

炬泉 MCU HT602X_FAQ

V2.32

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

		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 ADCBAT 8, 修改 3.3 提高 ESD 性能的方法 9, 更改仿真接口 ESD 防护
V1.91	2016-09-18	1, 增加 2.27 GPIO 输入电平判断 2, 增加 3.10: 5V tolerance IO 输入下拉的设计
V1.92	2016-12-23	1. 增加系统运行在 22M 时钟的应用建议; 2. 因为当前版本多功能编程器 (硬件版本: V0.3) 未加驱动芯片, 为保证正常下载程序, SWCLK (TCK) 建议不接 1000pF 电容。 3. 3.3.6 与后面可靠性描述内容重复, 删除 3.3.6 内容。
V2.0	2017-2-7	1. 增加 22M 打开 prefetch 下操作 flash 的应用建议; 2. 增加关于外部中断口的应用。
V2.1	2017-08-07	1. 增加 TBS 应用的建议和说明。
V2.2	2017-09-25	1. 增加 2.31 TBS 时钟选择推荐
V2.3	2018-10-23	1. 增加新旧版本 HT602X 的说明; 2. 修改 2.3 章节和 3.1 章节; 3. 删除 HT6027B 封装及说明; 4. 修改 4.4 Test 管脚小节, 增加风险说明及规避方法; 5. 修改 3.24 UART6 应用的推荐配置, 其他章节修改措辞及部分图。
V2.32	2019-01-30	1. 修改 4.2 中第 3 和第 4 条的内容, VDD1P5 外部串接电阻统一修改为 100ohm; 删除 SWIO 增加 1000pf 的电容的说明; 2. 增加两个误进测试模式的规避说明, 详见 3.32 和 3.33; 3. 增加 4.9 章节。

注: (1) 本文档适用于新旧版本的 HT6023/HT6025/HT6025B/HT6027/HT6027B/HT6029B

(2) HT602X 新旧版本差异比对详见 “新旧版本 HT602X 差异表”。

目 录

1	新旧版本 HT602X.....	6
2	Information Block.....	6
2.1	概况	6
2.2	Information Block 的操作说明.....	6
2.3	RTC 补偿系数寄存器和 Information Block 对应关系	7
3	软件	9
3.1	RTC 补偿参数.....	9
3.2	RTC 定时器定时	12
3.3	CLKCTRL0 配置	12
3.4	RTC 时间修改.....	12
3.5	系统时钟切换	13
3.6	在 HardFault 中不能加入软复位或者其他的强制复位的语句	13
3.7	系统时钟配置	13
3.8	ADC 检测功能.....	13
3.9	UART 使用.....	13
3.10	TMPEn 低功耗模式处理.....	13
3.11	startup.s 文件中 Flash 地址 FC0 处推荐配置.....	14
3.12	使用 INT 中断和 RX 中断的注意事项	14
3.13	仿真接口 SWIO 的配置	14
3.14	停振检测使用的建议	14
3.15	UART\7816 模块初始化配置.....	15
3.16	HRCADJ 初始化.....	15
3.17	低功耗模式 TOUT 输出.....	16
3.18	Toff 对温度测量的误差影响.....	16
3.19	关闭/开启 LRC 的操作.....	16
3.20	睡眠模式下, RTC 自动补偿的配置.....	18
3.21	外部 OSC 停振的系统状态.....	18
3.22	第二套 RTC 的应用.....	18
3.23	PMU 电源检测模块 (VCC_DET、LVDIN_DET 和 BOR_DET 模块) 配置	18
3.24	UART6 的推荐应用配置.....	18
3.25	重新开启 HRC 的配置	18
3.26	修改 PWM 初始电平位的推荐操作	19
3.27	GPIO 输入电平判断	19
3.28	Fcpu 主频工作在 22MHz 的建议	19
3.29	关于外部中断口的应用	21
3.30	TBS 模块应用建议和说明	21
3.31	新版本 HT602X TBS 时钟源的配置方法	22
3.32	模式切换导致 CLKOUT 无法正常输出的规避说明	22
3.33	误进 TDC 测试模式导致 PD.4/5/6 输出异常的规避说明	23
4	硬件	24
4.1	外部 32K 晶振应用说明	24
4.2	提升 HT6X2X 方案的 ESD 性能的方法.....	24

4.3	Test 管脚.....	25
4.4	VRTC 的使用.....	25
4.5	芯片版本识别方法.....	26
4.6	GPIO 带载能力描述.....	26
4.7	静电笔高频 ESD 防护规避.....	26
4.8	5V tolerance IO 输入下拉的设计.....	27
4.9	当 MCU 工作电压低于 5V 时，非 5V tolerance IO 外部上拉到 5V 影响 LCD 显示的规避方法	28
5	可靠性问题.....	29
5.1	仿真接口设计.....	29
5.2	IAR 环境下，仿真接口增加 RST 建议.....	30

1 新旧版本 HT602X

为了优化 EMC 等相关性能，新增了新版本的 HT602X 系列产品，新旧版本 HT602X 管脚 pin to pin 完全兼容，但新增了部分功能和寄存器，导致用户固件不能完全兼容，详细见新旧版本 HT602X 差异表说明。

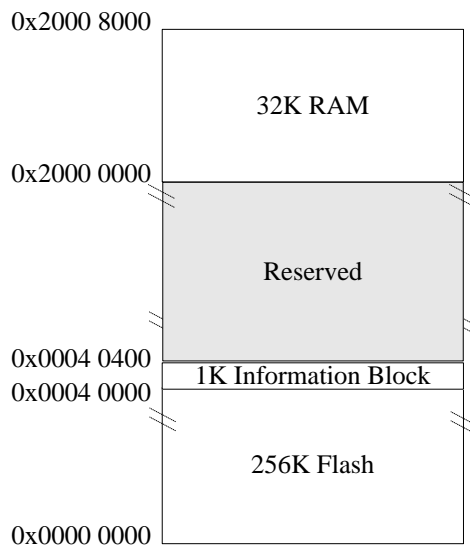
2 Information Block

2.1 概况

HT602X 内部有 1K bytes 的 Information Block 空间，用于存储出厂信息，地址位于 0x40000~0x403FF，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：



2.2 Information Block 的操作说明

将 Flash 的 00000FC1H 地址写入非 0FFH 的值后，256K Flash 被加密；Information Block 区不受影响，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。

