

HT32F125x 低功耗模式

文件编码: HA0284S

概述

简介

此应用范例用于帮助系统设计者对 HT32F125x 系列产品的低功耗模式在软硬件实现方面有简要地了解。用户可以学习如何使用 HT32F125x 产品及如何合理安排低功耗模式，以实现 HT32F125x 在低功耗应用方面的最优化处理。

相关文件

- HT32F125x 规格书
- HT32F125x 使用手册

功耗和唤醒时间测量

简介

阅读此节后，用户可以了解 HT32F125x 低功耗模式功耗和唤醒时间的测量问题。

功耗测量

用户可通过此应用范例附带的 Zip 文件中提供的固件来测量 HT32F125x 的功耗。该固件存放在 CurrentMeasurements 文件夹中。

可在以下低功耗模式中进行功耗测量：

- 休眠模式
休眠模式下仅有 Cortex™-M3 CPU 内核的时钟停止运行。此模式下的功耗可由时钟源及有效的外设决定。为涵盖此模式的所有功能，测量时用到不同时钟源（HSI 和 HSE）、时钟频率（1MHz~72MHz）、APB 外设配置（所有外设时钟开启或所有外设时钟关闭——包括 Flash 和 SRAM）。
- 深度休眠模式 1
深度休眠模式 1 中，V_{DD18} 电源域的所有时钟停止运行。高速振荡器（HSI、HSE）和 PLL 停止振荡，V_{DD18} 电压源为 LDO（处于低电流模式时）。
- 深度休眠模式 2
除了 V_{DD18} 电压源为 DMOS 外，深度休眠模式 2 与深度休眠模式 1 相同。
- 暂停模式：关闭 V_{DD18} 电源域，保留 V_{DD33} 和 V_{BAK} 电源域。

- 注意：1. 不同低功耗模式的更多详情请参考 HT32F125x 用户手册。
2. 处于休眠模式和深度休眠模式时，所有未使用的 I/O 脚都配置为输入浮空状态，斯密特触发输入除能，因此，这些 I/O 引脚的功耗为 0。

如何测量功耗

HT32F125x 开发板的功耗测量可通过两种方式测得：可利用电表代替跳线 J25，并且由一个外部供电给开发板；通过使用 USB 线。

固件描述

头文件（main.h）中，可通过选择几个#define 参数化范例。

- 取消相应行定义的注释，选择休眠模式所需的时钟配置


```
#define HSE_PLL_ON
#define HSE_PLL_ON_72MHz
//define HSE_PLL_ON_36MHz
//define HSE_PLL_ON_18MHz

//define HSE_PLL_OFF
//define HSE_PLL_OFF_8MHz
//define HSE_PLL_OFF_1MHz

//define HSI_PLL_ON
//define HSI_PLL_ON_72MHz
//define HSI_PLL_ON_36MHz
//define HSI_PLL_ON_18MHz

//define HSI_PLL_OFF
//define HSI_PLL_OFF_8MHz
//define HSI_PLL_OFF_1MHz
```
- 取消相应行定义的注释，选择所需的低功耗模式


```
#define SLEEP
//define DEEP_SLEEP_1
//define DEEP_SLEEP_2
//define POWER_DOWN
```
- 取消相应行定义的注释，选择所需外设时钟（门控时钟）


```
//define SLEEP_ALLPERIPH_ENABLE
#define SLEEP_ALLPERIPH_DISABLE
```
- 休眠模式下，用户可通过取消相应行定义的注释来选择所需的 FMC 时钟选项


```
//define SLEEP_FMC_ENABLE
#define SLEEP_FMC_DISABLE
```
- 休眠模式下，用户可通过取消相应行定义的注释来选择所需的 SRAM 时钟选项


```
//define SLEEP_SRAM_ENABLE
//define SLEEP_SRAM_DISABLE
```

