

炬泉 MCU HT6X1X 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. 删除 RTCIF 清 0 的 bug, 后续新版本已经修正该问题 2. 增加 VCC 与 VRTC 管脚设计问题 3. 增加 startup.s 的配置说明 4. 红外占空比问题后续版本已修改, 该项删除 5. 修改 RTC 定时器定时的描述
V1.4	2016.3.3	1. 不推荐使用 PLL 停振检测功能 2. 增加 OSC 停振后 NMI 处理推荐
V1.5	2016.4.21	1.增加 test 管脚设计推荐
V1.6	2016.5.9	1, 删除原 2.11 脉冲口配置; 2, 2.11uart 使用建议增加初始化顺序; TXEN 使用说明增加 H 版改进说明; 3, 增加 2.14 INT 开启数字滤波; 4, 增加 2.15 仿真口 TDO 配置; 5, 2.16PLL 停振检测增加 OSC 冷启动停振检测判断; 6, 3.1 增加 VCC 有电必需确保 VRTC 同时有电; 7, 增加 3.4 TPS 内部框图
V1.7	2016-08-18	1, 增加 4.2 仿真接口增加 RST 2, 增加 2.17 UART\7816 SREL 不可设置为 0 3, 修改 2.5 osc 冷启动条件的描述, 增加 POR 和 LBOR 描述 4, 增加 2.18 PMU 电源检测模块 (VCC_DET、LVDIN_DET 和 BOR_DET 模块) 配置
V1.8	2016-09-05	1, 增加 2.19 开启 HRC 增加延时 2, 增加 2.20 PWM 运行过程中不要修改初始电平位 3, 增加 4.3 仿真口软件配置 4, 修改 3.2 VRTC 描述, 增加 RTC 模块工作电压描述 5, 修改 2.11 低功耗下 TPS 应用 6, Toff 对温度测量的误差影响 7, 修改 2.13 外部数字滤波应用 8, 增加 2.22 HRC 初始化 9, 增加 2.23 关闭 WDT 和恢复 WDT 10, 增加 2.24 TOUT 在低功耗下输出 11, VDD 更名为 VDD1P5, ADCBAT 更名为 ADCBAT

注: 本文档适用于 HT6015/HT6017/HT6019/HT6215

目 录

1	Information Block.....	4
1.1	Information Block 的操作说明.....	4
1.2	RTC 补偿系数寄存器和 Info Block 对应关系.....	4
2	软件.....	5
2.1	RTC 补偿参数.....	5
2.2	RTC 定时器定时.....	8
2.3	CLKCTRL0 配置.....	8
2.4	RTC 时间修改.....	8
2.5	系统时钟切换.....	8
2.6	在 HardFault 中不能加入软复位或者其他的强制复位的语句.....	8
2.7	关于 LCD 配置建议.....	9
2.8	系统时钟配置.....	9
2.9	ADC 检测功能.....	9
2.10	uart 使用.....	9
2.11	TPS 低功耗模式处理.....	10
2.12	startup.s 文件中 Flash 地址 FC0 处推荐配置.....	10
2.13	使用外部中断和 RX 中断的注意事项.....	10
2.14	仿真接口 TDO 的配置.....	10
2.15	建议不使用 PLL 停振检测功能.....	10
2.16	OSC 停振之后 NMI 中断处理推荐.....	11
2.17	UART\7816 SREL 不可设置为 0.....	11
2.18	PMU 电源检测模块 (VCC_DET、LVDIN_DET 和 BOR_DET 模块) 配置.....	11
2.19	开启 HRC 应适当增加延时.....	12
2.20	PWM 运行过程中, 应避免修改初始电平位.....	12
2.21	Toff 对温度测量的误差影响.....	12
2.22	HRC 初始化.....	13
2.23	关闭 WDT 和恢复.....	13
2.24	低功耗模式秒脉冲 TOUT 输出.....	13
3	硬件.....	14
3.1	外部 32K 晶振应用说明.....	14
3.2	VCC 与 VRTC 管脚.....	14
3.3	VDD1P5 管脚 (HT6015 的 51pin/HT6017 的 39pin/HT6019 的 31pin).....	14
3.4	Test 管脚.....	14
3.5	TPS 内部框图.....	15
4	可靠性问题.....	16
4.1	仿真接口设计.....	16
4.2	仿真接口增加 RST 建议.....	17
4.3	仿真口的软件配置.....	18