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，用户若需要修改 information block 中的数据，应先把芯片出厂信息等数据备份再进行“擦—改—写”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]	Information Block 锁定控制 对该寄存器写入 0xF998 后，Information Block 被解锁，用户可以擦/写操作 Information Block。 写入非 0xF998 数据后，Information Block 被锁定，用户擦/写操作 Information Block 无效。 默认为锁定状态。用户写入的是 0xF998, 读出值为 1；写入的是非 0xF998, 读出值为 0。

2.3 RTC 补偿系数寄存器和 Information Block 对应关系

寄存器		Information block		
偏移地址 (基地址: 0x4000C000)	名称	偏移地址 (基地址: 0x00040000)	名称	说明和备注
0x50	DFAH	0x104	iDFAH	RTC 校正系数, 出厂值为 FF
0x54	DFAL	0x108	iDFAL	RTC 校正系数, 出厂值为 FF
0x58	DFBH	0x10C	iDFBH	RTC 校正系数, 出厂值为 FF
0x5C	DFBL	0x110	iDFBL	RTC 校正系数, 出厂值为 FF
0x60	DFCH	0x114	iDFCH	RTC 校正系数, 出厂值为 FF
0x64	DFCL	0x118	iDFCL	RTC 校正系数, 出厂值为 FF
0x68	DFDH	0x11C	iDFDH	RTC 校正系数, 出厂值为 FF
0x6C	DFDL	0x120	iDFDL	RTC 校正系数, 出厂值为 FF
0x70	DFEH	0x124	iDFEH	RTC 校正系数, 出厂值为 FF
0x74	DFEL	0x128	iDFEL	RTC 校正系数, 出厂值为 FF
0x78	TOFF	0x12C	iToff	TPS 温度传感器偏差校准 (出厂测试信息)

0x7C	MCON01	0x130	iMCON01	RTC 控制字 01, 出厂值为 FF
0x80	MCON23	0x134	iMCON23	RTC 控制字 23, 出厂值为 FF
0x84	MCON45	0x138	iMCON45	RTC 控制字 45, 出厂值为 FF
		0x13c	iChecksum	校验和, 出厂值为 FF
		0x140		内部高频 RC 时钟校准 HRCADJ (出厂测试信息)
		0x144		内部高频 RC 时钟校准 HRCADJ 补 码 (出厂测试信息)
		0x15c		TPS code 值 (出厂测试信息)
		0x15e		TMP275 的值 (出厂测试信息)
		0x160		芯片测试信息: 年/月/日/时 (出厂测试信息)
		0x164		芯片测试信息: 分/秒/siteNum (出厂测试信息)
		0x188	CHIP INFO	芯片版本号 (芯片信息)
		0x18C	~CHIP INFO	芯片版本号补码 (芯片信息)

注:

1. 由于不能确定客户实际应用中 32.768KHz 晶体的具体型号, 所以芯片出厂测试时 information block 中 RTC 校正系数 iDFAH ~ iDFEL (0x104~0x128) 均为 “FF”。用户进行 RTC 系数校正时, 需要计算校验和并更新 information block 中的相应数据。实际应用中请参考本 FAQ 的 3.1 章节。
2. 芯片出厂测试时 Information block 中的 RTC 控制字 iMCON01 ~ iMCON45 (0x130~0x138) 均为 “FF”。实际应用中 MCONxx 寄存器需要保持默认值 (MCON01=0x2000, MCON23=0x0588, MCON45=0x4488), 用户进行 RTC 系数校正时, 需要把 MCONxx 的默认值写入 information block 的相应地址中。
3. HT602X 芯片(除 HT6029B 外)在出厂测试中会进行 TPS 温度传感器偏差 TOFF 校准, 包括: iToff(0x12c), TPS code (0x15c) 以及 TMP275 (0x15e)。因 HT6029B 出厂时未做 TOFF 校准, 故用户需通过钜泉工具自行校准。
4. HT602X 芯片在出厂测试中会进行内部高频 RC 时钟的校准, 校准数据存储在 0x140 和 0x144 (0x144 是 0x140 的补码), 客户应用时不应更改或擦除。为了获得更高精度的 HRC 频率, 可将 information block 中内部高频 RC 时钟校准值 (0x140 和 0x144) 读出, 判断无误后, 把该值 (0x140) 写入 HRCADJ 寄存器。校准后, 可确保常温下芯片的 HRC 时钟频率在 11.01MHz±1.2%范围以内。
5. HT602X 芯片在出厂测试中会在 information block 中记录相关测试信息, 包括测试日期、测试时间和 siteNum (0x160 和 0x164), 客户应用时不应更改或擦除。
6. HT602X 芯片在出厂测试中会在 information block 中记录芯片信息, 包括芯片版本, 管脚等信息 (0x188 和 0x18C), 方便软件进行区分新旧版本 HT602X, 详细说明可见新旧版本 HT602X 差异表。
7. information block 中偏移地址 0x13c 处存放的是从 0x104 到 0x138 的 32 位无符号数累加和, 如果累加和比对正确, 则说明 info 中数据 (从 0x104 到 0x138) 有效; 否则有两种可能: (1) info 中数据 (从 0x104 到 0x138) 错误, (2) 芯片未进行 RTC 系数校正。详细请参考本 FAQ 的 3.1 章节。
8. HT602X 的 Flash 控制字默认为 0xFFFFFA2, 自动装载功能不使能。
9. 由于 information block 的特殊存储结构, 不能做单字节擦写操作, 只能采用页擦除的方式, 用户若需修改 information block 中的数据, 应先进行数据备份再进行“擦—改—写”操作。

3 软件

3.1 RTC 补偿参数

为了 RTC 温补功能正常运行，应确保 information block 中的校正系数、RTC 控制字、iToff 等参数被装载到对应寄存器中，用户程序需要在软件中进行如下操作：

1. 计算校验和：checksum = iDFAH+iDFAL+...+iMC0N23+iMC0N45; (iDFAH...iMC0N45 为 information block 中的数据)，并与 information block 中存储的校验和 iChecksum 对比。

2. 当校验和正确时（即 iChecksum 与 checksum 相等），从 information block 中加载信息 (iDFAH~iMCON45) 到 RTC 模块对应的寄存器中。

3. 当校验和不正确时（即 iChecksum 与 checksum 不相等），则向 RTC 模块对应的寄存器中写入相应的默认值（详细操作和默认值参考如下代码）；

//定义部分-----

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)
```

推荐的默认 RTC 校正系数如下:

HT602X(旧版本):

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

