

Introduction

HolyChip part of the Flash MCU uses Flash as an EEPROM-like design method. The ROM space of the MCU can be used for EEPROM-like data storage (IAP operation), and the user's flexibility in data storage is greatly improved.

During IAP operation, if relevant protective measures are not taken, it may cause abnormal program operation. This manual mainly introduces the precautions for the use of IAP.

- Chips applicable to this technical manual: HC89S003F4、HC89S105C6、HC89S105C8、HC89S105K6、HC89S105K8、HC89F0411P、HC89F0421、HC89F0431、HC89F0531、HC89F0541、HC89F301、HC89F302、HC89F303、HC89F3421、HC89F3531、HC89F3541.
- Related data manuals, tools and technical documents download URL: <http://www.holychip.cn/>.

目录

1、 BOR voltage and BOR debounce time setting.....	3
1.1 Principle description.....	3
1.2 BOR voltage setting method	3
2、 Option set IAP erasing protection settings.....	4
2.1 Principle description.....	4
2.2 Operating instructions.....	4
3、 Judge the MCU voltage before IAP operation	7
4、 Interrupt.....	7
5、 Double area to save and store data	8
5.1 Implementation principle.....	8
5.2 Operating instructions.....	8
6、 IAP function absolute address compilation function absolute address compilation and pointer call.....	9
6.1 Implementation Principle.....	9
6.2 Operating instructions.....	9
7、 Version description	11

1、 BOR voltage and BOR debounce time setting

1.1 Principle description

The MCU working voltage is 2.0V-5.5V. If the BOR is turned off, the MCU may work abnormally at 1.5V-2.0V, and the program may run away and cause false erasure. If the BOR voltage is set to 2.0V and above, the MCU can be guaranteed to work in a stable voltage range.

1.2 BOR voltage setting method

BOR voltage detection control register BORC

Bit number	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R	R	R	R/W	R/W	R/W
Reset value	1	0	0	0	0	0	0	0
Bit symbol	BOREN	BOR_DBC_EN	-			BORVS[2:0]		

Bit number	Bit symbol	Description
7	BOREN	BOR Enable bit 0: Disable BOR 1: Allow BOR
6	BOR_DBC_EN	BOR debounce enable bit 0: Disable 1: Allow
5-3	-	Reserved bit (read as 0, write invalid)
2-0	BORVS[2:0]	BOR detection voltage point selection bit 000: 1.8V 001: 2.0V 010: 2.4V 011: 2.6V 100: 3.0V 101: 3.6V 110: 3.9V 111: 4.2V

BOR voltage detection control register BORC

Bit number	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0
Bit symbol	BORDBC[7:0]							

Bit number	Bit symbol	Description
7-0	BORDBC[7:0]	BOR debounce control bit

	<p>Debounce time = BORDBC[7:0] * 8TCPU + 2 TCPU</p> <p>Note: BOR_DBC_EN needs to be enabled, otherwise BOR will not debounce.</p>
--	---

When setting the BOR voltage and BOR debounce time, set the BOR voltage to at least 2.0V. If the system allows, the higher the BOR voltage setting, the better, and try to set the debounce time as short as possible to ensure that the MCU is working reliably. Voltage environment.

The power consumption after BOR is enabled is about 8uA. If users need low power consumption, it is recommended to turn off BOR before entering power-down mode, and enable BOR after wake-up from power-down mode to save power consumption.

2、Option set IAP erasing protection settings

2.1 Principle description

Set the relevant IAP erasing protection in Option and enable the data protection bit in the program area, which can effectively ensure that the program area data will not be overwritten or erased by mistake

2.2 Operating instructions

2.2.1 Keil operation method

Take the HC89S003F4 "Flash as EEPROM-like read and write" example as an example.

