 1.21 外部 osc 停振的系统状态 13, 增加 1.22 第二套 RTC 的应用 14, 修改 1.13 错误描述 15, 更新 1,1RTC TAB_DfX[]默认值
V1.8	2016-08-29	1, 增加 1.23 PMU 电源检测模块 (VCC_DET、LVDIN_DET 和 BOR_DET 模块) 配置 2, 增加 1.24 不建议使用 UART6 自检功能 3, 增加 1.25 开启 HRC 增加延时

		<ul style="list-style-type: none">4, 增加 1.26 PWM 运行过程中, 应避免修改初始电平位5, 增加 Info block 说明6, 更改 3.6 G 版增加 HT6027B 相关说明7, VDD 更名为 VDD1P5, ADCBAT 更名为 ADCBAT8, 修改 3.3 提高 ESD 性能的方法9, 更改仿真接口 ESD 防护
--	--	--

注: 本文档适用于 HT6023/HT6025/HT6027

目 录

1	Information Block.....	6
1.1	概况	6
1.2	Information Block 的操作说明.....	6
1.3	RTC 补偿系数寄存器和 Info Flash 对应关系.....	7
2	软件.....	8
2.1	RTC 补偿参数.....	8
2.2	RTC 定时器定时.....	11
2.3	CLKCTRL0 配置.....	11
2.4	RTC 时间修改.....	11
2.5	系统时钟切换	11
2.6	在 HardFault 中不能加入软复位或者其他的强制复位的语句	11
2.7	系统时钟配置	12
2.8	ADC 检测功能.....	12
2.9	UART 使用.....	12
2.10	TMPEn 低功耗模式处理.....	12
2.11	startup.s 文件中 Flash 地址 FC0 处推荐配置.....	12
2.12	使用 INT 中断和 RX 中断的注意事项	13
2.13	仿真接口 SWIO 的配置	13
2.14	停振检测使用的建议	13
2.15	UART\7816 SERL 不能设置为 0.....	14
2.16	HRCADJ 初始化.....	14
2.17	低功耗模式 TOUT 输出.....	14
2.18	Toff 对温度测量的误差影响.....	14
2.19	关闭 LRC.....	15
2.20	睡眠模式下，RTC 可自动进行补偿.....	16
2.21	外部 OSC 停振的系统状态.....	16
2.22	第二套 RTC 的应用.....	17
2.23	PMU 电源检测模块（VCC_DET、LVDIN_DET 和 BOR_DET 模块）配置	17
2.24	不建议使用 UART6 的 485 极性判断状态功能.....	17
2.25	开启 HRC 需增加延时	17
2.26	PWM 运行过程中，应避免修改初始电平位	17
3	硬件.....	18
3.1	外部 32K 晶振应用说明	18
3.2	VCC 与 VRTC 管脚.....	18
3.3	提升 HT602X 方案的 ESD 性能的方法.....	18
3.4	Test 管脚.....	20
3.5	VRTC 的使用	20
3.6	HT6027B 封装及说明（G 版）	20
3.7	芯片版本识别方法	23
3.8	GPIO 带载能力描述.....	24

4	可靠性问题	25
4.1	仿真接口设计	25
4.2	仿真接口增加 RST 建议	26

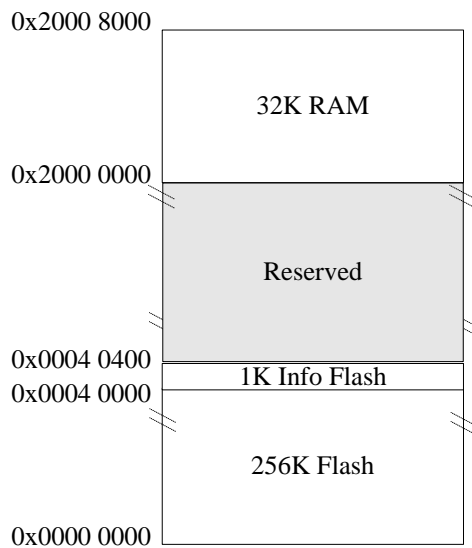
1 Information Block

1.1 概况

HT6X2X 内部有 1K bytes 的 Information Block 空间, 用于存储出厂信息, 地址位于 0x40000~0x40400, 特性如下:

- Information Block: 1k bytes/page
- 擦写次数: 100,000 次
- 数据保持时间: 20 年 (min)
- 操作温度: -40 度到+105 度

在内存中的位置:



1.2 Information Block 的操作说明

将 Flash 的 00000FC1H 地址写入非 0FFH 的值后, 开启读保护功能, 256K Flash 空间的数据无法读出, Information Block 中的数据依旧可以读出。

Information Block 的写/页擦除/全擦除操作与 Code Flash 的写/擦除操作方式是一样的, 区别在于 Flash 地址不一样, 以及对 Information Block 操作还需要再配置一个解锁的寄存器, 如下所示:

```
FLASHLOCK = 0x7A68;          //unlock flash memory
INFOLOCK = 0xF998;          //unlock information Block memory
```

上面的两个解锁寄存器都需要配置, 用户才可以操作 Information Block, 其他操作同上 Code Flash 操作。

Information Block 共 1024bytes (0x00040000~0x000403FF), 分 1 页, 1kByte/ 页, 其中地址 0x00040100~0x000401FF 存储有芯片出厂信息, 请勿进行写/擦除操作。

INFOLOCK (InfoFlash 锁定寄存器)		基地址: 0x4000F000 偏移地址: 50H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	KEY[15:8]							
Write:								

Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	KEY[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
KEY[15:0]	<p>Information Block 锁定控制</p> <p>对该寄存器写入 0xF998 后, Information Block 被解锁, 用户可以写操作 Information Block。</p> <p>写入非 0xF998 数据后, Information Block 被锁定, 用户禁止写操作 Information Block。</p> <p>默认为锁定状态, Information Block 不可执行写/页擦除/全擦除 操作</p> <p>用户写入的是 0xF998, 读出值为 1; 写入的是非 0xF998, 读出值为 0</p>

1.3 RTC 补偿系数寄存器和 Info Flash 对应关系

RTC 的补偿系数寄存器在芯片复位时 (除唤醒复位) 会自动装载, 考虑到系统的可靠性, 用户可以软件读取 Information Block 的对应地址的值, 然后再写入到对应的 RTC 模块的寄存器中。

寄存器偏移地址	寄存器名称	Information Block 对应偏移地址 (Information Block 基地址: 0x00040000)
0x50	DFAH	0x104
0x54	DFAL	0x108
0x58	DFBH	0x10C
0x5C	DFBL	0x110
0x60	DFCH	0x114
0x64	DFCL	0x118
0x68	DFDH	0x11C
0x6C	DFDL	0x120
0x70	DFEH	0x124
0x74	DFEL	0x128
0x78	Toff	0x12C
0x7C	MCON01	0x130
0x80	MCON23	0x134
0x84	MCON45	0x138
----	----	0x13c
HRCADJ 装载调整值	HRCADJ	0x140
----	~HRCADJ	0x144

