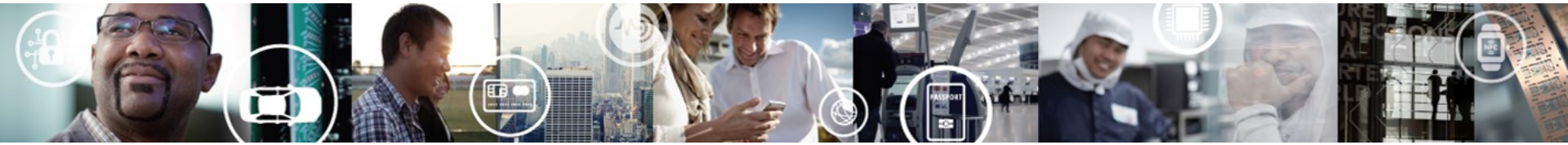
# LPC82X 培训资料

## 存储器及读写保护

MAY, 2016





SECURE CONNECTIONS
FOR A SMARTER WORLD

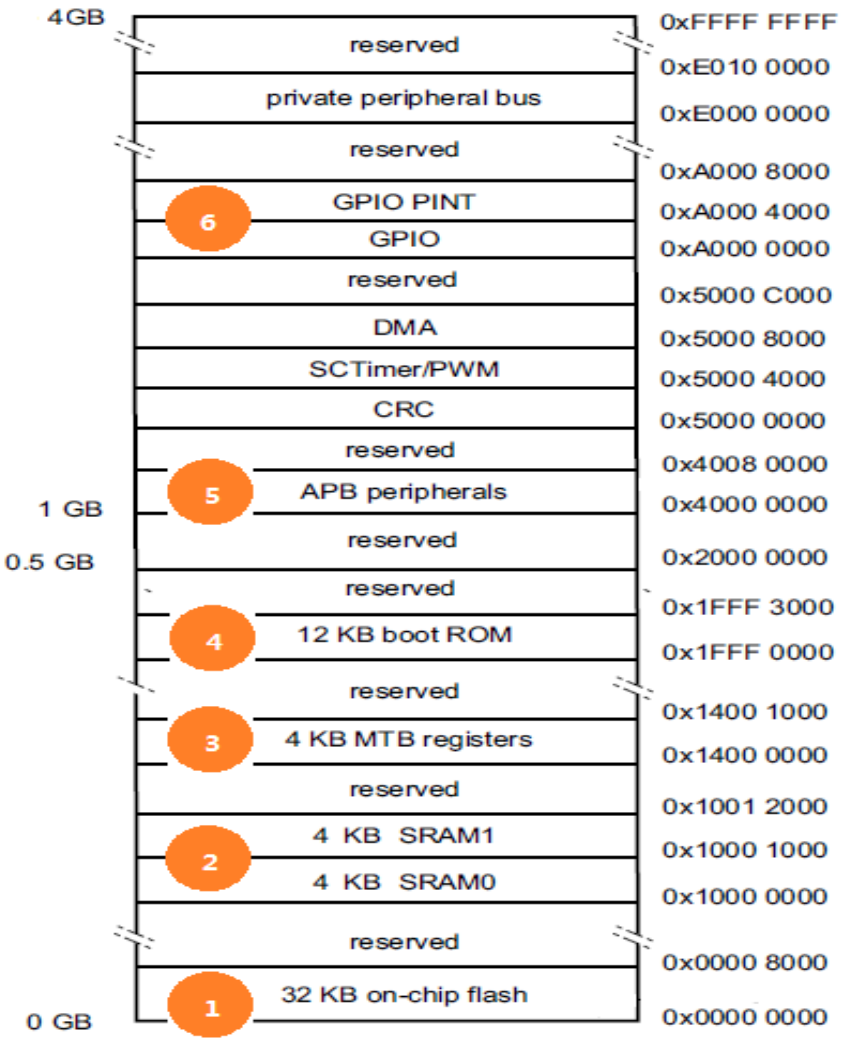
# 内容

- 存储器映射（Memory Map）
- 片上 FLASH 控制器
- ISP 和 IAP
- ROM驱动 ( ROM Driver)

# 存储器映射
# （MEMORY MAP）

# 存储器映射

- FLASH
  - LPC824 (32K 字节)
  - LPC822 (16K 字节)
  - 64 Bytes 擦写
- SRAM
  - LPC824 (8K 字节)
  - LPC822 (4K 字节)
- 调试记录缓存（MTB）
- 只读存储器（ROM）
  - Rom driver
  - ISP/IAP
- 异步外设总线（APB）
  - 定时器（Timer）
  - 模拟外设（模数转换，比较器）
  - 异步传输（SPI, I2C, UART）
- 通用IO口和IO中断