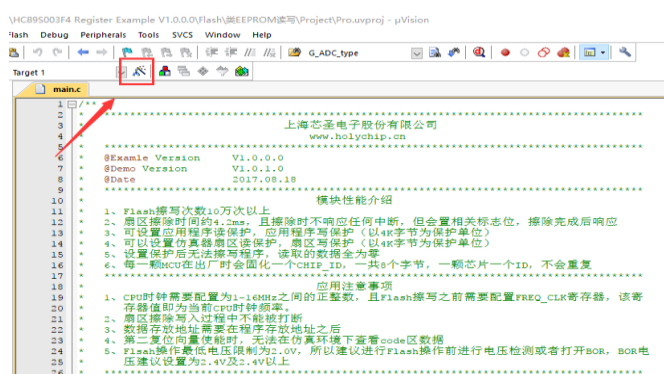


Figure 2-1

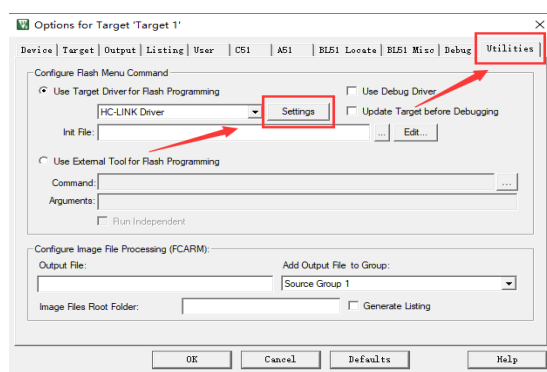


Figure 2-2



Figure 2-4

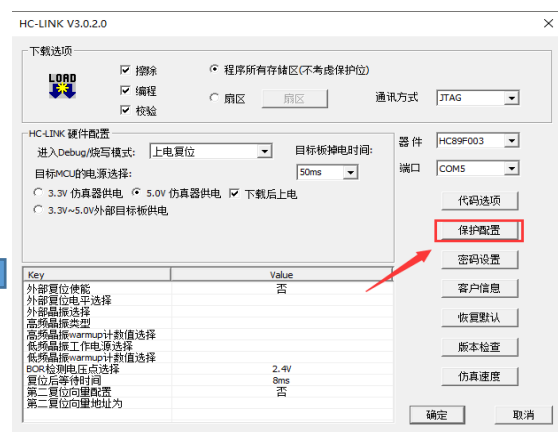


Figure 2-3

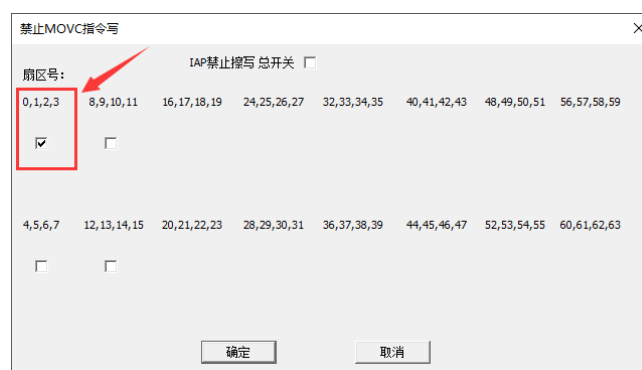
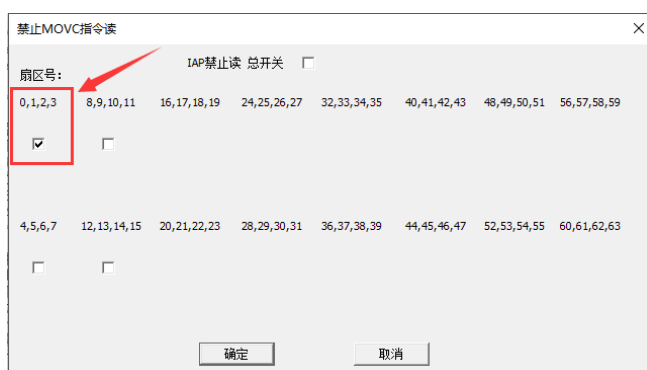


