

华中科技大学

电子与信息工程系

TI 文档

姓名	熊绪胜	学号	U200913668
姓名	熊倪	学号	U200913783
姓名	项明	学号	U200913845

指导老师：汪小燕老师

1. 背景介绍

1.1 MSP430 介绍

MSP430 系列 MCU 是 Texas Instruments (TI) 公司生产的 16bit RISC 超低功耗混合信号处理器，能为各种低功耗和便携式应用提供最终解决方案。作为混合信号和数字技术的领导者，TI 创新生产的 MSP430 使系统设计人员能够在保持独一无二的低功率的同时同步连接至模拟信号、传感器和数字组件，其典型应用包括实用计量、便携式仪表、智能传感和消费类电子产品。

MSP430 可提供 200 多种超低功耗微处理器器件。每个器件都具有灵活的时钟系统，启用了多达 7 种低功率模式 (LPM)，可提高优化性能。如果配以低于 1 μ s 的即时唤醒时间以及各种中断源，MSP430 可确保您的应用仅使用手动执行任务时所需的相应时钟和外设。

其主要的超低功耗度量标准：

- 各种低功耗工作模式
 - 超低功耗工作模式：最低 120 μ A/MHz@2.2V
 - 待机模式，具有自我唤醒功能、RAM 保持模式 (LPM3)：最低 0.7 μ A@2.2V
 - 待机模式，具有自我唤醒功能 (LPM4)：最低低于 100nA@2.2V
 - 停机模式，具有 RAM 保持模式 (LPM3.5)：最低低于 100nA@2.2V
- 低功率模式下低于 1 μ s 的即时唤醒时间
- 始终接通的零功耗掉电复位

MSP430 采用冯诺依曼架构，通过通用存储器地址总线 (MAB) 和存储器数据总线 (MDB) 将 16 位 RISC CPU、多种外设和灵活的时钟系统进行完美结合。MSP430 通过将先进的 CPU 与模块化内存映像模数外设相结合，为当今和未来的混合信号应用提供了解决方案。MSP430 平台内包括五代超低功耗、高度集成的微处理器产品，涵盖了 200 多款器件。每一代产品都提供各种级别的模拟集成、数字外设和通信协议，以帮助开发者查找用于各种应用的合适的微处理器。

1.2 MSP430F2619 介绍

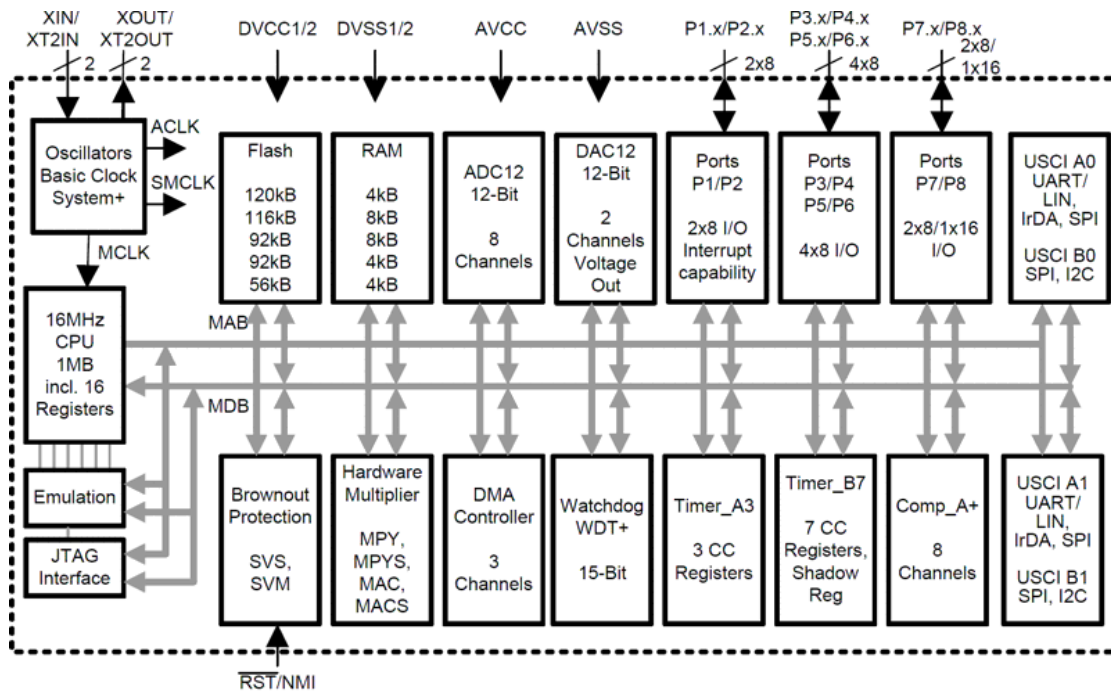
本文档通过演示基于 MSP430F2619 开发板的 Flash 操作案例，让读者了解使用 CCS v5.1 进行 MSP430 上的数据存储和传输特性。

