



# LPC1100 系列微控制器

---

## 第二十章 Flash 存储器编程固件

用户手册 Rev1.00

### 广州周立功单片机发展有限公司

地址：广州市天河北路 689 号光大银行大厦 12 楼 F4

网址：<http://www.zlgmcu.com>

## 销售与服务网络（一）

### 广州周立功单片机发展有限公司

地址：广州市天河北路 689 号光大银行大厦 12 楼 F4

邮编：510630

电话：(020)38730916 38730917 38730972 38730976 38730977

传真：(020)38730925

网址：[www.zlgmcu.com](http://www.zlgmcu.com)



### 广州专卖店

地址：广州市天河区新赛格电子城 203-204 室

电话：(020)87578634 87569917

传真：(020)87578842

### 南京周立功

地址：南京市珠江路 280 号珠江大厦 1501 室

电话：(025) 68123901 68123902

传真：(025) 68123900

### 北京周立功

地址：北京市海淀区知春路 113 号银网中心 A 座  
1207-1208 室（中发电子市场斜对面）

电话：(010)62536178 62536179 82628073

传真：(010)82614433

### 重庆周立功

地址：重庆市石桥铺科园一路二号大西洋国际大厦  
（赛格电子市场）1611 室

电话：(023)68796438 68796439

传真：(023)68796439

### 杭州周立功

地址：杭州市天目山路 217 号江南电子大厦 502 室

电话：(0571)89719480 89719481 89719482

89719483 89719484 89719485

传真：(0571)89719494

### 成都周立功

地址：成都市一环路南二段 1 号数码科技大厦 403 室

电话：(028)85439836 85437446

传真：(028)85437896

### 深圳周立功

地址：深圳市深南中路 2070 号电子科技大厦 C 座 4 楼 D 室

电话：(0755)83781788（5 线）

传真：(0755)83793285

### 武汉周立功

地址：武汉市洪山区广埠屯珞瑜路 158 号 12128 室  
（华中电脑数码市场）

电话：(027)87168497 87168297 87168397

传真：(027)87163755

### 上海周立功

地址：上海市北京东路 668 号科技京城东座 7E 室

电话：(021)53083452 53083453 53083496

传真：(021)53083491

### 西安办事处

地址：西安市长安北路 54 号太平洋大厦 1201 室

电话：(029)87881296 83063000 87881295

传真：(029)87880865

## 销售与服务网络（二）

### 广州致远电子有限公司

地址：广州市天河区车陂路黄洲工业区3栋2楼

邮编：510660

传真：(020)38601859

网址：[www.embedtools.com](http://www.embedtools.com) （嵌入式系统事业部）

[www.embedcontrol.com](http://www.embedcontrol.com) （工控网络事业部）

[www.ecardsys.com](http://www.ecardsys.com) （楼宇自动化事业部）



#### 技术支持：

##### CAN-bus:

电话：(020)22644381 22644382 22644253

邮箱：[can.support@embedcontrol.com](mailto:can.support@embedcontrol.com)

##### MiniARM:

电话：(020)28872684 28267813

邮箱：[miniarm.support@embedtools.com](mailto:miniarm.support@embedtools.com)

##### 无线通讯：

电话：(020) 22644386

邮箱：[wireless@embedcontrol.com](mailto:wireless@embedcontrol.com)

##### 编程器：

电话：(020)22644371

邮箱：[programmer@embedtools.com](mailto:programmer@embedtools.com)

##### ARM 嵌入式系统：

电话：(020) 22644383 22644384

邮箱：[NXPARM@zlgmcu.com](mailto:NXPARM@zlgmcu.com)

#### 销售：

电话：(020)22644249 22644399 22644372 22644261 28872524

28872342 28872349 28872569 28872573 38601786

#### 维修：

电话：(020)22644245

##### iCAN 及数据采集：

电话：(020)28872344 22644373

邮箱：[ican@embedcontrol.com](mailto:ican@embedcontrol.com)

##### 以太网：

电话：(020)22644380 22644385

邮箱：[ethernet.support@embedcontrol.com](mailto:ethernet.support@embedcontrol.com)

##### 串行通讯：

电话：(020)28267800 22644385

邮箱：[serial@embedcontrol.com](mailto:serial@embedcontrol.com)

##### 分析仪器：

电话：(020)22644375

邮箱：[tools@embedtools.com](mailto:tools@embedtools.com)

##### 楼宇自动化：

电话：(020)22644376 22644389 28267806

邮箱：[mjs.support@ecardsys.com](mailto:mjs.support@ecardsys.com)

[mifare.support@zlgmcu.com](mailto:mifare.support@zlgmcu.com)

## 目 录

第 20 章 Flash 存储器编程固件 .....	2
20.1 本章导读 .....	2
20.2 BOOT loader .....	2
20.3 特性 .....	2
20.4 应用 .....	2
20.5 描述 .....	2
20.5.1 复位后的存储器映射 .....	2
20.5.2 通信协议 .....	3
20.5.3 Boot 处理流程图 .....	4
20.5.4 扇区号 .....	5
20.6 代码读保护 (CRP) .....	6
20.6.1 ISP 入口保护 .....	7
20.7 ISP 命令 .....	7
20.7.1 解锁<解锁代码> .....	8
20.7.2 设置波特率<波特率><停止位> .....	8
20.7.3 回应<设定> .....	8
20.7.4 写 RAM<起始地址><字节数> .....	9
20.7.5 读存储器<地址><字节数> .....	9
20.7.6 准备写操作的扇区<起始扇区号><结束扇区号> .....	10
20.7.7 将 RAM 内容复制到 Flash<Flash 地址><RAM 地址><字节数> .....	10
20.7.8 运行<地址><模式> .....	11
20.7.9 擦除扇区<起始扇区号><结束扇区号> .....	11
20.7.10 扇区查空<起始扇区号><结束扇区号> .....	11
20.7.11 读器件标识号 .....	12
20.7.12 读 Boot 代码版本号 .....	12
20.7.13 比较<地址 1><地址 2><字节数> .....	12
20.7.14 读 UID .....	13
20.7.15 ISP 返回代码 .....	13
20.8 IAP 命令 .....	14
20.8.1 准备写操作扇区 .....	15
20.8.2 将 RAM 内容复制到 Flash .....	16
20.8.3 擦除扇区 .....	16
20.8.4 扇区查空 .....	17
20.8.5 读器件标识号 .....	17
20.8.6 读 Boot 代码版本号 .....	17
20.8.7 比较<地址 1><地址 2><字节数> .....	18
20.8.8 重新调用 ISP .....	18
20.8.9 读 UID .....	18
20.8.10 IAP 状态代码 .....	18
20.9 串行调试 (SWD) Flash 编程接口 .....	19
20.10 Flash 存储器访问 .....	19

