CMOSTEK AN211

CMT2380F16 Writer Operation Guide

Overview

The CMT2380F16 Writer is developed to improve production programming efficiency. This document provides basic operational guidance CMT2380F16 Writer users.

The CMT2380F16 supports the following operations.

- Chip online programming operation
- Chip offline programming operation
- Multi-machine serial programming operation.

The product models covered in this document are shown in the table below.

Table 1. Product Models Covered in This Document

Product Model	Frequency Range	Modulation Method	Tx Power	Sensitivity	Configuration Method	Package
CMT2380F16	127 – 1020MHz	OOK/(G)FSK	+ 20dBm	-120dBm	Wireless MCU	QFN48

Table of Contents

1 CMT2380F16 Writer Introduction

1.1 Hardware Introduction

The CMT2380F16 Writer is a programmer designed for the CMT2380F16 chip applying USB interface for programming. It can be programmed online using USB bus power, or programmed offline using an external power supply. Moreover , multiple units can be connected together for one-click batch programming (or for batch programming in factory). The Writer hardware structure is both simple and compact. The user-friendly software UI interface makes this programmer very easy to use.

The Writer's PCBA appearance is shown in the below figure.

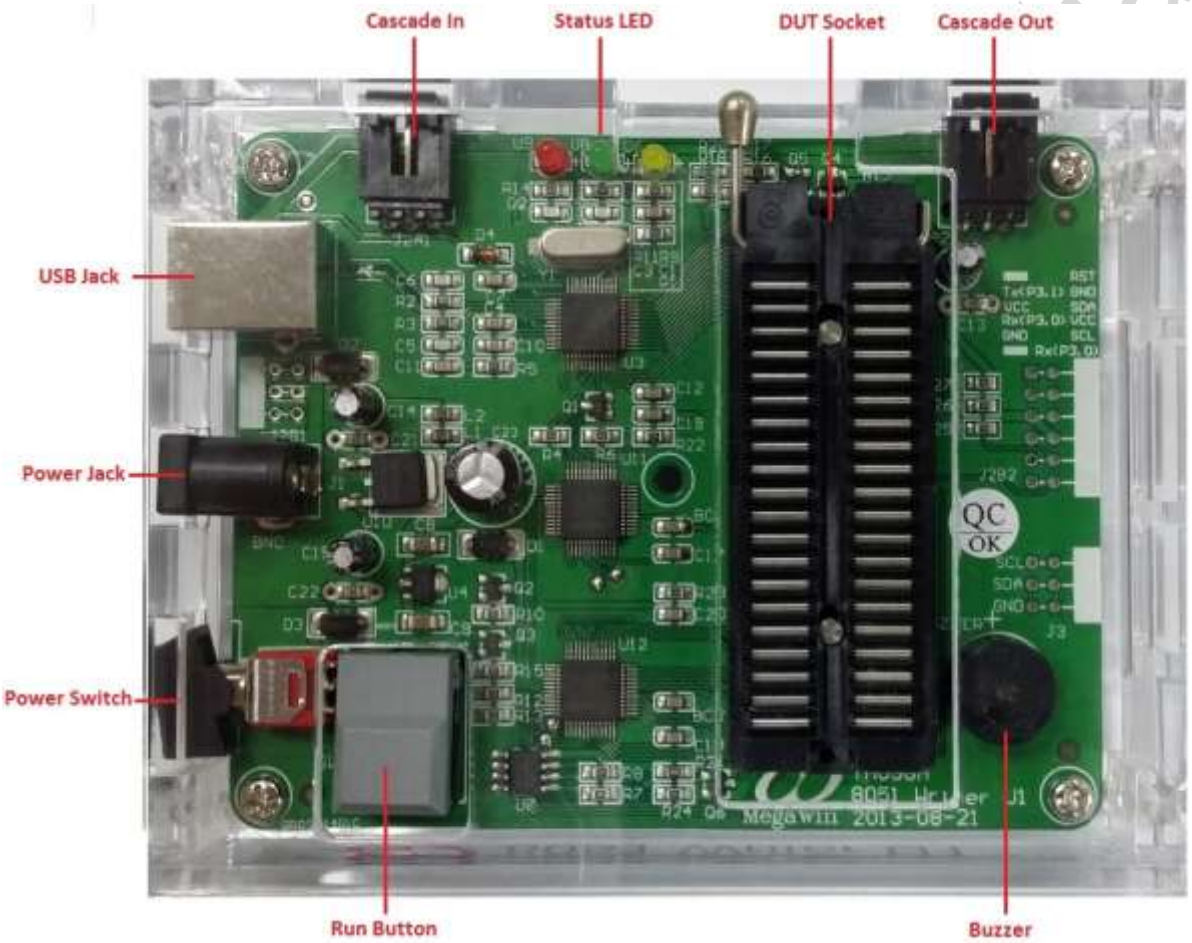


Figure 1. Writer PCBA Appearance

1.2 Writer PC Software

The CMT2380F16 Writer software is a portable one with free installation with no need for driver.

2 CMT2380F16 Writer Operation

The CMT2380F16 Writer UI interface is shown in the below figure.

