ISP Program Manual with UART

Ver:2.6

Contents

1 Operation Overview 3
1.1 Operation Flow Chat 3
1.2 Process Description 5
1.3 Hardware Configuration 5
2 Communication Protocol 6
2.1 Host Transfer Data Format 6
2.2 Chip System Program Send Data Format 7
2.3 Chip SRAM Program send Data Format 8
3 Implementation of the Protocol 9
3.1 Partial Instruction Data Flow 9
3.1.1 Handshake 9
3.1.2 Get Version and Configure Compression Baud Rate 9
3.1.3 Loading Configuration Information 9
3.1.4 Loading Configuration Data is not Last 9
3.1.5 Loading Configuration Data Last 9
3.1.6 Download Information 9
3.1.7 Download Program Data 10
3.1.8 Chip Encryption 10
3.1.9 Chip Initialization 10
3.1.10 Baud Rate Configuration 10
3.1.11 Program Jump 10
3.2 Additive Check 11
4 ISP Firmware Revision List 12
5 Document Revision History 13

1 Operation Overview

1.1 Operation Flow Chat

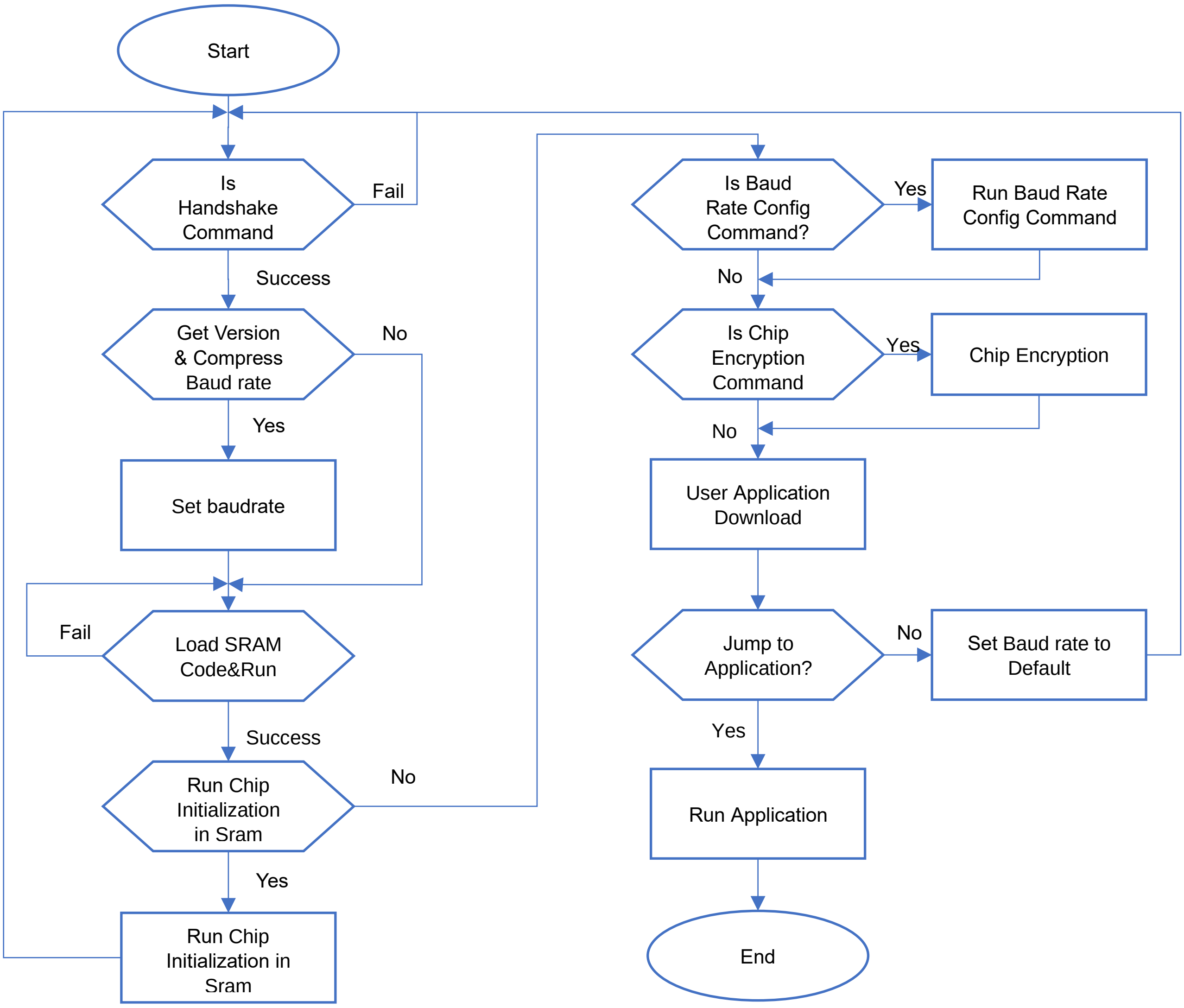
Start
Is Handshake Command
Fail
Success
Get Version & Compress Baud rate
No
Yes
Set baudrate
Fail
Load SRAM Code&Run
Success
Run Chip Initialization in Sram
No
Yes
Run Chip Initialization in Sram
Is Baud Rate Config Command?
Yes
Run Baud Rate Config Command
No
Is Chip Encryption Command
Yes
Chip Encryption
No
User Application Download
Jump to Application?
No
Set Baud rate to Default
Yes
Run Application
End

