

## 32 位微控制器

# HC32F4A0 系列的 Bootloader 使用

本产品支持芯片系列如下

F 系列	HC32F4A0
------	----------

# 目 录

<b>1</b>	<b>摘要 .....</b>	<b>3</b>
<b>2</b>	<b>HC32F4A0 系列的 Bootloader.....</b>	<b>4</b>
2.1	简介 .....	4
2.2	硬件连接.....	4
2.3	激活方式.....	5
2.4	程序配置.....	5
2.5	工作流程.....	6
2.6	协议命令.....	7
2.6.1	握手命令 .....	7
2.6.2	下载命令 .....	8
2.6.3	跳转命令 .....	9
2.6.4	ACK 列表 .....	9
2.7	错误处理机制 .....	10
2.7.1	接收到错误的命令数据帧.....	10
2.7.2	发生通信错误和超时错误.....	10
2.8	注意事项.....	11
2.8.1	Boot 模式下访问 MCU Flash .....	11
<b>3</b>	<b>总结 .....</b>	<b>12</b>
<b>4</b>	<b>版本信息 &amp; 联系方式 .....</b>	<b>13</b>

## 1 摘要

对 HC32F4A0 系列 MCU Flash 进行串行编程可分为以下 4 个部分：

- 1) 编程器与目标芯片建立连接
- 2) 编程器下载 RAMCODE 程序到目标芯片内置 SRAM
- 3) 目标芯片跳转到 RAMCODE 程序运行
- 4) 编程器通过 RAMCODE 程序提供的协议命令实现对 Flash 串行编程或其它操作

本篇应用笔记主要介绍通过 HC32F4A0 系列的 Bootloader 程序实现上述前 3 个部分的操作步骤和注意事项。

## 2 HC32F4A0 系列的 Bootloader

### 2.1 简介

HC32F4A0 系列的 Bootloader 程序的功能是通过 UART 外设将 RAMCODE 程序下载到 SRAM 中，以及运行下载的 RAMCODE。

### 2.2 硬件连接

支持 PB10、PB11 或 PA13、PA14 引脚通信。在可运行的最小系统环境内，与 HC32F4A0 系列的 Bootloader 程序通信所需的硬件连接如图 1 所示：

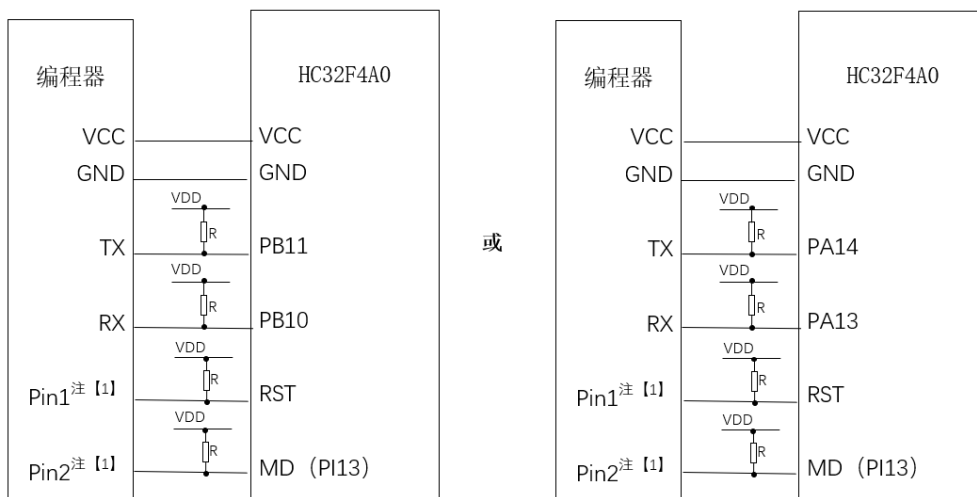


图 1 串行连接

注：

**【1】** 用于进入 Boot 模式（第 2.3 节将做介绍）

## 2.3 激活方式

Bootloader 程序在 Boot 模式下运行，按照图 1 所示做好硬件连接，并且按照图 2 所示时序进行操作，即可进入 Boot 模式从而激活 Bootloader 程序。