注: 1. 当 flash 地址 0xFC0 的 bit2 为 1 时, 上电后才会将 InfoFlash 中的数据自动装载到寄存器中。

2. info flash 偏移地址 0x13c 处存放从 0x104 到 0x138 的 32 位无符号数累加和, 如果累加和正确, 则

说明 info 中数据有效，否则认为 info 中数据无效

3.MCONxx 寄存器为内部控制寄存器，上电后也要一同装载。如果 Info 中数据不对（判断 Checksum），则这些数据写默认值。如果 Info 中数据正确，则装载 Info 中数据。

4. “HRCADJ 装载调整值”是不会芯片自动装载的，用户需要通过程序将 information block 的地址 0x140 中的数据读出，然后和 0x144（HRCADJ 装载调整值补码）中的值做比较，如果两个值互反的话，则说明 information block 的 0x140 中存储的对 HRCADJ 初始调整的数据是正确的，否则是错误的，用户不能装载其中的数据，详情参见 FAQ2.16。

5. 以上补偿系数在 HT622X 里面，已经经过出厂校正，补偿参数已经写入相应地址，无需再次校正。

2 软件

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=0x04; // //8s 打开 1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 DataArray[15];
}HT_Info_Typedef;

#define HT_InfoData_Base (0x00040000+0x104)
#define HT_Info ((HT_Info_Typedef *) HT_InfoData_Base )

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
}
else
{
    //RTC 补偿系数正确

    //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
}

```

若 information block 保存的 RTC 补偿系数遭到破坏，可初始化为下列常数，以获得正确的 RTC 补偿：
HT602X(外置晶振)：

```
static const uint16_t TAB_DFX[10] =
{
    0x0000, 0x0000,
    0x007F, 0xD64C
    0x007E, 0xD708,
    0x0000, 0x546E,
    0x0000, 0x04B0
};
#define C_Toff      0x0000           //温度偏置寄存器
#define C_MCON01   0x2000           //控制系数 01
#define C_MCON23   0x0588           //控制系数 23
#define C_MCON45   0x4488           //控制系数 45

```

//当 Info block 遭到破坏，RTC 补偿系数初始化用默认值：

```
HT_RTC->Toff    = C_Toff;
HT_RTC->MCON01=C_MCON01;
HT_RTC->MCON23=C_MCON23;
HT_RTC->MCON45=C_MCON45;

HT_RTC->DFAH=TAB_DFX[0];
HT_RTC->DFAL=TAB_DFX[1];

HT_RTC->DFBH=TAB_DFX[2];
HT_RTC->DFBL=TAB_DFX[3];

HT_RTC->DFCH=TAB_DFX[4];

```

```
HT_RTC->DFCL=TAB_DFx[5];
```

```
HT_RTC->DFDH=TAB_DFx[6];
```

```
HT_RTC->DFDL=TAB_DFx[7];
```

```
HT_RTC->DFEH=TAB_DFx[8];
```

```
HT_RTC->DFEL=TAB_DFx[9];
```

2.2 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)

2.3 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 (D 版更改为只有小功耗模式, 此位写 0 无效);

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。

2.4 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。

2.5 系统时钟切换

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 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, JTAG 也无法连接芯片以及擦除下载程序。

2.7 系统时钟配置

PLL 最高时钟频率为 44M，若 $f_{cpu}=44\text{MHz}$ ，需使能指令预取功能。
 介于 EMC 性能考虑，用户使用时，推荐使用 22M 时钟，或者更低频率。

2.8 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 3.3V，否则内部 ADC 模块可能会被烧坏。

2.9 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，必须 TXEN 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：

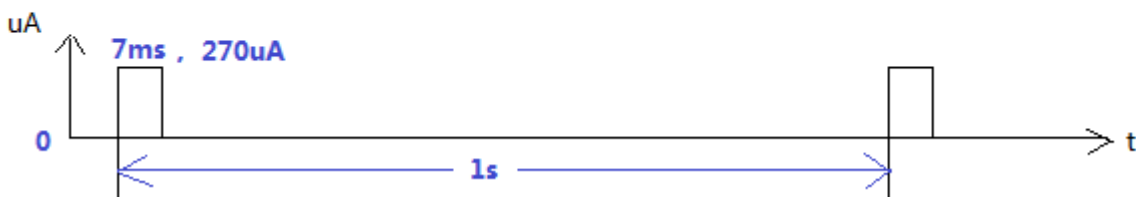
- 发生复位（比如软复位/WDT 复位等）
- 使能 TXEN 和 CLKCTRL0 寄存器中 UART 时钟使能位，让未发送完的数据发送完成即可

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

2.10 TMPEn 低功耗模式处理

为保证 RTC 正确补偿，建议始终保持温度采样使能（TMPEn 开启）状态，在进入低功耗之前，为降低功耗，可加宽 TPSPRD 分时开启的时间间隔以降低功耗。可根据实际需求配置分时开启的时间间隔，

TPS 分时1s开启一次的 功耗&开启时间 示意图



2.11 startup.s 文件中 Flash 地址 FC0 处推荐配置

关于 startup.s 文件中的 CRP_KEY 的配置推荐说明如下表格：

```

102         IF      :LNOT::DEF:NO_CRP
103         AREA   |.ARM.__at_0x0FC0|, CODE, READONLY
104 CRP_Key   DCD   0xF7F5FFA2
105         ENDIF
    
