

文档编号: AN2034

上海东软载波微电子有限公司

应用笔记

ES32H040x

修订历史

版本	修订日期	修改概要
V1.0	2021-12-01	初版发布
V1.1	2023-05-22	<ol style="list-style-type: none"> 1、修改外部晶振应用要求及注意事项 2、删除电源使用注意事项 3、增加低功耗模式注意事项 4、修改 GPIO 模块注意事项 5、修改 TIMER 模块注意事项 6、增加 RTC 模块注意事项 7、增加 IIC 模块注意事项 8、增加 LPUART 模块注意事项 9、修改 ADC 模块注意事项 10、增加 TK 模块注意事项 11、增加 LCD 模块注意事项 12、增加内部温感使用注意事项
V1.2	2024-01-17	<ol style="list-style-type: none"> 1、在外设 ADC 模块章节增加注意事项 6 2、在外设 GPIO 模块章节增加注意事项 5 和 6 3、在外设 LPUART 模块章节增加注意事项 2
V1.3	2024-07-29	<ol style="list-style-type: none"> 1、删除复位推荐电路 2、修改 GPIO 模块注意事项 3、修改 USART 模块注意事项 4、增加低功耗模式下 IWDG 使用注意事项 5、增加触控应用上电速率要求
V1.4	2025-07-14	<ol style="list-style-type: none"> 1、修改开发环境描述 2、更新 VLCD 管脚应用设计注意事项 3、增加 LCD 模块复用 LED 功能注意事项
V1.5	2026-04-20	<ol style="list-style-type: none"> 1、增加 GPIO 模块的注意事项 7

地 址：中国上海市徐汇区古美路 1515 号凤凰园 12 号楼 3 楼

E-mail: support@essemi.com

电 话：+86-21-60910333

传 真：+86-21-60914991

网 址：<http://www.essemi.com/>

版权所有©

上海东软载波微电子有限公司

本资料内容为上海东软载波微电子有限公司在现有数据资料基础上慎重且力求准确无误编制而成，本资料中所记载的实例以正确的使用方法和标准操作为前提，使用方在应用该等实例时请充分考虑外部诸条件，上海东软载波微电子有限公司不担保或确认该等实例在使用方的适用性、适当性或完整性，上海东软载波微电子有限公司亦不对使用方因使用本资料所有内容而可能或已经带来的风险或后果承担任何法律责任。基于使本资料的内容更加完善等原因，上海东软载波微电子有限公司保留未经预告的修改权。使用方如需获得最新的产品信息，请随时用上述联系方式与上海东软载波微电子有限公司联系

目 录

内容目录

第 1 章	概述	4
1.1	开发环境	4
1.2	库函数选择	4
1.3	寄存器写保护	4
1.3.1	系统写保护	4
1.3.2	RTC 写保护	4
1.3.3	IWDT 写保护	4
1.3.4	WWDT 写保护	5
1.4	写 1 清零寄存器	5
1.5	位带操作	5
第 2 章	系统控制	6
2.1	系统时钟选择	6
2.1.1	内部高速 24MHz (默认时钟)	6
2.1.2	外部时钟 HOSC (4~16MHz)	6
2.1.3	48MHz (使用 HRC 倍频)	6
2.1.4	48MHz (使用 HOSC 倍频)	6
2.1.5	外部低速时钟 (LOSC)	6
2.2	外部晶振	7
2.3	IAP 操作程序	7
2.4	FLASH 读保护	7
2.5	低功耗模式	7
2.6	配置字	8
第 3 章	外设	9
3.1	GPIO 模块	9
3.2	TIMER 模块	9
3.3	RTC 模块	9
3.4	I2C 模块	9
3.5	USART 模块	11
3.6	UART 模块	11
3.7	LPUART 模块	11
3.8	PIS 模块	11
3.9	SPI 模块	12
3.10	ADC 模块	13
3.11	TK 模块	13
3.12	LCD 模块	13
3.13	内部温感	14

第1章 概述

1.1 开发环境

推荐用户使用 Keil5、IAR8.11 或者 iDesigner 进行固件开发。

1.2 库函数选择

ES32 系列芯片提供 2 种类型库函数 ALD 和 MD:

ALD: 提供较为完善的封装, 提供更为人性化的 API, 适合大部分用户;

MD: 基本上只提供寄存器位域级别的“读”、“写”接口, 适合对芯片底层较为熟悉的用户。

如果用户对速度不是要求非常严格, 一般情况下推荐用户使用 ALD 库。可以减少用户学习时间, 增加代码可移植性, 最终缩短用户产品的开发周期。

1.3 寄存器写保护

为避免程序的异常导致运行错误, 芯片写保护寄存器用于阻止对被保护的寄存器误操作。

系统控制单元, GPIO, RTC, WDT 等模块支持寄存器写保护, 对被保护的寄存器进行写之前需要解除写保护状态(允许写), 否则无法对写保护寄存器写入。操作完成后, 再使能写保护(禁止写)。库函数中均提供相应宏定义进行解除保护和使能保护。

1.3.1 系统写保护

系统控制寄存器的访问操作会影响整个芯片的运行状态, 芯片提供系统设置保护寄存器 SYSCFG_PROT。对 SYSCFG_PROT 寄存器以字方式写入 0x55AA6996 会解除写保护, 对该寄存器写入其他任何值都会使能写保护。

可通过读 SYSCFG_PROT 寄存器确认写保护状态, 读出值为 0x1, 表示当前处于写保护状态; 读出值为 0x0 表示当前处于写保护解除状态。

SYSCFG_PROT 保护的寄存器为除 SYSCFG_PROT 寄存器外的 SYSCFG、PMU、CMU、RMU 模块所有寄存器。

1.3.2 RTC 写保护

对 RTC_WPR 寄存器以字方式写入 0x55AAAA55 会解除写保护, 写入其他值使能写保护。

可通过读 RTC_WPR 寄存器确认 RTC 模块是否处于写保护状态, 读出值为 0x1, 表示当前处于写保护状态; 读出值为 0x0 表示 RTC 模块处于写保护解除状态。

该寄存器保护除自身外的 RTC 所有寄存器。

1.3.3 IWDT 写保护

对 IWDT_LOCK 寄存器以字方式写入 0x1ACCE551 会解除写保护, 写入其他值使能写保护。

可通过读 IWDT_LOCK 寄存器确认 IWDT 模块是否处于写保护状态, 读出值为 0x1, 表示当前处于写保护状态; 读出值为 0x0 表示 IWDT 模块处于写保护解除状态。

该寄存器保护除自身外的 IWDT 所有寄存器。

执行 CLWDT 至少需要间隔 3 个时钟才能生效，在执行 CLWDT 后需要延时 128uS 后，再操作 WDT 或休眠指令。

1.3.4 WWDT 写保护

对 WWDT_LOCK 寄存器以字方式写入 0x1ACCE551 会解除写保护，写入其他值使能写保护。

可通过读 WWDT_LOCK 寄存器确认 WWDT 模块是否处于写保护状态，读出值为 0x1，表示当前处于写保护状态；读出值为 0x0 表示 WWDT 模块处于写保护解除状态。

该寄存器保护除自身外的 WWDT 所有寄存器。

1.4 写 1 清零寄存器

中断标志寄存器都是用“写 1 清零”的方式来操作。对于“写 1 清零”的寄存器，不可使用“读-修改-写”的方式来进行“写 1 清零”，否则会引起标志误清，进而产生漏中断的后果。对该类寄存器操作需要以字方式进行写。

例：清除 DMA 模块通道 0 的中断标志：

正确写法：DMAx->ICFR = 1;

错误写法：DMAx->ICFR |= 1;

1.5 位带操作

位带扩展区把每个 bit 扩展为一个 32-bits 的字，通过访问这些字可达到访问原始 bit 的目的；某个 bit 所在字的地址为 A，位序号为 N (0≤N≤31)，则该 bit 位带扩展后的地址为：

SRAM: AliasAddr=0x22000000+(A-0x20000000)x32+Nx4

外设: AliasAddr=0x42000000+(A-0x40000000)x32+Nx4

库函数中提供位带操作 API:

RAM 位带: void BITBAND_SRAM(uint32_t *addr, uint32_t bit, uint32_t val);

外设位带: void BITBAND_PER(volatile uint32_t *addr, uint32_t bit, uint32_t val);