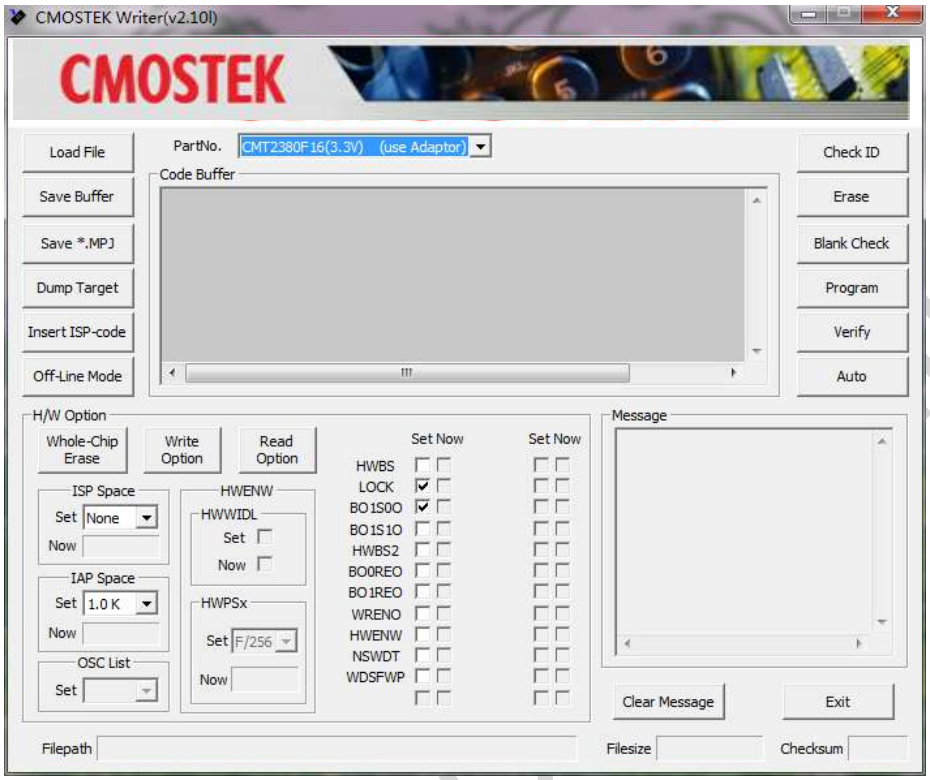
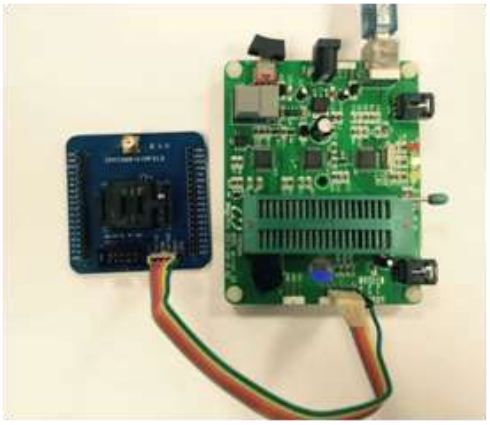


Figure 2. Writer UI Interface

2.1 Online Programming Operation

The online programming connection is shown in the below figure.



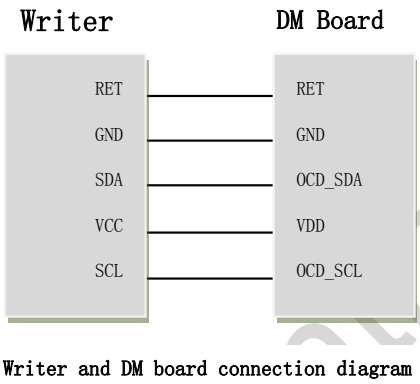


Figure 3. Online Programming Connection

The online programming operation flow is described in the below sections

2.1.1 Chip Model Selection

Select chip model CMT2380F16(3.3V) is selected automatically as shown in the below figure.

Figure 4. Select Chip Model

2.1.2 Load File

- Select AP/IAP Area

Click *Load File* button, the *Select AP/IAP Area* window pops up as shown in the below figure. *AP (Code)* refers to the user program. *IAP (Data)* refers to the data programmed to the chip along with the user program, such as configuration data, company name, product name, etc. IAP is an optional configuration item.

Notes:

1. For details on the IAP space, see details in Section 5.

Figure 5. Select AP (Code)

When selecting *IAP (Code),* users can set the storage area. An example of IAP(Code) storage area is shown in the below figure.

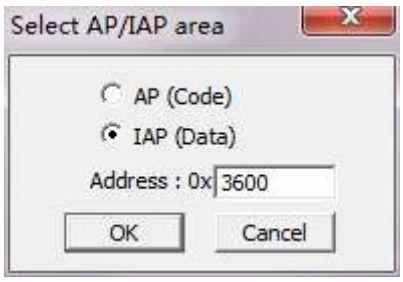


Figure 6. Select IAP(Data)

- Load file

After above AP/IAP Area settings, the file loading window pops up as shown in the below figure, to load the IC code files to be programmed to the chip (file formats are **.BIN**, **.HEX** and **.MPJ**).

1) Select the file to be programmed.
2) Click *Open* to load the files to be programmed. If a **.MPJ** file is loaded, users can click *Auto* button to perform Auto operation. See Section 2.1.5 for details.

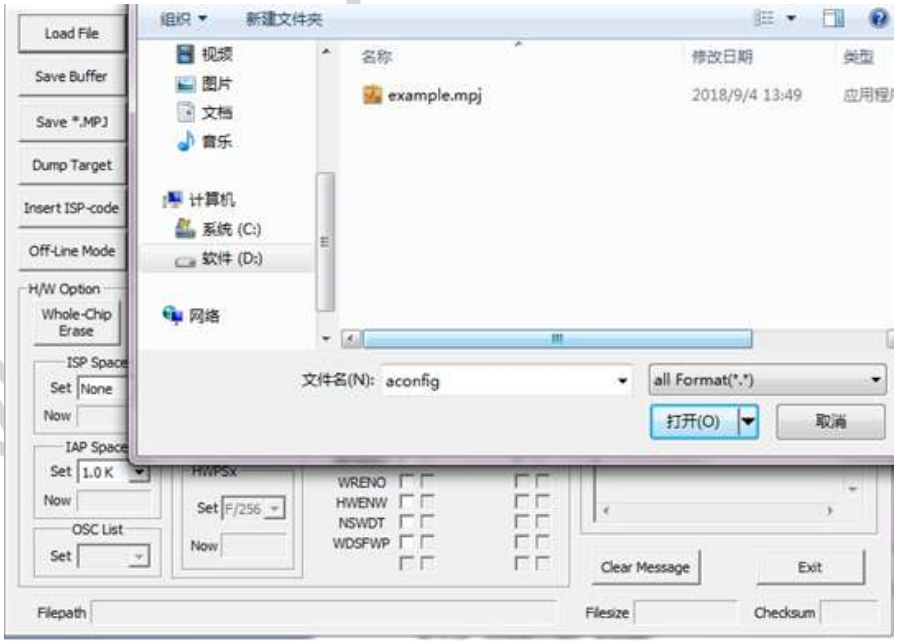


Figure 7. Load File

2.1.3 Insert ISP Code

If users do not need to program ISP code, please sip this step.

- Insert the CMOSTEK-provided ISP code

1) Click *Insert ISP Code*.

2) Select *CMOSTEK-provided ISP code* then click *Insert* button.