# 片上 FLASH 控制器

# 闪存（FLASH）控制器-1

- 控制闪存（FLASH）访问时间
- 当ARM时钟速度调整前，需要调整闪存访问时间
  - ARM 速度20MHz以上，闪存访问需要2个系统时钟
  - ARM 速度在20MHz及以下，闪存访问需要1个系统时钟
- 实例代码

```
typedef enum {
    /*!< 1 CPU clocks Flash accesses for up to 20 MHz CPU */
   FLASHTIM_20MHZ_CPU = 0,
    /*!< 2 CPU clocks Flash accesses for up to 30 MHz CPU */
   FLASHTIM_30MHZ_CPU = 1,
} FMC_FLASHTIM_T;
void Chip_FMC_SetFLASHAccess(FMC_FLASHTIM_T clks);
```

- 注意，不要在低功耗模式下调整闪存访问时间

# 闪存（FLASH）控制器-2

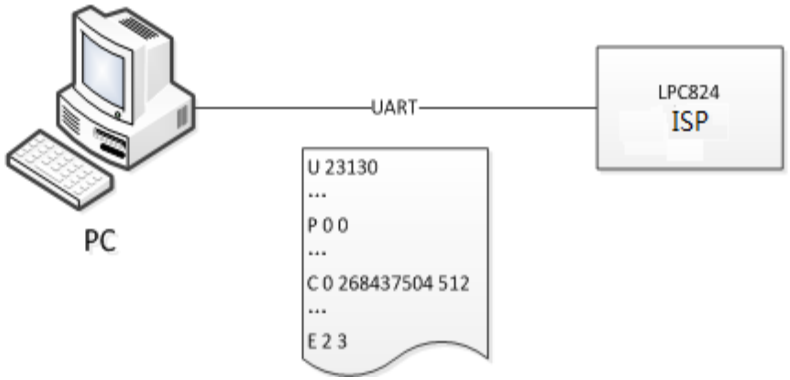
- 获得闪存（FLASH）内容签名
  - 提供16字节对齐起始闪存(FLASH)地址
  - 提供16字节对齐结束闪存(FLASH)地址，并开始计算签名
  - 等待判断签名计算完成
  - 获得32bits闪存内容签名
- 实例代码 （计算从0x4000地址到0x8000地址的签名）

```
const static uint32_t FLASH_SIGNATURE_START_ADDR = 0x1000;
const static uint32_t FLASH_SIGNATURE_END_ADDR = 0x4000;

Chip_FMC_ComputeSignature(FLASH_SIGNATURE_START_ADDR,
FLASH_SIGNATURE_END_ADDR);

while(Chip_FMC_IsSignatureBusy());

uint32_t flash_signature = Chip_FMC_GetSignature(0);
```

# ISP 和 IAP

# ISP模式和ISP的通常用法-1

- 通过串口（USART）ISP命令和ISP程序完成片上闪存（FLASH）的烧写编程
  - –ISP程序运行在ROM中
  - –进入ISP模式的方法
    - 芯片启动时拉低管脚P0_12
    - IAP命令 “Reinvoke ISP”




- 通常用法
  - –固件烧录（小批量）,固件升级
  - –LPC82X 应用程序进入低功耗状态时，调试端口失效，无法从调试端口下载程序，而进入ISP模式可以通过调试端口下载用户程序