1 Information Block

512Bytes Information Block 地址位于 0x00040000~0x00040200 的 reserve 区域。

1.1 Information Block 的操作说明

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 512bytes (0x00040000~0x000401FF)，分 2 页，256bytes/页，其中第 2 页 (0x00040100~0x000401FF) 存储有芯片出厂信息，请勿进行写/擦除操作。

INFOLOCK (InfoBlock 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]	Information Block 锁定控制 对该寄存器写入 0xF998 后，Information Block 被解锁，用户可以写操作 Information Block。 写入非 0xF998 数据后，Information Block 被锁定，用户禁止写操作 Information Block。 默认为锁定状态，Information Block 不可执行写/页擦除/全擦除 操作 用户写入的是 0xF998, 读出值为 1; 写入的是非 0xF998, 读出值为 0

1.2 RTC 补偿系数寄存器和 Info Block 对应关系

RTC 的补偿系数寄存器在芯片 POR 上电后会自动装载，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 自动装载调整值	HRCADJ	0x140
----	HRCADJ 用户装载调整值补码	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 block 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

4. “HRCADJ 用户装载调整值”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 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 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，用户可以设置 HRCADJ 的寄存器值为 0x2A,那么不同芯片的 HRC 的处值差异在 14M+-3%以内。information block 的 0x140 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 的初值在控制在+-1%的精度以内。

2 软件

2.1 RTC 补偿参数

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：

HT601X(外置晶振)：

```
static const uint16_t TAB_DFx[10] =
```

```
{
```

```
    0x0000, 0x0000,
```

```
    0x007F, 0xDC38,
```

```
    0x007E, 0xDBF3,
```

```
    0x0000, 0x4C66,
```

```
    0x0000, 0x0384
```

```
};
```

HT6215 (内置晶振)：

```
static const INT16U TAB_DFx[10] =
```

```
{
```

```
    0x0000, 0x0705,
```

```
    0x007F, 0xE80D,
```

```
    0x007F, 0x060C,
```

```
    0x0000, 0x7262,
```

```
    0x0000, 0x1E26
```

```
};
```

```
#define C_Toff      0x0000
```

```
//温度偏置寄存器
```

```
#define C_MCON01   0x6077
```

```
//控制系数 01
```

```
#define C_MCON23   0x2388
```

```
//控制系数 23
```

```
#define C_MCON45   0x2688
```

```
//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；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 VCC 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 300ms 的时间，主要是由于 OSC 晶振起振到正常的频率值需要 300ms 的时间。

POR 复位：VCC<=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, 1.9V)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 关于 LCD 配置建议

- 我们系统工作电压时 2.2V—5.5V，电池供电在 3.6V。为了在正常模式和电池模式下均能有较好的显示，推荐选择 3V 液晶。
- 通过配置 LCD 充电模式和对亮度适应不同电压状态工作

我们提供的 demo 使用的是 3.1V 的 LCD。相关配置如下，用户可以参考

5V 上电初始化全显，对比度需大点：

```
LCDCLK = 0x008C;           //1/3 偏压,6COM,帧扫描频率为 85.3Hz
LCDCR = 0x0007;           //大电流模式，对比度=71.37%VLCD
```

5V 上电轮显，对比度可稍调小：

```
LCDCLK = 0x008C;           //1/3 偏压,6COM,帧扫描频率为 85.3Hz
LCDCR = 0x000A;           ///大电流模式，对比度=62.47%VLCD
```

3.3V Hold/sleep 配置，选择快速充电模式以降低功耗，适当增加对比度以获得良好的显示效果：