Figure 2-5

Steps:

- 1、Open Option in KEIL. (Figure 2-1)
- 2、Click the Settings option in the Utilities tab. (Figure 2-2)
- 3、Click the Protection Configuration tab. (Figure 2-3)
- 4、Open the IAP-RD and IAP-EW tabs respectively. (Figure 2-4)
- 5、Tick the corresponding sector as required and click OK. (Figure 2-5)

2.2.2 HC-PM51 operation mode

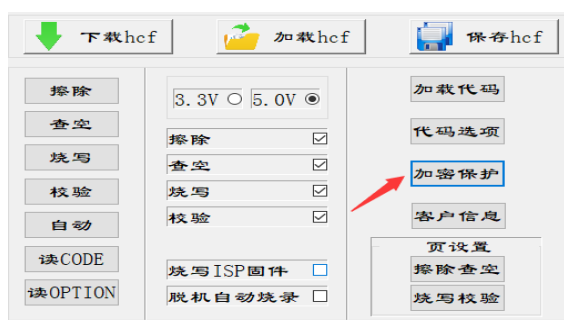


Figure 2-6

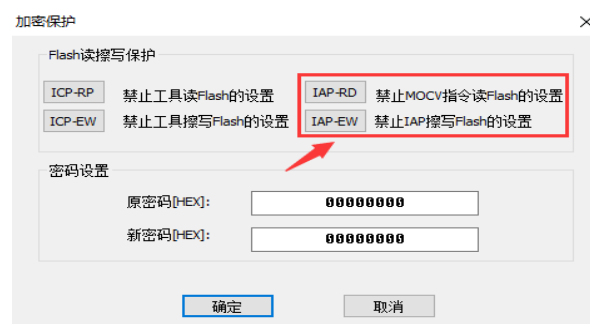


Figure 2-7

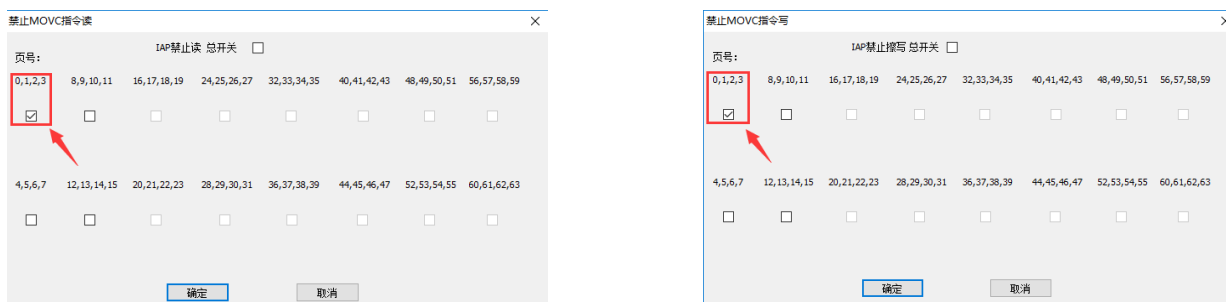


Figure 2-8

Steps:

1. Select the chip model, configure the code option, check the encryption protection option after loading the code, and click the encryption protection setting option. (Figure 2-6)
2. Open the IAP-RD and IAP-EW tabs respectively. (Figure 2-7)
3. Check the corresponding sector according to your needs and click OK. (Figure 2-8)

2.2.3、Protection area calculation method

```
Rebuild target 'Target 1'
compiling main.c...
..\user\main.c(106): warning C294: unreachable code
linking...
Program Size: data=28.0 xdata=0 code=585
creating hex file from "Pro"...
"Pro" - 0 Error(s), 1 Warning(s).
Build Time Elapsed: 00:00:03
```

Figure 2-9

HolyChip Flash series MCUs are 128 bytes and a sector, $8 \times 128 = 1\text{K}$ is 1 page.

