32-bit microcontroller

# HC32L110 / HC32F003 / HC32F005

## series of Flash serial programming



## Suitable

| series | Product number |
|---|---|
| **HC32L110** | HC32L110C6UA<br>HC32L110C6PA<br>HC32L110C4UA<br>HC32L110C4PA<br>HC32L110B6PA<br>HC32L110B4PA |
| **HC32F003** | HC32F003C4UA<br>HC32F003C4PA |
| **HC32F005** | HC32F005C6UA<br>HC32F005C6PA<br>HC32F005D6UA |

# content

# 1 Summary

This article mainly introduces the use of serial programming mode to program the internal Flash of HC32L110 / HC32F003 / HC32F005 series steps and precautions.

# 2 Introduction to Bootloader

In order to facilitate customers to serially program the internal Flash of HC32L110 / HC32F003 / HC32F005 series products, The Bootloader is designed to implement various interfaces for customers to develop.

# 3 Serial programming mode

## 3.1 Communication Standards

Serial programming mode adopts standard UART (Universal Asynchronous Receiver/Transmitter) communication, 1 start bits, 8 data bits, 1 stop bit.

## 3.2 Hardware connection

The hardware conditions for serial programming of the internal Flash of HC32L110 / HC32F003 / HC32F005 series MCU are:

1) Within the minimum operating system environment.

2) The signal line on the programmer side needs to be connected to the four signal lines of VCC, GND, TX and RX as shown in Figure 1 (using HC32L110 series as an example).

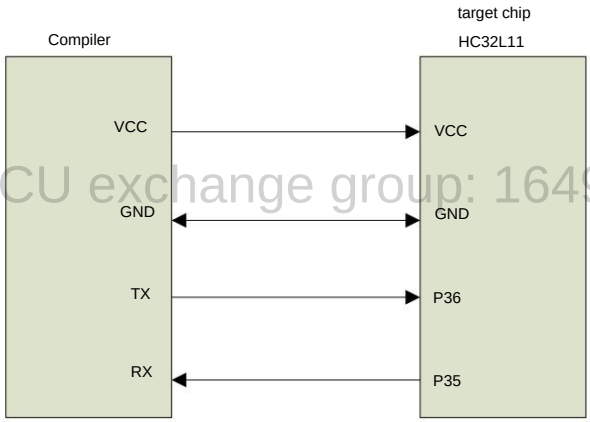




Figure 1 Serial programming connection

## 4 Flash programming process

The overall flow of serial programming to Flash is shown in Figure 2. The programmer uses the baud rate of 9600 to program the target chip RAMCODE download and baud rate reset, then use the reset baud rate for flash operation.

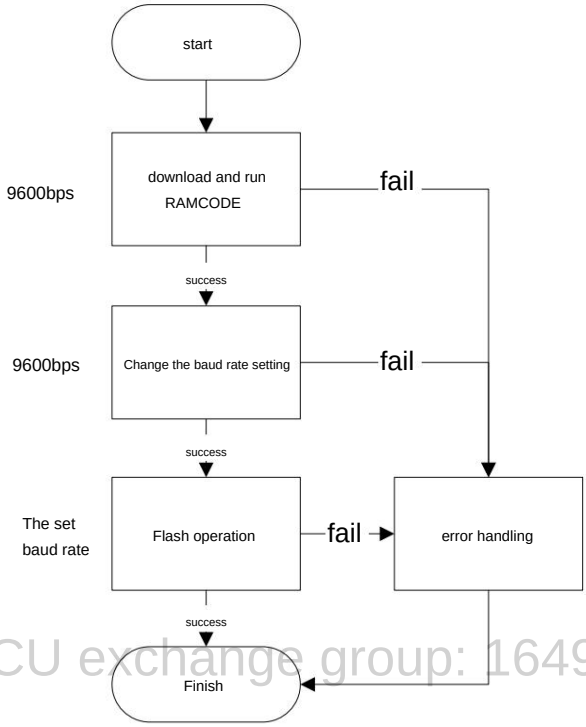




Figure 2 The overall flow of serial programming

## 4.1 Download and run RAMCODE

The process of downloading and running RAMCODE is shown in Figure 3.

