

megawin

OCD32_MLink

User Manual

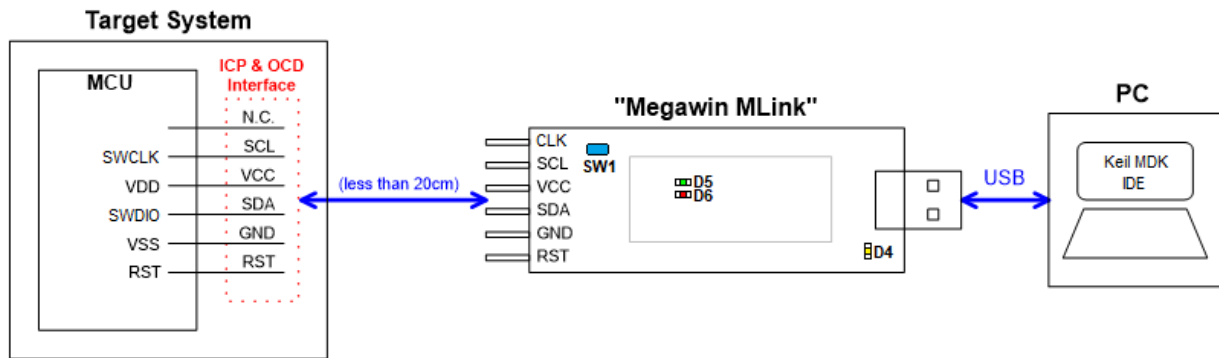
Index

Index	2
1. Introduction	3
2. Setup MLink and Pack for Keil MDK.....	4
3. Update Programmer	10
4. Update Target	16
5. Note	17
6. Revision History	18

1. Introduction

OCD32_MLink is an ICE tool used by megawin in Keil MDK. It supports only Cortex®-M0 series chips of megawin. This development kit provides OCD (On-Chip-Debug) real-time debugging function. In addition, the "ICP32_Programmer.exe" provided in the package is a software designed for megawin's MLink. Users can update the application code under the software tool without removing the mounted MCU chip from the actual end product. In addition, because the programming data to be programmed to the target can be saved in the non-volatile storage of MLink, this stand-alone programmer is able to work without host (PC) intervention. This feature is especially useful in the field without a PC. Users only need to reserve 5 pins such as SWDIO, SWCLK, VCC, GND and RST to connect to MLink.

PS. The VCC pin of MLink does not provide power to the target board, the users need to provide the power by himself.



(Figure 1)

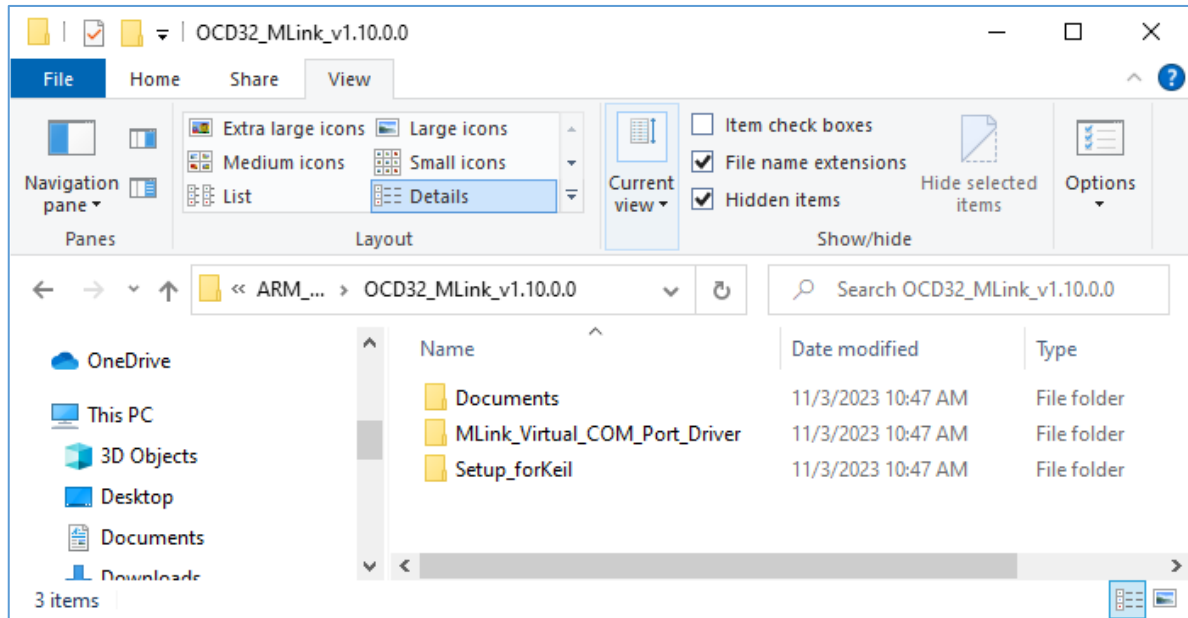
Light display	Always on	Always off	Flashing
D4	USB connection is normal	USB connection is not normal ^{*1}	
D5	1. Device initialization is normal 2. Programming the result is Pass		Programming in offline mode
D6	1. Device initialization is not normal ^{*2} 2. Programming the result is Fail		Programming in offline mode

PS1. When the USB connection is not normal, please check whether the power supply is normal and whether the system has recognized the device.

PS2. The device initialization is abnormal, indicating that the offline mode content is incorrect. Please set up the offline mode again.

