

32 位微控制器

HC32L15 系列的低功耗模式

适用对象

系列	产品型号
HC32L15	HC32L150KATA
	HC32L150JATA
	HC32L150FAUA
	HC32L156KATA
	HC32L156JATA

目 录

1	摘要	3
2	低功耗模式简介	3
3	HC32L15 系列的低功耗模式.....	4
3.1	简介	4
3.2	说明	5
3.2.1	各种模式介绍.....	5
3.2.2	各模式时钟工作状态.....	6
3.2.3	各模式唤醒源.....	7
3.2.4	各模式状态转换.....	8
3.2.5	寄存器介绍.....	9
3.2.6	工作流程介绍.....	10
4	样例代码	15
4.1	代码介绍	15
4.2	代码运行	17
5	总结	18
6	版本信息 & 联系方式.....	19

1 摘要

本篇应用笔记主要介绍如何使用 HC32L15 系列的低功耗模式的种类、性能及使用说明。

2 低功耗模式简介

什么是低功耗模式？

低功耗模式相对于芯片的正常工作模式，指利用芯片硬件设计实现的部分功能正常工作但芯片功耗相对较低的工作模式。

低功耗模式的作用？

低功耗模式可以保证某些必要功能（如 RTC、LCDC、RAM 数据保持等）正常工作的情况下，尽可能多地关掉空闲的耗电部分，降低产品整体功耗，提高产品待机时间，延长电池使用寿命，增加产品竞争力。

现阶段，低功耗模式在各产品方案中广泛应用于便携式电子设备、智能电水气表、车载娱乐等。特别是随着可穿戴设备的风靡，市场对芯片的低功耗模式需求越来越大，并且性能要求越来越高。

3 HC32L15 系列的低功耗模式

3.1 简介

华大 HC32L15 系列单片机支持多种低功耗模式，按照功能不同，分为 Idle 模式、Standby 模式和 Deep Standby 模式。其中 Standby 模式又分为 RTC 模式和 STOP 模式，Deep Standby 模式又分为 Deep RTC 模式和 Deep STOP 模式。功耗由高到低排序为：

- Idle 模式
 - HICR idle 模式
 - HSXT idle 模式
 - LICR idle 模式
 - LSXT idle 模式
- Standby 模式
 - RTC 模式
 - STOP 模式
- Deep Standby 模式
 - Deep RTC 模式
 - Deep STOP 模式

3.2 说明

该部分将详细介绍各种低功耗模式工作特点、低功耗模式相关寄存器和进入低功耗模式工作流程。

3.2.1 各种模式介绍

1) Idle 模式

Idle 模式是指 CPU 停止执行指令，可以降低功耗，AHB 和 APB 时钟正常工作，大部分外设都处于工作状态。

2) RTC 模式

RTC 模式下会将 HSXT 和 HICR 关闭，LSXT 和 LICR 会根据应用选择只开一个或者是两个都开启。在 RTC 模式下除了 RTC、LCD、红外 UART、PCNT、WDT、VC 和 LVD 工作外，其他功能都会关闭。

3) STOP 模式

STOP 模式下所有的时钟都会关闭，在此模式下只有 VC 和 LVD 可以工作。

4) Deep RTC 模式

Deep RTC 模式下会将 HSXT 和 HICR 关闭，LSXT 和 LICR 会根据应用选择只开一个或者是两个都开启。此模式下只有 RTC、LCD、LVD 和唤醒 IO 可以工作。CPU 电源域在此模式下会断电，RAM 电源域会根据设置选择是否断电。

5) Deep STOP 模式

Deep Stop 模式下所有时钟都会关闭，在此模式下只有 LVD 和唤醒 IO 可以工作。CPU 电源域在此模式下会断电，RAM 电源域会根据设置选择是否断电。