Figure 1-1-1: Program Work Flow Chat

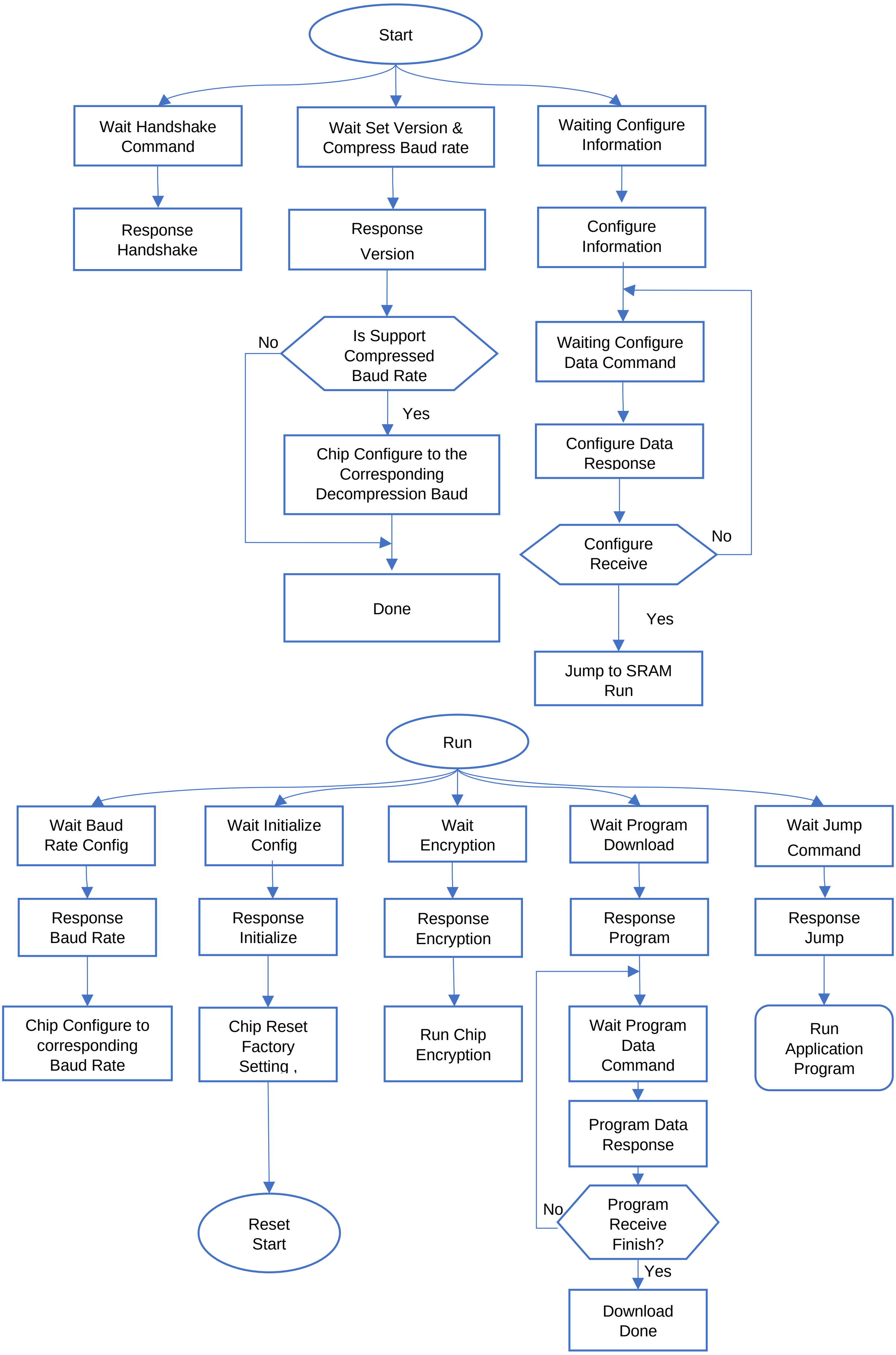

Figure 1-1-2: Communication thru system/SRAM program protocol

1.2 Process Description

Host side：According to section 1.3 configuration serial port, according to the second section Host to send instructions, reference 1.1 send process to obtain or configure the specified information to the MCU, complete the entire user program download process.

Device side：MCU is divided into two parts of the protocol, two-part protocol for the Run of two pieces of code, code 1 in the system space, follow the system space program protocol, code 2 in the SRAM space, follow the SRAM space protocol, wherein code 2 needs to run code 1 protocol to copy 2 files to