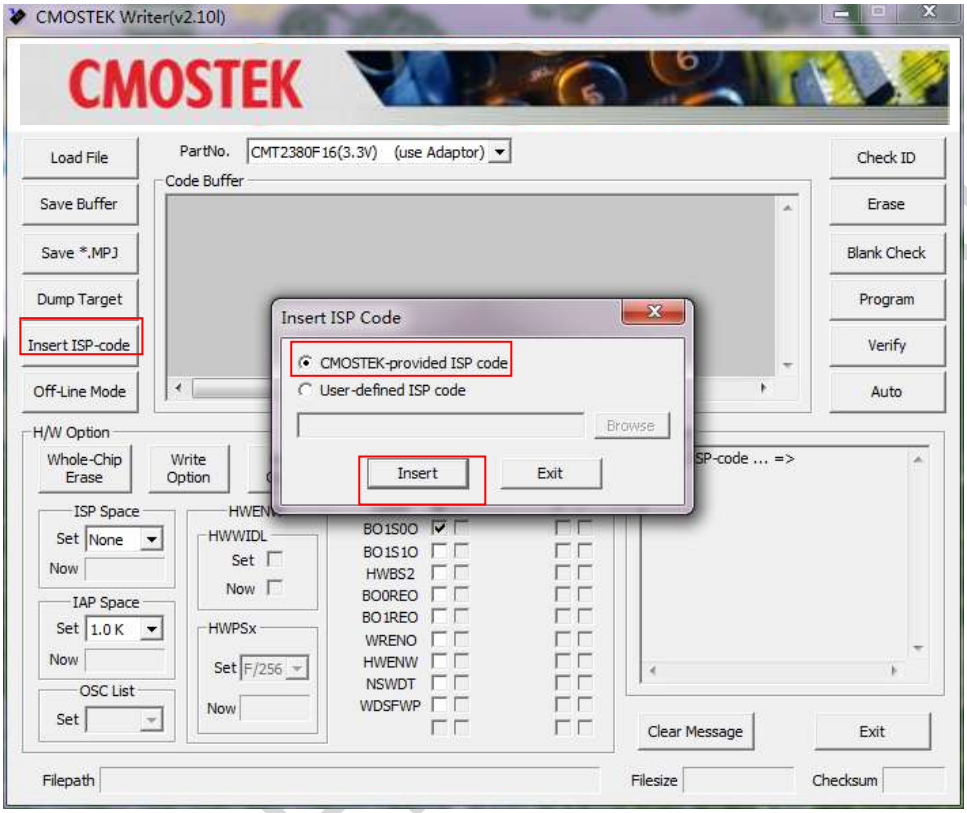


Figure 8. Insert ISP Code

- Insert *User-defined ISP code*

  1) Click on *Insert ISP Code*.

  2) Select *User-defined ISP code* then click *Browse.*

  3) Select the ISP code to be inserted then click *Open*.

  4) Click *Insert* button.

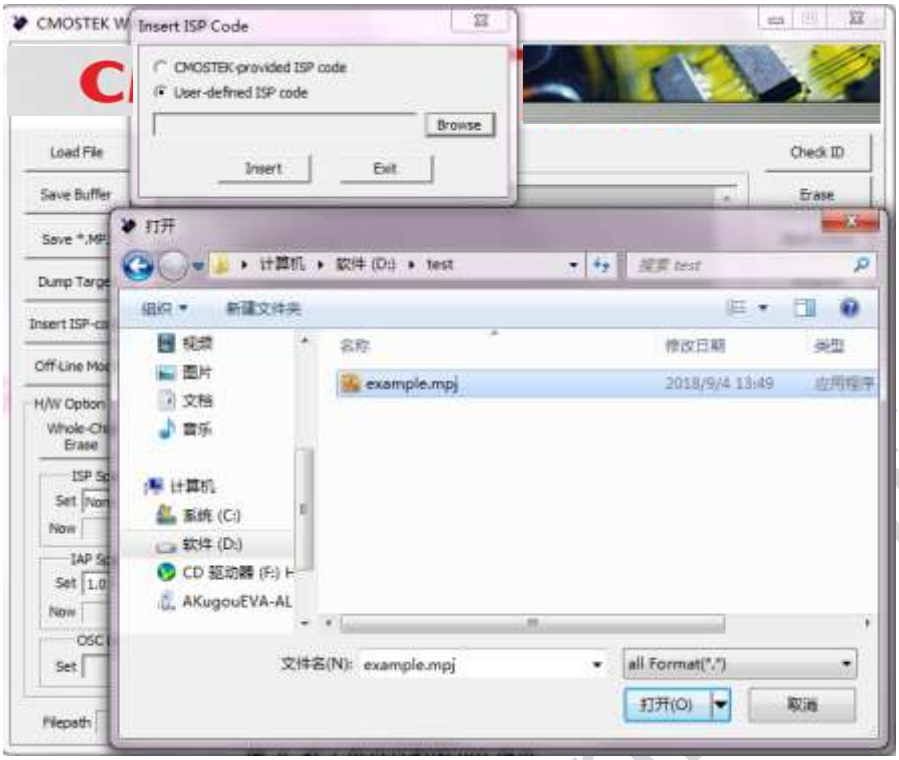


Figure 9. Insert User-defined ISP Code

Notes:

1. The function of ISP code is similar to the STM32 bootloader.
2. The chip supports U1 and ICE programming.
3. The ISP code will occupy some Flash space.
4. After the ISP code is inserted, the software will set *ISP space* autompatically with the *HWBS* flag ticked on.

2.1.4 Set Hardware Options

1) ISP space setting: If the ISP code function is needed, users need to set the corresponding space size for storing ISP code.

2) IAP space setting.: Set the space size of IAP (EEPROM)

3) Watchdog settings: it is valid only of *HWENW* in the IC option is ticked on.

4) IC option settings.

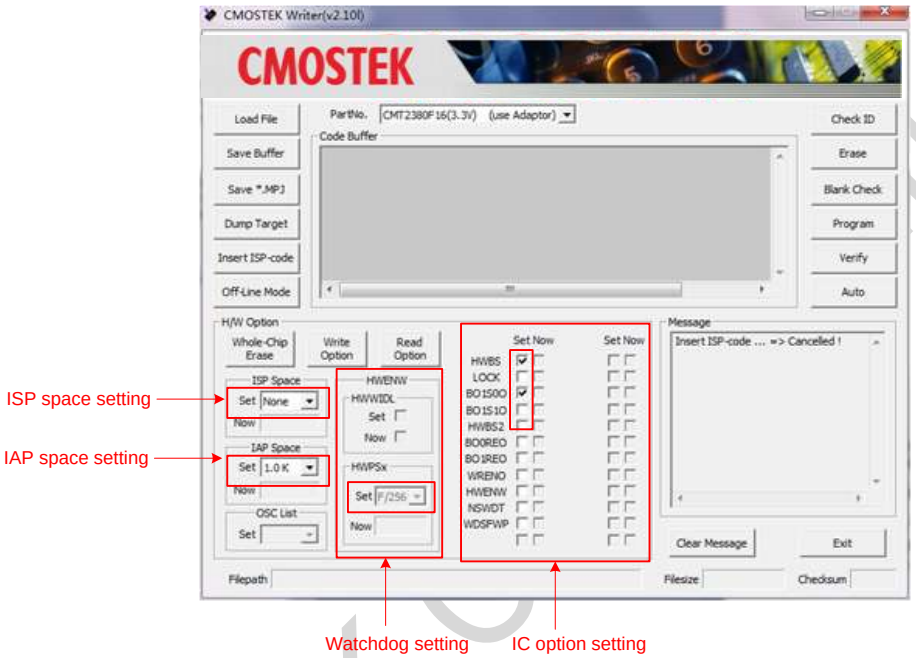


Figure 10. Hardware Options Setting

Notes:

1. In general, users only need to tick on *LOCK* (encryption), *BO1S0O* (voltage setting for low voltage detection) and *HWBS* (if *insert ISP code* operation is performed, this item should be ticked on. Otherwise, no need to tick on it).