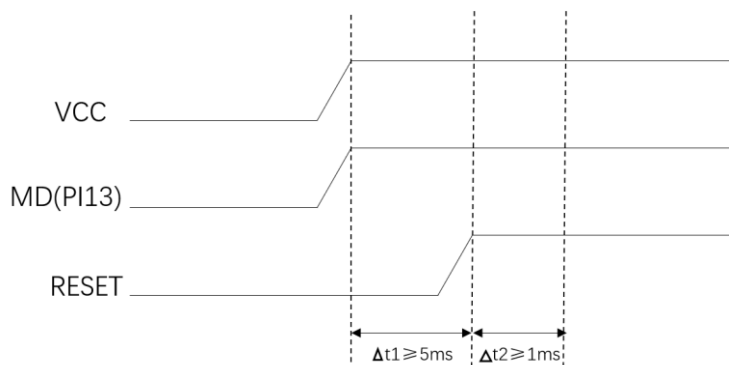


图 2 进入 Boot 模式的时序图

## 2.4 程序配置

HC32F4A0 系列的 Bootloader 程序初始化后使用的硬件资源及相关配置如下表所示：

功能/外设	状态	备注
PLL	使能	使用 PLL 作为系统时钟，频率为 240MHz
HRC	使能	使用 HRC 作为 PLL 时钟源
SRAM	-	自地址 0x1FFE0000 起至 0x1FFE7FFF 为 Bootloader 程序使用
UART3	使能	通信波特率：115200bps 通信配置：1 个起始位，8 个数据位，1 个停止位
UART4	使能	通信波特率：115200bps 通信配置：1 个起始位，8 个数据位，1 个停止位
UART3_RX 引脚	输入	PB11 引脚，用于 UART3 数据接收
UART3_TX 引脚	推挽输出	PB10 引脚，用于 UART3 数据发送
UART4_RX 引脚	输入	PA14 引脚，用于 UART4 数据接收
UART4_TX 引脚	推挽输出	PA13 引脚，用于 UART4 数据发送
SysTick 定时器	使能	用于通信超时机制

## 2.5 工作流程

通过 Bootloader 下载和运行 RAMCODE 使用到 3 类串行命令：握手命令、下载命令、跳转命令，工作流程如图 3 所示。

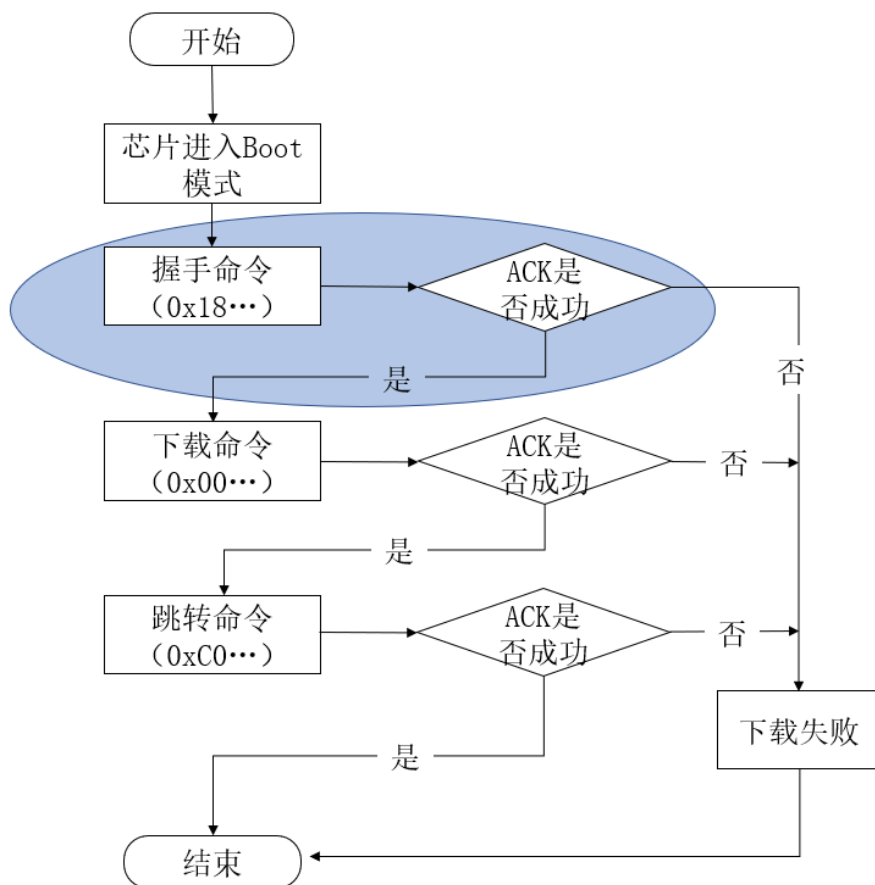


图 3 Bootloader 程序的工作流程

## 2.6 协议命令

### 2.6.1 握手命令

握手命令 0x18 用于编程器与目标芯片建立通信连接，为之后的数据下载做准备。编程器发送握手命令之后收到正确的 ACK 0x11，就表示与目标芯片成功建立了通信连接。

握手命令数据帧及对应 ACK 结构如下所示：

	编程器→目标芯片	目标芯片→编程器	
10 byte	握手命令（0x18 0x48 0x44 0x53 0x43 0x42 0x4F 0x4F 0x54 0x6E）		
1 byte		ACK	
		成功 校验错误	0x11 0x02

## 2.6.2 下载命令

下载命令 0x00 用于将 RAMCODE 程序下载到芯片内部 SRAM 相应位置，允许的下载地址范围为[0x1FFE8000, 0x2005FFFE]。

下载命令数据帧及对应 ACK 结构如下所示：

	编程器→目标芯片	目标芯片→编程器	
1 byte	下载命令 (0x00)		
4 byte	下载起始地址 (0x1FFE8000) <sup>【1】</sup>		
4 byte	RAMCODE 长度 <sup>【1】</sup> <sup>【2】</sup> (0XXXXXXXX)		
1 byte	命令校验和 (SUM1) <sup>【3】</sup>		
1 byte		ACK <sup>【5】</sup>	
		成功	0x01
		校验错误	0x02
		地址或长度错误	0x03
N byte	下载数据		
1 byte	数据校验和 (SUM2) <sup>【4】</sup>		
1 byte		ACK	
		成功	0x01
		校验错误	0x02