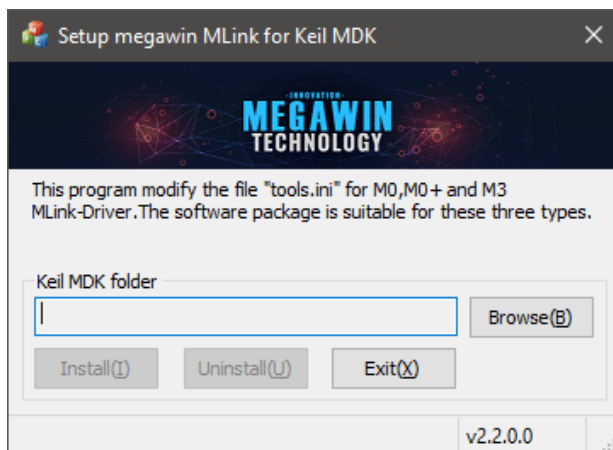
2. Setup MLink for Keil MDK

2.1. Install the package , there are some folders as figure 2.



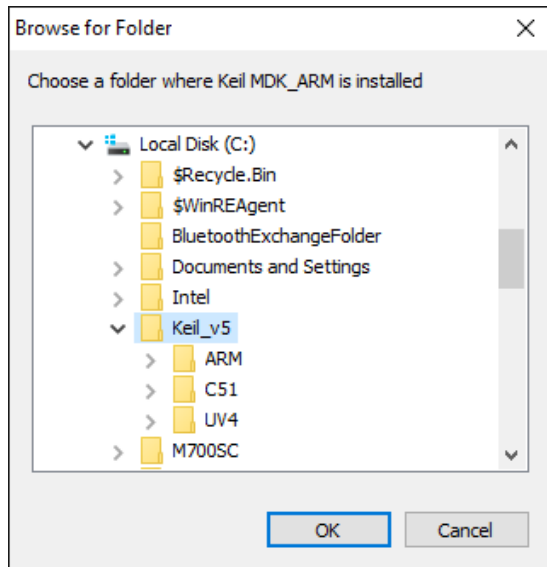
(Figure 2)

2.2. Please execute the “**SetupforKeilMDK.exe**” in “**Setup_forKeil**”, the program will show as figure 3. User can press button called “**Update MLink Firmware for CM0**” to update firmware.

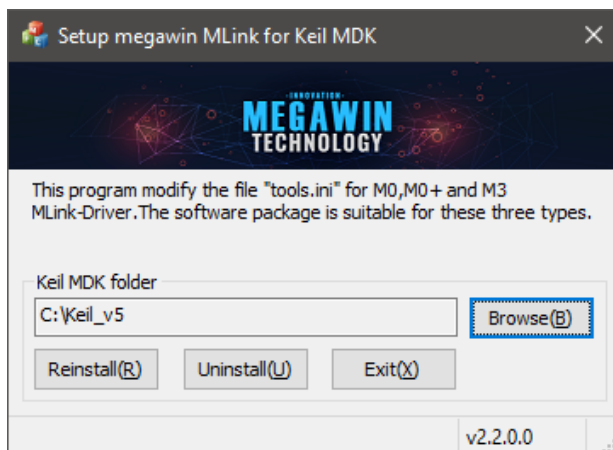


(Figure 3)

2.3. Click the **“Browse”** button to select the root directory of Keil MDK (only available for MDK4 and MDK5) as figure 4 and click **“OK”**, then it will be as figure 5.

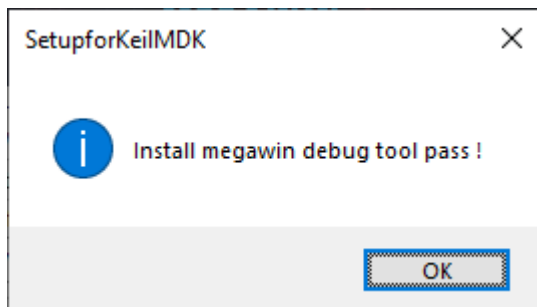


(Figure 4)



(Figure 5)

2.4. Click “Install” button, the application will add configuration in the file of “tools.ini”



(Figure 6)

2.5. Click “OK” button, when it is completed.

2.6. Click “Exit” to finish the install procedure.

2.7. Execute “UV4.exe” and open the sample project “Standard_Project”

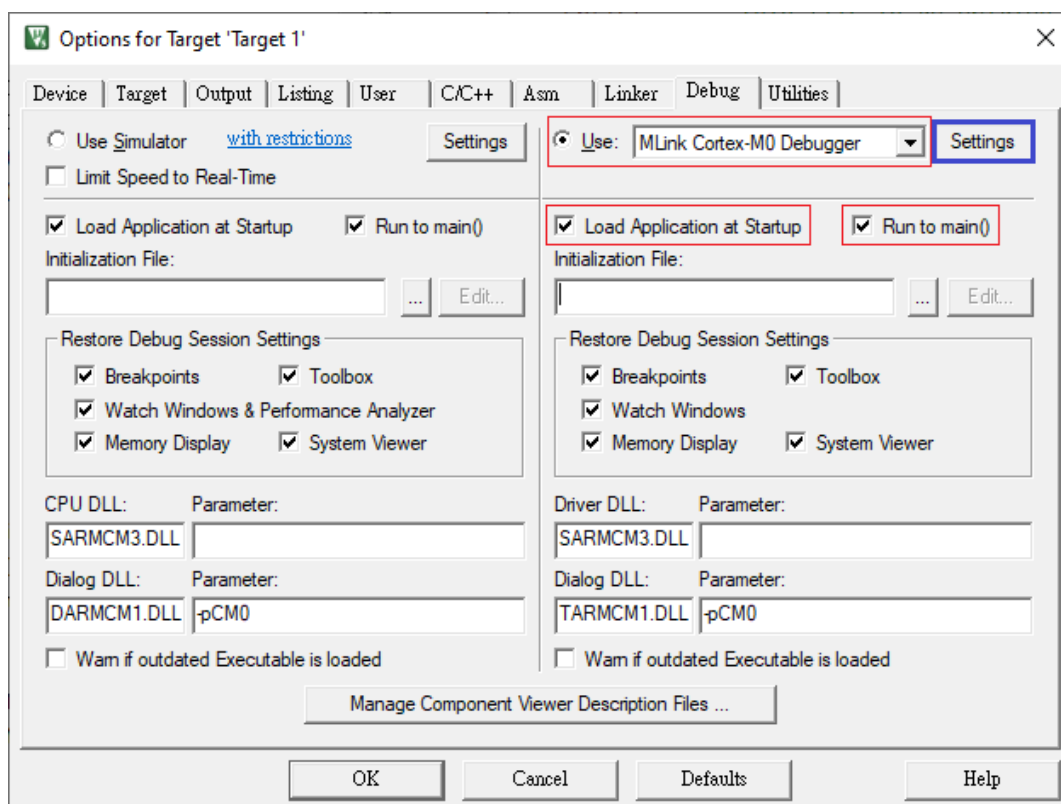
2.8. Click the menu of “Project” and select “Options for Target ‘Target 1’...”.