2.1.5 Perform *Auto* Operation

Click *Auto* button, then select the desired operations, and click *Run* after then. When one IC programming is done, clock Run again for programming another IC. In general, users can ticked on the the options as shown in the below figure.

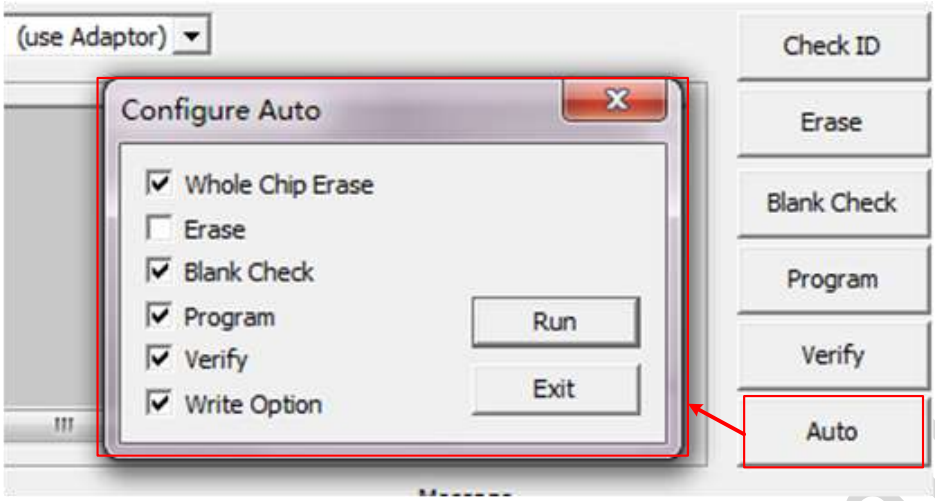


Figure 11. Perform Auto Operation

2.1.6 Save Project File

This operation is to save the above settings into a project file as shown in the below figure to facilitate the further programming or mass production programming.

- Click *Save Project*.MPJ*.
- The system packs the current *chip model, AP (program), IAP (data), ISP space, IAP space, setting* into an .mpj file.
- Open the .mpj later by performing Open File -> AP and the software will automatically load the above content.

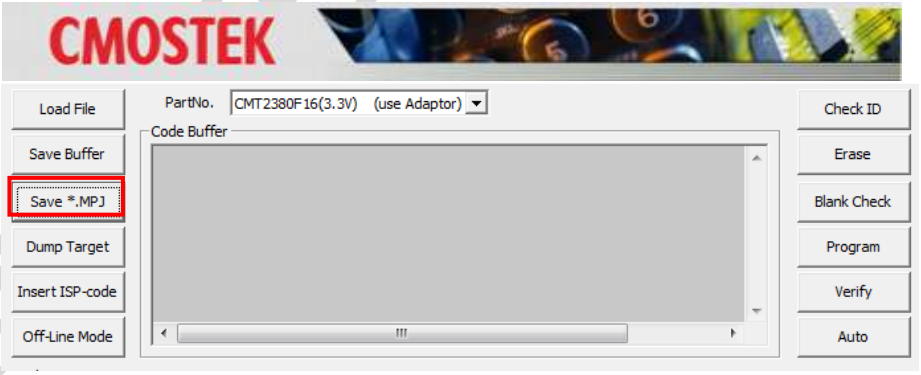


Figure 12. Save Project File

After clicking *Save*, the project file information window will pop up, where it can be printed or saved as a picture. When opening this project file for programming later, users can checking the printed file information or the saved image to confirm whether the project file is correct.

Figure 13. Project File Information

2.1.7 Read Target Chip Information

As shown in the below figure, if a chip is not encrypted, users can read Flash, ISP space, IAP space and Setting information by clicking *Dump Target* button.

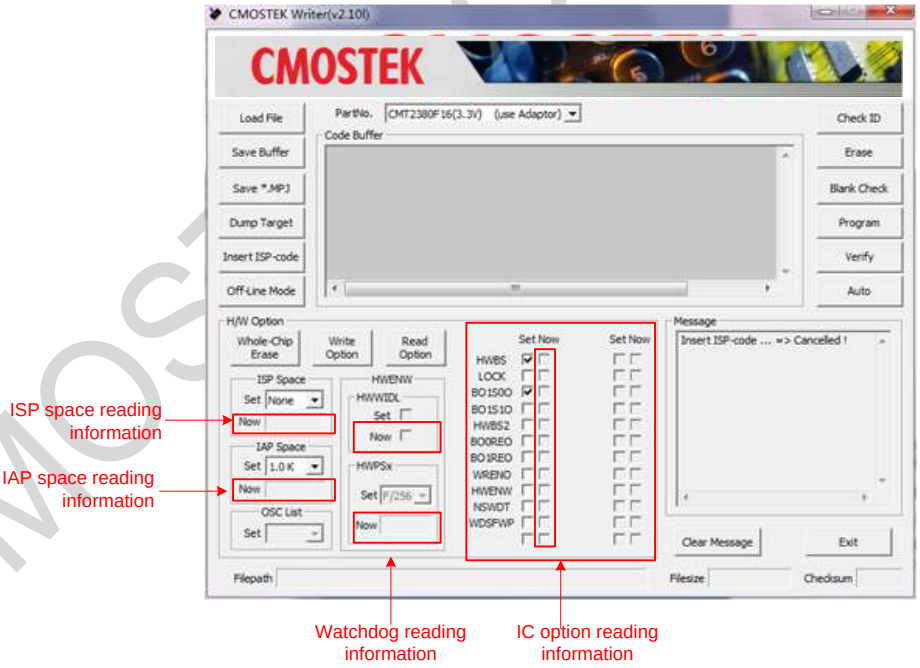
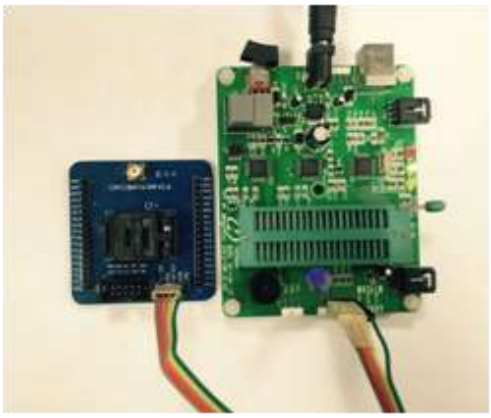
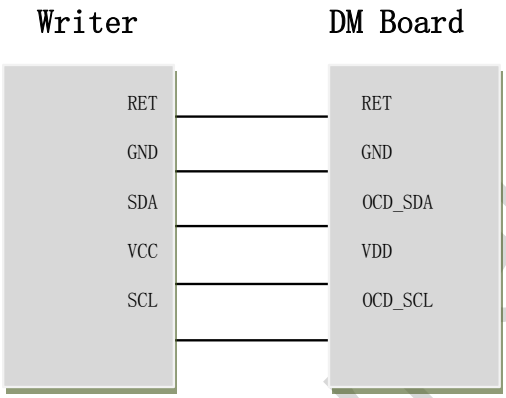