注意：无论时钟源来自 HSI 还是 HSE，若系统时钟频率等于或小于 8MHz，PLL 都将关闭。

执行低功耗范例后，若用户想重新载入 Flash 存储器的内容，需将启动模式由主 Flash 切换到 SRAM，并且要按下复位键。此举是因为当 HT32F125x 处于低功耗模式时，调试器不能连接到 HT32F125x。启动模式需重新配置到主 Flash。而后，通过重启目标板开始进行测量。

测量结果

测量结果如表 1 和表 2 所示。

表 1 休眠模式下功耗测量结果

条件	f _{HCLK}	所有 APB 外设使能	所有 APB 外设除能
运行 HSE。 AHB 预分频器用于减小频率， 休眠模式下 FMC 和 SRAM 时钟关闭	72MHz	26.42mA	6.76mA
	36MHz	15.64mA	5.25mA
	18MHz	10.1mA	4.49mA
	8MHz	4.39mA	1.43mA
	1MHz	2.21mA	1.13mA
运行 HSE。 AHB 预分频器用于减小频率， 休眠模式下 FMC 和 SRAM 时钟开启	72MHz	28.27mA	8.67mA
	36MHz	16.57mA	6.21mA
	18MHz	10.57mA	4.98mA
	8MHz	4.6mA	1.64mA
	1MHz	2.24mA	1.16mA
运行 HSI。 AHB 预分频器用于减小频率， 休眠模式下 FMC 和 SRAM 时钟关闭	72MHz	26.39mA	6.63mA
	36MHz	15.5mA	5.12mA
	18MHz	9.97mA	4.36mA
	8MHz	4.33mA	1.36mA
	1MHz	2.15mA	1.06mA

表 2 深度休眠模式和暂停模式下功耗测量

模式	条件	V _{DD33} /V _{BAT} = 3.3V
深度休眠 1 模式	V _{DD18} 来自 LDO, LDO 处于低功耗模式, RTC on, 使用 LSI	67uA
深度休眠 2 模式	V _{DD18} 来自 DMOS, LDO Off, RTC on, 使用 LSI	11.5uA
暂停模式	V _{DD18} Off, RTC on, 使用 LSI	4.5uA

唤醒时间测量

此节描述了如何测量 HT32F125x 由不同低功耗模式下唤醒的时间。此应用范例提供了相关的固件，固件位于提供的 Zip 压缩包内的 WakeUpTiming 文件夹下。

唤醒时间定义

- 休眠模式和深度休眠模式
唤醒时间起始于 RTCOUT (PB10) 的上升沿，结束于 WFE 后执行完第一条指令。
- 暂停模式
从暂停模式中唤醒后，与遇到复位条件相同，程序重新执行。暂停模式的唤醒时间介于 RTCOUT (PB10) 的上升沿与执行完第一条指令之间。

如何测量唤醒时间

系统由低功耗模式唤醒后，PA5 脚被设置以测量唤醒时间，示波器需连接在 PA5 和 PB10 之间。唤醒时间介于 PB10 (RTCOUT) 的上升沿和 PA5 的上升沿之间。

固件描述

头文件（main.h）中，可通过选择几个#define 参数化范例。

- 取消相应行定义的注释，选择所需的低功耗模式

```

#define SLEEP
#define DEEP_SLEEP_1
#define DEEP_SLEEP_2
#define POWER_DOWN
    
```

- 取消相应行的注释，选择所需的系统时钟源

```

/* Define the system clock */
#define HCLK_HSI
#define HCLK_HSI_PLL
#define HCLK_HSE
#define HCLK_HSE_PLL
    
```

在主文件（main.c）中。进入低功耗模式前，PA5 配置为输出推挽式，复位到低电压状态。

由低功耗模式唤醒后：

- 如果从休眠模式和深度休眠模式唤醒，则执行直接写入 GPIOA_SRR（输出设置复位控制寄存器）以设置 PA5 为高电平。
- 如果从暂停模式唤醒，则在代码启动时 PA5 脚应被设置（此部分代码位于启动文件中，如下所示）。

