



新浪博客

近5年中国男女择偶标准对比

[登录](#) [注册](#)

超简洁
新版发布器
你用过了吗?

云南之旅行
7月21日 出发

浮云的博客
http://blog.sina.com.cn/u/1412148625 [订阅] [手机订阅]

首页 博文目录 图片 关于我

字体大小: 大 中 小

浮云
微博

加好友 发纸条
写留言 加关注

博客等级: 15
博客积分: 611
博客访问: 85,268
关注人气: 32
获赠金笔: 8
赠出金笔: 0
荣誉徽章:

正文

终于我也可以对STM32低功耗进行总结 (2013-09-26 23:16:05) 转载 ▼

标签: 休闲 分类: STM32

STM32F103R8和RC的停机模式的休眠电流还不一样，R8停机模式实测为11UA,RC停机模式实测为30uA,还以为又是我的程序哪里没做好呢，仔细看了PDF，这两个芯片PDF上标的值的确有区别，和我测的值差不多，那我就没有再深究的意义了！

表14 运行模式下的最大电流消耗，数据处理代码从内部闪存中运行

符号	参数	条件	f _{HCLK}	最大值 ⁽¹⁾		单位
				T _A = 85℃	T _A = 105℃	
I _{DD}	运行模式下的供应电流	外部时钟 ⁽²⁾ ， 使能所有外设	72MHz	69	70	mA
			48MHz	50	50.5	
			36MHz	39	39.5	
			24MHz	27	28	
			16MHz	20	20.5	
			8MHz	11	11.5	
		外部时钟 ⁽²⁾ ， 关闭所有外设	72MHz	37	37.5	
			48MHz	28	28.5	
			36MHz	22	22.5	
			24MHz	16.5	17	
			16MHz	12.5	13	
			8MHz	8	8	



相关博文

9任男友揭秘汤唯的混乱情史
黛玲

- 街拍：身材傲人的性感美女曹作兰
- STM32F3的低功耗模式的耗电量是M阿七
- J-LINKtheconnectedemulatorisaj东东bh
- 修复山寨jlinkV8固件丢失问题__Lay
- 关于STM32通用定时器更新事件中崩summerchencong
- JLink修复过程中出现writeflash—玲声依旧美
- STM32AWURTC闹钟从停机模式唤醒C浮云
- STM32学习笔记——实时时钟RTC张彬
- 《ARMCortex-M3权威指南》中英文残翅天使
- stm32待机模式summerchencong
- STM32学习笔记之EXTI（外部中断）rianboe

更多>>



- 推荐资讯
- 福清西山学校面向全国招生
福清西山学校面向全国招收幼儿园
- 必看:初中高中正确学习方法
北大出版 百万家长推荐 孩子学习
- 出国不用怕一个月甩掉翻译出国游
不背单词不跑培训班 助你轻松学
- 沪指重返3900 大盘是否调整到位
散户怎么办 高手也迷惑 赚钱不能
- 指南针 教你如何逃离股灾
指南针18年精心打造安全放心 两



¥120.00

推荐博文

表17 停机和待机模式下的典型和最大电流消耗

符号	参数	条件	典型值 ⁽¹⁾		最大值		单位
			V _{DD} /V _{BAT} = 2.4V	V _{DD} /V _{BAT} = 3.3V	T _A = 85℃	T _A = 105℃	
I _{DD}	停机模式下的供应电流	调压器处于运行模式，低速和高速内部RC振荡器和高速振荡器处于关闭状态(没有独立看门狗)	34.5	35	379	1130	μA
		调压器处于低功耗模式，低速和高速内部RC振荡器和高速振荡器处于关闭状态(没有独立看门狗)	24.5	25	365	1110	
	待机模式下的供应电流	低速内部RC振荡器和独立看门狗处于开启状态	3	3.8	-	-	
		低速内部RC振荡器处于开启状态，独立看门狗处于关闭状态	2.8	3.6	-	-	
		低速内部RC振荡器和独立看门狗处于关闭状态，低速振荡器和RTC处于关闭状态	1.9	2.1	5 ⁽²⁾	6.5 ⁽²⁾	
I _{DD_VBAT}	备份区域的供应电流	低速振荡器和RTC处于开启状态	1.1	1.4	2 ⁽²⁾	2.3 ⁽²⁾	

1. 典型值是在T_A=25℃下测试得到。
2. 由综合评估得出，不在生产中测试。

结合下文的高手经验，反复摸索，

standby模式1.9uA，PWR_EnterSTOPMode(PWR_Regulator_LowPower, PWR_STOPEntry_WFI);
stop模式：11uA, PWR_EnterSTANDBYMode();

实验证明，将IO端口设成IPU/IPD/AIN/PPOUT=1/PPOUT=0/ODOUT=0，
电流是基本相同的，最可怕的就是GPIO浮空，且电路上未外接上拉下拉，这样
电流就会比较大。

原来以来PPOUT要是输出为0，就会浪费电流，其实只要不负载，电流和
ODOUT=0时一样样的。

以下为部分代码

```
void MUC_SLEEP(void)
{
    GPIO_PinRemapConfig(GPIO_Remap_SWJ_Disable, ENABLE); //完全失能JTAG/SWD 不关电流也好像不影响
    GPIO_PinRemapConfig( GPIO_Remap_PD01, ENABLE );//晶振为GPIO 不关电流也好像不影响
    RCC_LSEConfig(RCC_LSE_OFF);//关闭RTC 不关电流也好像也不影响
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_PWR, ENABLE);
    ADC_CONFIG(DISABLE);
    USART_DeInit(USART1);
    USART_DeInit(USART2);
    USART_DeInit(USART3);
```