2.9. Select the “Debug” tag as figure 7.

2.9.1. Please select the appropriate debug driver based on the device’s category.

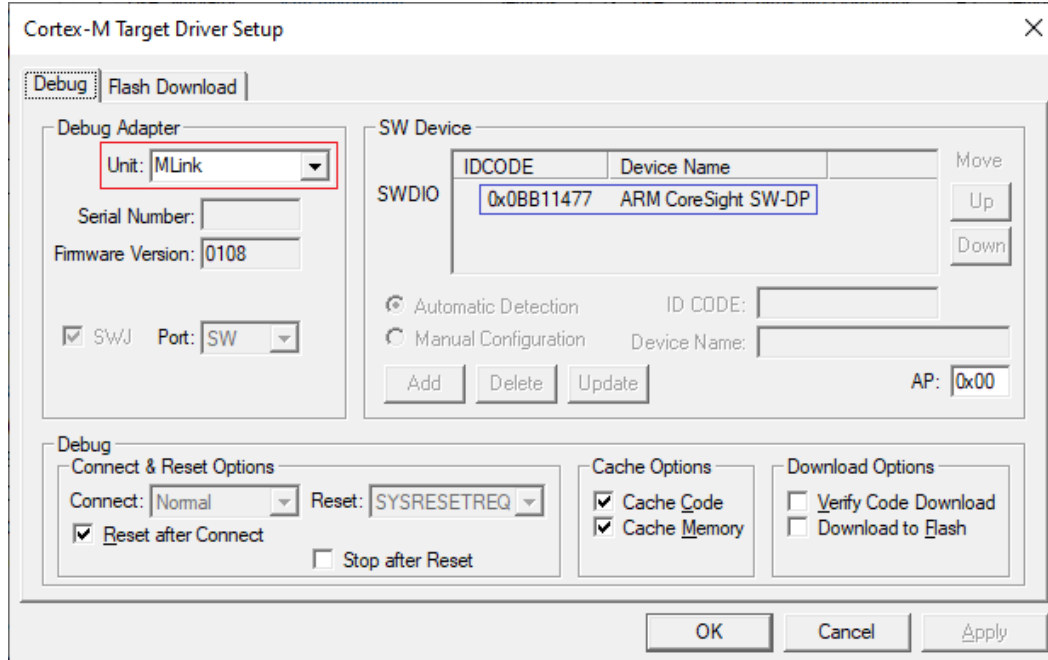
2.9.2. Check “Load Application at Startup”.

2.9.3. Check “Run to main()”.



(Figure 7)

2.10. Reference to figure 7, click the “**Setting**” button, another dialog will show as figure 8. Configure the debug adapter as figure 8. If the hardware are working, there will be a SW Device shows in list control on the right side of the dialog.



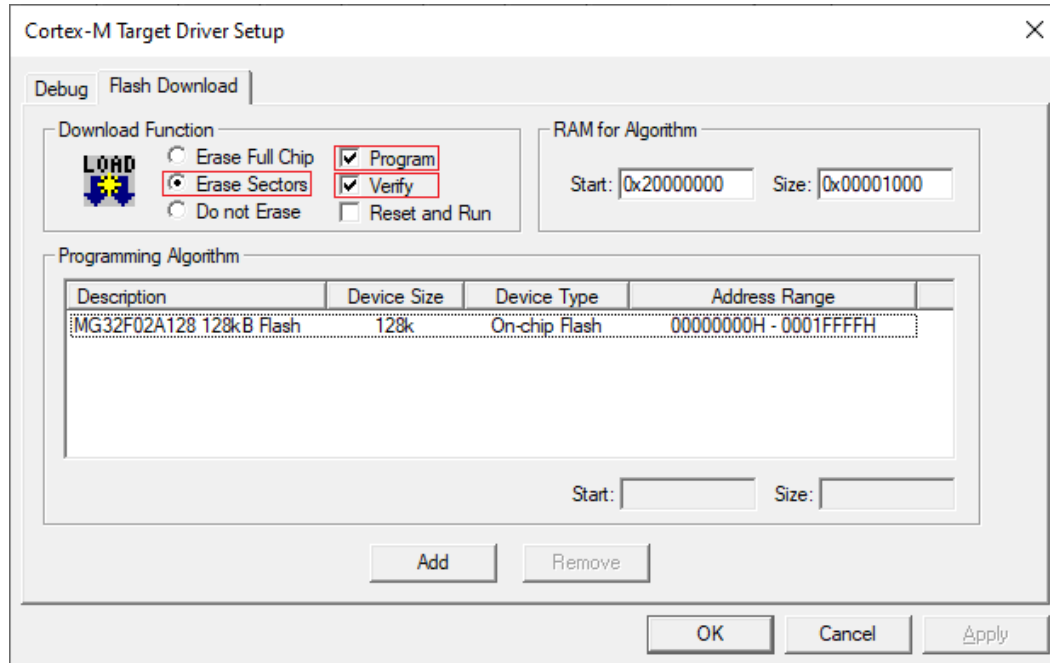
(Figure 8)

