1.Programmer Overview

This instruction is suitable for the Xgecu T48/T76 universal programmer, which supports multi-language menus. it can be run on WINXP/WIN7/WIN10/WIN11.

Copyright Notice

Software Copyright **2010-2025** XGecu

User's Guide Copyright **2010-2025** XGecu

The distribution and sales of the product are intended for use by the original purchaser under the terms of the License Agreement. This document may not, in whole or part, be copied, photocopied, reproduced, translated or reduced to any electronic medium of machine-readable form without prior consent in writing from XGecu.

XGecu® is the registered trademark of Xingong Electronics Co., Ltd..

2. Software Installation

2.1 Download and install the application software

Download the XgproVxxxx_setup.exe file from HTTP://www.xgecu.com, and double-click the file to enter the installation interface. As follows: (The installation file contains the contents of this manual.) Click [Install] to start installing the application, and the following screen will pop up: Enter the folder where you want to install. Click [Install] to start the installation of the application

---

IMPORTANT:

1. T76 can only achieve high-speed performance when connected via USB3.0.

2. Please insert the USB3.0 directly into the USB port on the motherboard, especially the front-mounted port of an assembled desktop computer. The internal connecting cables of the chassis generally do not meet the requirements of USB3.0 and should be disabled!

3. When using an external USB HUB, it is necessary to use products from reputable brands

---

Click [Install], after the file is installed to the hard disk. The USB driver installation dialog box is automatically performed. As below:

2.2 Device Installation

After the above installation is completed, connect programmer into the USB port and the operating system will discover the hardware as below:

Select [Auto Install Software], click [**next**] button, it will automatically install until the installation is successfully completed.

(Note: In this process, the driver will change the registry. If the firewall software is installed, please release this software. Windows will also prompt the driver have no digital visa, confirm and continue the installation).

If the automatic installation is not successful, you can also choose to install in <from list or specified location>, USB driver files are
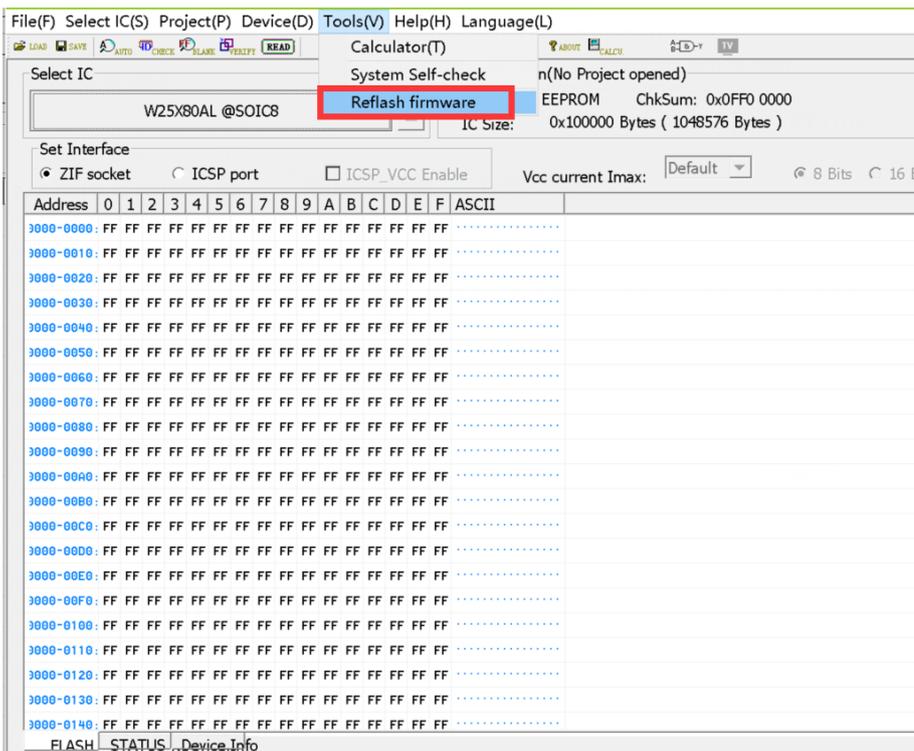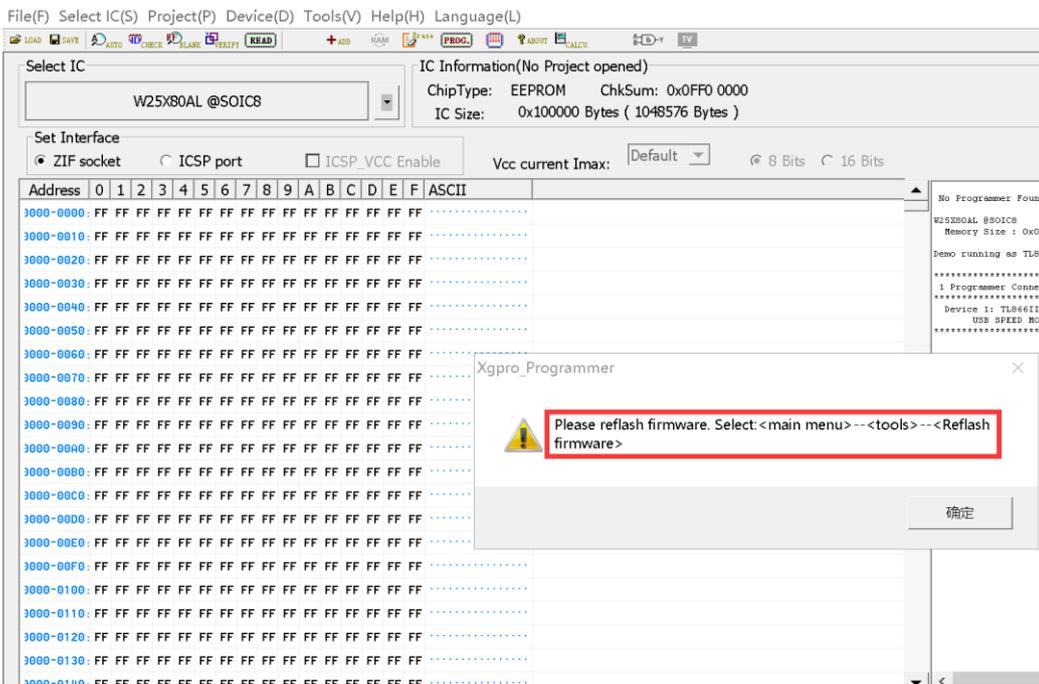
in the installation directory (D:\xgpro\drv). After installation, double-click the desktop icon and run the programmer software. Ready to use.

2.3 Firmware FLASH Refresh

When using at the first time or upgrading, you may need to refresh the firmware. In the main menu of the software, do a refresh operation as shown below (**Tools-Reflash firmware**):

IMPORTANT:

1. T76 can only achieve high-speed performance when connected via USB3.0.

2. Please insert the USB3.0 directly into the USB port on the motherboard, especially the front-mounted port of an assembled desktop computer. The internal connecting cables of the chassis generally do not meet the requirements of USB3.0 and should be disabled!

3. When using an external USB HUB, it is necessary to use products from reputable brands

---

File(F) Select IC(S) Project(P) Device(D) Tools(V) Help(H) Language(L)

File(F) Select IC(S) Project(P) Device(D) Tools(V) Help(H) Language(L)

If there is an error during firmware refresh, it is often because the programmer cannot be reset during the refresh and the programmer cannot do the refresh operation.

The reason of error:

It is usually caused by using an external USB hub. Please do not use an external HUB when refreshing. Please directly connect the programmer to the USB port of your computer to refresh. If it still cannot solve your problem, please try to restart your computer or do the refresh operation on another computer.

3. Quick Start

## 3.1 Hardware preparation and select the correct chip model

Firstly, please make sure the universal programmer application software and USB driver are correctly installed. Then connect programmer to your PC by using USB cable. At this point, the "POW" power indicator of the programmer keeps on and the "RUN" indicator goes off, that means the power connection is normal and you can go to the next step.

(If the RUN indicator blinks, that means the USB driver is installed incorrectly or don't install the USB device driver)

Note: Please be sure to use the original USB cable. If you use HUB, make sure it is the high quality HUB and the USB voltage should not be lower than 4.90V. Low voltage is usually caused by using poor quality USB cable or HUB, that will make the programmer work unstably.

As shown below：

1) Click the **[Select IC]** button, the select Device dialog box pops up.

2) In the search IC, type in the IC model (for example W29C020C)

3) Select the correct manufactory (if there are more than one)

4) Select the correct model. Note: @PLCC32 @TSOP32 refer to the IC package, none: dual-in-line package

5) Click   **[Select]** button

6) Note: Show that the programmer is connected
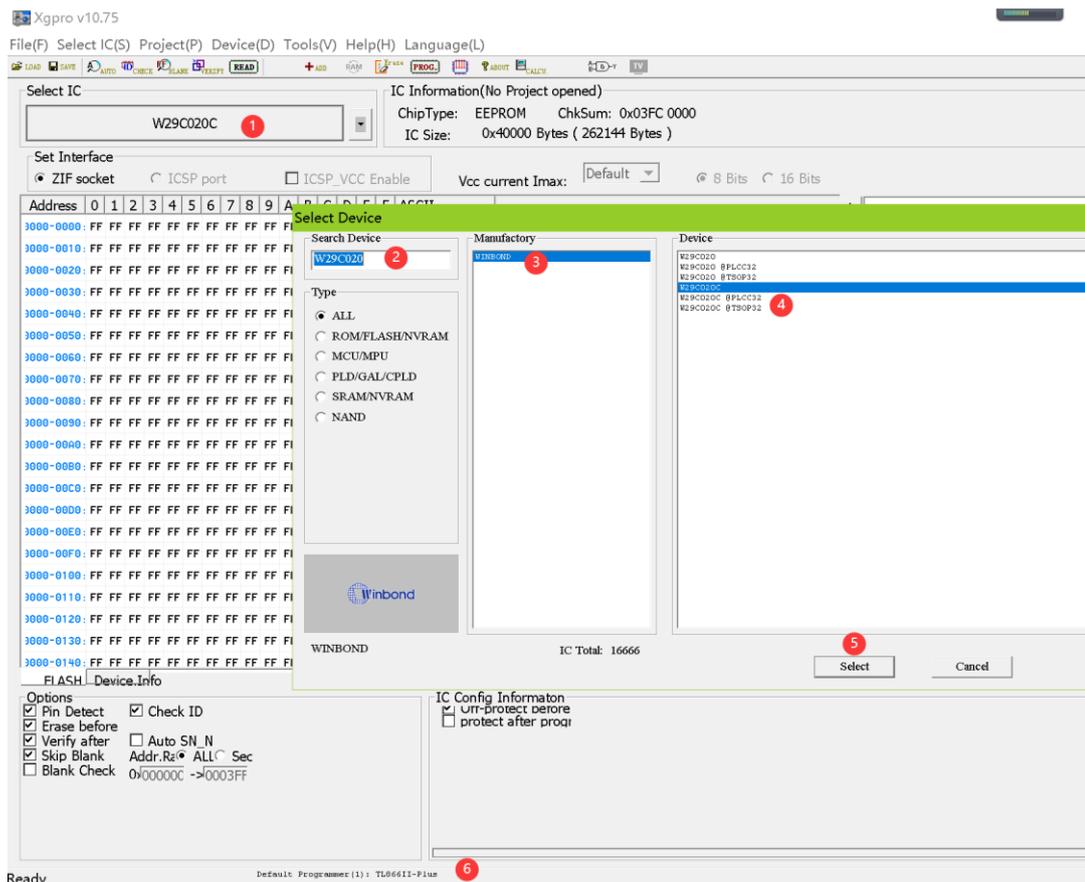
## 3.2 Read and Save

After selecting IC in section 3.1 steps

1) Click the **[Read]** button on the toolbar and pop up the Chip Read dialog box.

2) In the Chip Read dialog box, click the **[Read]** button.

3) Read finished and then click the **[Back]** button.

4) Click **[Save]** button on the toolbar (you can also select "File-Save" in the main menu), type in the file name to save the file.
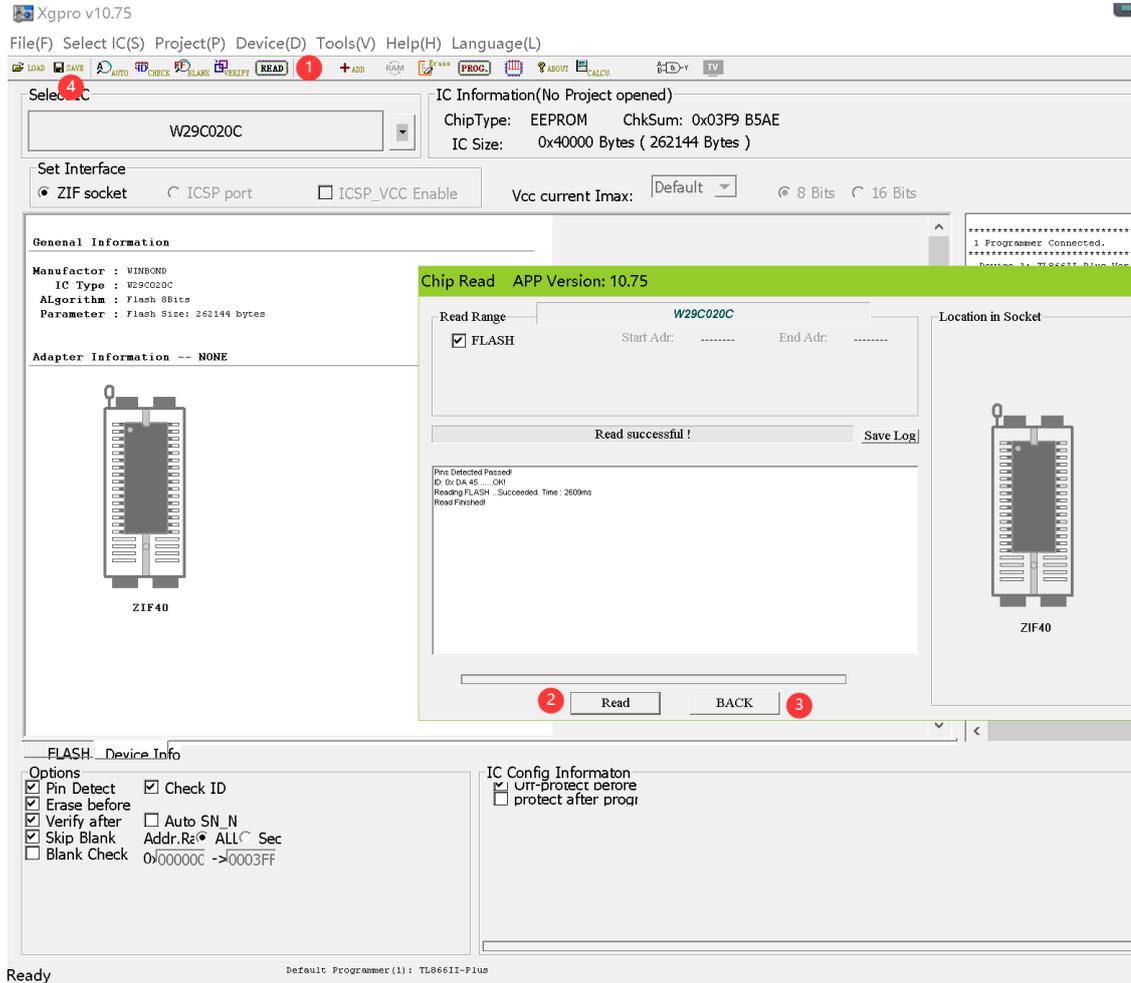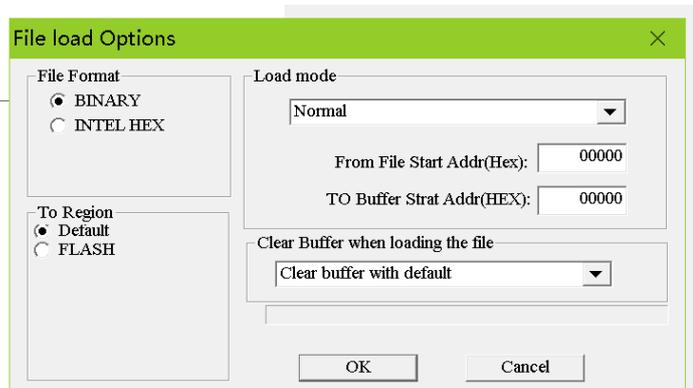
3.3 Programing a new chip

After completing the steps in section 3.2

1) Click **[Load]** button to bring up the Open File dialog box

2) Select the data file to be written (Test.BIN)

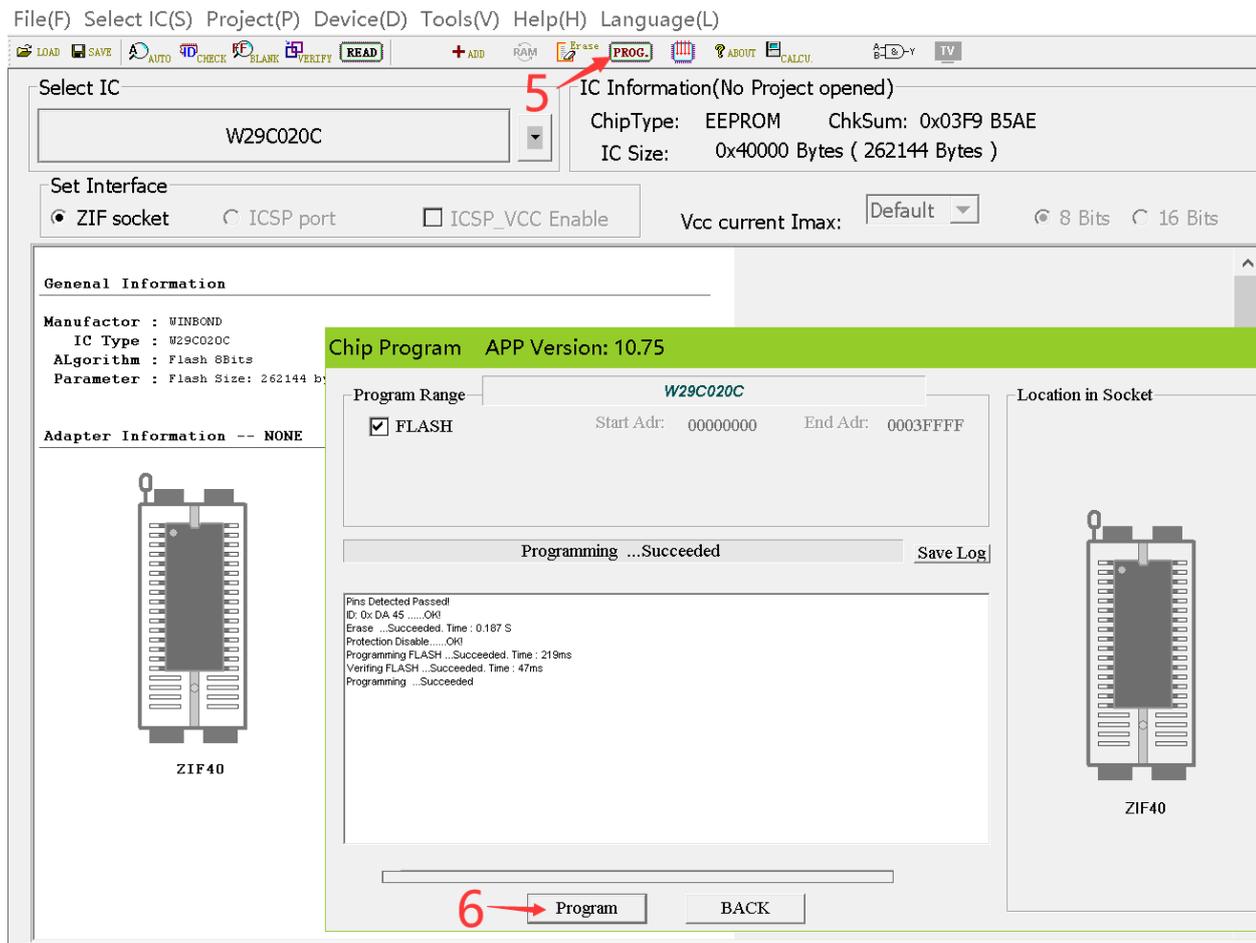3) Click **[Open]**, the file load options dialog box pops up
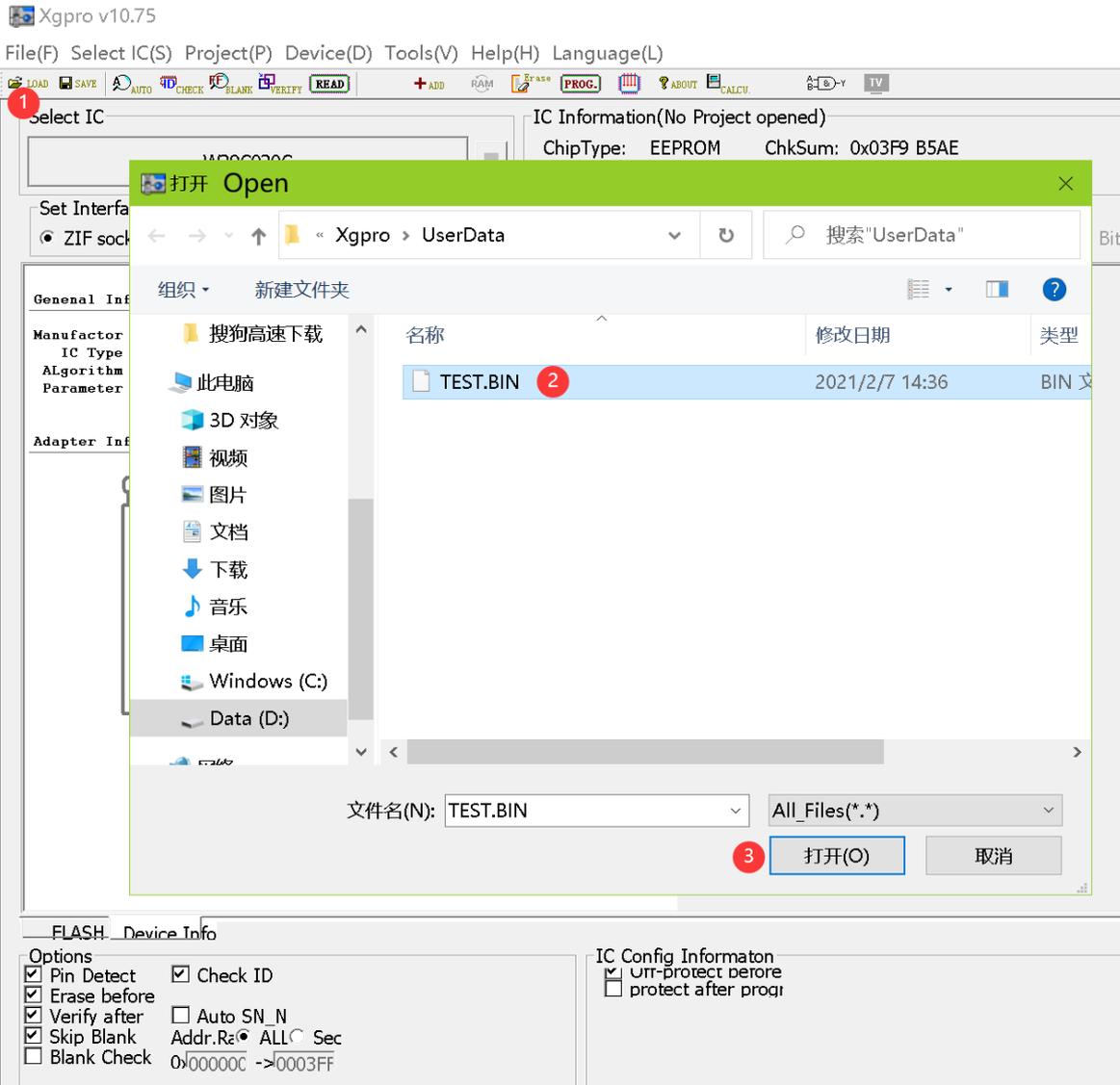
4) According to the default settings, click **[OK]**, at this time the file content has been loaded to the computer memory buffer.

5) Click **[PROG]** button, the chip programming dialog box pops up.

6) Click **[Program]** to get started to program the chip. If no error message appears, that means the programming succeeded.

Note:

  If you need to program more chips, replace the new chip and click [Program] button again to begin a new programming.

When programming in batch, you can press **[space]** instead of mouse click.

4. Programmer Function Details

4.1 Main Menu Summary

| Main Menu Summary	
File	Load File, Save, Find, Goto, Buffer operation, Exit
Select IC	Search and Select IC, 25 Flash Detect, Add IC by user
Project	Open Project, Save Project, Save Project As, Close Project, Project Attrib, Modify Password
Device	Read, Read ID, Verify chip, Verify, Program, Erase chip, Blank Check, Serials number, Testing, Multi Prog., NAND bad block check, logic IC test, TV/LCD Tools
Tools	Calculator, System self-check, Reflash firmware
Help	Help, About, Update On Line
Language	switch languages, English, Chinese, Russian, Polish, German, Spanish, Portuguese, Turkish, Czech
Right click menu Edit buffer	Find the corresponding hexadecimal value or ASCII character string in the buffer, data, copy, paste, fill in, partially save as TXT file

4.2 File Menu

- **Load file**:
  Loads a hex file or a binary file into the specified buffer, the GAL chip loads the file in JED format.
- **Save file**:
  Save the current buffer to HEX file in hex format or BIN file in binary format. For GAL chips, should be saved to JED format file.
- **Note:** For MICROCHIP microcontrollers, when loading HEX files, the HEX files generated by the MICROCHIP IDE development environment are loaded into all buffers. When saving, all buffers can be saved to the HEX file, and it is fully compatible with MICROCHIP IDE development environment.
- Save and open for Bin file, the default is Flash Memory + EEPROM Memory (if present)
  For MICROCHIP microcontroller, Bin saved configuration information (arranged by Microchip address)
- Can use shortcut keys or right-click in the buffer to invoke the menu functions.
- To get more details on Edit Function-->Edit Buffer
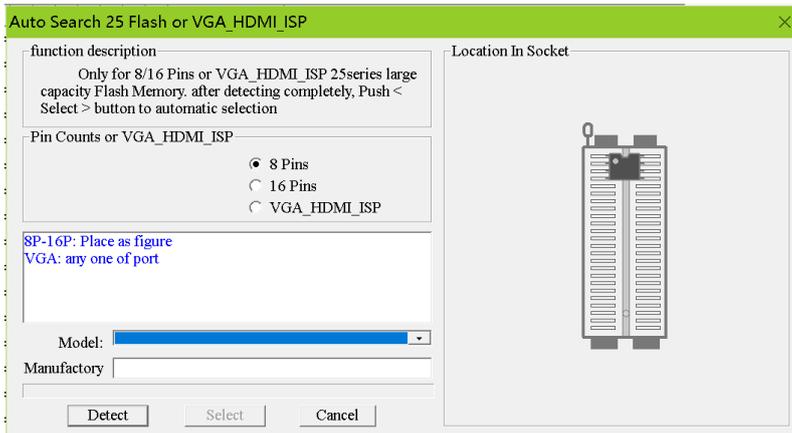
4.3 Select IC

The menu is shown below:
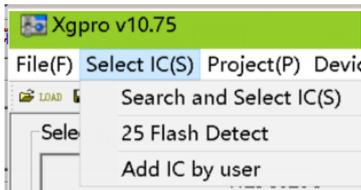
4.3.1 **Search and select IC**
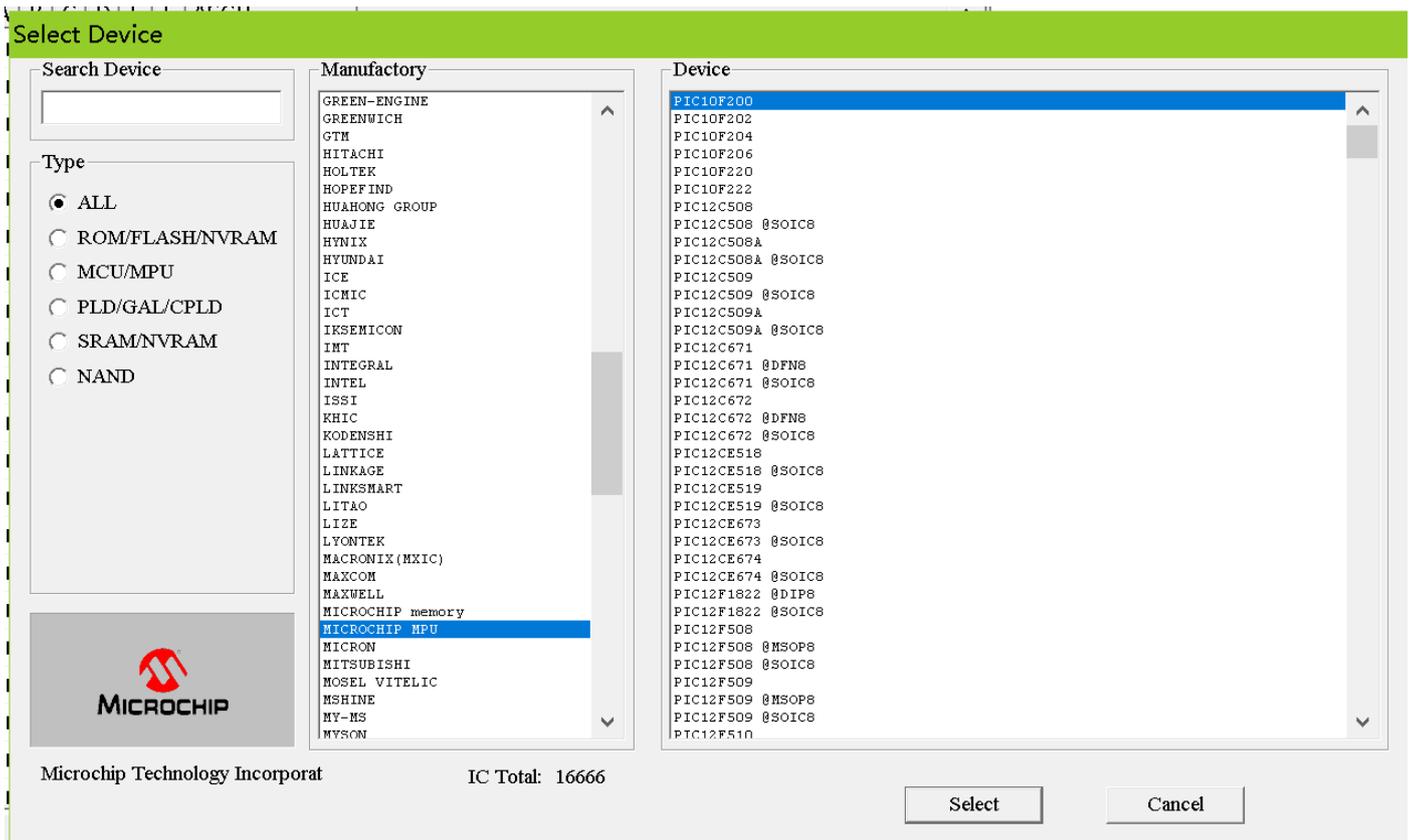
4.3.2 **Auto Detect 25 Flash**

25 Flash automatic detect for 8-pin, 16-pin and VGA HDMI. To detect 25 series SPI Nor/NAND Flash,

note: many chip IDs are the same and results have more than one, so you need to select an accurate model in the results.