为测量暂停模式的唤醒时间，需先配置和设置 PA5 脚。为此，添加如下汇编代码到工具启动文件：

```

/* Enable peripheral clocks of AFIO and GPIOA */
LDR R0,=0x4008802C
LDR R1,=0x00014000
STR R1,[R0]
/* PA5 output high */
LDR R0,=0x4001A000
LDR R1,=0x0020
STR R1,[R0,#0x24]
STR R1,[R0]
    
```

注意：执行低功耗范例后，若用户想重新载入 Flash 存储器的内容，需将启动模式由主 Flash 切换到 SRAM，并且要按下复位键。此举是因为当 HT32F125x 处于低功耗模式时，调试器不能连接到 HT32F125x。启动模式需重新配置到主 Flash。而后，通过重启目标板开始进行测量。

测量结果

休眠模式、深度休眠模式和暂停模式的唤醒时间测量结果如表 3 所示。

表 3 唤醒时间测量结果

符号	参数	条件	典型值
$t_{WUSLEEP}$	从休眠模式中唤醒	在 HSI 时钟下唤醒	1.04us
t_{WUDS1}	从深度休眠模式 1 中唤醒		17.4us
t_{WUDS2}	从深度休眠模式 2 中唤醒		17.4us
t_{WUPD}	从暂停模式中唤醒		118us
$t_{WUSLEEP}$	从休眠模式中唤醒	在 HSE 时钟下唤醒	1.04us
t_{WUDS1}	从深度休眠模式 1 中唤醒		5.04ms
t_{WUDS2}	从深度休眠模式 2 中唤醒		5.04ms
$t_{WUSLEEP}$	从休眠模式中唤醒	在 PLL 时钟下唤醒 PLL 使用 HSI	0.2us
t_{WUDS1}	从深度休眠模式 1 中唤醒		146us
t_{WUDS2}	从深度休眠模式 2 中唤醒		146us
$t_{WUSLEEP}$	从休眠模式中唤醒	在 PLL 时钟下唤醒 PLL 使用 HSE	0.2us
t_{WUDS1}	从深度休眠模式 1 中唤醒		5.16ms
t_{WUDS2}	从深度休眠模式 2 中唤醒		5.16ms

结论

根据不同的结果，用户可以在功耗和唤醒时间之间作一权衡，前提条件是：功耗更低、唤醒时间更长。

由于 HSE 和 PLL 需更长的准备时间，故当时钟源为 HSE 或 PLL 时，深度休眠模式唤醒时间更长。为缩短深度休眠模式的唤醒时间，用户可以先使用 HSI，然后切换到其它外设时钟源（系统时钟、HSE 或 PLL）。

根据应用的限制，用户应发现最好的权衡结果。

功耗优化

简介

事实上：单片机的功耗随着时钟频率的增大而增加。所以，用户须寻找到最优的功耗/性能比。很多应用中，可通过调整系统/外设频率达到所需的性能而降低功耗。若系统/外设工作无特别需求，可使用 HT32F125x 的低功耗模式。

应用中使用时钟配置

本节介绍如何使用 HT32F125x 的时钟配置，所使用的固件位于本应用范例的 Zip 压缩包内的 Run_Mode 文件夹中。

该程序使用 HT32F125x 的 USART 从 RTC 传送时间。

- 首先，用户必须使用超级终端调整时间。
- 时间显示在超级终端上，并且每秒刷新一次。RTC 设置为每秒产生一次中断。
- 当中断发生时，捕捉 RTC 计数器的值；计算时间，并使用 USART 传送出去。

硬件环境

在 HT32F125x 开发板上使用此范例：请参考 HT32F125x 开发板的用户手册

- 使用 HT32F125x 开发板包装中的 USART 线。
- DB9 连接器和 PC 串口间必须使用一个零调制解调器两端均为母端子的 RS232 线连接。
- 由一个电表代替跳线 J25 测量功耗。