```
LCDCLK = 0x008C;           //1/3 偏压,6COM,帧扫描频率为 85.3Hz
LCDCR = 0x0032;           //快速流模式，对比度=88.13%VLCD
```

2.8 系统时钟配置

PLL 最高时钟频率为 22M，用户使用时不推荐使用 22M 时钟，有可能会导致 EMC 性能不太好，建议使用 11M 时钟或者更低频率。

2.9 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。

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

2.10 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（H 版已更新为 TXEN 一直有效，写 TXEN=0 无效）

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

2.11 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。如果想在低功耗模式下开启 TPS 并获得最低功耗，可配置采用最大周期 32s 打开一次。

TBS 模块按此设定的采样周期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。

2.12 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：

```
CRP_Key      IF      :LMOT::DEF:NO_CRP
              AREA   |.ARM._at_0x0FC0|, CODE, READONLY
              DCD    0xFFFFF09
              ENDIF
```

芯片	推荐值	推荐值说明
HT601X	XXXXFF09	Flash 不加密，自动装载屏蔽（目前可能很多代码用的这个配置）
	XXXXFFE0	Flash 不加密，自动装载屏蔽。 这种配置 ESD 效果可能会比推荐配置 1 更好些

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 JTAG 口读出），需将红色 FF 变成非 FF 即可。比如：FFFF00E0

2.13 使用外部中断和 RX 中断的注意事项

使用 INT 外部中断或者 RX 引脚功能的时候，必须要将 PINFLT 寄存器中相应的引脚数字滤波功能打开，即 INTFLT/RXFLT。这个外部引脚数字功能仅对配置的 RX 或 INT 有效，对相应 PIN 脚的其他功能（GPIO 或其它复用功能）无效。

2.14 仿真接口 TDO 的配置

应注意 PB[12]/TDO 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。TDO 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。

2.15 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。

2.16 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。

2.17 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。

2.18 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。

2.19 开启 HRC 应适当增加延时

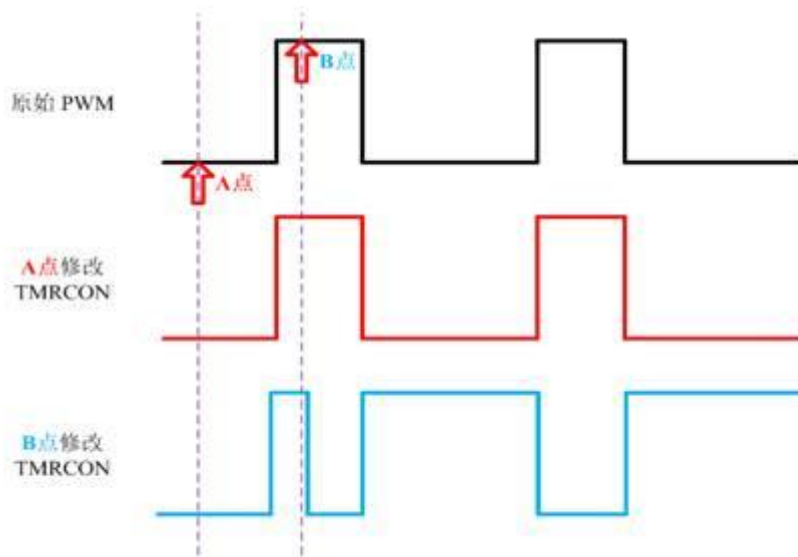
内部 HRC 时钟频率将在 H 版修改为 11.01MHz（Info block 中标志存储于 0x00040184，标志值为 0x0000C33C）。可以选择该高频 RC 时钟的分频输出时钟（分频设置位为 HRC_DIV[1:0]）作为系统时钟（SYSCLK_SEL[2:0]=010）。系统复位后，系统时钟默认选择 HRC。若关闭了 HRC，再次开启时，为确保 HRC 开启运行的稳定可靠，建议增加 1ms 左右延时，配置举例如下：

