SUPERPRO® 6100, 611S, 610P Universal Programmers

www.xeltek.com
sales@xeltek.com
Tel: +1 408-530-8080
Fax: +1 408530-0096

1296 Kifer Rd. Suite #605
Sunnyvale, CA 94086

Important Information:

1. This manual applies to SuperPro 6100, SuperPro 611S, SuperPro 610P.
2. In order to connect the programmer easily and quickly, please connect the hardware after installing the software.
3. Please read this manual carefully before using programmers.
4. Do not use self-made adapters. It is recommended to buy the adapter made by Xeltek.

The information in this document is subject to change without notice.
In the event of any discrepancy in respect of meaning between the software and the manual due to the upgrade of the software version, the software shall prevail.
SuperPro® is the registered trademark of XELTEK.

Chapter 1 General Description

1.1 INTRODUCTION

1.1.1 What is SUPERPRO

SuperPro is a series of cost-effective, reliable, and high-speed universal programmers. They are designed to communicate through a USB 2.0 port and to operate with most IBM-compatible desktop computers and notebook computers based on Pentium. Their menu-driven software interface facilitates the operation. **Programming hardware includes the following items:**

- A programming module with a 48-pin ZIF socket (SuperPro 6100 includes a DIP48 DX0001 adapter; SuperPro 610P and SuperPro 611S include a built-in DIP module.)
- Power supply unit
- USB connection cable
- Optional socket adapters with 144 pins universal pin-drivers design to support devices with 144 pins or less and accommodate PLCC, TSOP, SOIC, SOP, QFP, TSSOP and BGA package types are available

Software features (USB 2.0 Series):

- Support Windows XP , Windows Vista, Win 7/8.
- Support numerous (over 10,000) types of devices, including PROM, E/EPROM, PLD and MCU from more than 150 manufacturers.
- Support many types of formats such as Binary, Intel (linear & segmented) Hex, Motorola S, Tektronix (linear & segmented), JEDEC, POF, etc.
- Device insertion test (48 pins or less) to detect defective chips, improperly inserted devices and pins of poor contact (model dependent)
- Integrated full screen buffer editing environment with commands such as fill, copy, move, swap, etc.
- Auto-generation of electronic serial numbers. (Only authorized users can use the user-defined methods).

1.1.2 Manual Organization

This manual consists of three parts:
Part 1: Introduction to the SUPERPRO series, including system requirements, software and hardware installation.
Part 2: Detail explanations on the software commands and all the function items.
Part 3: Appendix, including customer support information and error messages.

1.1.3 System Requirements

The minimum system configuration is as follows:

- Desktop or laptop, with Intel Pentium or equivalent processor, has at least one standard (USB2.0) compliant connector.
- Windows XP/Vista operating system.
- One CD-ROM drive.
- A hard disk with at least 1 GB free space

1.1.4 Programmer Package

The standard package contains the following items:

- One programmer module.
- One USB connection cable.
- One power supply unit.
- One software installation CD.
- The User's Guide.
- A registration form.

Chapter 2 System Installation

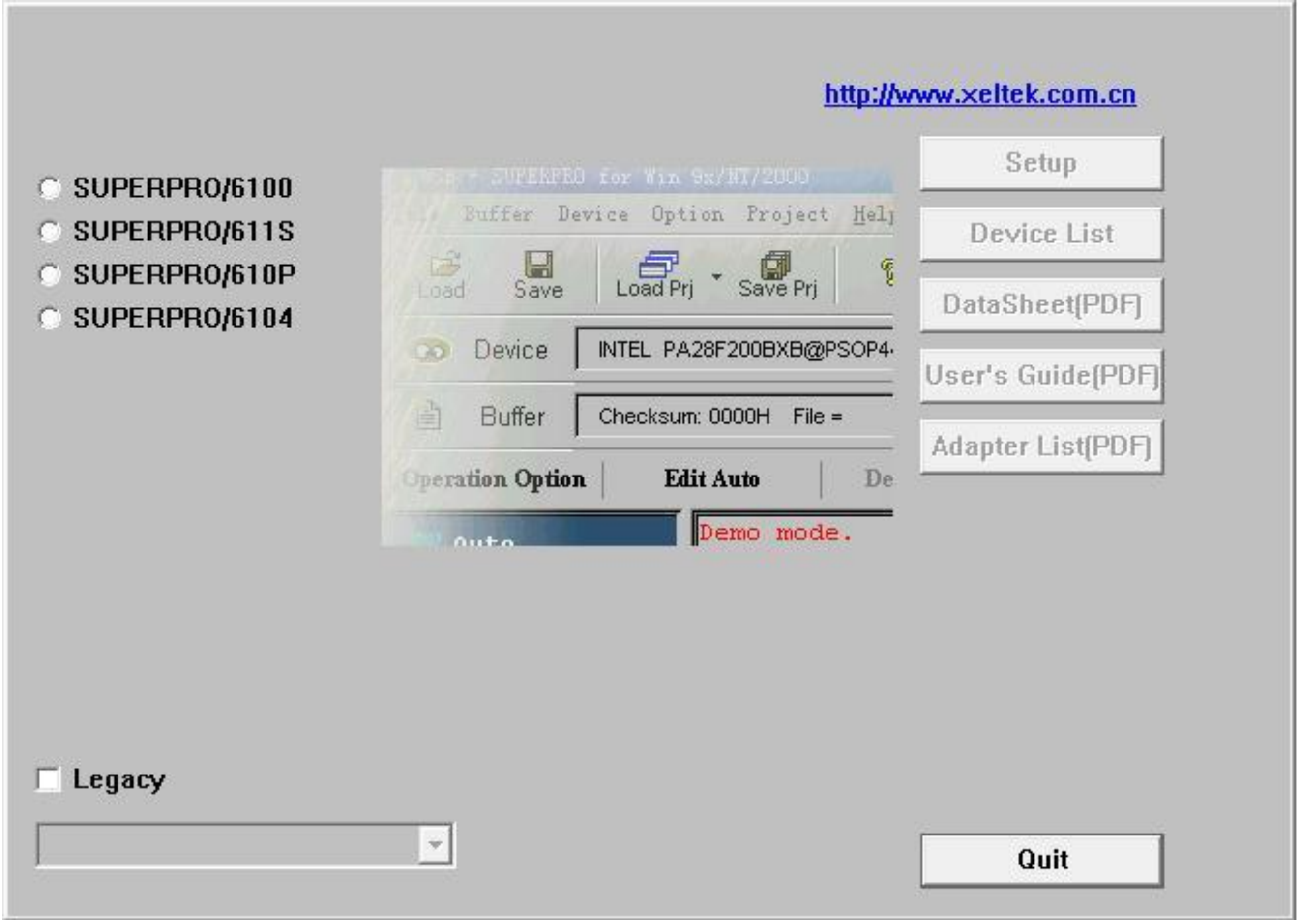
If it's your first time using the USB-based programmer from XELTEK, this chapter will help you to install the software and connect the programmer hardware properly. USB devices are PnP devices. At the first installation, Windows will start with "new hardware wizard" to scan all available INF files, and find out the appropriate drivers for the newly found hardware. In order to avoid unwanted complication during the installation of USB device, we strongly suggest that you install the software first. The setup program will handle the needed INF files and drivers automatically.