MSPF2619 简介：

- 低工作电压范围：1.8V—3.6V
- 超低功耗
 - Active 模式：365 μ A@1MHz, 2.2V
 - 待机模式 (VLO)：0.5 μ A
 - 掉电模式 (RAM 保持)：0.1 μ A
- 从待机模式唤醒时间小于 1 μ s
- 16bit RISC 架构，62.5ns 指令周期
- 3 通道内部 DMA
- 12bit A/D 转换器，包含内部参照、采样保持、自扫描特性
- 双通道异步 D/A 转换器

- 片上比较器
- 4 个通用串行通讯接口 (USCI)
 - USCI_A0 和 USCI_A1: 增强型 UART, 支持波特率自动检测, 红外通信编解码器, 同步 SPI
 - USCI_B0 和 USCI_B1: I2C, 同步 SPI
- 可编程检测的供电监视器
- 掉电检测器
- 引导程序 (BSL)
- 串行板上编程, 无需外部编程电源, 保险丝实现代码保护
- 120KB+256B Flash, 4KB RAM

MSPF2619 功能方框图:



Note: Memory sizes, available peripherals, and ports may vary depending on the device.

1.3 MSP-FET430UIF 简介



MSP-FET430UIF 是一个功能强大用于快速开发 MSP430 MCU 应用的 flash 仿真工具。它包含 USB 调试接口, 通过连接板上的 JTAG 接口或保存了 Spy Bi-Wire (2-JTAG) 协议的引脚以对 MSP430 进行编程和调试。仅需几次点击便能在数秒内对 Flash 进行擦除和编程, 由于 MSP430 flash 是超低功耗的, 无需外部电源供电。

MSP-FET430UIF 支持所有 MSP430 Flash 设备。USB 调试接口连接基于 Flash 的 MSP430 MCU 和 PC 以实时, 在线地进行编程和调试。

其技术特性如下:

- 软件可配置的供电电压：1.8V—3.6V@100mA
- 支持 JTAG 保险丝保护代码
- 支持所有有 JTAG 接头的 MSP430 开发板
- 支持 JTAG 和 Spy-Bi-Wire (2-Wire JTAG) 协议

1.5 Code Composer Studio (CCS) v5.1 简介

Code Composer Studio (CCStudio) 适用于德州仪器 (TI) 嵌入式处理器系列的集成开发环境 (IDE)，包含一整套用于开发和调试嵌入式应用的工具。它包含适用于每个 TI 器件系列的编译器、源码编辑器、项目构建环境、调试器、描述器、仿真器、实时操作系统以及多种其他功能。直观的 IDE 提供了单个用户界面，可帮助您完成应用开发流程的每个步骤。借助于精密的高效工具，用户能够利用熟悉的工具和界面快速上手并将功能添加至他们的应用。Code Composer Studio 以 Eclipse 开源软件框架为基础。Eclipse 软件框架最初作为创建开发工具的开发框架而被开发。Eclipse 为构建软件开发环境提供了出色的软件框架，并且逐渐成为备受众多嵌入式软件供应商青睐的标准框架。CCStudio 将 Eclipse 软件框架的优点和 TI 先进的嵌入式调试功能相结合，为嵌入式开发人员提供了一个引人注目、功能丰富的开发环境。为帮助更多新手开发者学习，CCS 提供了多项许可选项：