Figure 14. Read Target Chip Information

2.2 Offline Programming Operation

The offline programming connection is shown in the below figure.





Offline programming connection **Writer and DM board connection**

Figure 15. Offline Programming Connection

The operation flow of offline programming is described in the following sections.

2.2.1 Chip Model Selection

The specific steps are the same as in section 2.1.1.

2.2.2 Load File

The specific steps are the same as in section 2.1.2.

2.2.3 Insert ISP Code

The specific steps are the same as in section 2.1.3.

2.2.4 Set Hardware Options

The specific steps are the same as in section 2.1.4.

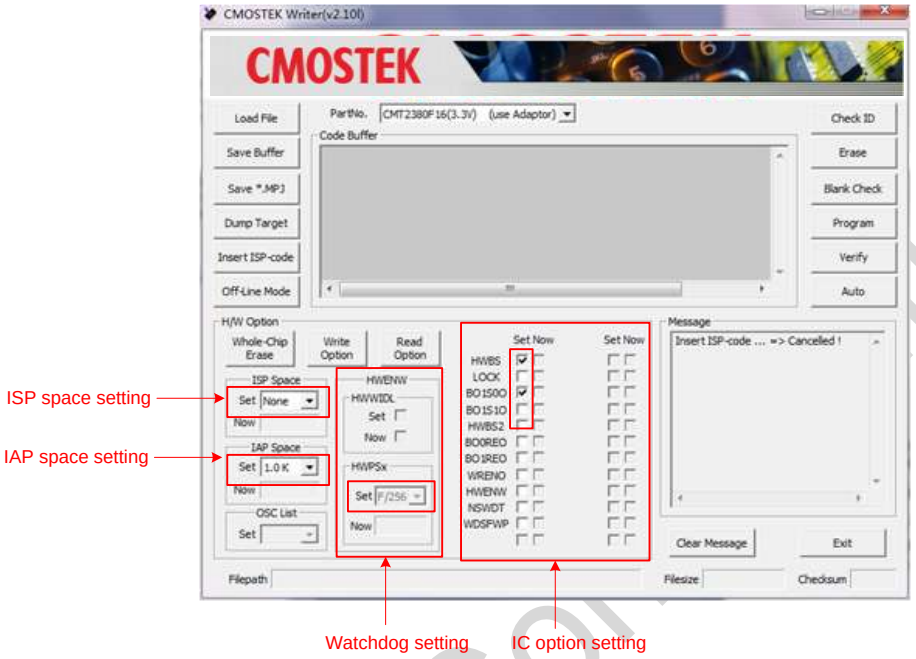


Figure 16. Set Hardware Options

2.2.5 Download Code and Hardware Settings to Programmer

1) Click *Offline mode.*

2) Set the maximum number of programming 0~65534). x. For example, if the *max counter* is set to 2000, the U1 tool will report an error during programming after successfully programming 2000 chips. The value of 0 represents unrestricted programming times. The maximum value of max counter is 65535.

3) Set the serial number (see Section 3 How to use the serial number for more details).

4) Click *Download* to download the code and hardware settings to the programmer

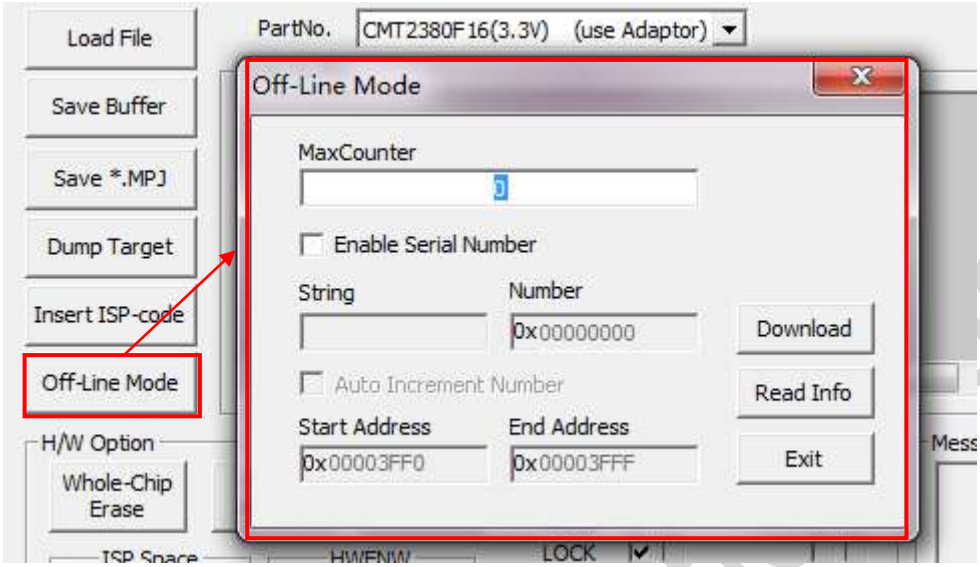


Figure 17. Offline Programming

2.2.6 Press Auto Key to Start Programming

The yellow light is on during programming. After programming completion, if successful, the green turns on and the buzzer will have one long beep, otherwise the red light turns on and the buzzer will have 10 short beeps.