```

芯片	推荐值	推荐值说明
HT602X	推荐配置 1 X7F5FFA2	Flash 不加密，

			Sleep/Hold 下 WDT 关闭
	推荐配置 2	X7FFFA2	Flash 不加密， Sleep/Hold 下 WDT 开启
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 SW 口读出），需将红色 FF 变成非 FF 即可。比如：F7FF00A2 注 3: 蓝色 5 为 Sleep/Hold 下 WDT 控制位，“5”表示 Sleep/Hold 下 WDT 关闭；非“5”表示 Sleep/Hold 下 WDT 开启 注 4: 黄色高亮为 BORreset 控制位，“7”为默认配置			

2.12 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。

2.13 仿真接口 SWIO 的配置

应注意 PB[13]/SWIO 引脚不能配置为开漏，否则会造成只能第一次下载成功，第二次就无法下载。SWIO 外部如果有加上拉电阻的话，第二次下载调低通信速度有可能能下载。

2.14 停振检测使用的建议

由于停振检测采用内部 LRC 作为检测时钟，当 LRC 误差较大时，会导致 PLL 停振检测误判。

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。

停振检测的详细描述如下：

1. 时钟 Flf, Fp11, Fhrc 具有停振检测功能。Flrc 默认为不会发生停振（设计保证），做为其他 3 个时钟停振检测模块的时钟源；
2. Flf 停振检测功能默认开启，3 个停振检测模块均可以由用户软件关闭，控制位分别为 CLKCTRL0.LF_DET_EN, CLKCTRL0.PLL_DET_EN, CLKCTRL0.HRC_DET_EN。
3. 当 Flf 停振检测开启并检测到停振，系统给出时钟停振标志 LF_FLAG；若系统时钟 Fsys 选择 Flf 或 Fp11（Flf 为 Fp11 的时钟源）时，系统会由硬件强制将系统时钟 Fsys 切换到内部低频 RC 时钟 Flrc，且产生中断（NMI 中断），同时将寄存器 SYSCLK_SEL[2:0] 的值置为 000；若系统时钟 Fsys 选择非 Flf 和 Fp11 时，不会切换时钟，也不会发生 NMI 中断。
4. 当 Fp11 停振检测开启并检测到停振，系统给出时钟停振标志 PLL_FLAG，若系统时钟 Fsys 选择 Fp11 时，

系统会由硬件强制将系统时钟 Fsys 切换到内部低频 RC 时钟 Flrc, 且产生中断 (NMI 中断), 同时将寄存器 SYSCLK_SEL[2:0] 的值置为 000; 若系统时钟 Fsys 选择非 Fp11 时, 不会切换时钟, 也不会发生 NMI 中断。

5. 当 Fhrc 停振检测开启并检测到停振, 系统给出标志位 HRC_FALG, 若系统时钟 Fsys 选择 Fhrc 时, 系统不会由硬件强制切换系统时钟, 此时系统将停止运行, 等待看门狗复位。

6. Flf 停振标志 LF_FLAG 控制 PMU 和 LCD 模块以及复位预热计时模块的时钟选择 (Flrc 或 Flf), 若 Flf 停振检测功能关闭, LF_FLAG 固定为 0, PMU 和 LCD 模块以及复位预热计时模块的时钟会强制选择 Flf, 若此时 Flf 停振, 会影响 PMU 和 LCD 模块以及复位预热计时模块功能。建议系统正常运行后, 开启 Flf 停振检测。

2.15 UART\7816 SERL 不能设置为 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。

2.16 HRCADJ 初始化

芯片启动时默认运行在 HRC, HRC 的初始频率即使没有经过 HRCADJ 的调整, 也不影响系统启动运行。如果客户需要提升内部 HRC 的频率准确性, 以适应特殊的应用 (例如作为串口的波特率时钟源), 建议加载 Info 的调整值到此寄存器, 以提高精度。

information block 的 0x00040140 地址存放的 HRCADJ 调整值主要是对 HRC 的初值做调整, 目的是让不同的芯片的 HRC 的初值控制在 11.01MHz±1% (常温), 11.01MHz±3% (全温度范围-40°C~85°C) 的精度以内。

“HRCADJ 高频 RC 装载调整值”芯片是不会自动装载的, 用户需要通过程序将 information block 的地址 0x00040140 中的数据读出, 然后和 0x00040144 (HRCADJ 用户装载调整值补码) 中的值做比较, 如果两个值互反的话, 则说明 information block 的 0x00040140 中存储的对 HRCADJ 初始调整的数据是正确的, 否则是错误的, 用户不能装载其中的数据, 用户可以设置 HRCADJ 的寄存器值为 0x40, 那么不同芯片的 HRC 的初值差异在 10.4MHz±3% (F 版 常温) 以内, 经过加载 information block 的校正值可将 HRC 频率调整到 11MHz±1% (常温)。

2.17 低功耗模式 TOUT 输出

系统进入 hold\sleep 模式之后, 如果 PLL_En 打开, TOUT 输出经高频补偿后的分频输出, 如果关闭 PLL_En, TOUT 可正常输出未经高频补偿的分频信号。

2.18 Toff 对温度测量的误差影响

温度传感器初始偏执 Toff: 定义为当环境温度为 0 度时, 内部温度传感器模块读到的 code 值, 即 TMPDAT 的值。每颗芯片不同, 需要校准。

(1) 对于一般的温度显示功能, 不用校准 Toff, 使用手册上的温度转换公式即可:

$$\text{温度 } Tr = 12.9852 - \text{TMPDAT} * 0.0028 \quad \text{其中: } Tr \text{ 为实际的温度 (}^\circ\text{C)}$$

上述公式没有计入 Toff 影响, 在全温状态下 (-40°C~+85°C) 引起的误差在 2°C 附近。

(2) 如果用户想提高温度测量精度, 可将 Toff 影响考虑在内, 使用如下公式:

$$\text{温度 } Tr = 12.9852 - (\text{TMPDAT} - \text{Toff}) * 0.0028 \quad \text{其中: } Tr \text{ 为实际的温度 (}^\circ\text{C)}$$

2.19 关闭 LRC

内部低频振荡器默认一直开启，功耗为 0.3uA。

对于一些特殊应用，需要关闭内部 LRC，可依照如下方法：

```
void close_wdt(void)
{
    HT_RTC->CTRLBYFLASH &= 0xFFFD;
    HT_RTC->LRCCOMAND = 0x5555;
    HT_RTC->LRCCOMAND = 0xAAAA;
    HT_WDT->WDTCLR = 0xAA3E; //clear WDT
}
```

恢复打开 LRC：