- 评估：免费的有限许可可用于评估 TI 工具和器件
- 节点锁定：许可颁发至特定计算机
- 浮动：许可可在多个计算机之间共享
- 代码大小限制：MSP430 具有免费 16KB 代码大小有限许可
- 捆绑包/开发套件：免费许可可与 EVM 和开发板（具有板载仿真）以及 XDS100 类仿真器一同使用
- 大学：联系 TI 大学计划相关人员以了解详细信息

2. 基于 MSP430F2619 的 flash 操作实验

2.1 实验模块原理

MSP430F2619 有 120KB+256B 的 flash 空间，256B 为信息段组成。120KB 为程序存储段，由块和段组成，其中一块为 64B，一段为 512B。对 flash 写入数据，必须先擦除相应段，擦除可以分为段擦除、主程序存储区擦除、全部擦除，在擦除开始时先要在首地址除执行一次空写操作，启动擦除操作；Flash 编程可以有两种方式：单字节或块写入。块写程序段必须放在 RAM 区，实验中使用段擦除和字节写模式将 C 段数据拷贝到 D 段中。

寄存器配置如下表：

寄存器	功能	说明
FCTL1	编程或擦除控制	
FCTL2	Flash 时序控制	Flash 时钟操作范围 257kHz—476kHz
FCTL3	Flash 空间锁定或解锁控制位	LOCKA 锁定信息段 A 不被擦除

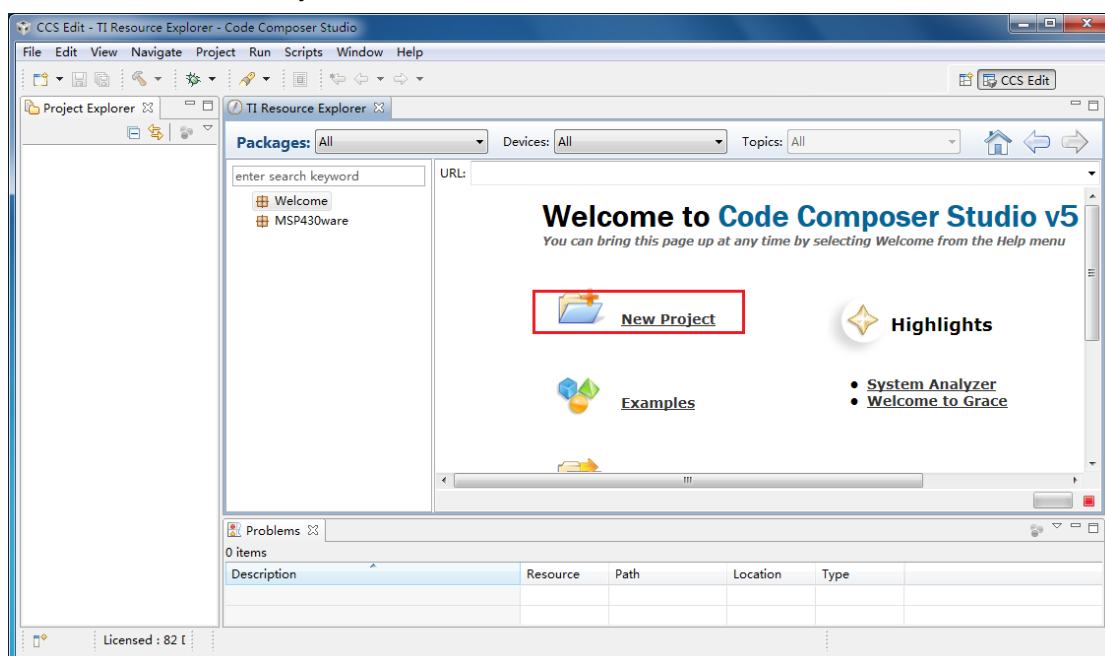
在对 FCTLX 寄存器进行操作时首先要写入安全值 FWKEYX(0x5A)，如果写错进入非屏蔽中断。