```
{
```

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

```
    0x007F, 0xD64C,
```

```
    0x007E, 0xD708,
```

```
    0x0000, 0x546E,
```

```
    0x0000, 0x04B0,
```

//推荐的 RTC 校正系数针对精工 VT200-F 晶体

```
};
```

HT602X(新版本):

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

```
{
```

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

```
    0x007F, 0xDA4B,
```

```
    0x007E, 0xD9AC,
```

```
    0x0000 0x4A2E,
```

```
    0x007F, 0xFC90,
```

//推荐的 RTC 校正系数针对精工 VT200-F 晶体

```
};
```

```
#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

```
//code 部分-----
```

```
uint32_t checksum;
```

```
INT16S toff, temp, code;
```

```
for(i=0;checksum=0;i<14;i++)
```

```
{
```

```
    checksum +=HT_Info->DataArray[i];
```

```
}
```

```
if(checksum== HT_Info->DataArray[14]) // 校验和正确 (RTC 系数已校正)
{
    // information block 中的数据写入到对应的寄存器中

    HT_RTC->DFAH (地址: :0x4000C050) = *(uint32_t *)0x40104;
    HT_RTC->DFAL (地址: :0x4000C054) = *(uint32_t *)0x40108;
    HT_RTC->DFBH (地址: 0x4000C058) = *(uint32_t *)0x4010C;
    HT_RTC->DFBL (地址: 0x4000C05C) = *(uint32_t *)0x40110;
    HT_RTC->DFCH (地址: 0x4000C060) = *(uint32_t *)0x40114;
    HT_RTC->DFCL (地址: 0x4000C064) = *(uint32_t *)0x40118;
    HT_RTC->DFDH (地址: 0x4000C068) = *(uint32_t *)0x4011C;
    HT_RTC->DFDL (地址: 0x4000C06C) = *(uint32_t *)0x40120;
    HT_RTC->DFEH (地址: 0x4000C070) = *(uint32_t *)0x40124;
    HT_RTC->DFEL (地址: 0x4000C074) = *(uint32_t *)0x40128;
    HT_RTC->TOFF (地址: 0x4000C078) = *(uint32_t *)0x4012C;
    HT_RTC->MCON01 (地址: 0x4000C07C) = *(uint32_t *)0x40130;
    HT_RTC->MCON23 (地址: 0x4000C080) = *(uint32_t *)0x40134;
    HT_RTC->MCON45 (地址: 0x4000C084) = *(uint32_t *)0x40138;

}

else // 校验和错误 (RTC 系数未校正; information block 中数据错误)
{
    //RTC 校正系数的默认值 TAB_DFx 写到相应的寄存器中
    HT_RTC->DFAH= TAB_DFx[0];           //地址同上
    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];
    //组合判断 information block 中的 iToff 值是否合理
    toff = *(INT16S*)0x4012C;           //iToff 值
```

```
code = *(INT16S*)0x4015C;           //TPS code 值
temp = *(INT16S*)0x4015E;           //TMP275 的温度值
if (((toff < -3000) || (toff > 3000)) //itoff 判断范围
    || (temp < 2000) || (temp > 3000) //TMP275 温度判断范围
    || (code < -7000) || (code > -1000)) //TPS code 判断范围
{
    HT_RTC->TOFF = C_Toff;           //判断错误，TOFF 用 默认值 C_Toff (0x00) 装载
    //理论上，已校正 TOFF 后的芯片不会出错，如果异常请用
    //炬泉工具重新校正 TOFF。
}
else
{
    HT_RTC->TOFF = toff;             //判断正确，TOFF 用 information block 中的值装载
}

//加载 RTC 控制字的默认系数
HT_RTC->MCON01=C_MCON01;
HT_RTC->MCON23=C_MCON23;
HT_RTC->MCON45=C_MCON45;
}
```

3.2 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)

3.3 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 版及后续版本更改为只有小功耗模式，此位写 0 无效)；

CLKCTRL0 的 1P5LB0R_EN 为内部电源控制位，保持默认配置为 1 即可，建议不要修改。

3.4 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，芯片会先判断 31 日对于 2 月是非法数据，就会导致修改时间不成功。

3.5 系统时钟切换

芯片在发生上电复位冷启动时，是从 HRC 开始启动，如果需要切换到 OSC 时钟或者 PLL 时钟运行，需要等待 500ms，主要是由于 OSC 晶振起振到正常的频率值需要 500ms。

3.6 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，导致芯片处于反复复位状态，不能恢复，仿真接口也无法连接芯片。

3.7 系统时钟配置

PLL 最高时钟频率为 44MHz，若 fcpu=44MHz，需使能指令预取功能。

介于 EMC 性能考虑，推荐用户使用 **fcpu = 22MHz 时钟**，或者更低频率。

3.8 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。

3.9 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 CLKCTRL1 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：

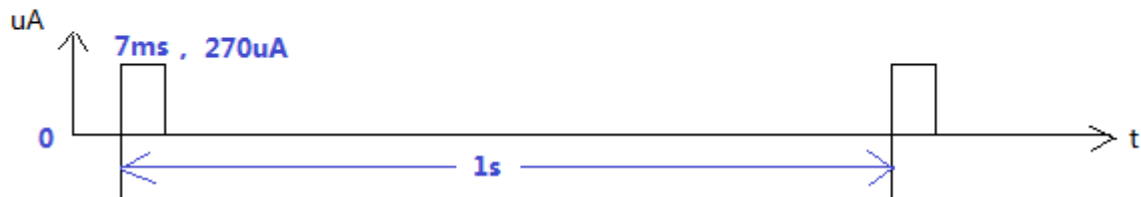
- A) 发生复位（比如软复位/WDT 复位等）
- B) 使能 TXEN 和 CLKCTRL1 寄存器中 UART 时钟使能位，让未发送完的数据发送完成即可

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

3.10 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，可通过 TBSPRD 寄存器（TMPPRD[2:0]位）配置更长的测量周期以降低功耗。用户可根据实际需求配置合理的测量周期。

TPS 分时1s开启一次的 功耗&开启时间 示意图



3.11 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	X7FFFFA2 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

3.12 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 版及后续版本已经修改为：内部强制使能 RX 和 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。

3.13 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，第二次下载时调低通信速度有可能能下载。

3.14 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 500ms 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 500ms 之后，再将 LF 停振检测打开（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, Fpll, Fhrc 具有停振检测功能。Flrc 默认为不会发生停振（设计保证），作为其他 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 Fpll（Flf 为 Fpll 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 Fpll 时，不会切换时钟，也不会发生 NMI 中断。
4. 当 Fpll 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 Fpll 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 Fpll 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_FLAG，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。