```
void open_wdt(void)
{
    HT_RTC->CTRLBYFLASH |= 0x02;
}
```

相关内部寄存器描述如下：

CTRLBYFLASH 即 Flash 控制寄存器，在 2.4 章节有相关描述，LRC_CTRL 位在手册上做了屏蔽。

CTRLBYFLASH Flash 控制功能说明			基地址：0x4000C000 偏移地址：100H					
Flash 地址	Bit7	6	5	4	3	2	1	Bit0
FC2H	RESERVE D	RESERVE D	RESERVE D	RESERVE D	WDT_EN [3]	WDT_EN [2]	WDT_EN [1]	WDT_EN [0]
Reset:	1	1	1	1	1	1	1	1
Flash 地址	Bit7	6	5	4	3	2	1	Bit0
FC1H	FLASH[7:0]							
Reset:	1	1	1	1	1	1	1	1
Flash 地址	Bit7	6	5	4	3	2	1	Bit0
FC0H	RESERVE D	x	RESERVE D	x	RTCRST	AUTOREL OAD	LRC_CTRL	x
Reset:	1	0	1	0	1	1	1	0

位	功能描述
FC2H BIT[7:4]	内部控制位，需保持 1111bit。
WDT_EN[3:0]	看门狗使能 =0101B: 看门狗在 Sleep 和 Hold 模式下关闭 =其他值: 看门狗在 Sleep 和 Hold 模式下开启 注：因为 WDT 是 LRC 驱动，如果关闭了 LRC，则看门狗自动关闭
FLASH[7:0]	代码空间加密位

	如果 Flash[7:0]=0xFF, 则 Flash 不加密 其他: Flash 加密, 该寄存器位只能读, 不能写
FCOH BIT[7: 5]	内部控制位, 需保持 101bit
RTRCST	RTC 复位控制位 =1, LBOR, POR 可以复位 RTC 计时寄存器 =0, LBOR, POR 不能复位 RTC 计时寄存器
AUTORELOAD	自动装载使能位 =1, 自动装载功能使能 =0, 自动装载功能屏蔽
LRC_CTRL	=1: 打开 LRC =0: 关闭 LRC

注: 其他位不可以修改, 保持默认值。

LRCCOMMAND (LRC 写关闭命令寄存器)		基地址: 0x400C000 偏移地址: 104H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
Bit[15:0]	在 LRC_CTRL 等于 0 情况下, 向 LRCCOMMAND 寄存器先写入 0x5555 再写入 0xAAAA, 才能真正关闭 LRC。 LRC_CTRL 参见 Flash 控制功能

2.20 睡眠模式下, RTC 可自动进行补偿

可以补偿, 但需要在睡眠模式下保持温度采样(TBSCON.TMPE_n=1)开启, 并根据需要确定温度传感器的采样周期(TBSPRD.TMPPRD[2:0])。系统在进入睡眠模式之后, TBS 模块按此设定的采样周自动开启并采样温度值, 采样时间持续 7ms, 结果更新于 TMPDAT 寄存器。RTC 温补模块硬件自动读取该采样值, 并计算实时 RTC 补偿值, 进行温补。温补功能不需客户软件参与, 不需唤醒。

2.21 外部 OSC 停振的系统状态

如果 Fosc 停振, 若当前系统时钟不是 Fosc 或 Fpll (由 Fosc 倍频), 那么 CPU 不会停止; 若当前系统时钟选择了 Fosc 或 Fpll, 同时开启的相应的停振检测, 那么硬件自动将系统时钟切换到 Flrc, 并进入 NMI 中断, 用户可以作相应的记录或措施; 若当前系统时钟选择了 Fosc 或 Fpll, 但没有开启相应的停振检测, 那么 CPU 会停止, 等待 WDT 复位。

2.22 第二套 RTC 的应用

第一套 RTC 以 Fosc 为时钟源，内部可进行高低频补偿；第二套 RTC 是以 Flrc 为时钟源，精度受 LRC 精度影响。两者精度显然有差异，但假设 OSC 停振后，第一套将不再计时，而第二套不受影响。MCU 中做两套 RTC 的本意就是，为了可靠性和精度，两套互为备份。但在两套都正常时，必须以第一套为准，如果需要采用第二套 RTC 作为备份，需要在第一套 RTC 正常运行时定期校准第二套 RTC。

2.23 PMU 电源检测模块（VCC_DET、LVDIN_DET 和 BOR_DET 模块）配置

在进入低功耗模式 Hold/Sleep 时，系统内建 PMU 检测模块自动进入分时开启模式来进一步降低系统功耗。配置寄存器 VDETPCFG，其中：

开启周期 VDET_TIME[1:0]，默认值为 00, 300us

开启时间 VDET_PRD[2:0]，默认值为 010, 67ms

VCC_DET、LVDIN_DET 和 BOR_DET 内置 200us 模拟滤波电路，在极端低温（-40℃）情况下，这个滤波时间存在变大的可能性，最大可达到 400us，VDET_TIME 设定的开启时间为数字电路产生，不会随温度变化。当模拟滤波时间超过设定的开启时间，即还未检测完毕即关闭，这样会产生误检信号而导致错误的复位，为可靠起见，设计应留有余量，推荐配置 VDET_TIME[1:0]=11，为最大检测时间 1068us，这时，分时开启周期可适当加长，总体功耗增加<1uA。

2.24 不建议使用 UART6 的 485 极性判断状态功能

串口状态寄存器 UARTSTA 的 POLASTA(bit4) 可以指示外接 485 的极性状态。

但建议 UART6 使用中，不要使用这个极性自检状态，UART6 的这个状态位不准确，可使用其它串口通过此位识别 485 极性。

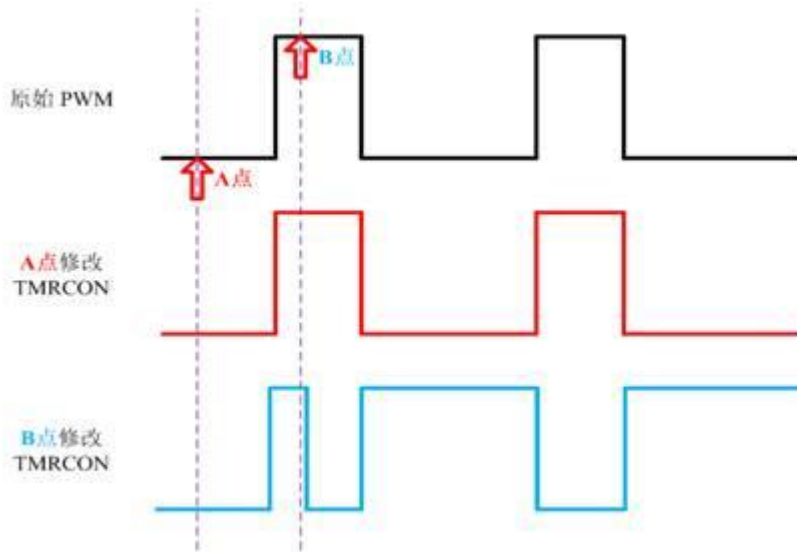
2.25 开启 HRC 需增加延时

内部 HRC 时钟频率将在 H 版修改为 11.01MHz（Info block 中标志存储于 0x00040184，标志值为 0x0000C33C）。可以选择该高频 RC 时钟的分频输出时钟（分频设置位为 HRC_DIV[1:0]）作为系统时钟（SYSCLK_SEL[2:0]=010）。系统复位后，系统时钟默认选择 HRC。若关闭了 HRC，再次开启时，为确保 HRC 开启运行的稳定可靠，建议增加 1ms 左右延时，配置举例如下：