信息段 A 保存有 DCO 校准信息，擦除后数据将不可恢复，因此在擦除时要小心信息段 A 不被擦除。LOCKA 标志位决定在擦除时信息段是否被擦除，如果为 1，信息段 A 将被保护，如果为 0，在擦除时信息段也将被擦除，在使用时要小心。例如下面程序中将 flash 配置为字节写模式。

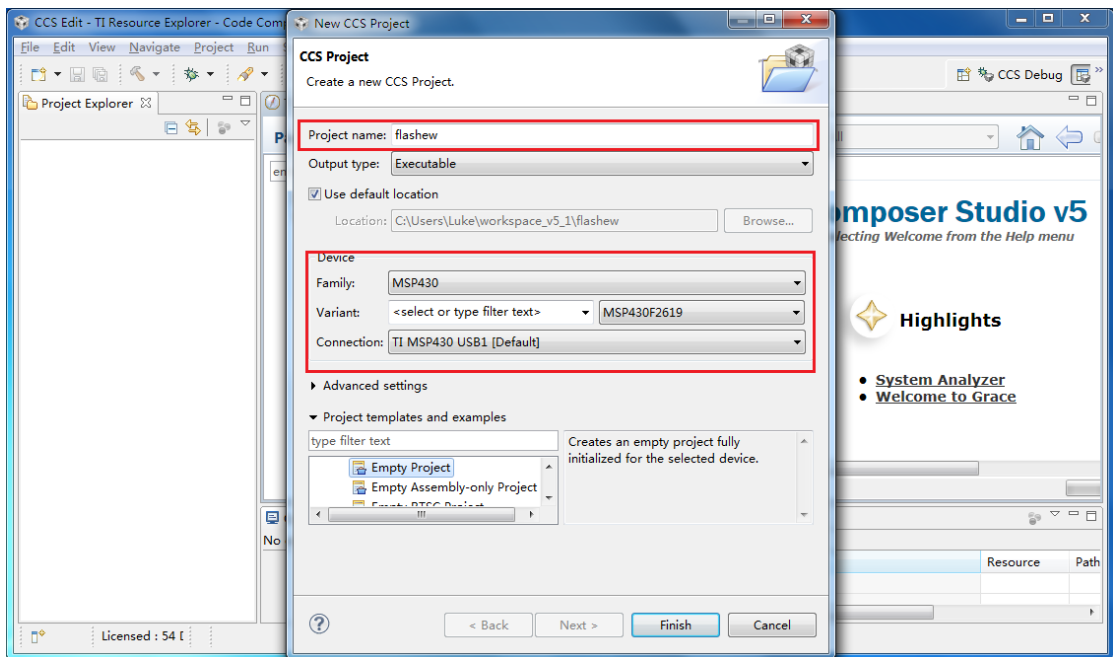
```
FCTL3 = FWKEY; //写入安全值并解锁
FCTL1 = FWKEY + ERASE; //段擦除功能
*Flash_ptr = 0; //空写操作，开始擦除
FCTL1 = FWKEY + WRT; //字节写功能
```