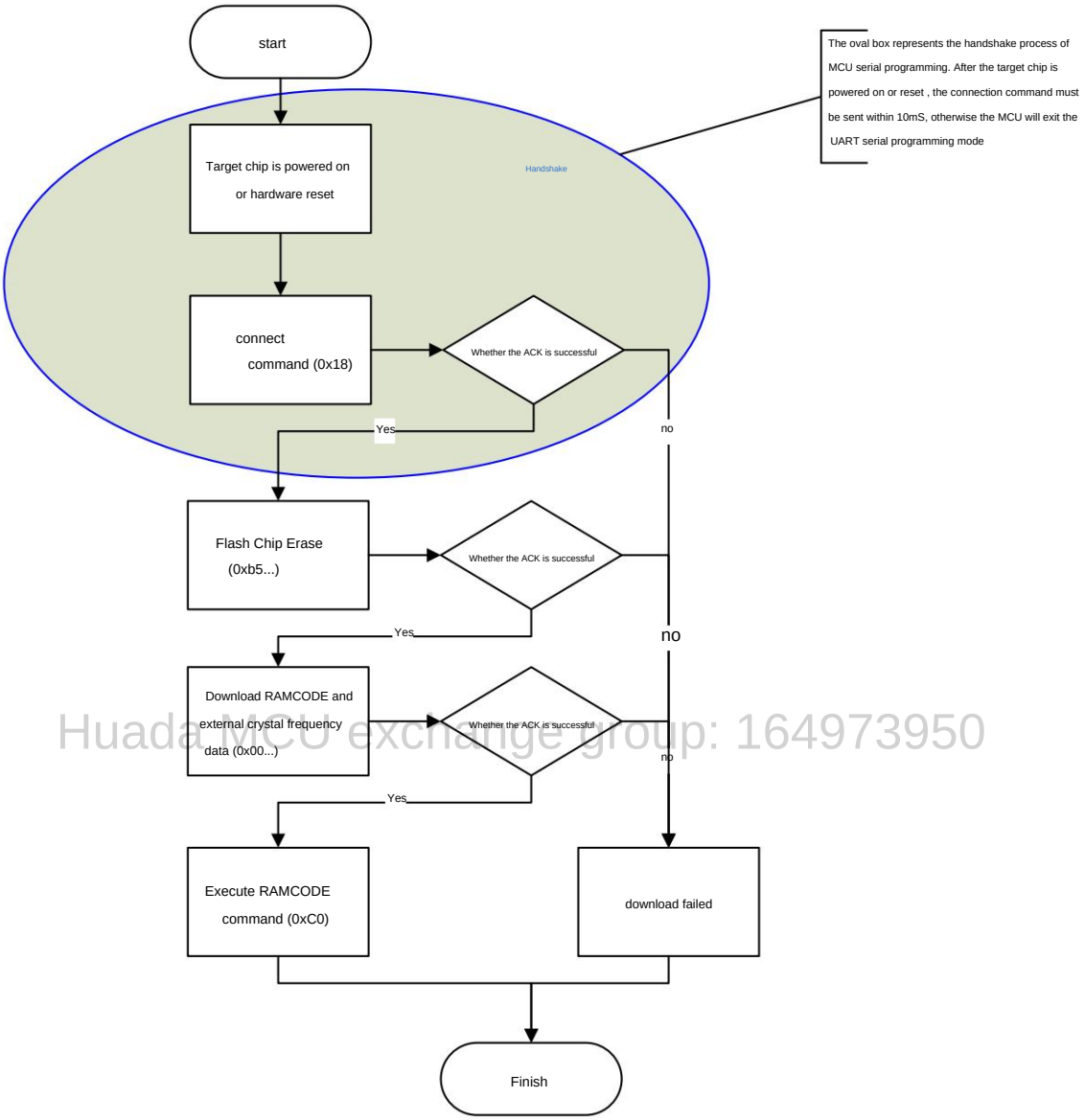




Figure 3 Download and run the RAMCODE process

Download and run RAMCODE using 4 serial commands: connect command, download command, execute command, full erase command make.