2.1 Software Setup

2.1.1 Setup from the CD-ROM

Insert the CD into the CD-ROM drive.

If the setup program starts automatically, a dialog box will show up for you to select the programmer model. If the setup program does not start automatically, please run SETUP.EXE in the root directory.



http://www.xeltek.com.cn

○ SUPERPRO/6100
○ SUPERPRO/611S
○ SUPERPRO/610P
○ SUPERPRO/6104

□ Legacy

Select the corresponding model of the programmer you have bought. Click Setup to install the application software for the programmer.

NOTE: Each model has its own software. Please do not mismatch them.

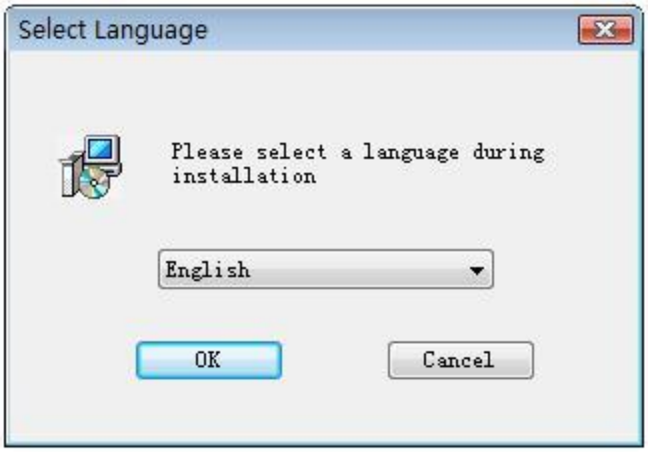
2.1.2 Download the installation software from the Internet

You can download the specific software for a certain model at Xeltek website: http://www.xeltek.com.cn/. It is usually a self-extracting file. You only need to run this file to setup the software.

2.1.3 Setup Process

The setup procedure goes step by step and you can change the default settings according to your needs in the setup process.

Step 1: Select the language that yon want to use in the setup process, as shown in the following language selection interface.

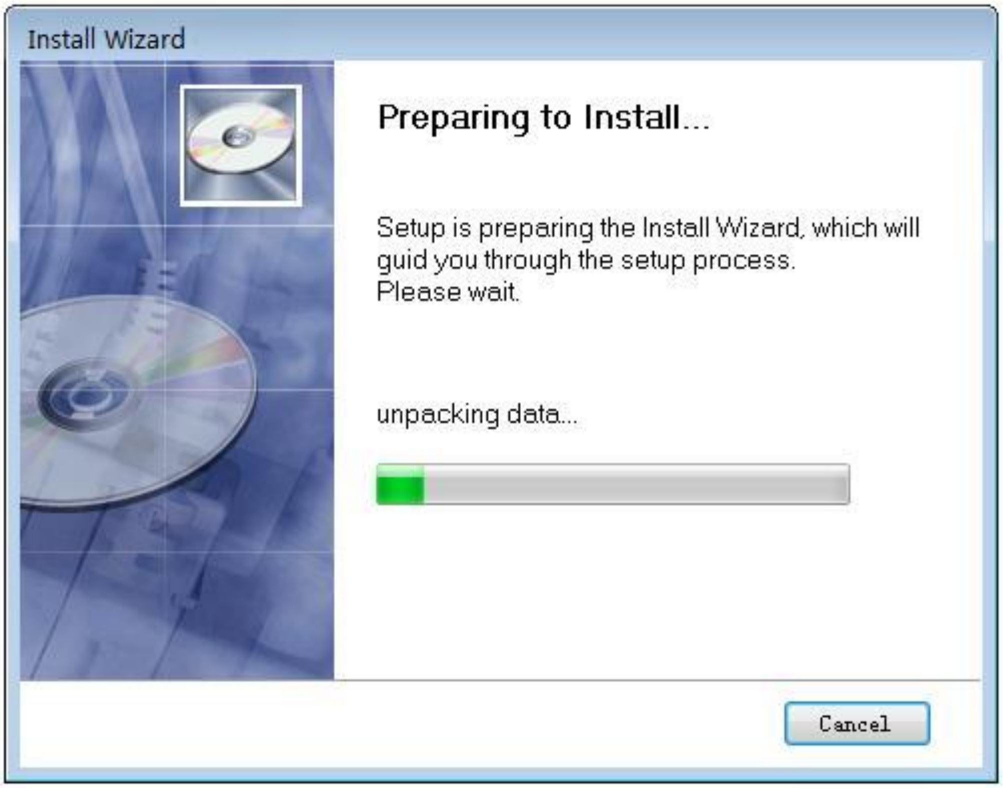


Select Language

Please select a language during installation

English

Click the “OK” button and a dialog box of “Preparing to install” appears.