4.4 Project Menu

The project file is created for better batch programming. It is a file that saves all current buffer data and working environment settings, including information about the current device, settings of all operation options, and settings related to auto serials number. It is convenient to use in daily mass production. The project can be set up with password protection, which can protect the security of the product program to a certain extent. The menu is shown below:
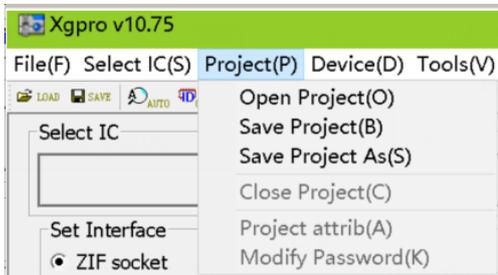
Save Project

After setting the required parameters of the chip in the application, select **[Project]** - **[Save-Project]**, that is, a project is created. And the project information is displayed in the information area of the window. Click **[Save Project]**, type in the project name and click OK, then the following project dialog box pops up.
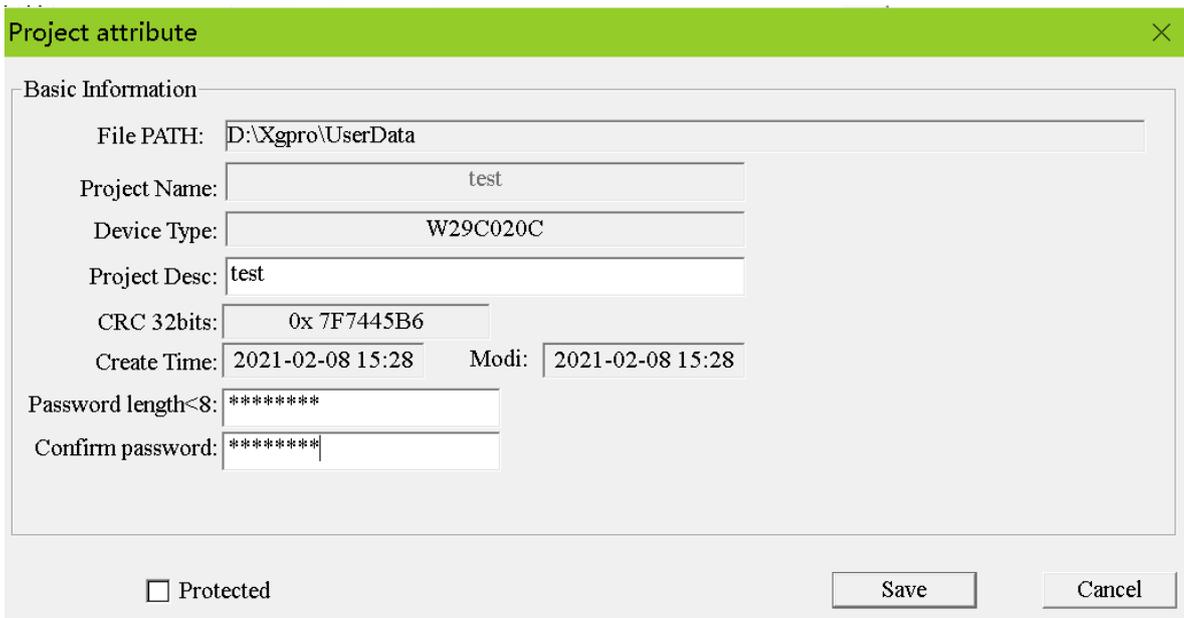
1) Password can be empty, that means the current project is not set with password protection.
2) If the project is set to protection mode, the password must be input. When the option "Protected" is checked, the relevant data of the project will not be changed. Some functions of the programmer are disabled.
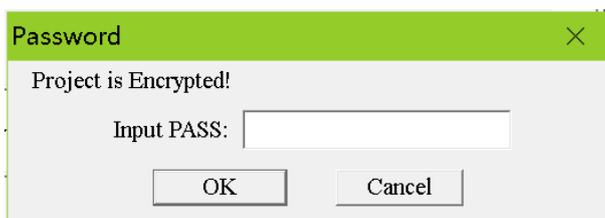
Open project

Open a current project, select [Project] - [Open Project], select the project name in the window and click OK. If the project has been set password, the following dialog box will pop up. (If no password is set, the project will be opened directly)

Save project As

Save the current project as another project name. Refer to the operation of [Save Project].

Close Project

Close the current project. Switch to normal programming mode, close project and clear data in the current buffer.
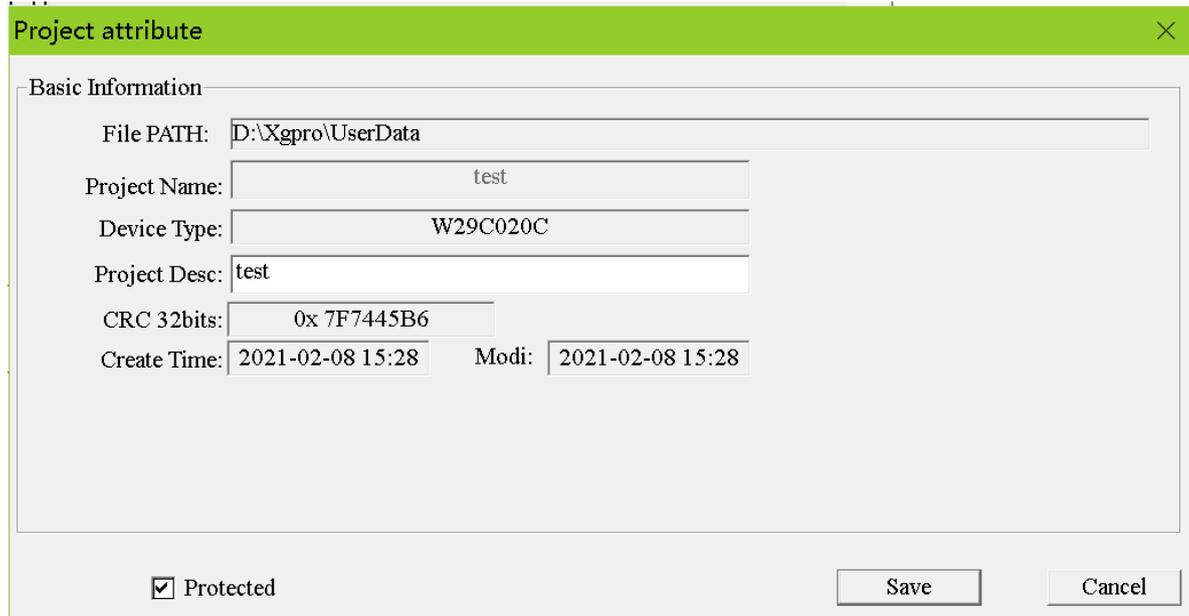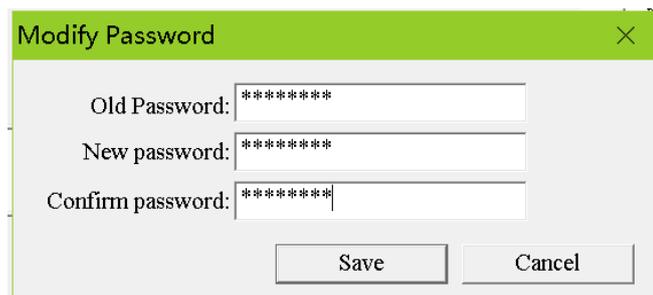
Project Attrib.

Select [Project] - [Project Attrib] to bring up the project attribute dialog box. The project attribute dialog box can change the project description and project protection mode. You need to input the project password when saving.

Project attribute   ✕

┌─ Basic Information ───┐
│   │
│   File PATH: D:\Xgpro\UserData   │
│   Project Name: test   │
│   Device Type: W29C020C   │
│   Project Desc: test   │
│   CRC 32bits: 0x 7F7445B6   │
│   Create Time: 2021-02-08 15:28   Modi: 2021-02-08 15:28   │
│   │
│   │
└──┘

   ☑ Protected   Save   Cancel

Modify Password

Reset project password

Modify Password   ✕

     Old Password: ********
    New password: ********
  Confirm password: ********

              Save   Cancel

┌──┐
│ Tips: │
│   1) Accurate engineering descriptions can distinguish from a large number of projects. Engineering models are generally │
│   used for mass production.   │
│   2) The project file has been encrypted to a certain extent, so the project password can be protected to prevent the project │
│   file from being copied. To a certain extent, it can protect the data.   │
│   3) When saving or opening project files, all contain 32-bit CRC check information, which avoid corrupting storage files or │
│   other uncertainties.   │
└──┘
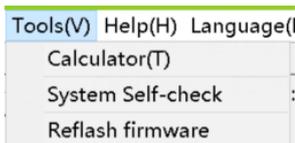
4.5 Device Menu

This menu function is the programming operation of the chips, please see section 4.10 Programmer operation function for details

The menu is shown below:
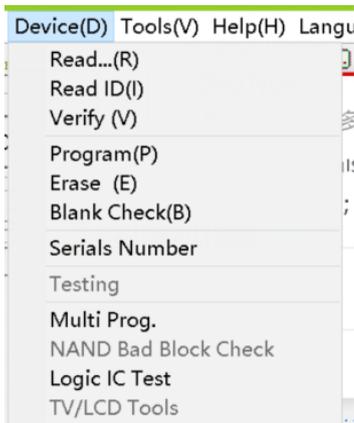
4.6 Tools Menu

The menu is shown below:

4.6.1 **Calculator**

This is a calculator that comes with Windows system and it is convenient for calculating various formats when switching to scientific mode.

4.6.2 **Self-check programmer hardware.**

Note: Before self-checking, please remove the IC and ICSP connecting wires on the ZIF socket, the programmer will apply VPP, VCC, GND to all pins for testing in self check. And test the internal short-circuit over current (apply VPP VCC and GND on each pin at the same time). If the IC is on the ZIF socket, may damage the IC. After the test is completed, the test condition of each power supply is displayed in the dialog box.

Note: The interface may vary slightly depending on the programmer model
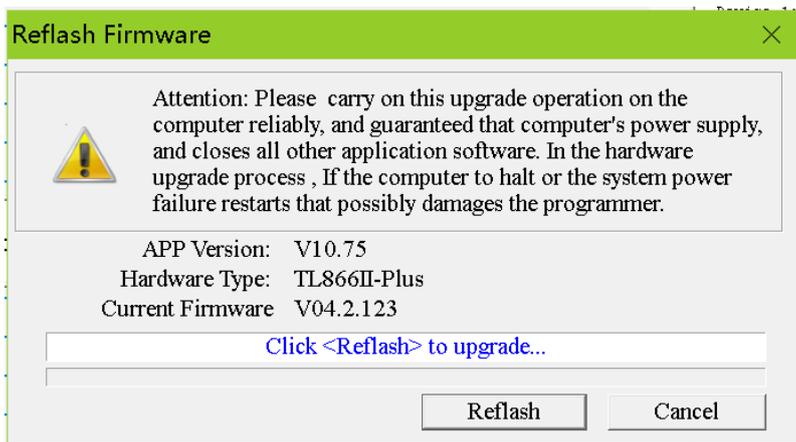
4.6.3 Reflash firmware

Reflash the firmware of the programmer hardware. The application is upgraded to a new version.

If the program in FLASH needs to be upgraded and reflashed, the application will prompt for a firmware reflash. If no refresh is needed, the reflash function is disabled.

Click <Tools-Reflash firmware> to bring up the following dialog box:

Reflash Firmware   ✕

> ⚠️ Attention: Please carry on this upgrade operation on the
> computer reliably, and guaranteed that computer's power supply,
> and closes all other application software. In the hardware
> upgrade process , If the computer to halt or the system power
> failure restarts that possibly damages the programmer.

APP Version: V10.75
Hardware Type: TL866II-Plus
Current Firmware V04.2.123

Click <Reflash> to upgrade...

[Reflash]   [Cancel]

Click <Reflash> button to start, the total reflashing time is about 15 seconds. Click <Cancel> to quit reflashing.

Note:

1. Please do not use an external USB hub when reflashing. Please directly connect the programmer to the USB port of your

computer. Using an external hub may not allow the reset upgrade.

2. Close other programs and make sure the power supply of computer is reliable!

4.7 Edit buffer

The programmer software can modify the contents in the buffer, the common functions as below:

- Define Block
- Fill Block
- Copy
- Block Save As
- Clear buffer
- Find
- Find Next
- GOTO address

4.7.1 **Define Block**

There are two ways to define block:

The first way is: Press the left mouse button in the buffer, move the mouse to select a piece of data and then press the right mouse

button, the following menu appears as below:
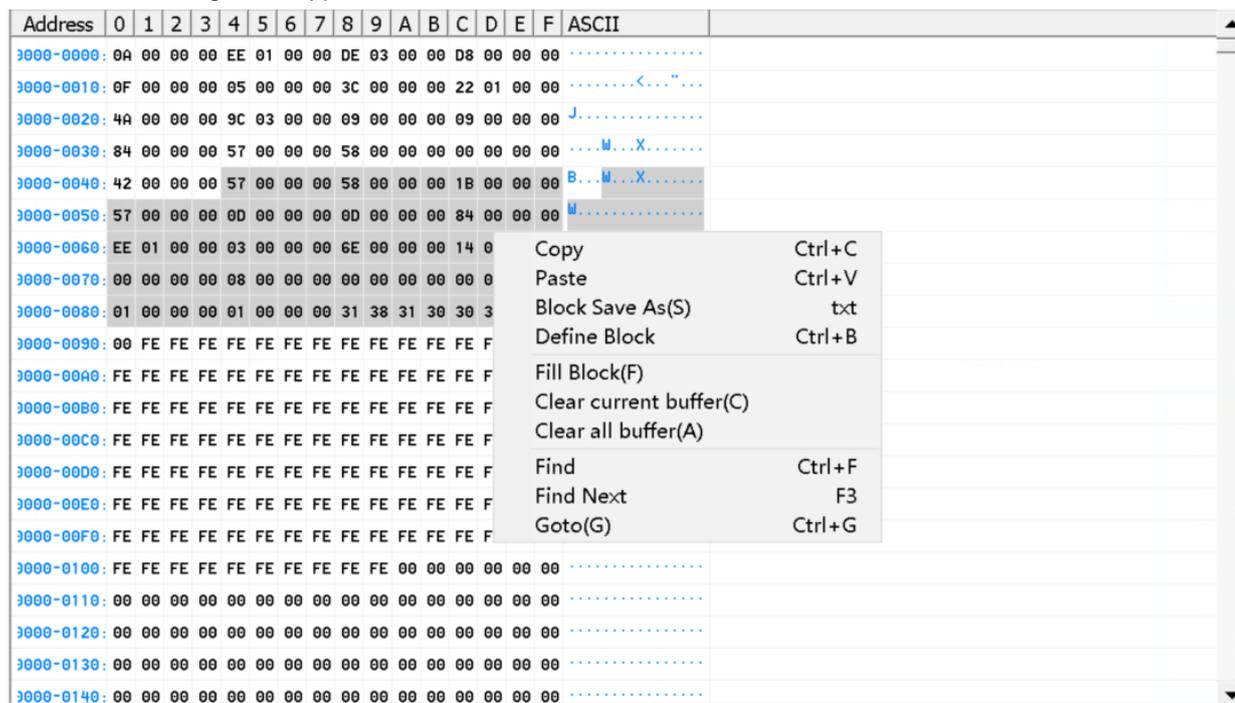
The second way is: in the main menu, click [File] button to select [Define Block], or directly press the shortcut key "CTRL + B" to bring up the following dialog box, type in the start address and end address, and press OK to complete the "Define Block".
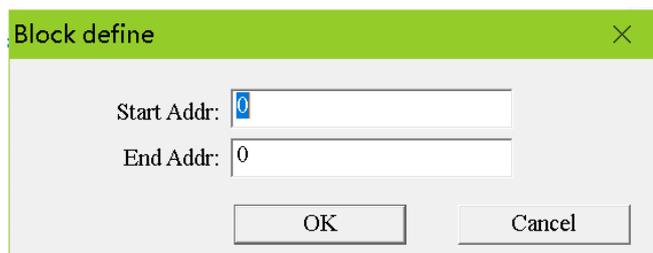
Subsequent block operations can be done on the defined data blocks.

Copy block, Fill block, Block save as TXT file

4.7.2 **Fill block**

"Fill block" is to replace all the data of a continuous address in the program code buffer with a fixed value. Right mouse click to bring up [right click menu], and select [Fill Block], the following dialog box pops up:

A.   Buffer Region, the default value is the buffer displayed in the current window.

B.   The start address and end address of the block to be filled. The default value is from the start to the end of the defined
     data block if a block is defined. If no block is defined, it is the full region of the current buffer.

C.   Fill value, the maximum allowed is 4 bytes. If the fill format is random, the value is irrelevant.

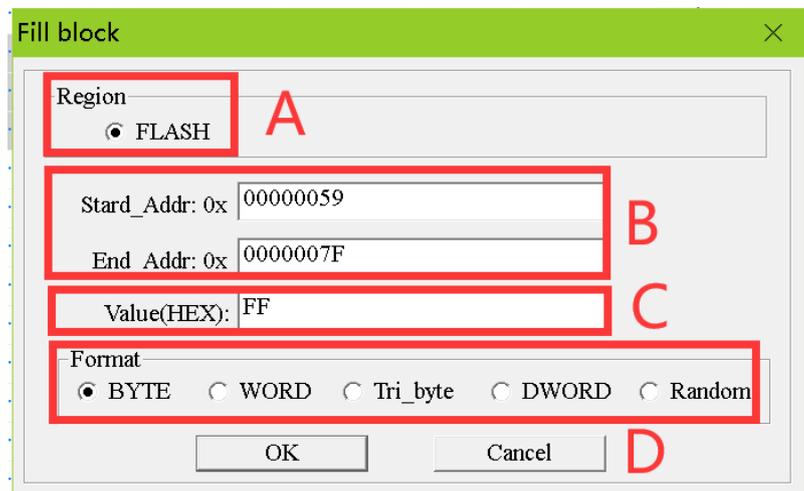D.   Filling format, default format: If the chip is programmed by 8-bit mode, the default is single-byte mode. If the chip is
     programmed by 16-bit mode, it is double-byte mode, you can change the filling format you want.

4.7.3 **Copy**

After defining block, right mouse click to bring up [right click menu] and select [Copy], then the content of the block has been copied
to the clipboard of WINDOWS. You can also directly press shortcut key "CTRL+C" to finish copying.

Copy data from area A to area B as below:
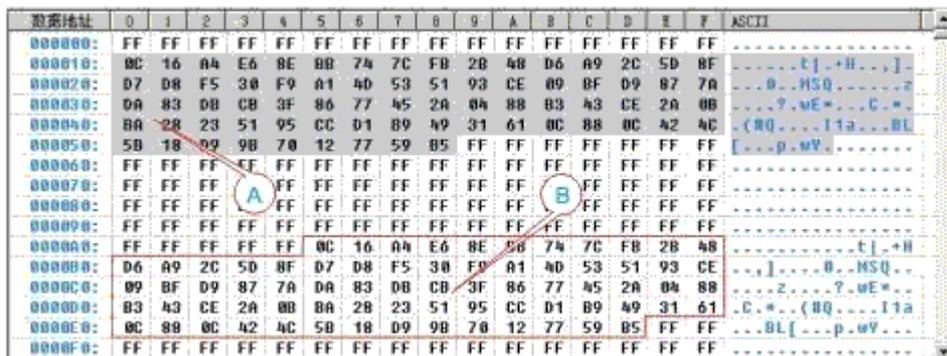
Note: In order to enhance the speed of pasting, the data format of clipboard is a special binary format for this programmer. So other
WINDOW applications cannot directly use the data copied to the clipboard by this program. If you want to copy the data block to
other applications, please use the [Block Save As] TXT file function.

Block Save As

You can output the defined data block as a TXT text file,   that is convenient for designers to apply when programming. The text
format is automatically arranged in 16-bit or 8-bit way. Operation method: After defining block, click the right mouse button to bring
up the [right click menu] and select [Save Block As].

4.7.4 **Clear Buffer**

Clear Current Buffer: clear the buffer displayed in the current window of the application.

Clear all buffers: clear all buffers of the chip with default values.

Find, Find Next

Function: Find the HEX or ASCII string in the current buffer.

Operation: Press shortcut key "CTRL+F" or click the [Find] in the right-click menu, the following dialog box pops up:
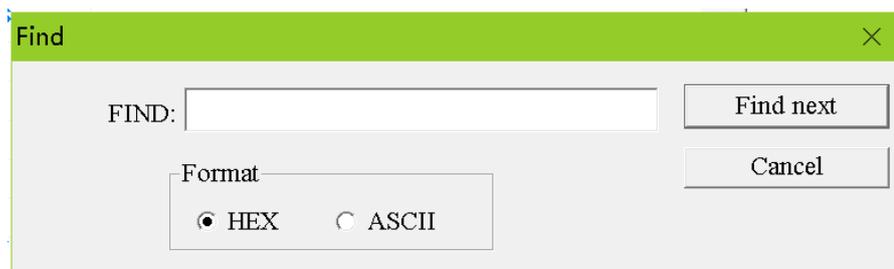
Type in the HEX value or ASCII string you want to find, and finally click the "Find" button in the dialog box. When there is a
matching hexadecimal data or string in the program code buffer, the cursor will be located to the address of the first matching
content and press F3 to continue finding the next one until the buffer is completely searched.

4.7.5 **Goto address**

Function: Move the cursor to the address specified in the current buffer.

Operation: Press shortcut key "CTRL+G", or press the right mouse button to bring up the menu and select [Goto], then the following dialog box pops up:

Goto	×
Goto Address 0x:	`00 00000000`
	OK Cancel
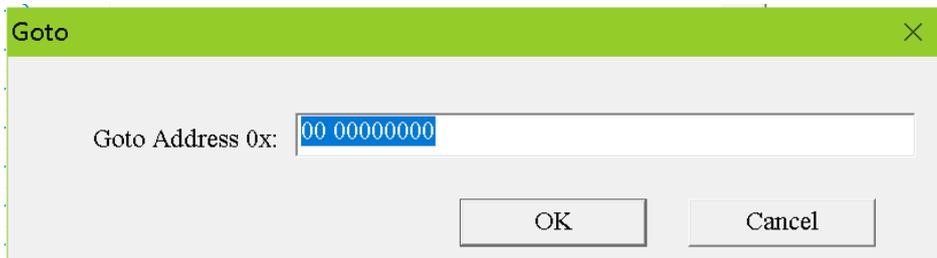
Type in the address, click OK button and the cursor will move to the address.

4.8 Options

Options of programming setting are shown below. You can set the options

Use default setting for mass production , don't need to modify.
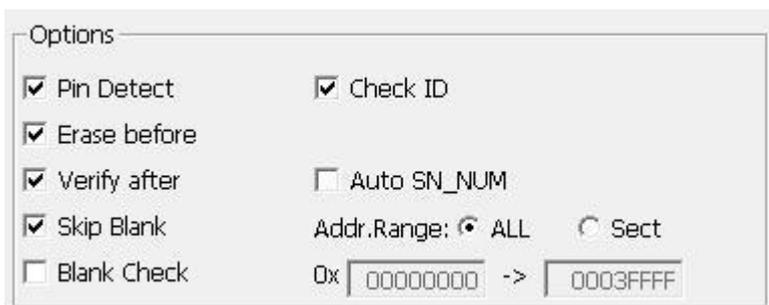
4.8.1 **Pin Detect**

Before operating, the software automatically checks if the chip pins are in good contact. It can avoid programming or reading errors caused by poor pin contact in most cases.

Special case of pin poor contact
Because of pin severe oxidation, some individual pins may not be completely disconnected, (for example, it is equivalent to a contact of 3-10K resistance). In this case it is not possible to detect if the contact is good, and errors may occur during programming. So for old chips, the pins should be handled carefully.

For very few chips, the pin detect may fail. You can temporarily uncheck this option "Pin Detect" while ensuring that the pin contact is good. In that case, no pin detection will be performed before reading or writing.

4.8.2 **Erase before (programming)**

Erase the entire contents of the chip before programming

Some chips do not have erase function, this option will be disabled, for example: most 27Cxxx chips are one time OTP ROM or UVROM. These chips are not erasable or do not have the electrical erase function.

For 24 series, 25 series, EEPROM 93 series chips, this function is also disabled when there is no special erase command. That means these chips do not need to be erased and can be rewritten directly. (If you must erase the data of the chip, you can write 0xFF into the entire memory of the chip)

4.8.3 **Verify after (programming)**

4.8.4 **Skip Blank**

The write operation will be skipped for FF data blocks, that can greatly improve the programming efficiency. When this option is checked, the verify operation will also be skipped.

4.8.5 **Check ID**

Many chips have an internal identification mark, the content of which is generally composed of two or more bytes: generally, the first byte is the Manufacture ID, followed by the chip type or chip capacity. Different chips have different IDs, this option can prevent incorrect chips put on. This option is to check the chip ID before reading or writing. If it is correct, continue; if incorrect, stop. This function is optional. The option is enabled by default.

> For chips of the same type, they have different manufacturers, but may be programmed in the same way. Therefore, for chips that are not in the support list, you can select other manufacturers of similar model chips with the same capacity for programming. Because the ID is different, before programming, just uncheck this option [Check ID]).

4.8.6 **Address range**

You can set the address range of the programming chip, that can improve the efficiency when mass production.
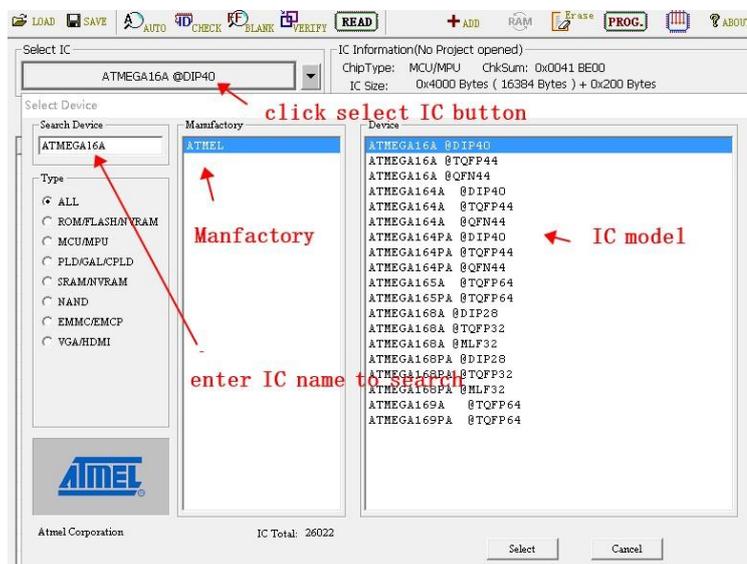
4.8.7 **Blank Check (before programming)**

Except for the 27 series UVROM, generally it is not necessary to check this option.

4.8.8 **Auto SN_N**

Refer to section 4.10.7 Setting the Serials number

4.9 Select IC

Click <Select IC> button in the upper left corner of the software interface

When selecting, the package of chip must be right. Sometimes, different packages of chips have different pin definitions. For the same type of IC, (none Package is DIP package chips.)

if the specific package is not listed, just check the IC data sheet and directly connect the corresponding pins to program.

4.10 Programmer operation function

4.10.1 **Read**

Read means reading out the data from the chip into the buffer. The reading window may be different from different models of chips.

for example, the chip ATMEGA16L, click [Read] button in the toolbar and the following dialog box pops up:
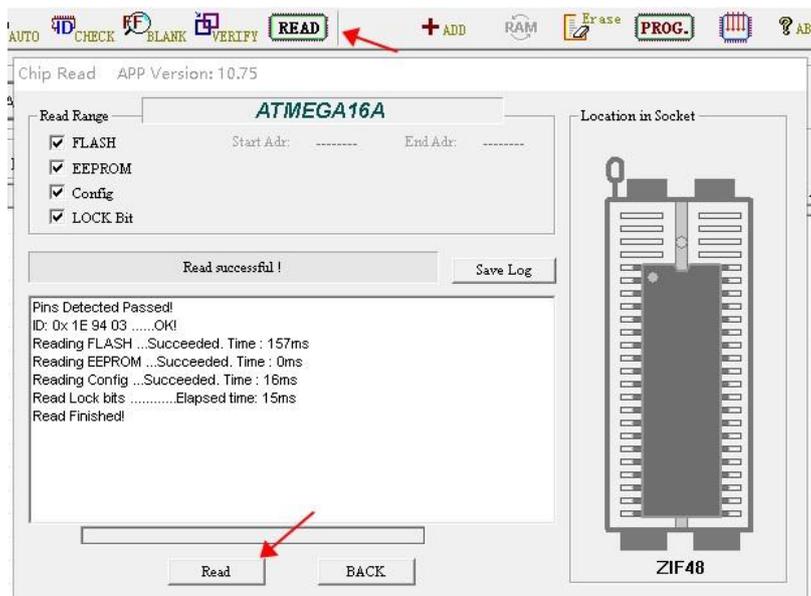
First of all, set the [Read Range], which includes [FLASH] [EEPROM] [Config.] [LOCK Bit].   do reading operation on the entire chip. Click [Read] button to start reading the data of chip, note: if the chip is encrypted, the read data is invalid.

(for   ATMEGA,   [LOCK Bit bytes] in config,   the datas are readable whether be encrypted or not.) After reading, the reading time of the individual buffers is displayed in the information list box.

Note: generally, the read operation will not report an error. If you want to verify whether the read data is correct, you can use the [Verify] function. Please see the verification instruction for more details. The chip ID value will be checked by default before operation.

4.10.2 **Program**

Load the data file to program chip

Click **[PROG.]** button in the toolbar, the chip program dialog box pops up as below:

Take W29C020 as an example,

Normal programming is shown below:

Chip Program APP Version: 10.75

Program Range W29C020

☑ FLASH Start Adr: 00000000 End Adr: 0003FFFF

Location in Socket

Programming ...Succeeded Save Log

```
Pins Detected Passed!
ID: 0x DA 45 ......OK!
Erase ...Succeeded. Time : 0.203 S
Protection Disable......OK!
Programming FLASH ...Succeeded. Time : 187ms
Verifing FLASH ...Succeeded. Time : 47ms
Programming ...Succeeded
```

ZIF40

Program   BACK
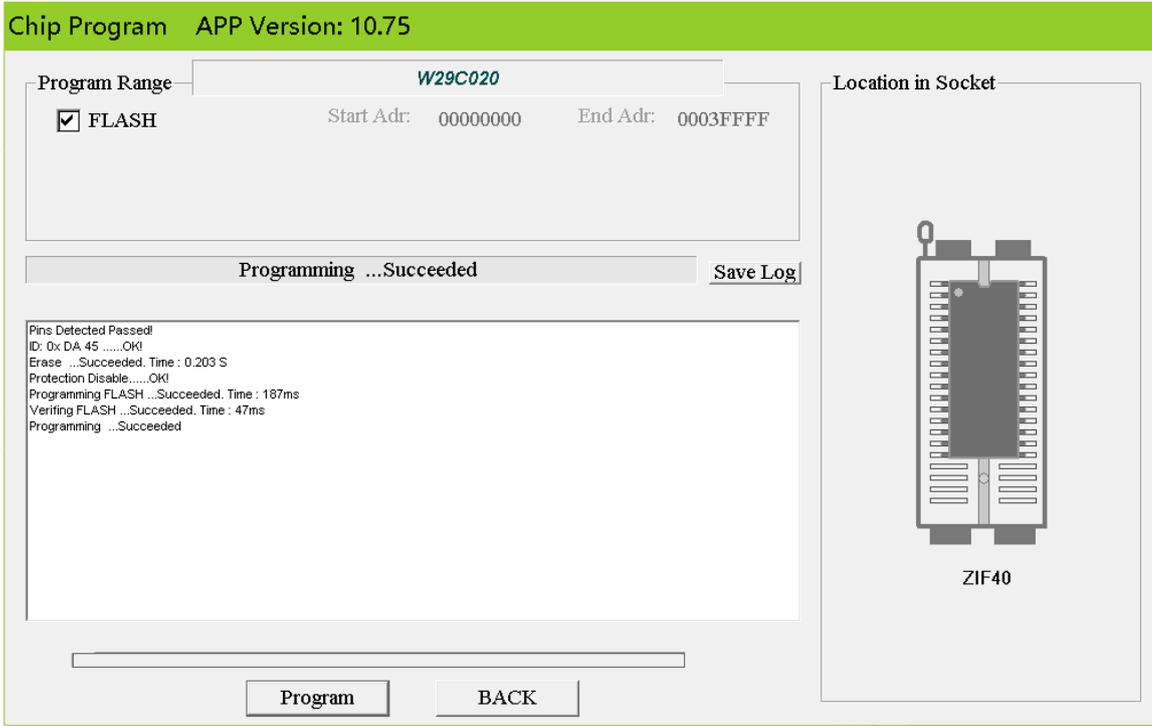
- For most of the chips, the programmer will automatically verify during the programming process. Will stop immediately if an error is found.
- If the option [Auto SN_N] is checked, the chip number in the buffer will be automatically incremented according to the setting after programming.
- To ensure the correctness of programming data, it is advisable to check the option **[Verify after]**.