3.2.2 各模式时钟工作状态

表 1 Idle 模式下时钟工作状态

	HICR Idle Mode	HSXT Idle Mode	LICR Idle Mode	LSXT Idle Mode
HICR	运行	根据设置决定	停止	停止
HSXT	根据设置决定	运行	停止	停止
LSXT	根据设置决定	根据设置决定	根据设置决定	运行
LICR	运行	运行	运行	运行
CPU clock	Stopped			
AHB bus clock	HICR clock	HSXT clock	LICR clock	LSXT clock
APB bus clock	HICR clock	HSXT clock	LICR clock	LSXT clock

表 2 Standby 模式下时钟工作状态

	RTC Mode	STOP Mode
HICR	停止	停止
HSXT	停止	
LICR	根据设置决定	
LSXT	根据设置决定	
CPU clock	停止	
AHB bus clock		
APB bus clock		

注意：

- RTC 模式下 LICR 和 LSXT 至少要开启一个，至于开启哪一个需要根据不同的设置来决定，也可以都开启。

表 3 Deep Standby 模式下时钟工作状态

	Deep RTC Mode	Deep STOP Mode
HICR	停止	停止
HSXT	停止	
LICR	根据设置决定	
LSXT	根据设置决定	
CPU clock	停止	
AHB bus clock		
APB bus clock		

注意:

- Deep RTC 模式下 LICR 和 LSXT 至少要开启一个，至于开启哪一个需要根据不同的设置来决定，也可以都开启。

3.2.3 各模式唤醒源

表 4 各模式唤醒源

唤醒方式	Idle Mode	RTC Mode	STOP Mode	DS-RTC Mode	DS-STOP Mode
复位唤醒	-RSTB Reset -LVD Reset -SWWDG Reset -HWWDG Reset -CSV Reset -FCS Reset	-RSTB Reset -LVD Reset -HWWDG Reset	-RSTB Reset -LVD Reset	-RSTB Reset -LVD Reset	-RSTB Reset -LVD Reset
中断唤醒	-Effective Interrupt from each Peripheral	-NMI Interrupt -External Interrupt(16) -PCNT Interrupt -RTC Interrupt -LVD Interrupt -VC Interrupt -HWWDG Interrupt -UART RxData Interrupt -UART CTSX IO Interrupt	-NMI Interrupt -External - Interrupt(16) -LVD Interrupt -VC Interrupt -UART CTSX IO Interrupt	-WKUP Pin Input(4) -RTC Interrupt -LVD Interrupt	-WKUP Pin Input(4) -LVD Interrupt

3.2.4 各模式状态转换

不同工作模式之间的转换关系如图 1 所示，Idle 状态与 Run 状态之间可以相互切换，Standby 状态与 Run 状态之间可以相互切换，但是 DeepStandby 状态被唤醒之后则只能进入 Reset 状态后才能转换为 Run 状态。