# ISP模式和ISP的通常用法-2

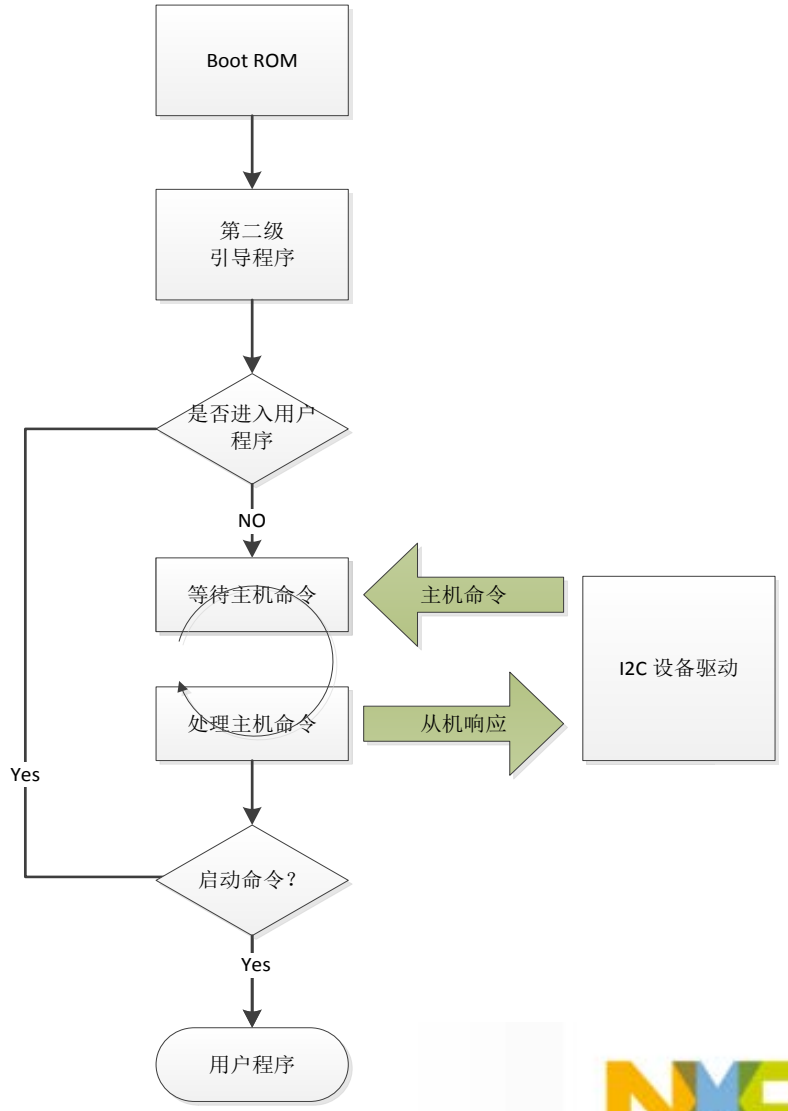
- 在用户应用代码中或者用户第二级引导程序使用IAP系统函数完成片上闪存（FLASH）的烧写编程
  - 用户程序在FLASH或RAM中调用执行IAP系统函数
  - 通常用法
    - 应用程序保存外设配置数据，校准数据到FLASH上
    - 应用程序保存用户数据到FLASH上
    - 第二级引导程序对应用程序升级

# 第二级引导程序

- 为什么需要第二级引导程序？
  - ISP 只可以完成MCU的固件升级，一般不能完成外围器件的固件升级
  - ISP 只可以通过串口0（USART0）完成片上闪存（FLASH）的烧写编程
  - ISP的命令是有限的，有时候不能完全满足用户的功能需求
    - 加密、校验启动
    - 故障现场分析（RAM内存导出）
    - 工厂测试

# 第二级引导程序（示例）

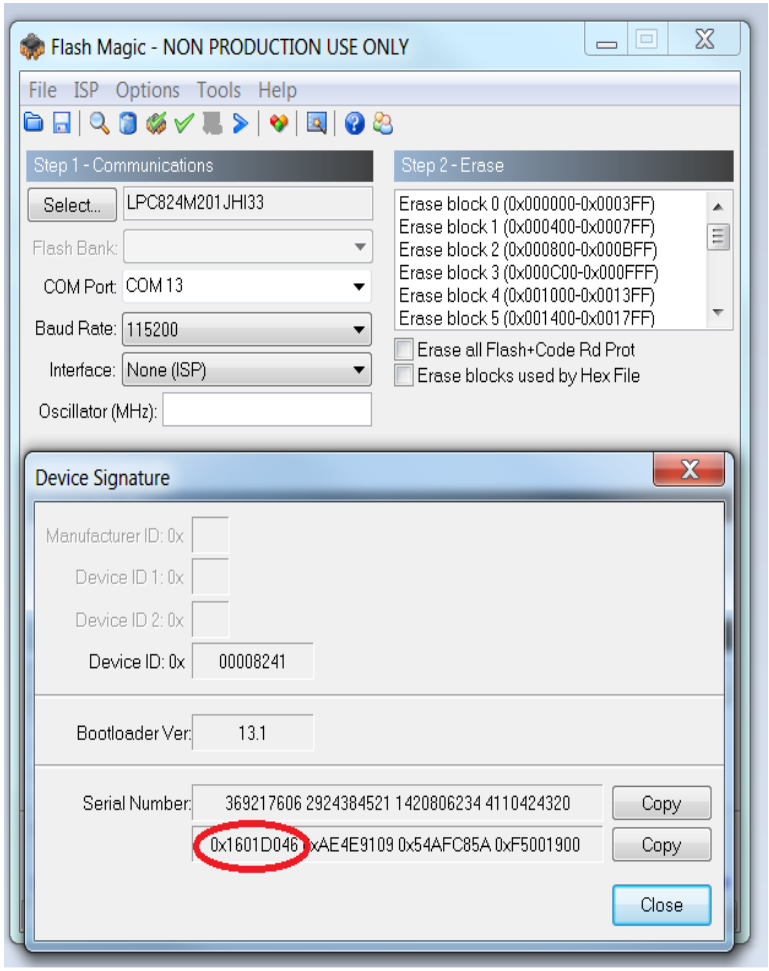
- 第二级引导程序实例完成基于I2C总线的主机从机通信
- 第二级引导程序等待主机指令，解析指令并处理主机指令
- 第二级引导程序响应主机启动命令可以启动用户程序



