

32 位微控制器

Flash 串行编程（TypeB 协议）

适用对象

系列	型号	系列	型号
HC32L07	HC32L072PATA HC32L072KATA HC32L072JATA HC32L072FAUA HC32L073PATA HC32L073KATA HC32L073JATA	HC32F07	HC32F072PATA HC32F072KATA HC32F072JATA HC32F072FAUA
HC32L17	HC32L176PATA HC32L176MATA HC32L176KATA HC32L176JATA HC32L170JATA HC32L170FAUA	HC32F17	HC32F176PATA HC32F176MATA HC32F176KATA HC32F176JATA HC32F170LATA HC32F170JATA HC32F170FAUA
HC32L19	HC32L196PCTA HC32L196MCTA HC32L196KCTA HC32L196JCTA HC32L190JCTA HC32L190FCUA	HC32F19	HC32F196PCTA HC32F196MCTA HC32F196KCTA HC32F196JCTA HC32F190JCTA HC32F190FCUA

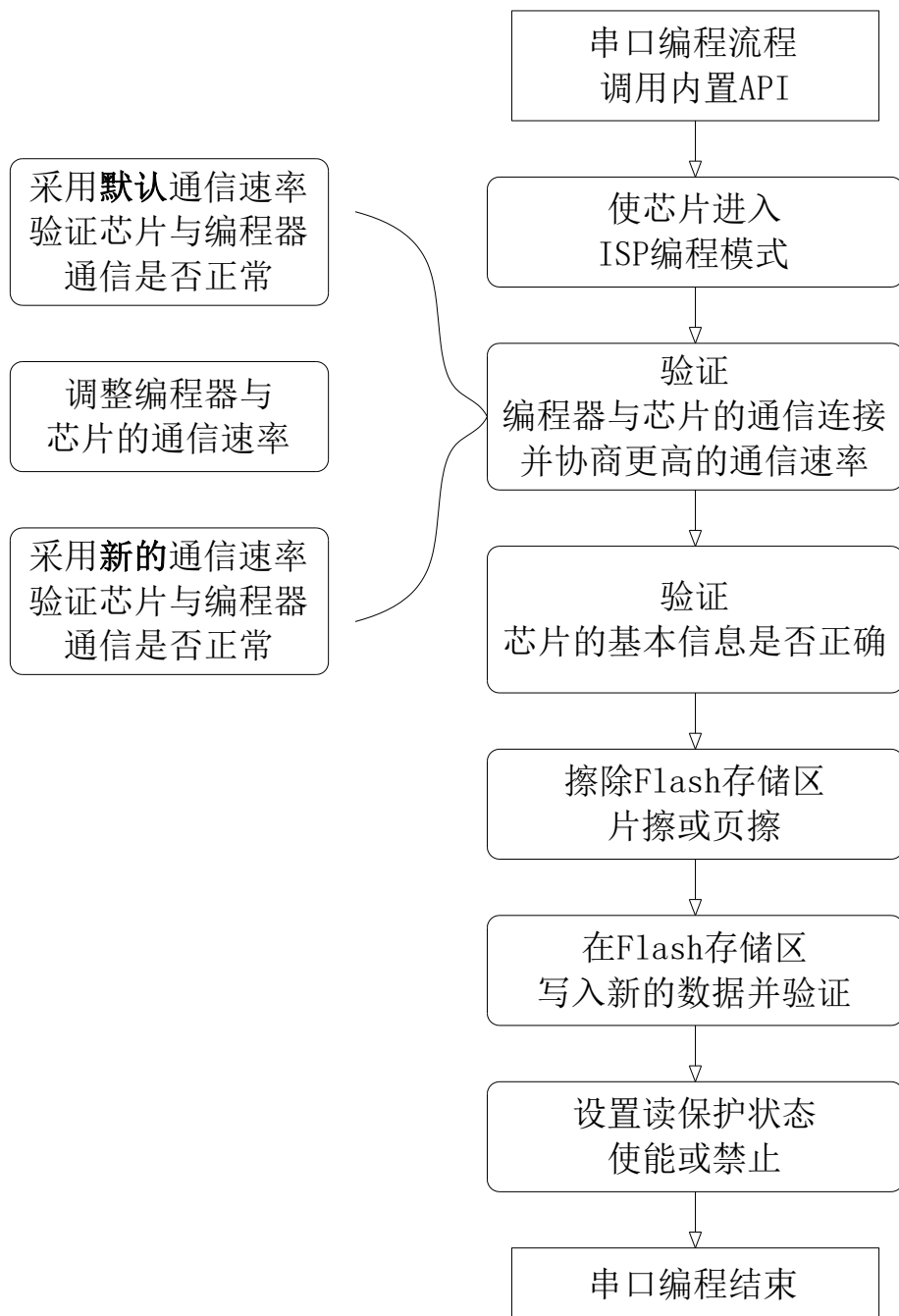
目 录

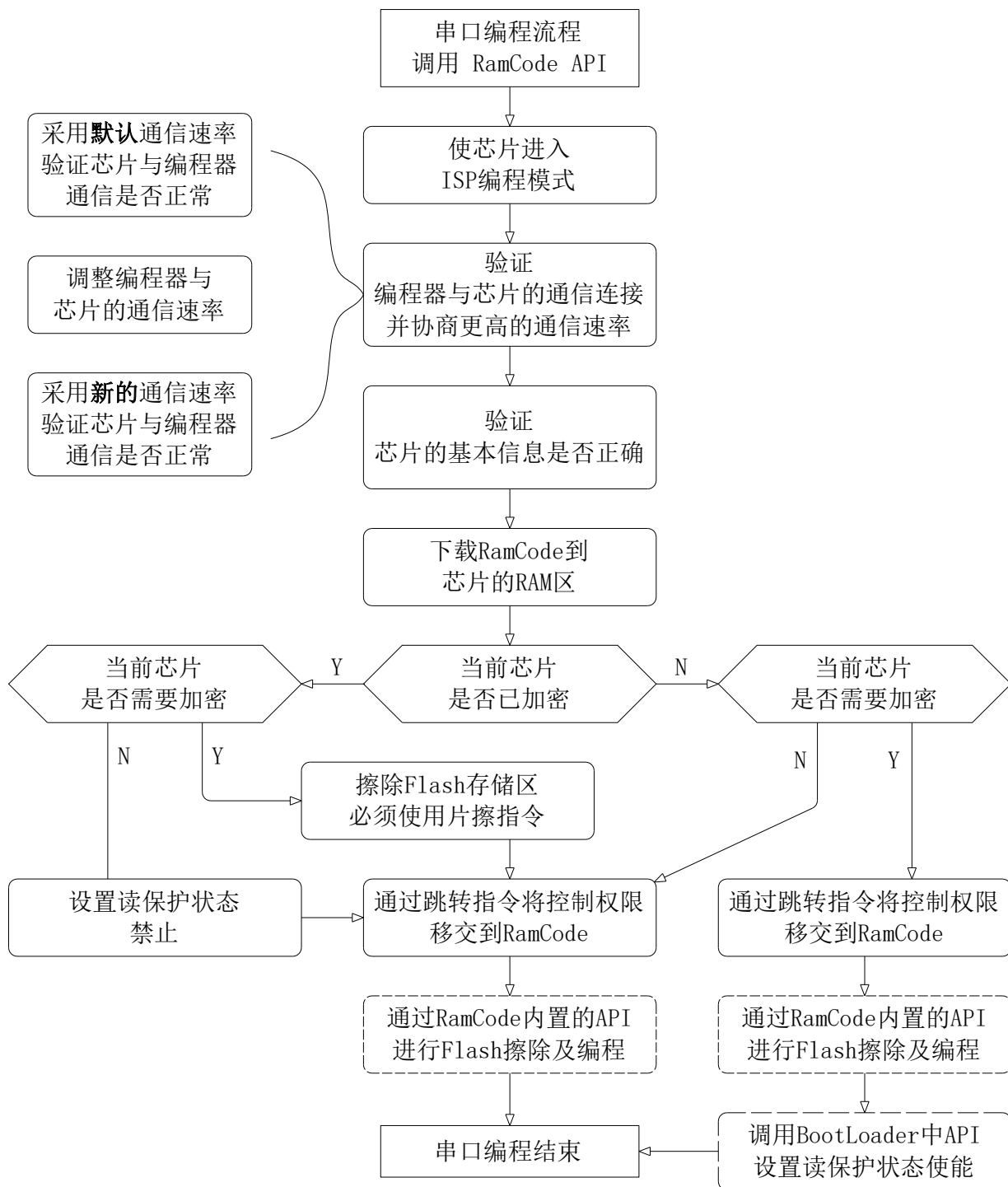
1. 芯片编程流程	3
2. 使目标 MCU 进入 ISP 模式的方法	5
3. UART 通信格式	5
3.1 物理层	5
3.2 协议层	5
4. 应用层指令	6
4.1 Query 指令	6
4.2 PPS 指令	7
4.3 Set BaseAddr 指令	7
4.4 ChipErase 指令	7
4.5 SectorErase 指令	7
4.6 Blank Check 指令	8
4.7 Write Data 指令	8
4.8 Read Data 指令	8
4.9 ReadOutProtection 指令	9
4.10 Jump 指令	9
4.11 返回状态字说明	9
5. 编程示例	11
5.1 完全调用 BooLoader 内置的 API 进行编程	11
5.2 混合调用 BootLoader 和 RamCode 内 API 进行编程	13
6. 版本信息 & 联系方式	17