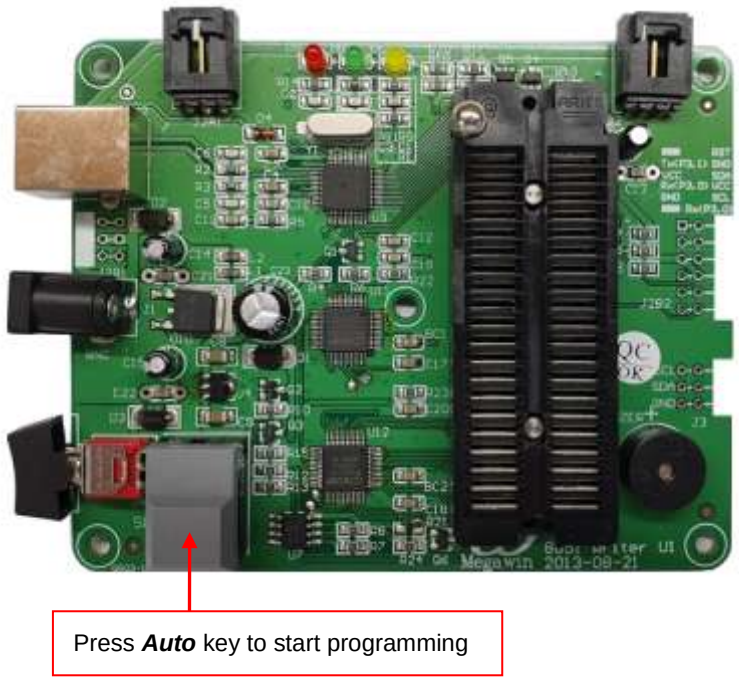


Figure 18. Press Auto Key to Start Programming

2.3 Offline Copying Programming Operation

The offline copying programming connection is shown in the below figure.

The offline copying programming operation is as follows.

- Download the code and hardware settings to the programmer following the offline programming operation.
- Connect the programmers as shown in the below figure
- Press Auto key to start programming.

  The yellow light is on during programming. After programming completion, if successful, the green turns on and the buzzer will have one long beep, otherwise the red light turns on and the buzzer will have 10 short beeps.

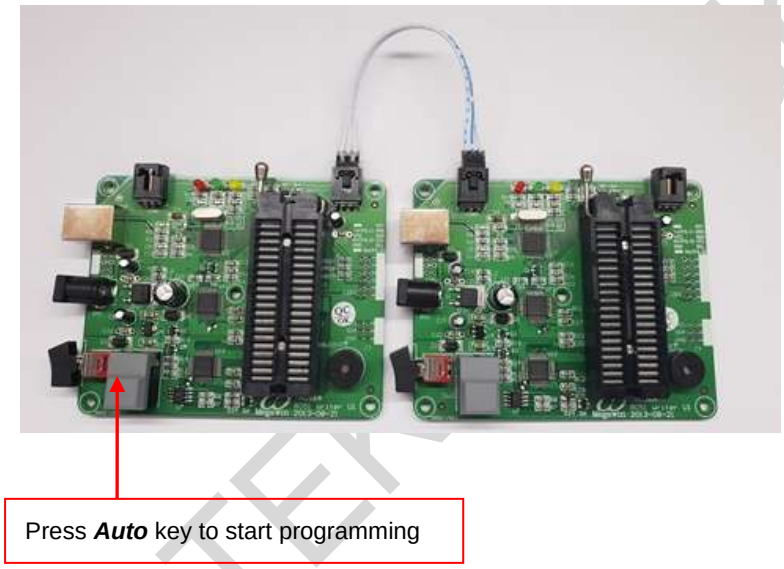


Figure 19. Offline Copying Programming Connection

3 Serial Number Usage

The serial number consists of 12 characters (ASCII code) plus a 4-byte number and is stored between the starting address and the end address. For different chips, the *string* is the same; if the auto increment number flag is checked, the number will be automatically incremented by 1 for each chip programmed.

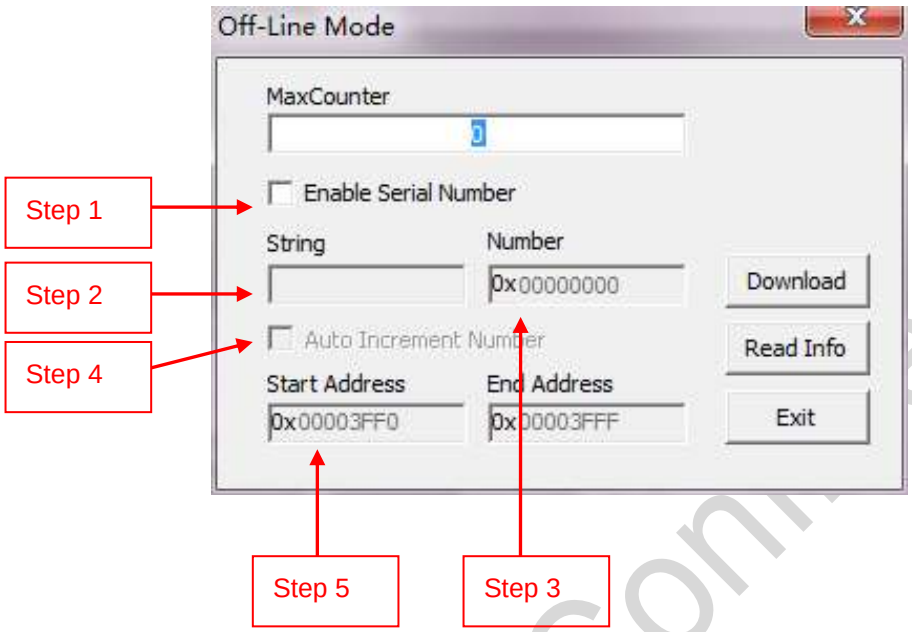


Figure 20. Series Number Usage

The operation flow is as follows.

Step 1: enable the serial number function. The serial number has a total of 16 bytes (a 12-byte string and a 4-byte number).

Step 2: the is string, up to 12 bytes of characters ,is used for factory or product name

Step 3: the number, up to 4 bytes with a valid value range as 0x00000000~0xFFFFFFFF.

Step 4: When automatic programming completes (see 2.2.6 Press 'Auto' in the programmer to start programming), the number set in the third step will be automatically increased by 1.

Step 5: The starting address of the serial number. Its maximum value is the chip size minus 16, and please make sure that the area range is not used yet.

- For example. based on the setting in the below figure, after programming the first chip, the information stored in 0x3500 ~ 0x350F is as follows.