After KEIL is compiled, the compiled byte size will be generated. Take the HC89S003F4 "Flash as EEPROM-like read and write" example as an example. The compiled bytes in this example are 585 (as shown in Figure 2-9). The protection is a unit, so check pages 0, 1, 2, and 3 when checking (Figure 2-5 and Figure 2-8). Some MCUs use 1K bytes as the protection unit, and users can choose according to their needs.

3、 Judge the MCU voltage before IAP operation

In order to prevent MCU power-on transient voltage instability causing program runaway and misoperation, it is recommended to use ADC or LVD to detect the current MCU voltage before each IAP operation. If the power supply is lower than 2.0V, no IAP operation is performed. For specific operation methods, please refer to the MCU data manuals and routines.

4、 Interrupt

Before the IAP operation, it is recommended to close the interrupt (EA=0) to ensure that it will not be affected by the interrupt during the IAP operation. After the IAP operation is completed, the interrupt is restored.

5、 Double area to save and store data

5.1 Implementation principle

When performing IAP operations, it is inevitable that after the data erasing is completed, the power will be lost before the data has been written. Therefore, it is recommended to use the dual-region data storage method. Even if the data in one region is erased, the other can be guaranteed. The data of one area is read normally.

5.2 Operating instructions

Erase and write operations

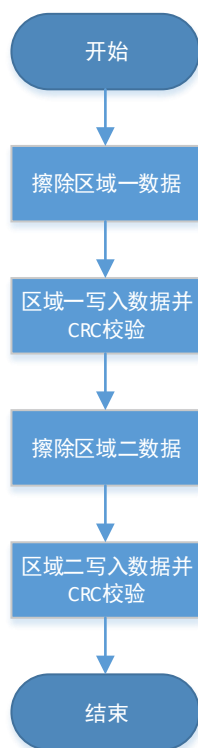


Figure 5-1

Read operations

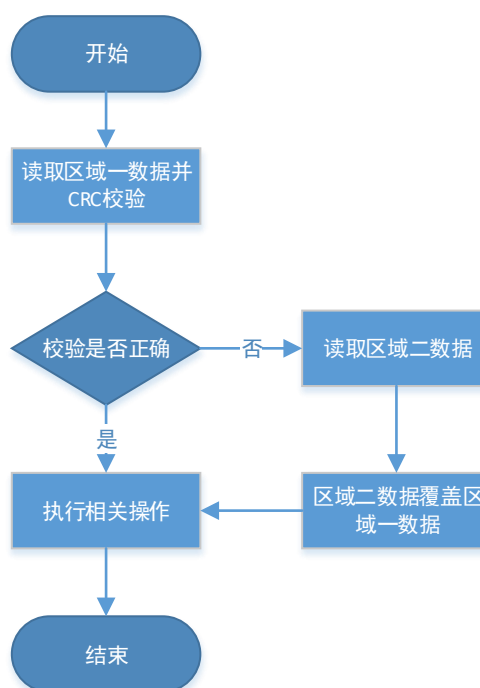


Figure 5-2

Erase and write operations:

When updating data, update the data in area one first, and then update the data in area two after the update is completed. (Figure 5-1)

Read operation:

When reading data, first read the data in area 1 and perform CRC check. If the check is wrong, discard the data in area 1 and read the data in area 2. (Figure 5-2)

6、 IAP function absolute address compilation function absolute address compilation and pointer call

6.1 Implementation Principle

Set the related IAP erasing protection in Option, and after enabling the sector protection bit where the user program is located, use absolute address compilation to put the IAP operation function in the area where IAP protection is not set, and use the function pointer to call the function to ensure that the MCU The user code will not be erased by mistake.