## 第20章 Flash 存储器编程固件

### 20.1 本章导读

各种 Flash 配置和功能见表 20.1。

表 20.1 LPC111x 的 Flash 配置

类型编号	Flash
LPC1111	8 KB
LPC1112	16KB
LPC1113	32KB

配置 Flash 存储器访问时间相关的内容见本章“Flash 存储器访问”小节。

### 20.2 BOOT loader

BOOT loader 控制芯片复位后的初始化操作，并提供对 Flash 存储器编程的方法。这可以对空片进行初始编程、对事先已编程的芯片进行擦除和再编程或者是在系统运行时通过系统中的应用程序对 Flash 存储器进行编程。

### 20.3 特性

- 在系统编程：在系统编程（ISP）是通过使用 BOOT loader 软件和 UART0 串口对片内 Flash 存储器进行编程/再编程的方法。这种方法也可以在芯片位于终端用户板时使用；
- 在应用编程：在应用编程（IAP）是通过终端用户的应用代码对片内 Flash 存储器进行擦除/写操作的方法；

### 20.4 应用

Boot loader 提供了在系统与在应用编程的接口，以便对片上 Flash 进行编程。

### 20.5 描述

每次芯片上电或复位都会执行 BOOT loader 代码。BOOT loader 可以执行 ISP 命令处理程序或用户的应用代码。复位后若 PIO0\_1 脚为低电平，会被系统认为是外部硬件请求执行 ISP 命令处理程序。假设在 RESET 管脚上出现上升沿时，电源引脚正处于标称的水平，那么在最多 3ms 后，会对 PIO0\_1 管脚信号进行采样并确定是否继续用户代码还是进入 ISP 处理程序。如果对 PIO0\_1 管脚采样的结果为低电平，同时看门狗溢出标志被置位，那么外部硬件启动 ISP 命令处理程序的请求将被忽略。如果并不存在执行 ISP 命令处理程序的请求

（PIO0\_1 在复位后的采样结果为高电平），那么会搜寻有效的用户程序。如果找到有效的用户程序，那么执行控制将由此程序接管。如果并未找到有效的用户程序，那么会调用自动波特率例程。

#### 20.5.1 复位后的存储器映射

引导块的大小为 16 KB，起始地址为存储器单元中的 0x1FFF 0000。Boot loader 被设计为从该存储器区域运行，但 ISP 和 IAP 软件都会占用部分的片内 RAM。本章后面部分描述 RAM 的使用情况。复位后，驻留在片内 Flash 存储器的引导块中的中断向量也变为有效。也就是说，引导块底部的 512 个字节的内容，将出现在起始地址为 0x0000 0000 的存储区域中。

### 1. 有效用户代码的判定标准

有效用户代码判定标准：保留的 ARM Cortex-M0 异常向量单元 7（在向量表中的偏移量为 0x0000 001C）应当包含表入口 0~6 的校验和的 2 的补码。这样就使前 8 个表入口的校验和为 0。Boot loader 代码校验 Flash 扇区 0 中的前 8 个单元。如果结果为 0，执行控制权便转移给用户代码。

如果信号为无效，那么自动波特率程序通过串口 0 与主机进行同步。主机应当发送一个同步字符“?”（0x3F）并等待响应。主机的串口应设定为 8 个数据位、1 个停止位且无校验位。自动波特率程序根据自身的频率测量接收到的同步字符的位时间并对串口波特率发生器进行编程。它还向主机发送一个 ASCII 字符串（“Synchronized<CR><LF>”），作为响应，主机应当发送接收到的字符串（“Synchronized<CR><LF>”）。自动波特率程序通过观察接收到的字符来验证是否同步。如果通过验证，则向主机发送“OK<CR><LF>”。主机应当通过发送正在运行部分的晶振频率（单位为 KHz）作为响应。例如，如果运行在 10MHz，主机的响应当为“10000<CR><LF>”。在接收到晶振频率后再向主机发送“OK<CR><LF>”。如果同步验证没有通过，那么自动波特率程序再次等待一个同步字符。在用户调用 ISP 的情况下要使自动波特率正确工作，CCLK 频率应当大于或等于 10MHz。

在接收到晶振频率后，器件初始化并调用 ISP 命令处理程序。出于安全性的考虑，在执行 Flash 擦除/写操作命令和“Go”命令之前必须执行“Unlock（解锁）”命令。执行其它命令时不需要执行解锁命令。每次 ISP 命令处理都要执行一次“Unlock（解锁）”命令。解锁命令在本章“ISP 命令”小节中介绍。

## 20.5.2 通信协议

所有 ISP 命令都以单个 ASCII 字符串形式发送。字符串应当以回车(CR)和/或换行(LF)控制字符作为结束符。多余的<CR>和<LF>将被忽略。所有 ISP 响应都是以<CR><LF>结束的 ASCII 字符串形式发送。数据是以 UU 编码格式发送和接收。