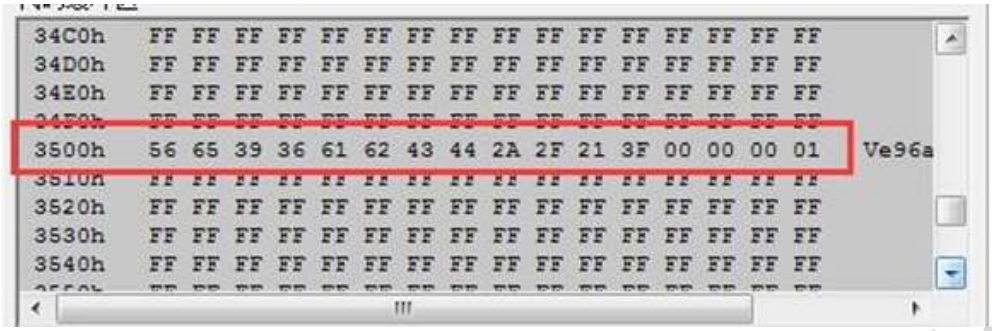


Figure 21. Information in Code Buffer - 1

Based on the setting in the below figure, after programming the second chip, the information stored in 0x3500~0x350F is as follows.

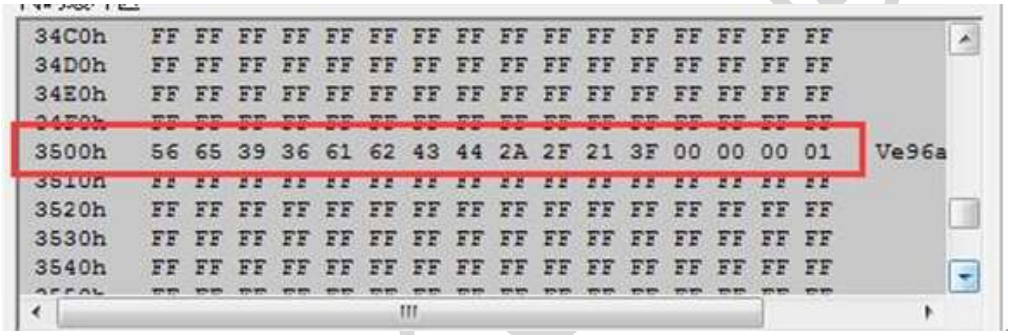
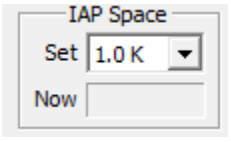


Figure 22. Information in Code Buffer - 2

4 Special Consideration

- IAP Space



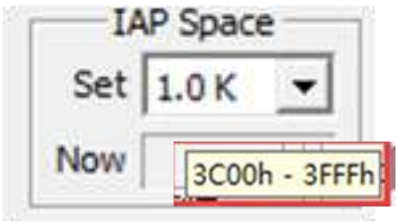


Figure 23. IAP Space

This value determines the the IAP space size. Users can also change the IAP space size by changing the register IAPLB in the program (IAP start address = IAPLB*256, IAPLB must be even).

2. The Chip provides 16 K Flash。 The space near to 0x0000 address is for user AP program, near to 0x3FFF (ie 16K) address is for ISP code and the space between the 2 spaces is for IAP (including IAP (data), serial number and IAP space). Pleas make sure the 3 spaces setting are not overlapped (Note that the minimum unit of Flash area division is a sector, which is 512 bytes).

3. If the contents of IAP (Data) and Serial Number are not changed after programming, it is highly recommended not put them into the same sector with IAP Space. Otherwise, when IAP is erased in the program, IAP (data) and serial number will also be erased. Suggest put them between tje AP (program) space and IAP space space.

- Firmware upgrading

When the following message shows up that means the U1 writer needs to be updated firmware. Please make sure the USB cable should **NOT** unplug during the updating process. Otherwise, The U1 writer will cause unrecoverable damaged.

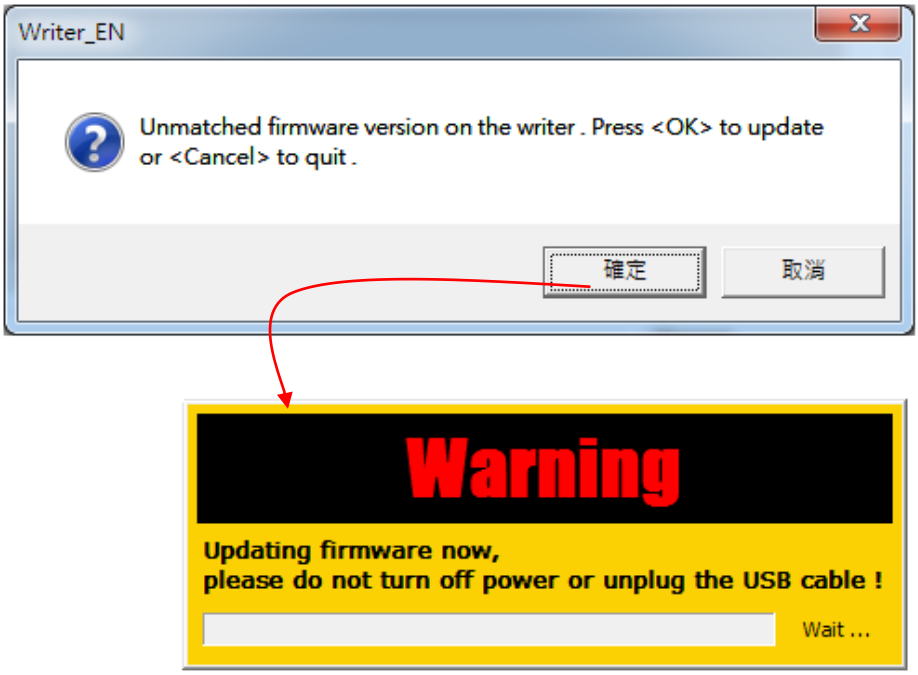


Figure 24. Firmware Upgrading Warning

5 Appendix

5.1 Chip Storage Space Configuration

The storage space of the CMT2380F16 is divided into the 3 categories.

- AP space: used to store client applications and data. This part of the data can be erased, read and written by the programmer and ISP program.
- IAP space: It is a non-volatile data storage space that can be used as an EEPROM. This part of the data can be erased, read and written by the programmer, ISP program and AP application.
- ISP space: It is a special storage space, where program code can run independently. It is generally used for online programming of AP and IAP space, and the storage space of ISP itself can only be programmed by the programmer.

CMT2380F16 provides a total of 8 Kbytes (0000h ~ 1FFFh) storage space as shown in the below figure.