4.10.3 **Erase**

For the electrically erasable chip, click [Erase] button in the toolbar to bring up the following dialog box:

Chip Erase APP Version: 10.75

Erase Range

ODEL: W29C020

Location in Socket

Erase ...Succeeded Save Log

```
Pins Detected Passed!
ID: 0x DA 45 ......OK!
Erase ...Succeeded. Time : 0.203 S
Erase ...Succeeded
```

ZIF40

Erase   BACK
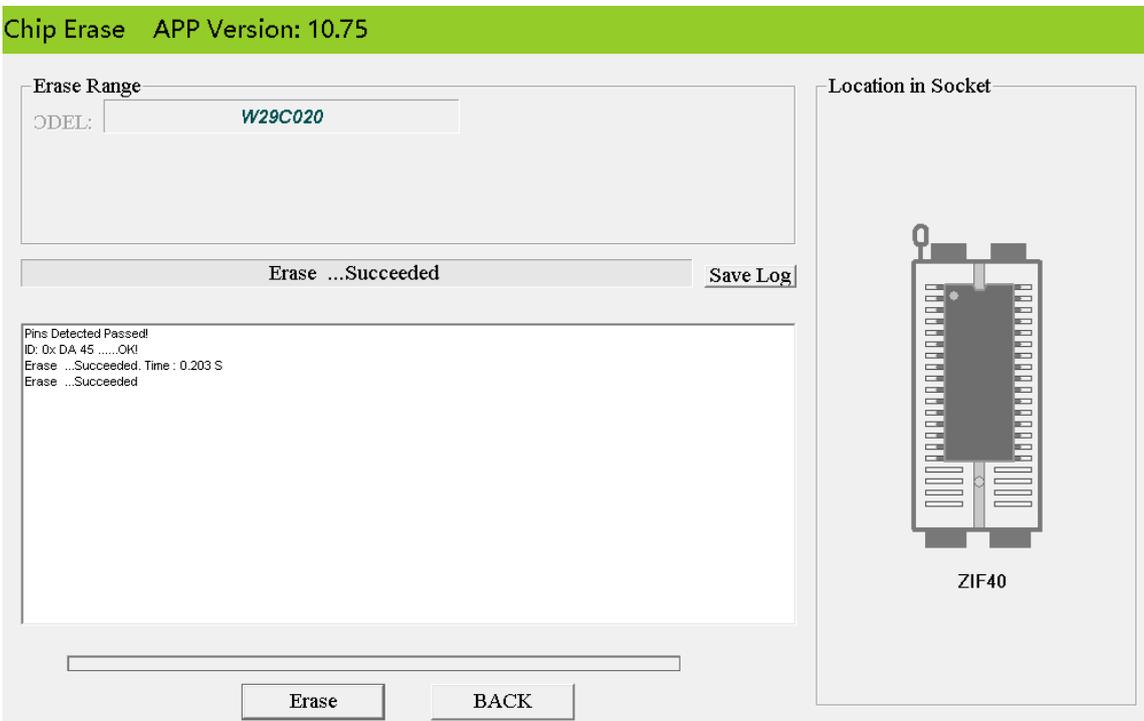
The option [Check ID] is checked by default. For few parts of microcontrollers, some chips while it be encrypted and the ID cannot be read out. In that case, please uncheck the option [Check ID], otherwise ID Error will be displayed. Click [Erase] button in the dialog box, after the erase succeeded, the erasing time is displayed.

4.10.4 **Blank Check**

Click [Check Blank] button in the dialog box to start the check blank operation. When it is finished, the elapsed time is displayed. You can select the region of the chip before checking, all region is by default. If the chip is found to be non-empty during the blank check, will stop operation and the address and stored value of the non-empty area will be displayed.

4.10.5 **Verify**

You can select the region of the chip before verifying, all region is by default. Click [Verify] button in the dialog box to start the operation. If all the data are the same, the elapsed time is displayed when verify finished.

If the chip datas are found to be different during verifying, the following dialog box will pop up:

Click [Continue]: The same dialog box will pop up again if there are still differences.

Click [Verify to all]: It will compare all the contents of the chip, and prompt will pop up if differences are found during the comparison process, until it is finished. It will prompt how many differences and show the different cells in red font in the buffer (except NAND) , as shown below:

Click [BACK]: It will exit directly without the later verifying.

4.10.6 **IC Config.**

Click [Config.] button, the interface is as below (select IC model: PIC18F4455):

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
0X300000:	–	–	☑ USBDIV	☑ CPUDIV1	☑ CPUDIV0	☑ PLLDIV2	☑ PLLDIV1	☑ PLLDIV0
0X300001:	☑ IESO	☑ FCMEN	–	–	☑ FOSC3	☐ FOSC2	☑ FOSC1	☐ FOSC0
0X300002:	–	–	☑ VREGEN	☐ BORV1	☐ BORV0	☐ BOREN1	☐ BOREN0	☐ /PWRTEN
0X300003:	–	–	–	☐ WDTPS3	☐ WDTPS2	☐ WDTPS1	☐ WDTPS0	☐ WDTEN
0X300004:	–	–	–	–	–	–	–	–
0X300005:	☐ MCLRE	–	–	–	–	☑ LPT1OSC	☐ PBADEN	☐ CCP2MX
0X300006:	☐ /DEBUG	☑ XINST	☑ ICPRT	–	–	☐ LVP	–	☐ STVERN
0X300007:	–	–	–	–	–	–	–	–
0X300008:	–	–	–	–	–	☐ CP2	☐ CP1	☐ CP0
0X300009:	☐ CPD	☐ CPB	–	–	–	–	–	–
0X30000A:	–	–	–	–	–	☐ WRT2	☐ WRT1	☐ WRT0
0X30000B:	☐ WRTD	☐ WRTB	☐ WRTC	–	–	–	–	–
0X30000C:	–	–	–	–	–	☐ EBTR2	☐ EBTR1	☐ EBTR0
0X30000D:	–	☐ EBTRB	–	–	–	–	–	–

Note: Bits CHECKED is Programmed as '0'

| FLASH | EEPROM | Config | Device.Info |

Options
☑ Pin Detect   ☑ Check ID
☑ Erase before
☑ Verify after   ☐ Auto SN_NUM
☑ Skip Blank   Addr.Range: ⦿ ALL   ○ Sect
☐ Blank Check   0x [00000000] -> [00005FFF]

Others
CONFIG:(0x300000-0x30000D)   USERID:(0x200000-0x200007)
-----------------------------   ----------------------------
00 05 1F 1F 00 83 85   FF FF FF FF FF FF FF FF
00 0F C0 0F E0 0F 40
Note:Checked = 0   USERID: [FFFFFFF]

Hardware Interface Ver: No Device!

> The value of the config information in the red circle corresponds to the config interface: click on the config interface, the value in the config information will change automatically. The config information varies from chip to chip, so please check the chip's data sheet at first, and configure IC correctly according to your specific needs.IC config, need to understand the role of each bit in the configuration bits

4.10.7 **Serials number**

The chip number setting function is to automatically add a number to a certain memory area of the chip during programming, that can easily program the chip at the factory. You can add time or unique serial numbers and so on, also can use a variety of numbering algorithms. Click <Device>--<Serials Number> in main menu to bring up the following dialog box:

Firstly, select the automatic numbering algorithm, there are four modes in the dialog box. And then introduce the every mode separately as below:

4.10.7.1 Default INC. (Default Incremental Algorithm)

The setting interface of this algorithm is shown above. The function of this algorithm is to put a number of set length at the place where the set storage area of the start address, and the number increases according to step length at a time. After setting the parameters, you can click <Test> button to test and see the change rule under various parameter settings.

The initial value and step length of the number can be set in the dialog box.

After setting the parameters and testing them well, set the record file. If it is empty, the record file is invalid and the data will not be recorded into the file when programming. If the record file is set, the number you set will be added to the end of the record file after each programming.

Finally, check the option <Enable Auto.SN> and click<Save> button.

4.10.7.2 Time Mode (algorithm)

The interface of the Time Mode dialog box is as below:

at 0x2000, a time data is put in
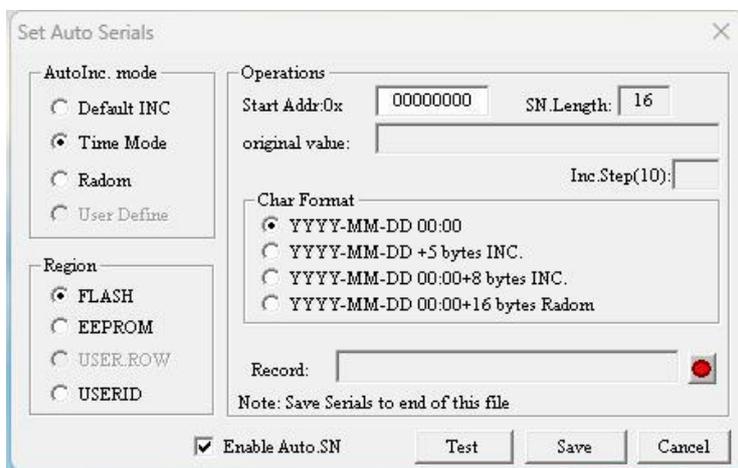
There are four types of time encoding formats, which are as below:

    (1) Date + time   Total length of number: 16bytes

        The date and time are added at the specified location during programming, the total length is 16bytes.

    (2) Date + 5 bytes increment   Total length of number: 16 bytes

        The increment part can set the initial value and step length.

    (3) Date + time + 8 bytes increment   Total number length: 24 bytes

        The increment part can set the initial value and step length.

    (4) Date + time + 16 bytes of random numbering   Total length of numbering: 32 bytes

16 bytes are randomly generated bytes, which can generate a unique serial number of product. In this way, the record file must be set so that the generated code is recorded in the file. Otherwise, you don't know what the product number is.

The next steps are the same as above <Default Incremental Algorithm>

4.10.7.3 Random (Random number algorithm)

The algorithm generates a random number of the specified length, as below: 16 bytes of random data are placed at the start address 0x2000
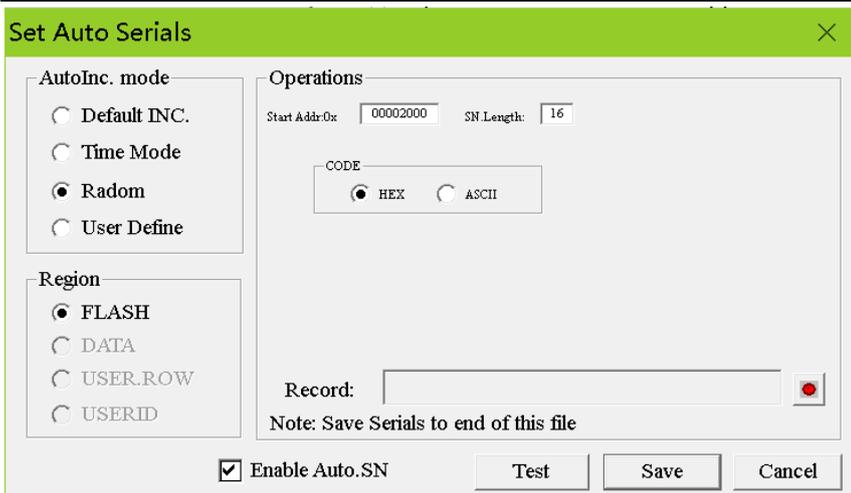
4.10.7.4 User Define algorithm

The user-defined algorithm is very flexible, you can set any numbering algorithm, see the catalog for details about how to use it. \\Serialnumber\source_dll in the VC++ source program for detailed instruction. The interface is as below:

After making the DLL file, you just need to copy it to the application directory of ... \\Serialnumber subdirectory. When setting, choose the algorithm file you made in the user algorithm file list. The following steps are the same as the other algorithms. Important Notes:
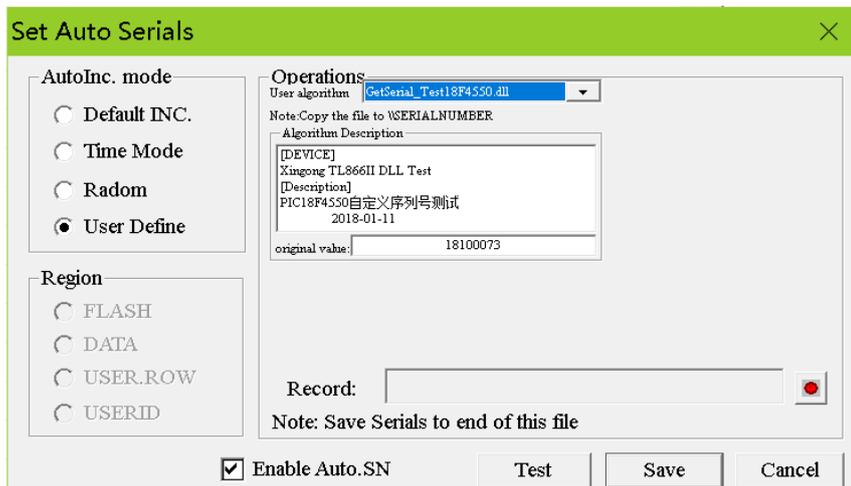
1) After setting the above content, the programmer will not automatically add the number you set to the chip when programming. If need to automatically add the number, you must check the option [Auto SN_N] in the main interface.

2) In the main interface, the condition that you can check the option [Auto SN_N] is that you must firstly set the numbering algorithm and check the option [Enable Auto.SN] in the set auto serials dialog box.

3) After checking the option [Auto SN_N] in the main interface, if you reset the numbering algorithm, the option [Auto SN_N] must be re-checked in the main interface.

4) If IC model is changed, you must reset the numbering algorithm.

5) In the engineering mode, the algorithm is saved in the project file. You can directly open the project file and automatically transfer to the automatic numbering algorithm. Therefore, it is generally used in the project mode for mass production, it is simple and convenient.

4.11 ICSP Programming

For chips that can be in-circuit programmed serially, the programmer can program the chips on the target board through the unique ICSP interface. Before programming, check the option [ICSP port] in the main interface, at this time the option [ICSP_VCC_Enable] becomes available. The option [ICSP_VCC_Enable] is checked by default, that means the programmer provides VCC power to the target board. Please note: a maximum current that VCC power supply of programmer can provide is 120ma. If the target board needs a larger power supply, please use the power supply from target board itself.

If you do not need power supply from programmer, you can uncheck the option [ICSP_VCC_Enable]. In this case, there will be no VCC power on VCC pins when programming. It is shown as below:
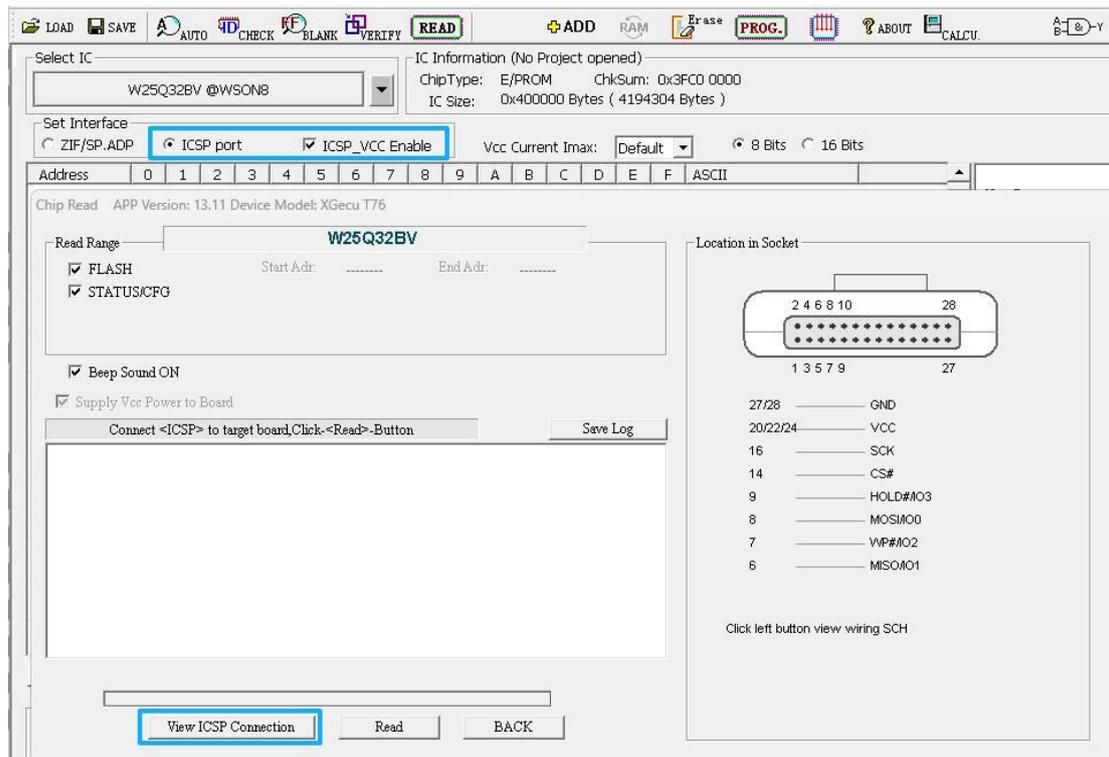
Next, Read, Erase, Verify, and Program operations will work through the ICSP port.

Click [Read] button in the toolbar, the following dialog box pops up as below (Please note: the dialog box for read, erase, verify and other operations is the same as write operation):

On the right side, the ICSP port wiring schematic is displayed. For more details, you can click [View ICSP Connection] button.
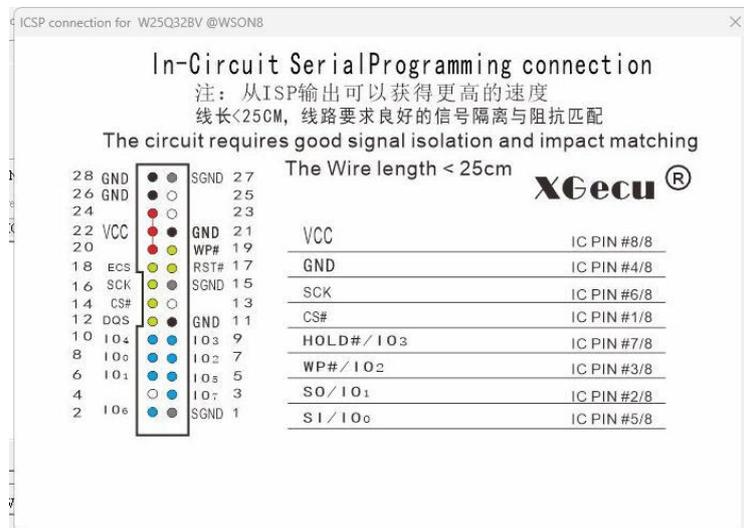
4.12 Multi Programming

The software support simultaneously working with 4 Programmers on each computer. For USB expansion, please use a high-quality USB hub with power supply. Multi programming interface is as below. The automatic numbering function can be used normally when multi programming.
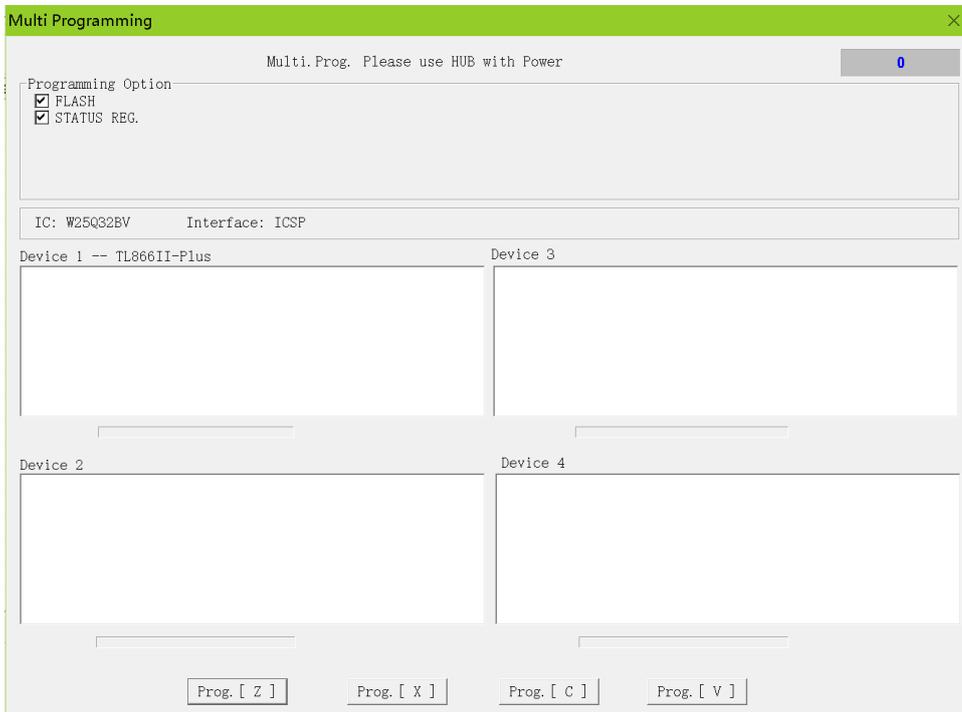
You can use the shortcut keys [Z] [X] [C] [V] to start the corresponding programmer when multi programming.

4.13 RAM Testing

After selecting the corresponding model of RAM chip, click [Device] - [Testing] button in the toolbar, the following dialog box pops up.
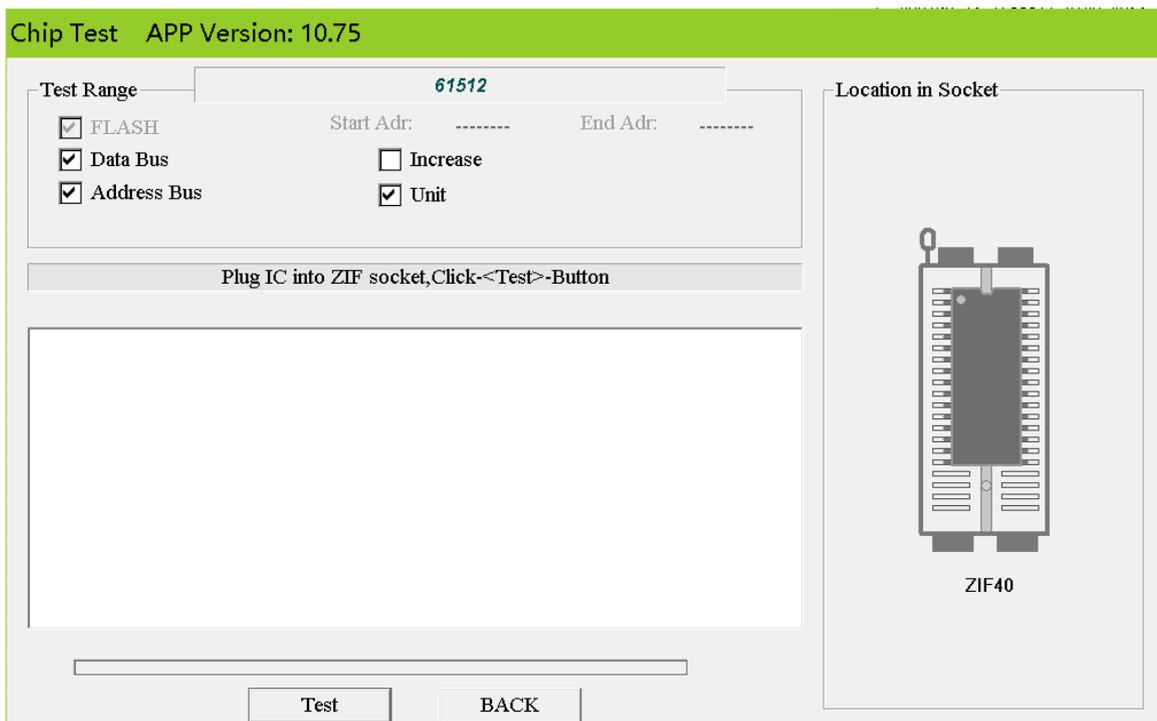
There are four methods of RAM testing. Through these four methods, you can basically test whether the RAM cell function is normal. General programmer can only read and write the RAM cell and cannot test whether the RAM is good or bad.

For example, if two adjacent address lines are short-circuited, it is impossible to check out the problem by simply reading and writing data from the RAM cell. So only one way to test is not a complete test. Plug chip and click [Testing] button to test RAM.

5. EMMC/EMCP programming

eMMC (Embedded Multi Media Card) is a standard specification for embedded memory developed by the MMC Association, primarily for products such as cell phones or tablet computers. By integrating a controller into the package, providing a standard interface and managing flash memory, eMMC allows handset manufacturers to focus on other parts of product development and shorten the time to market.

EMCP is a higher-level memory device than EMMC, which combines EMMC and LPDDR into one package, reducing the volume and design for circuit connection.

EMMC/EMCP are available in various BGA packages, general: BGA153/BGA169/BGA162/BGA221/BGA100/BGA254/BGA529

5.1 EMMC Memory

EMMC flash memory can have up to 8 storage areas, named as BOOT1/BOOT2/RPMB/GPP1/GPP2/GPP3/GPP4/USER. Among them, GPP1-4 are user-defined partitions, which are not configured at the factory.

- BOOT1/BOOT2 is generally used as system boot area and boot data backup partition
- RPMB is a special storage partition, data can be read always, but writing requires a 32-byte authentication key. It must be written through a complex HMAC_SHA256 authentication algorithm, in order to prevent hacking software from illegally modifying the data.
- GPP1-4 are user-definable hardware partitions, which are used in the same way as the USER partition
- USER partition, the largest user data partition
- Partition write protection, GPP1-4 and USER partitions can be written protection in groups, and the write protection modes are temporary write protection, permanent write protection, and power-on write protection.   After setting permanent write protection, the protected WPG group can't be erased or modified data permanently.

- EMMC password protection function, if the password is set, the password must be unlocked before accessing the USER area, otherwise it cannot access.

- ECSD and CSD are configuration registers. The various configurable parameters of EMMC are set in [Device config]. Great part of the registers in ECSD and CSD are read-only registers.

- CID is identification mark of the chip at the factory, which can only be read and cannot be modified.

5.2 EMMC function supported

5.2.1 **Analysis of the original chip**

This function allows you to get a full overview of the original chip and do operations the way you want.

The content of the original chip is analyzed and the following results are displayed:

- Display chip manufacturer information and life cycle

- Visually display the usage of up to 8 partitions of the chip

- Each partition Displays whether it is write-protected, readable, rewritable or erasable

- Displays whether the whole chip is temporarily write protected or permanently write protected

- Display whether the chip is password protected or not

- Display whether the chip uses 32-bit authentication key written to the RPMB.

- Display other important parameters of ECSD CSD CID

5.2.2 **Read/Program function**

All functions can be done with one key

- Read CID/CSD/ECSD.

- Clear temporary write protection before programming

- Clear temporary write protection group before programming

- Erase chip before programming

- Blank check before programming

- Program 32-bit authentication key

- Check the chip password (T48 no support)

- New partitioning of the chip (partition Config)

- Read and write BOOT1

- Read and write BOOT2

- Read and write RPMB

- Read and write GPP1

- Read and write GPP2

- Read and write GPP3

- Read and write GPP4

- Read and write user data (User)

- Setting temporary write protection group

- Setting permanent write protection group

- Setting chip password (or reset password) (T48 no support)

- Program ECSD

- Program CSD

- Automatic verification of all the above operations after programming

5.2.3 **ISP in-circuit programming**

- Ultra-stable ISP in-circuit programming with line length up to 40CM, working frequency 50MHZ, reliable work

- ISP can also use 4-bit wide mode, operation speed increased to 4X faster
- See details on Section 5.9 ISP in-circuit programming example.

5.2.4 **EMMC ONE-KEY-GHOST**

When reading EMMC, the one-key-ghost project can be automatically generated to realize the one-key copying of the original chip content to the new chip. See details on section 5.8 one-key-ghost example.

5.2.5 **Other EMMC Functions**

- The working frequency and working voltage of EMMC can be adjusted manually.
- Data in the user area can be read and written in blocks and multiple files
- If there is a password protection when deleting, perform forced erase
- Password can use ordinary password or SHA1 encrypted password
- The value of ECSD/CSD can be imported from the ECSD_CSD file or manually set by the user
- Data CRC error can be ignored during read operation
- Optional automatic verification after the reading operation is completed.
- There are high-definition pin diagrams and ISP wiring diagrams in the device information. The pin check and locate the BGA pin.
- EMMC has real-time accurate rate display during read and write operations, and the final total time

5.3 EMMC select IC

5.3.1 **The usual method to select IC**

Type in IC model (for example, MTFC4GMVEA-1M), as shown below (You can also choose AUTO EMMC):
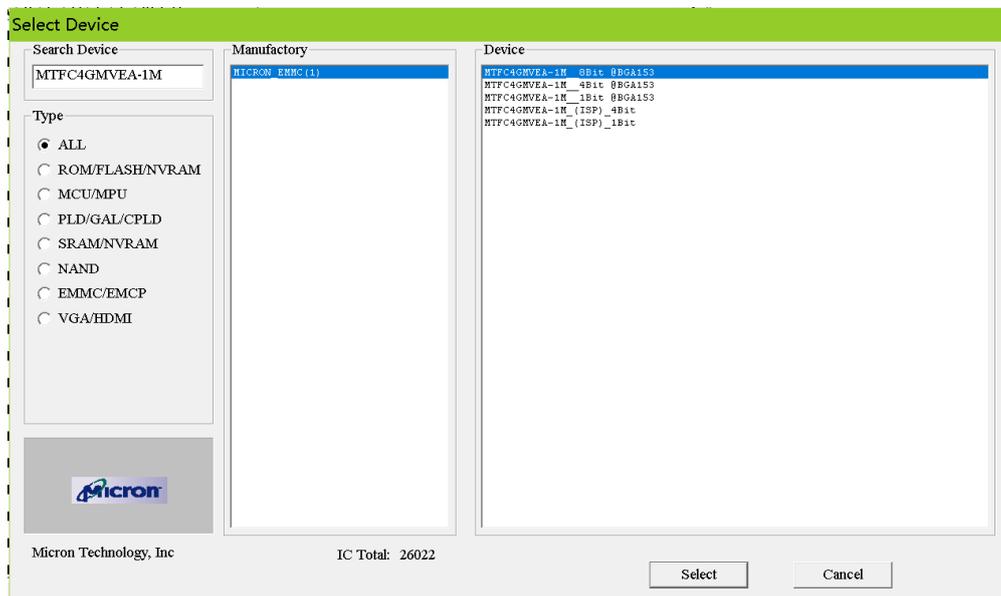
In general, there are 5 options for the same model EMMC:

- 8bit means 8 bits of data bus width is used when programming, and works at the fastest speed.
- 4bit means 4 bits of data bus width is used when programming, it is available when chip pin D7:4 is damaged.
- 1bit means 1 bit data bus width is used when programming, and can be used when chip pin D7:1 is damaged, and is the slowest.
- ISP_4Bit: 4 bits of data bus width is used when programming through ISP. At this time, the programming speed is also faster through ISP.
- ISP_1Bit: 1bit of data bus width is used when programming through ISP.