2.11. Click the tag of “Flash Download” as figure 9.

2.11.1. Click “Erase Sectors”

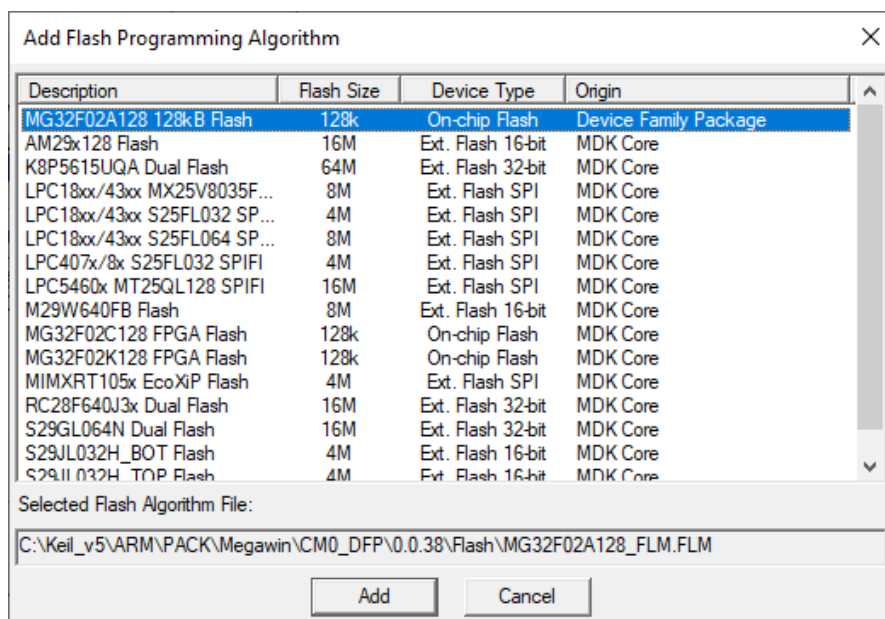
2.11.2. Check “Program” and “Verify”

2.11.3. Click “OK” button



(Figure 9)

If program algorithm is not showed as figure 9, please click “Add” to select the flash programming algorithm as figure 10



(Figure 10)

2.12. Rebuild the sample project “Standard_Project”

2.13. Click the menu of “Flash” and select “Download”.

2.14. Start to debug the project.

Note 1 : Before using MLink at the first time. Please execute “**ICP32_Programmer.exe**” and press “**Check ID**” to update the firmware of MLink

Note 2 : When the chip is locked, we can’t run in debug mode with MLink in Keil IDE.
Please execute “**ICP32_Programmer.exe**” and click “**Update Target**” to unlock the chip.

3. Update Programmer

The following figure shows the graphic user interface of the PC-site application program. The following sections will demonstrate how this tool can be used very easily.

Step 1: Choose a “MCU Series” and then choose the “MCU Part No”.