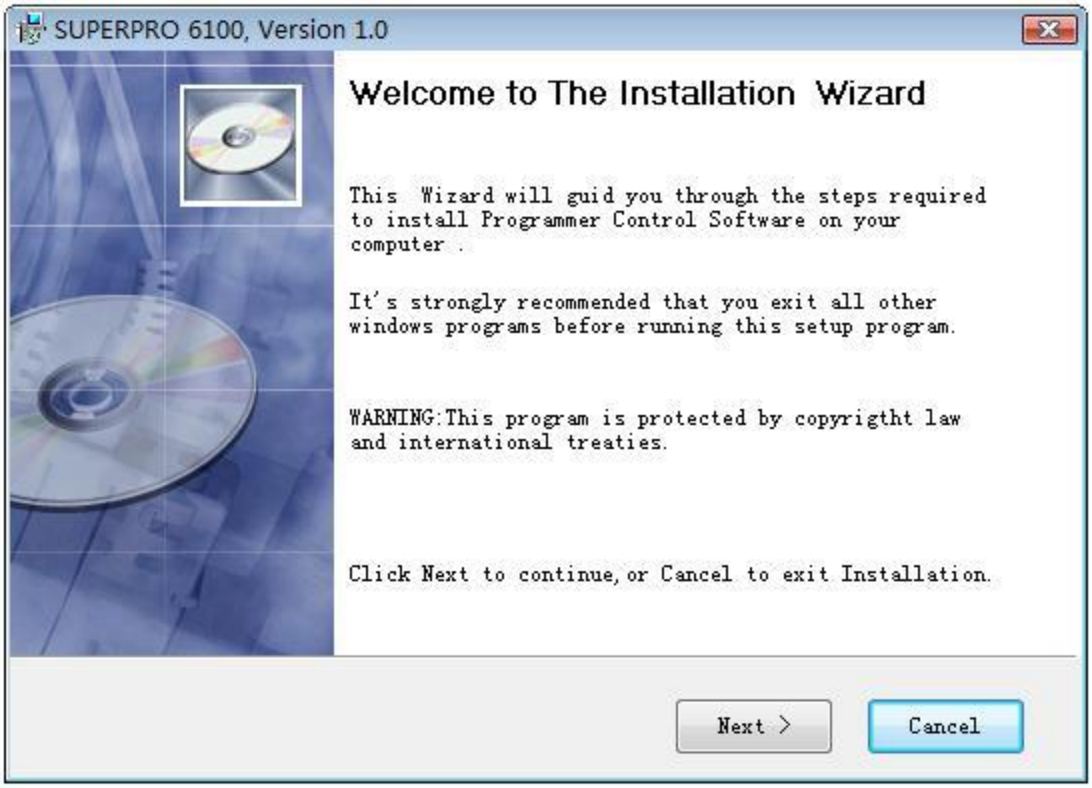
Install Wizard

Preparing to Install...

Setup is preparing the Install Wizard, which will guid you through the setup process.
Please wait.

unpacking data...

Then a welcome screen appears as shown below.



Welcome to The Installation Wizard

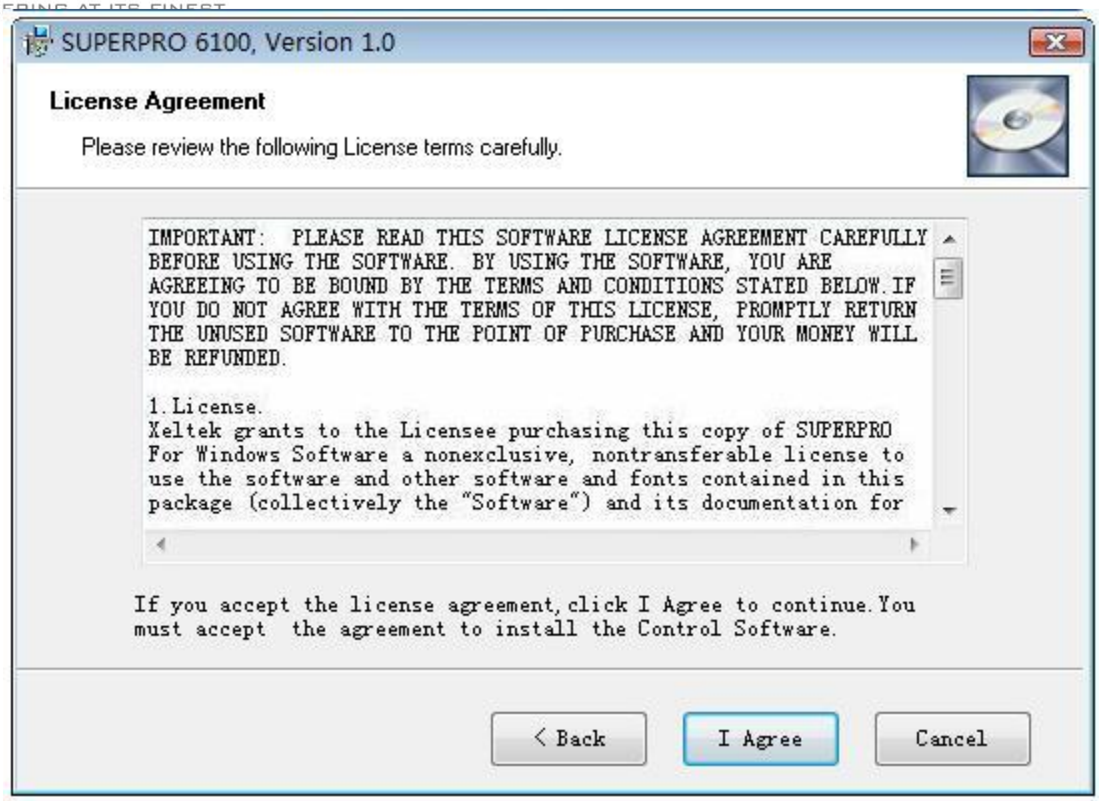
This Wizard will guid you through the steps required to install Programmer Control Software on your computer .

It's strongly recommended that you exit all other windows programs before running this setup program.

WARNING:This program is protected by copyrigtht law and international treaties.

Click Next to continue, or Cancel to exit Installation.

Step 2: Click the "Next" button. The License Agreement appears.



License Agreement

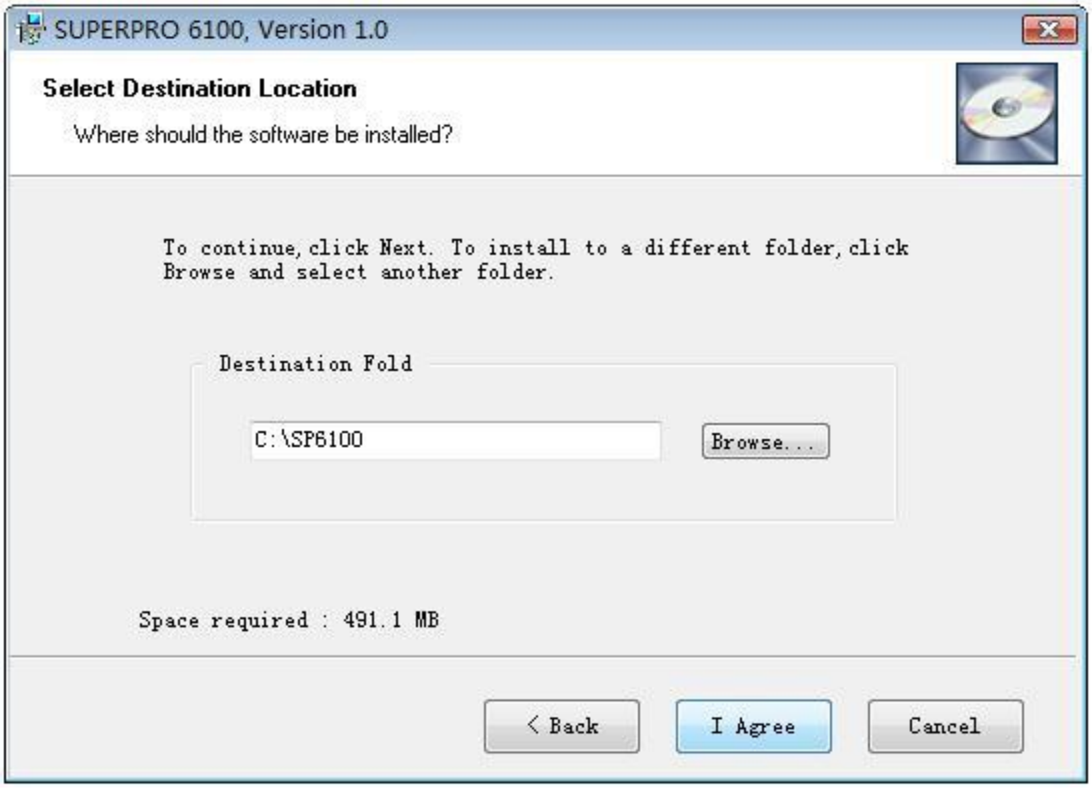
Please review the following License terms carefully.