2.2 实验步骤及结果

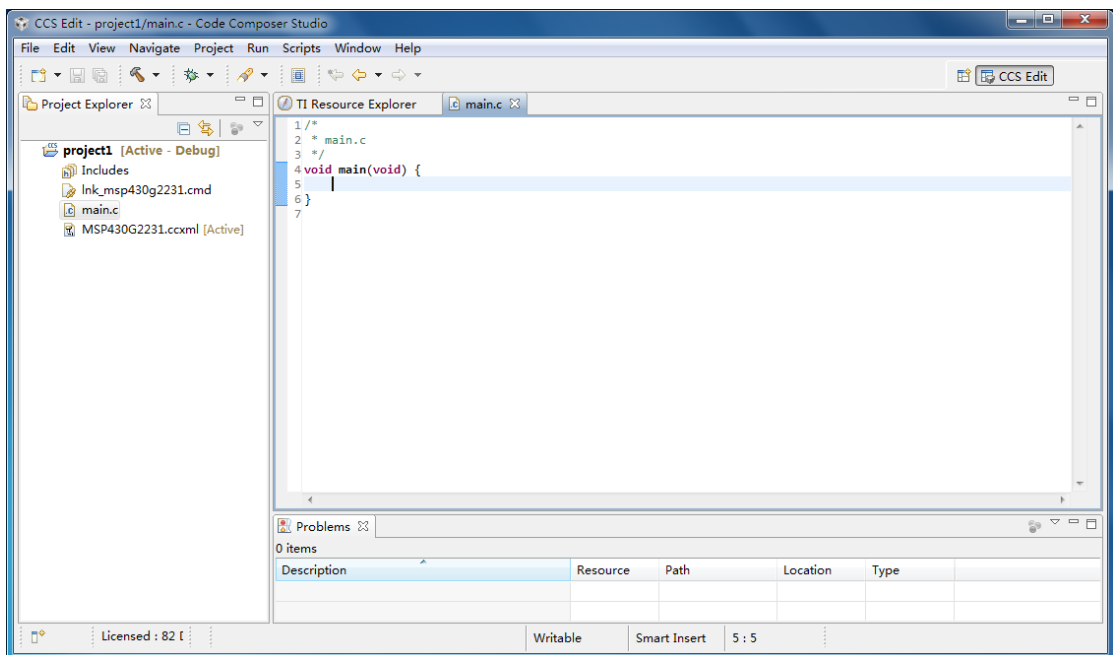
1. 运行 Code Composer Studio v5，新建一个 CCS 工程，点击导航界面中的 New Project，也可点击 File/New/CCS Project 以新建，抑或添加已存在的工程。

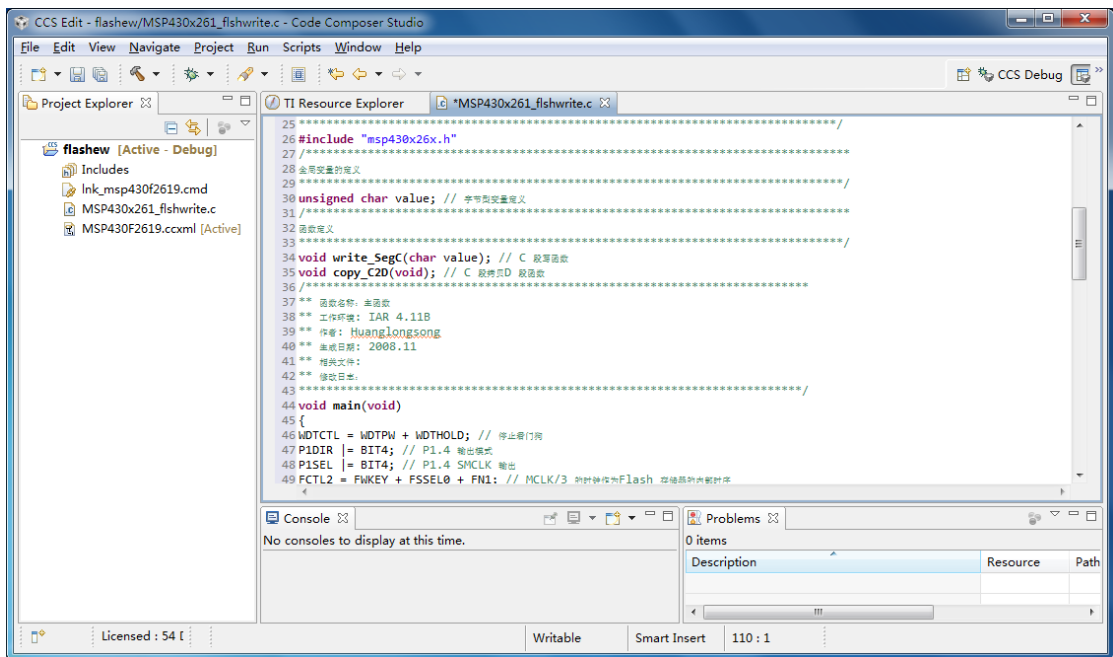


2. 在新建工程导航界面里输入工程名，保存目录，选择相应的 MSP430 目标装置型号，在 Connection 中选择 TI MSP430 USB1[default]，以确保调试器有效连接到 MSP430 MCU，点击 Finish。