第2章 系统控制

2.1 系统时钟选择

系统上电默认使用内部 24MHz 高速时钟（HRC）作为系统时钟。

若系统运行 48MHz/32MHz 主频，建议将 APB2 总线进行 2 分频，否则可能会导致低速外设运行不正常，分频操作建议使用 ALD 库中 API：`ald_cmu_div_config(CMU_PCLK_2, CMU_DIV_2)`;

几种常用系统时钟配置：

2.1.1 内部高速 24MHz（默认时钟）

此种系统时钟不需要用户做任何配置。

2.1.2 外部时钟 HOSC（4~16MHz）

外部高速时钟要求为 4MHz 的倍数，如：4MHz、8MHz、12MHz、16MHz。

首先要确认焊接了外部高速时钟，并已知外部高速时钟的频率，假如外部高速时钟为 12MHz，则配置方式如下：

```
ald_cmu_clock_config(CMU_CLOCK_HOSC, 12000000);
```

2.1.3 48MHz（使用 HRC 倍频）

配置方式如下：

```
ald_cmu_pll1_config(CMU_PLL1_INPUT_HRC_6, CMU_PLL1_OUTPUT_48M);
```

```
ald_cmu_clock_config(CMU_CLOCK_PLL1, 48000000);
```

2.1.4 48MHz（使用 HOSC 倍频）

外部高速时钟要求为 4MHz 的倍数，如：4MHz、8MHz、12MHz、16MHz。

首先要确认焊接了外部高速时钟，并已知外部高速时钟的频率，假如外部高速时钟为 12MHz，则配置方式如下：

```
ald_cmu_pll1_config(CMU_PLL1_INPUT_HOSC_3, CMU_PLL1_OUTPUT_48M);
```

```
ald_cmu_clock_config(CMU_CLOCK_PLL1, 48000000);
```

2.1.5 外部低速时钟（LOSC）

首先要确认焊接了外部低速时钟，配置方式如下：

```
ald_cmu_clock_config(CMU_CLOCK_LOSC, 32768);
```

需要注意的是，当系统时钟配置为低速时钟时（低于 1MHz），SysTick 中断将会被迫关闭。ALD 提供的延迟类函数禁止使用。

2.2 外部晶振

外部高速振荡器的典型应用连接：

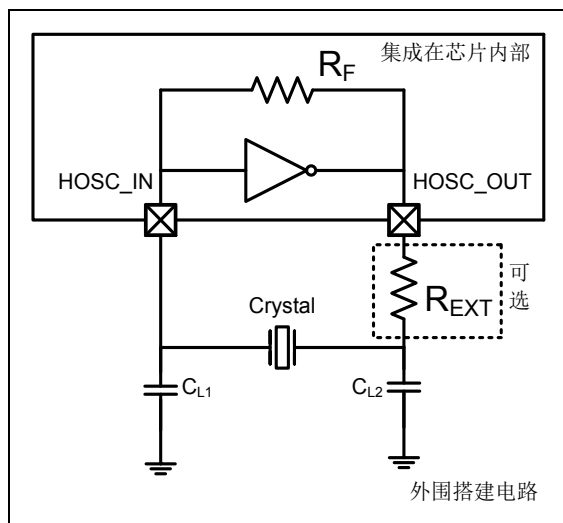


图 2-1 外部高速振荡器连接图

注 1: R_{EXT} 为可选电阻，其阻值取决于具体晶振规格特性

注 2: 当晶振频率为 1~8M，要求 $ESR \leq 150 \Omega$ ，且 C_{L1} 、 C_{L2} 容值需小于 15pF

注 3: 当晶振频率为 8~16M，要求 $ESR \leq 40 \Omega$ ，且 C_{L1} 、 C_{L2} 容值需小于 12pF

2.3 IAP 操作程序

芯片内置 IAP 自编程固化模块，由硬件电路实现。推荐使用 IAP 方式对 FLASH 进行擦、写操作，可以减少用户代码量。

2.4 FLASH 读保护

规则描述：当 FLASH 的读保护级别设置为 level1 或 level2 时，运行在 SRAM 上的程序不能有读 FLASH 操作。

典型应用 1：运行在 SRAM 上的程序想读取 FLASH。将读 FLASH 操作放在 FLASH 上执行；

典型应用 2：程序运行在 SRAM 上，响应中断请求。将中断向量表拷贝至 SRAM 中，并设置中断向量偏移地址（SYSCFG_VTOR 和 SYSCFG_MRMP.VTOEN）。