固件描述

- 在 PC 上配置超级终端
 - 字长 = 8 bits
 - 一个 Stop 位
 - 无奇偶校验
 - 波特率 = 115200
 - 流程控制：无
- 配置固件
 - 头文件 (main.h) 中，可通过选择几个 #define 参数化范例。
 - 定义外设选择 (门控时钟)


```

                                #define ALL_PERIPHERIALS_ENABLE
                                #define ONLY_USART_RTC_ENABLE
                            
```
 - 定义频率选择


```

                                #define HCLK_72MHz
                                #define HCLK_8MHz
                            
```
 - 定义应用程序等待 RTC 中断时切换到休眠模式


```

                                #define SLEEP_WFI_ON
                            
```

注意：执行低功耗范例后，若用户想重新载入 Flash 存储器的内容，需将启动模式由主 Flash 切换到 SRAM，并且要按下复位键。此举是因为当 HT32F125x 处于低功耗模式时，调试器不能连接到 HT32F125x。启动模式需重新配置到主 Flash。而后，通过重启目标板开始进行测量。

测量结果

表 4 在 25°C 时运行模式测量举例

外设时钟	频率	休眠	典型功耗
ALL On	72MHz	No	51.8mA
仅 RTC On	72MHz	No	37.4mA
仅 RTC On	72MHz	Yes	12.7mA
仅 RTC On	8MHz	No	11.1mA
仅 RTC On	8MHz	Yes	2.05mA

结论

为减少功耗，HT32F125x 必须根据用户需求以最优化配置进行初始化。用户必须着眼于应用的要求，并配置相应的 HT32F125x。从这个范例中，用户可以借鉴可能的 HT32F125x 时钟配置和优化应用程序的功耗。

配置描述如下：

- 系统和外设频率
若应用程序无需在最大频率下运行，用户可通过使用 PLL 或预分频器分频以减小 HCLK。
- 门控时钟
为优化功耗，用户应通过 HT32F125x 门控时钟选项除能未使用的外设。
- 休眠模式
减少功耗的另一个方法是当应用程序等待事件或者中断时切换到 HT32F125x 的休眠模式。

在电池产品应用中使用深度休眠模式和暂停模式

简介

由电池供电的一些应用程序不是一直处于运行中的。此种应用中，单片机等待外部事件，且需要在暂停运行期间减少功耗。

此应用范例提供了两个在电池供电应用中如何使用 HT32F125x 的范例（深度休眠模式和暂停模式）。这些范例中，一旦应用程序无需处理，HT32F125x 就切换到低功耗模式。这两个低功耗模式基于 Cortex-M3 内核的深度睡眠功能和 WFE 指令。

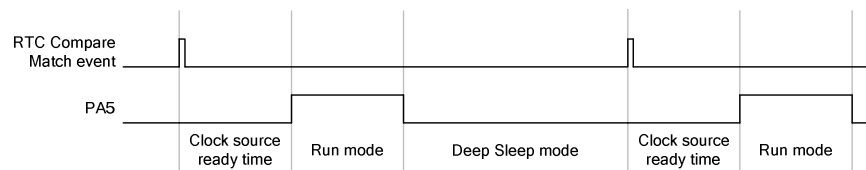
深度休眠模式

本节介绍如何使用 HT32F125x 的 WFE 指令和深度休眠模式，所使用的固件位于本应用范例的 Zip 压缩包内的 DeepSleepMode 文件夹中。

该范例执行定期 ADC 转换并将转换结果存储在 RAM 缓冲中。它采用 RTC 比较匹配事件以自动唤醒深度休眠模式。

- 首先，配置备份域、GPIO 和 ADC。
- RTC 正在运行，且由主循环中的 RTC 比较匹配触发每次 ADC 转换和 ADC 转换结果的存储。
- RTC 比较匹配事件将 HT32F125x 由深度休眠模式中唤醒。
- PA5 I/O 显示了 ADC 转换和 RTC_CMP 寄存器重新载入所需的时间。

