

## In-Circuit Programmer of STM8 based on MCU

Li Zhao, Wenbin Xie, Guichu Wu, Xiaoqin Zhang\*  
Wenzhou University, Zhejiang, China

**Abstract**—In industrial production, it often encounters problems in updating program of MCU (Micro Controller Unit), and the process parameters can not be updated as needed in real time. In order to facilitate the online updates of STM8 microcontroller, the paper designed a STM8 online downloader based on STM32 microcontroller and SWIM protocol. The downloader does not rely on a computer or a dedicated download device. It has a simple circuit structure, and is easy to download online, accurately and fastly. When STM32 connects to a touch screen, the downloader can modify parameters online. Through practical application, the downloader has achieved a good result.

**Keywords**- SWIM; STM8; Downloader

### I. OVERVIEW

STM8 microcontroller has an unique manufacturing process, with advanced 8-bit CPU core for cost-effective and energy-saving benefits, excellent anti-jamming performance, superior reliability, the most advanced peripheral interfaces. All the advantages make it use widely in automotive electronics, industrial application, consumer electronics, medical equipment and other fields.

In industrial production, we often need to update programs, or to adjust the parameters according to demands of the factory. The programmers often need to take computers and programming tools to maintain the device individually; or recover the device to update. This will not only spend a lot of time but affect the production<sup>[1]</sup>. STM8 microcontroller has the single wire interface module (SWIM for short) and the debug module (DM for short), which do well preparation for the user to update program of STM8. It only needs a single data line, the user can conveniently program STM8 microcontroller online and debug it non-intrusive.

In this paper, we designed an online downloader of STM8 microcontroller, which uses the SWIM and DM module, and the PWM mode of MCU. The downloader does not rely on a computer and dedicated downloading devices.

### II. SINGLE WIRE INTERFACE MODULE(SWIM)

Online programming is used to update the FLASH memory and EEPROM memory contents. In-circuit debugging mode or in-circuit programming mode is managed through a single wire hardware interface featuring ultra-fast memory programming. Coupled with an in-circuit debugging module, it also offers a non-intrusive emulation mode, making the in-circuit debugger extremely powerful, and close in performance to a full-featured emulator.

\*Corresponding author.

E-mail address: xqzhang@wzu.edu.cn.

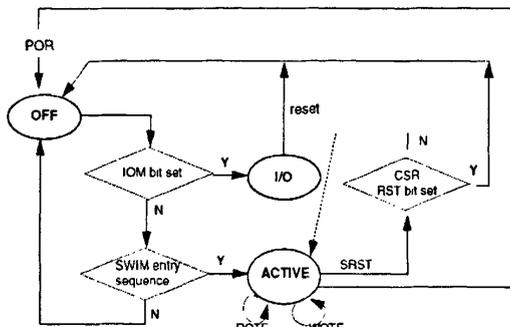


Fig. 1. Operation mode

The SWIM is a single wire interface based on asynchronous, high sink (8 mA), open-drain, bidirectional communication. While the CPU is running, the SWIM allows non-intrusive read/write accesses to be performed on-the-fly to the RAM and peripheral registers for debug purposes. In addition, while the CPU is stalled, the SWIM allows read/write accesses to be performed to any other part of the MCU's memory space (Data EEPROM and program memory). CPU registers (A, X, Y, CC, SP) can also be accessed. These registers are mapped in memory and can be accessed in the same way as other memory addresses<sup>[2]</sup>.

#### A. Operation Mode

The SWIM has three Operation modes: OFF, I/O, ACTIVE. Three modes can be switched each other, which is shown in figure 1. After a Power-On Reset (powering of the device), the SWIM is reseted, and it enters into its OFF mode.

**OFF:** In this mode the SWIM pin must not be used as an I/O by the application. It is waiting for the SWIM entry sequence or to be switched to I/O mode by the application software.

**I/O:** This state is entered by the software application by setting the IOM bit in the core configuration register (MCR). In this state, the user application can use the SWIM pin as a standard I/O pin, the only drawback is that there is no way to debug the functionality of this pin with the built-in debug capabilities. In case of a reset, the SWIM goes back to OFF mode.

**ACTIVE:** This mode is entered when a specific sequence is detected on the SWIM pin while in OFF state. In this state, the SWIM pin is used by the host to control the STM8 with 3 commands (SRST System Reset, ROTF Read On The Fly, WOTF Write On The Fly).