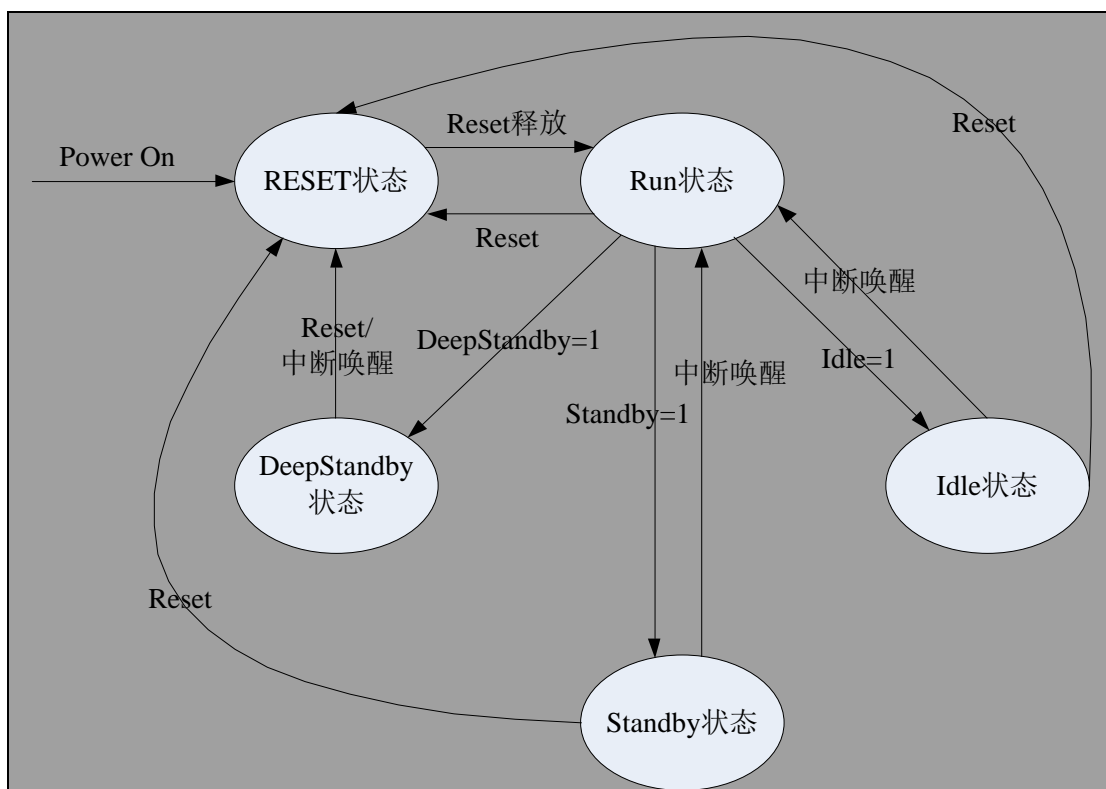


图 1 模式状态转换图

3.2.5 寄存器介绍

对于低功耗模式的操作主要通过以下寄存器进行，寄存器的详细介绍请参照 HC32L15 用户手册。

1) 模式控制寄存器

LPM_CTL

低功耗模式控制寄存器，用于控制进入 Standby 和 Deep Standby 模式。

CLK_CTL1

RTC 模式设置寄存器，用于设置是否使能 RTC 模式

2) 配置寄存器

BGR_CTL

用于控制 BGR 电源域的开关寄存器。

CLK_CTL2

外设始终设置寄存器

DS_RAMP

RAM 时钟设置寄存器

WAKE_IER

唤醒功能使能寄存器

WAKE_LVR

外部唤醒功能电平设置寄存器

3) 状态寄存器

WAKE_RSR

复位状态寄存器，用于表示有哪些复位请求产生。

WAKE_ISR

中断唤醒状态寄存器，用于表示有那些中断唤醒信号将 MCU 唤醒。

3.2.6 工作流程介绍

Idle 模式进入流程如图 2 所示，按照应用需求配置完 Idle 模式需要的唤醒方式后，Cortex-M0+ 系统控制寄存器 SLEEPDEEP 写为 0，然后调用 WFI 函数，则芯片进入 Idle 模式。

在 Idle 模式下，系统任何的复位和有效中断都可以将系统唤醒，中断唤醒后系统继续执行 WFI 之后的代码，复位唤醒后系统复位重新执行代码。

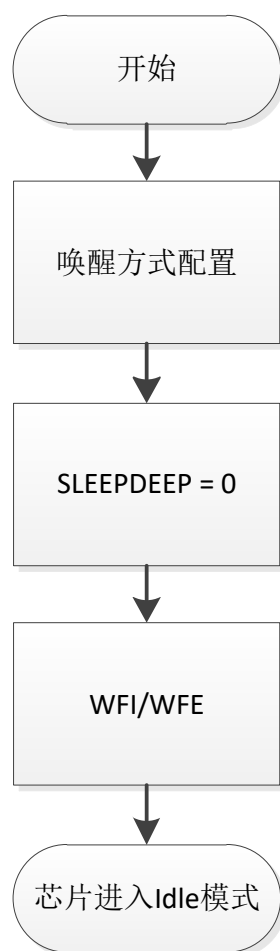


图 2 Idle 模式进入配置流程

RTC 模式的配置流程如图 3 所示，进入 RTC 模式前需要使能 RTC 功能，如果 RTC 功能选用的 LSXT 时钟，需要保证在进入 RTC 模式时，LSXT 时钟已经配置好，即 CLK_STB 寄存器的 SUB_RDY 位已经为 1。