图 1 深度休眠模式举例



固件描述

头文件（main.h）中，可通过选择几个#define 参数化范例。

- 取消相应模式定义的注释，选择所需的低功耗模式


```

                //#define DEEP_SLEEP_1
                //#define DEEP_SLEEP_2
            
```
- 选择循环时序，取消相应行的注释


```

                //#define LOOP_50mS
                //#define LOOP_500mS
                //#define LOOP_1S
                //#define LOOP_3S
                //#define LOOP_5S
            
```

注意：执行低功耗范例后，若用户想重新载入 Flash 存储器的内容，需将启动模式由主 Flash 切换到 SRAM，并且要按下复位键。此举是因为当 HT32F125x 处于低功耗模式时，调试器不能连接到 HT32F125x。启动模式需重新配置到主 Flash。而后，通过重启目标板开始进行测量。

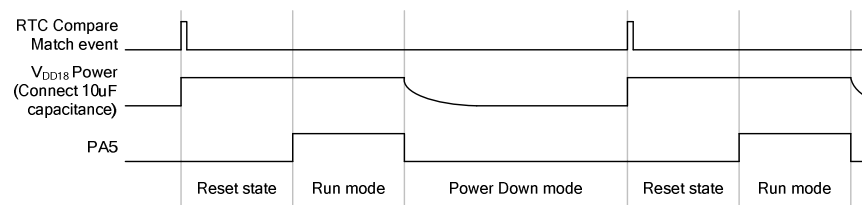
暂停模式

本节介绍如何使用 HT32F125x 的暂停模式，所使用的固件位于本应用范例的 Zip 压缩包内的 PowerDownMode 文件夹中。

该范例执行定期 ADC 转换并将转换结果存储在 RAM 缓冲中。它采用 RTC 比较匹配事件以自动唤醒暂停模式。

- 首先，配置备份域、GPIO 和 ADC。
- RTC 运行且 RTC 比较匹配事件将 HT32F125x 由暂停模式中唤醒。
- 每次由暂停模式中唤醒后，HT32F125x 从复位状态重启，且在 HT32F125x 已配置后执行每次 ADC 转换。
- PA5 I/O 显示了程序运行所需的时间。

图 2 暂停模式举例



固件描述

头文件（main.h）中，可通过选择几个#define 参数化范例。

- 选择循环时序，取消相应行的注释


```

                //#define LOOP_50mS
                //#define LOOP_500mS
                //#define LOOP_1S
                //#define LOOP_3S
                //#define LOOP_5S
            
```

注意：执行低功耗范例后，若用户想重新载入 Flash 存储器的内容，需将启动模式由主 Flash 切换到 SRAM，并且要按下复位键。此举是因为当 HT32F125x 处于低功耗模式时，调试器不能连接到 HT32F125x。启动模式需重新配置到主 Flash。而后，通过重启目标板开始进行测量。

如何测量电流消耗

HT32F125x 开发板的功耗测量可通过两种方式测得：可利用电表代替跳线 J25，并且由一个外部供电给开发板；通过使用 USB 线。

测量结果

测量结果如表 5 所示。

表 5 在 25°C 下低功耗模式测量结果举例

低功耗模式	50ms	500ms	1s	3s	5s
深度休眠模式 1	77.2uA	69.3uA	69.2uA	68.6uA	68.5uA
深度休眠模式 2	23.4uA	14.3uA	13.8uA	13.4uA	13.4uA
暂停模式	363.2uA	46.2uA	26.4uA	10.7uA	8.8uA

结论

由于在 Cortex-M3 CPU 内核集成了用于低功耗应用（休眠模式和深度休眠模式）的高效核心级指令，同时结合 HT32F125x 的低功耗特性，使用户可以根据应用的需求来优化系统功耗。

测量表明，必须考虑唤醒时间和功耗之间的最优化。