### 1. ISP 命令格式

“命令 参数\_0 参数\_1 ... 参数\_n<CR><LF>” “数据”（只适用于写命令）。

### 2. ISP 响应格式

“返回\_代码<CR><LF>响应\_0<CR><LF>响应\_1<CR><LF> ... 响应\_n<CR><LF>” “数据”（数据只适用于读命令）

### 3. ISP 数据格式

数据流采用 UU 编码格式。UU 编码算法将 3 字节的二进制数据转换成 4 字节可打印的 ASCII 字符集，该编码的效率高于 Hex 格式（Hex 格式将 1 字节的二进制数据转换成 2 字节的 ASCII Hex 数据）。发送器在发送 20 个 UU 编码行之后发送校验和。任何 UU 编码行的长度都不应超过 61 个字符（字节），也就是说它可以保持 45 个数据字节。接收器应当将该校验和与接收字节的校验和进行比较。如果校验和匹配，接收器响应“OK<CR><LF>”来继续下一次发送。如果校验和不匹配，接收器响应“RESEND<CR><LF>”。作为响应，发送器还应当重新发送字节。

UU 编码的描述可以再 wotsit 的网页上找到。

### 4. ISP 流量控制

软件 XON/XOFF 流程控制机制可防止缓冲区溢出时的数据丢失。当数据快速到达时，发送 ASCII 控制字符 DC3（停止）使数据流停止。发送 ASCII 控制字符 DC1（启动）恢复数据流。主机也应支持相同的流控制机制。

### 5. ISP 命令中止

命令可通过发送 ASCII 控制字符“ESC”来中止。该特性在“ISP 命令”一节中并没有作为一个命令。一旦接收到退出代码，ISP 命令处理器就会等待新的命令。

### 6. ISP 过程中的中断

在任何复位后，位于 Flash 引导块内的引导块中断向量都有效。

### 7. IAP 过程中的中断

在擦除/写操作过程中，片内 Flash 存储器不可访问。只有当用户应用程序代码启动执行时，用户 Flash 区的中断向量才有效。在调用 Flash 擦除/写 IAP 之前，用户应当禁止中断或确保用户中断向量在 RAM 中有效且中断处理程序位于 RAM 中。IAP 代码不能使用或禁止中断。

### 8. ISP 命令处理器使用的 RAM

ISP 命令使用片内地址 0x1000 017C 到 0x1000 025B 范围内的 RAM。用户可以使用该区域，但是在复位时内容可能会丢失。Flash 编程命令使用片内 RAM 最顶端的 32 字节。堆栈位于 RAM 顶端减去 32 字节。可使用的最大堆栈为 256 字节，堆栈是向下递增的。

### 9. IAP 命令处理器使用的 RAM

Flash 编程命令使用片内 RAM 最顶端的 32 字节。可使用的最大堆栈为 128 字节，堆栈是向下递增的。

## 20.5.3 Boot 处理流程图

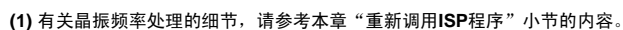


图 20.1 Boot 处理流程图

#### 20.5.4 扇区号

有些 IAP 和 ISP 命令以“扇区”为单位进行操作,同时还指定扇区号。下表列出了 LPC111x 系列 ARM 的扇区号和存储器地址之间的对应关系。

表 20.2 LPC111x 系列 ARM 器件的扇区

扇区号	扇区大小 (KB)	地址范围
0	4	0x0000 0000 – 0x0000 0FFF
1	4	0x0000 1000 – 0x0000 1FFF
2	4	0x0000 2000 – 0x0000 2FFF
3	4	0x0000 3000 – 0x0000 3FFF
4	4	0x0000 4000 – 0x0000 4FFF
5	4	0x0000 5000 – 0x0000 5FFF
6	4	0x0000 6000 – 0x0000 6FFF
7	4	0x0000 7000 – 0x0000 7FFF



## 20.6 代码读保护（CRP）

代码读保护是允许用户在系统中通过使能不同的安全级别来限制对片内 Flash 的访问和 ISP 的使用的一种机制。需要时，可通过在 Flash 地址单元 0x0000 02FC 编程特定的格式来调用 CRP。IAP 命令不受代码读保护的影响。

要点：CRP 所作出的任何改变只有在器件经过一个电源周期之后才会生效。

表 20.3 代码读保护选项

名称	在 0x000002FC 处编程的格式	描述
NO_ISP	0x4E64 7370	阻止对 PIO0_1 管脚进行采样而进入 ISP 模式。PIO0_1 管脚作其他用途
CRP1	0x12345678	禁止通过 JTAG 引脚访问芯片。该模式允许在下列条件约束下使用下列 ISP 命令来进行局部的 Flash 更新： 写 RAM 命令不能访问在 0x10000200 以下的 RAM； 将 RAM 内容复制到 Flash 命令不能写扇区 0； 只有在选择擦除所有扇区时，擦除命令才能擦除扇区 0； 比较命令禁能。 禁能读取存储器命令 当需要 CRP 且要更新 Flash 字段时可使用该模式，但不能擦除所有扇区。 由于在 Flash 部分更新的情况下比较命令禁能，因此次级装载程序应执行校验和机制来验证 Flash 的完整性
CRP2	0x87654321	禁止通过 JTAG 引脚访问芯片。下列的 ISP 命令被禁能： 读存储器； 写 RAM； 运行； 将 RAM 内容复制到 Flash； 比较。 使能 CRP2 时，ISP 擦除命令仅允许擦除所有用户扇区的内容
CRP3	0x43218765	禁止通过 JTAG 引脚访问芯片。如果 Flash 扇区 0 中存在有效用户代码，则禁止通过拉低 PIO0_1 来进入 ISP。 该模式有效的禁止了通过 PIO0_1 脚来强行进入 ISP 的行为。用户的应用程序可决定是调用 IAP 来进行 Flash 更新还是通过 UART0 重新调用 ISP 命令来进行 Flash 更新 <b>注：如果选择了 CRP3，此后该器件将无法执行厂商测试</b>

表 20.4 代码读保护硬件/软件的相互作用

CRP	用户代码是否有效	复位时 PIO0_1 脚电平	JTAG 是否使能	LPC1700 系列 ARM 是否进入 ISP 模式	ISP 模式下部分 Flash 是否更新
否	否	X	是	是	是
否	是	高	是	否	NA
否	是	低	是	是	是
CRP1	是	高	否	否	NA
CRP1	是	低	否	是	是