5.3.2 **Select AUTO_EMMC**

When IC model cannot be found in the list, you can type in "AUTO" to select the corresponding package as below:
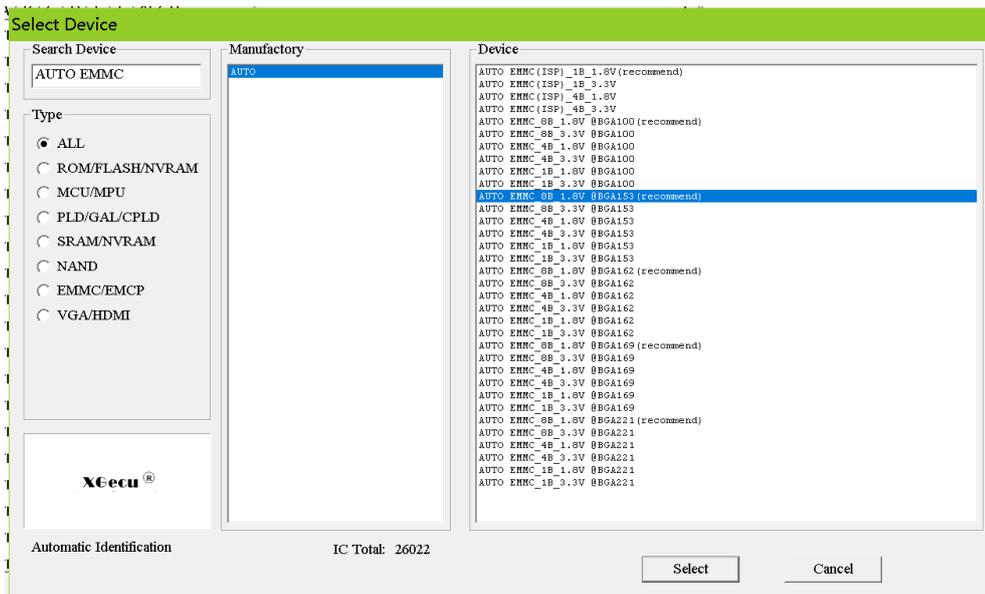
- If you program with adapter, select IC of the corresponding package, such as BGA153:
- There are 6 options under BGA153, usually choose the first 8B_1.8V, which represents 8-bit of data bus width is used, IO voltage is 1.8V.
- If you in-circuit program via ISP, usually choose ISP_1B_1.8V (in general, the motherboard IO voltage is 1.8V). If the motherboard IO voltage is 3.3V, choose ISP_1B_3.3V.
- If choose 1.8V, cannot be programmed normally on motherboard, you can try to change to 3.3V for testing.

5.4   EMMC Operation Options

5.4.1 **Programmer Operation Options**
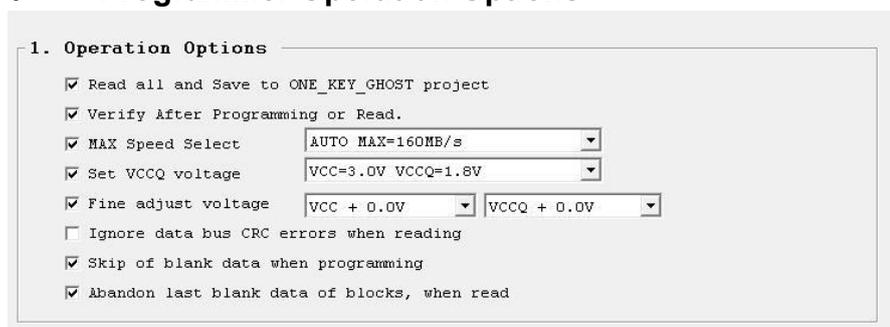
- Read all areas and save one-key-ghost project

  while reading, automatically analyze the chip, read all the data. This option will ignore the next (read, program, erase) option in the part of the selection, will automatically select according to the chip content. After reading, in the data folder a project file EMMC_GHOST.MPJ will be automatically generated. If the user wants to clone a chip, just open this ghost project file and directly program it to the new chip.

- Automatically verify after programming or read

  after reading or programming, the data will be automatically verified once.

- EMMC Select Clock

  Select the clock frequency, AUTO is default. If the reading of data is unstable, you can also reduce the clock frequency for testing.

- EMMC select Power voltage

Select the IC power voltage, a great part of the chip works at VCC=3.3V/VCCQ=1.8V, with extremely high stability.

- EMMC fine adjust voltage

  If the reading is unstable, can fine-tuning the voltage. It is possible to get a better stable effect. Users can do various tastings of voltage fine-tuning.

- Ignore data bus CRC errors when reading

  CRC verify error will be ignored when reading, this option is not recommended.

- Skip of blank data when programming

  during the verifying or programming, the blank data is skipped in order to increase the programming speed.

- Abandon last blank data of blocks

  When reading data, if the last data of a block is entirely blank (consisting entirely of 0x00 or 0xFF), then when saving the data file, the last blank data will be discarded and not saved in the data file. If the entire chip block is blank, the length of the data file may be 0, which can typically save a significant amount of hard disk space. and other software can also use directly. If the user wishes to save the last blank data, they can uncheck this option.

5.4.2 **Read, Program, Erase options**
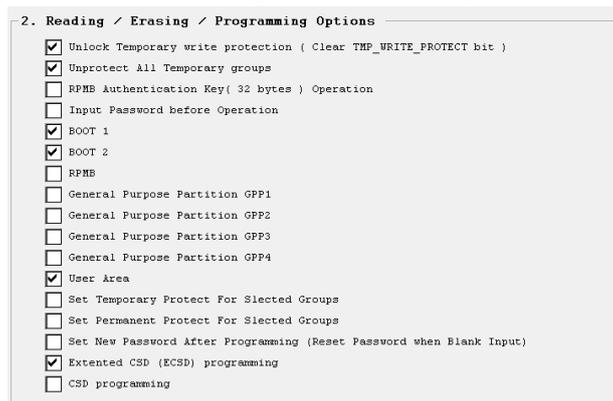
Select the chip area to read or programmiing. If an item is not selected, no operation will be performed for this item.

The key option RPMB: when programming RPMB, the option [RPMB Authentication Key Operation] must be checked and the correct 32 bytes authentication key must be loaded in the file menu before programming. After the authentication key is written, the chip will use this authentication key permanently and cannot be rewritten with a new authentication key.

5.4.3 **Erase, Blank Check option**
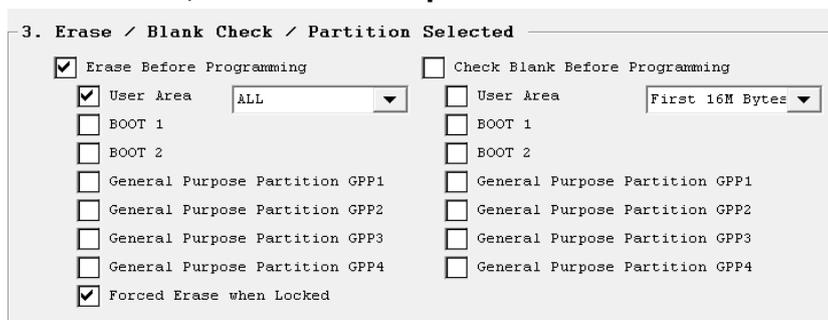
The erase and blank check operations performed before programming, generally do not need to be changed, just use the default settings. If the chip is new while mass production, the erase can also be unchecked.

5.4.4 **File path and file name settings**

```
4. File name / Path / Address Assigned

            File Directory :                          Select Directory

   D:\Xgpro_T56\UserData\EMMC_Data

        ECSD+CSD File :   ECSD_CSD.BIN              ...    Analysys ECSD
          BOOT1 File :    BOOT1.BIN                 ...
          BOOT2 File :    BOOT2.BIN                 ...
           RPMB File :    RPMB.BIN                  ...
           GPP1 File :    GPP1.BIN                  ...
           GPP2 File :    GPP2.BIN                  ...
           GPP3 File :    GPP3.BIN                  ...
           GPP4 File :    GPP4.BIN                  ...

           User Area :   (●) Single File Mode ,All User Area
                         ( ) Multiple Partition(Address aligned by 64K)

   Single User Aear File :   UserData.BIN          ...

            Start Address      End Address        File Name      Partition Size
   [✓] <1>  00 0000 0000      00 001F FFFF      UserPart_1.BIN      2048 KB
   [✓] <2>  00 0020 0000      00 002F FFFF      UserPart_2.BIN      1024 KB
   [✓] <3>  00 0100 0000      00 04FF FFFF      UserPart_3.BIN      65536 KB
   [ ] <4>  00 0000 0000      00 0000 0000      UserPart_4.BIN         0 KB
   [ ] <5>  00 0000 0000      00 0000 0000      UserPart_5.BIN         0 KB
   [ ] <6>  00 0000 0000      00 0000 0000      UserPart_6.BIN         0 KB
   [ ] <7>  00 0000 0000      00 0000 0000      UserPart_7.BIN         0 KB
   [ ] <8>  00 0000 0000      00 0000 0000      UserPart_8.BIN         0 KB
   [ ] <9>  00 0000 0000      00 0000 0000      UserPart_9.BIN         0 KB
   [ ] <10  00 0000 0000      00 0000 0000      UserPart_10.BIN        0 KB
   [ ] <11  00 0000 0000      00 0000 0000      UserPart_11.BIN        0 KB
```

- Select the data folder

   set a new folder before reading or programming, the folder name should preferably be device name + IC model for future use, all the data read will be stored under this folder. Including the automatically generated one-key-ghost project files. Of course, the folder name can be changed after reading. When programming, just select the data folder and make sure that all the data files be located under this folder.

- File name, read generally do not need to change the file name.

   If new product development, mass production, select the data file of the corresponding area.

   Note: All files must be set in the same folder.

- Analyze ECSD button

   Check out the information related to ECSD file under the folder.

- If the development of production

   ECSD file may not exist and user can set the relevant configuration in [Device Config].

- Blocks mode

   it can be sliced up to 16 blocks, block address must be 64K aligned, 1K=1024 bytes

   Users can write or read the USER Area in blocks, which is convenient for flexible applications when programming in bulk. The blocks mode also has a special function: if the unstable chip has nearly reached the end of its useful life and you want to read the data inside, you can read in blocks. After reading of the all data in the good blocks is completed at one time, then the bad blocks can be read separately for several times (while reading, disable the option "Read all and Save to ONE_KEY-GHOST project").

5.5 EMMC Device Configuration

5.5.1 **EMMC Password Setting**

```
┌─ 1. Password  Set ──────────────────────────────────────────────────┐
│                                                                      │
│           Password In the Device (ASCII) :  [                    ]   │
│   New Password (ASCII, Reset Password when Blank) :  [            ]   │
│                                                                      │
│                        [ ]  Use SHA-1 Secure Hash Algorithm          │
│                                                                      │
└──────────────────────────────────────────────────────────────────────┘
```

EMMC can use password protection, that only protects the USER Area, BOOTx/GPPx/RPMB partition, cannot be protected by password.

5.5.2 **ECSD Setting**

```
                          [✓]  Load ECSD_CSD from BIN file(else user define)when Programming
┌─ 2. Extended CSD ─────────────────────────────────────────────────────┐
│                                                                        │
└────────────────────────────────────────────────────────────────────────┘
```

 [ECSD CSD values are loaded from the file when programming] button

When checked, all fields of ECSD and CSD are loaded from the file and the set values in the interface are invalid.

set fields by user while unchecked.

☑ 编程时 ECSD CSD值从ECSD_CSD文件载入(未选时用户设定)

2. Extended CSD

☐ **PARTITION_SETTING_COMPLETED[155]** (<OTP> bits0 is valid , bit7:1 Reserved)	00
☐ **WR_REL_SET[167]** (<OTP> bits4:0 is valid , bit7:5: Reserved)	1F
☐ **GP_SIZE_MULT_GP1[145-143]** (<OTP> defines GPP1 size)	000000
☐ **GP_SIZE_MULT_GP2[148-146]** (<OTP> defines GPP2 size)	000000
☐ **GP_SIZE_MULT_GP3[151-149]** (<OTP> defines GPP3 size)	000000
☐ **GP_SIZE_MULT_GP4[154-152]** (<OTP> defines GPP4 size)	000000
☐ **ENH_SIZE_MULT[142-140]** (<OTP> defines Enhanced User Data Area Size)	000000
☐ **ENH_START_ADDR[139-136]** (<OTP> defines Enhanced User Data Start Address)	00000000
☐ **PARTITIONS_ATTRIBUTE[156]** (<OTP> bits4:0 is valid ,sets enhanced attribute in GPPx)	00
☐ **EXT_PARTITIONS_ATTRIBUTE[53-52]** (<OTP> sets extended attribute in GPPx)	0000
☐ **USE_NATIVE_SECTOR[62]** (<OTP> Valsue is 0x00 or 0x01 , 0x02-0xFF: Reserved)	00
☐ **PRODUCT_STATE_AWARENESS_ENABLEMENT[17]** (Bit5:4 can be set, bits1:0 is read only, others Reserved)	01
☐ **PRE_LOADING_DATA_SIZE[25-22]** (0x00000000-0xFFFFFFFF , value reset after a power cycle)	00000000
☐ **PRODUCTION_STATE_AWARENESS[133]** (the host reports to the device its production state)	00
☐ **PARTITION_CONFIG[179]** (This register defines the configuration for partitions)	00
☐ **BOOT_CONFIG_PROT[178]** (<OTP> This register defines boot configuration protection)	00
☐ **BOOT_BUS_CONDITIONS[177]** (This register defines the bus width for boot operation)	00
☐ **BOOT_WP[173]** (<OTP> Boot area write protection register)	00
☐ **USER_WP[171]** (<OTP> User area write protection register)	00
☐ **FW_CONFIG[169]** (<OTP> Bit[0]: Update_Disable Bit[7:1]: Reserved)	00
☐ **BKOPS_EN[163]** (<OTP> Bit[0]:=1 ENABLE Bit[7:1]: Reserved)	00
☐ **RST_n_FUNCTION[162]** (<OTP> Bit[1:0]: valid Bit[7:2]: Reserved)	00
☐ **SEC_BAD_BLK_MGMNT[134]** (<OTP> Bad Block Management mode, bit7:1 Reserved)	00
☐ **PERIODIC_WAKEUP[131]** (Periodic Wake-up timer)	00
☐ **SECURE_REMOVAL_TYPE[16]** (<OTP> Bit [5:4]: Cfg. Secure Removal Type, B3:B0 read only)	01

- All ECSD register fields options, only when checked, the chip will be programmed and verified. Unchecked registers will not be programmed and verified.

- The first major option **[PARTITION SETTING COMPELTED 155],** when checked, the value must be 01, and it's subparts must be all checked, and the subparts must meet the EMMC standard, otherwise they cannot be written. This major option is to set the EMMC partition and partition attributes, the chip is not allowed to repeatedly set the partition size and attributes.

- All registers marked with OTP are written at once, so please set them carefully

- When **[USE_NATIVE_SECTOR 62]** button is checked, the value must be 01, otherwise it is the error.

- All other registers can be checked, or only the non-00 registers be checked

- Note: The above options will be checked automatically when the original chip is read with the one- key-ghost project is generated automatically.

5.5.3 CSD Setting

3. CSD

|	
FILE_FORMAT_GRP <OTP> :	0 = as FILE_FORMAT ▾
FILE_FORMAT <OTP> :	00 = Hard disk-like file system ▾
COPY <OTP> :	0 = Original ▾
PERM_WRITE_PROTECT <OTP> :	0 = Disable ▾
TMP_WRITE_PROTECT :	0 = Disable ▾
ECC :	00 = Default ▾

- If any option in CSD is non-zero, check the option **[CSD programming]** in EMMC Operation Options (2.Reading/ Erasing/ Programming Options)
- When auto-generating one-key-ghost project, the software will check the all ECSD/CSD options automatically.

5.5.4 **Write Protect Groups (WPG)**
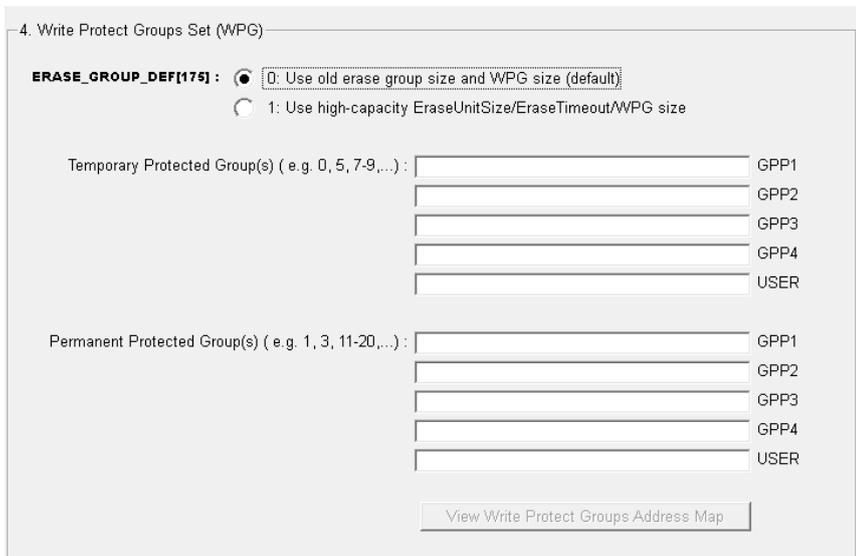
- Write Protect is for the part of chip data, the protection type and protection group are set according to the user's requirements.
- While Analysis of the original chip function: you can check which groups of the original chip was protection set. If you need to set the same write protection group as the original chip, please type in manually.
- When the ECSD file has been loaded, you can check the specific address corresponding to the write protection group.

5.6 EMMC Analysis
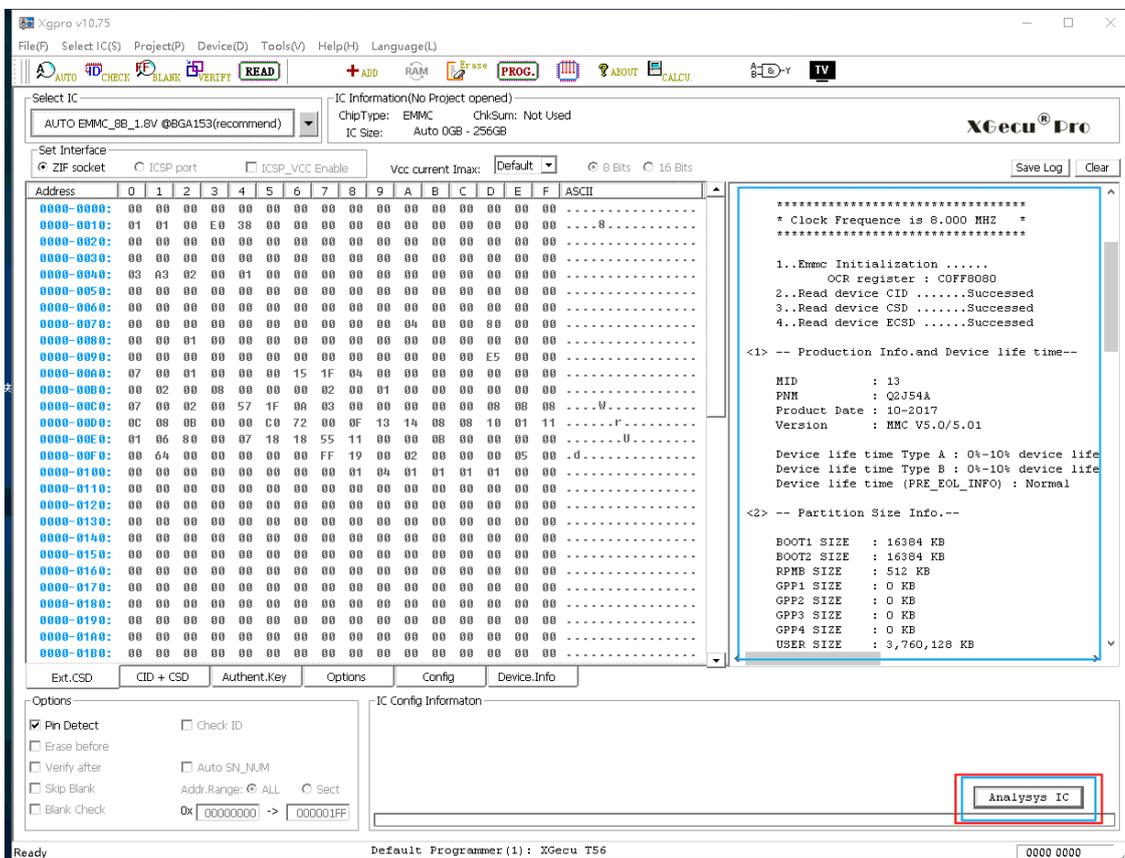
- Analysis of the original chip is carried out at a lower clock frequency of 8MHZ. You can test whether the connection is normal: If not, try to change the working voltage, or fine-tune the working voltage or select a different bus width for several tests. In that way, for the chips with unstable performance, may be you can get the correct result.
- Connect the adapter or ISP, click button **[Analysis IC]**
- Click button **[SAVE LOG]** to save the analysis result
- From the result, you can get the use condition of the chip
  - Display information about the IC's manufacturing
  - Visually display IC use condition of up to 8 partitions
  - On each partition display whether it is write-protected, readable, rewritable or erasable
  - Display whether the entire chip is temporarily write-protected or permanently write-protected
  - Display if the IC is password protected
  - Display whether the IC uses 32-bit authentication key to write for RPMB.
  - Display other important parameters of ECSD CSD CID

Example: **The IC analysis results are shown below**:

```
**************************************************
 1 Programmer Connected.
**************************************************
  Device 1: XGecu T56 Ver: 00.01.37
       USB POWER VOLTAGE: 04.91V
       USB SPEED MODE: HS 480MHZ
**************************************************

  < Analysis  IC >

   Pin Detected Passed.

       *********************************
       * Clock Frequence is 8.000 MHZ   *
       *********************************

     1..Emmc Initialization ......
          OCR register : C0FF8080
     2..Read device CID .......Successed
     3..Read device CSD .......Successed
     4..Read device ECSD ......Successed

  <1> -- Production Info.and Device life time--

     MID         : 13
     PNM         : Q2J54A
     Product Date : 10-2017
     Version     : MMC V5.0/5.01

     Device life time Type A : 0%-10% device life time used
     Device life time Type B : 0%-10% device life time used
     Device life time (PRE_EOL_INFO) : Normal

  <2> -- Partition Size Info.--

     BOOT1 SIZE  : 16384 KB
     BOOT2 SIZE  : 16384 KB
     RPMB SIZE   : 512  KB
     GPP1 SIZE   : 0 KB
     GPP2 SIZE   : 0 KB
     GPP3 SIZE   : 0 KB
     GPP4 SIZE   : 0 KB
     USER SIZE   : 3,760,128 KB
                   ( 0x 00_E580_0000 )

  <3> -- RPMB Security and Password --

     Authentication Key not yet programmed, RPMB not used

     Password Protect Features : YES
     Password Protected       : Not Locked
```

```
<4> -- Write Protection --

   ERASE_GROUP_DEF        : 00
   Disable  PERM_WP_PROTECT : No
   User Permanent protection: Enable
   Whole Chip Permanent Protect  : Not protected
   Whole Chip Temporary Protect  : Not protected
   BOOT1 Write Protection Status : Not protected
   BOOT2 Write Protection Status : Not protected

   Write Protect Group Size : 0x800000 ( 8192 KB )

   User Write Protection Group (0=Not Protected  TE=Temp Protected
   ----------------------------------------------------------------
    WPG  Start Addr.   0  1  2  3  4  5  6  7  8  9  10 11 12 13 :
   ----------------------------------------------------------------
   00000 00-00000000   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   00032 00-10000000   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

   ..............

   00448 00-E0000000   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   ----------------------------------------------------------------
   USER Partiton --- Not any Write Protected . Total Write Protect

<5> -- other Informations --

   MAX_READ_BL_LEN   : 512 bytes
   MAX_WRITE_BL_LEN  : 512 bytes
   MAX_TRAN_SPEED    : 55.000 MHZ

   ENH_START_ADDR            : 00000000
   ENH_SIZE_MULT             : 000000
   MAX_ENH_SIZE_MULT         : 0000E5
   PARTITIONS_ATTRIBUTE      : 00
   WR_REL_SET                : 1F
   WR_REL_PARAM              : 15
   PARTITION_SETTING_COMPLETED : 00

   HC_WP_GRP_SIZE    : 16
   HC_ERASE_GRP_SIZE : 1
   WP_GRP_ENABLE     : 1
   WP_GRP_SIZE       : 15
   ERASE_GRP_MULT    : 31
   ERASE_GRP_SIZE    : 31
   CCC               : 00F5
   DSR implemented   : 1
   PARTITION_ACCESS  : 30 ms
   ERASED_MEM_CONT   : 00
   DYNCAP_NEEDED     : 00
   SECURE_WP_INFO    : 00
   SEC_ERASE_MULT    : 18
   ERASE_TIMEOUT_MULT: 11
   NATIVE_SECTOR_SIZE: 00 (512B)
   INI_TIMEOUT_AP    : 10000 ms
   INI_TIMEOUT_EMU      : 0 ms

<6> -- Result -- View More Information in ECSD and CSD
   ----------------------------------------------------------------
   1 . BOOT1 Partition Not be Permanent write-protected, it can be
   2 . BOOT2 Partition Not be Permanent write-protected, it can be
   3 . GPP1 GPP2 GPP3 GPP4 The Partition Invalid
   4 . USER  Partition Not be Permanent write-protected, it can be
   5 . RPMB  Partition is Blank, not be Used.
   6 . This chip's content can be completely copied to another chi
   ----------------------------------------------------------------
```
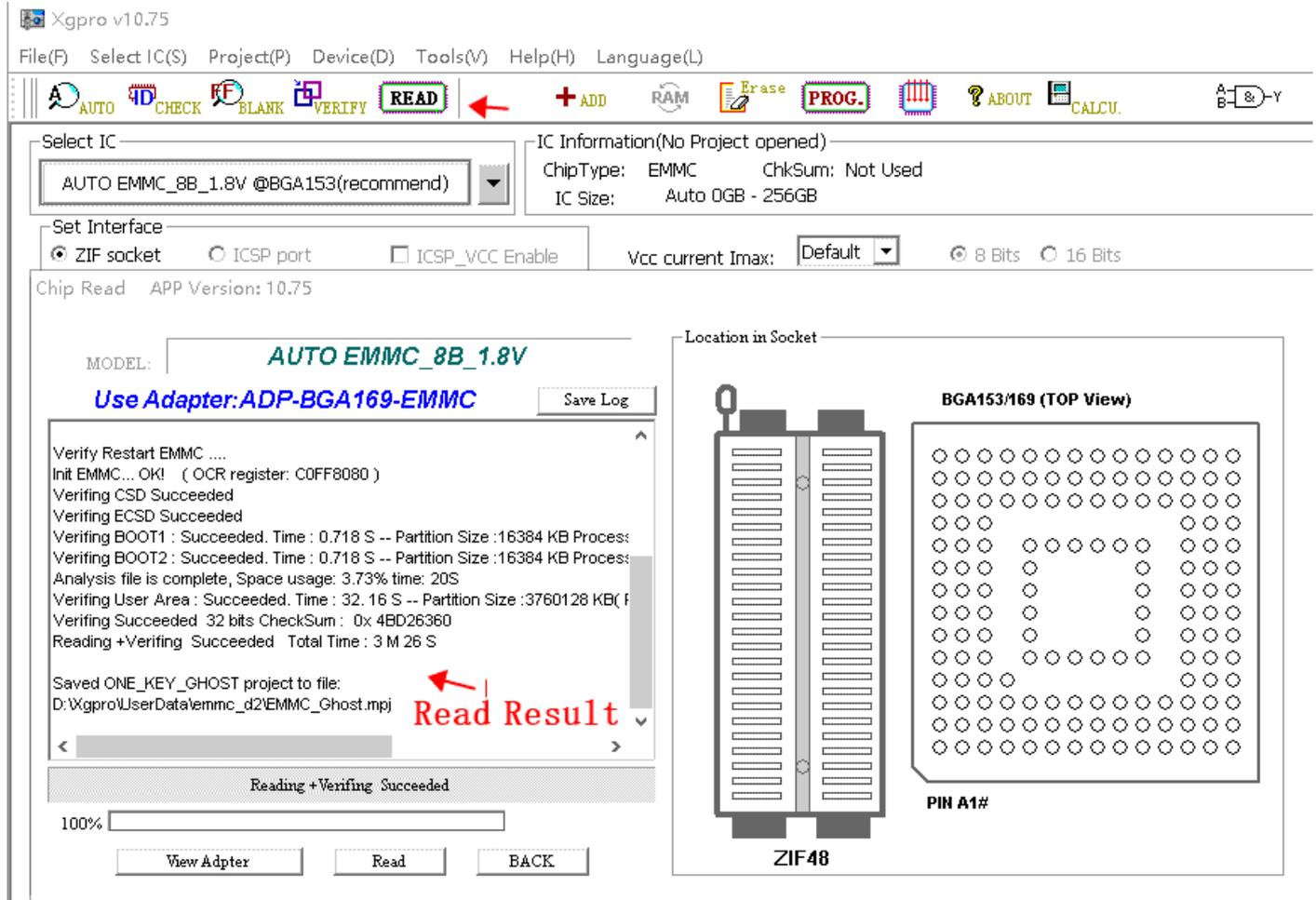
5.7 Read EMMC

- Select IC (example: JY001_8bits, this is a 4GB IC)
- Set the folder for data storage in [Operation Options] (EMMC_DATA, you can set it arbitrarily)
- Don't need to change any other settings, just click **[Read]** button to start reading.

Reading results as below:

- 4GB IC, the total time to "Read + Verify" is 4 minutes 28 seconds, this is a more typical time for actual operation for your reference. Generally, the IC of more larger capacity will get the faster reading and writting speed.
- After reading is finished, an EMMC_GHOST.mpj project file is automatically generated as shown above, just open this project when programming.

After reading, the contents of the folder are as below: at least 5 files will be generated, one of which is a one-key-ghost project file. If IC uses RPMB or GPPx, there will be more files.