### 4.1.1 Connection Command

The connection command 0x18 establishes a communication connection with the target chip to prepare for the subsequent data download. The connection establishment process requires a

Send multiple times within a certain time until the correct ACK 0x11 is received, indicating that the target chip is ready.

The connection command data frame and the corresponding ACK structure are as follows:

| | ProgrammerÿTarget Chip | Target chipÿProgrammer | |
|---|---|---|---|
| 1 byte | connect command (0x18) | | |
| 1 byte | | ACKÿ | |
| | | success | ÿ0x11 |

## 4.1.2 Download RAMCODE

The download command 0x00 downloads the RAMCODE data and the external crystal oscillator frequency data to the corresponding position in the internal RAM of the MCU. Down

The command data frame and the corresponding ACK structure are as follows:

| | ProgrammerÿTarget Chip | Target chipÿProgrammer | |
|---|---|---|---|
| 1 byte | Download command (0x00) | | |
| 4 byte | RAM start address (0x20000000) | | |
| 4 byte | RAMCODE file length<br>ÿ0x00000XXXÿ | | |
| 1 byte Command Checksum (SUM1) | | | |
| 1 byte | | ACKÿ | |
| | | success | ÿ0x01 |
| | | Checksum error | ÿ0x02 |
| N byte download data | | | |
| 1 byte data checksum (SUM2) | | | |
| 1 byte | | ACKÿ | |
| | | success | ÿ0x01 |
| | | Checksum error | ÿ0x02 |



Note:

- The RAM start address and RAMCODE length data in the table are sent in little-endian mode. The starting address is

  0x20000000 sends 0x00, 0x00, 0x00, 0x20 in sequence

- RAMCODE length: provide the data length of RAMCODE file for Huada Semiconductor (in byte)

- SUM1: refers to the 9-byte data of download command, RAM start address, and RAMCODE length, which are accumulated byte by byte result

- SUM2: refers to the result of accumulating the download data of N bytes in bytes

### 4.1.3 Execute command

RAMCODE starts to run after command 0xC0 is sent, if the command is sent and RAMCODE runs correctly, ACK will Returns 11 bytes of random data, the data frame and the corresponding ACK structure are shown below.

| | ProgrammerÿTarget Chip | Target chipÿProgrammer | |
|---|---|---|---|
| 1 byte | Execute command (0xC0) | | |
| 8 byte dummy data (all 0x00) | | | |
| 1 byte command checksum (0xC0) | | | |
| 1 byte or 11byte | | ACKÿ | |
| | | execution succeed | : 11 bytes of random data |
| | | Checksum error | ÿ0xC2 |

### 4.1.4 Full Erase Command



The full erase command is used to erase the Flash in the internal user area of the MCU during serial programming. The data frame and corresponding ACK structure are as follows shown.

| | ProgrammerÿTarget Chip | Target chipÿProgrammer | |
|---|---|---|---|
| 1 byte Command code 1 (0xB5) | | | |
| 1 byte Command code 2 (0x34) | | | |
| 1 byte Command code 3 (0x84) | | | |
| 1 byte Command code 4 (0x52) | | | |
| 1 byte | Check code (0xBF) | | |
| 1 byte | | ACKÿ | |
| | | execution succeed | ÿ0x01 |
| | | mistake | ÿ0x02 |

## 4.2 Change the baud rate setting

After RAMCODE is downloaded and run successfully, the API (application interface) implemented by RAMCODE takes effect, and its API interface

Please refer to Appendix 1: RAMCODE API for the format and supported commands.

The programmer can reconfigure the baud rate of the target system through API commands to improve programming efficiency. It is recommended to configure the baud rate.