3.15 UART7816 模块初始化配置

程序初始化 UART7816 模块时，波特率设置寄存器 SREL 不能设置为 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。

注：新版本 HT602X 无该问题。

3.16 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），必须加载 information block 中的调整值到此寄存器。

information block 的 0x00040140 地址存放的 HRCADJ 调整值是对 HRC 的频率做调整，目的是让不同的芯片的 HRC 的频率控制在 $11.01\text{MHz} \pm 1.2\%$ （常温）， $11.01\text{MHz} \pm 3\%$ （全温度范围 $-40^{\circ}\text{C} \sim 85^{\circ}\text{C}$ ）的精度以内。

“HRCADJ 的调整值”芯片不会自动装载。为了获得更高精度的 HRC 频率，可将 information block 中内部高频 RC 时钟调整值（0x140 和 0x144）读出，判断无误后，把该值（0x140）写入 HRCADJ 寄存器。

3.17 低功耗模式 TOUT 输出

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

3.18 Toff 对温度测量的误差影响

温度传感器初始偏置 Toff: 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) 对于一般的温度测量功能, 温度计算时可不调用 Toff, 直接使用手册上的温度转换公式:

温度 $T_r = 12.9852 - \text{TMPDAT} * 0.002828$ 其中: T_r 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 ($-40^{\circ}\text{C} \sim +85^{\circ}\text{C}$) 引起的误差在 2°C 附近。

(2) 如果用户需要提高温度测量精度, 需将 Toff 影响考虑在内, 建议使用如下公式:

温度 $T_r = 12.9852 - (\text{TMPDAT} - \text{Toff}) * 0.002828$ 其中: T_r 为实际的温度 ($^{\circ}\text{C}$)

3.19 关闭/开启 LRC 的操作

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_WDT->WDTCLR = 0xAA3E; //clear WDT
    HT_RTC->CTRLBYFLASH &= 0xFFFD;
    HT_RTC->LRCCOMAND = 0x5555;
    HT_RTC->LRCCOMAND = 0xAAAA;
}
```

关闭后再开启 LRC:

void open_wdt(void)

```
{
    HT_RTC->CTRLBYFLASH |= 0x02;
    HT_WDT->WDTCLR = 0xAA3E; //clear WDT
}
```

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
FC0H BIT[7: 5]	内部控制位, 需保持 101bit
RTCRST	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: 其他位请保持默认值。

LRCCOMMAND (LRC 写关闭命令寄存器)			基地址: 0x4000C000 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

3.20 睡眠模式下，RTC 自动补偿的配置

在睡眠模式下，保持温度采样(TBSCON.TMPEn=1)开启,并根据功耗需求配置温度传感器的采样周期(TBSPRD.TMPPRD[2:0])。芯片 RTC 温补模块硬件自动计算实时 RTC 补偿值,进行温补。

3.21 外部 OSC 停振的系统状态

Case1: 如果 Fosc 停振，若当前系统时钟不是 Fosc 或 Fpll（由 Fosc 倍频），则 CPU 不会停止；

Case2: 如果 Fosc 停振，若当前系统时钟选择了 Fosc 或 Fpll，同时开启了相应的停振检测，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；

Case3: 如果 Fosc 停振，若当前系统时钟选择 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。

3.22 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。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。

3.23 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 前，程序必须配置寄存器 VDETPCFG（0x4000F408）的 VDET_TIME[1:0]（bit4:3）=11B，

检测时间 VDET_TIME[1:0]，默认值为 00B, 336us

开启周期 VDET_PRD[2:0]，默认值为 010B, 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，这个滤波时间可能会变大，产生误检信号导致错误的复位，故用户程序必须配置 VDET_TIME[1:0]=11B，为最大检测时间 1068us。用户可根据功耗需求配置开启周期 VDET_PRD[2:0]。

3.24 UART6 的推荐应用配置

问题 1:

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

但 UART6 使用中，不能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。

问题 2:

在新版 HT602x 中，uart6 在时钟使能后，其他配置未配之前，如果 RX 上有上升沿，会产生 RX 中断标志，该中断标志无法清 0，直到下一个下降沿才能清 0。

规避方案：

方案 1：不要将 uart6 时钟使能和 uart6 配置分开，即 uart6 使能后，立刻配置波特率等。该规避方案出现问题的概率极低。

方案 2：先将 uart6 全部配置好，再将 GPIO 配置为 uart 功能。该规避方案可以完全规避。

3.25 重新开启 HRC 的配置

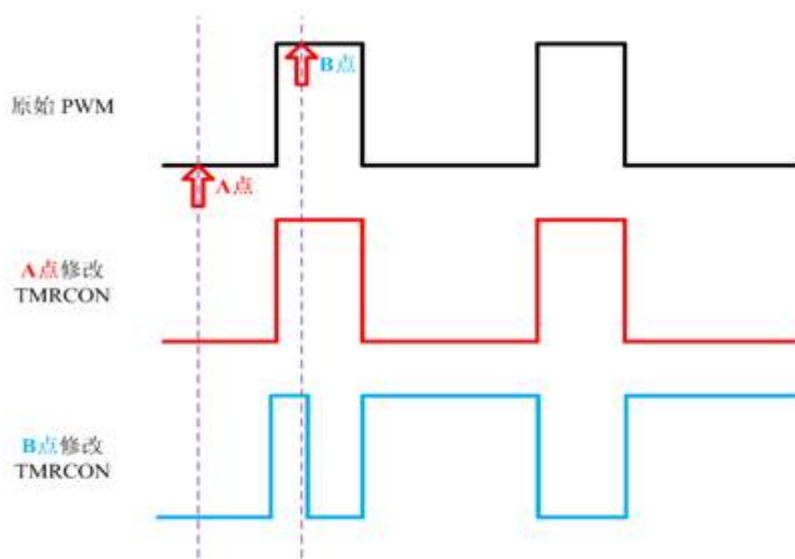
用户程序若关闭了 HRC，再次开启时，为确保 HRC 开启运行的稳定可靠，建议增加 1ms 左右延时，配置

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
```

3.26 修改 PWM 初始电平位的推荐操作

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：



规避操作方法如下：

需要首先关闭计数器使能位 TMRCON.CNTEN，再修改初始电平控制位 TMRCON.PWMHL (BIT6)，最后开启计数器使能。

3.27 GPIO 输入电平判断

对于一般的 GPIO，高电平输入 (VIH) 为 0.7VCC。

当 MCU 的系统电压 VCC=3V，外接 5V 器件，只能选用具有 5V tolerance 功能的 IO 作为输入端口，这些 GPIO 只有 8 个：PC4\5\6\7\9\10\13\14；