[查看更多>>](#)

谁看过这篇博文

 加载中...

```
USART_DeInit(UART5);
EXTI_INITIAL(ENABLE);//以下端口的设置最有影响
GPIO_PIN_INITIAL(GPIOA,GPIO_Pin_All,GPIO_Mode_AIN,0); //此处没做外部唤醒 仅用于测试 EXTI合理设置GPIO不影响电
流
GPIO_PIN_INITIAL(GPIOB,GPIO_Pin_All,GPIO_Mode_AIN,0);
GPIO_PIN_INITIAL(GPIOC,GPIO_Pin_All,GPIO_Mode_AIN,0);
GPIO_PIN_INITIAL(GPIOD,GPIO_Pin_All,GPIO_Mode_AIN,0);
//PWR_EnterSTOPMode(PWR_Regulator_LowPower, PWR_STOPEntry_WFI);//STOP模式

PWR_EnterSTANDBYMode();//standby模式

}
```

以下来自 [这个总结似乎更更贴合用户](#)
<http://bbs.21ic.com/icview-558242-1-1.html>

[dzxxlxd](#) 发表于 2013-7-15 16:10:33 | [只看该作者](#) | [返回版面](#)

找到方法了，最终板级数据：
stop模式，外部中断唤醒：13uA
standby模式，2.5uA

[dzxxlxd](#)
实习生
专家等级：
结帖率:0%

6#
[dzxxlxd](#) 发表于 2013-7-15 16:22:08 | [只看该作者](#) | [返回版面](#)

具体要点为：
1、所有IO管脚，如果高阻状态端口是高电平，就设成上拉输入，如果高阻状态是低电平，就设成下拉输入，必须的。作为输出就免了，待机你想输出个什么东西，一定要输，硬件上上拉。
2、两个晶振输入脚要remap成普通IO!!! 使用内部晶振。
3、pwr的时钟要使能，即RCC_APB1PeriphClockCmd(RCC_APB1Periph_PWR, ENABLE);
4、关闭jtag口，并设成普通IO；
5、注意助焊膏的质量!!! 注意电路板层之间是否进水!!!!
掌握这几项要点，再设中断什么的都行，整个世界清静了!!!! 完全低能耗。

=====以下摘自网络=====

STM32实现低功耗待机总结(电流低至5.7uA)

分类: [STM32](#) 2013-01-16 10:05 670人阅读 评论(0) [收藏](#) [举报](#)

刚开始进入STOPMode后，整机功耗有300uA的，此时外围其他硬件电路电流已经可以肯定漏电流在nA级，因此调试方向在主芯片，经过实际测试，都是GPIO配置的问题，比如某个GPIO为中断输入，闲置为低电平，而我们配置成了IPU，因此内部的40K上拉就会在这里消耗3/40k =75uA，另外将N.C的GPIO配置成Floating Input，也会有一些漏电流，实际测试漏电流不大;另外将STM32F05x直接PIINtoPIN替代STM32100，所以Pin35,36的PF6,PF7为之前的VCC, GND，因此要相应的配置为IPU, IPD，才不会有拉电流/灌电流;外部不使用晶振，因此必须将其配置为IPU/IPD或者输出Low，如果配置成Floating，实测消耗200uA+的电流，这个特别注意。另外不需要关闭不用的外设的CLK，因为STOPMODE会将内部1.8V的core关闭，因此该步骤不影响功耗。

因此在进入STOPMODE之前，需要做：

- 1、将N.C的GPIO统一配置为IPU/IPD;
- 2、检查一些Signal的输入Active是High/Low，相应进行配置为IPD/IPU，即避免在内部上/下拉电阻上消耗电流，而且该电流理论值为VCC/R = 3/40 =75uA;
- 3、如果外部晶振不使用，必须将GPIO配置为IPU/IPD/PPLow，不允许配置为floating，否则会消耗极大的电流200uA+;
- 4*、加入进入STOPMODE前，不允许将PWR的CLK关闭，这部分牵涉低功耗模式，实际测试关闭能用，也能唤醒，但是电流会增加10uA+;
- 5、配置GPIO为输出时，根据输出的常态选择上拉/下拉，如闲置输出为0，则配置为下拉，输出闲置为1，则配置上拉;

6、另外特别说明的是->从Stopmode唤醒后，系统会自动切换到HSI，如果进入前使用的是外部晶振/PLL(PLL的clksource = HSI/HSE)因此必须调用System_Init()，对RCC重新初始化，否则唤醒后主频发生改变，会影响系统;

7

2

喜欢

赠金笔

分享:

阅读(3370) | 评论 (0) | 收藏(0) | 转载(2) | 喜欢▼ | 打印 | 举报

已投稿到: 排行榜

前一篇: 双向可控硅驱动运用 (我总是忘记这个)
后一篇: STM8S103 105唯一序列码的读取 (转的)

评论 重要提示: 警惕虚假中奖信息

[发评论]

评论加载中,请稍候...

发评论



顶



顶



顶



顶



顶



顶



顶



顶

登录名: 密码: [找回密码](#) [注册](#) ☒ 记住登录状态

☒ 分享到微博 ☐ 评论并转载此博文

验证码: [请点击后输入验证码](#) [收听验证码](#)

发评论

以上网友发言只代表其个人观点，不代表新浪网的观点或立场。

< 前一篇 后一篇 >
双向可控硅驱动运用 (我总是忘记这个) STM8S103 105唯一序列码的读取 (转的)