2.5 低功耗模式

注意事项 1：进入 STOP1/STOP2 模式之前，置位 CR.LPSTOP 位可降低 STOP 功耗。

注意事项 2：配置寄存器 PMU_BKPCR0、PMU_BKPCR1、PMU_BKPSR 时除了解锁外还需将 bit15~bit8 配置为 0x5A 并同其他配置一起写入寄存器才可生效。

注意事项 3：根据具体应用注意将外设时钟低功耗模式使能（CMU_LPENR）。

注意事项 4：在进入低功耗模式前需关闭 SysTick，否则可能导致 MCU 无法被唤醒。具体操作方法可参考 ALD 库 STOP 及 STANDBY 模式的配置方式：

```
SysTick->CTRL &= ~SysTick_CTRL_ENABLE_Msk;
```

```
__WFI();
```

```
SysTick->CTRL |= SysTick_CTRL_ENABLE_Msk;
```

注意事项 5: 若要使用 IWDG 进行唤醒并复位，须保证 FLASH 低功耗暂停（FSTOP）和掉电（FPD）未使能。

2.6 配置字

注意事项 1: 使用 ELink II/ELink II mini 修改芯片配置字之后，芯片需要断电后重新上电，才能正常调试。

注意事项 2: 使用 ESBurner 擦除芯片后，需要执行配置字编程，否则将导致芯片运行不正常，即点击“擦除”按钮后，需要点击“配编”按钮。

第3章 外设

3.1 GPIO 模块

注意事项 1: 未使用的 GPIO 管脚建议设置为输出固定电平并悬空，若设置为输入，须加上拉或下拉电阻接到电源或地。

注意事项 2: 因 4 线 SWD 做调试接口时不能被占用，故烧写引脚复用为 GPIO 功能后，keil 里面不能下载程序，可使用 5 线制 SWD 并在 keil 中使用 under Reset 功能。或者使用我司 ESlinkII 工具用 ESburner 软件进行下载。

注意事项 3: 输出口若需要上下电钳位，请选用 1K--10K 电阻下拉至 GND。

注意事项 4: 芯片上电后在载入芯片配置字之前 PC14、PC15 为 LOSC 功能，有短暂输出驱动能力，应用时避免作为关键控制脚使用。

注意事项 5: 应用时注意避免出现端口过压或负压的情况。

注意事项 6: 在上下电过程中端口可能存在弱驱动的脉冲输出，对地挂接 10K 以下电阻可有效避免脉冲。

注意事项 7: 针对 TK 的应用方案，PA0/PA1/PA2/PB6 优先配置为 TK 通道，如配置为其他功能，应避免 IO 持续翻转。

3.2 TIMER 模块

注意事项 1: 应用 PWM 模式时若需改变周期或占空比，建议使能输出比较预载，否则输出 PWM 波形可能不连续。

注意事项 2: PWM 模式未输出时通道的空闲电平会受到上下拉控制影响（上拉为高电平，下拉为低电平）。

3.3 RTC 模块

注意事项 1: 设置时间和日期时需对寄存器连续写入两次，该项应用注意已在库中做了处理。

注意事项 2: 在 standby 模式下使用 RTC，需要配置字 LOSCEN 配置为强制使能。在 stop2 模式下使用 RTC，需要配置 CMU->CLKENR.LOSCEN=1 同时 CMU->LPENR.LOSCEN=1，或者配置字 LOSCEN 配置为强制使能。否则，在低功耗阶段，LOSC 为禁止。