IMPORTANT: PLEASE READ THIS SOFTWARE LICENSE AGREEMENT CAREFULLY BEFORE USING THE SOFTWARE. BY USING THE SOFTWARE, YOU ARE AGREEING TO BE BOUND BY THE TERMS AND CONDITIONS STATED BELOW. IF YOU DO NOT AGREE WITH THE TERMS OF THIS LICENSE, PROMPTLY RETURN THE UNUSED SOFTWARE TO THE POINT OF PURCHASE AND YOUR MONEY WILL BE REFUNDED.

1. License.
Xeltek grants to the Licensee purchasing this copy of SUPERPRO For Windows Software a nonexclusive, nontransferable license to use the software and other software and fonts contained in this package (collectively the "Software") and its documentation for

If you accept the license agreement, click I Agree to continue. You must accept the agreement to install the Control Software.

Step 3: Please read the license agreement carefully and click “I Agree” to continue the installation. Otherwise, click Cancel to exit the installation.



SUPERPRO 6100, Version 1.0

Select Destination Location

Where should the software be installed?

To continue, click Next. To install to a different folder, click Browse and select another folder.

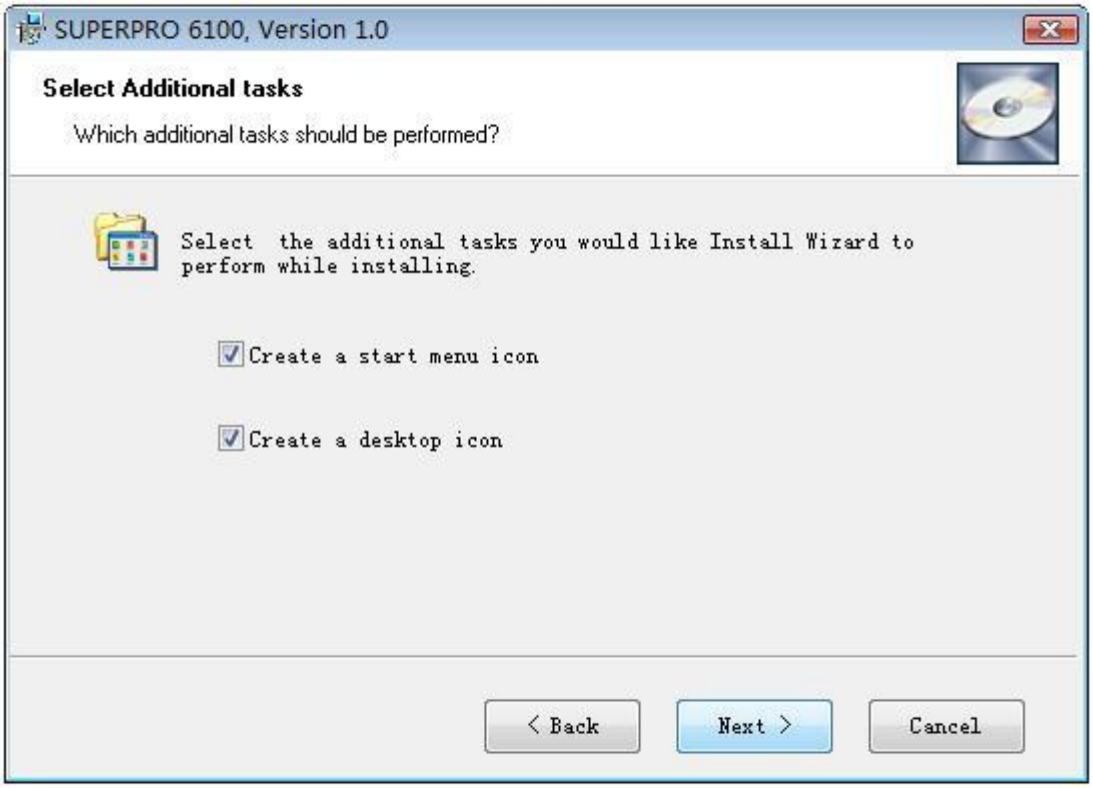
Destination Fold

C:\SP6100

Space required : 491.1 MB

Step 4: You can select the installation path in this step. Click the “Browse” button to change the default destination folder. Click the “Next >” button to continue the installation.

Step 5:



SUPERPRO 6100, Version 1.0

Select Additional tasks

Which additional tasks should be performed?

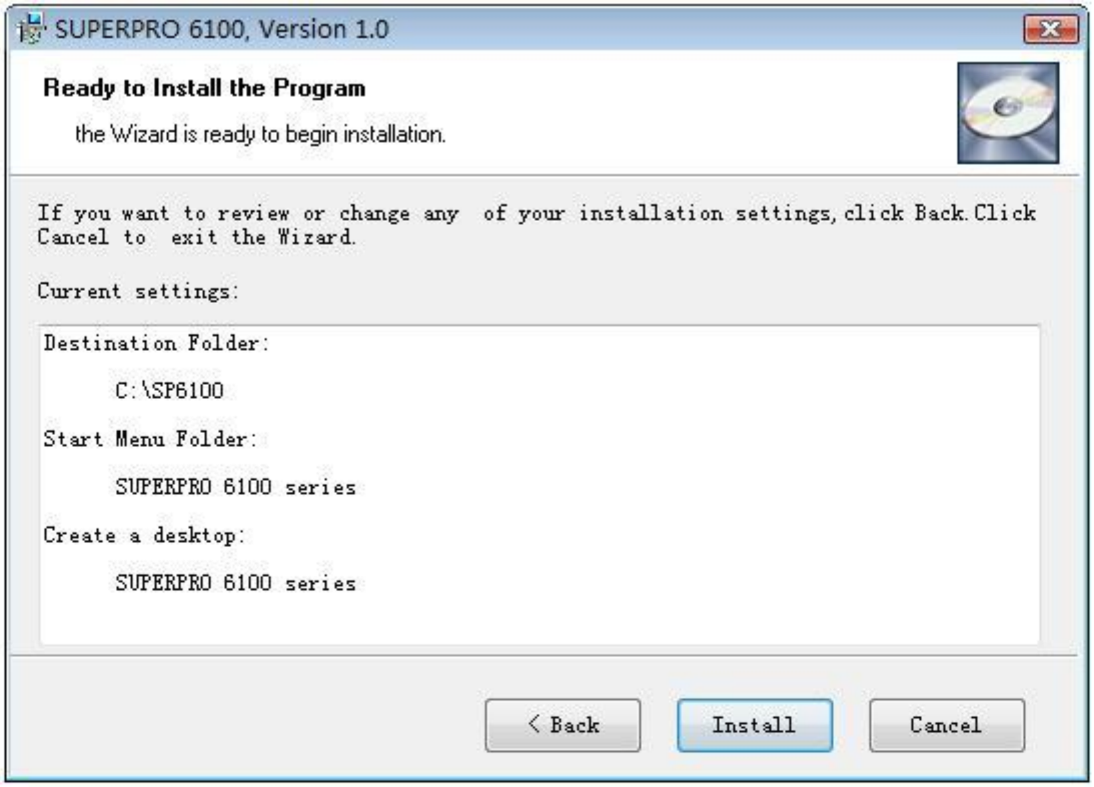
Select the additional tasks you would like Install Wizard to perform while installing.