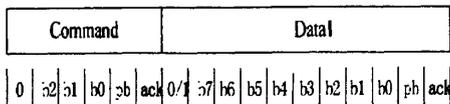


Fig 2 command and data format

TABLE I. ROTF/WOTF STRUCTURE

ROTF/WOTF	N	@E	@H	@L	D[@]	D[@+N]
-----------	---	----	----	----	------	--------

N: The 8 bits are the number of bytes to read or write (from 1 to 255)

@E/H/L: This is the 24-bit address to be accessed.

D[...]: These are the data bytes to read or write in the memory space

### B. SWIM communication Protocol

In order to control the STM8, SWIM must be in ACTIVE mode. While in ACTIVE, communication can be initiated by host( the downloader) or device(STM8). The host can start the transfer only if there is no transfer on-going. The command and data format between host and STM8 are shown in figure 2.

- command format

**Header:** 1 bit at '0'

**b2-b0:** 3-bit command

**pb:** parity bit, XOR between all b(i)

**ack:** acknowledge (1 bit at '1'). The receiver must send the not-acknowledge value if it has detected a parity error (NACK: not acknowledge = 1 bit at '0'), or it is not yet ready.

- data format

**Header:** 1 bit at '0'(Host to STM8) or 1 bit at '1'(STM8 to Host)

**b7-b0:** 8-bit data

**pb:** parity bit sent after data, XOR between all b(i)

**ack:** acknowledge

In ACTIVE mode, there are three commands: SRST, ROTF, WOTF, that are used by host to program STM8. The Device(STM8) must send acknowledge and data back to host. The whole structure of these commands is shown as follow:

**SRST:** system reset. Binary Code 0b000. This command is from host to target, and has no parameter. SRST command generates a system reset only if SWIM\_CSR/SWIM\_DM bit is set

**ROTF:** read on the fly. Binary Code 0b001. The structure is shown in table 1.

**WOTF:** write on the fly. Binary Code 0b010. The structure is also shown in table 1.

### III. SYSTEM HARDWARE DESIGN

According to the theory of SWIM module and the characteristics of communication protocol, we can conclude that it can control and program the FLASH of STM8 microcontroller, by generating required PWM pulse of command and data. In order to generate PWM pulse, we can use analog electronics and MCU. In this paper, we select the MCU to design the downloader of STM8 microcontroller. There are some advantages: 1) simple hardware, the frequency

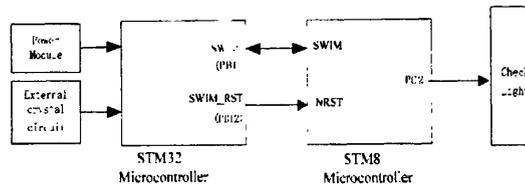


Fig 3 system hardware diagram

can be changed easily; 2) easy to update the program, computer and special program downloader are not needed; 3) it can be changed online, when connected to a touch screen in the industry.

We adopted the STM32 microcontroller to design the downloader of STM8 microcontroller. The STM32 microcontroller has excellent real-time performance, power control, ease of development and large FLASH space, which is very easy to update the whole FLASH of STM8. We can only use two I/O Ports to generate SWIM signal and SWIM\_RST( reset) signal to program the FLASH of STM8.

The whole hardware design of the system is shown in figure 3 in the below. In order to make the STM32 and STM8 work, there should be power module, which supply 3.3V or 5V voltage, and external crystal circuit, which can provide more accurate system master clock<sup>[3]</sup>.

In figure 3, the STM32 microcontroller is the host which is used to control the STM8. To program the STM8, it needs two ports: SWIM and SWIM\_RST. The SWIM is the data signal, which is asynchronous, high sink(8mA), open-drain, bidirectional communication interface. The SWIM\_RST is the reset signal of STM8, it is unidirectional. The Check Light is used to show the result of data check. When it is lighted, it shows that the data of downloading is correct, otherwise there is something wrong in the downloading data.

### IV. SOFTWARE DESIGN

There are two methods for writing (or erasing) Flash program memory<sup>[4]</sup>:

- Byte/word programming (1 or 4 bytes)

– Advantages: offers small area programming, programming is done from Flash

– Disadvantages: program stops during programming, programming speed is slow

- Block programming (128 bytes or one Flash block for a given STM8 type)

– Advantages: offers large area programming with high speed (large blocks)

– Disadvantages: programming routine must run from RAM (need to copy programming routine into RAM)

In this paper, we select the block programming mode to write. Block programming mode can be performed only by code located in RAM. First we must copy the programming code into RAM and then run this code. The RAM code then uses block mode to program the Flash. In this mode, programming one block takes the same time as programming one byte/word in byte/word mode. As a result programming speed is very fast and code execution is not stopped (because it is running from RAM). The only disadvantage of this