在 RTC 模式下，系统任何的复位和有效中断都可以将系统唤醒，具体复位源如表 4 所示，中断唤醒后系统继续执行 WFI 之后的代码，复位唤醒后系统复位重新执行代码。

STOP 模式的配置流程如图 4 所示，在 STOP 模式下，系统任何的复位和有效中断都可以将系统唤醒，具体复位源如表 4 所示，中断唤醒后系统继续执行 WFI 之后的代码，复位唤醒后系统复位重新执行代码。

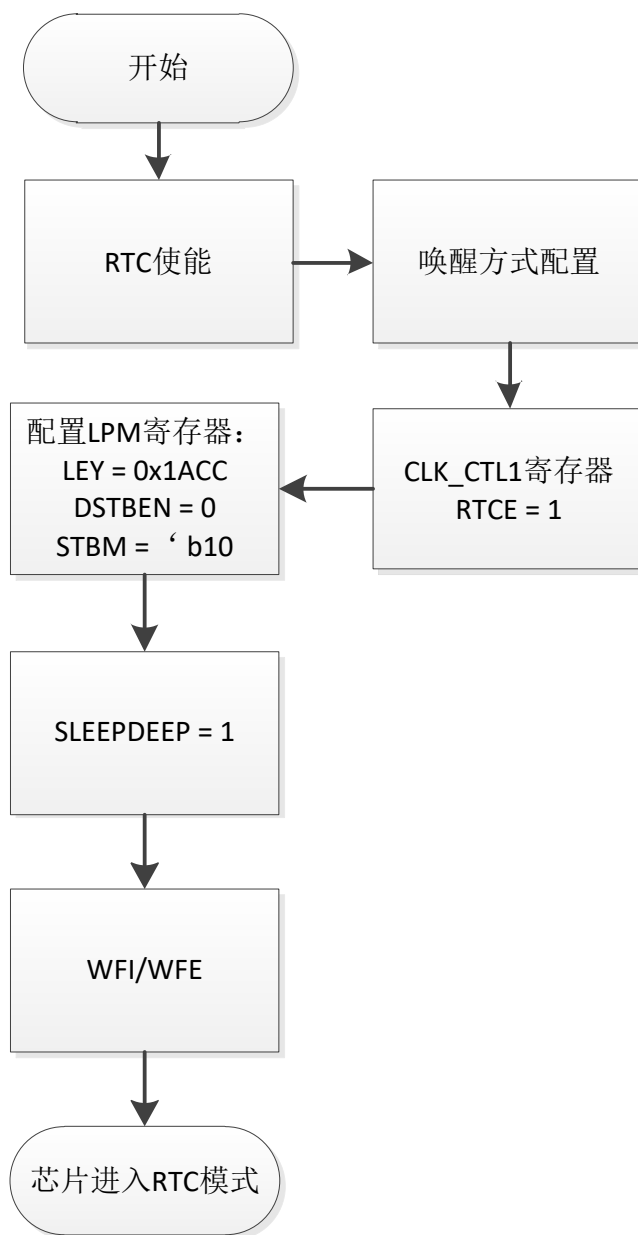


图 3 RTC 模式配置流程

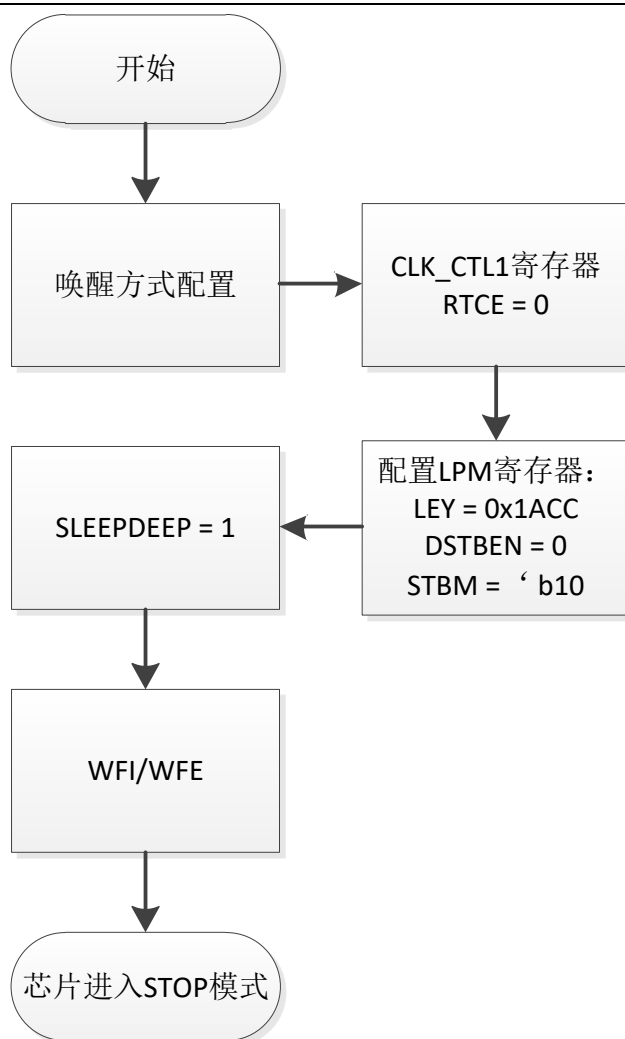


图 4 STOP 模式配置流程

Deep RTC 模式配置流程如图 5 所示，与 RTC 模式的区别仅仅在于 LPM 控制寄存器的 DSTBEN 位配置为 1。在 Deep RTC 模式下，LVD 复位、RSTB 复位以及 RTC 中断、LVD 中断、WKUP 中断可以将系统唤醒。，具体复位源如表 4 所示，中断唤醒和复位唤醒后系统均复位重新执行代码。

Deep STOP 模式配置流程如图 6 所示，与 STOP 模式的区别仅仅在于 LPM 控制寄存器的 DSTBEN 位配置为 1。在 Deep STOP 模式下，仅有 LVD 复位、RSTB 复位以及 WKUP 中断可以将系统唤醒。，具体复位源如表 4 所示，中断唤醒和复位唤醒后系统均复位重新执行代码。