```
HT_CMU->WPREG = 0xA55A;           //解除写保护
HT_CMU->CLKCTRL0 |= 0x0020;        //使能 HRC
HT_CMU->HRCDIV = 0x0000;           //HRC 时钟分频设置
Wait1mS();
HT_CMU->SYSCLKCFG = 0x0082;         //Fsys = Fhrc
HT_CMU->SYSCLKDIV = 0x0000;        //Fcpu = Fsys
HT_CMU->WPREG = 0x0000;           //写保护有效
```

2.26 PWM 运行过程中，应避免修改初始电平位

Timer 选择 PWM 模式输出，若在 PWM 输出过程中修改初始电平控制位 TMRCON.PWMHL（BIT6），输出电平会被重新配置为设置电平，会有较大概率导致 PWM 输出异常，示意图如下：



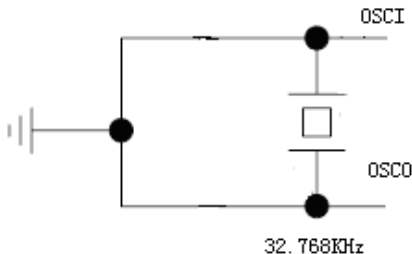
为避免这种问题出现，建议软件操作：

在维护 registers 期间发现读出 TMRCON 的值发生了变化，或者功能上需要更改 TMRCON 的值，需要首先关闭计数器使能位 TMRCON.CNTEN，重新写 TMRCON 之后，再次开启计数器使能，输出 PWM。

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，可能会影响芯片的可靠性。

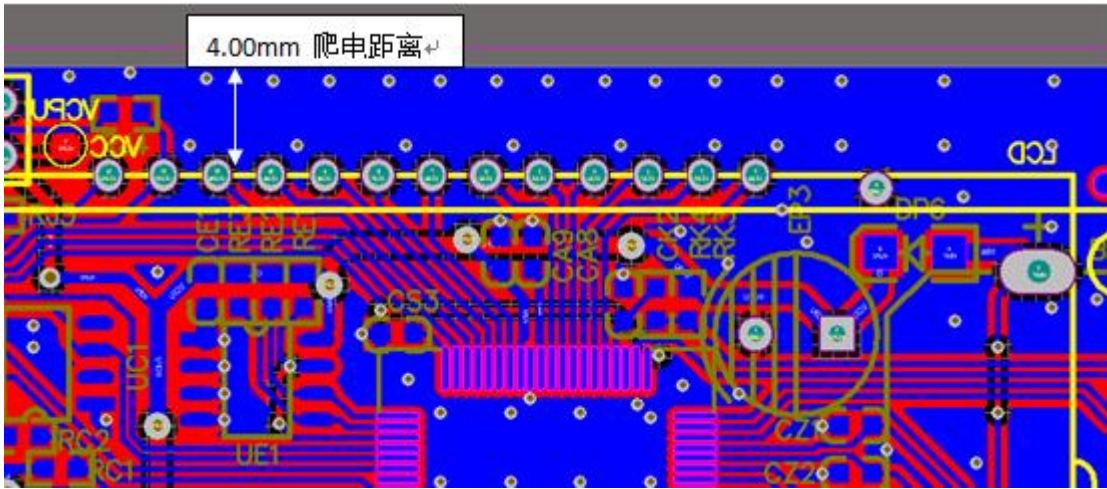


3.2 VCC 与 VRTC 管脚

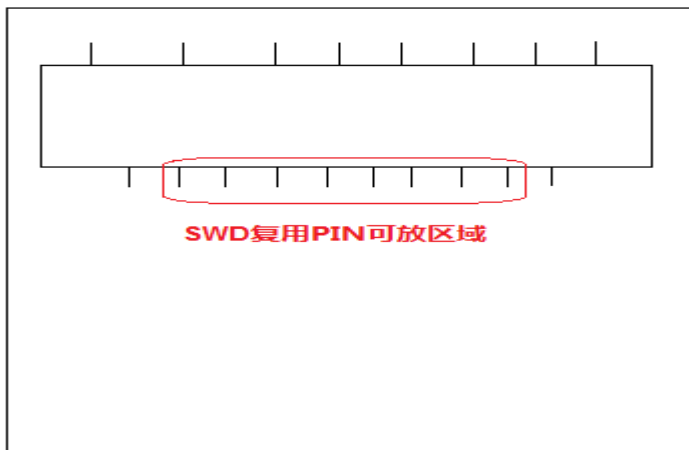
硬件电路设计时，必须将 VCC 与 VRTC 管脚连在一起。如果是做 RTC 独立供电设计，建议 VRTC 一直保证有电。

3.3 提升 HT602X 方案的 ESD 性能的方法

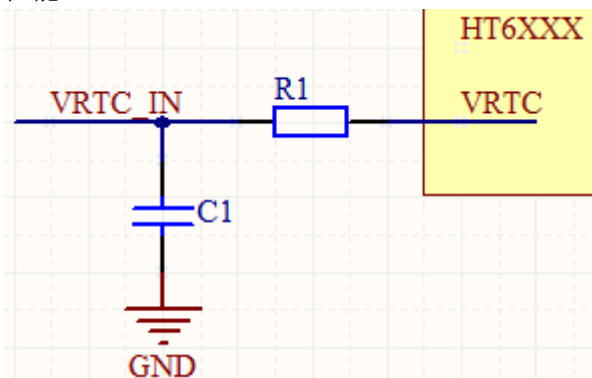
- 1, LCD 焊盘距离板边保证至少 4mm 的爬电距离，使芯片不易受到 ESD 辐射的影响。芯片尽量下移并拓宽顶部接地面积。如下图所示，其中白色双向箭头表示的是爬电距离。



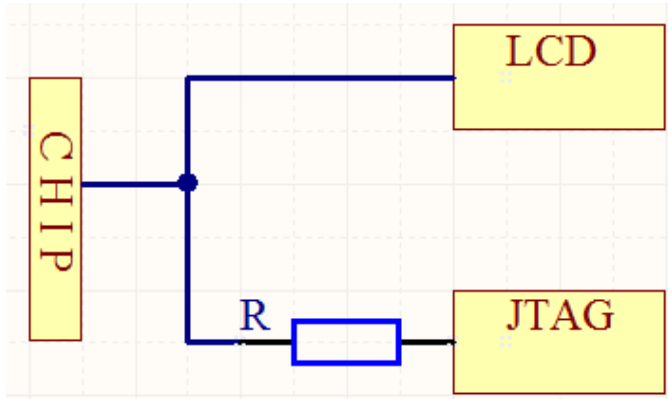
- 2, 往下移动芯片和避开电表挂钩, 使芯片尽量远离表壳, 可改善 800KV 高压包对电表的影响和整表 ESD 性能。