☑ Create a start menu icon

☑ Create a desktop icon

You are asked whether to create a start menu icon and a desktop icon in this step. If you select the default setting, click the “Next” button to continue.

Step 6:



SUPERPRO 6100, Version 1.0

Ready to Install the Program

the Wizard is ready to begin installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the Wizard.

Current settings:

Destination Folder:

C:\SP6100

Start Menu Folder:

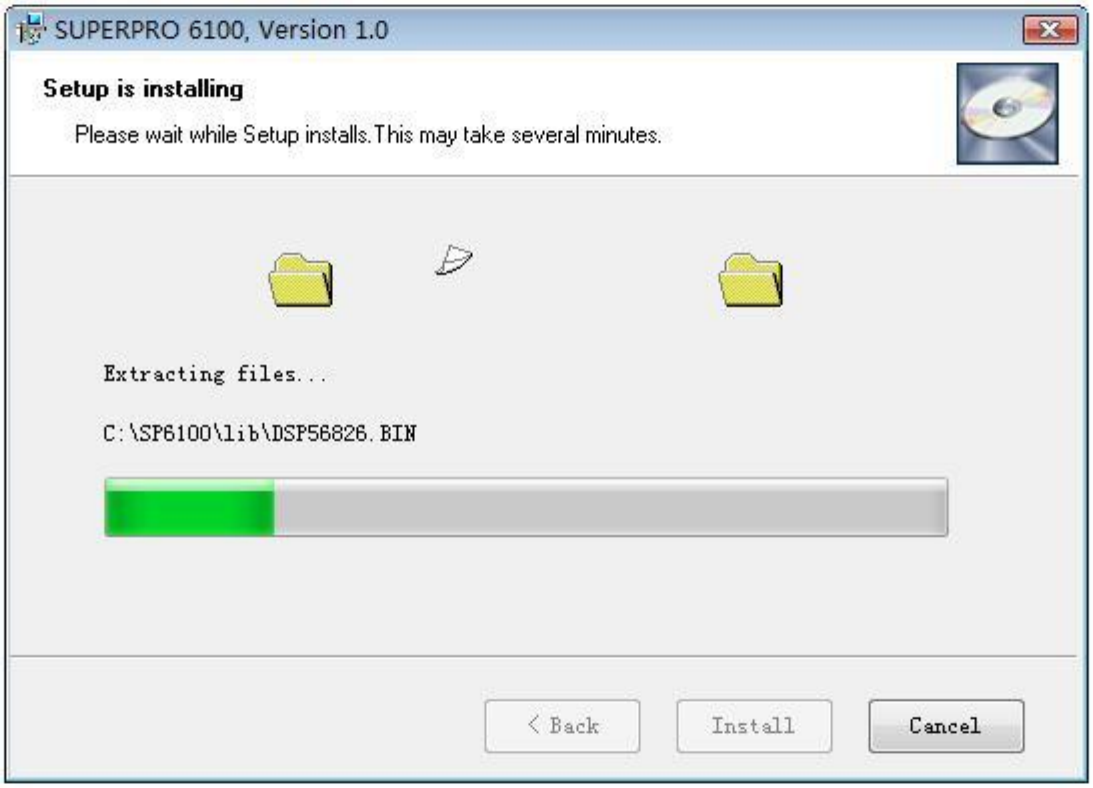
SUPERPRO 6100 series

Create a desktop:

SUPERPRO 6100 series

Check the above settings that you set in the setup process. Click the “Install” button to proceed to the next step.

Step 7:



SUPERPRO 6100, Version 1.0

Setup is installing

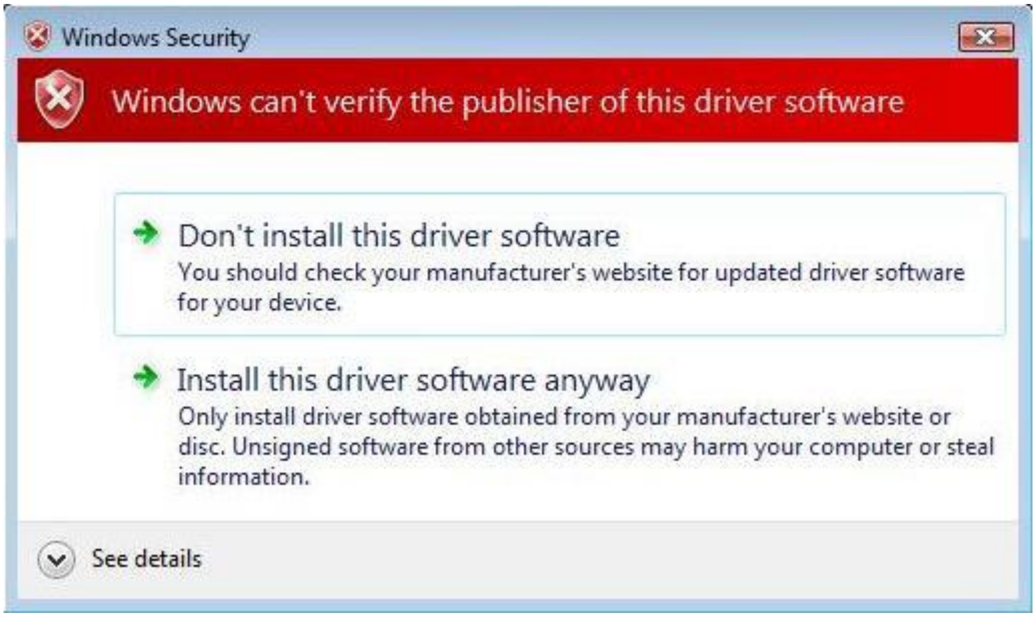
Please wait while Setup installs.This may take several minutes.

Extracting files...

C:\SP6100\lib\DSP56826.BIN

In this step, the files are being copied, the drivers are installed, the software is registered, etc.