续上表

CRP	用户代码是否有效	复位时 PIO0_1 脚电平	JTAG 是否使能	LPC1700 系列 ARM 是否进入 ISP 模式	ISP 模式下部分 Flash 是否更新
CRP2	是	高	否	否	NA
CRP2	是	低	否	是	否
CRP3	是	X	否	否	NA
CRP1	否	X	否	是	是
CRP2	否	X	否	是	否
CRP3	否	X	否	是	否

表 20.5 不同 CRP 等级下允许使用的 ISP 命令

ISP 命令	CRP1	CRP2	CRP3
解锁	是	是	否
设置波特率	是	是	否
回应	是	是	否
写 RAM	是；只在 0x1000 0200 以上	否	否
读存储器	否	否	否
准备写保护扇区	是	是	否
将 RAM 复制到 Flash	是；不到扇区 0	否	否
运行	否	否	否
擦除扇区	是；当有在擦除所有扇区时才能擦除扇区 0	是；只有所有扇区	否
扇区查空	否	否	否
读器件 ID	否	否	否
读 Boot 代码版本	否	否	否
比较	否	否	否
读 UID	否	否	否

若 CRP 模式使能，并且允许通过 ISP 访问芯片，不支持或受限制的 ISP 命令将被终止执行并同时返回代码 CODE\_READ\_PROTECTION\_ENABLED。

### 20.6.1 ISP 入口保护

除了 3 种 CRP 模式外，用户可防止对 PIO0\_1 采样进入 ISP 模式，从而释放 PIO0\_1 脚，使其用于其它方面。这称为 NO\_ISP 模式。可通过对 0x0000 02FC 单元的 0x4E69 7370 编程进入 NO\_ISP 模式。

## 20.7 ISP 命令

下面的命令是 ISP 命令处理器所接受的命令。每个命令都支持具体的状态代码。当接收到未定义命令时，命令处理程序会发送返回代码 INVALID\_COMMAND。命令和返回代码为 ASCII 格式。

只有当接收到的 ISP 命令执行完毕时，ISP 命令处理器才会发送 CMD\_SUCCESS，这时主机才能发送新的 ISP 命令。但“设置波特率”、“写 RAM”、“读存储器”和“运行”命令除外。

表 20.6 ISP 命令总结

ISP 命令	用法	描述
解锁	U<解锁代码>	见表 20.7
设置波特率	B<波特率><停止位>	见表 20.8
回应	A<设定>	见表 20.9
写 RAM	W<起始地址><字节数>	见表 20.10
读存储器	R<地址><字节数>	见表 20.11
准备写操作的扇区	P<起始扇区号><结束扇区号>	见表 20.12
将 RAM 内容复制到 Flash	C<Flash 地址><RAM 地址><字节数>	见表 20.13
运行	G<地址><模式>	见表 20.14
擦除扇区	E<起始扇区号><结束扇区号>	见表 20.15
扇区查空	I<起始扇区号><结束扇区号>	见表 20.16
读器件 ID	J	见表 20.17
读 Boot 代码版本	K	见表 20.19
比较	M<地址 1><地址 2><字节数>	见表 20.20
读 UID	N	见表 20.21

### 20.7.1 解锁<解锁代码>

表 20.7 ISP 解锁命令

命令	U
输入	解锁代码: 23130 <sub>10</sub>
返回代码	CMD_SUCCESS   INVALID_CODE   PARAM_ERROR
描述	该命令用于解锁 Flash 写、擦除和运行命令
举例	“U 23130<CR><LF>” 解锁 Flash 写/擦除和运行命令

### 20.7.2 设置波特率<波特率><停止位>

表 20.8 ISP 设置波特率命令

命令	B
输入	波特率: 9600   19200   38400   57600   115200   230400 停止位: 1   2
返回代码	CMD_SUCCESS   INVALID_BAUD_RATE   INVALID_STOP_BIT   PARAM_ERROR
描述	该命令用于改变波特率。新的波特率在命令处理程序发送 CMD_SUCCESS 返回代码之后生效
举例	“B 57600 1<CR><LF>”设置串口波特率 57600bps 和 1 个停止位

### 20.7.3 回应<设定>

表 20.9 ISP 回应命令

命令	A
输入	设定：打开=1   关闭=0
返回代码	CMD_SUCCESS   PARAM_ERROR
描述	回应命令的默认设定是打开。当打开时，ISP 命令处理器将接收到的串行数据发送回主机
举例	"A 0<CR><LF>"回应关闭

#### 20.7.4 写 RAM<起始地址><字节数>

主机应只在接收到 CMD\_SUCCESS 返回代码后才发送数据。当发送完 20 个 UU 编码行之后主机应当发送校验和。校验和是通过把原始数据（UU 编码前）字节的值叠加起来产生的，当发送完 20 个 UU 编码行后该值会重置。任何 UU 编码行的长度不应超过 61 个字符（字节），也就是说，它可以携带 45 个数据字节。当数据少于 20 个 UU 编码行时，校验和按照实际发送的字节数进行计算。ISP 命令处理器将它与接收字节的校验和进行比较。如果校验和匹配，那么 ISP 命令处理器响应“OK<CR><LF>”来继续下一次发送。如果校验和不匹配，那么 ISP 命令处理器响应“RESEND<CR><LF>”。作为响应，主机应当重新发送字节。

表 20.10 ISP 写 RAM 命令

命令	W
输入	<b>起始地址：</b> 要写入数据字节的 RAM 地址。该地址应当以字为边界 <b>字节数：</b> 写入的字节数。计数值应当为 4 的倍数
返回代码	CMD_SUCCESS   ADDR_ERROR（地址不是以字为边界）   ADDR_NOT_MAPPED   COUNT_ERROR（字节计数值不是 4 的倍数）   PARAM_ERROR   CODE_READ_PROTECTION_ENABLED
描述	该命令用于将数据下载到 RAM。数据应当为 UU 编码格式。当代码读保护使能时该命令被禁止
举例	"W 268436224 4<CR><LF>"向地址 0x1000 0300 写入 4 个字节数据