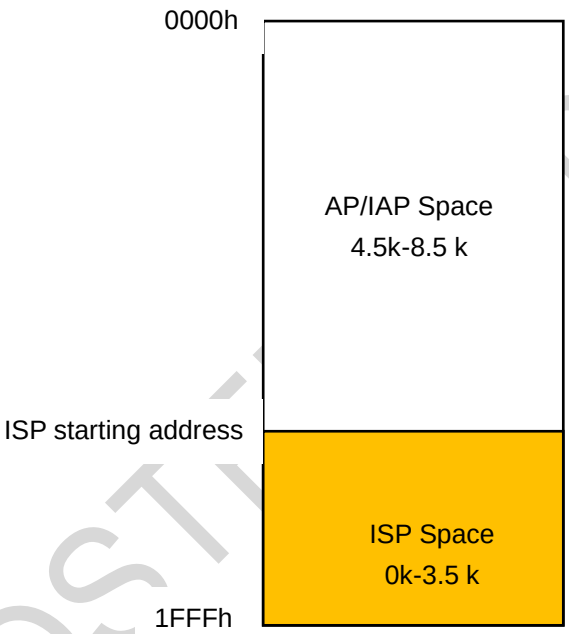


Figure 25. CMT2380F16 Storage Space

The ISP space is configured by the user through the programmer with ISP space ranging from 0 kbytes to 3.5 kbytes. As the IAP space and AP space of the CMT2380F16 are shared, apart from the ISP space, the rest of 8K space is the AP and IAP space.

5.2 Chip Hardware Options

- **LOCK**

  Enable : Code dumped on a Writer is locked to 0xFF for security

  Disable : Code dumped on a Writer is not locked

- **SB**

  Enable : Code dumped on a Writer is scrambled for security

  Disable : Code dumped on a Writer is not scrambled

- **MOVCL**

  Enable : MOVC instruction executed from external program memory is disabled for security.

  Disable : MOVC is always enabled.

- **EN6T**

  Enable : MCU runs at 6T mode (each machine-cycle has 6 clocks).

  Disable : MCU runs at 12T mode (each machine-cycle has 12 clocks)

- **HWBS**

  Enable : When power-on, MCU will boot from ISP-memory if ISP-memory exists

  Disable : (No action)

- **OSCDN**

  Enable : Used under 16MHz for EMI reduction. (The gain of oscillating amplifier is reduced.)

  Disable : The gain of crystal oscillator is enough for higher Fosc oscillating

- **AUX_RAM**

  Enable : The internal auxiliary RAM access is disabled when the ERAM bit =0 (AUXR.bit1) and when ERAM bit =1 the internal auxiliary RAM access is enable

  Disable : The internal auxiliary RAM access is enable when the ERAM bit =0 (AUXR.bit1) and when ERAM bit =1 the internal auxiliary RAM access is disabled

- **FZWDTCR**

  Enable : The WDTCR register will be initialized to its reset value only by power-on reset

  Disable : The WDTCR register will be initialized to its reset value by all reset (including power-on, H/W, S/W and WDT reset)

- **ENLVR**

  Enable : Enable Low-Voltage Reset (LVR) , the LVR is around 2.4V for 3.3V device and 3.7V @12MHz for 5.0V device

  Disable : Disable LVR

- **LVFWP**

  Enable : Enable LVFWP (Low-Voltage Flash-Write Protection) while IAP or ISP programming

  Disable : Disable LVFWP

- **ENROSC**

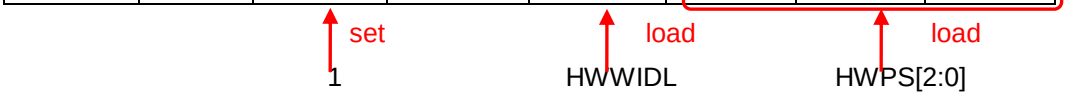
  Enable : MCU will used the internal 6MHz oscillator

  Disable : MCU will used the external oscillator

- **HWENW**

  Enable : Automatically enable Watch-dog Timer by hardware when MCU is powered up

7	6	5	4	3	2	1	0
WRF	-	ENW	CLRW	WIDL	PS2	PS1	PS0



Disable : (No action)

- **WDSFWP**

  Enable : To Write the WDTCR will be deny

  Disable : To Write the WDTCR will be accept

- **HWBS2**

  Enable : In addition to power-up, the reset from RST-pin will also force MCU to boot from ISP-memory, if ISP-memory is configured

  Disable : Where MCU boots from is determined by **HWBS**

- **ENLVRO**

  Enable : Enable Low-Voltage Reset (LVR) when Vdd less then 3.7V

  Disable : Disable LVR

- **ENLVRC**

  Enable : Enable Low-Voltage Reset (LVR) when Vdd less then 2.5V

  Disable : Disable LVR

- **BODRE**

  Enable : Enable Low-Voltage Reset (LVR)

  Disable : Disable LVR

- **BODWP**

  Enable : Enable LVFWP (Low-Voltage Flash-Write Protection) while IAP or ISP programming

  Disable : Disable LVFWP

- **P40IOE**

  Enable : Enable the P4.0 is in “Input Only mode” after Power On Reset

  Disable : The P4.0 is in default “Quasi-Bidirectional mode”

- **P41IOE**

  Enable : Enable the P4.1 is in “Input Only mode” after Power On Reset

  Disable : The P4.1 is in default “Quasi-Bidirectional mode”

- **{BO1S1O,BO1S0O}**

  {0, 0} : BOD1 detects the level at 2.0V on VDD

  {0, 1} : BOD1 detects the level at 2.4V on VDD

  {1, 0} : BOD1 detects the level at 3.7V on VDD

  {1, 0} : BOD1 detects the level at 4.2V on VDD

- **BO0REO**

  Enable : BOD0 will trigger a RESET event to CPU on AP program start address

  Disable : BOD0 can not trigger a RESET to CPU

- **BO1REO**

  Enable : BOD1 will trigger a RESET event to CPU on AP program start address

  Disable : BOD1 can not trigger a RESET to CPU

- **NSWDT**

Enable : Enable the WDT to run in power down mode

Disable : Disable the WDT to run in power down mode

6 Revise History

Table 2. Revise History Records

Version No.	Chapter	Description	Date
0.5	All	Initial version	2019-07-19

7 Contacts

CMOSTEK Microelectronics Co., Ltd. Shenzhen Branch
Address: 2/F Building 3, Pingshan Private Enterprise S.T. Park, Xili, Nanshan District, Shenzhen, Guangdong, China

Tel: +86-755-83231427
Post Code: 518071
Sales: sales@cmostek.com
Supports: support@cmostek.com
Website: www.cmostek.com