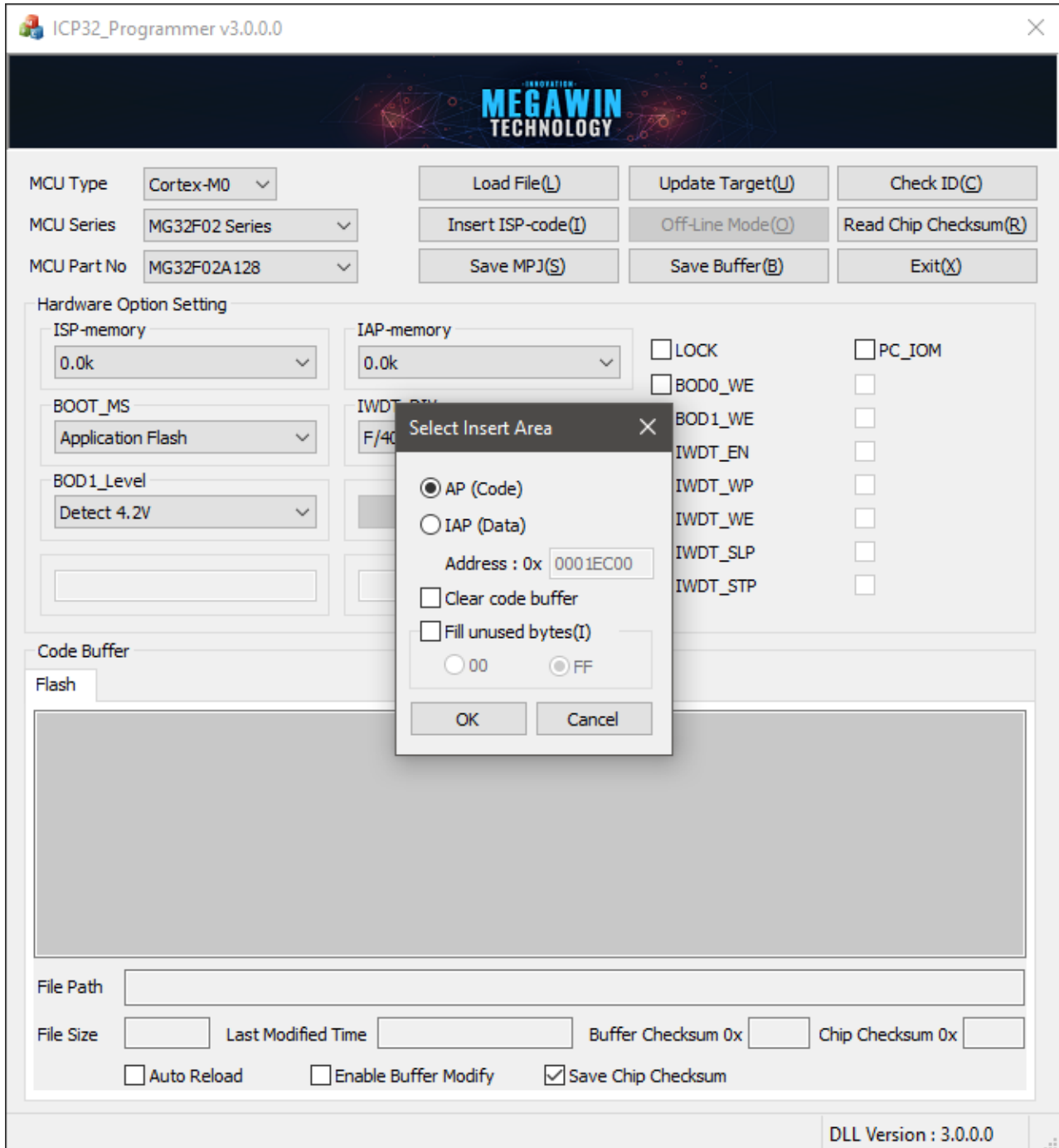


(Figure 11)

Note 1 : Auto Reload : When the application is from inactive to active. The code buffer will be reload.

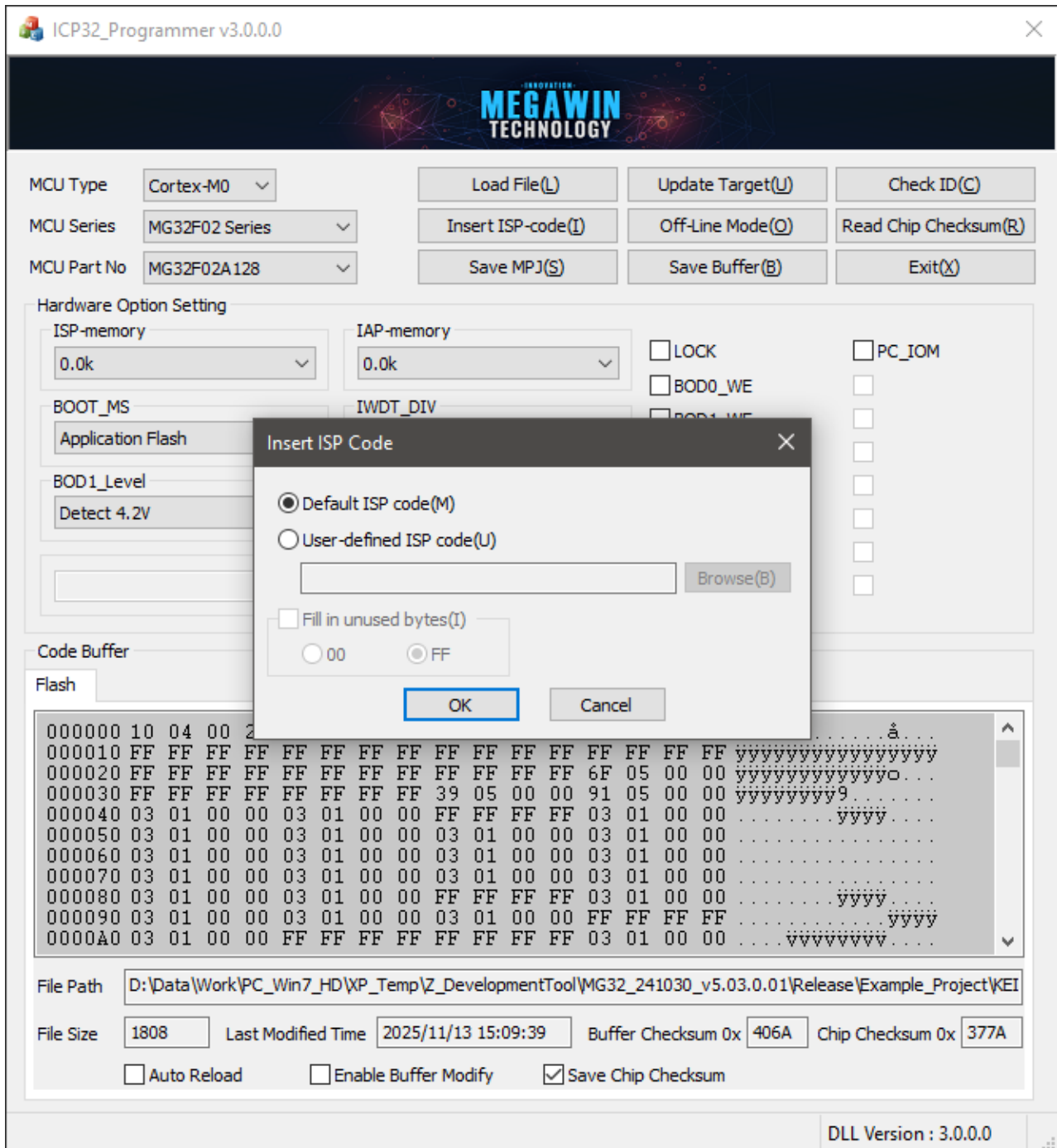
Note 2 : Enable Buffer Modify : User can modify the code buffer directly with keyboard.

Step 2 : Click “Load File” to select a file then choose loading AP(Code) or IAP(Data). “Load File” can be clicked repeatedly to load different files. While loading IAP(Data), user have to key in address. HEX and BIN data formats are supported for file loading.