当 MCU 的系统电压 VCC=5V，外接 3V 器件，要想正确识别外部器件的高电平输入，只能选用具有 TTL 输入功能的 GPIO，其输入高电平最小值为 0.5VCC，HT6023 具有 TTL 输入功能的 GPIO 共有 17 个：PC4\5\6\7\8\9\10\13\14, PH0\1\2\3\4, PE0\3\6。

3.28 Fcpu 主频工作在 22MHz 的建议

用户如果 Fcpu 主频工作在 22MHz，为提高系统的可靠性，建议使能指令预取功能。使能后，实际性能约为 22MHz 的 80% 左右。

PREFETCH	基地址：0x4000F000
----------	----------------

(写保护) (指令预取使能控制寄存器)			偏移地址: 3CH					
	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	PreFetch_EN
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PreFetch_EN	指令预取使能控制位 =0: 禁止 =1: 使能

当Fcpu工作在22MHz且打开PREFECH时，若需要对内部flash进行写操作，可选择操作方法如下：

方法1：需要先将CPU时钟降到11M，关闭prefech再进行flash写操作，待flash操作完成之后先打开prefech，再切回22M；

方法2：在不降频时将写FLASH的底层C驱动函数HT_Flash_WordWrite(const uint32_t* pWriteWord, uint32_t Address, uint32_t Num)更改为汇编指令，具体函数如下：

注：该函数值仅 适用于IAR编译环境

```
void HT_Flash_WordWrite(const uint32_t* pWriteWord, uint32_t Address, uint32_t Num)
{
    asm(
        "PUSH      {R4-R6}\n"
        "MOVS      R5, #1\n"
        "LDR       R3, dat0\n"
        "LDR       R4, dat3\n"
        "LDR       R6, [R3]\n"
        "ANDS      R6, R6, R5\n"
        "LSLS      R6, R6, #2\n"
        "LDR       R4, [R4, R6]\n"
        "LDR       R6, dat1\n"
        "STR       R6, [R3]\n"
        "LDR       R6, dat2\n"
        "STR       R6, [R3, #0x38]\n"
        "STR       R5, [R3, #0x34]\n"
        "CMP       R2, #0\n"
        "BEQ       LOOP0\n"
        "MOVS      R5, #3\n"
        "BICS      R1, R1, R5\n"
        "LOOP3: LDR      R5, [R0]\n"
        "STR       R5, [R1]\n"
```

```

"LOOP1:LDR      R5, [R3, #0x34]\n"
  "LSLS      R5, R5, #29\n"
  "BMI      LOOP1\n"
  "ADDS      R0, R0, #4\n"
  "ADDS      R1, R1, #4\n"
  "SUBS      R2, R2, #1\n"
  "BNE      LOOP3\n"
"LOOP0:MOVS     R0, #0\n"
  "STR      R0, [R3, #0x34]\n"
  "STR      R0, [R3, #0x38]\n"
  "STR      R4, [R3]\n"
  "POP      {R4-R6}\n"
  "BX      LR\n"
"dat0:dc32      0x4000f000\n"
"dat3:dc32      0x0\n"
"dat1:dc32      0xA55A\n"
"dat2:dc32      0x7a68\n"
  );
}

```

3.29 关于外部中断口的应用

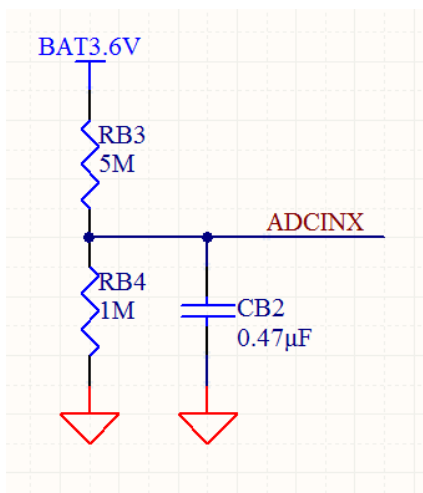
用户将 GPIO 口复用为外部中断口之后，如果需要再配回 GPIO 口的话，必须先关闭外部中断使能然后再配置为 GPIO 口，以免发生误触发。

3.30 TBS 模块应用建议和说明

HT602X 内部 ADC 会应用到内部温度，外部电池电压 电源等方面的测量。建议客户应用时注意如下：

1. 当 HT602X 内部 ADC 用来测量外部电池电压时，其满量程为 800mv，电表上一般采用 5M + 1M 电阻分压取样经过电容后接入芯片 ADC 管脚。

由于是弱源输入，ADC 过采样会造成 ADC 采样电压一定的跌落。建议外接 0.47uF 电容以避免采样电压的跌落。 原理图如下：



2. 新版本 HT602X 芯片已修改这一问题，但同时 TBS 相关的系数会与现有版本的不一样，为避免国网送样软件比对出现问题，客户必须把 TBS 相关系数:包括温度 TMPDAT、7 路 ADC（ADCBAT VCCDAT VADCIN0 VADCIN1 等）的计算系数 存入芯片 information block 中（除芯片出厂相关信息外的地址）。

3.31 新版本 HT602X TBS 时钟源的配置方法

新版本 HT602X 的 TBS 模块时钟可选择 OSC 或 PLL 时钟的 168 分频（TBSCON. Bit5）作为时钟源。用户在配置快速触发请求前，需选择工作时钟源为 PLL 时钟的 168 分频。

当 TBS 模块的时钟源选择为 PLL 的 168 分频时，如 PLL 时钟已被关闭，则 TBS 模块没有时钟源，标志寄存器和数据寄存器将维持原来的值，不会继续更新数据，中断也不会产生，TBS 模块无法正常工作。

推荐配置方法如下：

- 1) 若需要配置 TBS 模块的时钟源为 PLL 的 168 分频（TBSCON.bit5=1），先确保 PLL 时钟已稳定运行（PLL_EN=1，PLL 锁定已稳定）；
- 2) 若用户需关闭 PLL 时钟，应先确保 TBCLK_SEL 为 OSC 时钟（TBSCON.bit5=0）。

3.32 模式切换导致 CLKOUT 无法正常输出的规避说明

问题现象：芯片上电后，未正常进入 Normal 模式，而是进入测试模式，之后在未发生掉电复位的情况下进入 Normal 模式时，可能会发生 PE.3 即使配置为 CLKOUT 也无法正常输出时钟的情况。