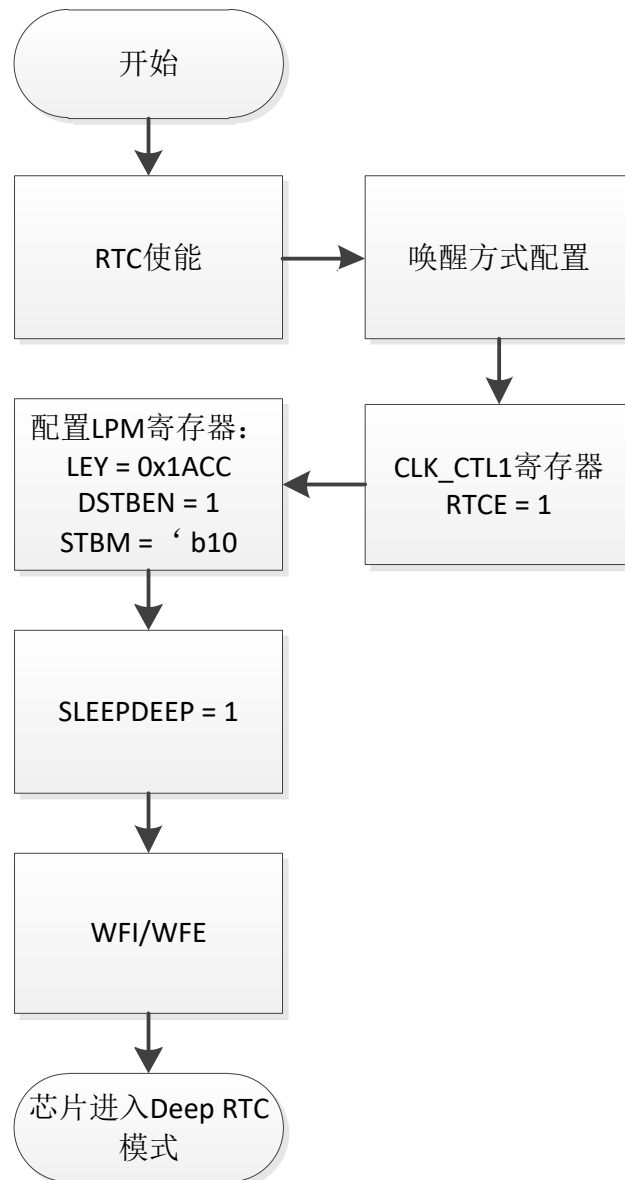


图 5 Deep RTC 模式进入配置流程

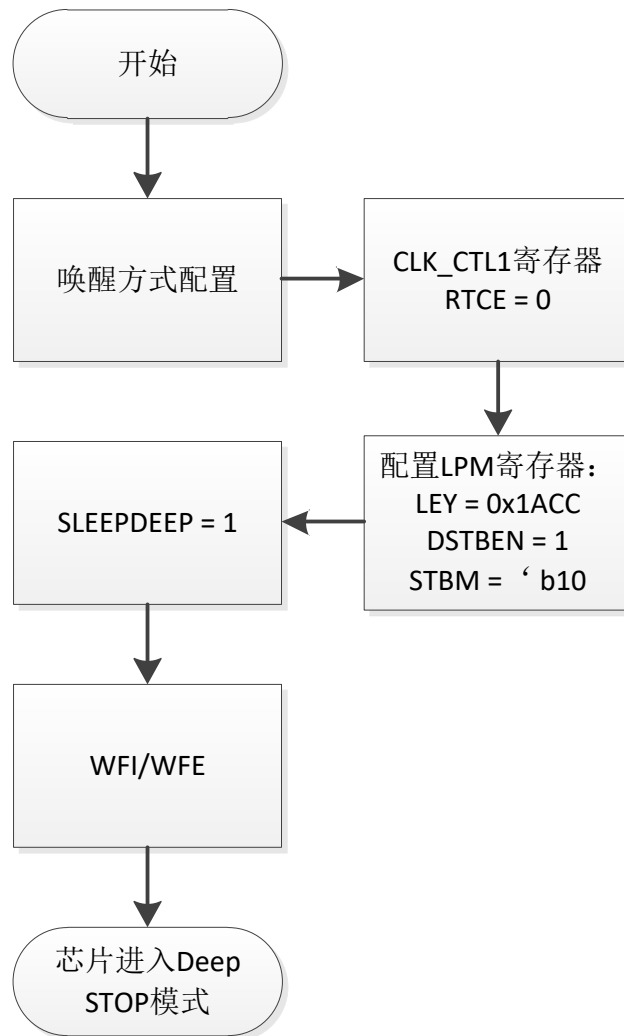


图 6 Deep STOP 模式进入配置流程

4 样例代码

4.1 代码介绍

此模块提供 3 个样例工程分别为：

1) lpm_idle

Idle 模式样例工程

2) lpm_standby

RTC 模式样例工程

3) lpm_deepstandby

Deep RTC 模式样例工程

用户可以根据上述的工作流程编写自己的代码来学习验证该模块，也可以直接通过华大半导体的网站下载样例工程。

本文档以 lpm_standby 工程为例简要介绍该代码的各个部分的功能：

- 1) 系统时钟配置，同时 LSXT 时钟使能，如果 RTC 功能选择使用 LSXT 时钟，此时钟必须使能。

```
stcCfg.enBaseClkDiv = ClkBaseDiv1;  
stcCfg.enAPB1Div = ClkApb1Div1;  
stcCfg.enClkWaitMain = ClkCswtMain1ms;  
stcCfg.enAPB0Div = ClkApb0Div1;  
stcCfg.enClkWaitSub = ClkCswtSub256ms;  
Clk_Init(&stcCfg);  
  
Clk_Enable(ClkLSXT, TRUE);
```