# 芯片唯一ID（UID）

- UID 是芯片的唯一ID，128bits, 每个LPC82X芯片都有不同的唯一ID，相当于芯片的身份证
- 获得UID的方法
  - –在ISP模式下，通过ISP命令'N'
  - –通过FLASHMAGIC
  - –通过IAP系统函数

```
uint32_t Chip_IAP_ReadUID (uint32_t* uid);
```

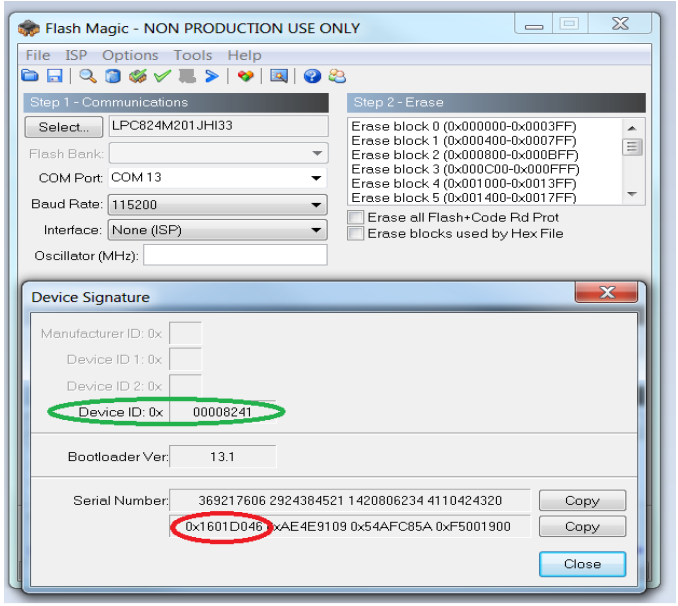


# 产品型号ID (PART ID)

- LPC82X系列有以下产品型号，每个型号的PID不同

| 设备 | 16进制产品ID |
|---|---|
| LPC824M201JHI33 | 0x00008241 |
| LPC822M101JHI33 | 0x00008221 |
| LPC824M201JDH20 | 0x00008242 |
| LPC822M101JDH20 | 0x00008222 |

- 获得产品型号ID
  - –在ISP模式下，通过ISP命令’J’
  - –通过FLASHMAGIC
  - –通过IAP系统函数



# 代码读取保护（CRP）-1

- 允许客户设置不同安全级别的闪存访问
- CRP模式

| NO ISP | CRP1 |
| --- | --- |
| • 禁止进入“在系统编程”（ISP）模式 | • 禁止调试端口（SWD）<br>• 受限的”在系统编程”（ISP）模式<br>✓ RAM地址0x10000000~0x10000300禁止写入<br>✓ RAM地址0x10000000~0x10000200禁止访问<br>✓ 受限擦除FLASH闪存的0块<br>✓ 禁止写入FLASH闪存的0块<br>✓ 禁止ISP命令“Compare”<br>✓ 禁止ISP命令“Read Memory” |
| **CRP2** | **CRP3** |
| • 禁止调试端口（SWD）<br>• 受限的”在系统编程”（ISP）模式<br>✓ 禁止以下ISP命令 “Read Memory”， “Write to RAM”，“Go”, ”Copy RAM to flash”, “Compare”<br>✓ 只允许ISP命令擦出所有用户闪存分区 | • 禁止调试端口（SWD）<br>• 当用户代码存在时，禁止ISP管脚入口<br>• 无法进行工厂测试 |