- 3, VDD1P5pin 到 GND 之间串接 150-200ohm 电阻和 0.1uF 电容(靠近芯片), 确保 VDD1P5 良好的接地, 可改善 800KV 高压包对电表的影响。
- 4, 尽量避免 SW 和 SEG 复用, 如果 SW 和 SEG 复用, 尽量不用液晶边缘的管脚(如下图)和 SWIO、SWCLK 两个管脚, 推荐对地各加 1000pF 电容(靠近芯片), 可进一步提升整表 ESD 性能。



- 5, VRTC 管脚串接小于 100 欧姆电阻(推荐 50 欧姆, 靠近芯片), 电容推荐 105pF, 可进一步提升整表 ESD 性能。



- 6, 现推荐靠近 JTAG 接口串接 100 欧姆电阻滤波, 串接电阻作为辅助功能, 防护 JTAG 插拔影响, 也防护从 JTAG 端引进的 ESD 影响, 此项在不影响 ESD 性能的前提下可选。



3.4 Test 管脚

TEST 管脚接一个 1K 电阻上拉到 VCC 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。

3.5 VRTC 的使用

建议在 VCC 有电的情况下，确保 VRTC 同时供电，否则会造成 Flash 无法操作。

MCU 芯片内部分 2 个电源域：VCC 与 VRTC。在芯片外部是 2 个独立的供电引脚。

VCC 供 CPU 部分；VRTC 供 RTC 相关部分（包括 32.768KHz 晶体, TBS 温度采样模块，温补分频电路）；

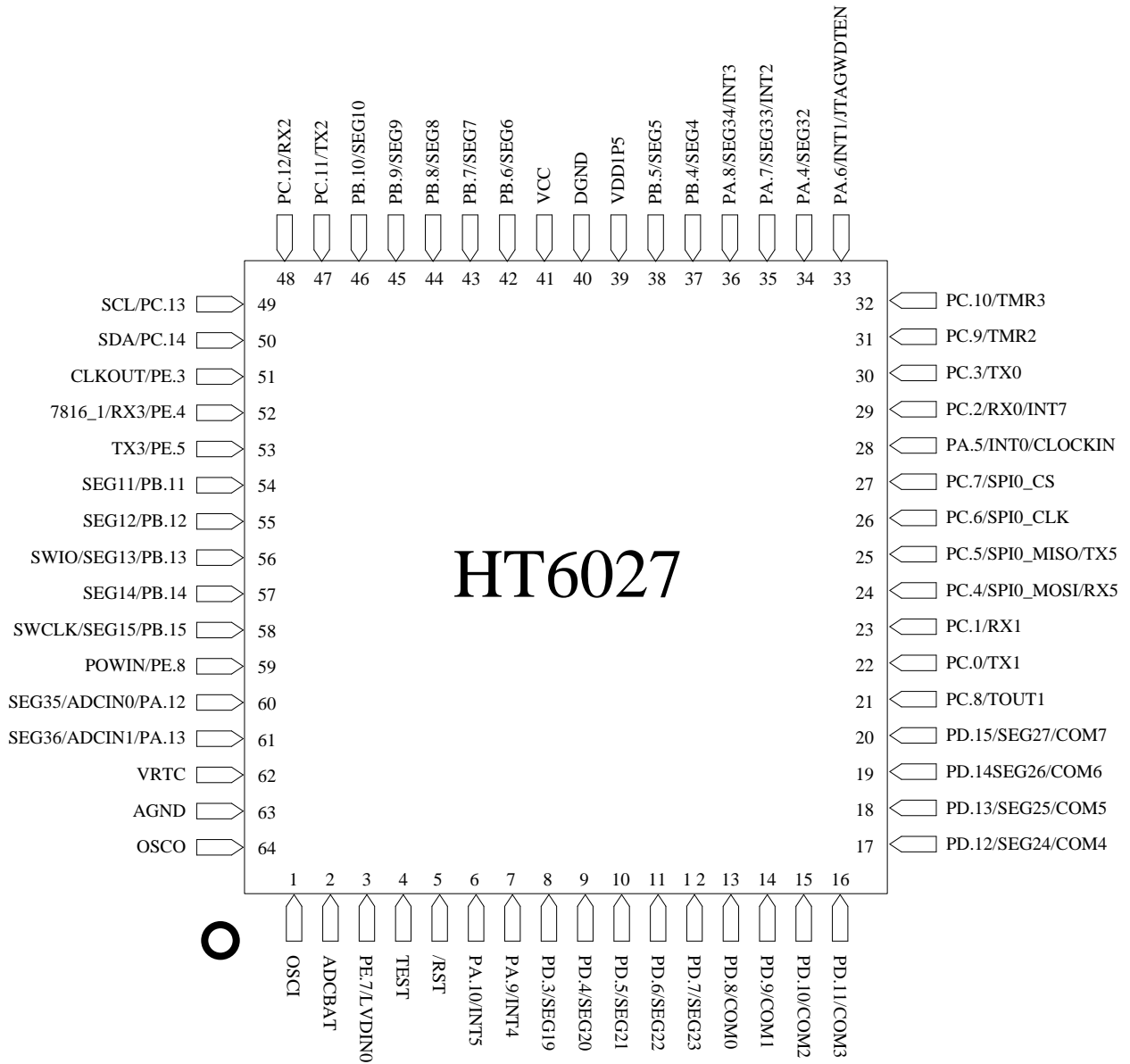
VRTC 供 RTC 部分本身只要高于 1V，就能计时，否则时间数据会丢失。

VRTC 供电 > 2V 时，如果已开启了温度采样功能，那么温补功能正常。

VRTC 供电在 1V~2V 时，因为温度采样不准，所以温度补偿功能失效，一般情况下此时误差 30s/d，极端情况下偏差 360s/d。

3.6 HT6027B 封装及说明（G 版）

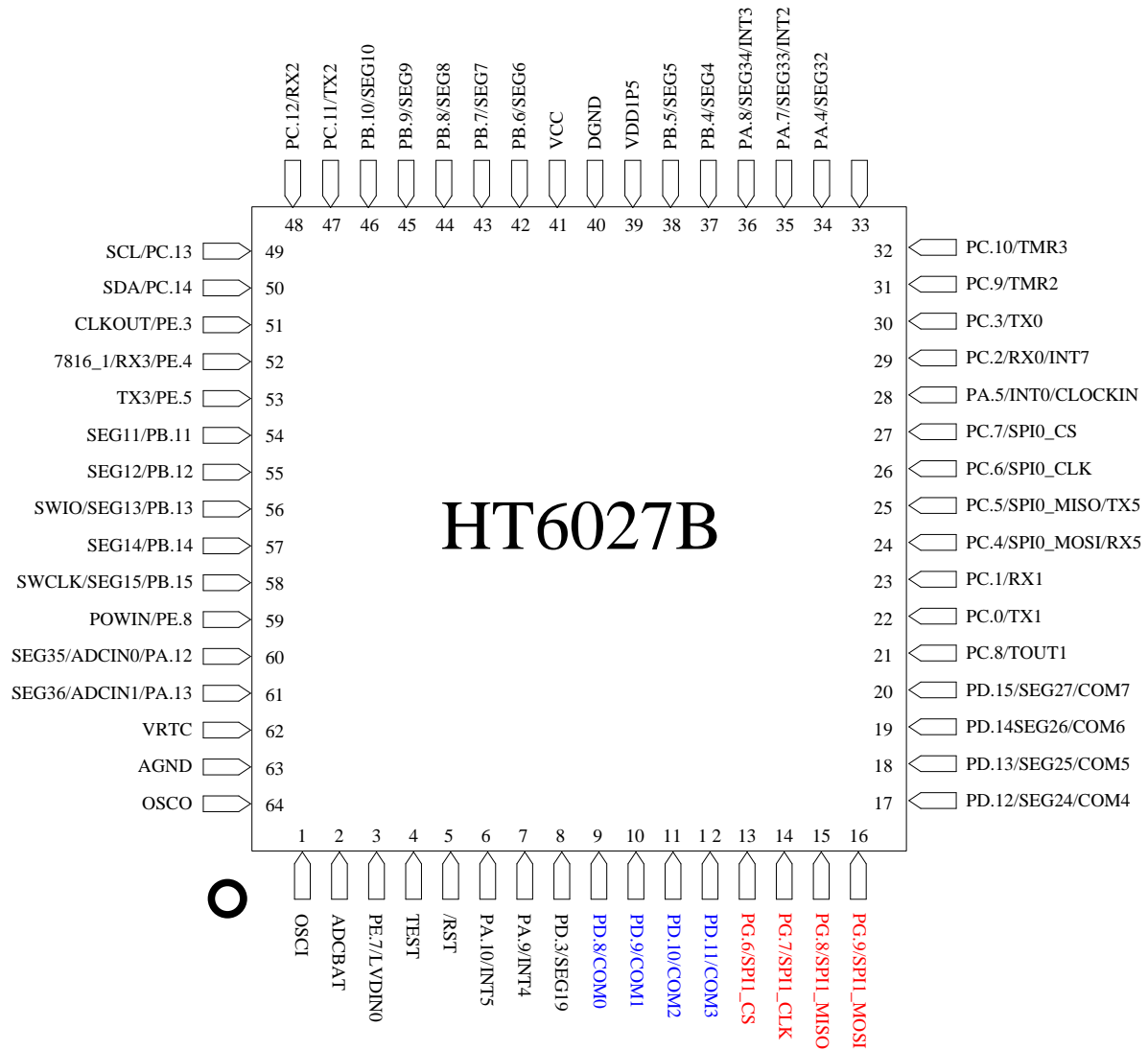
原 HT6027 64PIN 封装如下：



HT6027 引脚定义 (原有):

14↔	13↔	9↔	PD. 4↔	I/O↔	SEG20↔	
15↔	14↔	10↔	PD. 5↔	I/O↔	SEG21↔	
16↔	15↔	11↔	PD. 6↔	I/O↔	SEG22↔	