#### 20.7.5 读存储器<地址><字节数>

数据流之后是命令成功返回代码。发送完 20 个 UU 编码行之后发送校验和。校验和是通过把原始数据（UU 编码前）字节的值叠加起来产生的，当发送完 20 个 UU 编码行后该值会重置。任何 UU 编码行的长度不应超过 61 个字符（字节），也就是说，它可以携带 45 个数据字节。当数据少于 20 个 UU 编码行时，校验和按照实际发送的字节数进行计算。主机将它与接收字节的校验和进行比较。如果校验和匹配，那么主机响应“OK<CR><LF>”来继续下一次发送。如果校验和不匹配，主机响应“RESEND<CR><LF>”。作为响应，ISP 命令处理器应当重新发送字节。

表 20.11 ISP 读存储器命令

命令	R
输入	<b>起始地址：</b> 被读出数据字节的地址，该地址应当以字为边界 <b>字节数：</b> 读出的字节数。计数值应该为 4 的倍数

续上表

命令	R
返回代码	CMD_SUCCESS, 后面是<实际数据 (UU 编码)>  ADDR_ERROR (地址不是以字为边界)   ADDR_NOT_MAPPED   COUNT_ERROR (字节计数值不是 4 的倍数)   PARAM_ERROR   CODE_READ_PROTECTION_ENABLED
描述	该命令用于读出 RAM 或 Flash 存储器的数据。当代码读保护使能时该命令禁能
举例	"R 268435456 4<CR><LF>"从地址 0x1000 0000 读出 4 个字节数据

## 20.7.6 准备写操作的扇区<起始扇区号> <结束扇区号>

该命令使 Flash 写/擦除操作分成两个步骤处理。

表 20.12 ISP 准备写操作命令的扇区

命令	P
输入	起始扇区号 结束扇区号: 应当大于或等于起始扇区号
返回代码	CMD_SUCCESS   BUSY   INVALID_SECTOR   PARAM_ERROR
描述	该命令必须在执行“将 RAM 内容复制到 Flash”或“擦除扇区”命令之前执行。“将 RAM 内容复制到 Flash”或“擦除扇区”命令的成功执行会使相关的扇区再次被保护。该命令不能用于引导块。要准备单个扇区, 可将“起始”和“结束”扇区号设置为相同值
举例	"P 0 0<CR><LF>"准备 Flash 扇区 0

## 20.7.7 将 RAM 内容复制到 Flash<Flash 地址><RAM 地址><字节数>

表 20.13 ISP 复制命令

命令	C
输入	Flash 地址 (DST): 要写入数据字节的目标 Flash 地址。目标地址的边界应当为 256 字节 RAM 地址 (SRC): 读出数据字节的源 RAM 地址 字节数: 写入的字节数目。应当为 256   512   1024   4096
返回代码	CMD_SUCCESS   SRC_ADDR_ERROR (地址不以字为边界)   DST_ADDR_ERROR (地址边界错误)   SRC_ADDR_NOT_MAPPED   DST_ADDR_NOT_MAPPED   COUNT_ERROR (字节计数值不是 256   512   1024   4096)   SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION   BUSY   CMD_LOCKED   PARAM_ERROR   CODE_READ_PROTECTION_ENABLED

续上表

命令	C
描述	该命令用于编程 Flash 存储器。“准备写操作的扇区”命令应当在该命令之前被执行。当成功执行复制命令后，受影响的扇区将自动再次受到保护。不能通过该命令来写引导块。当代码读保护使能时该命令被禁止
举例	"C 0 268468224 512<CR><LF>"将 RAM 地址 0x1000 8000 开始的 512 字节复制到 Flash 地址 0

## 20.7.8 运行<地址><模式>

表 20.14 ISP 运行命令

命令	G
输入	<b>地址：</b> 代码执行起始的 Flash 或 RAM 地址。该地址应当以字为边界 <b>模式：</b> T（执行 Thumb 模式下的程序） A（执行 ARM 模式下的程序）
返回代码	CMD_SUCCESS   ADDR_ERROR   ADDR_NOT_MAPPED   CMD_LOCKED   PARAM_ERROR   CODE_READ_PROTECTION_ENABLED
描述	该命令用于执行位于 RAM 或 Flash 存储器当中的程序。一旦成功执行该命令，就有可能不再返回 ISP 命令处理程序。当代码读保护使能时该命令禁能
举例	"G 0 A<CR><LF>"跳转到 ARM 模式下的地址 0x0000 0000 处

## 20.7.9 擦除扇区<起始扇区号><结束扇区号>

表 20.15 ISP 擦除扇区命令

命令	E
输入	<b>起始扇区号</b> <b>结束扇区号：</b> 应当大于或等于起始扇区号
返回代码	CMD_SUCCESS   BUSY   INVALID_SECTOR   SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION   CMD_LOCKED   PARAM_ERROR   CODE_READ_PROTECTION_ENABLED
描述	该命令用于擦除片内 Flash 存储器的一个或多个扇区。引导块不能使用该命令来擦除。当代码读保护使能时，该命令只允许擦除所有用户扇区的内容
举例	"E 2 3<CR><LF>"擦除 Flash 扇区 2 和 3

## 20.7.10 扇区查空<起始扇区号><结束扇区号>

表 20.16 ISP 扇区查空命令

命令	I
输入	起始扇区号: 结束扇区号: 应当大于或等于起始扇区号
返回代码	CMD_SUCCESS   SECTOR_NOT_BLANK (后跟<第一个非空字的偏移量> <非空字的内容>)   INVALID_SECTOR   PARAM_ERROR
描述	该命令用于对片内 Flash 存储器的一个或多个扇区进行查空 对扇区 0 查空总是失败, 这是由于前 64 字节重新映射到 Flash 引导块
举例	"I 2 3<CR><LF>"对 Flash 扇区 2 和 3 进行查空