(Figure 12)

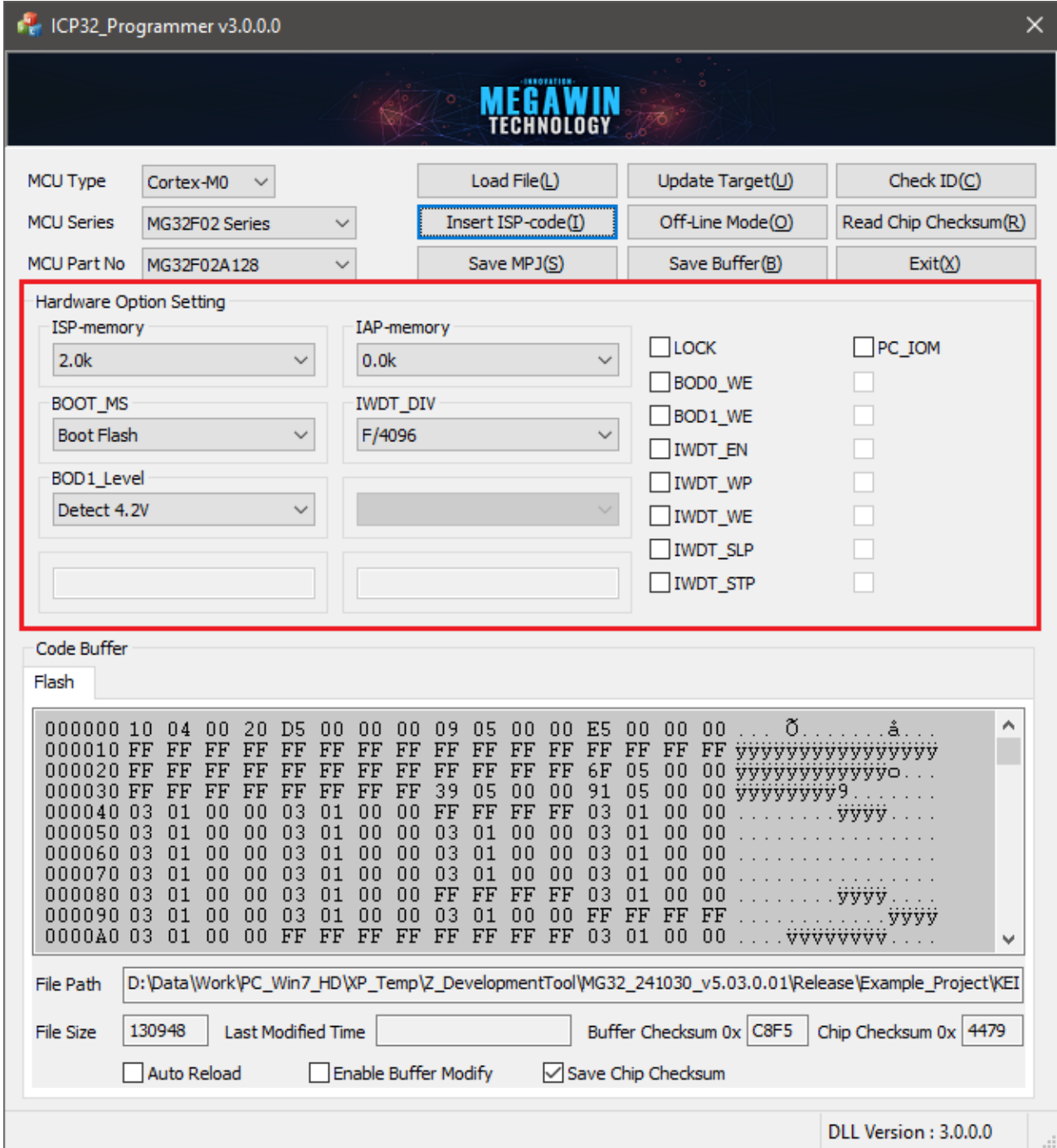
Step 3: Click “Insert ISP-Code” may choose to insert megawin-provided ISP code or User-defined ISP Code. If ISP function is not needed, Step 3 can be omitted.



(Figure 13)

Step 4: H/W Option setting

The hardware option defines the chip default behavior those are not volatile after power off. For details of the hardware options, please refer to user guide.



ICP32_Programmer v3.0.0.0

MEGAWIN TECHNOLOGY

MCU Type: Cortex-M0
 MCU Series: MG32F02 Series
 MCU Part No: MG32F02A128

Buttons: Load File(L), Update Target(U), Check ID(C), Insert ISP-code(I), Off-Line Mode(O), Read Chip Checksum(R), Save MP1(S), Save Buffer(B), Exit(X)

Hardware Option Setting

ISP-memory: 2.0k
 IAP-memory: 0.0k
 BOOT_MS: Boot Flash
 IWDI_DIV: F/4096
 BOD1_Level: Detect 4.2V

Options:

- ☐ LOCK
- ☐ BOD0_WE
- ☐ BOD1_WE
- ☐ IWDI_EN
- ☐ IWDI_WP
- ☐ IWDI_WE
- ☐ IWDI_SLP
- ☐ IWDI_STP
- ☐ PC_IOM

Code Buffer

Flash

```

000000 10 04 00 20 D5 00 00 00 09 05 00 00 E5 00 00 00 ... 0. ....
000010 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF yyyyyyyyyyyyyyyy
000020 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF yyyyyyyyyyyyyyyy
000030 FF FF FF FF FF FF FF FF FF 39 05 00 00 91 05 00 00 yyyyyyyy9
000040 03 01 00 00 03 01 00 00 FF FF FF FF 03 01 00 00 ..... yyy
000050 03 01 00 00 03 01 00 00 03 01 00 00 03 01 00 00 .....
000060 03 01 00 00 03 01 00 00 03 01 00 00 03 01 00 00 .....
000070 03 01 00 00 03 01 00 00 03 01 00 00 03 01 00 00 .....
000080 03 01 00 00 03 01 00 00 FF FF FF FF 03 01 00 00 ..... yyy
000090 03 01 00 00 03 01 00 00 03 01 00 00 FF FF FF FF ..... yyy
0000A0 03 01 00 00 FF FF FF FF FF FF FF FF 03 01 00 00 ..... yyy
  