```

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
    
```

2.20 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。

2.21 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：

温度 $T_r = 12.9852 - \text{TMPDAT} * 0.0028$ 其中： T_r 为实际的温度（℃）

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

温度 $T_r = 12.9852 - (\text{TMPDAT} - \text{Toff}) * 0.0028$ 其中： T_r 为实际的温度（℃）

2.22 HRC 初始化

从 H 版开始, HRC 的频率从原来的 14MHz 调整到 11MHz。

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 11.01M+-1% (H 版, 常温), 11.01M+-3% (H 版, 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 11M+-3% (H 版 常温) 以内, 经过加载 information block 的校正值可将 HRC 频率调整到 11M+-1% (常温)。

2.23 关闭 WDT 和恢复

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。

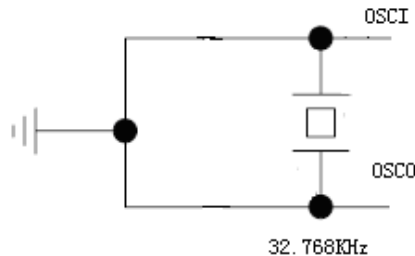
2.24 低功耗模式秒脉冲 TOUT 输出

系统进入 hold 模式之后, 如果 PLL_En 打开, TOUT 无输出; 如果关闭 PLL_En, TOUT 可正常输出未经高频补偿的分频信号。

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 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。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，极限情况下偏差 360s/d。

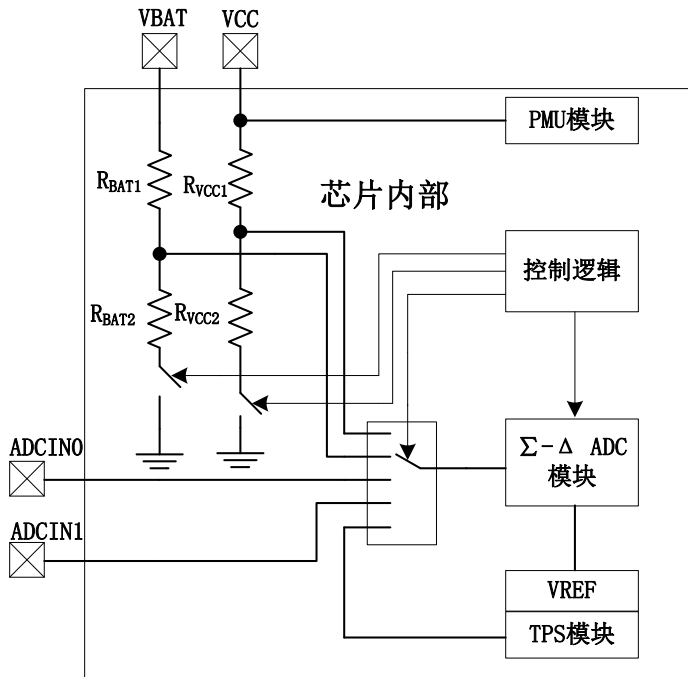
3.3 VDD1P5 管脚 (HT6015 的 51pin/HT6017 的 39pin/HT6019 的 31pin)

VDD 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 0-200 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 200 欧姆。

3.4 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。

3.5 TPS 内部框图

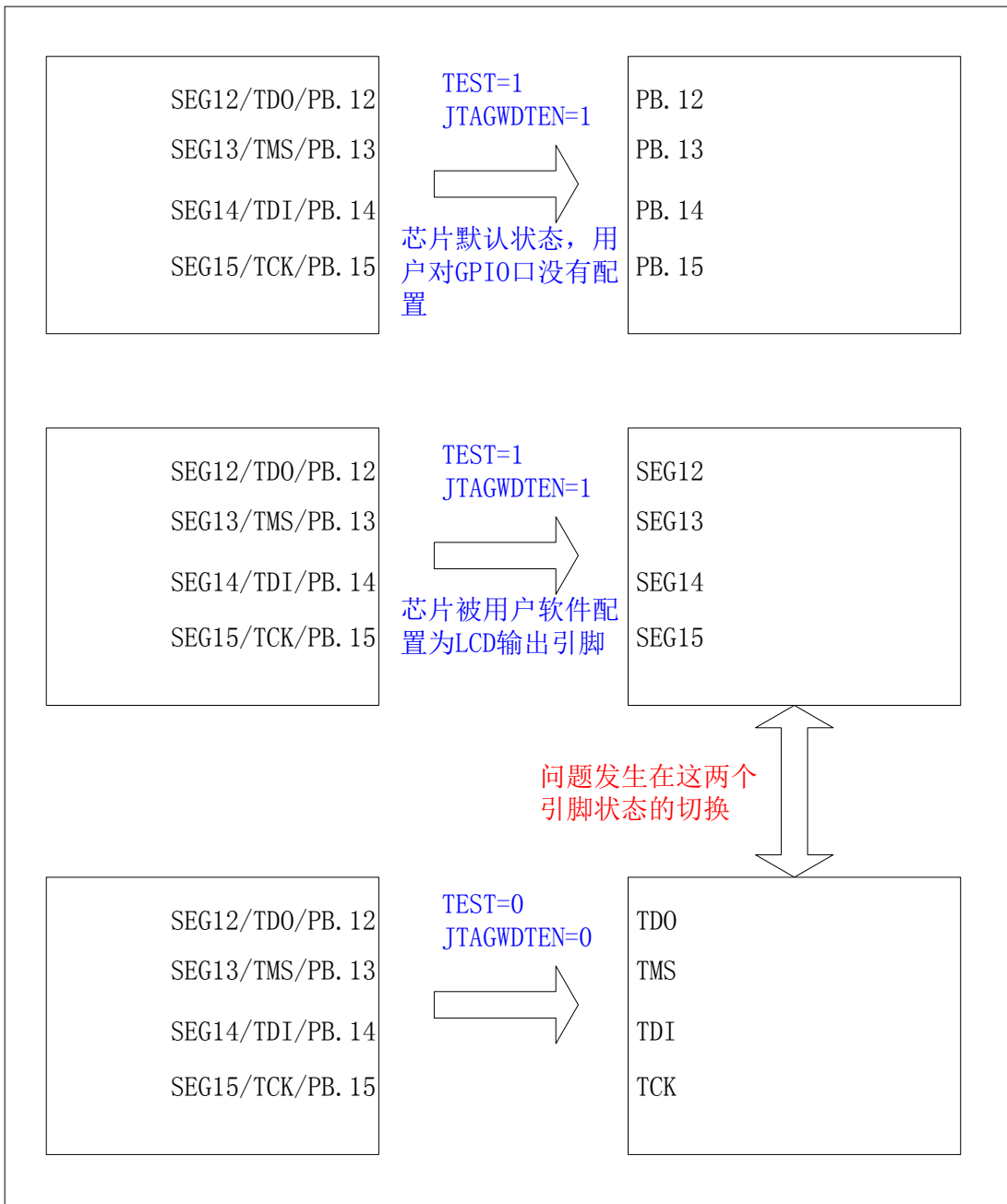


内部分压电阻阻值

Chip Num	ADCBAT 通道				VCC 通道			
	R_{BAT1}	R_{BAT2}	$R_{BAT1}:R_{BAT2}$	$R_{BAT1}+R_{BAT2}$	R_{VCC1}	R_{VCC2}	$R_{VCC1}:R_{VCC2}$	$R_{VCC1}+R_{VCC2}$