问题原因：此条件下，芯片在进入 Normal 模式后，芯片 RTC 模块的 HFCFG 寄存器有概率初始值不为 0，导致芯片开启了自测频模式。自测频模式下，芯片的 PE.3 会强制为输入模式，CLKOUT 功能不受寄存器控制。

规避方案：芯片上电后将 RTC 模块的 HFCFG 寄存器清零。

```
#define HT_HFCFG (*(volatile unsigned int*)(0x4000C0F0))
```

系统上电初始化该寄存器：

```
HT_CMU->WPREG = 0xA55A; //关写保护
```

```
HT_HFCFG = 0x00; //初始化为 0
```

```
HT_CMU->WPREG = 0x0000; //开写保护
```

系统运行定时校验：

```
If((HT_HFCFG & 0x80) != 0)
```

```
{
```

```
    HT_CMU->WPREG = 0xA55A; //关写保护
```

```
    HT_HFCFG = 0x00; //初始化为 0
```

```
    HT_CMU->WPREG = 0x0000; //开写保护
```

```
}
```

注：HFCFG 受写保护寄存器影响，修改前需要关闭写保护。

3.33 误进 TDC 测试模式导致 PD.4/5/6 输出异常的规避说明

问题现象：PD4 信号口异常脉冲，2s 左右出现一个 1V 高脉冲，持续 8ms，擦除芯片 flash 程序后，现象依然存在。

问题原因：PD.4/5/6 是 TDC 测试模式的测试结果输出引脚。

将 TBS 模块的 TBSTEST 寄存器配置为 0x0220，即 tdc_test<1:0>控制位选择 01，可进入 TDC 测试模式。若同时将 TBS 模块的温度检测周期设置为 2s，可看到 PD.4 每 2s 输出一个持续 7ms、电平为 2.3V 的信号，同时 PD.5 和 PD.6 会输出低电平（正常不配置时 IO 呈高电平）。

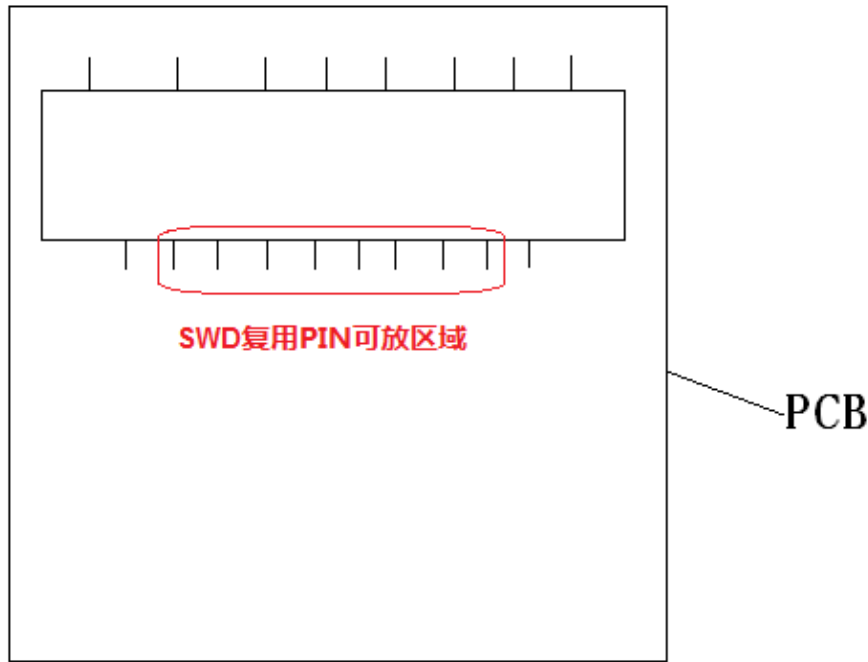
若管脚对地接 5k 电阻，输出电平将从 2.3V 下降至 1.36V。（VCC 供电 3.32V）

此现象只要 RTC 不掉电，会一直保持。

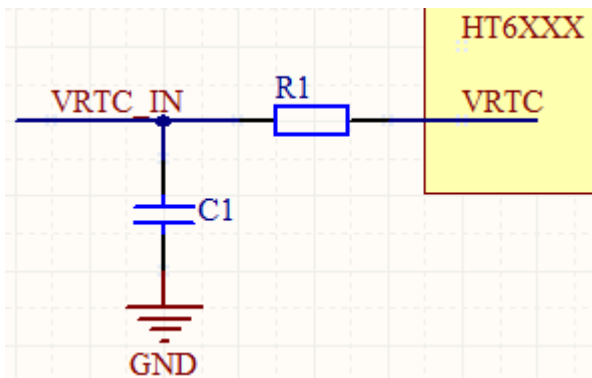
规避方案：上电后将寄存器 TBSTEST 赋一次默认值。

寄存器 TBSTEST 的地址与默认值请参考下表：

Chip	Address	Default
HT6x2x	0x4000E024	0x0200



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。



4.3 Test 管脚

问题：当芯片在上电时，TEST、PC9、PA5、PA10、PA6 管脚初始电平的配置满足下表条件时，芯片会进入某种测试模式，导致 RTC 时钟等寄存器出现异常值。

管脚名称	TEST	PA6	PC9	PA5	PA10
管脚电平	0	1	1	0	0

硬件规避方法：

1. TEST 管脚需外接 1K 上拉电阻（内部有约 88K 上拉电阻）和一个 0.01uF 的电容到地。
2. 硬件设计上确保 PC9 PA5 PA10 在上电过程中电平为非“100”。注：第 2 点在新版本 HT6X2X 已修改。

4.4 VRTC 的使用

建议在 VCC 有电的情况下，确保 VRTC 同时供电，否则会造成无法下载程序。

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。

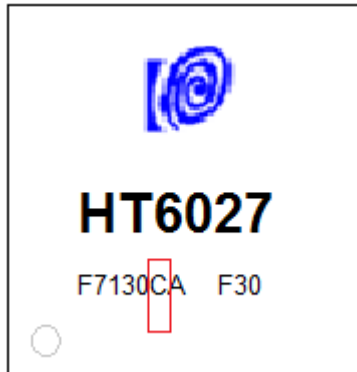
VCC 供 CPU 部分；VRTC 供 RTC 相关部分（包括 32.768KHz 晶体,TBS 温度采样模块，温补分频电

路);