1. 芯片编程流程

具有 2 种芯片编程流程：完全调用 BootLoader 内置 API、混合调用 BootLoader 和 RamCode 内 API。

具体编程流程图如下方所示。





2. 使目标 MCU 进入 ISP 模式的方法

Step1. 编程器拉低 MCU 的 RST 引脚;

Step2. 编程器向 MCU 的 BOOT 引脚提供高电平;

Step3. 编程器拉高 MCU 的 RST 引脚, 延时不小于 50ms。

Step4. MCU 进入 ISP 编程模式。

3. UART 通信格式

目标 MCU 进入 ISP 编程模式后, 编程器与芯片之间的通信格式如下方所示。

3.1 物理层

使用 UART 进行通信, UART 初始通信参数为: 115200 - 8 - N - 1。

可通过 PPS 指令修改通信速率。

3.2 协议层

上下行时数据包格式均为 Head - Len - Info - CRC

Head 为帧头, 其值固定为 0x65

Len 的值代表 Info 有多少个字节, 取值范围为 0-255

Info 为实际的应用层数据, 其长度为 Len。上行时, Info 的第一个字节代表执行状态

CRC 为 Head - Len - Info 的所有数据的计算值, 低字节先发

收发数据示例:

下行: 65 01 10 65F3

上行: 65 09 00 1800080001010600 BA2B

其中黄色为 Head, 蓝色为 Len, 绿色为 Info, 灰色为 CRC。

4. 应用层指令

指令名称	指令格式	指令功能说明
Query	10	编程器查询目标 MCU 是否能正常通信。
PPS	11 DIVN	编程器与目标 MCU 协商通信速率。
Set BaseAddr	27 BaseAddr	设置读、写、擦操作的基地址。
ChipErase	20	擦除目标 MCU 闪存内的所有数据。
SectorErase	21 Offset	擦除 BaseAddr + Offset 地址所在的页面。
Blank Check	22	检查目标 MCU 闪存内数据是否全为 FF。
Write Data	28 Offset D1-Dn	将 N 字节的数据写入 BaseAddr + Offset 处，并验证读出与写入的数据是否相同。N = 1 - 248。
Read Data	29 Offset N	从 BaseAddr + Offset 处，读出 N 个字节。N = 1 - 255。
ReadOutProtection	2B RdEn	设置或读取芯片的读保护状态。
Jump	30 Addr	跳转到 Addr 所指定的地址开始执行程序。

4.1 Query 指令

- 指令功能：编程器查询目标 MCU 是否能正常通信。
- 下行格式：10
- 上行格式：状态字 + HCLK + PRSC + BootLoaderId
- 指令说明：HCLK 为 2 个字节，低字节先发，以 MHz 为单位。

PRSC 为 2 个字节，低字节先发，UART 时钟 = HCLK / PRSC。

BootLoaderId 为 4 个字节，低字节先发。

4.2 PPS 指令

- 指令功能：编程器与目标 MCU 协商通信速率。
- 下行格式：11 DIVN
- 上行格式：状态字
- 指令说明：DIVN 为 2 个字节，低字节先发。

MCU 返回状态字后，将通信波特率调整为 $HCLK / RPSC / DIVN$ 。

4.3 Set BaseAddr 指令

- 指令功能：设置读、写、擦操作的基地址。
- 下行格式：27 BaseAddr
- 上行格式：状态字
- 指令说明：BaseAddr 为 4 个字节，低字节先发。

0x000xxxxx 代表闪存存储区。

0x2000xxxx 代表 RAM 区。

4.4 ChipErase 指令

- 指令功能：擦除目标 MCU 闪存内的所有数据。
- 下行格式：20
- 上行格式：状态字
- 指令说明：擦除完成后，闪存内的数据全部变成 0xFF。

4.5 SectorErase 指令

- 指令功能：擦除 BaseAddr + Offset 地址所在的页面
- 下行格式：21 Offset
- 上行格式：状态字
- 指令说明：Offset 为 2 个字节，低字节先发。