### 20.7.11 读器件标识号

表 20.17 ISP 读器件标识命令

命令	J
输入	无
返回代码	CMD_SUCCESS 后跟 ASCII 格式的器件标识号 (见表 20.18)
描述	该命令用于读取器件的标识号

表 20.18 LPC111x 系列 ARM 器件标识号

器件	ASCII/dec 编码	Hex 编码
LPC1111FHN33/101	68571179	0x0416 502B
LPC1111FHN33/201	68571179	0x0416 502B
LPC1112FHN33/101	69554219	0x0425 502B
LPC1112FHN33/201	69554219	0x0425 502B
LPC1113FHN33/201	70537259	0x0434 502B
LPC1113FHN33/301	70520875	0x0434 102B
LPC1113FBD48/301	70520875	0x0434 102B
LPC1114FHN33/201	71585835	0x0444 502B
LPC1114FHN33/301	71569451	0x0444 102B
LPC1114FBD48/301	71569451	0x0444 102B
LPC1114FA44/301	71569451	0x0444 102B

### 20.7.12 读 Boot 代码版本号

表 20.19 ISP 读 Boot 代码版本号命令

命令	K
输入	无
返回代码	CMD_SUCCESS 后跟 2 字节 ASCII 格式的 Boot 代码版本号 将其解释为<字节 1 (主)>.<字节 0 (次)>
描述	该命令用于读取 Boot 代码版本号

### 20.7.13 比较<地址 1><地址 2><字节数>

表 20.20 ISP 比较命令

命令	M
输入	地址 1 (DST)：要比较的数据字节的起始 Flash 或 RAM 地址。该地址应当以字为边界 地址 2 (SRC)：要比较的数据字节的起始 Flash 或 RAM 地址。该地址应当以字为边界 字节数：待比较的字节数；计数值应当为 4 的倍数
返回代码	CMD_SUCCESS   (源和目标数据相同) COMPARE_ERROR   (后跟第一个不匹配字节的地址) COUNT_ERROR (字节数不是 4 的倍数)   ADDR_ERROR   ADDR_NOT_MAPPED   PARAM_ERROR
描述	该命令用来比较两个地址单元的存储器内容 当源或目的地址包含从地址 0 发起的前 512 个字节时，比较结果可能不正确。前 512 个字节重新映射到 boot ROM
举例	"M 8192 268468224 4<CR><LF>"将 RAM 地址 0x1000 8000 开始的 4 个字节与 Flash 地址 0x2000 开始的 4 个字节进行比较

## 20.7.14 读 UID

表 20.21 读 UID 命令

命令	N
输入	无
返回代码	CMD_SUCCESS 后跟 E 类测试信息的 32 位字 (ASCII 格式)。先发送低位地址的字
描述	该命令用于读唯一的 ID

## 20.7.15 ISP 返回代码

表 20.22 ISP 返回代码总览

返回代码	符号	描述
0	CMD_SUCCESS	成功执行命令。只有成功执行了主机发出的命令后，才由 ISP 处理器发送该代码
1	INVALID_COMMAND	无效命令
2	SRC_ADDR_ERROR	源地址没有以字为边界
3	DST_ADDR_ERROR	目标地址的边界错误
4	SRC_ADDR_NOT_MAPPED	源地址的映射不在存储器映射中。无法考证运算值适用之处
5	DST_ADDR_NOT_MAPPED	目标地址的映射不在存储器映射中。无法考证运算值适用之处
6	COUNT_ERROR	字节计数值不是 4 的倍数或是一个非法值
7	INVALID_SECTOR	扇区号无效或结束扇区号大于起始扇区号
8	SECTOR_NOT_BLANK	扇区非空
9	SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION	为写操作准备扇区的命令未执行
10	COMPARE_ERROR	源和目标数据不相等
11	BUSY	Flash 编程硬件接口忙
12	PARAM_ERROR	参数不足或无效参数



续上表

返回代码	符号	描述
13	ADDR_ERROR	地址没有以字为边界
14	ADDR_NOT_MAPPED	地址的映射不在存储器映射中。无法考证运算值的适用之处
15	CMD_LOCKED	命令被锁定
16	INVALID_CODE	解锁代码无效
17	INVALID_BAUD_RATE	无效波特率设定
18	INVALID_STOP_BIT	无效停止位设定
19	CODE_READ_PROTECTION_ENABLED	代码读保护使能

## 20.8 IAP 命令

对于在应用编程来说，应当通过寄存器 `r0` 中的字指针来调用 IAP 程序，该字指针指向含有命令代码和参数的存储器（RAM）。IAP 命令的结果返回到寄存器 `r1` 所指向的结果表。用户可以把寄存器 `r0` 和 `r1` 中的指针赋予相同的值，如此便能将命令表复用来存放结果。参数表应当大到足够保存所有的结果以防结果的数目大于参数的数目。参数传递见图 20.2。参数和结果的数目根据 IAP 命令而有所不同。参数的最大数目为 5，传送到“将 RAM 内容复制到 Flash”命令。结果的最大数目为 4，由“读 UID”命令返回。命令处理器在接收到一个未定义的命令时发送状态代码 `INVALID_COMMAND`。IAP 程序是 Thumb 代码，驻留在地址 `0x1FFF1FF0`。

可按下面方式使用 C 调用 IAP 函数。

定义 IAP 程序的入口地址。由于 IAP 地址的第 0 位是 1，因此，当程序计数器转移到该地址时会使当前指令集变为 Thumb 指令集。

```
#define IAP_LOCATION 0x1ffff1ff1
```

定义数据结构或指针，将 IAP 命令表和结果表传递给 IAP 函数：

```
unsigned long command[5];
```

```
unsigned long result[4];
```

或

```
unsigned long *command;
```

```
unsigned long *result;
```

```
command = (unsigned long *) 0x....
```

```
result = (unsigned long *) 0x....
```

定义函数类型指针，其包含 2 个参数，无返回值。需要注意的是 IAP 将函数结果和 `R1` 中的表格基址一同返回。

```
typedef void (*IAP) (unsigned int [], unsigned int []);
```

```
IAP iap_entry;
```