- 2) 调用 RTC 初始化函数，使能 RTC timer 中断功能。

```
/* Configure and start RTC */  
InitRtc();
```

- 3) 在 while(1)循环中，调用进入低功耗模式函数，进入 RTC 模式，等待被 RTC timer 中断唤醒。Timer 中断唤醒后，使能 RTC 半秒中断，半秒中断产生后，重新进入 RTC 模式，等待下一次被 RTC timer 中断唤醒，如此循环。代码中 LED 的操作用于观察程序运行情况。

```
while(1)
{
    /* Enter RTC mode */
    Lpm_GoToStandByMode(StbRtcMode, TRUE);
    if(u8TimerIrqtFlag == TRUE) // wakeup here!, RTC interrupt flag is set in the callback function
    {
        u8TimerIrqtFlag = FALSE;
        /* Turn on LED */
        LED_ON();
        Rtc_ClrIrqtFlag(RtcHalfSecondIrqt);
        while(Rtc_GetIrqtFlag(RtcHalfSecondIrqt) != TRUE); // Wait 0.5s
        /* Turn off LED */
        LED_OFF();
    }
}
```

4) RTC timer 中断处理函数

```
static void SampleRtcTimerCb(void)
{
    // Set timer interruption flag
    u8TimerIrqtFlag = TRUE;
}
```