```

File Path: D:\Data\Work\PC_Win7_HD\XP_Temp\Z_DevelopmentTool\MG32_241030_v5.03.0.01\Release\Example_Project\KEI

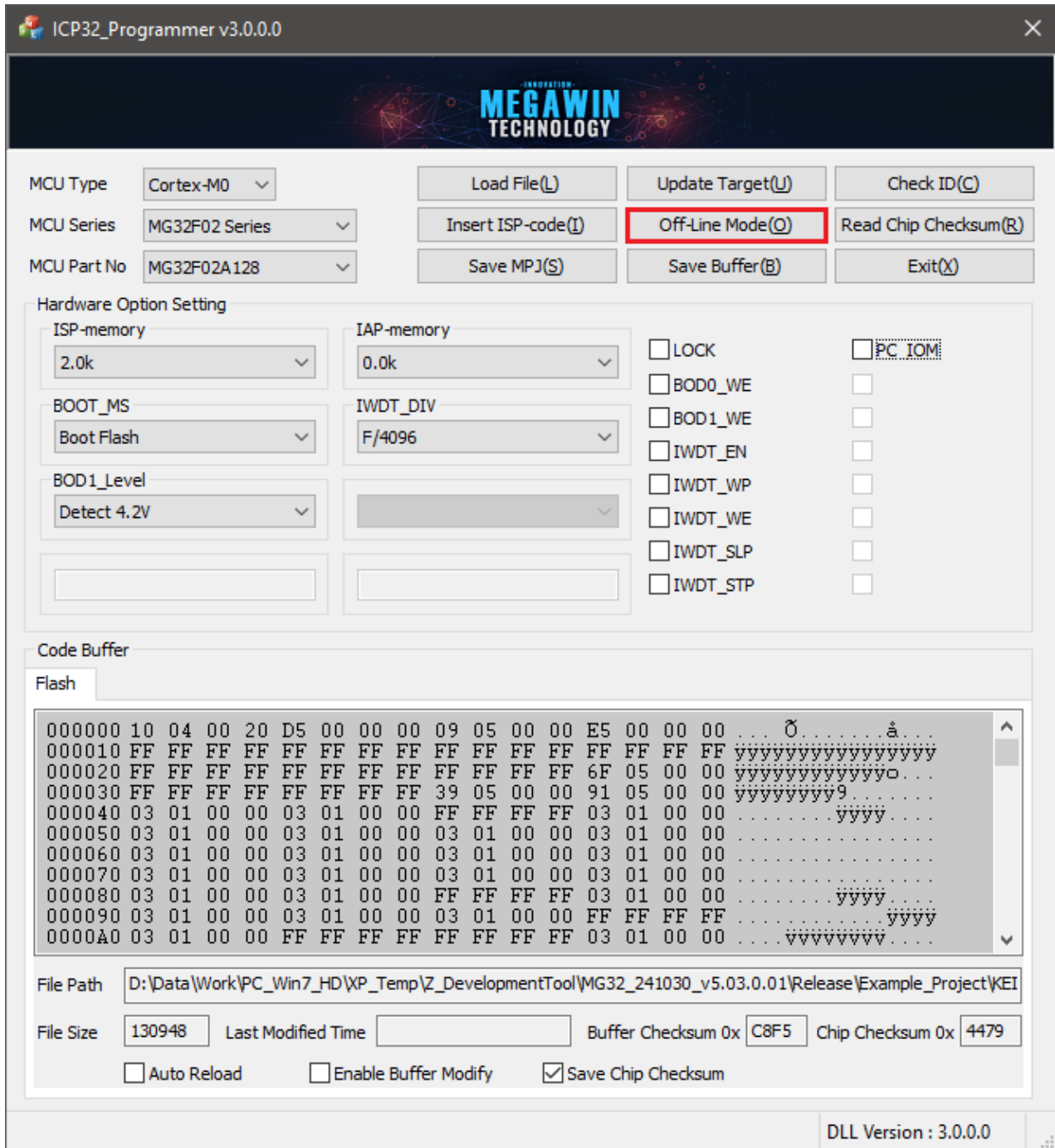
File Size: 130948 Last Modified Time: Buffer Checksum 0x: C8F5 Chip Checksum 0x: 4479

☐ Auto Reload ☐ Enable Buffer Modify ☒ Save Chip Checksum

DLL Version : 3.0.0.0

(Figure 14)

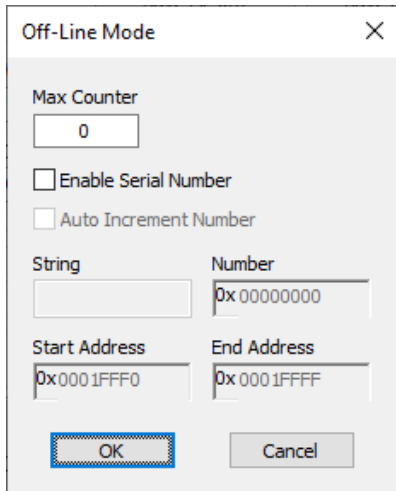
Step 5: Click “Off-Line Mode” to download programming data to the MLink.



(Figure 15)

Step 6: Setup “Max Counter” and “Serial Number”.

The “Max Counter” will be used to limit the number of off-line programming. The “Serial Number” will be programmed along with the code. If the function is not needed, just press “OK” to update the MLink.