设置函数指针：

```
iap_entry=(IAP) IAP_LOCATION;
```

可使用下面的语句来调用 IAP

```
iap_entry (command , result);
```

根据 ARM 规范（ARM Thumb 过程调用标准 SWS ESPC 0002 A-05），可分别通过 r0、r1、r2 和 r3 寄存器来传递最多 4 个参数。另外的参数通过堆栈传递。最多 4 个参数可以分别通过 R0、R1、R2 和 R3 寄存器返回。其它的参数通过存储器间接返回。有些 IAP 调用需要多于 4 个参数。如果使用 ARM 建议的机制来传递/返回参数，则有可能因为不同厂商所提供的 C 编译器不同而产生问题。使用建议的参数传递机制便可降低这种风险。

在写或擦除操作过程中不能访问 Flash 存储器。那些导致 Flash 写/擦除操作的 IAP 命令将使用片内 RAM 顶端的 32 个字节空间来执行。如果应用程序中允许 IAP 编程，那么用户程序不应使用该空间。

表 20.23 IAP 命令总览

IAP 命令	命令代码	描述
准备写操作的扇区	50 <sub>10</sub>	见表 20.24
将 RAM 内容复制到 Flash	51 <sub>10</sub>	见表 20.25
擦除扇区	52 <sub>10</sub>	见表 20.26
扇区查空	53 <sub>10</sub>	见表 20.27
读器件 ID	54 <sub>10</sub>	见表 20.28
读 Boot 代码版本	55 <sub>10</sub>	见表 20.29
比较	56 <sub>10</sub>	见表 20.30
重新调用 ISP	57 <sub>10</sub>	见表 20.31
读 UID	58 <sub>10</sub>	见表 20.32

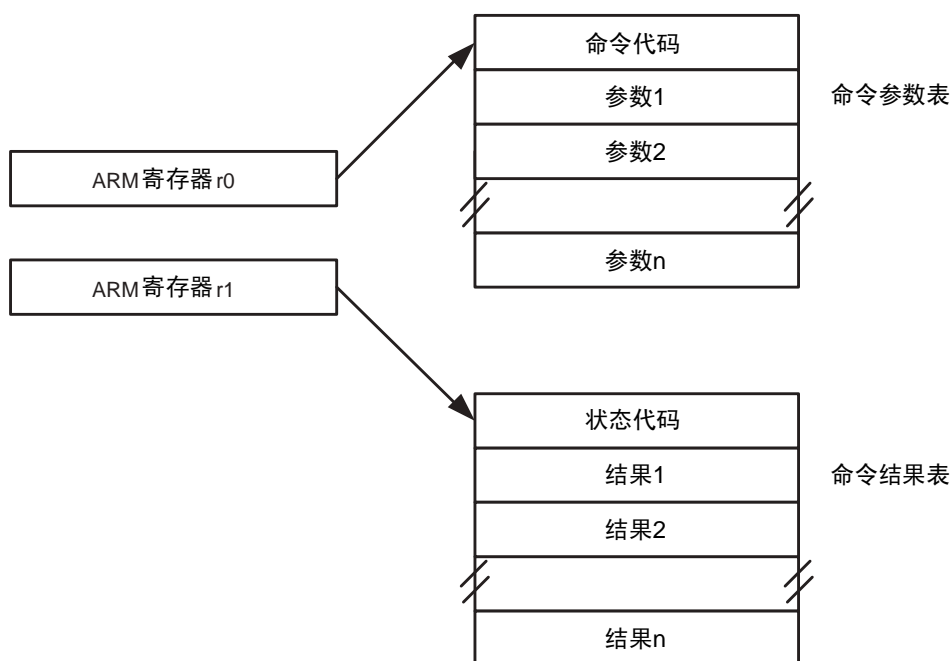


图 20.2 IAP 参数传递

### 20.8.1 准备写操作扇区

该命令使 Flash 写/擦除操作分两步执行。

表 20.24 IAP 为写操作准备扇区命令描述

命令	准备写操作扇区
输入	命令代码: 50 <sub>10</sub> 参数 0: 起始扇区号 参数 1: 结束扇区号 (应当大于或等于起始扇区号)
返回代码	CMD_SUCCESS   BUSY   INVALID_SECTOR
结果	无
描述	该命令必须在执行“将 RAM 内容复制到 Flash”或“擦除扇区”命令之前执行。“将 RAM 内容复制到 Flash”或“擦除扇区”命令的成功执行会导致相关的扇区再次被保护。不能使用该命令准备启动扇区。要准备单个扇区, 可将起始和结束扇区号设为相同值

## 20.8.2 将 RAM 内容复制到 Flash

表 20.25 IAP 将 RAM 内容复制到 Flash 命令描述

命令	将 RAM 内容复制到 Flash
输入	命令代码: 51 <sub>10</sub> 参数 0 (DST): 要写入数据字节的目标 Flash 地址。目标地址的边界应当为 256 字节 参数 1 (SRC): 读取数据字节的源 RAM 地址。该地址应当以字为边界 参数 2: 写入字节的数目。应当为 256   512   1024   4096 参数 3: 系统时钟频率 (CCLK) (单位: KHz)
返回代码	CMD_SUCCESS   SRC_ADDR_ERROR (地址不以字为边界)   DST_ADDR_ERROR (地址边界错误)   SRC_ADDR_NOT_MAPPED   DST_ADDR_NOT_MAPPED   COUNT_ERROR (字节计数值不是 256   512   1024   4096)   SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION   BUSY
结果	无
描述	该命令用于编程 Flash 存储器。受影响的扇区应先通过调用“准备写操作扇区”命令准备好。当成功执行复制命令后, 扇区将自动受到保护。该命令不能写引导扇区

## 20.8.3 擦除扇区

表 20.26 IAP 擦除扇区命令

命令	擦除扇区
输入	命令代码: 52 <sub>10</sub> 参数 0: 起始扇区号 参数 1: 结束扇区号 (应当大于或等于起始扇区号) 参数 2: 系统时钟频率 (CCLK) (单位: KHz)