If the user’s operating system is Windows Vista, the following dialog box will appear on the screen:



Windows Security

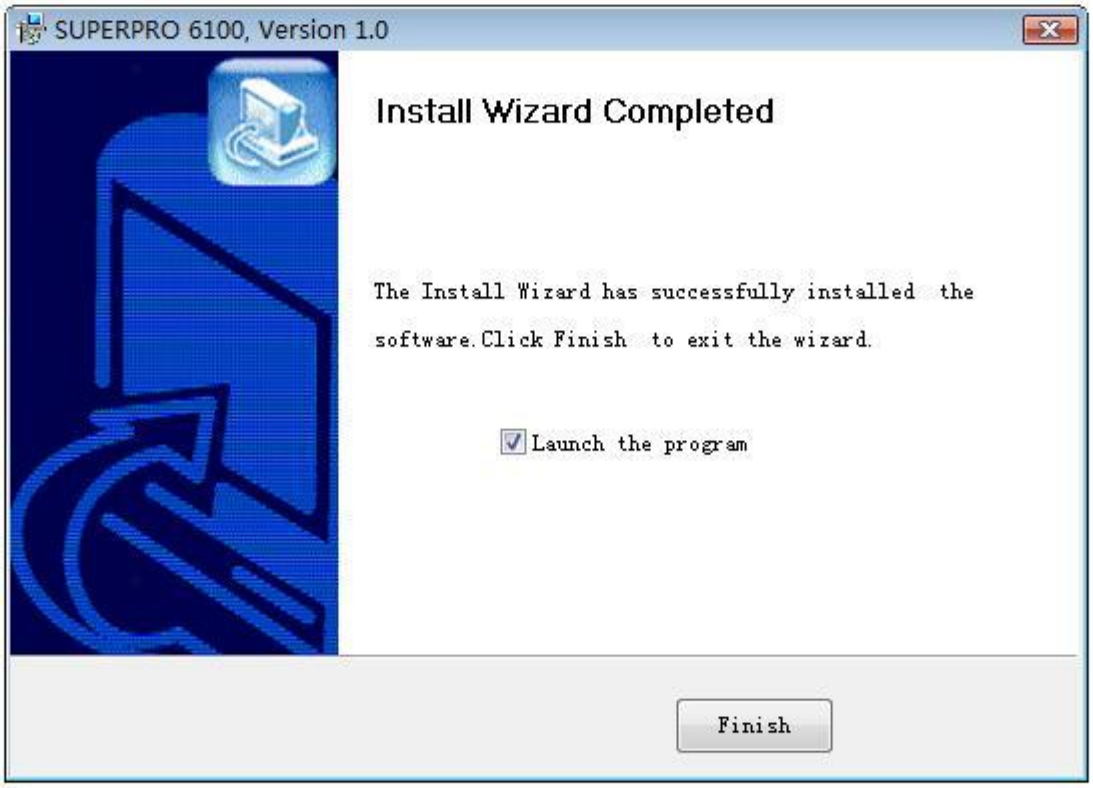
Windows can't verify the publisher of this driver software

Don't install this driver software
You should check your manufacturer's website for updated driver software for your device.

Install this driver software anyway
Only install driver software obtained from your manufacturer's website or disc. Unsigned software from other sources may harm your computer or steal information.

Now please select “Always install the driver software (I)”. Then wait until the next dialog box appears.

Step 8:



SUPERPRO 6100, Version 1.0

Install Wizard Completed

The Install Wizard has successfully installed the software. Click Finish to exit the wizard.

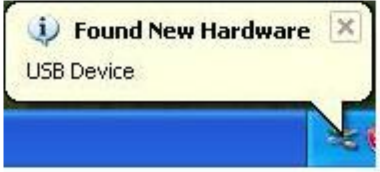
☑ Launch the program

Click the “Finish” button to finish the setup.

2.1.4 The hardware and driver installation

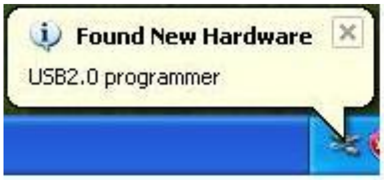
Connect the programmer module to your computer through the USB port. Turn on the programmer power switch.

After the application software is installed, the driver installation for the first time connection between the computer and the programmer module becomes very simple. You only need to wait until new hardware wizard finishes the installation. The following interfaces will appear on the screen one after another:



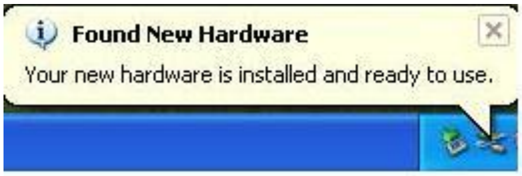
Found New Hardware

USB Device



Found New Hardware

USB2.0 programmer



Found New Hardware

Your new hardware is installed and ready to use.

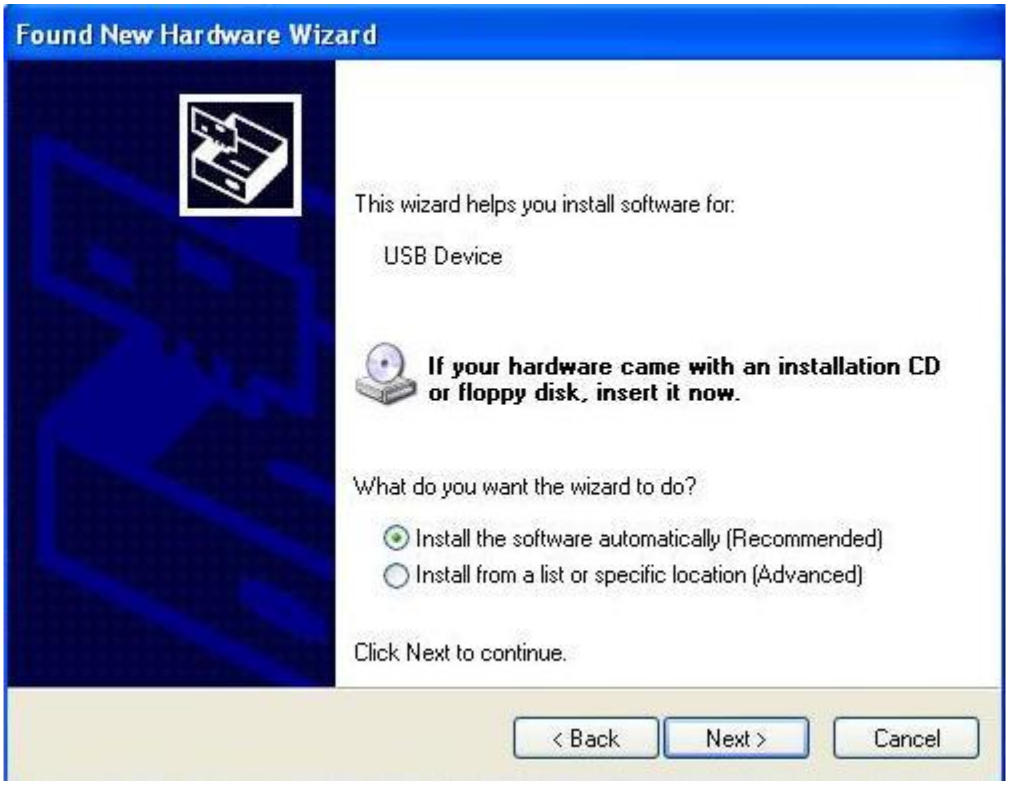
The actual interfaces may be somewhat different from the above ones. Anyway the user should wait until the installation finishes. Please don't run any other programs during the installation

2.1.5 Running the application software for the programmer

The application software for the programmer has a standard Windows UI, including pull down menus and buttons, etc. The user should have some knowledge about the MS Windows OS and be able to use the computer mouse well.