17↔	16↔	12↔	PD. 7↔	I/O↔	SEG23↔	
18↔	17↔	13↔	PD. 8↔	I/O↔	COM0↔	
19↔	18↔	14↔	PD. 9↔	I/O↔	COM1↔	
20↔	19↔	15↔	PD. 10↔	I/O↔	COM2↔	
21↔	20↔	16↔	PD. 11↔	I/O↔	COM3↔	
22↔	↔	↔	PG. 6↔	I/O↔	SPI1_CS↔	
23↔	↔	↔	PG. 7↔	I/O↔	SPI1_CLK↔	
24↔	↔	↔	PG. 8↔	I/O↔	SPI1_MISO↔	
25↔	↔	↔	PG. 9↔	I/O↔	SPI1_MOSI↔	

G 版之后，增加 64PIN HT6027B 封装将更改为双 SPI 驱动模式，封装图更新为（新增）：



HT6027B 引脚定义如下（G 版之后）：

14↔	13↔	↔	PD. 8↔	I/O↔	SEG20↔
15↔	14↔	↔	PD. 5↔	I/O↔	SEG21↔
16↔	15↔	↔	PD. 6↔	I/O↔	SEG22↔
17↔	16↔	↔	PD. 7↔	I/O↔	SEG23↔
18↔	17↔	9↔	PD. 8↔	I/O↔	COM0↔
19↔	18↔	10↔	PD. 9↔	I/O↔	COM1↔
20↔	19↔	11↔	PD. 10↔	I/O↔	COM2↔
21↔	20↔	12↔	PD. 11↔	I/O↔	COM3↔
22↔	↔	13↔	PG. 6↔	I/O↔	SPI1_CS↔
23↔	↔	14↔	PG. 7↔	I/O↔	SPI1_CLK↔
24↔	↔	15↔	PG. 8↔	I/O↔	SPI1_MISO↔
25↔	↔	16↔	PG. 9↔	I/O↔	SPI1_MOSI↔

此更改产生的变化：

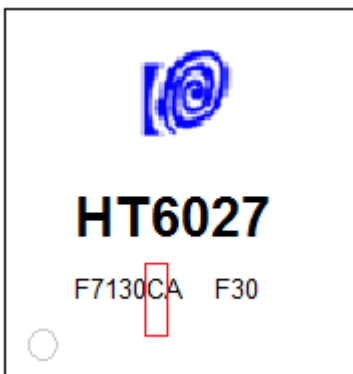
LCD 驱动的 COM0~3 PIN 脚改变，同时改动将减少 4 个 seg 口 (SEG20\21\22\23)，液晶驱动段数将减少为：

4COM*22SEG = 88 段

6COM*24SEG = 120 段

8COM*22SET = 144 段

3.7 芯片版本识别方法



例如上图：上图为芯片正面俯视图，红框字母代表了此芯片的版本号，上图红框字母为 C，说明此芯片为 C 版本。

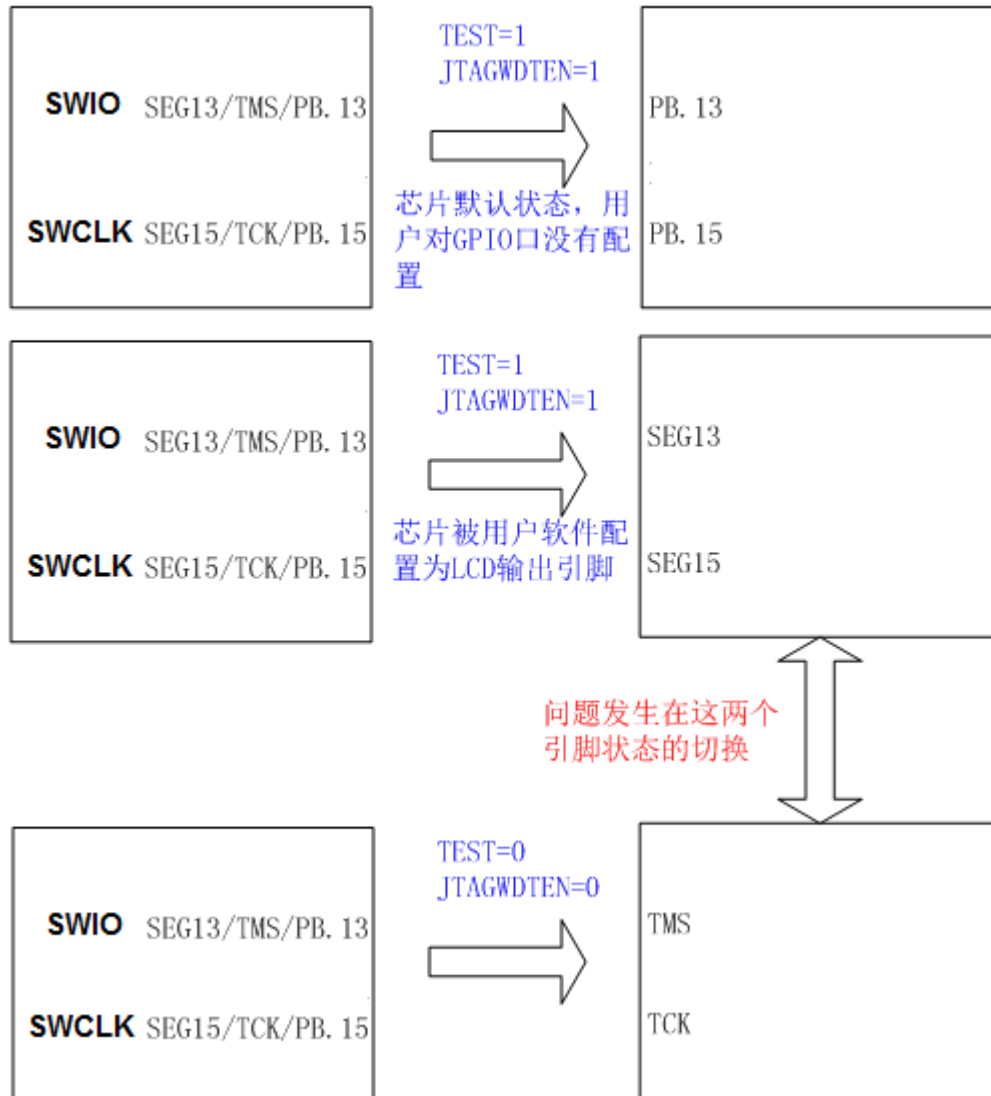
3.8 GPIO 带载能力描述

HT602X 的 GPIO 根据带载能力分为 3 中类型，具体如下：