37
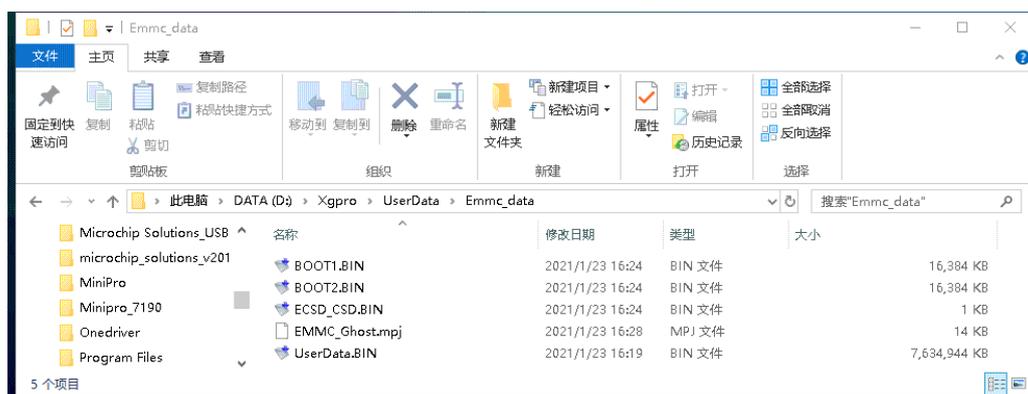
Note: Before reading, it will check the remaining capacity of the disk, if the capacity is not enough, select the data folder in the operation options and save it to another disk
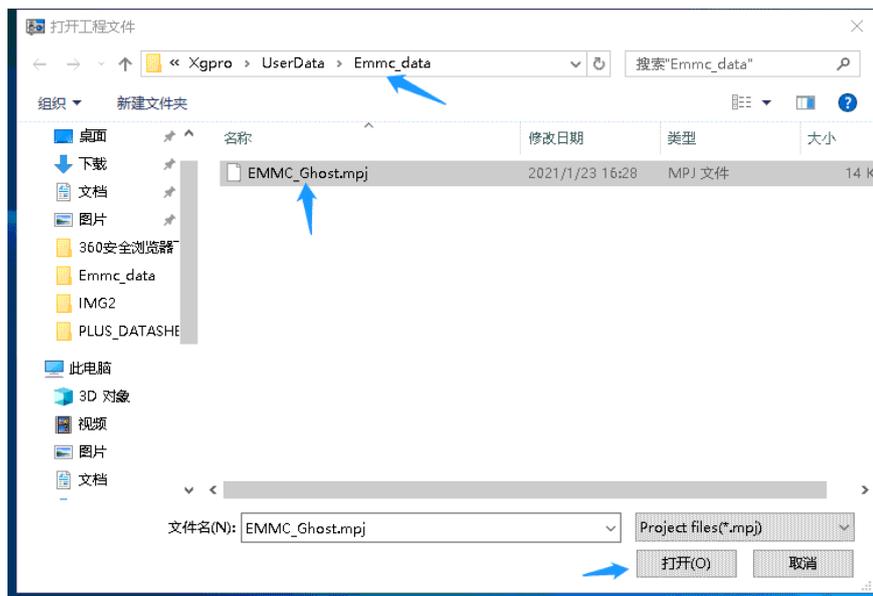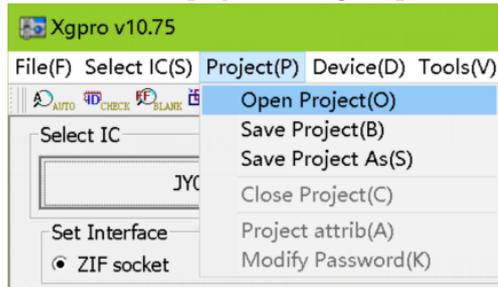
5.8 Program EMMC :   one-key-ghost example

One-key copy of new IC read in the previous section

Just open the project file [EMMC_GHOST.mpj]] previously read and copy it with one key.

5.8.1 **Select IC (for example: JY001_8bits)**

How to select IC correctly, you can refer to the previous contents.

### 5.8.2 **Click [Open Project] in Project Menu**

After clicking **[Open]**, the project has been loaded

5.8.3 **Program new IC (done by one key)**

Don't need to make any other settings, at this time the data folder, ECSD, CSD and other operation options have all been set automatically, directly click **[PROG.]** button to copy and program.

Note: For projects that need to program the RPMB partition, please see section 5.8.4 for details
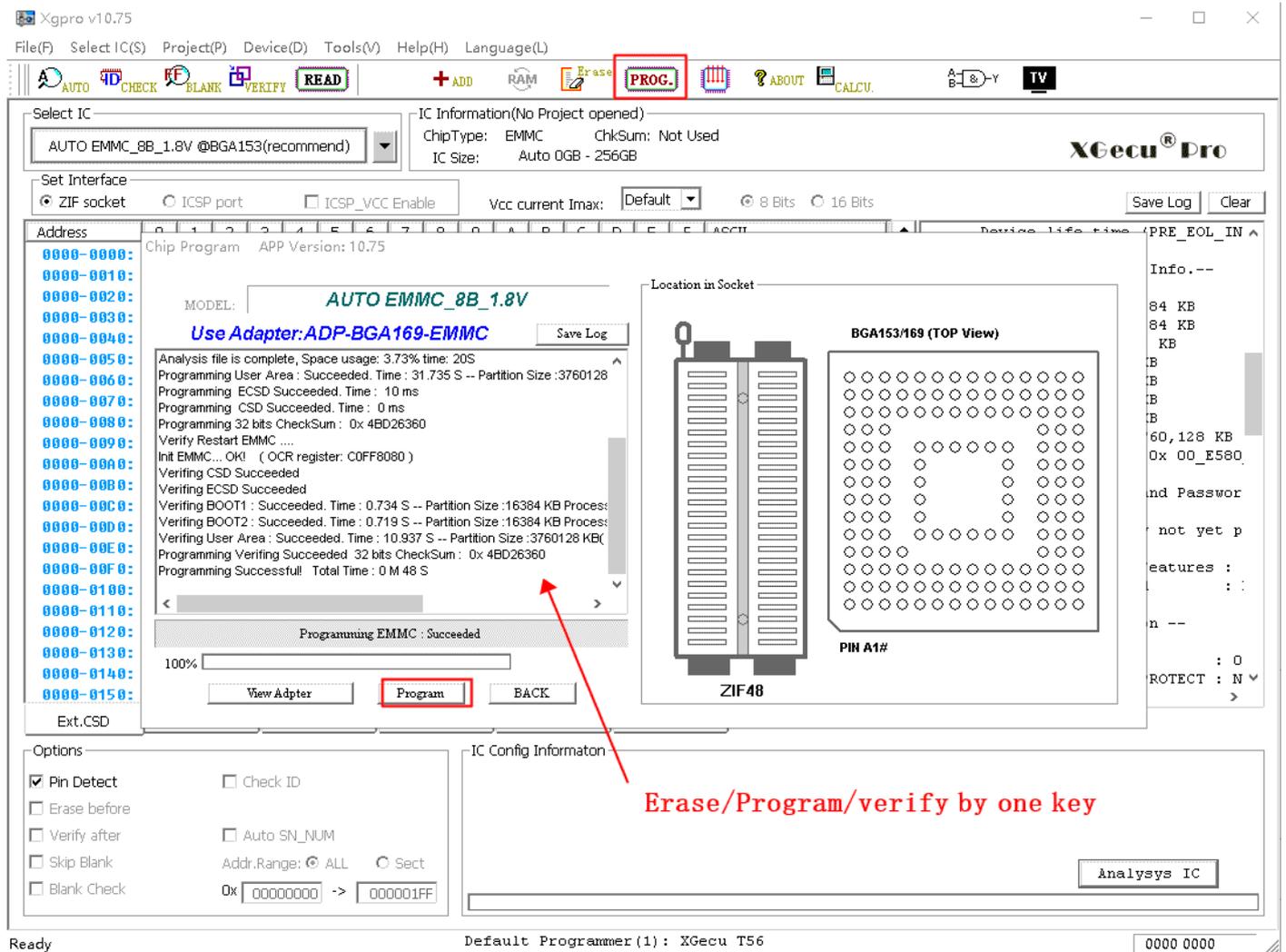
Programming process as shown above, "Program + Verify" total time of 0 minutes and 48 seconds, it is faster than reading, because the blank data was skipped when programming.

The new chip copy is completed.

5.8.4 **One-key-ghost project notes:**

- When programming the project, the capacity of the BOOT, RPMB, and USER partitions of the original chip will be compared with that of the programmed chip. If the capacity of the new chip in any one partition is smaller than the original chip, will stop programming.

- When programming the project, the EMMC version of the original chip and the new chip will be compared. If the version of the new chip is lower than that of the original chip, the programming will be stopped.

- For projects that need to program RPMB partitions, the correct 32-byte authentication key is required. The user can load the authentication key in the File menu after opening the project, and save the project after loading. This authentication key will be used as the new authentication key when programming the RPMB. (Authentication key can also be loaded before reading. Save to project file when saving a project automatically with key.)

- If the authentication key is not loaded, when programming, the all-blank data 0xFF will be used as the authentication key.

- Important Note: authentication key, one chip is only allowed to be programmed once, it cannot be read and cannot be erased. If the authentication key is lost, the RPMB data will be permanently immutable. For specific devices, incorrect authentication key will not pass.

- If new chip's RPMB has been used when programming, but there is no authentication key, it will prompt key error. At this time,

the RPMB data cannot be programmed. If you have correct authentication key , the RPMB can be programmed normally.

5.9   EMMC ISP in-circuit programming example

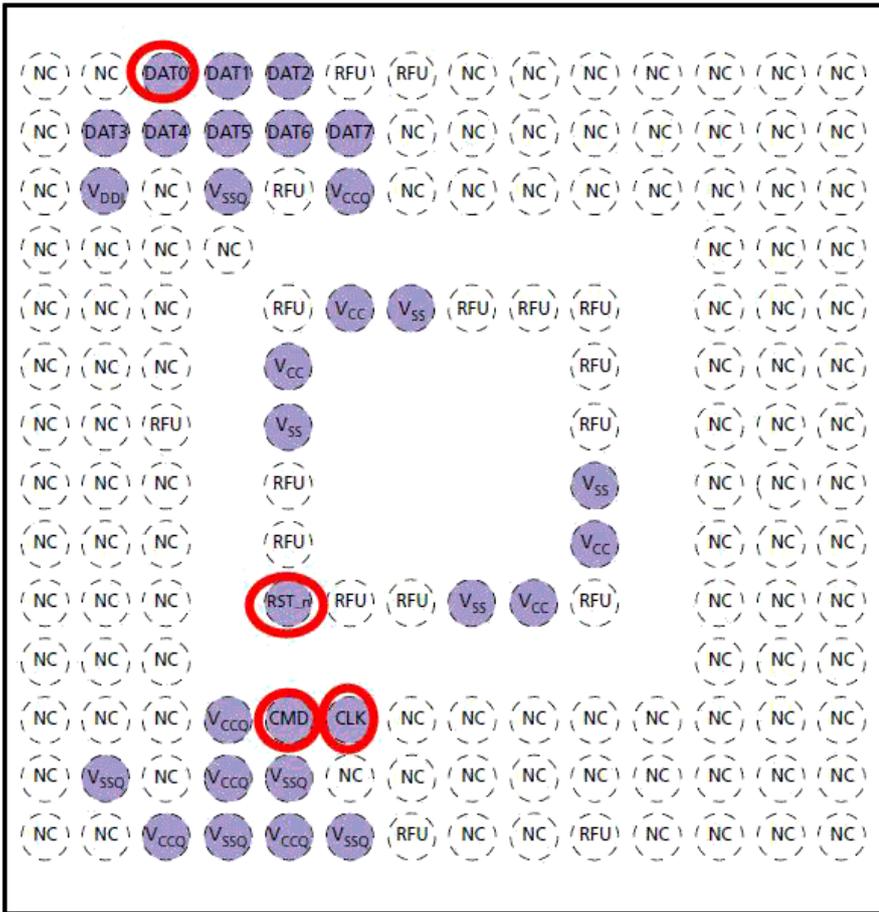### 5.9.1 **EMMC ISP connection**
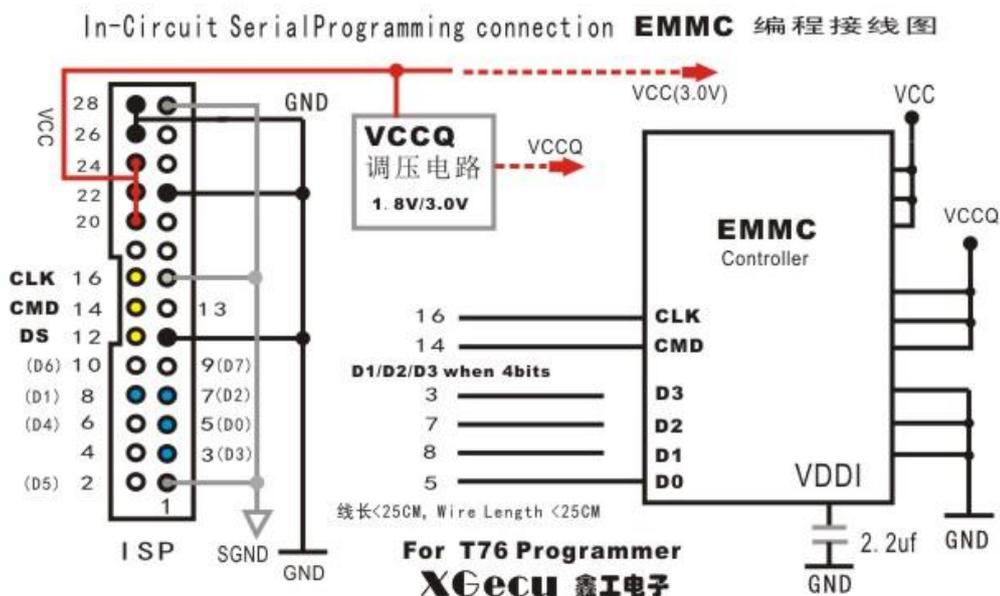
In general, 1bits only need to connect GND / CLK / CMD / D0 to the motherboard, then power the motherboard.
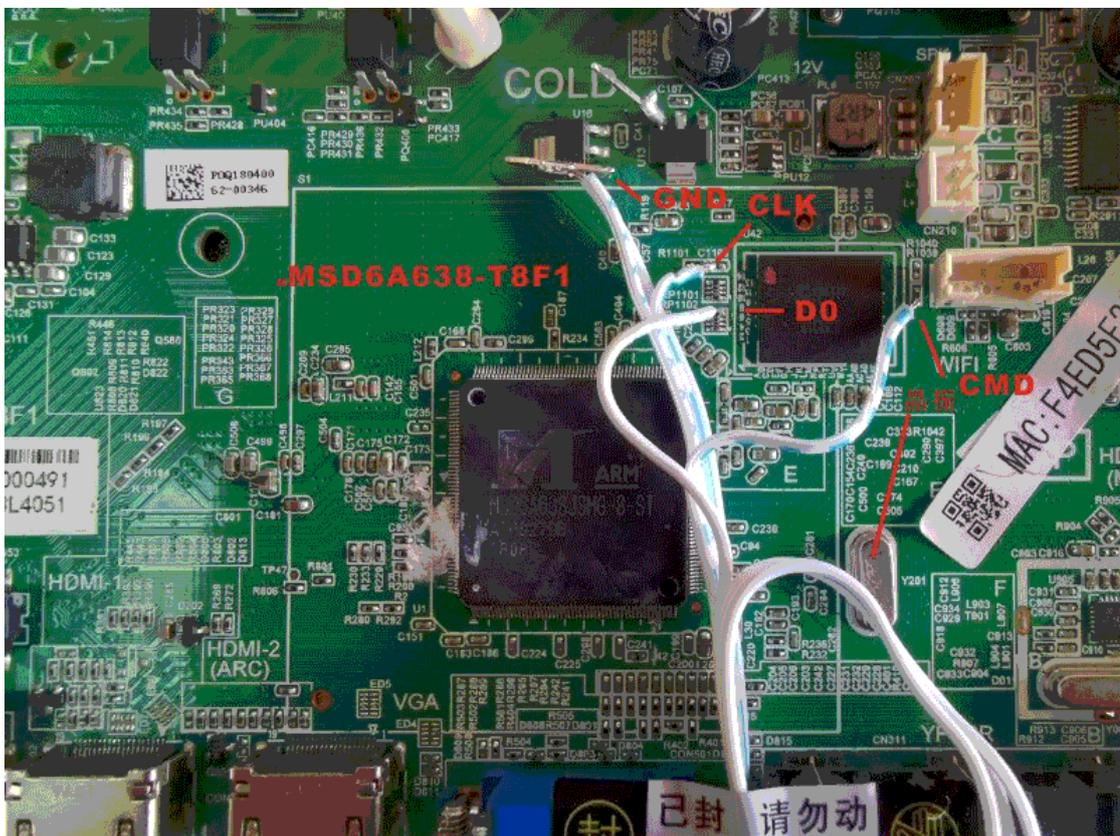
Notes：

- ISP has two grounds, both grounds need to be connected, and the ground point should be as close to the CLK line as possible.
- CLK line and the two GND lines do not separate, and CLK line should not cross with other lines, CLK line generally has a series of resistors R, you had better remove it.
- Both ends of the crystal of the MCU in PCB, connected to ground, stop the MCU from working.
- Connect the lines, power up the motherboard and check if the RST_n pin of the EMMC is high. If the RST_n voltage is 0, you need to pull up the pin, otherwise EMMC will not work. You had better connect 1K resistor to VCCQ (1.8V or 3.3V). This pin is the reset pin of EMMC.
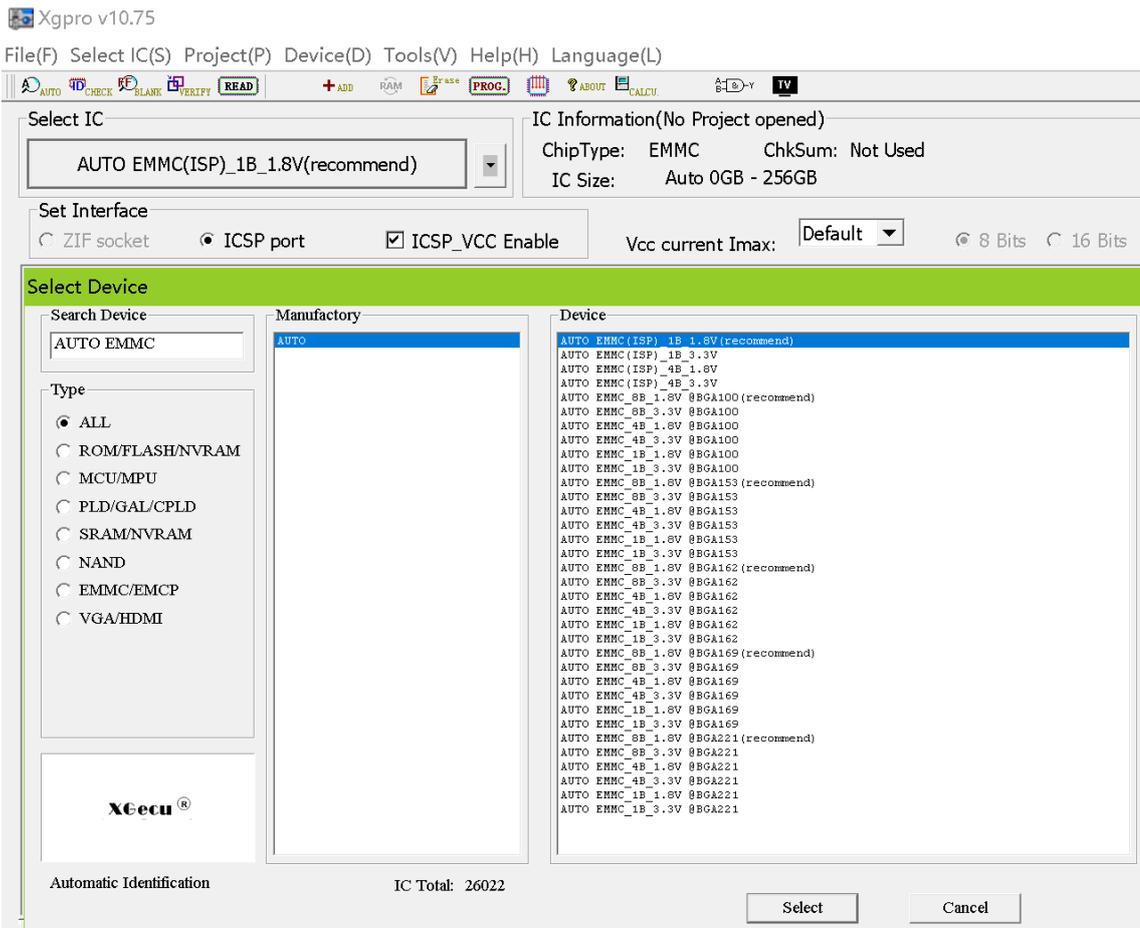
EMMC BGA153/169 IC PINS:

The following photo shows the wiring example of the MSD6A638-T8F1 TV board:

5.9.2 **Select IC and Analysis IC**

Xgpro v10.75

File(F) Select IC(S) Project(P) Device(D) Tools(V) Help(H) Language(L)

[toolbar: AUTO CHECK BLANK VERIFY READ ADD RAM Erase PROG. ABOUT CALCU. TV]

Select IC

AUTO EMMC(ISP)_1B_1.8V(recommend) ▾

IC Information(No Project opened)
ChipType: EMMC ChkSum: Not Used
IC Size: Auto 0GB - 256GB

Set Interface
○ ZIF socket ● ICSP port ☑ ICSP_VCC Enable Vcc current Imax: Default ▾ ● 8 Bits ○ 16 Bits

Select Device

Search Device
AUTO EMMC

Type
● ALL
○ ROM/FLASH/NVRAM
○ MCU/MPU
○ PLD/GAL/CPLD
○ SRAM/NVRAM
○ NAND
○ EMMC/EMCP
○ VGA/HDMI

Manufactory
AUTO

Device
```
AUTO EMMC(ISP)_1B_1.8V(recommend)
AUTO EMMC(ISP)_1B_3.3V
AUTO EMMC(ISP)_4B_1.8V
AUTO EMMC(ISP)_4B_3.3V
AUTO EMMC_8B_1.8V @BGA100(recommend)
AUTO EMMC_8B_3.3V @BGA100
AUTO EMMC_4B_1.8V @BGA100
AUTO EMMC_4B_3.3V @BGA100
AUTO EMMC_1B_1.8V @BGA100
AUTO EMMC_1B_3.3V @BGA100
AUTO EMMC_8B_1.8V @BGA153(recommend)
AUTO EMMC_8B_3.3V @BGA153
AUTO EMMC_4B_1.8V @BGA153
AUTO EMMC_4B_3.3V @BGA153
AUTO EMMC_1B_1.8V @BGA153
AUTO EMMC_1B_3.3V @BGA153
AUTO EMMC_8B_1.8V @BGA162(recommend)
AUTO EMMC_8B_3.3V @BGA162
AUTO EMMC_4B_1.8V @BGA162
AUTO EMMC_4B_3.3V @BGA162
AUTO EMMC_1B_1.8V @BGA162
AUTO EMMC_1B_3.3V @BGA162
AUTO EMMC_8B_1.8V @BGA169(recommend)
AUTO EMMC_8B_3.3V @BGA169
AUTO EMMC_4B_1.8V @BGA169
AUTO EMMC_4B_3.3V @BGA169
AUTO EMMC_1B_1.8V @BGA169
AUTO EMMC_1B_3.3V @BGA169
AUTO EMMC_8B_1.8V @BGA221(recommend)
AUTO EMMC_8B_3.3V @BGA221
AUTO EMMC_4B_1.8V @BGA221
AUTO EMMC_4B_3.3V @BGA221
AUTO EMMC_1B_1.8V @BGA221
AUTO EMMC_1B_3.3V @BGA221
```

XGecu ®

Automatic Identification IC Total: 26022

[Select] [Cancel]

- Type in "AUTO EMMC"
- Select AUTO_EMMC(ISP)_1B_1.8V
- Click **<Analyze IC>**button and check if the connection is normal. If there is no problem, will go to the next step

5.9.3 **ISP read data from original IC**

Click the "Read" button in the toolbar to read the clock settings. The T76 can operate at up to 80MHZ, and in 4b mode, it can achieve a maximum speed of 35MB/s. If there are reading errors, you can appropriately reduce the frequency

5.9.4 **ISP program IC**

- *Select IC AUTO_EMMC(ISP)_1B_1.8V
- *Open the one-key clone project file read
  Under normal condition, open the project file and program all the data read from EMMC to finish the copy with one key.
  If you don't need to program all the data, please uncheck the option that you don't need to write.
  Another method: After selecting IC, do not use the project file, just select the data folder. Check the area to be programmed in the operation options, that is completely set by user
- Click [PROG.] button in the toolbar to start programming IC
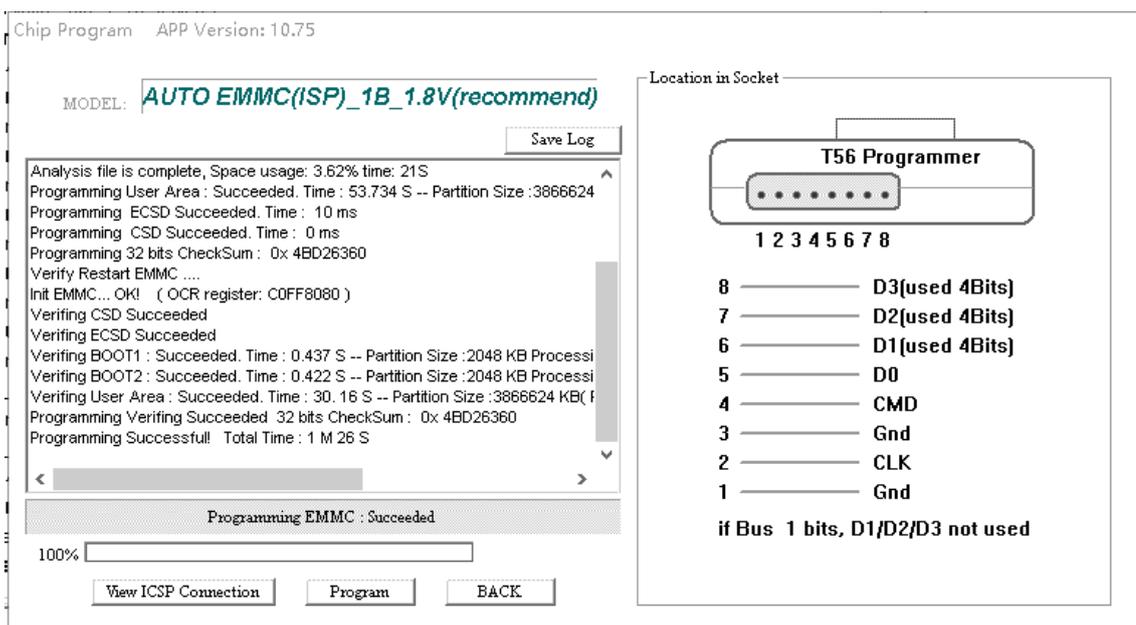
Open Project
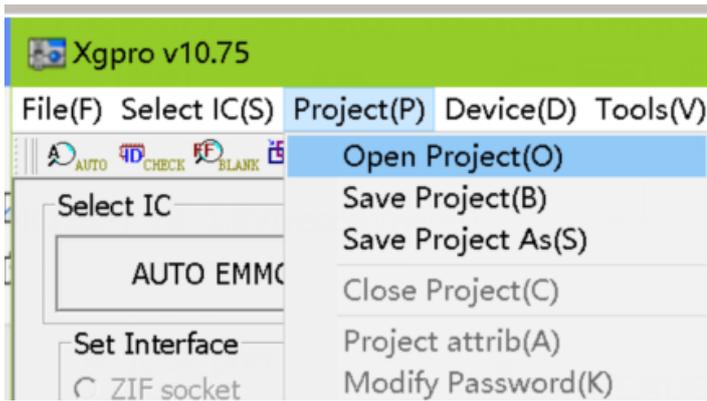
Open project file: EMMC_GHOST.MPJ
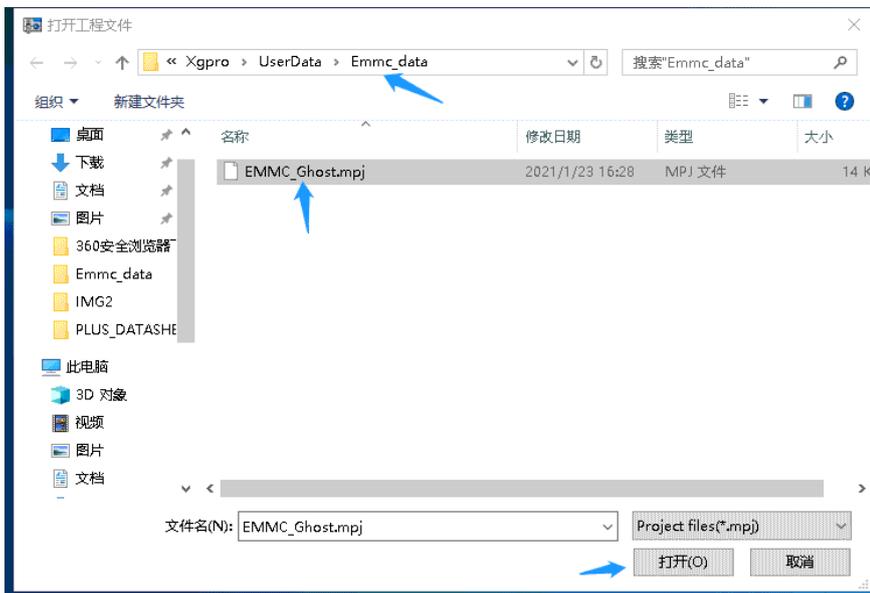
Click **[PROG.]** on toolbar

The whole programming process, as shown below, Erase, Program and Verify all partitions with one key.

6.NAND/SPI_NAND Programming

6.1 Characteristics of NAND FLASH

- NOR Flash randomly accesses all memory maps and dedicated interfaces (such as EPROM) address and data lines. But there is no dedicated address line in NAND flash. It is made up of an 8/16-bit wide interface bus that sends commands, addresses and data to internal registers, that provides a more flexible configuration for many original chips. The NAND flash architecture, emphasis on lower cost per bit, higher performance, and disk can easily upgrade through the interface, NAND is lower cost and higher capacity.

- NAND, bad blocks are allowed. Due to the production process of NAND, bad blocks will randomly appear in the factory chip. Bad blocks are initialized at the factory and marked as bad blocks in the special area. If bad blocks appear during use, they also need to be marked.

- Bit flip: If bit flip occurs on critical files, it will cause the system to hang up. Therefore, while using NAND FLASH, algorithms such as ECC/EDC must be used for data correction to ensure reliability.

- There is spare area. Just because NAND FLASH has the above two special features, spare area plays the role of putting bad block flags, ECC values, IC information and file information.

- NAND FLASH bad blocks manage: There are many ways to manage bad blocks in NAND FLASH. Different system providers may choose different management methods of bad block to meet the needs of product development. The software defines three common methods for user to choose from. The software defines three common methods for users to select and use, and enables user-defined ECC algorithms for flexible applications.

6.2 Structure of NAND

The following figure shows the MT29F4G08ABAEA as an example:

MT29F4G08 has 2 Planes *2048 blocks , the programmer is operated in blocks, index number 0-4095 blocks. Each block has 64 pages , and the processing of each block is separately written to the chip sequentially.