115200bps. (Currently, the baud rates supported by this series of RAMCODE are 9600, 19200, 38400, 115200bps)

Command example

For example, if the baud rate is set to 115200, the command frame and the ACK data of the successful baud rate setting are as follows:

| ProgrammerÿTarget Chip | | Target chipÿProgrammer | |
|---|---|---|---|
| Start code | 0x49<br>0x53 | | |
| Frame length | 0x00<br>0x09 | | |
| CMD | 0x01 | | |
| Address | 0x00<br>0x00<br>0x00<br>0x00 | | |
| Data length | 0x00<br>0x01 | | |
| Baudrate | 0x06 | | |
| Checksum | 0x11 | | |
| | | Start code | 0x49<br>0x53 |
| | | Frame length | 0x00<br>0x07 |
| | | CMD | 0x01 |
| | | Status | 0x00 |
| | | Address | 0x00<br>0x00<br>0x00<br>0x00 |
| | | Checksum | 0x08 |

## 4.3 Flash operation

After changing the baud rate setting successfully, the serial communication performed by the Flash operation process is performed with the latest configured baud rate.

The Flash operation flow is shown in Figure 4.

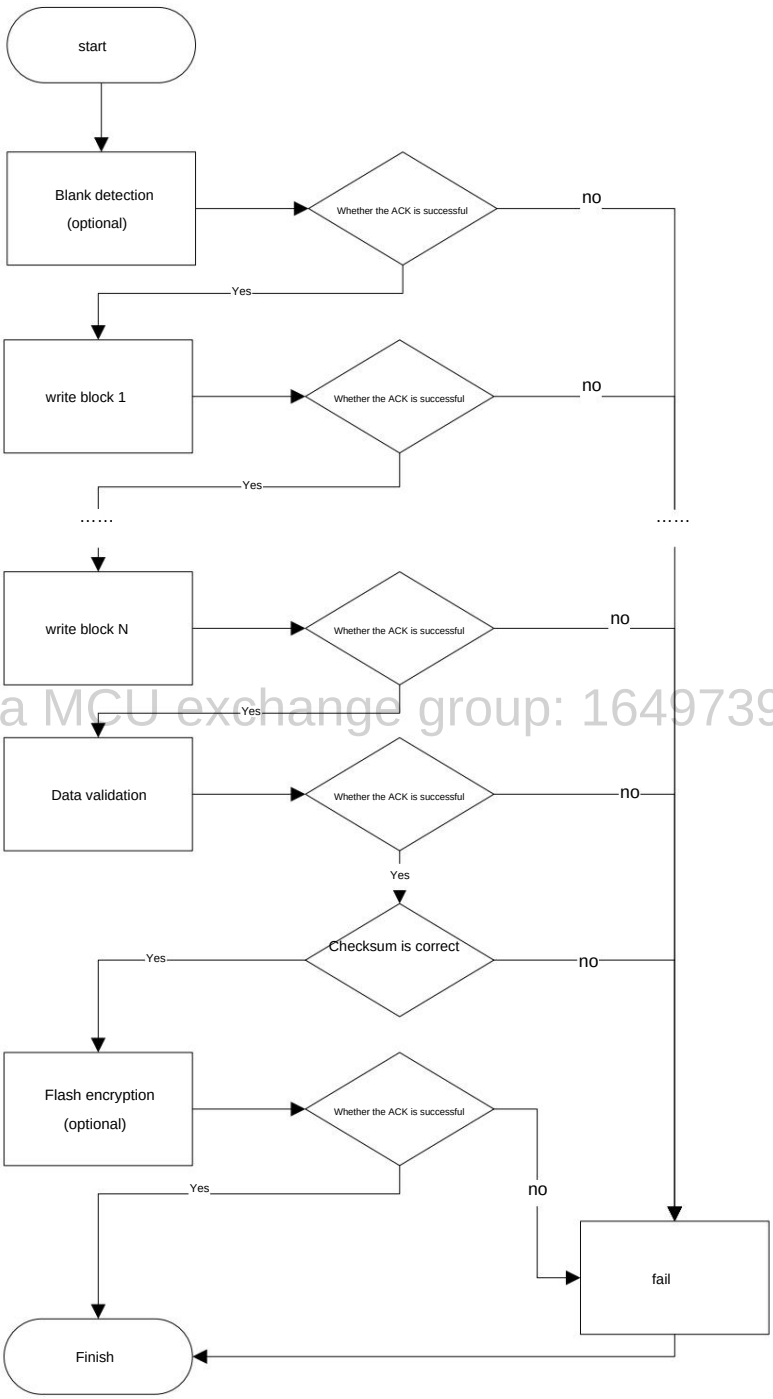




Figure 4 Flash operation flow

### 4.3.1 Flash Blank Detection

Flash blank detection is an optional step. After the detection, ACK will return whether the detected area is blank (full FF).

Command example

After the chip erase is successful, the blank detection and ACK data of the entire flash data are as follows:

| ProgrammerÿTarget Chip | | Target chipÿProgrammer | |
|---|---|---|---|
| Start code | 0x49<br>0x53 | | |
| Frame length | 0x00<br>0x0C | | |
| CMD | 0x07 | | |
| Check address | 0x00<br>0x00<br>0x00<br>0x00 | | |
| Data length | 0x00<br>0x04 | | |
| Check Lengh | 0x00<br>0x02<br>0x00<br>0x00 | | |
| Checksum | 0x19 | | |
| | | Start code | 0x49<br>0x53 |
| | | Frame length | 0x00<br>0x08 |
| | | CMD | 0x07 |
| | | Status | 0x00 |
| | | Check address | 0x00<br>0x00<br>0x00<br>0x00 |
| | | Result | 0x01 |
| | | Checksum | 0x10 |