SRAM, the run of the copy process instruction is called “load configuration”, and the program is automatically run by jumping from the code 1 area to the code 2 space after the configuration load is complete. MCU in code 2 in accordance with the SRAM run protocol to complete the entire user program download. The new version of code 1 supports the porter rate switching, can improve the code 2handling speed, the specific mode of communication see section 2.

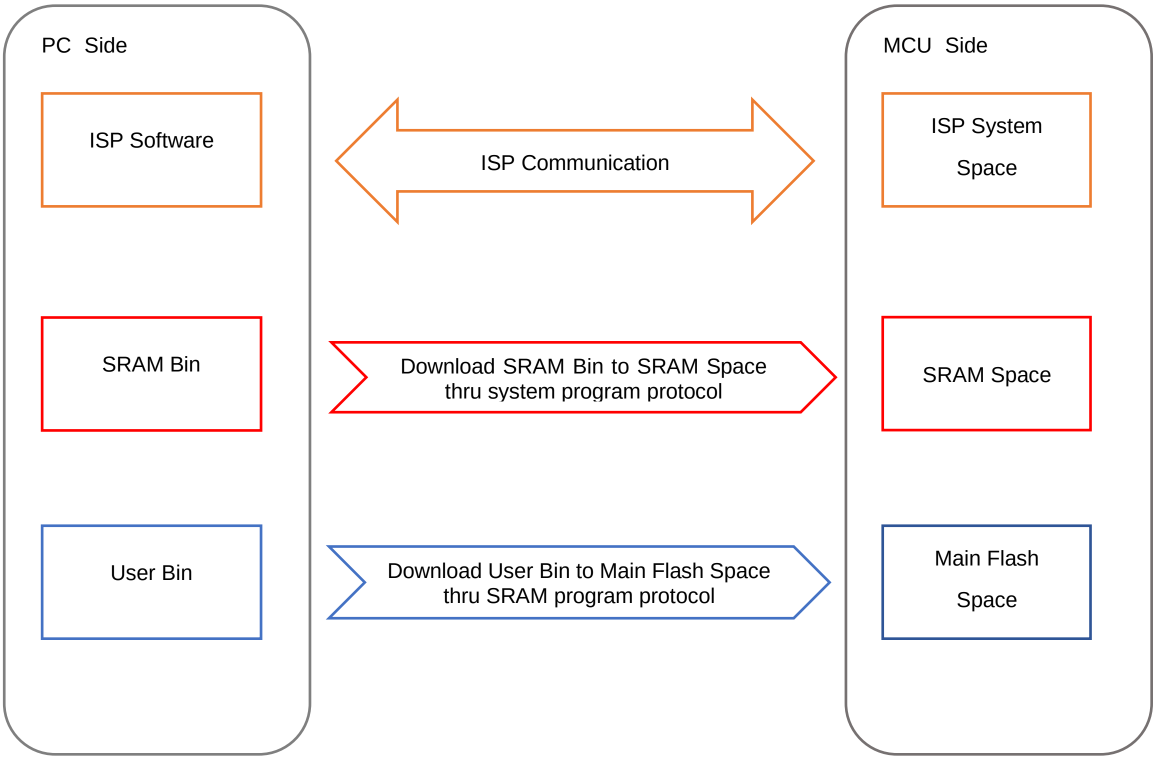
PC Side
ISP Software
SRAM Bin
User Bin
ISP Communication
Download SRAM Bin to SRAM Space thru system program protocol
Download User Bin to Main Flash Space thru SRAM program protocol
MCU Side
ISP System Space
SRAM Space
Main Flash Space