续上表

命令	擦除扇区
返回代码	CMD_SUCCESS   BUSY   SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION   INVALID_SECTOR
结果	无
描述	该命令用于擦除片内 Flash 存储器的一个或多个扇区。该命令不能擦除引导扇区。要擦除单个扇区可将“起始”和“结束”扇区号设定为相同值

## 20.8.4 扇区查空

表 20.27 IAP 扇区查空命令描述

命令	扇区查空
输入	命令代码: $53_{10}$ 参数 0: 起始扇区号 参数 1: 结束扇区号 (应当大于或等于起始扇区号)
返回代码	CMD_SUCCESS   BUSY   SECTOR_NOT_BLANK   INVALID_SECTOR
结果	结果 0: 状态代码为 SECTOR_NOT_BLANK 时第一个非空字位置的偏移量 结果 1: 非空字位置的内容
描述	该命令用于对片内 Flash 存储器的一个或多个扇区进行查空。要查空单个扇区可将“起始”和“结束”扇区号设定为相同值

## 20.8.5 读器件标识号

表 20.28 IAP 读器件标识命令描述

命令	读器件标识号
输入	命令代码: $54_{10}$ 参数: 无
返回代码	CMD_SUCCESS
结果	结果 0: 器件标识号
描述	该命令用于读取器件的标识号

## 20.8.6 读 Boot 代码版本号

表 20.29 IAP 读 Boot 代码版本号命令

命令	读 Boot 代码版本号
输入	命令代码: $55_{10}$ 参数: 无
返回代码	CMD_SUCCESS
结果	结果 0: 2 字节 Boot 代码版本号 (ASCII 格式) 将其解释为 <字节 1 (主) >.<字节 0 (次) >

续上表

命令	读 Boot 代码版本号
描述	该命令用于读取引导代码版本号

## 20.8.7 比较<地址 1><地址 2><字节数>

表 20.30 IAP 比较命令描述

命令	比较
输入	<b>命令代码:</b> 56 <sub>10</sub> <b>参数 0 (DST):</b> 要被比较的数据字节的 Flash 或 RAM 起始地址。该地址应当以字为边界 <b>参数 1 (SRC):</b> 要被比较的数据字节的 Flash 或 RAM 起始地址。该地址应当以字为边界 <b>参数 2:</b> 待比较的字节数。计数值应当为 4 的倍数
返回代码	CMD_SUCCESS   COMPARE_ERROR   COUNT_ERROR (字节数不是 4 的倍数)   ADDR_ERROR   ADDR_NOT_MAPPED
结果	<b>结果 0:</b> 当状态代码为 COMPARE_ERROR 时第一个不匹配字节的偏移地址
描述	该命令用来比较两个地址单元的存储器内容 当源或目标地址包含从地址 0 开始的前 512 字节中的任意一个地址时, 比较的结果可能不正确。因为前 512 字节是可以重新映射到 RAM 中的

## 20.8.8 重新调用 ISP

表 20.31 重新调用 ISP

命令	比较
输入	<b>命令代码:</b> 57 <sub>10</sub>
返回代码	无
结果	无
描述	该命令用来调用 ISP 模式中的引导装载程序。它会映射引导向量, 设定 PCLK = CCLK, 配置 UART 管脚 RXD 和 TXD, 复位计数器/定时器 CT32B1 和 U0FDR。当内部 Flash 存储器中出现有效的用户程序且 PIO0_1 脚不可访问时, 可使用该指令强制进入 ISP 模式

## 20.8.9 读 UID

表 20.32 IAP 读 UID 命令

命令	比较
输入	<b>命令代码:</b> 58 <sub>10</sub>
返回代码	CMD_SUCCESS
结果	<b>结果 0:</b> 第 1 个 32 位字 (在最低地址) <b>结果 1:</b> 第 2 个 32 位字 <b>结果 2:</b> 第 3 个 32 位字 <b>结果 3:</b> 第 4 个 32 位字
描述	该命令用于读唯一 ID

## 20.8.10 IAP 状态代码

表 20.33 IAP 状态代码总结

返回代码	符号	描述
0	CMD_SUCCESS	成功执行命令
1	INVALID_COMMAND	无效命令
2	SRC_ADDR_ERROR	源地址不是以字为边界
3	DST_ADDR_ERROR	目标地址的边界错误
4	SRC_ADDR_NOT_MAPPED	源地址的映射不在存储器映射中。无法得到运算值适用对象的位置
5	DST_ADDR_NOT_MAPPED	目标地址的映射不在存储器映射中。无法得到运算值适用对象的位置
6	COUNT_ERROR	字节计数值不是 4 的倍数或是一个非法值
7	INVALID_SECTOR	扇区号无效
8	SECTOR_NOT_BLANK	扇区非空
9	SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION	为写操作准备扇区命令未执行
10	COMPARE_ERROR	源和目标数据不相同
11	BUSY	Flash 编程硬件接口忙

## 20.9 串行调试（SWD）Flash 编程接口

调试工具可将一部分 Flash 映像写入 RAM，然后按照正确的偏移地址重复调用 IAP 命令“将 RAM 内容复制到 Flash”。

## 20.10 Flash 存储器访问

根据系统时钟频率,通过写入位于 0x4003 C010 地址的 FLASHCFG 寄存器,可以对 Flash 存储器访问的访问时间作出多种不同的配置。

注：若此寄存器设置不当，将导致 LPC111x 的 Flash 存储器的不正确运行。

表 20.34 Flash 配置寄存器（FLASHCFG，地址 0x4003 C010）各个位的描述

位	标识	值	描述	复位值
1:0	FLASHTIM		Flash 存储器访问时间。FLASHTIM+1 等于 Flash 访问所占用的系统时钟数目	10
		00	1 个系统时钟的 Flash 访问时间（用于高于 20MHz 的系统时钟）	
		01	2 个系统时钟的 Flash 访问时间（用于高于 40MHz 的系统时钟）	
		10	3 个系统时钟的 Flash 访问时间（用于高于 50MHz 的系统时钟）	
		11	保留	
31:2	-	-	保留。用户软件不可更改这些位的值。位 31:2 必须以所读取的原值写回。	<tbid>