### 4.3.2 Flash Write

Flash write Write the data to be programmed to the corresponding position of the Flash, and can be subpackaged and written multiple times with the Flash write command.

Command example

Write a packet of data with a data length of 64Byte to the Flash address 0x0200 and the ACK data as follows:

| ProgrammerÿTarget Chip | | Target chipÿProgrammer | |
|---|---|---|---|
| Start code | 0x49<br>0x53 | | |
| Frame length | 0x00<br>0x48 | | |
| CMD | 0x04 | | |
| Write address | 0x00<br>0x00<br>0x02<br>0x00 | | |
| Write length | 0x00<br>0x40 | | |
| Write data | Omitted (64byte) | | |
| Checksum | 0xE3 | | |
| | | Start code | 0x49<br>0x53 |
| | | Frame length | 0x00<br>0x07 |
| | | CMD | 0x04 |
| | | Status | 0x00 |
| | | Write address | 0x00<br>0x00<br>0x02<br>0x00 |
| | | Checksum | 0x0d |



Note:

- In the command frame, Write data is the 64-byte data that needs to be written into the Flash.

ÿ In the command frame, Checksum is the value in this example. Please calculate according to the frame structure in the appendix in the application.

### 4.3.3 Read Flash Checksum

Read Flash checksum command start address and detection length can be set. After reading, ACK will return to the detected area checksum (accumulated in bytes).

Command example

Flash starts from address 0 and has a length of 3610 bytes. Checksum and ACK data are as follows:

| ProgrammerÿTarget Chip | | Target chipÿProgrammer | |
|---|---|---|---|
| Start code | 0x49<br>0x53 | | |
| Frame length | 0x00<br>0x0C | | |
| CMD | 0x06 | | |
| Check address | 0x00<br>0x00<br>0x00<br>0x00 | | |
| Data length | 0x00<br>0x04 | | |
| Check Lengh | 0x00<br>0x00<br>0x0E<br>0x1A | | |
| Checksum | 0x3E | | |
| | | Start code | 0x49<br>0x53 |
| | | Frame length | 0x00<br>0x09 |
| | | CMD | 0x06 |
| | | Status | 0x00 |
| | | Check address | 0x00<br>0x00<br>0x00<br>0x00 |
| | | Checksum result | 0xEC<br>0xB2 |
| | | Checksum | 0xAD |



Note: Checksum result is the checksum in the Flash check area. In this example, the checksum is 0xECB2.