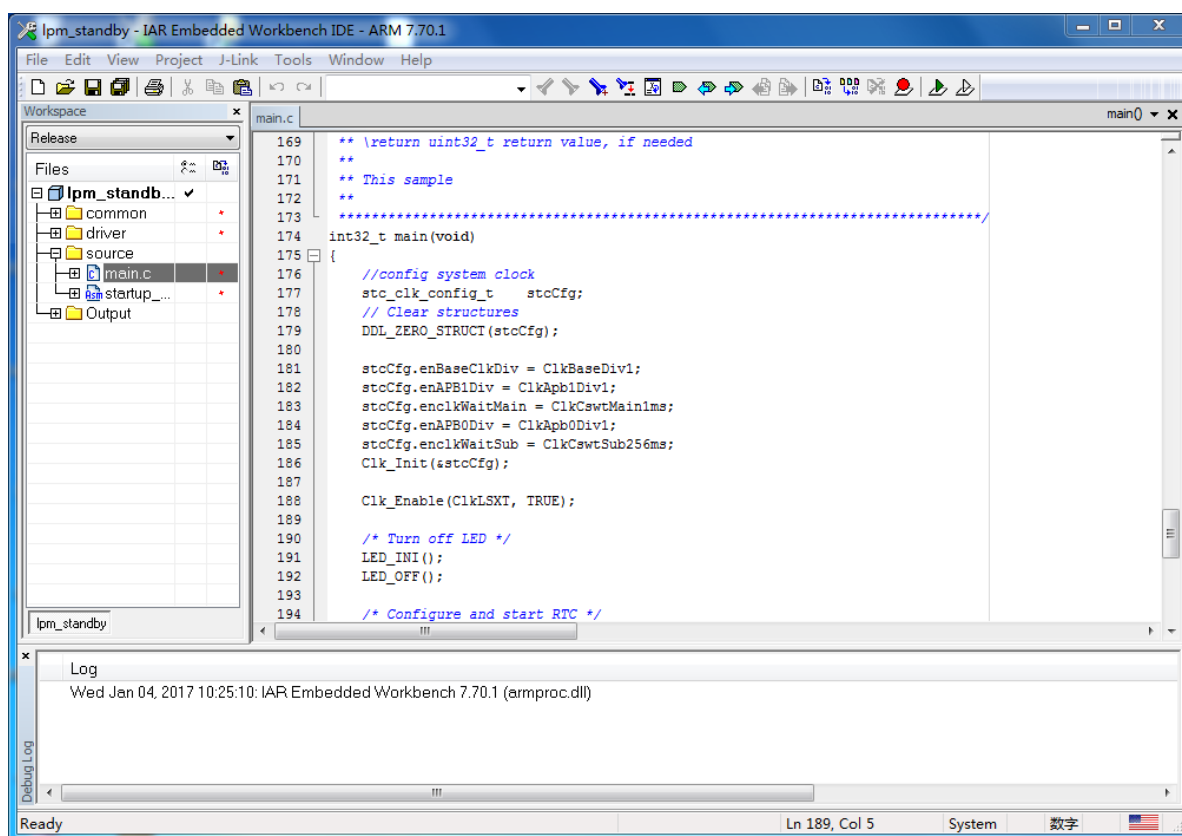

4.2 代码运行


用户可以通过华大半导体的网站下载到低功耗模式的样例代码，建议在最小系统板运行相关代码学习低功耗模块。

以下部分主要介绍如何在最小系统板上运行低功耗模式样例并观察结果：

- 确认安装正确的 IAR 工具（请从华大半导体完整下载相应的安装包并参考用户手册进行安装）。
- 准备好 HC32L15 系列最小系统板。
- 从华大半导体网站下载低功耗模式的 lpm_rtc 样例代码。
- 下载并运行样例代码：

1) 打开 lpm_rtc 样例工程，并打开‘main.c’如下视图：



2) 点击  重新编译整个项目。

3) 点击  将代码下载到 MCU，并使代码全速运行。

- 4) 用示波器观察 LED 对应的 GPIO 得到如图 7 所示波形，GPIO 拉低后，芯片进入 RTC 模式，RTC timer 中断使芯片唤醒，GPIO 拉高，通过半秒中断延时 0.5 秒，再次拉低 GPIO 进入 RTC 模式，等待下一次唤醒。