3.4 I2C 模块

注意事项 1: I2C 工作在主机模式时，若从机一直拉低时钟线，主机可以通过关闭 I2C 模块来释放时钟线。

注意事项 2: I2C 作为主机与外部 EEPROM 进行通信，若 SDA 线一直被 EEPROM 器件拉低，可以通过以下 2 种方式解除死锁现象：

1. 通过重新给 EEPROM 上电解除死锁
2. 主机 SCL 线模拟输出 9 个时钟信号后，再模拟发送一个停止信号，EEPROM 会终止

此次通信并释放 SDA 线。

注意事项 3: I2C 工作在从机模式时，会响应 stop 命令的中断，需在地址匹配标志置起后再使能 stop 中断，否则会误响应其他地址的 stop 命令。

注意事项 4: I2C 工作在从机模式时，主机发送读取命令之前写入发送 buffer 的值无效。

注意事项 5: I2C 工作在主机模式，若发送数据后从机返回的是 NACK，默认会重发当前字节，直至收到 ACK 为止。如果应用时无需此种处理方式，建议使用 GPIO 软件模拟 I2C 主机。

3.5 USART 模块

注意事项 1: IDLE 中断需要在接收到一个数据之后再打开，否则该中断会立即触发，并且中断标志不能被清除。

注意事项 2: 使用 IRDA 模式特别是低波特率通讯时，注意超时时间配置不可过短。

3.6 UART 模块

注意事项 1: 在 LIN 模式下，SR.BF 标志无效，可通过 RIF.LINBKIF 标志判断总线是否出现断开符。

注意事项 2: TC 中断在每发送完一个字节后均被置起，而不是在数据帧发送完成后被置起，所以不能通过 RIF.TCIF 标志判断一帧是否发生完成。可通过联合判断 SR.TEM 标志和 RIF.TCIF 标志，来确定一帧数据是否发送完成。

注意事项 3: 配合 DMA 使用时，DMA 的 burst 必须为 1，R_power 必须为 0。

注意事项 4: 在中断模式下，发送时若向 TX_FIFO 中写入多个数据，则每次写数据前需确认 FIFO 是否满，避免 FIFO 上溢。接收时需要在接收中断函数中将 RX_FIFO 读空，否则会导致后续再收到数据也进不了 RX 中断。该项应用注意在 ALD 库中已经做了处理，使用 MD 库时需自行处理。

注意事项 5: 当发生溢出错误时，需要将 RX_FIFO 进行复位。该项应用注意在 ALD 库中已经做了处理，使用 MD 库时需自行处理。

注意事项 6: 使用中断方式发送数据时，需要按照如下顺序进行：先清除 TXS 中断，再向 TBR 寄存器写入数据。该项应用注意在 ALD 库中已经做了处理，使用 MD 库时需自行处理。

注意事项 7: 数据接收时最大波特率误差需在 $\pm 3\%$ 内。

注意事项 8: 不支持硬件流控模式下的单线半双工通信功能。

注意事项 9: 硬件流控模式下，若全双工模式下数据接收错误，则在轮询接收时，不判断 UART_STAT 寄存器的 RFTH，改判断 UART_RIF 寄存器的 RFTH。

3.7 LPUART 模块

注意事项 1: LPUART 主要应用于使用低速时钟（如 LRC）的情况，若使用其他高速时钟（如 HRC）作为时钟源，请避免轮询标志的使用方式，且需确认 CLKDIV 配置是否正确，若不正确请直接对 CLKDIV 赋值。

注意事项 2: 接收时需要在接收中断函数中将 RX_FIFO 读空，否则会导致后续再收到数据也进不了 RX 中断，具体可参考 LPUART 模块的 ALD 例程

3.8 PIS 模块

注意事项 1: PIS 模块不支持 TIMER 的触发事件，控制另外一个 TIMER。

3.9 SPI 模块

注意事项 1: SPI 在只读模式下 (CON1.RXO 置位), STAT.BUSY 标志会一直被置起, 故不能通过该位判断通信是否完成。若需要清除 STAT.BUSY 标志, 需要将 CON1.RXO 清零。

注意事项 2: 不支持 DMA 模式下的 CRC 功能。

3. 10 ADC 模块

注意事项 1: 若发现在输入某一电压后数据转换的结果有较大波动，在对采样结果做处理时请使用采样多次去最大最小后取平均的方式，推荐使用 ALD 库中的数据处理方式。