### 4.3.4 Flash encryption settings

Flash encryption setting command, encrypt the flash of the target chip, after the success, the RAMCODE and SWD interface cannot be read Flash internal data.

Command example

| ProgrammerÿTarget Chip | | Target chipÿProgrammer | |
|---|---|---|---|
| Start code | 0x49<br>0x53 | | |
| Frame length | 0x00<br>0x08 | | |
| CMD | 0x09 | | |
| Address | 0x00<br>0x00<br>0x00<br>0x00 | | |
| Data length | 0x00<br>0x00 | | |
| Checksum | 0x11 | | |
| | | Start code | 0x49<br>0x53 |
| | | Frame length | 0x00<br>0x07 |
| | | CMD | 0x09 |
| | | Status | 0x00 |
| | | Address | 0x00<br>0x00<br>0x00<br>0x00 |
| | | Checksum | 0x10 |

## 4.4 Error Handling

As shown in Figure 2, RAMCODE download and other processes may fail.

Flash programming of the target chip requires power-off or hardware reset of the chip and re-handshake and RAMCODE download process.

The suggested error handling process is shown in Figure 5. Either mode 1 or mode 2 can be selected to restore the state of the target chip.

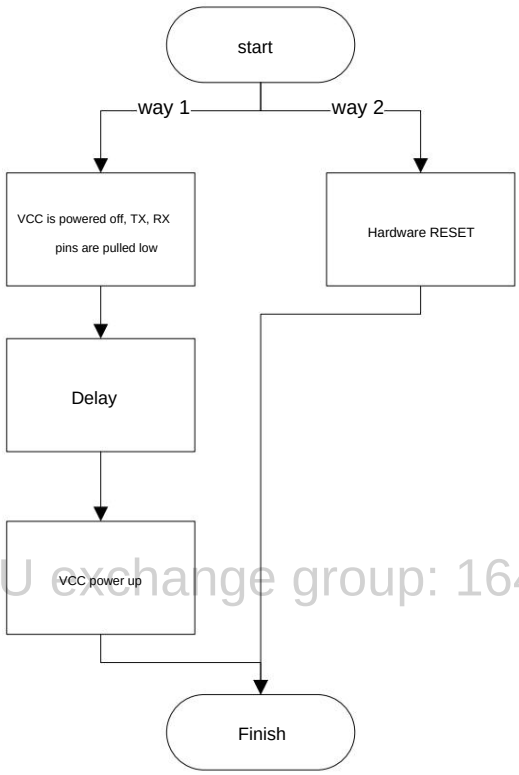




Figure 5 Error handling

Note:

- The length of the delay time is related to the hardware environment of the board, and it needs to meet the complete VCC of the system where the MCU is located within the delay time. until power down.
- Hardware RESET refers to the reset through the hardware reset pin of the MCU. Please refer to the specifications of the specific chip for the reset timing requirements. Book.

# Appendix 1: RAMCODE API Interface

**command frame**

The command frame refers to the data frame sent by the upper computer (PC or programmer) to the lower computer (MCU).

| D0D1 | D2D3 | D4 | D5D6D7D8 D9D10 … | | | Dn |
|---|---|---|---|---|---|---|
| Start code | Frame Length | CMD Address | | Data Length | Data (optional) | Check Sum |
| 0x4953 XXXX XX | XXXXXXXX XXXX XX…XX XX | | | | | |

Start code: Data frame start code 0x4953, D0 – 0x49, D1 – 0x53 (mentioned in the RAMCODE API interface word, long data are sent according to this big endian structure)

Frame Length: The length of the data frame, the number of bytes from CMD to Checksum

CMD: Command code, defined as follows:

| CMD command | | CMD | Order |
|---|---|---|---|
| 0x01 | Serial port baud rate setting | 0x06 | Read Flash checksum |
| 0x02 | Flash chip erase | 0x07 | Flash whitespace check |
| 0x04 | Flash write | 0x09 | Flash encryption settings |



Addressÿ Flash address, this field is invalid for some commands, it is 0

Data Length: data length, this field is invalid for some commands, it is 0

Data: data, some commands may default to this item

Check Sum: Data frame checksum (Frame Length, CMD, Address, Data Length, Data by byte add).