# 代码读取保护（CRP）-2

- CRP模式配置
  -模式配置地址 0x000002FC

| **CRP 模式** | **配置地址（0x000002FC）的数值** |
|---|---|
| Disable | 0xFFFFFFFF |
| NO ISP | 0x4E697370 |
| CRP1 | 0x12345678 |
| CRP2 | 0x87654321 |
| CRP3 | 0x43218765 |

- CRP 模式的改变在MCU重新上电后生效
- CRP 不影响IAP命令
- 实例代码

```
;//   keil_startup_lpc82x.s
;//   Code Read Protection level (CRP), 0x000002FC
;//   <0xFFFFFFFF=> Disabled
;//   <0x4E697370=> NO_ISP
;//   <0x12345678=> CRP1
;//   <0x87654321=> CRP2
;//   <0x43218765=> CRP3 (Are you sure?)
CRP_Level    EQU    0xFFFFFFFF
```

# ISP 命令-1

| ISP 命令 | 用法 | 实例 |
| --- | --- | --- |
| 解锁（FLASH擦写命令，GO命令） | U <解锁码> | 解锁FLASH擦写，GO 命令<br>U 23130<CR><LF> |
| 设置通信串口波特率 | B <波特率> <停止位> | 设置波特率57600，1停止位<br>B 57600 1<CR><LF> |
| 设置回显 (ISP 回发给主机命令) | A <是否回显> | 关闭回显功能<br>A 0<CR><LF> |
| 写 RAM | W <起始地址> <字节数> | 写4字节到0x10000300 RAM<br>W 268436224 4<CR><LF> |
| 读存储器 | R <地址> <字节数> | 从0x10000000 RAM 读取4字节<br>R 268435456 4<CR><LF> |
| 闪存块写操作准备 | P <块起始地址> <块结束地址> | 擦出FLASH块0<br>P 0 0<CR><LF> |
| 拷贝RAM到FLASH | C <FLASH 地址> <RAM 地址> <字节数> | 从RAM地址0x10000800拷贝512字节到FLASH地址0x0<br>C 0 268437504 512<CR><LF> |
| 跳转 | G <地址> <模式> | 用于跳转到FLASH或RAM地址运行程序，地址需要4字节对齐<br>G 512 T<CR><LF> |

# ISP 命令-2

| ISP 命令 | 用法 | 实例 |
| --- | --- | --- |
| 擦除FLASH块 | E <块起始地址> <块结束地址> | 擦除FLASH块2~3<br>E 2 3<CR><LF> |
| FLASH空白检测（不适用FLASH块0） | I <块起始地址> <块结束地址> | 检测FLASH块2~3是否为空白<br>I 2 3<CR><LF> |
| 读取芯片型号 | J | |
| 读取引导程序版本 | K | 获得引导程序版本号<br>2字节(主版本.从版本) |
| 存储比较 | M <地址1> <地址2> <字节数> | 比较FLASH地址0x2000与RAM地址0x10008000 (比较4字节)<br>M 8192 268468224 4<CR><LF> |
| 读取芯片唯一ID | N | 返回128bits芯片唯一ID |
| 读取CRC校验值 | S <地址> <字节数> | 读取从0x10000500地址开始4字节的CRC值<br>S 268436736 4<CR><LF> |