The image shows a software dialog box titled "Off-Line Mode" with a close button (X) in the top right corner. Inside the dialog, there are several input fields and checkboxes. At the top, there is a "Max Counter" section with a text box containing the value "0". Below this are two unchecked checkboxes: "Enable Serial Number" and "Auto Increment Number". Further down, there are two columns of input fields. The left column has a "String" label above an empty text box, and a "Start Address" label above a text box containing "0x0001FFF0". The right column has a "Number" label above a text box containing "0x00000000", and an "End Address" label above a text box containing "0x0001FFFF". At the bottom of the dialog are two buttons: "OK" and "Cancel". The "OK" button is highlighted with a blue dashed border.

(Figure 16)

How to use the Serial Number :

- (1) Enable the Serial Number function and totally 16 bytes. (12 bytes for **String** and 4 bytes for **Number**) will be used.
- (2) Totally 12 bytes for the **String** and it could be used for manufactory or product string
- (3) Totally 4 bytes for the **Number** and the value could be from 0x00000000 to 0xFFFFFFFF.
- (4) The Number (as set in step 3) will be automatically added one when finish the “Download”.
- (5) **Start Address** for the Serial Number. It is limited from the chip size minus sixteen, please make sure this range from start to end is unused.

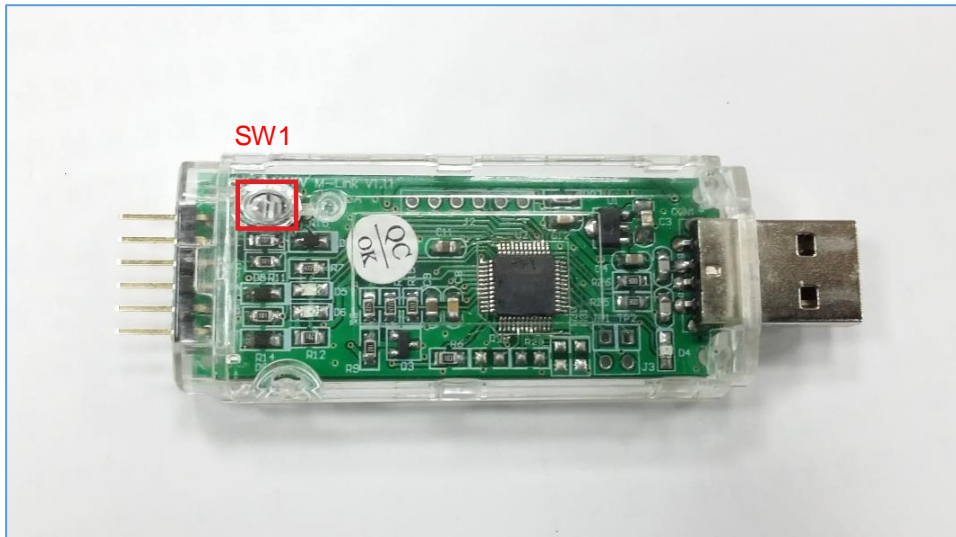
4. Update Target

How to update the target? User may:

4.1. Click “**Update Target**” to program on-line update, referring to steps 1 through 4 of 3.

Update Programmer, or

4.2. Click “**SW1**” of MLink to program off-line update, referring to 3. Update Programmer.



(Figure 17)

5. Note

- 5.1. After using ICP32_Programmer to update the target, the chip normally is unlock.
- 5.2. When the applications are using more I/O, it may configure the PC4, PC5 and PC6 as GPIO. But these 3 PINs are also using for the communication for MLink and target chip. After program PC4, PC5 and PC6 as GPIO, it will disable SWD interface, and it will disconnect the debug mode (ICE).
And due to SWD interface is disabled, the ICP function will also be disabled too. In this case, we suggest to finish all other function debug in your project and then test PC4, PC5 and PC6 GPIO function without debug mode. When you want to reload the code after modify to do the retest, it should follow the steps show in following:
 - 5.2.1. Power off the target chip
 - 5.2.2. Press and hold the reset button (RSTN/PC6 Pin) on the target board
 - 5.2.3. Power on the target chip
 - 5.2.4. Click “Update Target” in ICP32 Programmer AP, and release the Reset button after two seconds.
- 5.3. Auto Reload: When the source of code buffer is changed, the application will be notified to reload the file.
- 5.4. Enable Buffer Modify : Enable users to modify the code buffer.
- 5.5. Save Checksum : only support MG32F02A032, MG32F02A128, MG32F02A064, MG32F02U128, MG32F02U064 and MG32F02V032. Store the checksum of code buffer in a specific location.
- 5.6. Fill in unused bytes : If the item is checked, users can specify to fill the unused bytes with 0x00 or 0xFF. Except the loaded file, the remaining space will be filled with the unused bytes. The code buffer size will be the same as the chip size.

6. Revision History

Revision	Description	Date
v2.00	New version of application release. Support MLink Hardware V14 and V16.	2024/09/26
v2.10	Add support MG32F003F4U7, MG32F003F4P7, MG32F005K6U7, MG32F005G6P7	2024/11/01
v2.20	Fix the start address check mechanism issue in off-line mode	2024/11/29
v2.30	Fix a bug in hex file conversion.	2025/06/06
v2.42	Add “Save Checksum” function	2025/07/09
v2.43	Fix a bug about MG32F02U064	2025/08/07
v2.44	Fix a bug.	2025/08/11
v2.45	Add “Fill unused bytes” in “Load File” and “Insert ISP-code”. See the description in Note 5.6.	2025/09/25
v2.50	Add MG32F02N064/N128/K064/K128	2025/11/03
v3.0.0	Integrate M0, M3 and M0+.	2026/01/12