After the application software for the programmer is started, it will communicate with the programmer immediately and start the initialization. If the communication fails, please check according to the following steps:

- One programmer module. Check the connection between the programmer hardware and the computer, and turn on the power switch.
- Follow the correct installation steps. If the programmer hardware is connected to the computer before the programmer application software is installed, the computer will detect the new hardware and the following window will appear.



Found New Hardware Wizard

This wizard helps you install software for:

USB Device

If your hardware came with an installation CD or floppy disk, insert it now.

What do you want the wizard to do?

● Install the software automatically (Recommended)

○ Install from a list or specific location (Advanced)

Click Next to continue.

Please click the “Cancel” button lest the OS should install the driver with a compatible ID and cause some problems.

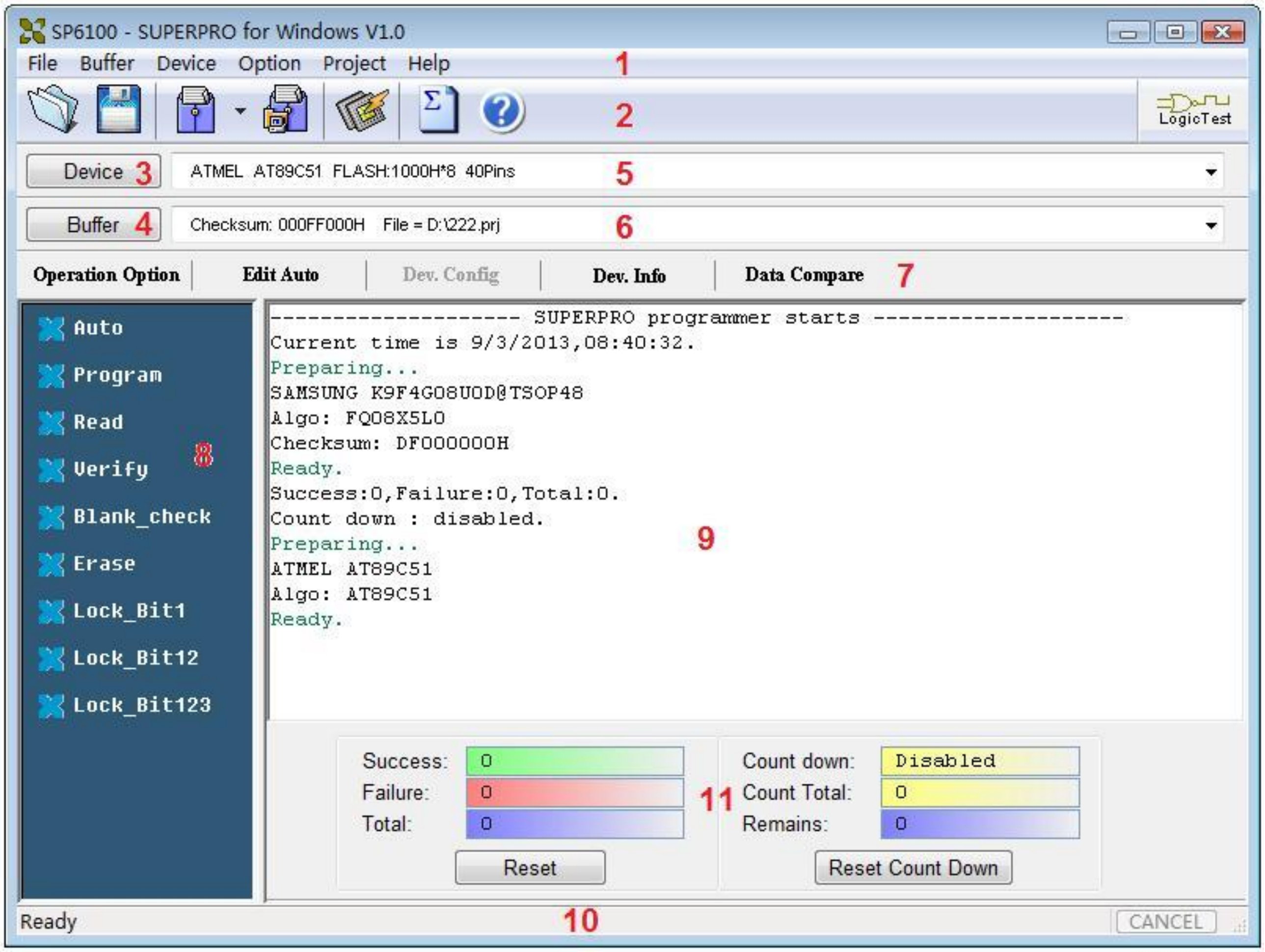
Chapter 3 Quick Guide

This chapter will help you understand the whole operation process of the chip programming. The content includes:

Before you start to program any devices, make sure that you have correctly install the programmer and the computer has succeeded in communicating with the programmer.

3.1 Description of the User Interfaces

The following interface will appear after the application software for the programmer is started.



Device: ATMEL AT89C51 FLASH:1000H*8 40Pins

Buffer: Checksum: 000FF000H File = D:\222.prj

```
-------------------- SUPERPRO programmer starts --------------------
Current time is 9/3/2013,08:40:32.
Preparing...
SAMSUNG K9F4G08U0D@TSOP48
Algo: FQ08X5L0
Checksum: DF000000H
Ready.
Success:0,Failure:0,Total:0.
Count down : disabled.
Preparing...
ATMEL AT89C51
Algo: AT89C51
Ready.
```

Success: 0
Failure: 0
Total: 0
Count down: Disabled
Count Total: 0
Remains: 0

Ready

1. Main Menu
2. Toolbar
3. Device Selection
4. Data Buffer for Editing
5. Device Messages
6. File Message
7. Operation Option Toolbar
8. Function Items for Operations on Devices
9. Operation Message Window
10. Progress Indicator
11. Statistical Window

Hotkey: SPACE = "Auto"

3.2 The steps of the programming

3.2.1 Hardware Preparation

Before using the programmer to program some device, please make sure the programmer is installed properly, and communication is established successfully between PC and programmer.