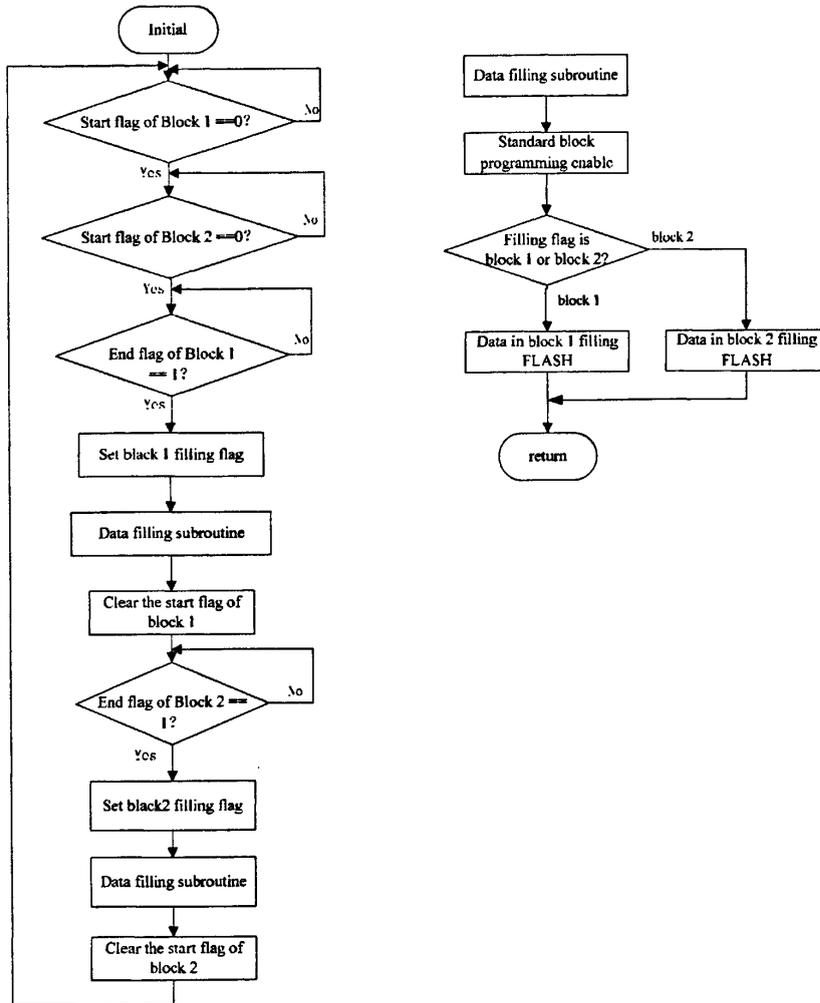


Fig 4 data mapping

method is the RAM code management<sup>[5]</sup>: 1) Copying the executable code to RAM, 2) Storing the RAM code, 3) Allocating RAM space for code, 4) Compiling the code to be able to run from RAM.

We first write executable code and data into RAM space by SWIM. The RAM space is smaller than the FLASH space, we designed the data mapping method. The mapping detail is: 1) the data downloaded by SWIM is stored in a special space of RAM; 2) Mapping the data in the RAM into FLASH corresponding space, 3) Filling the entire FLASH by many times of mapping. Figure 4 shows the mapping method, data block of 128 bytes as an example.

In order to ensure the accuracy of data downloading, we designed a simple data checking method. Firstly before downloading, all the data is added in the XOR sum, the result is filled in the last address of FLASH space. After downloading, the STM8 CPU checks all the data in FLASH

by XOR sum. If the XOR sum result is zero, then all the data downloaded is correct, otherwise there is wrong, all the data should be downloaded once again.

The data mapping, filling method and check method should be translated into machine code<sup>[6]</sup>, and the machine code is downloaded into the RAM of STM8 by SWIM. The whole process of downloading is: STM32 writes the data and programs into the RAM of STM8 through the WOTF command; the STM8 executes the program in the ram, and fill the data into FLASH by block programming; after downloading is completed, STM8 continues to check the data and output the result.

Through SWIM and DM programming the FLASH, it should first set the related registers of FLASH using the SWIM protocol, and ensure STM8 microcontroller is not running other tasks<sup>[7]</sup>. The steps of programming the FLASH are shown in figure 5.