Figure 1-2-1：Program Data Flow Chat

1.3 Hardware Configuration

1. Device BOOT0 pulls high, BOOT1 pulls low.

2. The host serial port is configured to be 8-bit, stop 1-bit, no parity.

2 Communication Protocol

2.1 Host Transfer Data Format

Table 2-1：Host Transfer Protocol

Header **1 byte**	**Data Length** **2~3 byte**	**4 byte**	**…**	**…**	**…**	**n-1 byte**	**Cumulative and checksum**
ASCII 'P'	VAL_FRE_LEN (Actual Data Length)	0x00 (Handshake command)[3]	None	None	None	None	XX
	0x10D (269)[5]	0x01 (FLASH Download)	0x00 (Program Information 4B)	Start Address (4B)[1]	Total Program Length (4B)[1][5]	0x00(1B)[11]	XX
	0x10D (269)[5]		Total Packages Sent (4B)[1]	Current number of package (4B)[1]	Data (N)[4]	…	XX
	0x10D (269)[5]	0x02 (Loading Configure)[1][3]	0x00 (Configure Information 4B)[1]	Start Address (4B)[1][8]	Data Length (4B)[1][2]	…	XX
	0x10D (269)[5]		0x01 (Send configure 4B)[1]	Current Data Address (4B)[1][8]	Data (N)[4]	…	XX
	0x10D (269)[5]		0x02 (Configure Finish 4B)[1]	Current Data Address (4B)[1]	Data (N)[4]	…	XX
	VAL_FRE_LEN (Actual Data Length)	0x03 (Configure Baud Rate)	Baud Rate(4B)[1]				XX
		0x05 (Write Register)	0x00 (Reserved 4B)	Address (4B)	Data (4B)	None	XX
		0x06 (Read Register)	0x00 (Reserved 4B)	Address (4B)	None	None	XX
		0x07 (Write Memory)[10]	0x00 (Not Erase 2B) 0x01 (Page Erase2B)	Length (2B)	First Address (4B)	Data（N）	XX
		0x09 (Program Jump)	Jump Address(4B)[1]	None	None	None	XX
		0x0A (Chip Encryption)	Data (16B)	None	None	None	XX
		0x0F (Get Check Value)	0x00 (Reserved 4B)	None	None	None	XX
		0x20 (Get ISP Version)[1][3]	0x00(1B)	None	None	None	XX
			0x03(1B)[6]	Compression Baud Rate (1B)[7]			
		0x21 (Get configure Version)[2]	None	None	None	None	XX
		0x5A (Chip Initialization)[9]	0x00 (Reserved 4B)	None	None	None	XX

Note: 1. When the instruction (byte 4) is 0x1, 0x2, 0x3, 0x09, the data is sent in large-end mode, and the rest is in small-end mode (e.g. command 0x03, want to modify the baud rate of 14400, 16 to modify the bautid rate of 0x3840, and actually send should be 0x00, 0x00, 0x8, 0x40); 4B means four Bytes;

2. When the instruction (byte 4) is 0x02, the data "data length" is the number of bytes given by the official download profile (bin).