6.2 Operating instructions

Take HC89S003F4's "Flash as EEPROM-like read and write" example as an example,

1、 Open Flash for EEPROM-like read and write, find FLASH erase (Flash_EraseBlock) and write in the main function.(FLASH_WriteData) function, open the Project -> Options for Target menu, select the BL51 Locate tab, and enter in Code:: ?PR?_Flash_EraseBlock?MAIN(0X2000),?PR?_FLASH_WriteData?MAIN(0X2080). The format is: ?PR?_function name?MAIN (address to be compiled), note that the input text (including punctuation) must be in English format, otherwise the compiler will report an error, and then click OK. After the program is compiled, in the M51 file Relevant information can be found in, as shown in Figure 6-3
The program shown has been compiled at the relevant address.

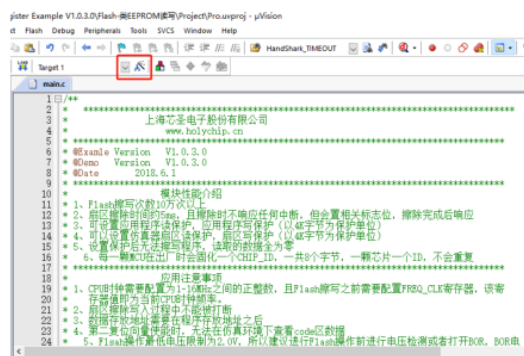


Figure 6-1

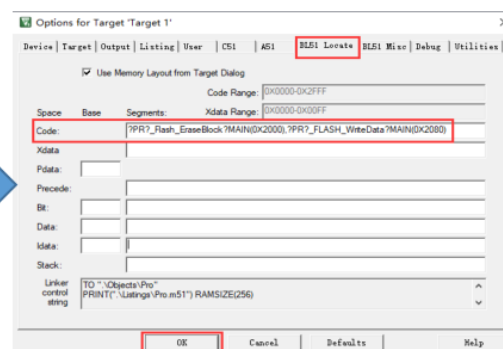


Figure 6-2

```

***** C O D E   M E M O R Y *****
CODE 0000H 0003H ABSOLUTE
CODE 0003H 0019H UNIT ?C_INITSEG
CODE 001CH 0007H *** GAP ***
CODE 0023H 0003H ABSOLUTE
CODE 0026H 00BAH UNIT ?PR?MAIN?MAIN
CODE 00E0H 008CH UNIT ?C_C51STARTUP
CODE 016CH 0044H UNIT ?PR? FLASH_WRITEARR?MAIN
CODE 01B0H 0037H UNIT ?PR?UART1_RPT?MAIN
CODE 01E7H 0037H UNIT ?PR? FLASH_READARR?MAIN
CODE 021EH 0031H UNIT ?C?LIB_CODE
CODE 024FH 1DB1H *** GAP ***
CODE 2000H 0022H UNIT ?PR? FLASH_ERASEBLOCK?MAIN
CODE 2022H 005EH *** GAP ***
CODE 2080H 0024H UNIT ?PR? FLASH_WRITEDATA?MAIN

```



Figure 6-3

2. Use the function pointer to call the function at the absolute address, please refer to the routine for details

- (1) Define the function prototype: `void Flash_EraseBlock(unsigned int fui_Address);`
- (2) Define the corresponding function pointer variable: `void (*CALL_FLASH_ERASE)(unsigned int i);`
- (3) Function pointer variable assignment, the function that points to the absolute address we locate:
`CALL_FLASH_ERASE=Flash_EraseBlock;`
- (4) Call of function pointer: `(*CALL_FLASH_ERASE)(0x2B00);`

7、 Version description

Version	Date	Description
V1.00	2018/1/25	First edition
V1.01	2018/6/26	1. Add related models 2. Modify some errors

Holychip reserves the right to make change without further notice to any products herein to improve reliability, function or design. Holychip does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. Holychip products are not designed, intended, or authorized for use as components in system intended, for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Holychip product could create a situation where personal injury or death may occur. Should Buyer purchase or use Holychip products for any such unintended or unauthorized application. Buyer shall indemnify and hold Holychip and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, cost, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use even if such claim alleges that Holychip was negligent regarding the design or manufacture of the part.

Holychip

June 2018