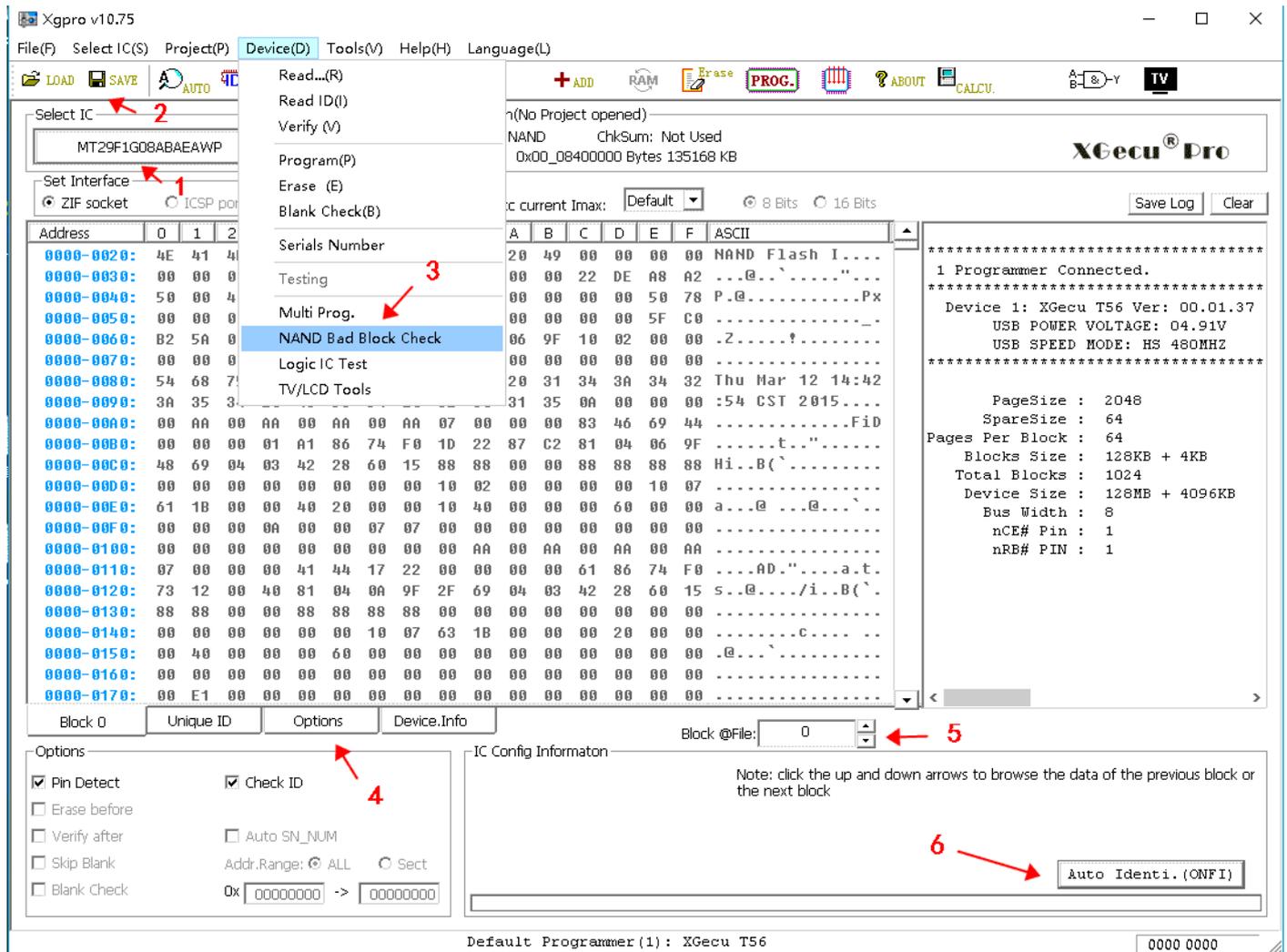The size of each page: 2048 bytes (Page size) + 64 bytes (Spare size) = 2112 bytes

The total capacity of IC is: 2112 * 64 (Page) * 4096 (Block) = 4224MB.

Figure 9: Array Organization – MT29F4G08 (x8)
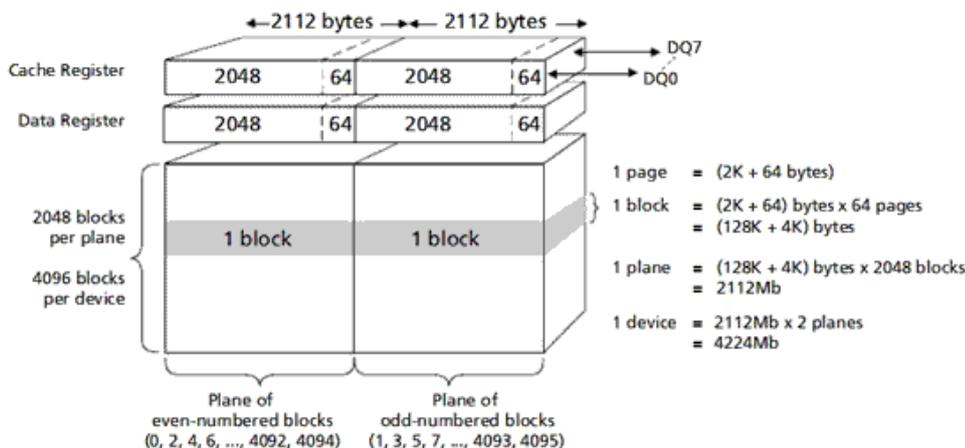
6.3　NAND Program main interface

After selecting IC, the software interface is as below:

1) Click select button to select IC model

2) Load and save: data files to load or save to file after reading

3) **<NAND bad block check>** function in the menu is only valid for standard bad block markers, for a small amount of specific device data, if the bad block marker is not standard, the check result is not accurate, you can ignore it.

4) Configuration and options: configuration of NAND programming method, see the next section for details: configuration and options (for equipment maintenance personnel, generally settings by default, do not need to change)

5) Data block switch: user browse the data buffer by block, after loading file or reading IC, user can use the software to directly browse the data in the set block.

6) ONFI automatic identification of IC parameters: the IC model not listed in the software, for the NAND chip in line with the ONFI standard, the user can use this function to automatically identify the chip parameters, automatically adjust the software parameters after identifying. You can read and write directly, and can be saved as a customized IC model, see Chapter 10 Add customized IC for details.

6.4 NAND Configuration and Options

Click **[Options]** button in Figure 4 to enter the programming configuration interface.

6.4.1 **Programming Operation Options**

Settings when reading or programming

Prog. Speed: if an error occurs during reading or programming, please try to select "Low Speed".

```
┌─ 1. Operation Options ──────────────────────────────────────────────────┐
│                                                                          │
│  ☑ Erase Before Programming              Prog.Speed:  ● Default          │
│     ☐ Not clear Bad Block Mark when Erase             ○ Low Speed        │
│  ☑ Skip Blank                                         ○ High Speed       │
│  ☑ Auto Verify after reading or Programming                              │
│  ☐ Blank Check Before Programming                                        │
│  ☐ Read Device Unique ID (UID)                                           │
│                                                                          │
└──────────────────────────────────────────────────────────────────────────┘
```

6.4.2 **Bit flip Permission**

Set by the required ECC error correction bit, usually set by default, no need to modify.

Note: When reading or writing to the chip, the NAND will be bit flipped, so the data file read may not be exactly the same every time.The file data is valid as long as the auto-verify passed when reading, that means the error data is in allowed range.

```
┌─ 2. Bit Flip Permission ────────────────────────────────────────────────┐
│                                                                          │
│  ○ 1b/512B    ○ 4b/512B    ○ 8b/512B    ○ 24b/1024B    ○ 40b/1024B  ● Disable │
│                                                                          │
└──────────────────────────────────────────────────────────────────────────┘
```

When SPI NAND enables the internal ECC error correction function, error bits are no permission.

6.4.3 **Customize NAND parameters**

The software allows user to set NAND parameters according to the IC data sheet. Theoretically, user can add any NAND IC by customize setting, see Chapter 10 Adding Custom IC for details.

```
┌─ 3. Customize NAND parameters ──────────────────────────────────────────┐
│                                                                          │
│  ☐ Customize Setting                                                     │
│                                                                          │
│       Page Size :  [4096   ▼]        VCC Voltage :  [3.0V      ▼]         │
│      Spare Size :  [224    ▼]          Bus Width :  [8 Bits    ▼]         │
│  Pages per Block:  [64     ▼]   PIN CONFIGURATION :  [ONFI STD  ▼]        │
│ BlocksPer#CE(Die):[2048   ▼]            PACKAGE :  [TSOP48    ▼]          │
│      #CEn (Die) :  [1      ▼]       Internal ECC :  [Disable   ▼]         │
│      #RBn Pins  :  [1      ▼]         Device ID :  [   2C DC   ]          │
│                                                                          │
│            [ View Config Resault ]                                       │
│                                                                          │
└──────────────────────────────────────────────────────────────────────────┘
```

6.4.4 **File name/ Path/ Partition Assigned**

6.4.4.1 Temporary Buffer Folder

The folder where temporary files are stored when a single file is accessed.

NAND files may take up a lot of disk space. Only while the capacity of disk is low, you need to set the folder to another disk, normally no changes are needed.

6.4.4.2 Data access method

- **Single file Mode, all data**: all data is from a single buffer file when programming, or stored in single buffer file when reading. , generally uses this mode by hard copy.
- **Single file, partitions mode:**   data is from a single buffer file (user loaded) when programming, or stored in the buffer file when reading. When programming or reading, the data is handled by partition.
    - PART: Select the partition to use
    - ST_BLK: Start block of partition
    - END_BLK: End block of partition
    - CNT_BLK: The number of data blocks in the file for current partition (the actual number of data blocks be processed)
    Note: The size of the partition (END_BLK-ST_BLK+1) must be larger than CNT_BLK when setting.
    The next partition memory cannot overlap with the previous partition.
- **Multiple files and partitions:** Partition mode is the same as the above, but each partition uses its own file, which does not use the buffer file.
    In mass production, the NAND partition mode is generally used for bad block skipping process, which can be set by user as needed, see Bad Block Management for details.

6.4.5 **Set OTP Area**

Some NAND has a special OTP area where data can be written, generally used to store the serial numbers.   fixed or unchanging data, etc.   OTP is the abbreviation of "One Time Programmable", that means one-time programmable. Once the data in this area has been programmed, it cannot be programmed again.

When reading the original IC, you can check this option, after reading, check whether the OTP data file data is completely empty. If there is data, OTP data must be written to the new NAND together when copying.

OTP data file: Disable or enable OTP

OTP pages: refers to the maximum number of pages in this area, only the smaller number of pages can be set.

6.4.6 **Spare Area in the File**

Due to the special nature of NAND FLASH, the data file is whether it includes the spare data.

If not included: the file will not include the spare area data

6.4.7 **Bad Block Manage Mode**

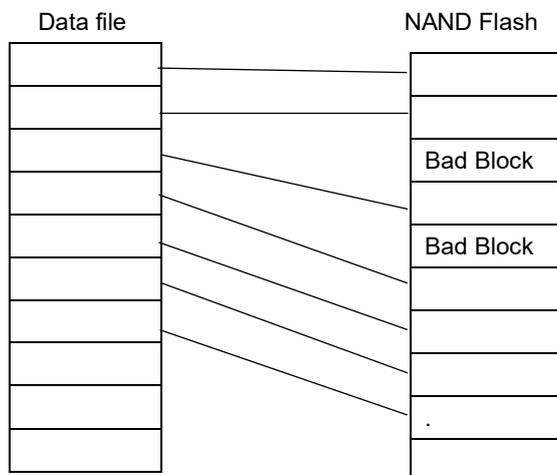3 manage modes for bad block

6.4.7.1 Skip Bad Block

If a bad block is found during reading or writing IC, the programmer will skip the bad block. And directly proceed to the next block. as below:

Data file   NAND Flash

6.4.7.2 Hard Copy

Force data reads or writes regardless of whether the block is marked with a bad block or not. If the verifying displays error during writing, programming will be stopped. This method is generally used for technical maintenance when replacing memory chips.

6.4.7.3 Replace Bad Block

Bad blocks are found to be written to the beginning of a specific block set,   that allows the system to logically feel that the memory is still a contiguous area, and is generally used in partitioning mode.

See detailed instructions in the example configuration as below. The number type in the start block (example: 900) is the actual replacement block index that is written to FLASH when the bad block is found.

6.4.7.4 Enable user-defined algorithm file

In the customized algorithm DLL file, before programming user can perform data processing or automatic sequence number calculation for any data in the block, as well as perform ECC algorithms, etc. It is also possible to auto-fill the BBT table with data according to the skipped bad blocks during programming.

For details, please contact us and we can customize it according to your requirements.

6.4.8 **Configuration Example 1 (MT29F1G08ABAEA)**

Operation method: **single file, partitions mode, skip bad block**

1) program NAND FLASH with partition
2) bad block processing mode: **skip bad block**
3) File data: file includes spare area data.
4) Use 2 partitions:

   The first partition: from 0# to199#, a total of 200 blocks, the actual length of the written data is 10 blocks

   The second partition: from 1000# to 1023#, a total of 24 blocks, the actual length of written data is 5 blocks

After the above settings, you can start programming. The results of the operation are shown in Table 1-2.

If the length of the buffer file is not enough, the programming is terminated.

If the buffer file length exceeds the length of the BLOCK (Total 15 blocks) data to be written, the programming is terminated.

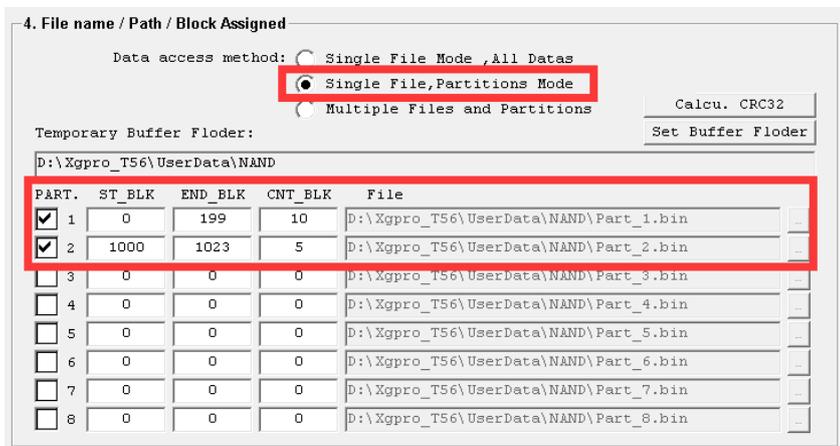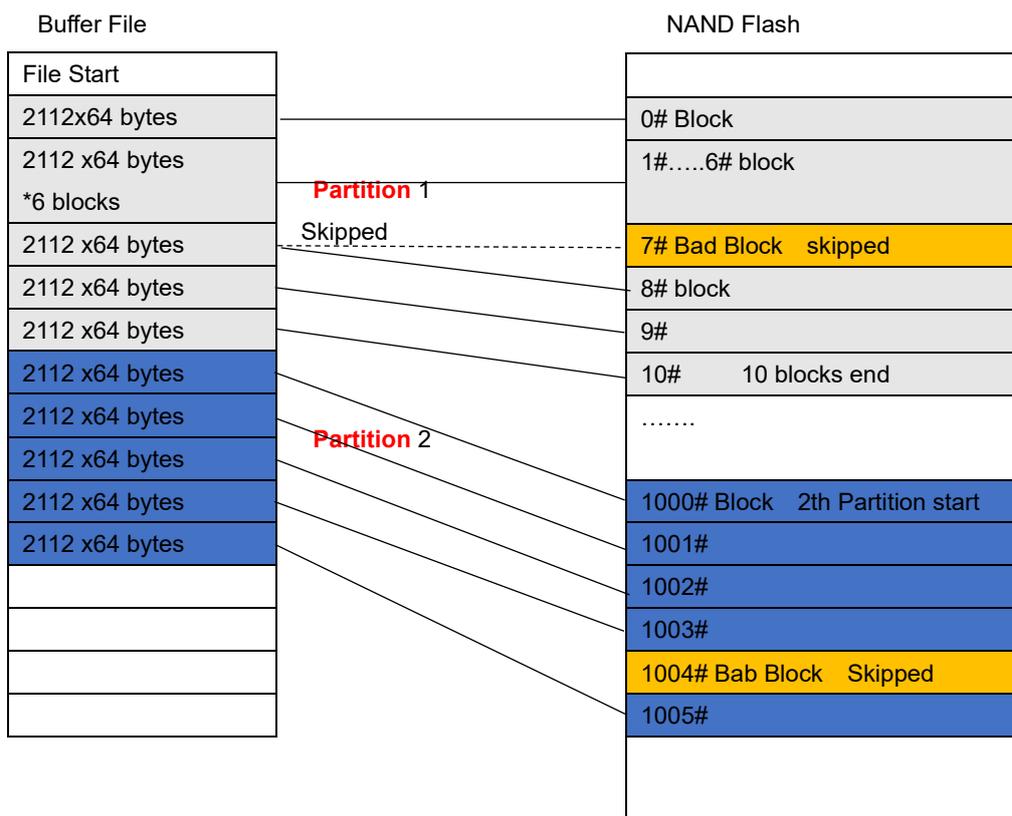The configuration interface is shown as below:

Table 1-2: Programming Process diagram

Notes:

In this configuration example, when reading or verifying, it is also done by partition

    1. Read 10 Blocks of partition 1 and save it to the file

    2. Read 5 Blocks of partition 2 and save them to the file

    3. Read finish.

By setting partitions flexibly, you can individually read or write data of one or more blocks.

6.4.9 **Configuration Example 2 (K9F1208U0C)**

Operation method: multiple files, partitions mode, replace bad blocks

K9F1208U0C parameter:

IC capacity: (512+16)*32*4096 blocks

Page Size: 512 bytes

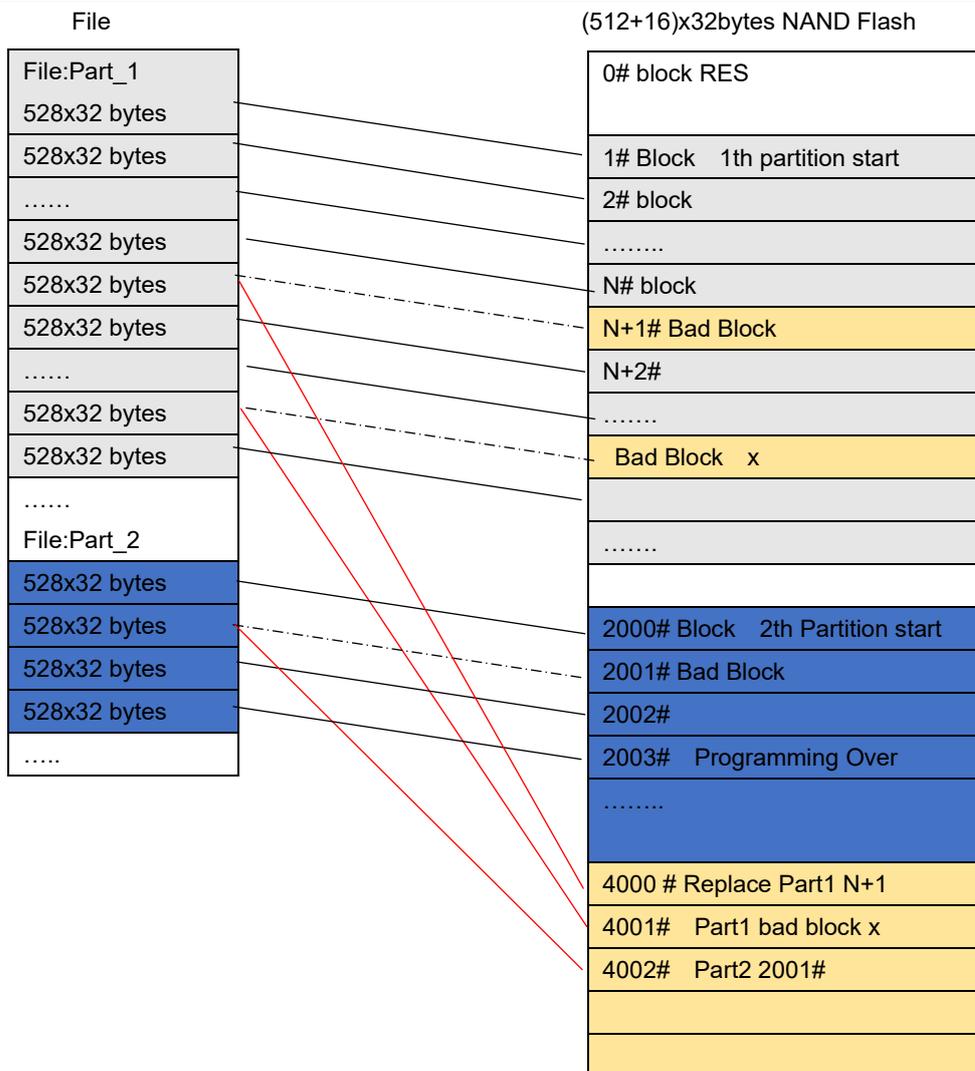Spare size: 16 bytes

Pages Per Block : 32 pages

Block Size : 4096

1) Use partition table to write NAND FLASH
2) Bad block handle mode: replace block, if bad blocks are found, write to block 4000# start position
3) File data: includes spare area
4) Use 2 partitions,

The first partition: from 1# to 1000#, a total of 1000 blocks, the actual length of the written data is 100 blocks

The second partition: from 2000# to 3999#, a total of 1000 blocks, the actual length of written data is 3 blocks

5) The written data file is: PART_1.BIN Partition 1

PART_2.BIN Partition 2.

6) Programming process is shown in Table 1-3.

File		(512+16)x32bytes NAND Flash
File:Part_1 528x32 bytes		0# block RES
528x32 bytes		1# Block 1th partition start
……		2# block
528x32 bytes		……..
528x32 bytes		N# block
528x32 bytes		N+1# Bad Block
……		N+2#
528x32 bytes		…….
528x32 bytes		Bad Block x
……		
File:Part_2		…….
528x32 bytes		
528x32 bytes		2000# Block 2th Partition start
528x32 bytes		2001# Bad Block
528x32 bytes		2002#
…..		2003# Programming Over
		……..
		4000 # Replace Part1 N+1
		4001# Part1 bad block x
		4002# Part2 2001#

6.5 Program NAND through ISP

ISP programming only supports SPI NAND IC, select X1 mode, check option <ICSP port> to program. ISP programming process is no different from using adapter. Do not separate the three wires, while connect the two GND lines and CLK clock lines.

6.6 Save project file

For all configuration parameters, you can "Save Project" to project file. When use it again, just open the project file and you can directly program. It is suitable for mass production.

6.7 NAND Flash Copy From Original Chip

6.7.1 **Notes:**

In repairing equipment, we often need to read out data from an original chip, copy it to another new chip, and then solder it to the board. To ensure that the chip can work properly, users need to pay attention to the following points:

1.   Copy chip configuration: Use the default settings of the programmer software.

Option settings: must use hard copy, the data in the file must include spare area, all other options are not checked by default. When reading by this mode, the programmer can read the entire contents of the chip (including the contents of the bad block) to the file.

2.   Some NAND FLASH chips have the unique ID. Embedded system applications may read the unique ID of the chip and perform encryption operations in the program, because the unique ID is set by the chip manufactory at the factory and cannot be changed or copied. In this case, even if you have copied all the contents of the chip correctly (including the OTP contents), it will not work properly after replacing with the new chip.

For example: MT29F4G08ABAEA has the unique ID (16+16 Bytes) , which can only be read, cannot be rewritten.

If the application is encrypted by this method,   it cannot be replaced with the new chip unless you can modify the software of the embedded system.

3.   OTP Area (OTP One-Time Programmable Storage Area)

OTP Area only exists on some NAND Flash chips, please check the IC data sheet for details.

For example: MT29F4G08ABAEA has the OTP data of 30 pages (30Pages*2112 bytes). When reading, you should read the OTP content at the same time. Check whether the OTP file is completely empty (FF), if there is OTP data, you need to write OTP at the same time when programming.
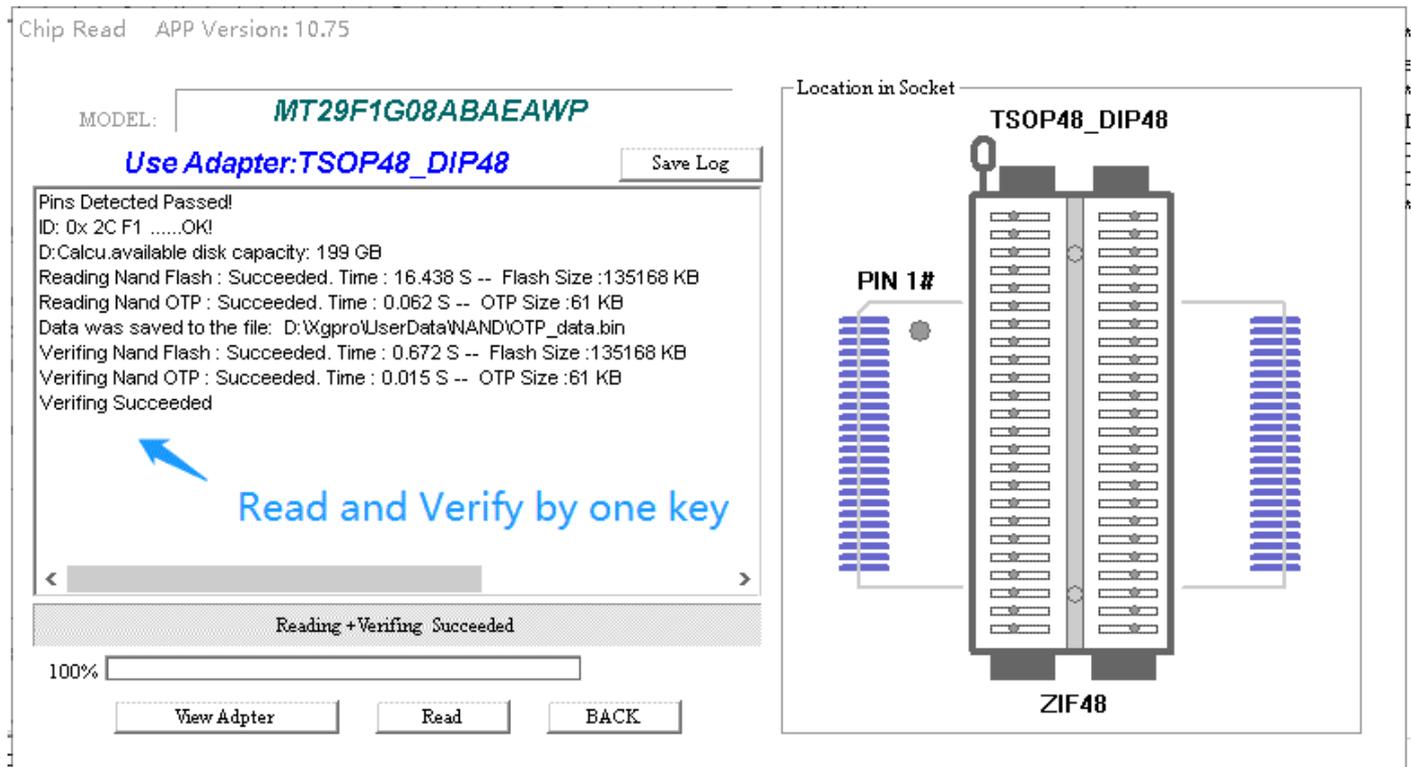
Read the OTP content, as shown below:

6.7.2 **Configuration and options**

Default settings: hard copy, single file mode all data, Includes Spare Area (OOB)

6.7.3 Read the original chip and save

Click [Read] button in the toolbar to start reading the chip, as follows:

```
Chip Read    APP Version: 10.75

        MODEL:        MT29F1G08ABAEAWP          ┌─ Location in Socket ─
                                                │       TSOP48_DIP48
        Use Adapter:TSOP48_DIP48    Save Log

Pins Detected Passed!
ID: 0x 2C F1 ......OK!
D:Calcu.available disk capacity: 199 GB
Reading Nand Flash : Succeeded. Time : 16.438 S -- Flash Size :135168 KB
Reading Nand OTP : Succeeded. Time : 0.062 S -- OTP Size :61 KB      PIN 1#
Data was saved to the file:  D:\Xgpro\UserData\NAND\OTP_data.bin
Verifing Nand Flash : Succeeded. Time : 0.672 S -- Flash Size :135168 KB
Verifing Nand OTP : Succeeded. Time : 0.015 S --  OTP Size :61 KB
Verifing Succeeded

              Read and Verify by one key

    <                                    >
                 Reading +Verifing  Succeeded
                                                       ZIF48
100% [                                    ]

    [ View Adpter ]    [ Read ]    [ BACK ]
```
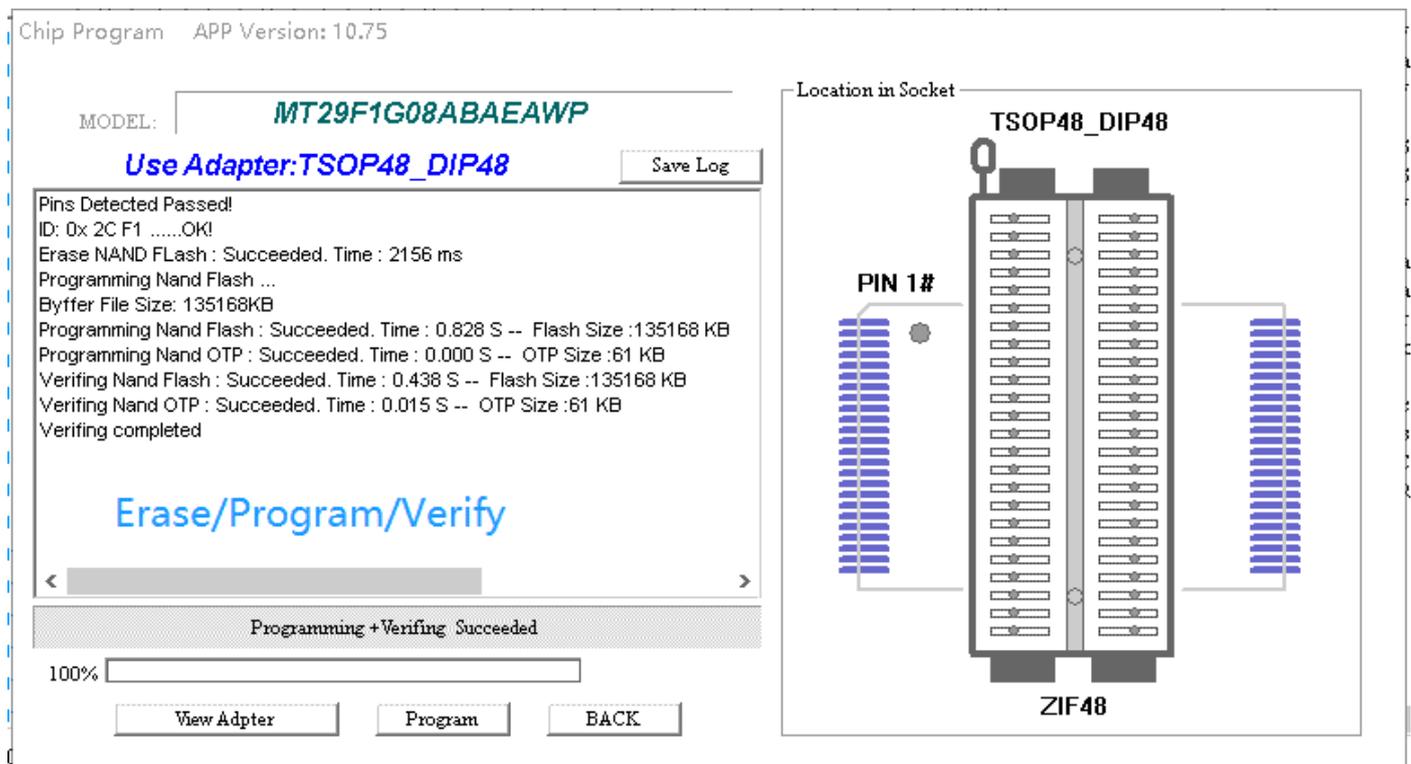
Exit the Read dialog box, Click **[Save]** button in the toolbar to save flash to a new file

6.7.4 6. Copy new chip

Selecting correct IC model, and **[Load]** the data file

Put on the new chip to the socket, click **[PROG.]** button in the toolbar, the following interface will pop up: click the **[PROG.]** button to start programming.

```
Chip Program    APP Version: 10.75

        MODEL:        MT29F1G08ABAEAWP          ┌─ Location in Socket ─
                                                │       TSOP48_DIP48
        Use Adapter:TSOP48_DIP48    Save Log

Pins Detected Passed!
ID: 0x 2C F1 ......OK!
Erase NAND FLash : Succeeded. Time : 2156 ms
Programming Nand Flash ...
Byffer File Size: 135168KB                               PIN 1#
Programming Nand Flash : Succeeded. Time : 0.828 S -- Flash Size :135168 KB
Programming Nand OTP : Succeeded. Time : 0.000 S --  OTP Size :61 KB
Verifing Nand Flash : Succeeded. Time : 0.438 S -- Flash Size :135168 KB
Verifing Nand OTP : Succeeded. Time : 0.015 S --  OTP Size :61 KB
Verifing completed

         Erase/Program/Verify

    <                                    >
                 Programming +Verifing  Succeeded
                                                       ZIF48
100% [                                    ]

    [ View Adpter ]    [ Program ]    [ BACK ]
```

Notes:

1. It is possible that the speed of programming the chip is faster than reading, because the software automatically skips the content of the empty data during programming.