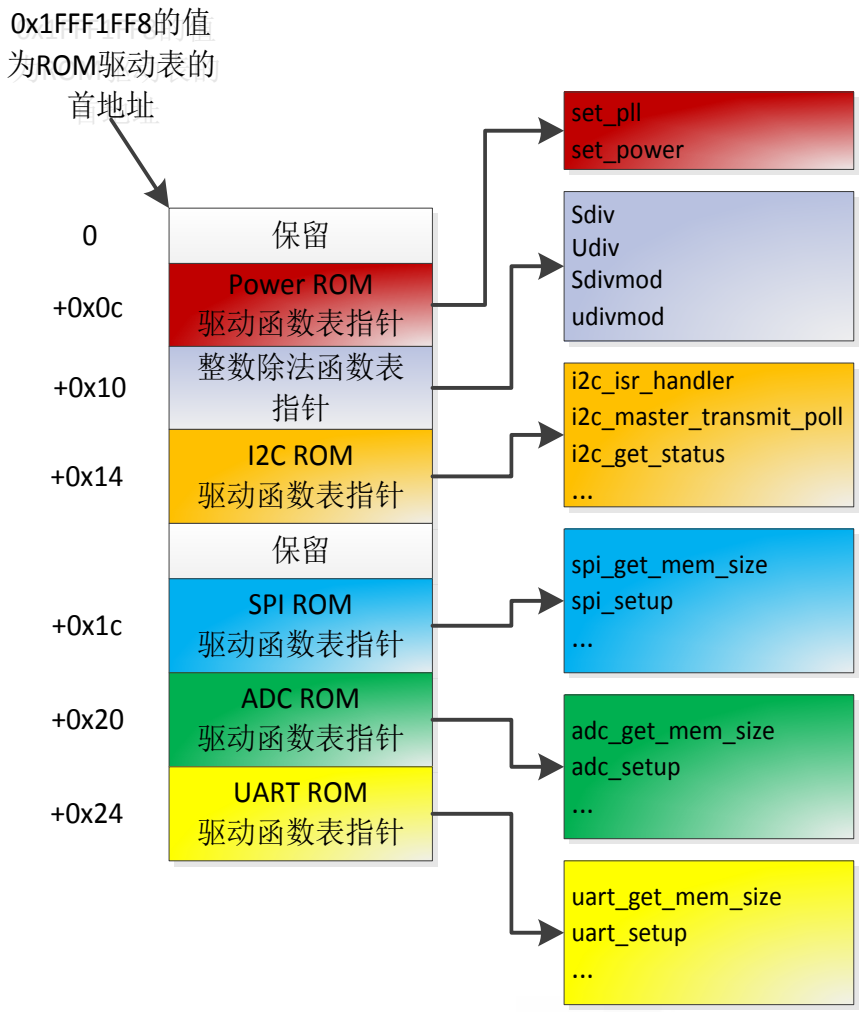
# IAP 命令

| IAP 命令 | 命令号 | 参数 |
| --- | --- | --- |
| 解锁（FLASH擦写命令，GO命令） | 50 | 参数0：起始FLASH块， 参数1：结束FLASH块<br>返回值：无 |
| 拷贝RAM到FLASH | 51 | 参数0：FLASH目标写入地址（64字节对齐）<br>参数1：RAM源读取地址<br>参数2：拷贝字节数（64 或 128 或 256 或 512 或 1024）<br>返回值：无 |
| 擦除FLASH块 | 52 | 参数0：起始FLASH块，参数1：结束FLASH块<br>返回值：无 |
| FLASH空白检测 | 53 | 参数0：起始FLASH块，参数1：结束FLASH块<br>返回值0：状态码（FLASH空？非空？） 返回值1：非空白FLASH字地址 |
| 读取芯片型号 | 54 | 参数：无<br>返回值：芯片型号 |
| 读取引导程序版本 | 55 | 参数：无<br>返回值：引导程序版本号，2字节，主版本.从版本 |
| 存储比较 | 56 | 字比较存储地址1和存储地址2<br>参数0：FLASH或者RAM地址1<br>参数1：FLASH或者RAM地址2<br>参数2：比较字节数，必须是4字节的整数倍 |
| 重入ISP模式 | 57 | 参数：无，返回值：无 |
| 读取芯片唯一ID | 58 | 返回128bits芯片唯一ID<br>参数：无， 返回值0～3：芯片唯一ID |
| 擦除FLASH页 | 59 | 参数0：起始页地址， 参数1：结束页地址 |

# ROM驱动
# ( ROM DRIVER)

# ROM 驱动

- 提供I2C/SPI/UART/ADC ROM 版本的驱动程序
- 提供控制功耗模式的Power 函数
- 提供有符号和无符号的整数乘法函数



# Power ROM 驱动

| void (*set_pll) (unsigned int command[], unsigned int response[]) | |
|---|---|
| 输入参数 | 参数0：系统PLL输入频率（单位kHz）<br>参数1：期望设置的系统时钟（单位kHz）<br>参数2：模式<br>参数3：系统PLL锁频超时 |
| 返回值 | 返回值0：错误标识<br>返回值1：设置的系统时钟（单位kHz） |
| **void (*set_power) (unsigned int command[], unsigned int response[])** | |
| 输入参数 | 参数0：主时钟频率（单位mHz）<br>参数1：模式<br>参数2：系统时钟频率（单位kHz） |
| 返回值 | 返回值0：错误标识 |

SECURE CONNECTIONS
FOR A SMARTER WORLD