4.6 Blank Check 指令

- 指令功能：检查目标 MCU 闪存内数据是否全为 FF。
- 下行格式：22
- 上行格式：状态字
- 指令说明：无。

4.7 Write Data 指令

- 指令功能：将 N 字节的数据写入 BaseAddr + Offset 处，并验证读出与写入的数据是否相同。
- 下行格式：28 Offset D1-Dn
- 上行格式：状态字
- 指令说明：Offset 为 2 个字节，低字节先发。

$N = 1 - 248$ 。

当 BaseAddr+Offset 与待编程字节数均为 Word 对齐时，调用 Word Program 操作，否则调用 Byte Program 操作。Word Program 仅需 Byte Program 一半的时间。

4.8 Read Data 指令

- 指令功能：从 BaseAddr + Offset 处，读出 N 个字节。
- 下行格式：29 Offset N
- 上行格式：状态字 + D1-Dn
- 指令说明：Offset 为 2 个字节，低字节先发。

N 为 1 个字节，取值范围为 1 - 255。

读取 0x00100C60~0x00100C6F 可以获取商业编号。

读取 0x00100C70~0x00100C73 可以获取 Flash 的容量。

读取 0x00100C74~0x00100C77 可以获取 Ram 的容量。

读取 0x00100C78~0x00100C79 可以获取 FlashSectorSize。

读取 0x00100C7A~0x00100C7B 可以获取引脚数量。

4.9 ReadOutProtection 指令

- 指令功能：设置或读取芯片的读保护状态。
- 下行格式：2B RdEn
- 上行格式：状态字 + RdState + Cnt
- 指令说明：RdEn 为 0x00 则闪存内的任何数据都不能读出；SWD 接口禁止。

RdEn 为 0xFF 则闪存内的任何数据都可以读出；SWD 接口使能。

RdEn 为 0x55 则不操作当前读保护状态，只返回当前保护状态。

RdState 为 0x00 代表闪存为加密态，其内的数据不可读出。

RdState 为 0xFF 代表闪存为解密态，其内的数据可以读出。

Cnt 代表芯片读保护状态可以改写的剩余次数，最多可改写 64 次。

如 MCU 已处于加密态，收到 RdEn 为 0xFF 的指令时，会将闪存内的所有数据擦除。

4.10 Jump 指令

- 指令功能：跳转到 Addr 所指定的地址开始执行程序。
- 下行格式：30 Addr
- 上行格式：状态字
- 指令说明：Addr 为 4 个字节，低字节先发。

Addr 只能是 0x2000xxxx 或 0x00000000。

4.11 返回状态字说明

0x00，代表执行成功

0x10，代表通信 CRC 检验错误，需要重发该命令

0x11，代表 UART 通信出错

0x20，代表指令不支持

0x21，代表参数不支持

0x30，代表没有读权限

0x31, 代表没有写权限

0x32, 代表没有跳转权限

0x40, 代表 Write Data 失败

0x41, 代表 Blank Check 失败

0X42, 代表验证失败

48004700000000002008ED00E00D020000B901000000F00DF8002801D000BF00BF00BF0
0BF002000BF00BFFFF79DFF00F002F80120704780B500F002F801BDFEE70746384600F0
02F8FBE7FEE780B500BF00BF024A11001820ABBEFBE72600020000BF00BF00BF00BF
FFF7D4FFFEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE76D
AF

IN 650100E4E3

OUT 653128E001FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7
FEE7FEE7FEE7FEE7FEE7FEE7FEE7FEE7000070479C3B

IN 650100E4E3

OUT 650527000001004485

IN 650100E4E3

OUT 650728000064000000CC18

IN 650100E4E3

OUT 650429000004BD79

IN 65050064000000D927

OUT 65022B004E79

IN 650300003D3B87

5.2 混合调用 BootLoader 和 RamCode 内 API 进行编程

//以下为 BootLoader 指令

-->65011065F3

<--6509001800080010011500594B

-->65031103007C98

<--650100E4E3

-->650527000010000D09

<--650100E4E3

-->650429600C10F583

<--651100484333324C3139365043544100000000F1EA

-->650527000800205C7B

<--650100E4E3

-->65F3280000B81E00201108002000000000970F002072B60448016881F3088803490860034800
4700000008002008ED00E0B90F0020C068016873480162734A8918734A914201D306218171704
780B500F0F5F87048016803220A430260002109700168CA06D20FFBD101BD10B504006068802
1C902884205D36749401880218900884210D200F0DAF8624801680322914302220A4302600021
626811700168CA06D20F03D0FAE705205749887110BD80B501004A6948688023DB029A4205
D3574B984207D3564B9A4204D20A89C96800F012F901BD05204C49887101BD80B50189494A
91800289C168406800F02FF901BD1CB500213B23