2. In the process of reading, erasing, or programming, if chip has bad blocks, the tip will appear, but it does not affect the programming of the chip.

3. It is not necessary to check the bad block before copying. As long as the verifying is correct and there is no error when programming, the new copied chip will be available.   If there is a bad block in the new chip, the software will skip writing if the bad block is empty data when copying. If the new chip has a bad block, and the original chip has data to be written to the location of this bad block, the data can't be written into this bad block, the programming error will occur. Please replace with another new chip, (the location of the new chip bad block is generally random).

4. Repairers usually do not know how to deal with bad blocks, so only copy directly with hard copy mode.

6.8 NAND internal ECC setting

Notes: (ECC option is disabled for chips without ECC function)

1. All ECC is enabled by default, if you want, you can uncheck the option.
   Enable Internal ECC is the best choice, high reliability, the data read every time is the same, there is no bit flipping.

2. Enable ECC Read +   Enable ECC Write, (same reliability as the original chip)

3. Enable ECC Read + Disable ECC write   (data may be flipped, less reliable than original chip data)

4. Disable ECC Read + Enable ECC write **(don't use this method)**

5. Disable ECC Reading + Disable ECC writing, the probability of flipping is twice as high as the third method, and the possibility of error is greater, and the reliability is worse than that of the original chip data.

6. Chip has internal ECC correction, but the device does not use internal ECC correction, Must disable internal ECC when reading or writing. This situation should be rare, if reading and writing via enabled ECC, the chip does not work normally, test with [disable ECC read + disable ECC write]. ECC function switching is shown in the following figure:
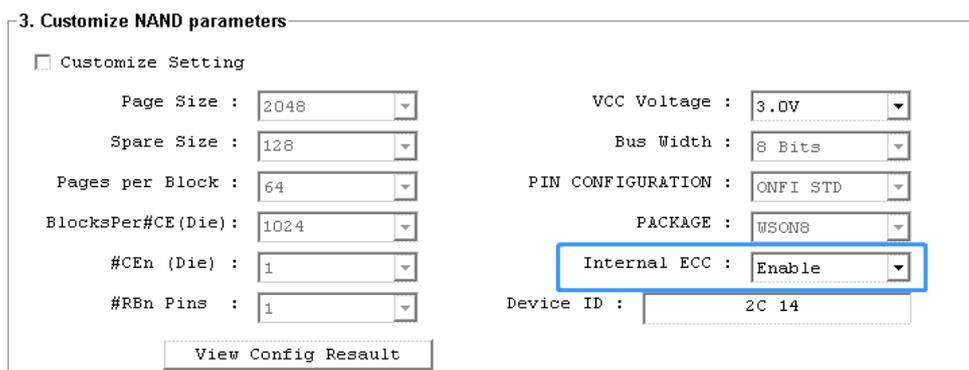
6.9 Calculation of NAND file address

Calculated in two cases:

- File includes Spare Area

  The location of the data in the file = n Block x (Page Size + Spare Size) x the number of pages per block

- File does not include Spare Area

  Location of data in the file = n Block x Page Size x the number of pages per block

7. VGA_HDMI ISP and Tools

7.1 VGA_HDMI ISP wiring diagram

Use the VGA_HDMI adapter to connect a VGA cable or HDMI cable, or ISP cable to the TV motherboard. When in-circuit programming, the TV motherboard is normally required to be in standby mode and to use external power supply.

General wiring diagram as below:

According to the requirements of the motherboard, VGA in-circuit programming can be used in a variety of ways to connect. T76 can automatically recognize the hardware connection, do not need to make any settings in the software.
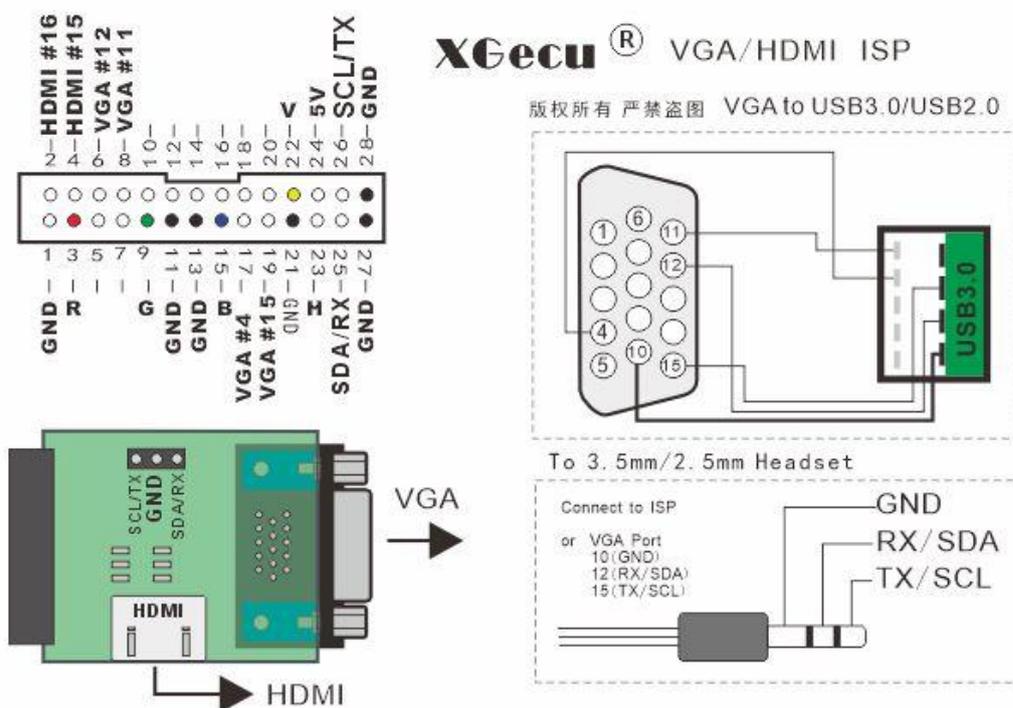
Connection photo via VGA cable
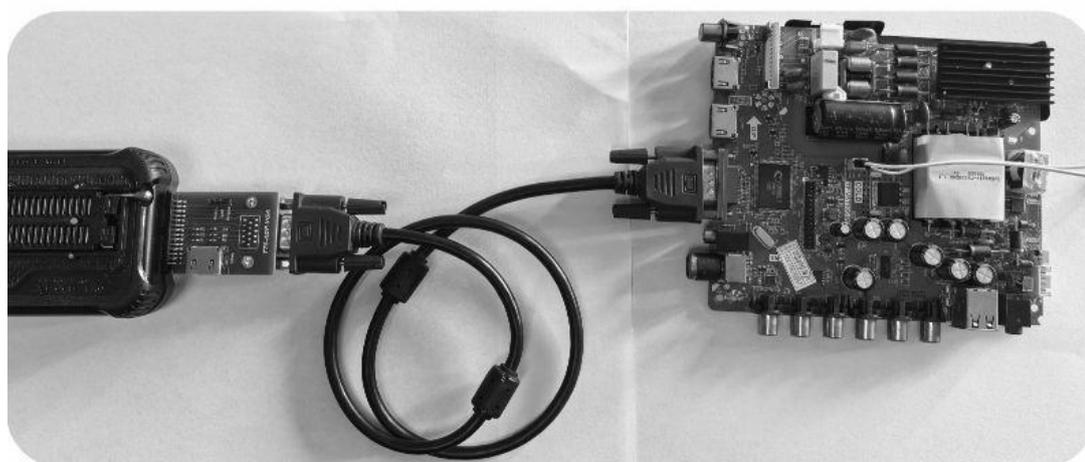
7.2 VGA_HDMI in-circuit programming

Note: VGA in-circuit programming 25 FLASH, currently only support **MSTAR** part of the motherboard,

Connected to the motherboard, power supply to the motherboard, keep TV in standby state

7.2.1 **Auto-identify Chip**

Click **[AUTO]** button, the dialog box will pop up, select **VGA _HDMI_ISP** port.

Click **[Detect]** button to automatically find the chip model and select the chip

7.2.2 **In-circuit Read chip data**

Click **[Read]** button, the read dialog box pops up, click **[Read]**

When reading, it will automatically find the motherboard communication port and protocol to enter SPI mode.

7.2.3 **VGA in-circuit programming**

Click **[LOAD]** in the tool bar to load the data file to be written, as shown in the following figure:
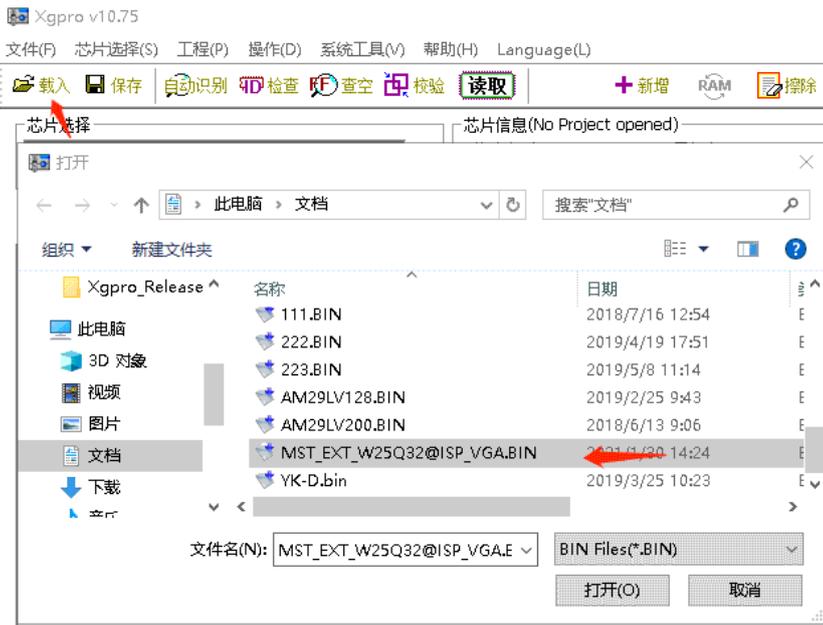
After loading the file, click **[PROG.]** in the toolbar to bring up the programming dialog box. Click **[Program]** to start programming

7.3 EDID Programming

Connect TV or monitor, click **[Select IC]** button, in the search device, type in **"EDID"**.

Select **EDID_256B @ISP_VGA**, as shown below:
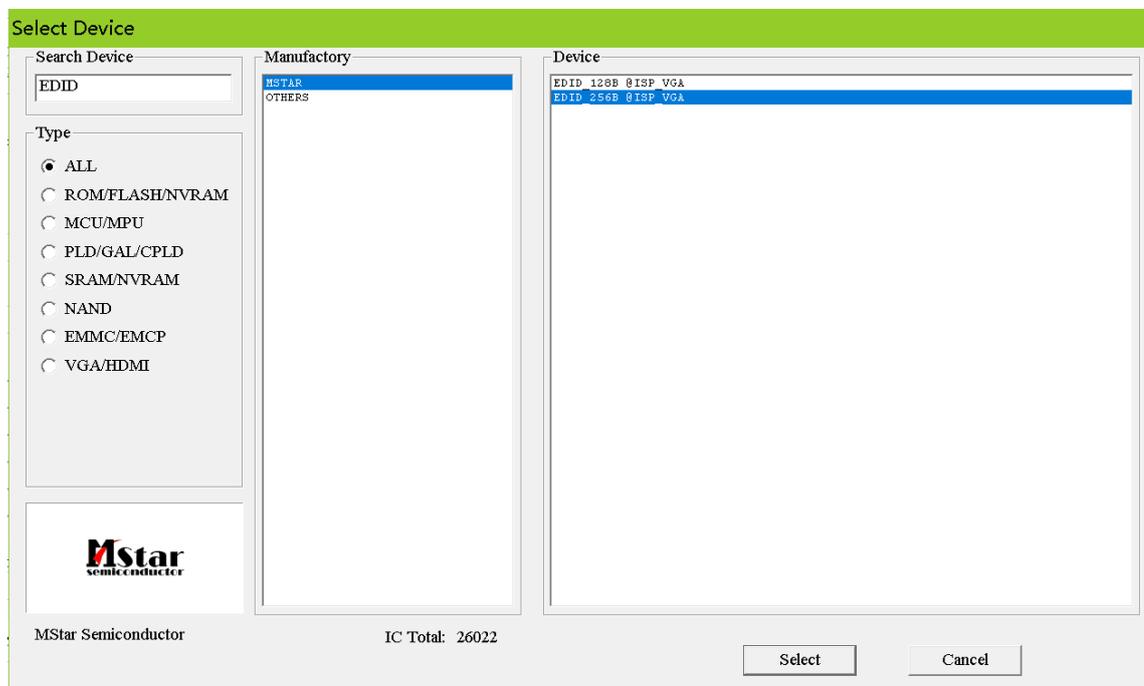
After selecting, you can read, modify, program and verify the EDID.

The operation method is exactly the same as ordinary FLASH chips, so not describe here.

7.4 TV Tools: Serial Printing

Connect the TV board, click the icon in the upper right corner

Click the **[Monitor]** button and start　to receive power-on diagnostic information from the TV board.

Turn on the TV power switch. The power-on information will be displayed on the screen, you can click **[Save Log]** to save.

The line sequence is auto found by default, do not need to select it manually.

Baud rate is 115200bps by default, please switch the baud rate if there is a messy code, as follows:

Click **[STOP]** button to stop receiving messages.

7.5 TV tools: read monitor or TV parameters

Connect TV or monitor, power standby state.

Click on the icon in the upper right corner, as below,

click on **[Read Monitor/TV Parameter]**

Show as follow:

7.6 TV tools: HDMI, VGA video test signal

Video test signal connection, you must use the T76-ADP-VGA adapter to connect VGA

Click the icon 　[TV]　 in the upper right corner of the window, click **[Video Signal Start]** button, various kinds of VGA signal will be out.

Click **[STOP]** and back

8. Automatic Identification

The automatic-identificatiom is for the SPI 25 series flash. Realize fast selection of 25 nor/NAND flash chips.

Identification interface: 8-pin and 16-pin chip on the ZIF socket

8.1 ZIF Socket Identification

Click the **[AUTO]** icon 　[AUTO]　 button in the toolbar, as shown below:
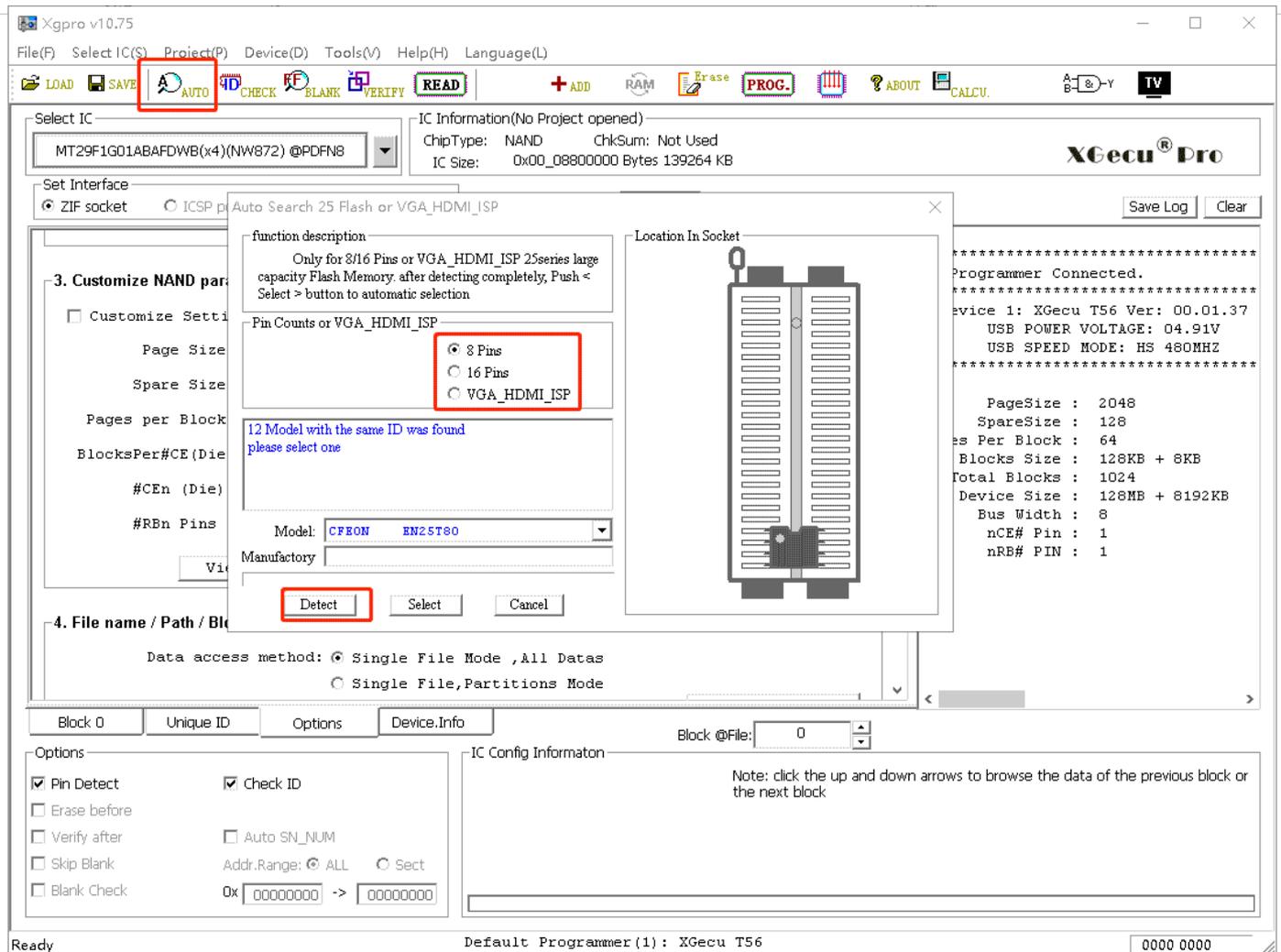
- Select 8-pin or 16-pin chip

- Click **[Detect],** if the chip is recognized, the list of chip with the same ID will be displayed, select the appropriate one

- Click **[Select]** button in the dialog box,　selecting IC is finished, next to reading or programming

8.2 VGA_HDMI In-circuit Identification

Click **[AUTO]** icon ⬚ button in the toolbar to check "the VGA_HDMI ISP" interface.

For details, see: Chapter 7 VGA_HDMI ISP in-circuit programming.

9. Digital Logic IC Testing

Functions:

- Test 54/74 series, CD4xxx digital logic chips, test results positioned to the pin. Adjustable VCC voltage.
- Automatically identify the logic chip.
- User-defined test vectors.

9.1 Test Logic Chip

Click **[Logic Test]** icon ⬚ in the toolbar to bring up the test dialog box, as below:
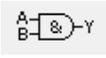
- After selecting IC model, directly click **[TEST]** button to see the test results of the chip vector
- VCC voltage is variable
- Click the **[NEW]** or **[COPY]** button to add a new customized chip
- The customized chips can be imported or exported for sharing. You can also delete or modify.

9.2 Customize the logic chip

Click **[NEW]** or **[COPY]** button to edit the logic test vector table,   As figure shows:
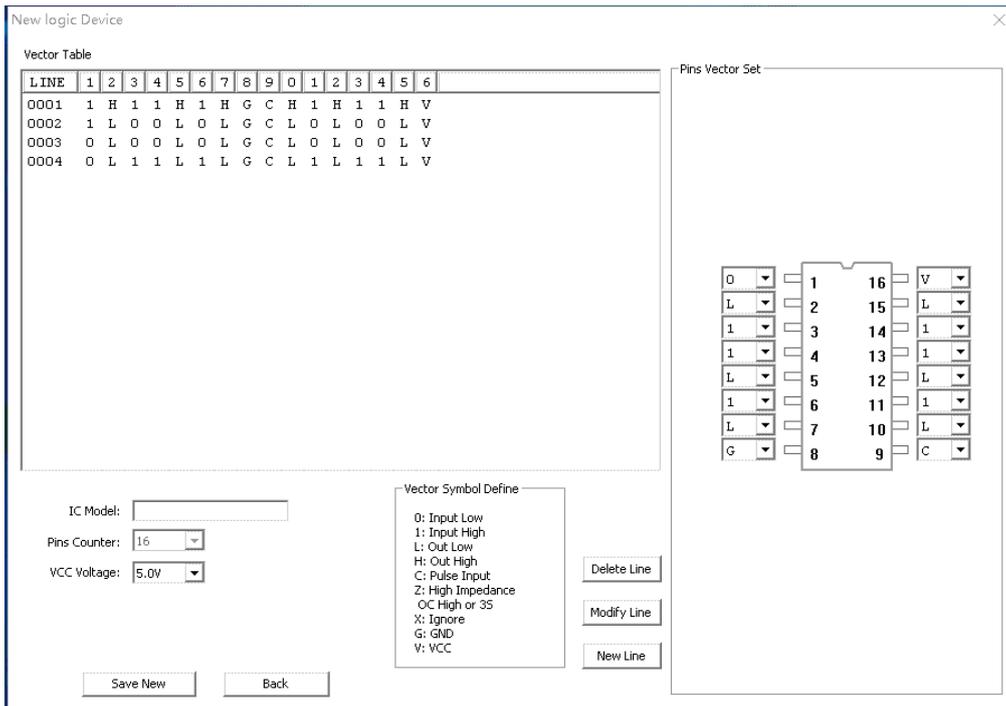
Modify and add the test vector table. then type in the new chip model and save the new.

9.3 Logic chip identification

Put the chip in the ZIF socket of the programmer and click **[Auto Find]** , the logic matching chip model will be listed automatically, as below:
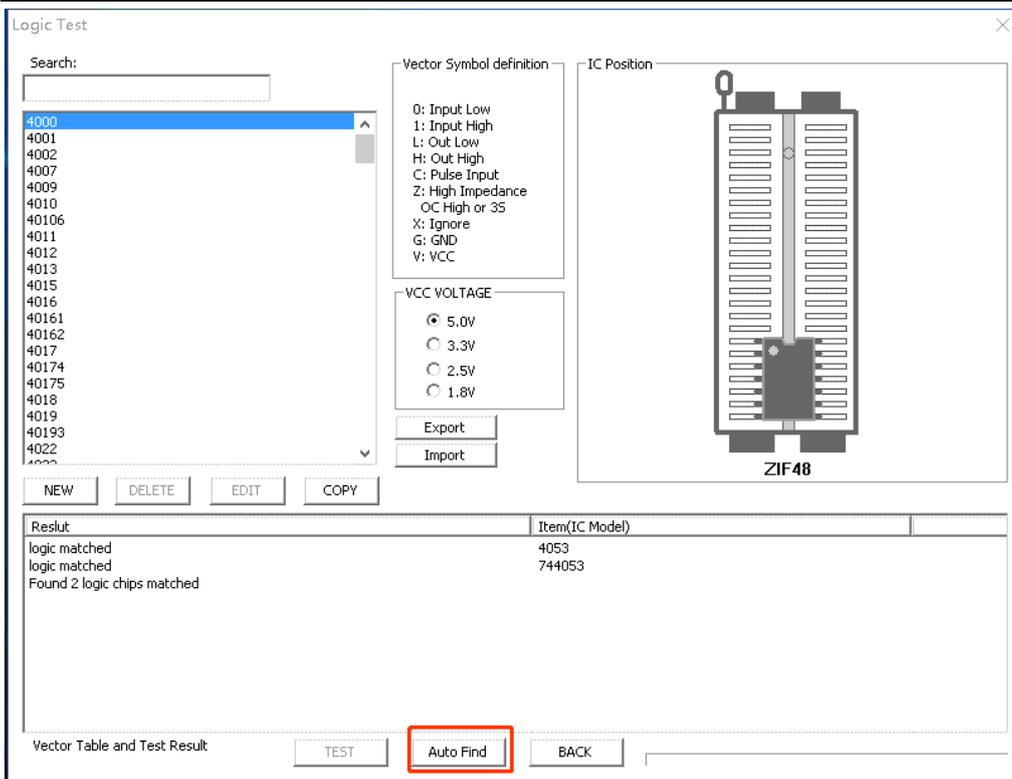
10. Add customized chips[FAVORITES]

● For chips that are not in the software support list, there are many chips that may have programming algorithms fully compatible with one of the chips in the list. In this case, you can add them to the custom list after the test is normal, so that you can easily use them later or export the list to share.

● For chips that the software support parameter setting, such as NAND, 27C and other series, various parameters can be set by user. The user can change the parameter settings, add it to user list after getting the programming test to pass.

● New customized chips can also be used as favorites of commonly used chips.

10.1 The common method of adding chips

The same type of chips may be the same programming algorithms exactly . But in general due to different manufactories, the identification ID of the chip is different, at this time, just select the compatible chip model, uncheck the option **[check ID]** to read and write for test.

10.1.1 **Test algorithm**

For example: XM25QH32B (assume, this chip is not in the support list)

We can select the same type of chip W25Q32BV for testing, as below:
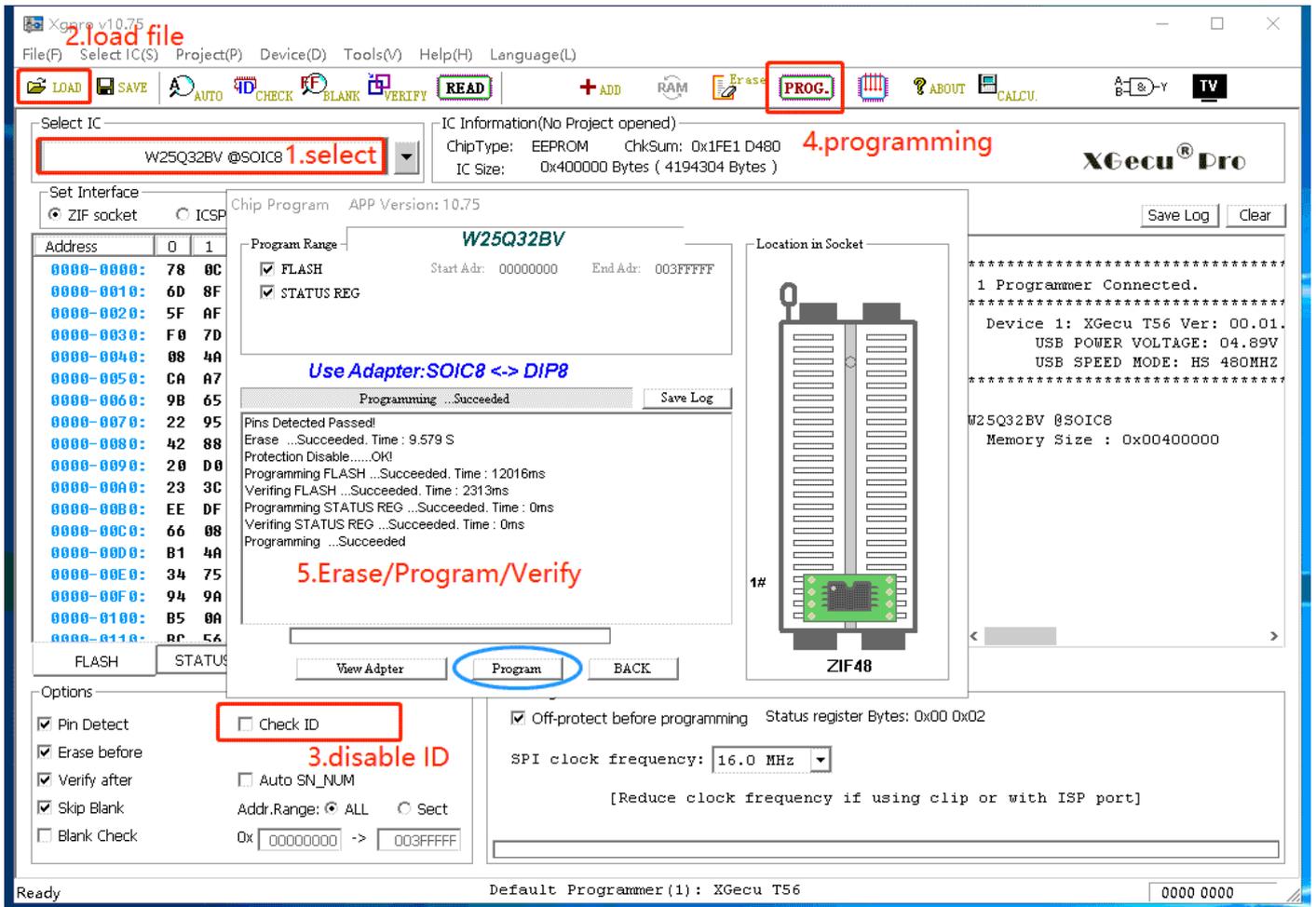
As the above, the test is completed, that means the chip is programmed correctly by selecting W25Q32B.

STEP:

1) Select IC W25Q32BV

2) Load a data file

3) Uncheck the option **[check ID]**

4) program for testing

5) Erase, write and verify completed, if testing is correct, you can go to the next step. If it failed, you can select another similar chips for testing.