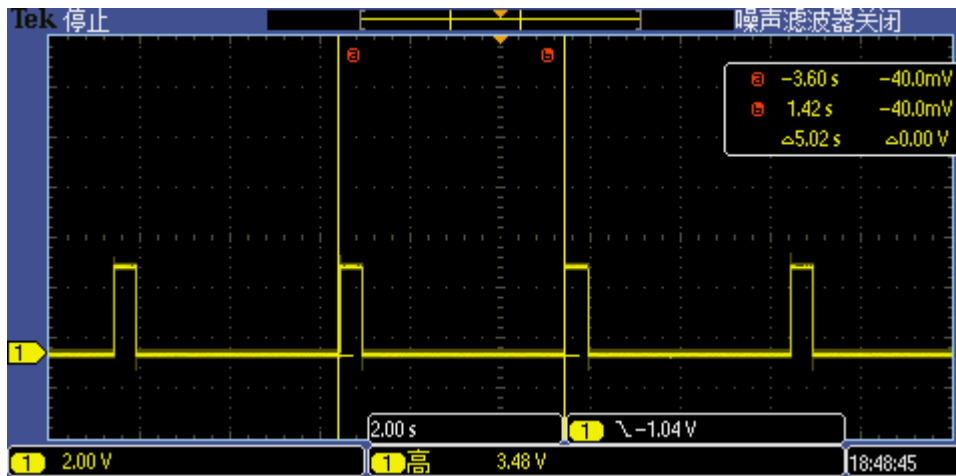


图 7 运行 lpm_rtc 样例 GPIO 波形

运行完毕后可以关闭项目文件。

5 总结

以上章节介绍了 HC32L15 系列芯片的低功耗模式的类型和使用方法，并以 lpm_standby 模式样例代码为例详细说明了芯片进入 RTC 模式流程，以及被 RTC Timer 中断唤醒的效果。其他两个 lpm_idle 模式和 lpm_deepstandby 模式的应用样例，用户可以以相同的方式进行代码下载和运行，详细的运行步骤可以参考样例工程的 readme 文件。

6 版本信息 & 联系方式

日期	版本	修改记录
2018/8/9	Rev1.0	初版发布。



如果您在购买与使用过程中有任何意见或建议，请随时与我们联系。

Email: mcu@hdsc.com.cn

网址: www.hdsc.com.cn

通信地址: 上海市张江高科园区碧波路 572 弄 39 号

邮编: 201203