## ACK frame

The ACK frame refers to the data frame returned by the lower computer (MCU) to the upper computer (PC).

| D0D1 | D2D3 | D4 | D5 | D6D7D8D9 … | | Dn |
|---|---|---|---|---|---|---|
| Start code | Frame Length | CMD Status | | Address | Data (optional) | Check Sum |
| 0x4953 XXXX | XX | | XX | XXXXXXXX XX…XX | XX | |

Start code: same as command frame

Frame Length: same as command frame

CMDÿ same as command frame

Statusÿ ACK status

0 – success;

1 – Command frame check error;

2 – unsupported command code;

3 – Flash address is not in the supported range

4 - Frame length not within supported range

5 – Data length not within supported range

6 – Baud rate not supported;

7 – In encrypted state, read, write or page erase operations are prohibited;

Address: same as command frame

Data: same as command frame

Check Sum : same as command frame

1) Serial port baud rate setting

command frame

| D0D1 | D2D3 | D4 | D5D6D7D8 D9D10 | | D11 | D12 |
|---|---|---|---|---|---|---|
| Start code | Frame Length | CMD Address | | Data Length | Data (optional) | Check Sum |
| 0x4953 | 0x0009 | 0x01 | 0x00000000 0x0001 | | Baud rate XX | |

Baud rate: The baud rate set by the host computer, the detailed definition is as follows:

| Baud rate | baud rate |
|---|---|
| 0x01 | 9600 |
| 0x02 | 14400 |
| 0x03 | 19200 |
| 0x04 | 38400 |
| 0x05 | 57600 |
| 0x06 | 115200 |
| 0x07 | 128000 |
| 0x08 | 76800 |
| 0x09 | 256000 |



**ACK** frame

| D0D1 | D2D3 | D4 | D5 | D6D7D8D9 | | D10 |
|---|---|---|---|---|---|---|
| Start code | Frame Length | CMD Status | | Address | Data (optional) | Check Sum |
| 0x4953 | 0x0007 | 0x01 | Std | 0x00000000 | N/A | XX |

Std: 0 - Indicates that the command frame was received correctly

non-0 - failure (see ACK frame description for details)

2) Flash chip erase

command frame

| D0D1 | D2D3 | D4 | D5D6D7D8 D9D10 | | | D11 |
|---|---|---|---|---|---|---|
| Start code | Frame Length | CMD Address | | Data Length | Data (optional) | Check Sum |
| 0x4953 | 0x0008 | 0x02 | 0x00000000 0x0000 | | N/A | XX |

**ACK** frame

| D0D1 | D2D3 | D4 | D5 | D6D7D8D9 | | D10 |
|---|---|---|---|---|---|---|
| Start code | Frame Length | CMD Status | | Address | Data (optional) | Check Sum |
| 0x4953 | 0x0007 | 0x02 | Std | 0x00000000 | N/A | XX |

Std: 0 - Chip Erase Successful

non-0 - failure (see ACK frame description for details)

3) Flash write

command frame



| D0D1 | D2D3 | D4 | D5D6D7D8 D9D10 | | D11 ~ Dn-1 Dn | |
|---|---|---|---|---|---|---|
| Start code | Frame Length | CMD Address | | Data Length | Data (optional) | Check Sum |
| 0x4953 XXXX 0x04 | | | Write Adr | Write Length | Write Data XX | |