VRTC 供电>2V 时，如果已开启了温度采样功能，那么自动温补功能正常。

VRTC 供电在 1V~2V 时，TBS 温度采样不准，温度补偿功能异常，一般情况下此时误差 30s/d，极限情况下偏差更大。

4.5 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。

4.6 GPIO 带载能力描述

HT6X2X 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

4.7 静电笔高频 ESD 防护规避

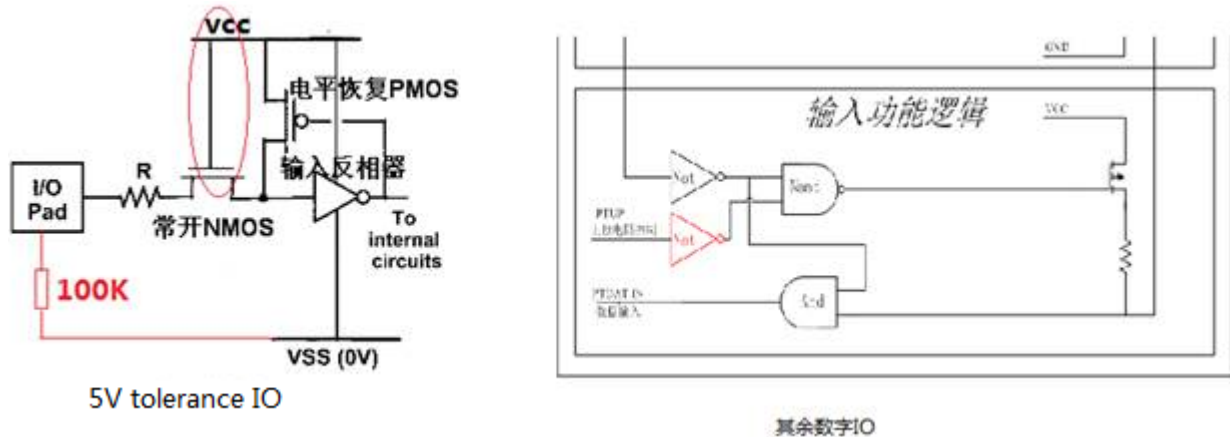
改进方式一：尽量降低 MCU 的系统工作电压，可有效减小 ESD 干扰信号的泄放量，而保护 MCU；

改进方式二：MCU 输入电源 VCC 和 VRTC 的高频滤波电容摆放位置远离 MCU 相应电源 PIN 脚，推荐距离电源 PIN 1cm 位置摆放，以防止静电笔对高频滤波电容进行干扰时导致 MCU 电源管脚损坏的情况发生。

4.8 5V tolerance IO 输入下拉的设计

具有 5V tolerance 功能的数字 IO 当配置为数字输入且上拉无效的情况下，通过 100K 欧姆的下拉电阻无法拉低，而其余 GPIO 可在相同配置下通过外加 100K 欧姆的下拉电阻获得低电平。原因如下：

GPIO 内部等效为如下的电路：



在 HT6X2X 系列芯片中，具有 5V tolerance 功能的 IO 电路如上图左边所示，内部常开 NMOS 上拉到 VCC（其余数字 IO 不存在这个无法关闭的上拉），即使配置数字输入上拉无效，这个上拉还是存在的，所以，必需在外部 PIN 脚上增加一个较强的下拉才能将 PIN 脚电平拉为低电平。对于此类 IO，如果要实现有效下拉，5V 电源电压下，下拉电阻不能大于 10K；3V 电源电压下不能大于 20K。相关 GPIO 列表：

100 PIN	80 PIN	64 PIN	标识	引脚类 型	第一复用 功能	第二复用 功能	引脚说明
					IOCFG=1 AFCFG=0	IOCFG=1 AFCFG=1	
38	28	24	PC.4	I/O	SPI0_MOSI	RX5	滤波 2us，输出驱动 5mA TTL 电平输入
39	29	25	PC.5	I/O	SPI0_MISO	TX5	TTL 电平输入
40	30	26	PC.6	I/O	SPI0_CLK	SPI0_CLK	TTL 电平输入
41	31	27	PC.7	I/O	SPI0_CS	SPI0_CS	TTL 电平输入
49	39	31	PC.9	I/O	TMR2	TMR2	TTL 电平输入
50	40	32	PC.10	I/O	TMR3	TMR3	TTL 电平输入
76	61	49	PC.13	I/O	SCL	SCL	TTL 电平输入
77	62	50	PC.14	I/O	SDA	SDA	TTL 电平输入

4.9 当 MCU 工作电压低于 5V 时，非 5V tolerance IO 外部上拉到 5V 影响 LCD 显示的规避方法

问题描述：

客户表上 MCU_VCC 工作电压小于 5V，非 5V tolerance IO 配置为 OD 输出高，外接载波模块为 5K 电阻上拉至 5V，导致 LCD 显示异常；

问题分析：

非 5V tolerance IO 在 OD 输出高时，外部上拉高于 MCU_VCC 电压（大于约 0.5V），会从外部反灌电流到 MCU 内部，且拉高 MCU 内部电压。在 LCD 对比度不变的情况下，会导致 LCD 驱动电压整体被抬高，从而 LCD 显示异常（产生“鬼影”）。

当外部上拉高于 MCU_VCC 电压小于 0.4V 时，内部保护二极管未导通，MCU_VCC 未被外部上拉拉高。但由于该非 5V tolerance IO 是 LCD 功能的复用管脚，外部上拉的 5V 经过上拉电阻影响 LCD 的每一档电压，但没有直接连接到 VCC，所以 VCC 电压并未被影响，而 VLCD 的电压由于上拉的存在，会被拉高。（上拉电阻越小，对 LCD 的影响越大；5V 至 VCC 的压差越大，对 LCD 影响也越大）