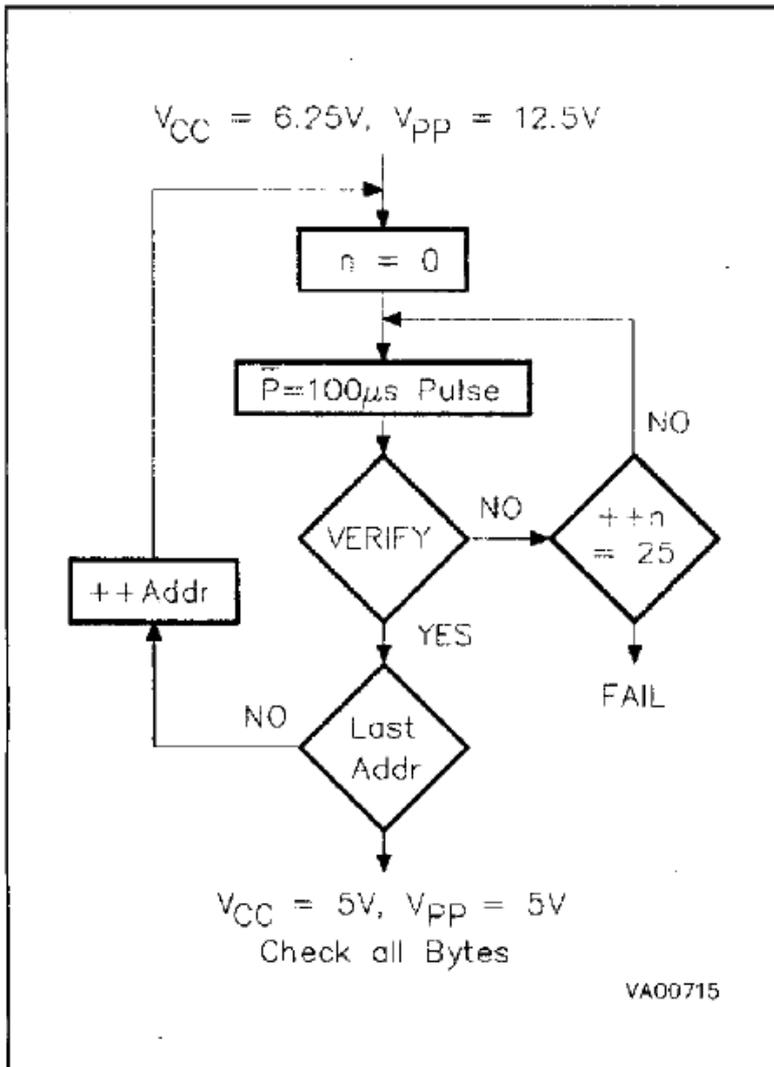## 10.1.2 **Read ID and add the chip model**

As shown below:
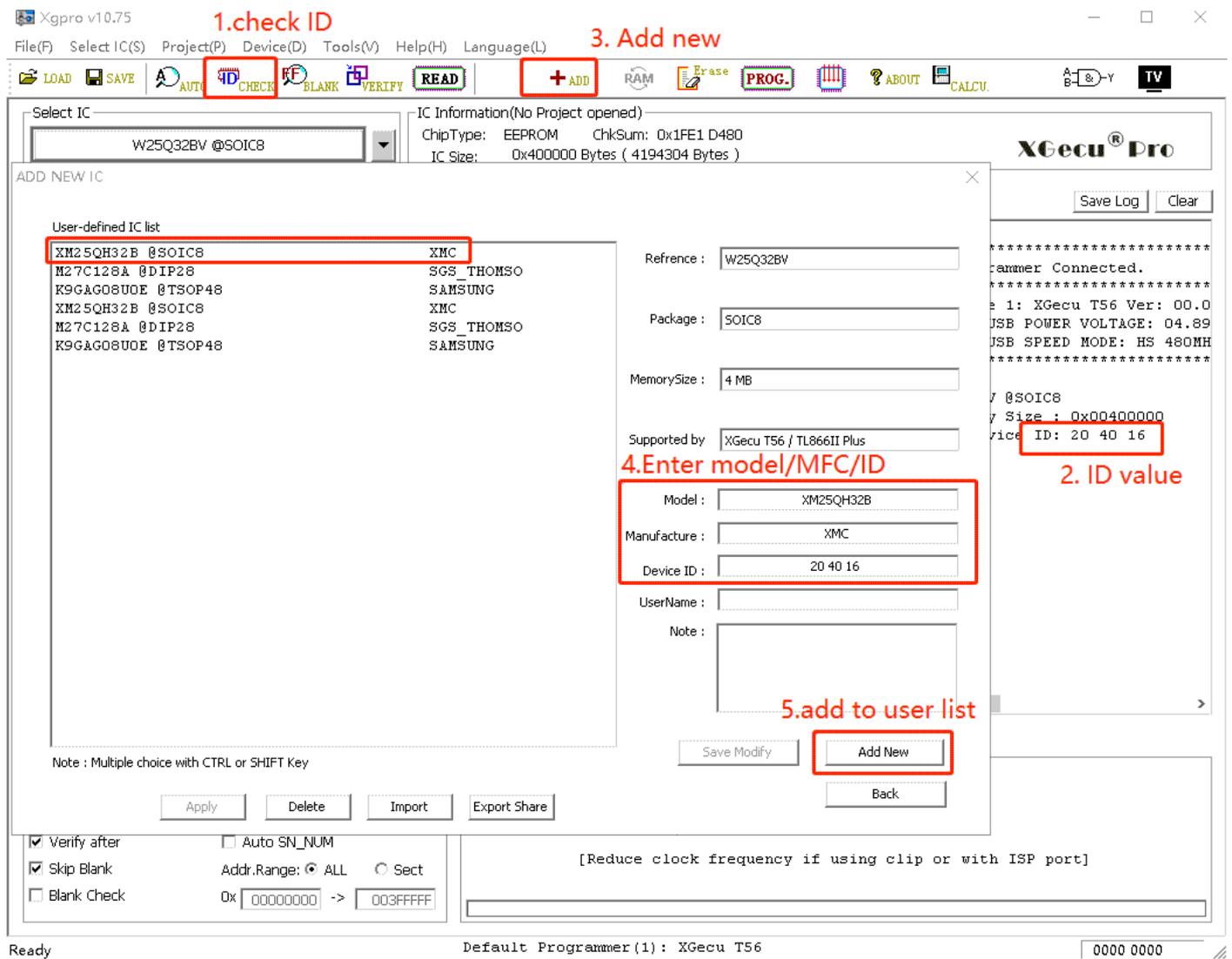
STEP:

1) Click **[Check ID]** to read the current chip ID

2) Display the current ID value

3) Click **[ADD]** in the toolbar

4) Type in the chip model, manufactory, the chip ID read, and other information

5) Click **[Add New]**, a new model will be added in the user list

6) Just double click the chip model in the custom list to use it.

10.2 Add 27 series device

Example: M27C128A is not in the supported list, the basic method of adding chips is the same as before, more reliable method of adding is as below:

1. Select IC: Type in "27C128" to find, select AM27C128 @DIP28 Note: the chip package and chip capacity must be the same.
2. Download the M27C128A IC data set from the Internet
   From the data sheet, you can find:
   Chip ID: 20 0A, VCC=6.25V when programming, VCC=5V when verifying and reading, VPP=12.50V for programming, pulse time is 100us

Figure 7. Programming Flowchart

$V_{CC} = 6.25V, V_{PP} = 12.5V$

n = 0

$\overline{P} = 100\mu s$ Pulse

VERIFY

++Addr

++n = 25

Last Addr

FAIL

$V_{CC} = 5V, V_{PP} = 5V$
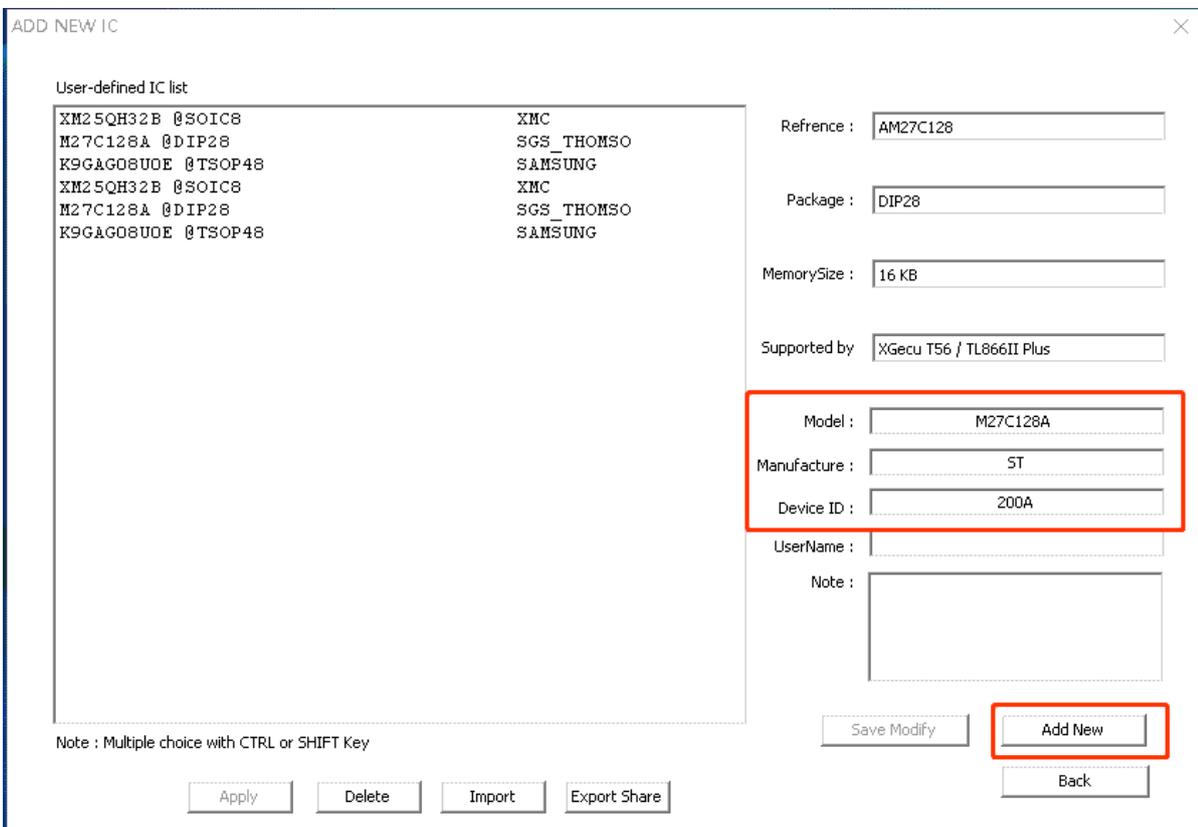Check all Bytes

VA00715

3.   According to the parameters in data sheet, modify the chip parameters as shown below:
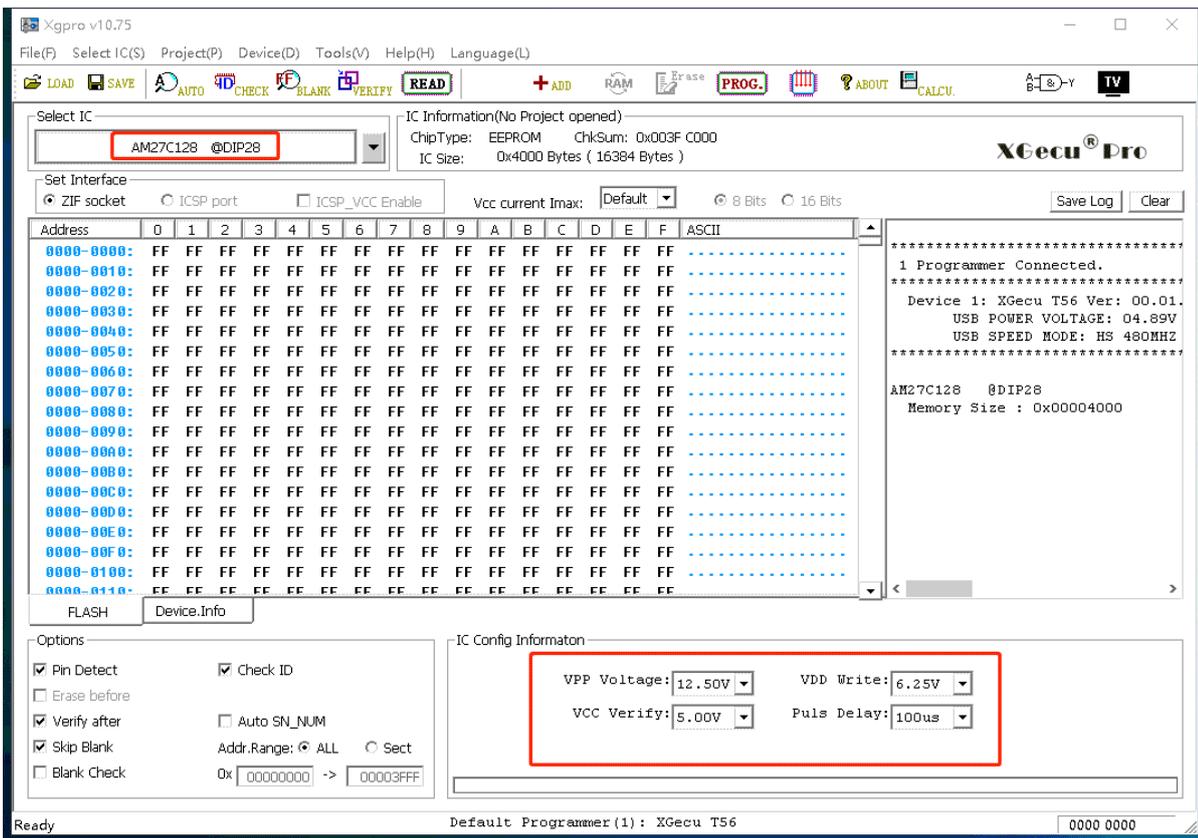
4. Click **[ADD]** button in the toolbar

In the following position, type in the chip model: M27C128A Manufactory: SGS-THOMSON Chip ID: 20 0A, as shown below:

click **[Add new]** button


5. Double click the new chip to apply using it.

10.3 Add NAND chips

For NAND chips, theoretically you can customize all NAND chips. For ONFI standard chips, you can also automatically identify and add to the custom list. For non-ONFI standard chips, you can add it according to the IC data sheet.

10.3.1 Select a NAND chip of any type with the same package

K9GAG08U0E TSOP48 package chip as an example to add custom.

Arbitrarily select a K91G08U0M @TSOP48 (you can select any other TSOP48 NAND chip)
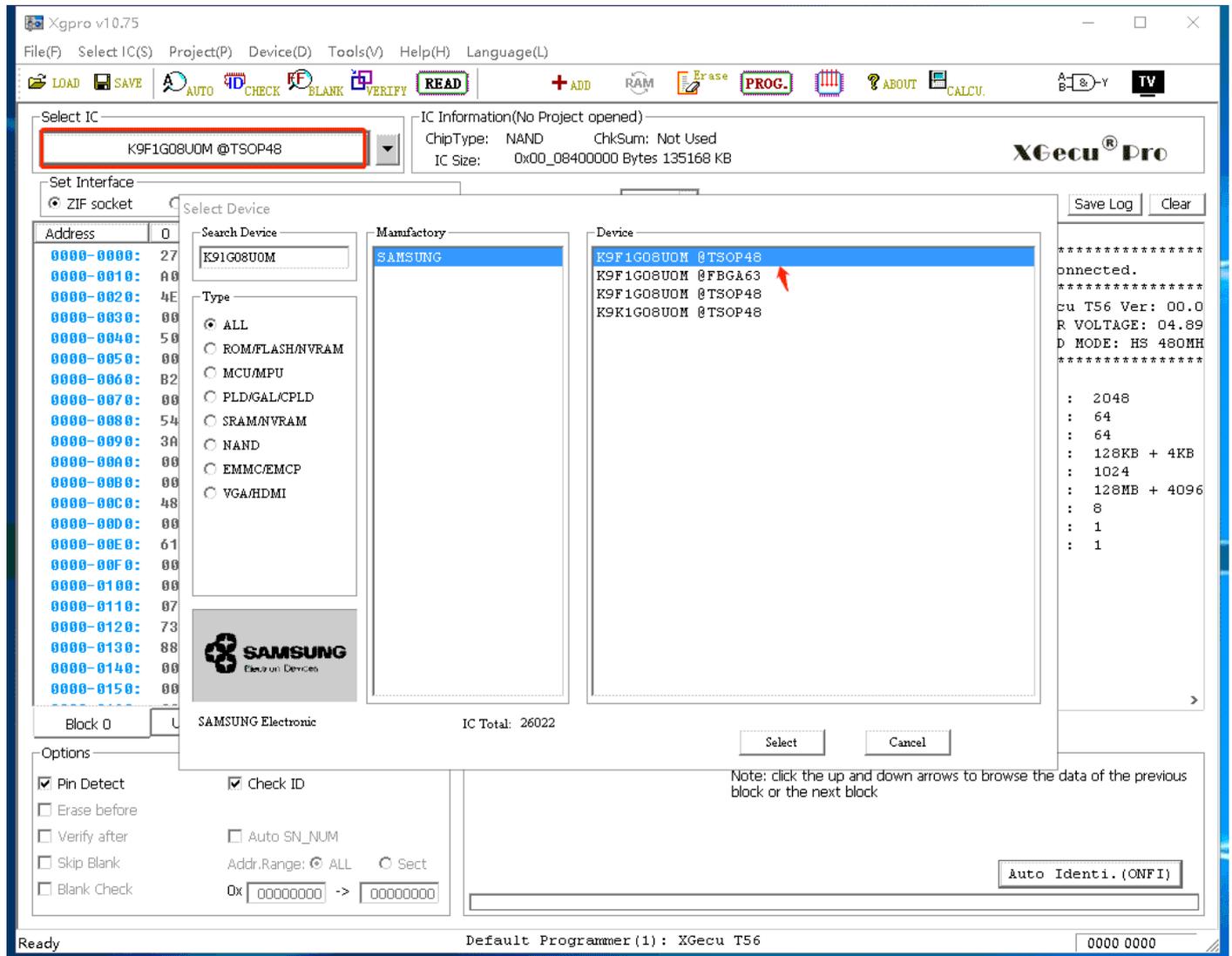
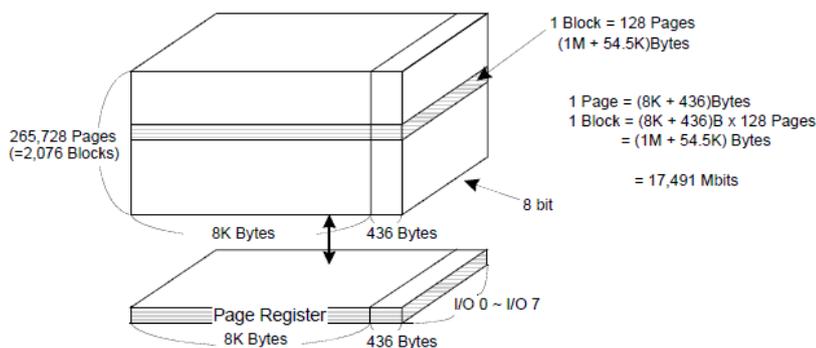## 10.3.2 Check IC data sheet to find the corresponding parameters

Figure 2. K9GAG08U0E Array Organization

68

From the above figure you can see,
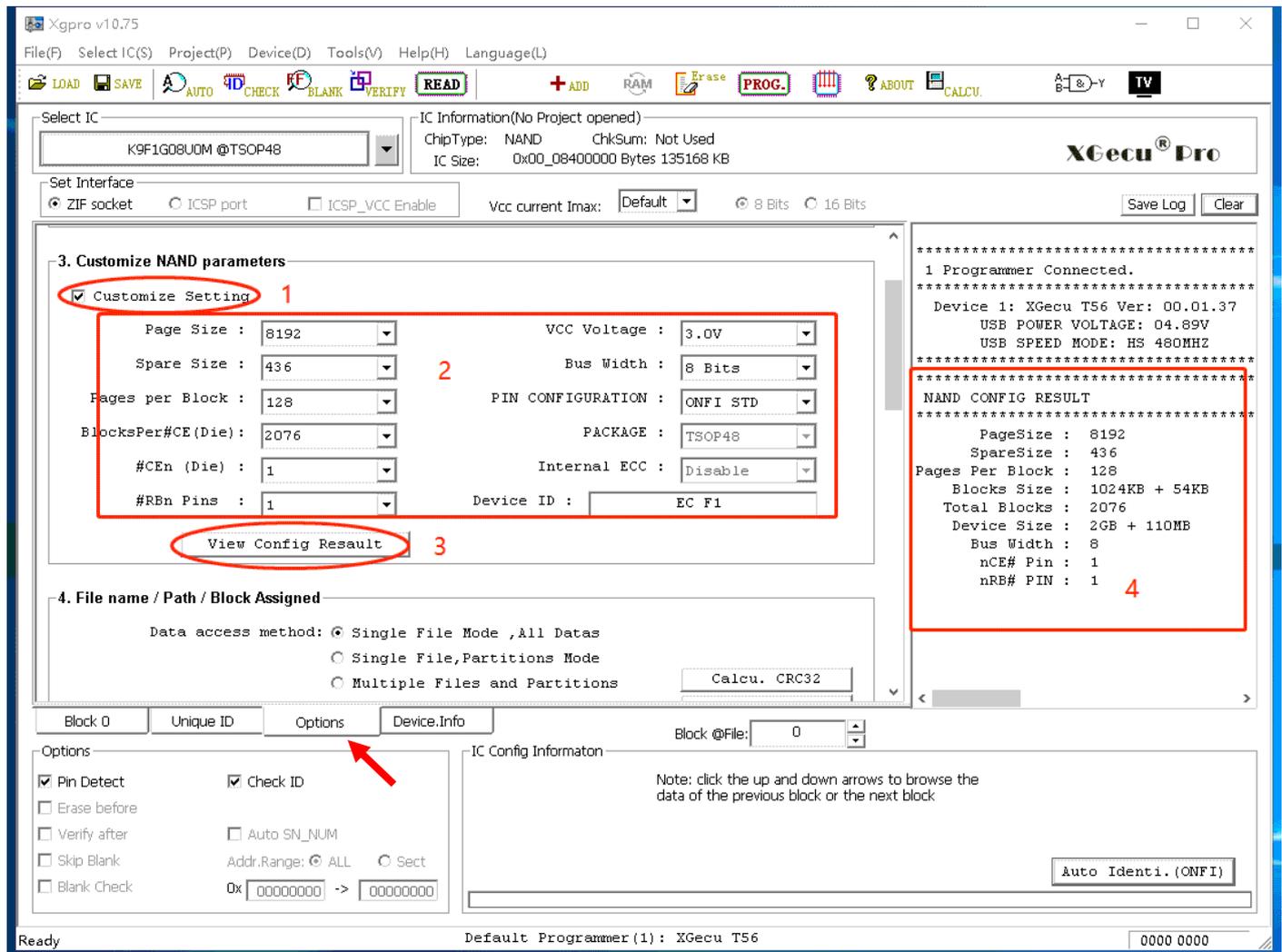
K9GAG08U0E Parameter:

Page Size 8K (8192) Spare Size=436 Number of pages per block: 128 pages, total number of blocks 2076 blocks

In addition, from data sheet you can find, the operating voltage is 3.3V 1CE pin, 1 RB pin, 8-bit data width

ID: EC D5 84 72 (only read the first 4 bytes)

10.3.3 Set NAND parameters

Click [Options] to set as below **(3. Customize NAND parameters)**:

1. Check option **[customize setting]**
2. Set value parameter according to IC data sheet
3. Click **[View Config Result]** and you can view the final result in the right information column **[4]**.

10.3.3 Add to User List

After the parameters are set, add new list in the same way as before, click the **[ADD]** button ![+ADD] in the toolbar to enter the ADD NEW IC dialog box, as below:
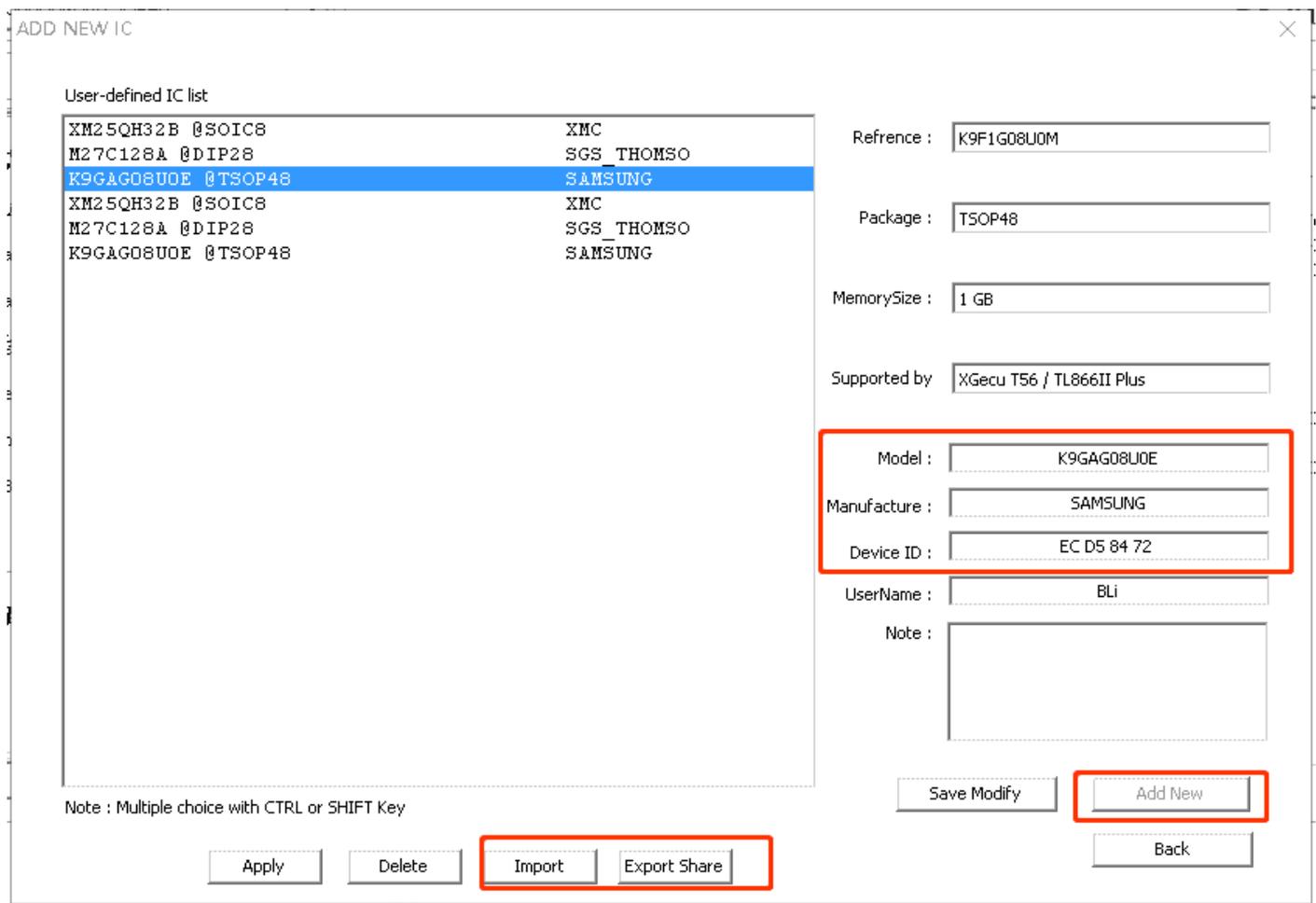
Respectively type in the IC name, manufacturer and device ID. And click **[Add New]** to add to the list.

Double click the added model, you can program the IC. If the read/write is normal, it means the addition is correct.

10.4 Import and Export Share

● For customized chips, user can export the database for sharing, just send the exported data file to the third party, who uses the import function to import the list.

● Export data, use CTRL or SHIFT key + left mouse button to multi-select and export multiple chips at one time. The extension of exported file is xxx.ULI.

10.5 Commonly-used chip favorites

For commonly-used chips, there is a drop-down list of the last 10 chips to the right of the **[Select IC button]**:
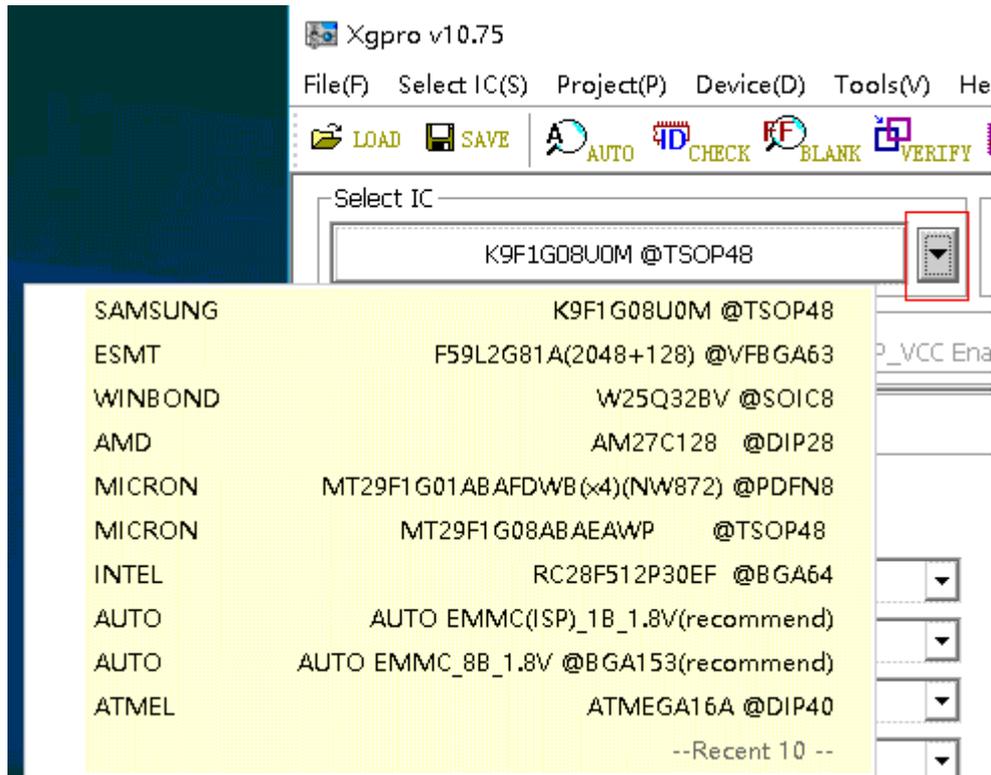
If it is not enough to use, you can add to this custom list.

Selection method:

After selecting IC, directly click **[+ADD]** button in the toolbar, without modifying the content, directly add to the custom list. Next time, user can select the IC in the custom list. User can select the IC model more quickly.

11. Appendix

11.1 Disclaimer

(1) Products that have been repaired, modified by any organization or individual not officially authorized by us are not covered by the warranty.

(2) The ZIF socket of the product and other man-made damages are not covered by the warranty.

(3) Joint and several extended liability caused by hardware failure or software defects.

Special Declare

Due to the variety of chips supported by the programmer and the complexity of the software system, errors or defects are inevitable. In mass production, especially for OTP one-time programming chips, you must make sure that the programming is correct before mass production. We are not responsible for collateral expansion losses due to hardware failure or software defects. If you find any problems or have valuable opinions about our products, please contact us.

11.2 Warranty

Within one year from the date of purchase, if there is a failure during normal operations, can repair for free.

11.3 Contact Us

As the software is constantly updated, the manual content is for reference only and is subject to the product application. The latest version software can be downloaded from http://www.xgecu.com/en.

Name: Haikou Xingong Electronic Co.,Ltd.
Address: 4th Floor, Jingrun Hotel, No.80, Qiuhai Avenue West,
Haikou, Hainan Province, China
EMIAL: 532007590@QQ.com
TEL: 0898-68681816

11.4 FAQ (Frequently Asked Questions)

1) Device unavailable or error

It is usually because the USB driver is not installed, or the driver is prohibited by antivirus software, at this time the RUN light on the USB programmer flashes rapidly.
Solution: Refer to section 2.1 and re-install the application software.

2) Error when updating firmware, unable to refresh

It is usually caused by using a USB hub. Please do not use an external HUB when refreshing. Please directly connect the programmer to the USB port of your computer to refresh. Restart the computer to test. If it still cannot solve the problem, please try to do the refresh operation on another computer.

3) Poor pin contact

Situation 1: Old chips, often there are some tips of poor pin contact, need to handle with pins carefully.

Situation 2: If using an adapter, please use a multimeter to measure whether the adapter pins are normal.

Situation 3: Individual pins of individual chips may not support pin contact check. In this case, you can temporarily uncheck the option **<Pin Detect>** in the lower left corner of the main interface.

Note: Pin detect can only check if the pin is on, cannot check the contact resistance.

If the contact resistance is too large, especially for NAND high-speed chips, it will also cause working errors. So be sure to clean the adapter, please do not easily trust any clip-type connection.

Note: Pin detect can only check if the pin is on, but cannot detect the contact resistance. If the contact resistance is too large, especially for NAND high-speed chips, it will also cause working errors. So make sure to clean the adapter and please do not easily trust any clip-type connection. After clamping, hard twist the adapter so that there is a small amount of frictional sliding between cooper sheets of the ZIF socket and pin headers or between the pins of integrated circuit, to make a good contact.

4) Check ID error

Many chips have an internal chip identification mark (Identification). The content generally has two or more bytes, generally the first byte is the Manufacturer ID, followed by the chip type or chip capacity. Different chip has the different ID and the function of this option is to prevent incorrect chips put in. This option is to check the chip ID before reading or writing. If it is correct, continue. If ID incorrect, abort. This function is optional. The option is checked by default.

> For the same type of chip, different manufactory may program in the same way. For chips not supported in the list, you can select same capacity same type ICs of other manufactories to program. Because ID is different, before programming, uncheck the option <Check ID>.

For few parts of microcontrollers, the chip ID cannot be read after chip is encrypted. At this time, the option <Check ID> should be unchecked in the main interface, otherwise check ID error will appear.