3. These commands can only be used until the download configuration data is sent. The program code can only be downloaded after Download configuration file.(Important)

4. The "program data" length per send is fixed at 256 bytes, so the maximum length per package is 269 bytes.

5. The "total program length" in the download program refers to the total number of program data, the total number of packages sent refers to the number of packages packed with 256 data, "total number of packages sent" = "the total length of the program" / 256+1

6. The compression baud rate function can only by supported if the ISP version number is higher than V022.

7 .”Compressed baud rate” = “baud rate to be updated” / 2400. For example, if you want to update the baud rate of 115200, then the compression baud rate should be transmitted by the number of 115200/2400=48, converted to 16-decimal, i.e.0x30. It is important to note that the baud rate supported by this command can only be used if it can be divided by 2400.

8. The “start address” in the configuration is fixed at 0x20000400, and for each successful lying of a data “current address” plus 0x100, for example, after the “current data address” is sent 0x20000400, the second send “current data address” should be 0x20000500, the maximum is 0x200000f00.

9. De-chip protection and erasing Flash user space, chip receipt of this command will result in reset, if the need to continue to use is ISP function needs to reload the configuration.

10. To write the number of data in Flash, in bytes, for example, there are 4 data to write: 0x12, 0x13, 0x14, 0x15, then the length should be 0x04.

11. ”Total program length(4B)” followed by a byte “0x00” cannot be omitted.

2.2 Chip System Program Send Data Format

Table 2-2：Chip System Transfer Protocol

Header **1 byte**	**Data Length** **2~3 byte**	**4 byte**	**…**	**…**	**…**	**n-1** **byte**	**Cumulative and checksum**
ASCII ‘S’	VAL_FRE_LEN (Actual Data Length)	0x00 (Handshake command)	0x00	None	None	None	XX
		0x02 (Loading Configure)	0x01 (Go on 1B)	None	None	None	XX
			0x02 (Finish 1B)[3]	None	None	None	XX
		0x20 (Get ISP Version)	ISP Version(4B)	None	None	None	XX
				0x03(1B)[1]	Compression Baud Rate (1B)[2]		

Note: 1. Older version ISP protocol does not support the compression baud rate function in command 0x20, see is the description of the ISP version on the tail page for details. 1B means one Bytes;

2. Command 0x20 gets the compressed baud rate in the ISP version. For example, if you want to update the baud rate of 115200, then the compression baud rate should be transmitted by the number of 115200/2400 = 48, converted to 16-decimal, i.e. 0x30.

3. After the configuration file(SRAM program) is loaded, the run program is performed to the configuration address, where the protocol adheres to the chip SRAM send code.(Important)

2.3 Chip SRAM Program send Data Format

Table 2-3：Chip SRAM Transfer Protocol

Header 1 byte	Data Length 2~3 byte	4 byte	…	…	…	n-1 byte	Cumulative and checksum
ASCII ‘S’	VAL_FRE_LEN (Actual Data Length)	0x00 (Compatible)	0x00	None	None	None	XX
		0x01 (FLASH Download)	0x00 (Program Information 4B)	Start Address (4B)	Total Program Length (4B)	None	XX
			Total Packages Sent (4B)	Current number of packages (4B)	None		XX
		0x02 (Compatible)	0x01 (Go on1B)	None	None	None	XX
			0x02 (Finish 1B)	None	None	None	XX
		0x03 (Baud Rate Configure)	XX（Reserved 4B）				
		0x05 (Write Register)[2]	0x00 (Reserved 4B)	None	None	None	XX
		0x06 (Read Register) [2]	Return Data (4B)	None	None	None	XX
		0x07 (Write Memory) [2]	0x00 (Reserved 4B)	First Address（4B）	Data（N）	None	XX
		0x09 (Program Jump)	0x00 (Reserved 4B)	None	None	None	XX
		0x0A (Chip Encryption) [2]	0x01 (Success 4B) 0x02 (Failed 4B)	None	None	None	XX
		0x0F (Get Check Value)	0x00 (Reserved 4B)	0x00 (Program Accumulate sum 4B)	None	None	XX
		0x20 (Compatible)	None	None	None	None	XX
				0x03 (1B)	Compression Baud Rate (1B)		
		0x21 (Get configure Version)	Configure Version (8B)	None	None	None	XX
		0x5A (Chip Initialization)	0x00 (Reserved 4B)	None	None	None	XX