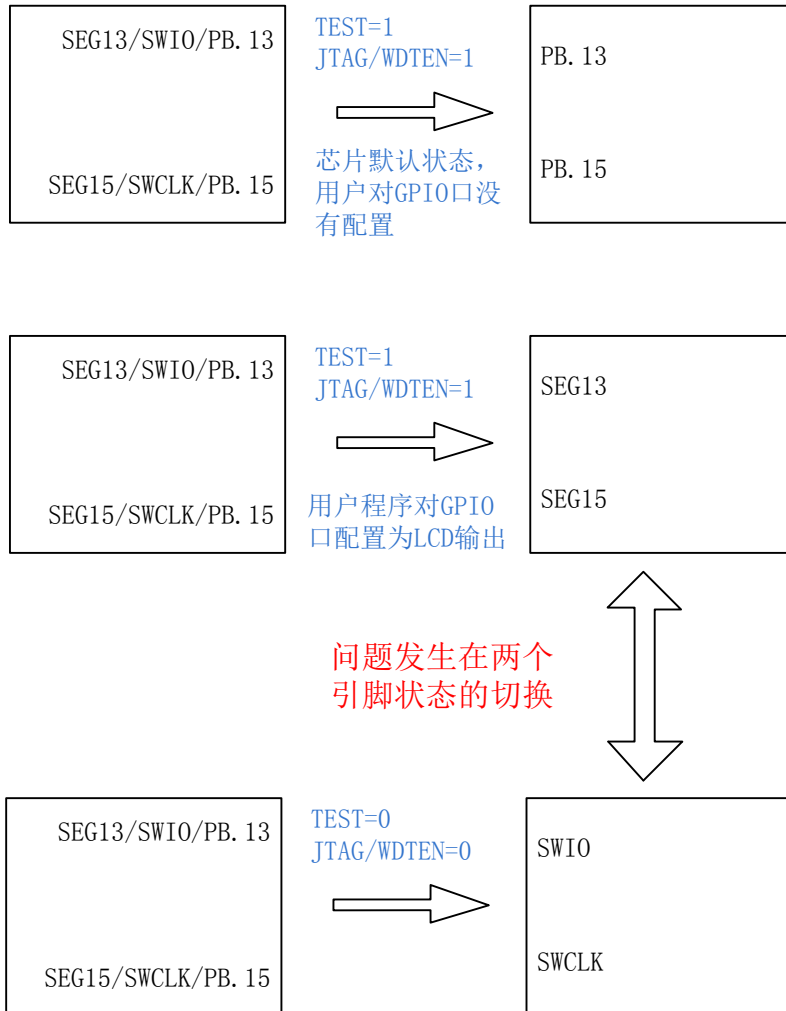
规避方法：

方法 1：提高 MCU 工作电压，与外部上拉电压一致，可避免此种情况；

方法 2：使用 5V tolerance IO 配置为 OD 输出，外接 5V 上拉，可避免此种情况。

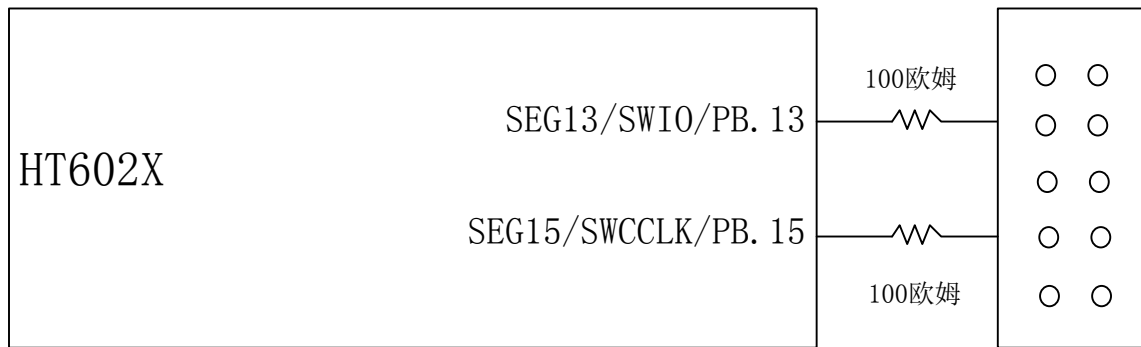
5 可靠性问题

5.1 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, 如果产生静电则有可能导致芯片 LCD 模块内部损坏。规避方法如下:

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, 靠近 SWD 接口串接 100 欧姆电阻滤波, 如下图所示:



仿真下载
10PIN接口

用户PCB



5.2 IAR 环境下，仿真接口增加 RST 建议

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，会造成在 IAR 环境下，仿真器与目标板无法再次建立连接。

为了解决该问题，同时增加仿真烧写的便利，推荐规避方法如下：

- 硬件上，建议拉出 MCU 的外部复位脚 RST 连接到仿真接口（在仿真状态下，该仿真接口与 ULINK/JLINK 的 PIN15 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。
- 软件上，用户程序检测到 MCU 外部 RST 引脚产生的 reset 信号之后延时至少 200ms，保证进入 sleep/hold 前，J-LINK 发出的“hold 住 cpu”指令完成。

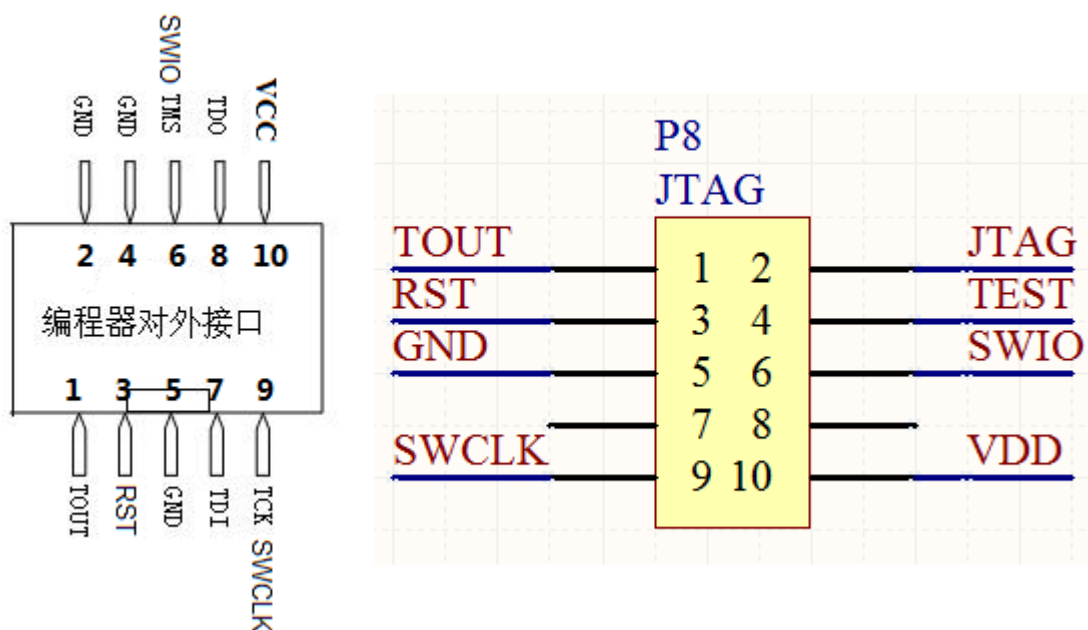
General Ulink Port

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

推荐以及鉅泉工具的仿真接口：



注：RST 用于连接 MCU RST PIN 和仿真器的 PIN 15