注意事项 2: 使用外部输入做参考的情况下也需要将内部参考电压 VREFEN 使能。

注意事项 3: 为应对测量信号驱动能力弱的情况，在应用函数例程中初始化使能 TRMEN、IREFEN、VCMBUFEN、VRBUFEN。

注意事项 4: 为增强输入电平信号稳定性可根据需要在检测口增加一滤波电容。

注意事项 5: 正向参考电压选择 VREFP 管脚时需选择经过 Buffer。

注意事项 6: IO 引脚的灌拉电流对其他引脚电压有拉高低影响，会影响 ADC 采样结果，每灌/拉 1mA 电流最大约拉高/低 2mV。

3. 11 TK 模块

注意事项 1: 充放电频率不宜过高否则充放电不完全，不建议使用高于 4M 的充放电频率。

注意事项 2: 使用 TK 功能时普通 IO 上的电压不可高于 VDD，否则将影响 TK 结果数据。

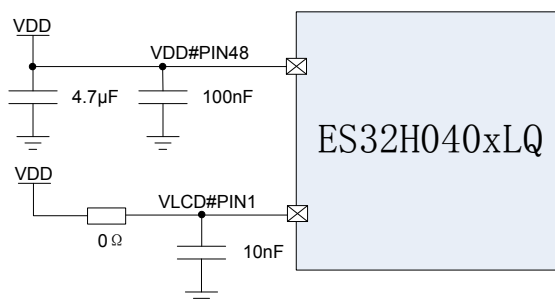
注意事项 3: TK 推荐的工作电压范围为 2.7V~5.5V，在电压范围外的采样数据波动较大。

注意事项 4: 针对 tk_ES32H0403_V1.1 版本的触控库，要求上电速率需快于 7.5V/S。

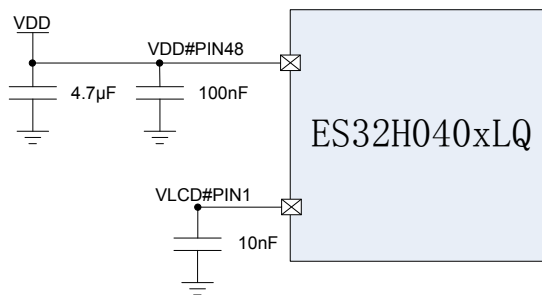
3. 12 LCD 模块

注意事项 1: VLCD 管脚的应用电路

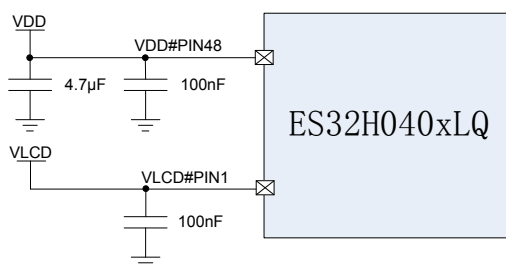
1.未使用内部升压泵的情况下，VLCD 管脚挂接 10nF 电容到 VSS，并串 0Ω 电阻到 VDD。



2.使用内部升压泵的情况下，VLCD 管脚挂接 10nF 电容到 VSS，且与 VDD 不连接。



3.LCD 使用外部供电时，VLCD 管脚挂接 100nF 去耦电容并接到外部供电电压源。



注意事项 2: 高驱动模式电压驱动器必须使能，驱动电流需配置为最大，驱动电阻需配置为最小。

注意事项 3: 如 LCD 电源选择 charge pump 供电，LCD 时钟需选 1MHz，并且以下 IO 不支持复用为 SEG 口：PC0 (SEG22)，PC1 (SEG23)，PB10 (SEG10)，PB11 (SEG11)，PB12 (SEG12)，PB8 (SEG21)。

注意事项 4: 如 LCD 电源选择 VLCD 且 VLCD 电压高于 VDD，以下 IO 不支持复用为 SEG 口：PC0 (SEG22)，PC1 (SEG23)，PB10 (SEG10)，PB11 (SEG11)，PB12 (SEG12)，PB8 (SEG21)。

注意事项 5: 复用为 LED 功能时，不可驱动共阳型的数码管，且最大支持 16 个显示段 (SEG0~15) 和 8 个公共端 (COM0~7)

3.13 内部温感

注意事项 1: 使用 ADC 采样内部温感通道时，正向参考电压需选择内部 2V。推荐使用 ALD 库中的使用方式。