Note:

1. This agreement takes effect after the system space jumps to SRAM. 4B means four Bytes;

2. These directives are not currently in effect in M0.

3 Implementation of the Protocol

3.1 Partial Instruction Data Flow

3.1.1 Handshake

Host: 0x50 0x00 0x05 0x00 0x55 // Handshake command.

Device: 0x53 0x00 0x06 0x00 0xFF // (Can be omitted)0x58 Responding to a handshake.

3.1.2 Get Version and Configure Compression Baud Rate

Host: 0x50 0x00 0x07 0x20 0x03 0x30 0xAA // Configuration Baud Rate to 115200 （0x30*2400）.

// Response does not support automatic compression of baud rates.
Device: 0x53 0x00 0x09 0x20 0x56 0x33 0x32 0x31 0x68

// Response support for automatic compression of baud rates.
Device: 0x53 0x00 0x09 0x20 0x56 0x33 0x32 0x32 0x03 0x30 0x9E

3.1.3 Loading Configuration Information

Host: 0x50 0x00 0x11 0x02 0x00 0x00 0x00 0x00 0x20 0x00 0x04 0x00 0x00 0x00 0x12 0xB4 0x4D // Loading configuration information command.

Device: 0x53 0x00 0x06 0x02 0x01 0x5C // Response configuration information.

3.1.4 Loading Configuration Data is not Last

// [xx … xx (Total 256 byte)] xx(Check) Loading configuration data command.
Host: 0x50 0x01 0x0D 0x02 0x00 0x00 0x00 0x01 0x20 0x00 0x04 0x00

// Response configuration data, Host continues to send data.
Device: 0x53 0x00 0x06 0x02 0x01 0x5C

3.1.5 Loading Configuration Data Last

// xx [xx … xx (Total 256 byte)] xx(Check) Loading configuration data command.
Host: 0x50 0x01 0x0D 0x02 0x00 0x00 0x00 0x02 xx xx xx

Device: 0x53 0x00 0x06 0x02 0x02 0x5D // Response configuration data.

3.1.6 Download Information

Host: 0x50 0x00 0x15 0x01 0x00 0x00 0x00 0x00 0x08 0x00 0x00 0x00 0x00 0x00 0x10 0x64 0x00 0x00 0x00 0x01 0xE3 // Loading program information command.

Device: 0x53 0x00 0x11 0x01 0x00 0x00 0x00 0x00 0x08 0x00 0x00 0x00 0x00 0x00

0x10 0x64 0xE1 // Response download program information

3.1.7 Download Program Data

0x01 [xx … xx (Total 256 byte)] xx(Check) // Loading program data command.
Host: 0x50 0x01 0x0D 0x01 0x00 0x00 0x00 0x11 0x00 0x00 0x00

Device: 0x53 0x00 0x11 0x01 0x00 0x00 0x00 0x11 0x00 0x00 0x00 0x01 0x90 0x06
0x00 0x20 0x2D // Response download data.

3.1.8 Chip Encryption

Host: 0x50 0x00 0x25 0x0A 0x00 0xFF 0xFF 0x00 0xFF 0x00 0xFF 0x00 0xFF 0x00
0xFF 0x00 0xFF 0x00 0xFF 0x00 0x00 0x06 0x4F 0xF0 0x00 0x07 0x4F 0xF0 0x00
0x08 0x4F 0xF0 0x00 0x0B 0x21 0xF0 0x65 // Chip Encryption command.

Device: 0x53 0x00 0x09 0x0A 0x00 0x00 0x00 0x02 0x68 // Chip encryption response.

3.1.9 Chip Initialization

Host: 0x50 0x00 0x05 0x5A 0xAF // Initialization command

Device: 0x53 0x00 0x09 0x5A 0x00 0x00 0x00 0x00 0xB6 // Responding to initialization command.