<--650100E4E3

-->65F328F0006A461180C16809684068424C4218521E8023DB029A4202D30520A07102E06A4
600F0D0F8424869460988017269460988090A41720220A08013BD7CB500220092C1680968334E
0125B580394C22726A46406800F0C4F8002802D00098306073BD257273BD01202B4988803149
314A1278FF2A00D100200872704780B500202E490968884701BD704710B5224C00F08BF900F0
0EF90128F9D118212000083000F09FF92000083000F01AF9A071E16821600021A18000280BD1
2048617A89004158002903D020000830884701E00220A071200000F041F9A08800F05DF9607A0
12805D1A0790028D1D1206A00F081F9607A0FE8

<--650100E4E3

-->65F328E0010A28CBD1A0790028C8D1FA200001401EFDD1BFF34F8F0E480E490860BFF34
F8F00BFFDE7101E002080DAFFFFC11C0F002000024000F6EFFF000A1000000C1000041C0020
FC0B10009C0B1000F80E00200400FA050CED00E04A484B4901604B490160704700E00000494
84749C162474AC262C0230360C162C2628A234360C162C262A2238360C162C262424BC360C1

62C262414B0361C162C26290234361C162C262B423DB008361C162C2623B4BC361C162C2620
0230362C162C262DB430363C162C2624363C162C2620364C162C2624364704730B5002300240
3E005785B19401C641C8C42F9D31380002030BD30B5238F

<--650100E4E3

-->65F328D0020300002400E0641C8C4208D21D006B1C2D78FF2DF7D000191060012030BD00
2030BDF0B5224B1C4CDC601C4CDC6001241D680326B54325431D6095070BD1150415D01D6
82D092540FBD10D680560091D001D121FF3E7002507E01E6836092640FBD14E5D47193E706
D1C9542F5D3186800092040FBD1F0BD10B5002304E004780C70401C491C5B1C9CB29442F7D
3002010BD00002C0002405A5A0000A5A5000000000240E0A5010040D10C00701700002000024
010B50A000021002303E004780919401C5B1C9CB29442F8D3C8B210BD4848018809290BDB4
7494A78052A09D08A79C9790902114309310088814204D0EF55

<--650100E4E3

-->65F328C0030020704700880928FAD10120704738B50400002000253B4A118810803A480278
2270427862708278C3781B021343027912041A4343791B06134363608379C27912021A43228163
689B185B1E63612378492B16D163780B2B01DA002B01D1022510E0002A02D002000832E2604
218521E12782274491E89B2FFF7A4FF217C884200D00125280032BD38B504001E4DA0796870
2068A8702068000AE8702068000C28712068000E6871A088A871A088000AE871A188083189B2
2800FFF783FFA1886918087231BD80B50100093189B20E4800F086F801BD38B500F096F80023
09490A88002A01D149280AD1074A0C88074DECE1

<--650100E4E3

-->65F328B004AC4204DA0B885C1C0C80D05431BD13700B8031BD341E0020041C002009020
000002200BF094202D0491E4254FCD1704730B500210022234BD4B2A5005D59A84209D0521C
D4B20C2CF6DB00BF38A0405A1D49086030BD6100F7E770B51B48002101601B491B4ACA61
1B4A4A601B4A4A600A68830D13430B6019491A4ACB681340CB60194B02240725ED435E682
E4026435E600E6832400A60CA68144E1640CE601A6815402C431C60420C0B6813430B600B6A
1A430A629421890001609C214900416170BD980E0020180000400400004004200040010080925A
5A0000A5A50000000D0240FFBFFFFF340C0240FFDFFFFFC210

<--650100E4E3

-->65F328A00530B500228023DB050AE0045D1C601C69A507ED0FFBD05C690225AC435C61
521C94B28C42F1D330BD8020C0050169C907FCD541690122914341610068C0B270473801D00
09C004E0034001A00170027000C000D000300020038B5124812498160124A82600368E024E400
1C430460104B104CDD882540C5609D882540C5601B881C40C4608160826001680B4A0A40026
00620FFF707FEFFF768FFFFF79DFD002032BD00BF002000405A5A0000A5A50000000C1000F

6. 版本信息 & 联系方式

日期	版本	修改记录
2019/11/27	Rev1.0	初版发布
2020/6/29	Rev1.1	更新支持型号，修正部分笔误



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: 上海市浦东新区中科路 1867 号 A 座 10 层

邮编: 201203