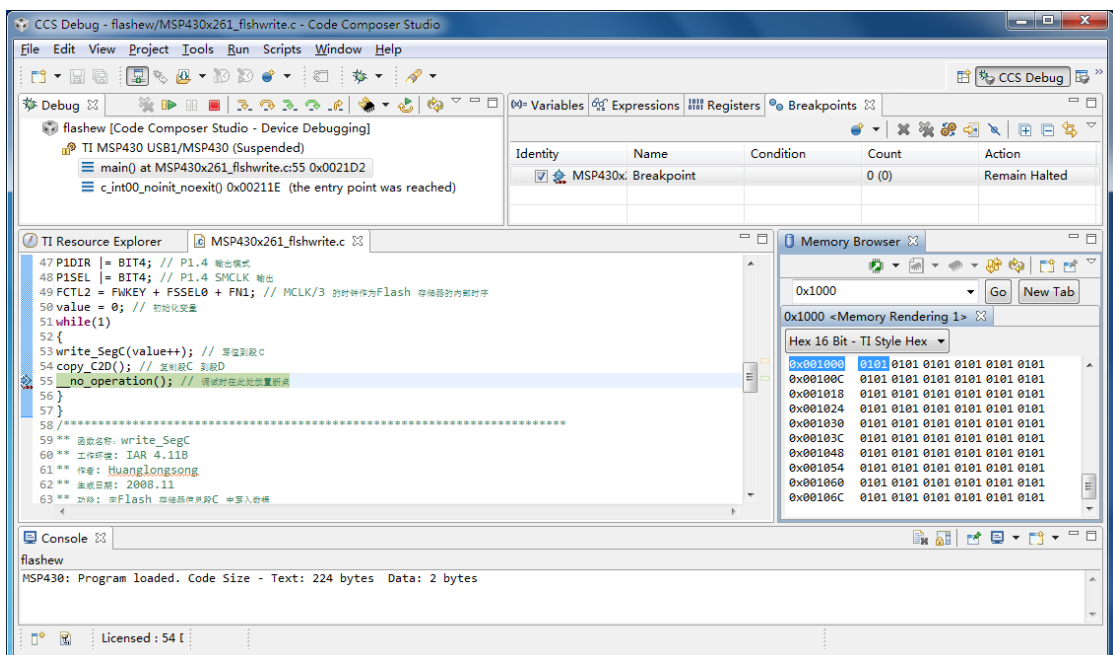


3. 新建完成，工程自动新建 main.c 并在主界面中打开，在这里输入程序代码，这里使用的是别人已经调试好的代码，只需复制进 main.c 文件里去，并重新命名为 MSP430x261_flashwrite.c





4. 在 Project Explorer 面板中点击选中刚才新建的工程，右键选择 Properties，在弹出框中核查工程的各个属性，确保目标 MCU 型号正确选择，Connection 选项为 TI MSP430 USB1[default]，然后点击 Debug/MSP430 properties，在 Download 选项中有四种选择，这里选择 Erase main and information memory，这样在 Debug 时，目标 MCU 中的 flash 会被清空并写入新的代码和数据。
5. 点击 Run/Debug (F11)，程序将被编译、连接，然后下载到目标板上的 flash 中，通过 CCSv5.1 的实时软件调试功能可以观察到程序运行过程中 flash 上的数据情况，为方便观察，在代码中间__no_operation();处设了一个断点，打开 Memory Browser，观察 0x1000 处的数据情况：



可以在 Memory Browser 窗口中观察到 0x1040 和 0x1000 处的数据相同，这与实验的预计是相同的。多次恢复运行后也是同样的情况，可以判断程序正确执行，即完成向 0x1000

后的 64B 空间写入数据并将其复制到 0x1040 后的地址中去。