Write Adrÿ Flash write start address

Write Length: The length of data written by Flash

Write Dataÿ Data written by Flash

**ACK** frame

| D0D1 | D2D3 | D4 | D5 | D6D7D8D9 | | D10 |
|---|---|---|---|---|---|---|
| Start code | Frame Length | CMD Status | | Address | Data (optional) | Check Sum |
| 0x4953 | 0x0007 | 0x04 | Std | Write Adr | N/A | XX |

Std: 0 – Flash write successfully

non-0 - failure (see ACK frame description for details)

Write Adr: Flash write start address

4) Read Flash checksum

command frame

| D0D1 | D2D3 | D4 | D5D6D7D8 D9D10 | | D11D12D13D 14 | D15 |
|---|---|---|---|---|---|---|
| Start code | Frame Length | CMD Address | | Data Length | Data (optional) | Check Sum |
| 0x4953 | 0x000C | 0x06 | Check Adr | 0x0004 | Check Length XX | |

Check Adr: The starting address of the flash for calculating the checksum

Check Length: Calculate the length of the checksum (in byte)

**ACK** frame

| D0D1 | D2D3 | D4 | D5 | D6D7D8D9 | D10 ~ D11 D12 | |
|---|---|---|---|---|---|---|
| Start code | Frame Length | CMD Status | | Address | Data (optional) | Check Sum |
| 0x4953 | 0x0009 | 0x06 | Std | Check Adr | Checksum XX | |

Std: 0 – Computation of checksum succeeded

non-0 - failure (see ACK frame description for details)

Read Adr: The starting address of the flash for calculating the checksum



Checksum: Calculated checksum

5) Flash blank check

command frame

| D0D1 | D2D3 | D4 | D5D6D7D8 D9D10 D11D12D13D14 D15 | | | |
|---|---|---|---|---|---|---|
| Start code | Frame Length | CMD Address | | Data Length | Data (optional) | Check Sum |
| 0x4953 | 0x000C | 0x07 | Check Adr | 0x0004 Check Length XX | | |

Check Blank Adr: Flash start address for blank check

Check Blank Length: length of blank check (in byte)

**ACK** frame

| D0D1 | D2D3 | D4 | D5 | D6D7D8D9 | D10 | D11 |
|---|---|---|---|---|---|---|
| Start code | Frame Length | CMD Status | | Address | Data (optional) | Check Sum |
| 0x4953 | 0x0008 | 0x07 | Std | Check Adr | Result | XX |

Std: 0 – Blank check succeeded

non-0 - failure (see ACK frame description for details)

Check Adr: Flash start address for blank check



Result: Blank check result

TRUE(1) – check range blank

FALSE(0) – check range is not blank

6) Flash encryption settings

command frame

| D0D1 | D2D3 | D4 | D5D6D7D8 D9D10 | | | D11 |
|---|---|---|---|---|---|---|
| Start code | Frame Length | CMD Address | | Data Length | Data (optional) | Check Sum |
| 0x4953 | 0x0008 | 0x09 | 0x00000000 0x0000 | | N/A | XX |

**ACK** frame

| D0D1 | D2D3 | D4 | D5 | D6D7D8D9 | | D10 |
|---|---|---|---|---|---|---|
| Start code | Frame Length | CMD Status | | Address | Data (optional) | Check Sum |
| 0x4953 | 0x0007 | 0x09 | Std | 0x00000000 | N/A | XX |

Std: 0 - encryption set successfully

non-0 - failure (see ACK frame description for details)

# 5 Summary

The above chapters briefly introduced Bootloader serial programming and explained HC32L110 / HC32F003 / HC32F005 in detail Series of Bootloader programming interface, users can use the Bootloader according to their actual needs during development.

## 6 Version Information & Contact Information

| date | Version revision record |
|---|---|
| 2018/6/4 | Rev1.0 initial release |
| | |
| | |

If you have any comments or suggestions in the process of purchasing and using, please feel free to contact us.

Email:ymcu@hdsc.com.cn



Website: www.hdsc.com.cn

Mailing address: No. 39, Lane 572, Bibo Road, Zhangjiang Hi-Tech Park, Shanghai

Postcode: 201203