注：

**【1】** 下载起始地址和 RAMCODE 长度均按小端方式发送

如下载起始地址为 0x1FFE8000，则依次发送 0x00、0x80、0xFE、0x1F

**【2】** RAMCODE 长度：RAMCODE 程序的数据长度 (in byte)

**【3】** SUM1：下载命令、SRAM 起始地址、RAMCODE 长度这 9 bytes 数据按字节累加的结果

**【4】** SUM2：N bytes 的下载数据按字节累加的结果

**【5】** 发送 SUM1 之后，收到的 ACK 如果不是 0x01，则本次下载命令以失败结束，目标芯片不再接收剩余的 N+1 bytes 数据



### 2.6.3 跳转命令

跳转命令用于目标芯片程序跳转到 RAMCODE 运行。

- 命令检查成功后，目标芯片先返回 11 个 bytes 的固定数据 ACK “happycoding”，然后程序跳转至 RAMCODE 运行。
- 命令校验错误或检查失败后，目标芯片仅返回 1 个 byte 的错误或失败 ACK，程序不跳转至 RAMCODE 运行。

跳转命令数据帧及对应 ACK 结构如下所示：

	编程器→目标芯片	目标芯片→编程器	
1 byte	跳转命令 (0xC0)		
8 byte	虚拟数据 (全 0x00)		
1 byte	命令校验和 (0xC0)		
1 byte 或者 11byte		ACK	
		跳转命令检查成功，正在跳转到 RAMCODE 运行	字符串 “happycoding”
		校验错误	0x02
		跳转命令检查失败	0x04

### 2.6.4 ACK 列表

值	含义
0x11	握手成功
0x01	成功
0x02	校验错误
0x03	地址或长度错误
0x04	跳转命令检查失败，请确认握手命令之后的最近一次下载命令已成功完成
“happycoding”	跳转命令检查成功，正在跳转到 RAMCODE 运行

## 2.7 错误处理机制

### 2.7.1 接收到错误的命令数据帧

#### 1) 不支持的命令

Bootloader 仅支持握手、下载和跳转命令，接收到其它命令会直接忽略。

#### 2) 校验和检查失败

命令数据帧接收完成，且校验成功，才被判定为有效，若校验和检查失败，则 Bootloader 通过 ACK 指明发生校验错误。

校验方法：按字节累加接收到的数据，累加结果与命令数据帧中的校验和字段进行比较，比较不一致则校验失败。

### 2.7.2 发生通信错误和超时错误

#### 1) 通信错误：ORE 接收上溢错误、FE 接收帧错误。

#### 2) 超时错误：命令数据帧未接收完成的情况下，Bootloader 接收到一个数据之后超过 5 秒钟没有接收到下一个数据，或 Bootloader 回复下载命令有效 ACK 后，超过 5 秒钟没有接收到 RAMCODE 数据。

Bootloader 检测到以上错误后会放弃本次命令数据帧的接收，而进入等待下一命令数据帧的状态，但不返回任何 ACK。

编程器需自行对应通信错误和超时错误的处理。例如，编程器在等待 ACK 时进行超时判断，超时后没有接收到来自 Bootloader 的 ACK 则尝试重新发送，或提示编程失败，或重启目标板等处理。

## 2.8 注意事项

### 2.8.1 Boot 模式下访问 MCU Flash

进入 Boot 模式后，若想对 MCU Flash 资源进行正常访问，请执行以下步骤：

- 1) 解除 Flash 寄存器的写保护

```
M4_EFM->FAPRT = 0x00000123;
```

```
M4_EFM->FAPRT = 0x00003210;
```

- 2) 解除重映射寄存器的写保护

```
M4_EFM->MMF_REMPRT = 0x00000123;
```

```
M4_EFM->MMF_REMPRT = 0x00003210;
```

- 3) 设置地址重映射功能无效

```
bM4_EFM->MMF_REMCR0_b.EN0 = 0;
```

```
bM4_EFM->MMF_REMCR1_b.EN1 = 0;
```

### 3 总结

以上章节简要介绍了 HC32F4A0 系列的 Bootloader 程序，说明了编程器与 HC32F4A0 系列的 Bootloader 通信所需的硬件连接方式、Bootloader 的激活方式、Bootloader 使用到的硬件资源和相关配置、Bootloader 程序的工作流程和协议命令等。在实际开发中用户可以根据自己的需要，通过 Bootloader 下载 RAMCODE 并运行 RAMCODE，进而通过 RAMCODE 实现对 HC32F4A0 系列 MCU Flash 进行编程等操作。

## 4 版本信息 & 联系方式

日期	版本	修改记录
2021/7/27	Rev1.0	初版发布



---

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](mailto:mcu@hdsc.com.cn)

网址: [www.hdsc.com.cn](http://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

---