符号	参数说明	测试条件	最小	典型	最大	单位
Ioh (大电流管脚)	高电平输出电流	VCC=5V I/O 口上电压 Vio 降低到 0.9VCC 测试引脚为： PA. 6, PA. 7, PA. 8, PC. 0		15		mA
Iol (大电流管脚)	低电平输出电流	VCC=5V I/O 口上电压 Vio 升高到 0.1VCC 测试引脚为： PA. 6, PA. 7, PA. 8, PC. 0		30		mA
Ioh (普通管脚)	高电平输出电流	VCC=5V I/O 口上电压 Vio 降低到 0.9VCC		5		mA
Iol (普通管脚)	低电平输出电流	VCC=5V I/O 口上电压 Vio 升高到 0.1VCC		9		mA
Idd	输入电流	VCC 电源引脚			50	mA
Iss	地上电流	所有 GND 引脚			50	mA

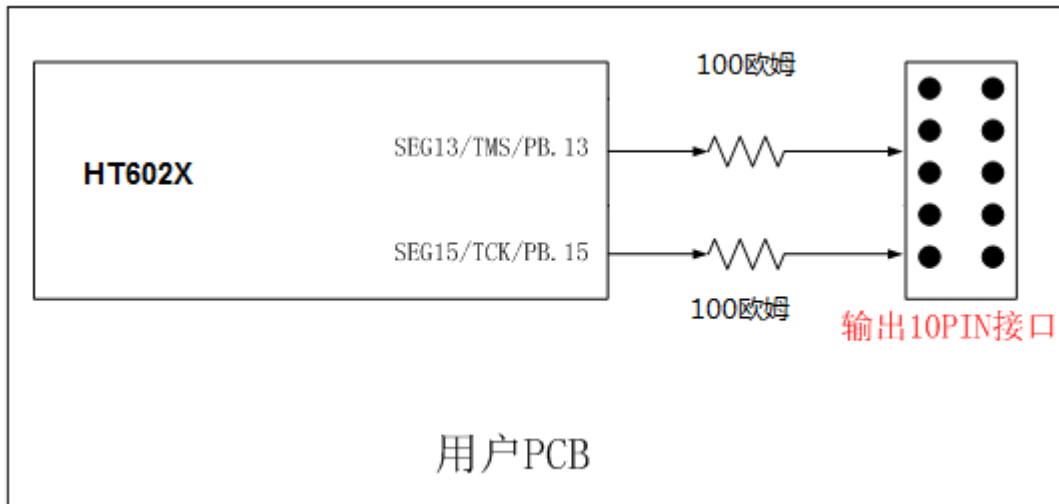
4 可靠性问题

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则有导致 LCD 模块内部损坏的可能。规避方法如下：

- (1) 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 的其它 GPIO 接口与客户 PCB 的 TMS, TCK 调试接口相连。
- (2) 在客户的 PCB 板上，从芯片的引脚 TMS, TCK 引出到调试的外部插针的位置，靠近 SWD 接口串接 100 欧姆电阻滤波，如下图所示：



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 MCU 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

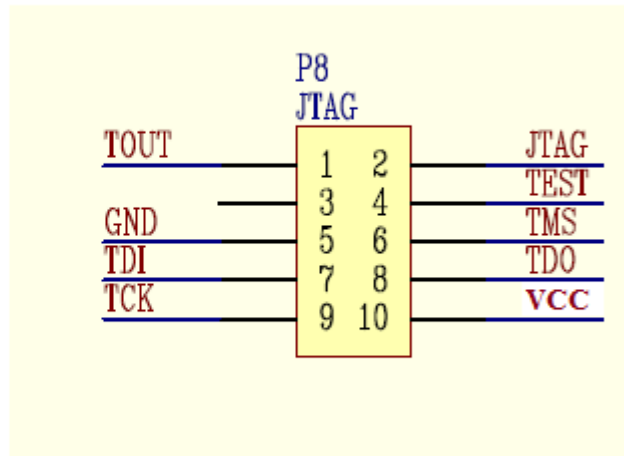
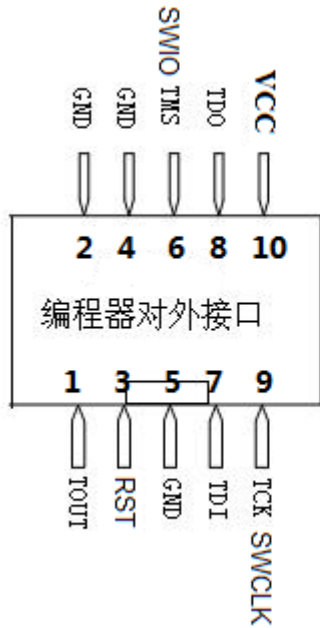
1	VCC	2	NC
3	TRST	4	GND
5	TDI	6	GND
7	TMS	8	GND
9	TCK	10	GND
11	RTCK	12	GND
13	TDO	14	GND
15	Reset	16	GND
17	NC	18	GND
19	NC	20	GND

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