HT6X1X	9.6K	2.4K	4:1	12K	36K	6K	6:1	42K

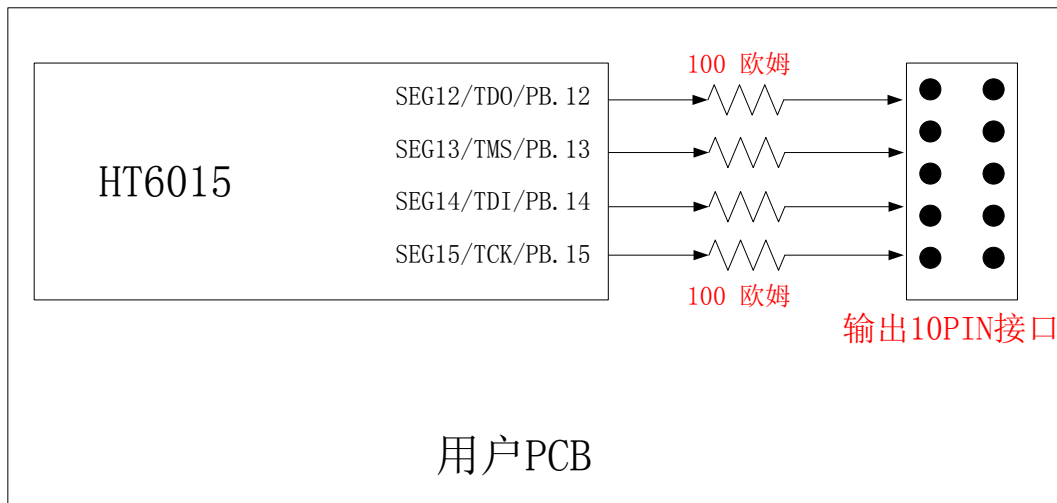
4 可靠性问题

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 4 个 GPIO 接口与客户 PCB 的 TMS, TDI, TDO, TCK 调试接口相连。
- (2) 在客户的 PCB 板上, 从芯片的引脚 TMS, TDO, TDI, TCK 引出到调试的外部插针的位置, 串入一个 100 欧的电阻, 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。

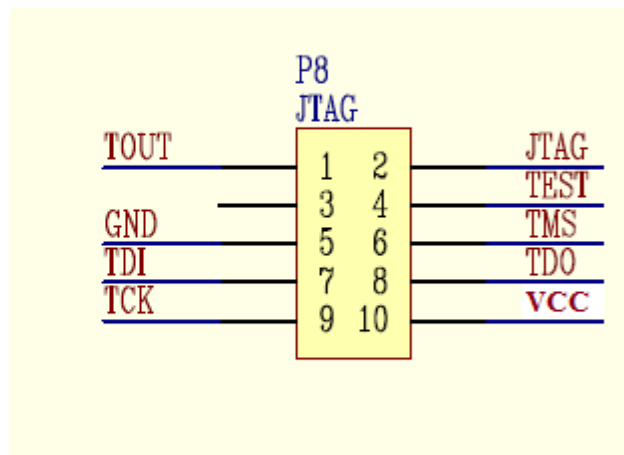
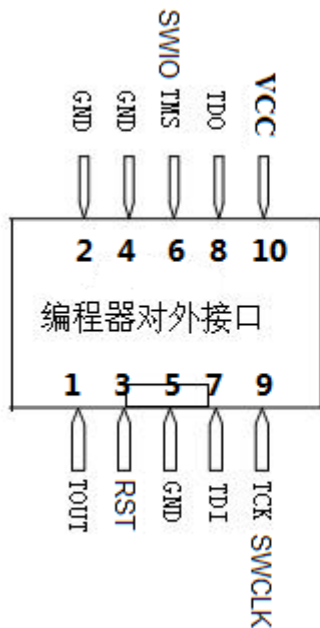
General Ulink Port

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 Reset

4.3 仿真口的软件配置

当 TEST 和 JTAGWDTEN 同时拉低时，系统进入仿真调试模式，TMS\TDO\TDI\TCK 这 4 个 pin 将作为仿真口线。开漏配置在 JTAG 模式下依然有效，软件配置建议不要将仿真口线配置为 OD 模式，否则会造成只能第一次下载成功，第二次就无法下载。仿真口线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。