3.1.10 Baud Rate Configuration

// Configuration baud rate of 115200.
Host: 0x50 0x00 0x09 0x03 0x00 0x01 0xC2 0x00 0x1F

// (Can be omitted) 0xC0 Response configuration baud rate.

Device: 0x53 0x00 0x09 0x03 0x00 0x01 0xC2 0x00 0x22

3.1.11 Program Jump

// Jumps to 0x0800000 run program command.
Host: 0x50 0x00 0x09 0x09 0x08 0x00 0x00 0x00 0x6A

Device: 0x53 0x00 0x09 0x09 0x00 0x00 0x00 0x00 0x65 // Response to jump command.

3.2 Additive Check

```
u8 get_checksum(u8 *p, u16 len)
{
        u8 i = 0;
        u32 sum_tmp = 0;
        while (len > 0)
        {
                sum_tmp += *p++;
                len--;
        }
        sum_tmp &= 0x000000ff;
        return (u8)sum_tmp;
}
```

4 ISP Firmware Revision List

Chip Model	FW Version	Modify Content	Note
MM32F031xx_n MM32L0xx_n MM32W0xx_n MM32SPIN0x_n	V021	Basic Function： 1. Download configuration. 2. Join FW version number send. 3. Support for variable SRAM address. 4. UART1 supports PA9.10.	For example: “V021” Byte 1：version ‘V’ means version Byte 2：Chip model (Cortex-M3/M0) ‘0’ means Core is Cortex-M0 ‘3’ means Core is Cortex-M3 Byte 3：Chip version (n/o/p/q version) ‘1’ means version ‘m’ ‘2’ means version ‘n’ ‘3’ means version ‘o’ ‘4’ means version ‘p’ ‘5’ means version ‘q’ ‘6’ means version ‘s’ Byte 4：ISP firmware version
MM32F103xx_n MM32L3xx_n MM32W3xx_n MM32SPIN4x_n	V321	Basic Function： 1. Download configuration. 2. Join FW version number send. 3. Support for variable SRAM address. 4. UART1 supports PA9.10.	
MM32F103xx_o MM32L3xx_o MM32SPIN4x_o	V331	Basic Function： 1. Download configuration. 2. Join FW version number send. 3. Support for variable SRAM address. 4. Support baud rate configuration. 5. UART1 supports PA9.10 and PB6.7.	
MM32SPIN2x_p MM32F031xx_p MM32L0xx_p MM32W0xx_p	V041	Basic Function： 1. Download configuration. 2. Join FW version number send. 3. Support for variable SRAM address. 4. Support baud rate configuration. 5. UART1 supports PA9.10 and PB6.7.	
MM32F031xx_q MM32L0xx_q MM32W0xx_q MM32SPIN0x_q	V051	Basic Function： 1. Download configuration. 2. Join FW version number send. 3. Support for variable SRAM address. 4. Support baud rate configuration. 5. UART1 supports PA9.10 and PB6.7.	
MM32F032xx_s	V061	Basic Function： 1. Download configuration. 2. Join FW version number send. 3. Support for variable SRAM address. 4. Support baud rate configuration. 5. UART1 supports PA9.10 and PB6.7 and PA13.14.	

5 Document Revision History

Revision	Date	Author	Comment	Notes
1.0	2015/x/x	tm	1.Includes FLASH programming operations.	
1.1	2016/4/x	tm	1.Add version information.	
2.1	2017/4/5	tm	1.Increase read and write protection commands. 2.Increase reading, writing, and erasing commands.	
2.1a	2017/4/5	tm	Increase automatic baud rate confirmation command	
2.1b	2017/4/7	tm	Increase the verification of enabling and deactivation items.	
V2.1b	2017/4/14	sjh	Modify configuration data protocol errors.	
V2.2	2017/12/8	sjh	Check the correction confirmation.	
V2.3	2018/10/12	tyx	Add new chips	
V2.4	2019/05/14	tm	One-click download when repairing protection(only protected byte encryption)	
V2.5	2019/07/18	tm	Add new chips	
V2.6	2019/11/18	HHH	Translate to English Version Move m Version in “4 ISP Firmware Revision List”, add more description	