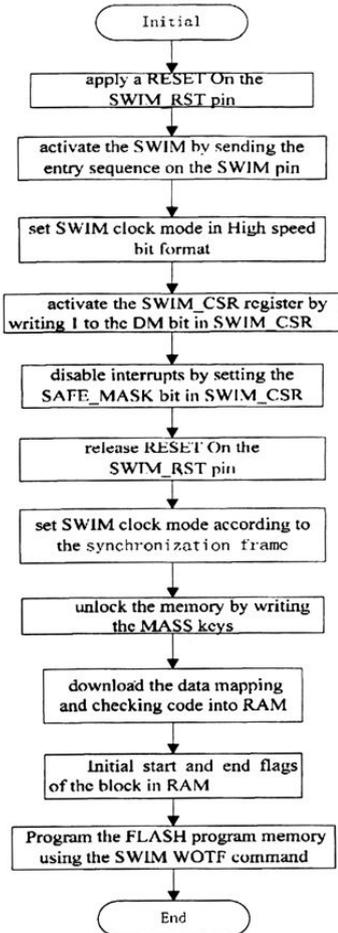


Fig 5 programming steps

## V. APPLICATION DEBUGGING

After the software and hardware are ready, we begin to test and apply. In the Application debugging, we select the STM8S103F3 as the target device, in which the RAM is 1K, and the FLASH is 8K. We select the STM32F101RB as the host, in which the RAM is 20K, the FLASH is 128K.

The STM32 first saves the pre-define data (binary data) in own FLASH, then it runs the program to download the data into the STM8 through SWIM commands. In the debugging, we connect the Logical Analytic device to capture the data pulse, then we examine the correctness of data.

In figure 6, the upper pulse is SWIM signal; the lower pulse is SWIM\_RST signal. figure 6 a) and b) are the overall diagram by captured. In figure 6 c), STM32 generated the correct entry sequence, and the STM8 identified it and sent the synchronization frame to the host followed. Figure 6 d) shows that when STM32 sent the ROTF or WOTF command, the STM8 sent back the 'ACK' and data. At the same time

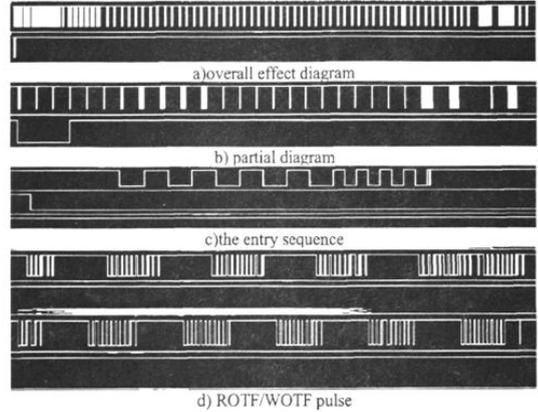


Fig 6 downloading pulse

the check LED is lighted, which shows that the data downloaded is correct. In addition, we verified the data through ST-Link (a dedicated downloader). The read data are the same with the data that needed to download. So the application debugging shows that the design in this paper is possible and effective.

## VI. APPLICATION DEBUGGING CONCLUSION

In the paper, we designed an online downloader of STM8, through the STM32 microcontroller and SWIM protocol. The downloader has a simple circuit, and only needs one data download cable. It can write any memory space of STM8. The download speed is fast, by using block programming which reduced the speed greatly; it is easy to update online, when STM32 microcontroller connect to the touch screen, you can modify the parameters at any time online. Through practical application in Tengen Company, the online downloader device achieved a good result.

## ACKNOWLEDGMENT

This work is supported by NSFC (Grant No. 61074123) and Zhejiang Provincial Top Key Discipline of Information Processing and Automation Technology.

## REFERENCES

- [1] Sun Lu, Jiang Jiu-chun, Yin Hang. Design of Procedure Downloads System Based on MC9S12 series MCU[J], Micro Computer and Information, 2010, 4-2:64
- [2] STM8 SWIM communication protocol and debug module Technical Data[Z]. 2009
- [3] Zhe Tian. Development and Application Tutorial of Embedded Systems[M]. Beijing University of Aeronautics & Astronautics Press, 2005
- [4] STM8S microcontroller family Technical Data[Z]. 2009
- [5] STM8S in-application programming (IAP) using a customized bootloader Technical Data[Z]. 2009
- [6] STM8 CPU programming manual Technical Data[Z]. 2009
- [7] How to program STM8S FLASH program memory and data EEPROM Technical Data[Z]. 2009

# In-Circuit Programmer of STM8 based on MCU

作者: [Li Zhao](#), [Wenbin Xie](#), [Guichu Wu](#), [Xiaoqin Zhang](#)

作者单位: [Wenzhou University, Zhejiang, China](#)

本文链接: [http://d.g.wanfangdata.com.cn/Conference\\_WFHYXW485981.aspx](http://d.g.wanfangdata.com.cn/Conference_WFHYXW485981.aspx)