Some SMD devices need adapters. Please make sure that they are available for the devices. Insert the chip correctly (Device information for SMD and non-standard insertion will appear on the screen when the device is selected, if device information does not appears, check if chip insertion orientation follows the guideline of the standard chip insertion markings near the pin-driver socket).

3.2.2 Select the device

Click the "Device Selection" button or click the "Device" from the main menu and select the device. The “Device Selection” window will appear. First select the Device Type (such as E/EPROM, BPROM, SRAM, PLD, or MCU), then select the Manufacturer and Device Name in the window. Click OK button or double-click the device to confirm your selection. You can also enter the device part name in the “Search” box so that the selection is narrowed and can be done quickly.

3.2.3 Load data into Buffer

The process of programming a chip is the process in which the buffer data are written into the storage units of the chip according to the requirements of the manufacturer. Data can be loaded into the buffer in the following two ways:

1) By reading data from a file

You may load a file into the buffer by clicking “File” in the main menu and selecting ‘Load File’. The dialog box for loading a file appears after the ‘Load File’ is selected. In the dialog box, select the folder and file name you wish to load. Then select the appropriate file type in the ‘File Type’ dialog box appearing subsequently. After confirmation, the selected data will be loaded into the memory buffer. Please go to the Buffer edit window and check if the data is loaded correctly.

Note: Some Hex or S record files contain non-zero file start address. In this case, the start address should be entered in the File Address box.

2) Reading data from master chip:

Insert the master chip in the socket and make the selection of the device to read from. In the screen of “Function items for operations on the devices”, click Read to copy the data from the master chip into the memory buffer. At this point, you may go to the memory buffer edit screen and check if the data is loaded correctly. The data may be saved to a disk for later use.

Note: Some devices have no read function or are encrypted, under this condition data cannot be read from.

3.2.4 Set Options

1) Operation Option, including:

- Insertion Test option. Checking the pin contact before programming.
- Device ID Check option. Checking the device’s ID before programming.

- Beeper option. Checking whether the beeper sounds on or off when the operation succeeds or fails.
- Auto Increment Function option. If it is enabled, the label number increases automatically while programming and it is written to a designated place so that each chip has a different label number.
- Address Change option. Changing the start address and end address of programming zone of the device.
- Verification Mode option. Selecting a specific VCC voltage according to the device requirements in order to ensure that the chip is programmed correctly.

2) Edit Auto. In the window of "Function Items for Operations on the devices", each device has a basic batch-processing Auto operation, which automatically executes all the other operations of the device one after another in a preset order. Normally the following auto chip programming methods are selected:

- Erase
- Blank_check
- Program
- Verify
- Security or Protect

3) Dev.Config. For the chip that has the configuration word, the configuration word must be properly set before programming to ensure that the chip can be used on the target system. The configuration words for some devices are contained in the data file. The configuration words are automatically loaded into the configuration word buffer when the data file is loaded. For some devices, the selection must be made manually.

4) Dev. Information. After a device is selected, the dialog box of the device pops up and the vital information about the device is displayed in the box. Some devices must meet some special requirements while programming or there are some special conventions about the device algorithms. Therefore, the users must adjust operation steps or buffer data after carefully reading the text in the Dev. Information box. When an adapter is needed for programming the device, the adapter information will also be shown.

5) Production Mode. In order to program chips under production mode, the programmer always conducts the chip insertion test. Once the chip is inserted correctly, the auto batch-processing command is automatically started and the users do not need to use keyboard and mouse repeatedly.

3.2.5 Programming, writing the Data in Memory Buffer into a Chip

If the chip package is not DIP encapsulation, user may need to buy the adapter according to the device information and insert the chip properly into the socket.. (Contact Xeltek for selection and purchase of the adapters.)

The operation steps for programming a chip are as follows:

- Blank Check. This step can be skipped if the chip is brand new.
- Program.
- Verify. This step is necessary. The programming cannot be implemented

unless the chip passes the Verification. Some chips do not provide the unit-to-unit check function, but only provide the accumulation check function, such as Verify CRC, etc. Some chips even do not provide the accumulation check function.

- For the electrically erasable chips containing some information, the blank check will fail. In this case, the Erase operation should be performed first before the blank check. For some EPROM chips, the data can be overwritten. So no Erase operation needed.
- If encryption is required, add Security or Protect function after Verify.

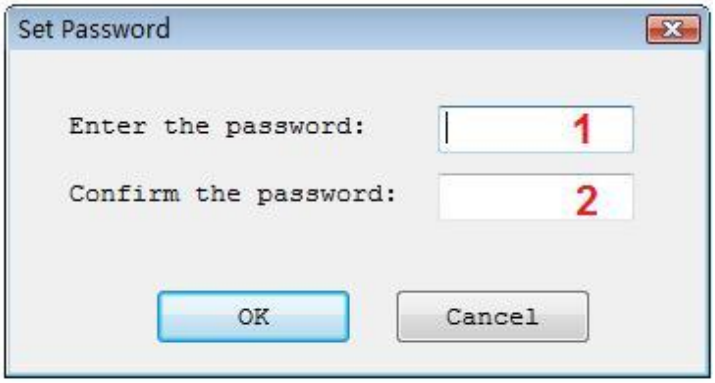
Note: For some devices, the configuration word should be set before the encryption is carried out. Refer to the detailed explanations on the function.

User can choose Auto to finish all the operations in one step.

3.2.6 Security and Password Input

To prevent the user's data from being used by others, sometimes the programmer asks the user to input the password in the operation process.

- Set the password: Input a new password.

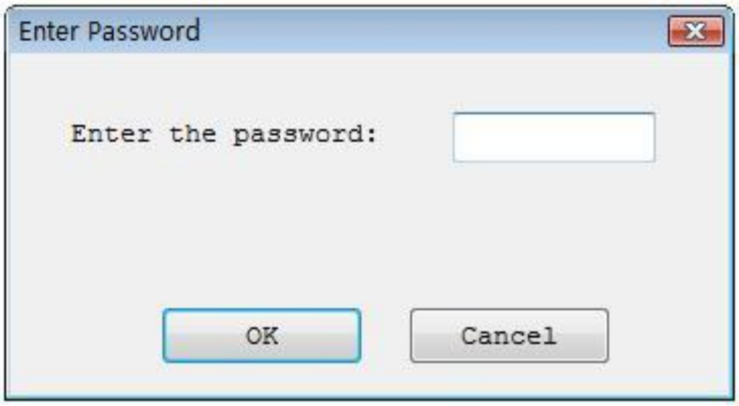


1. Enter the new password 2. Enter the password for confirmation

If the input password in 1 is the same and the same as in 2, the password is valid. If there is no input in 1 and 2, it means that no password is needed here.

Note: Spaces are characters and they can be used in the password.

- Verify the password: When the data protected with some password are operated on, the password needs to be input.

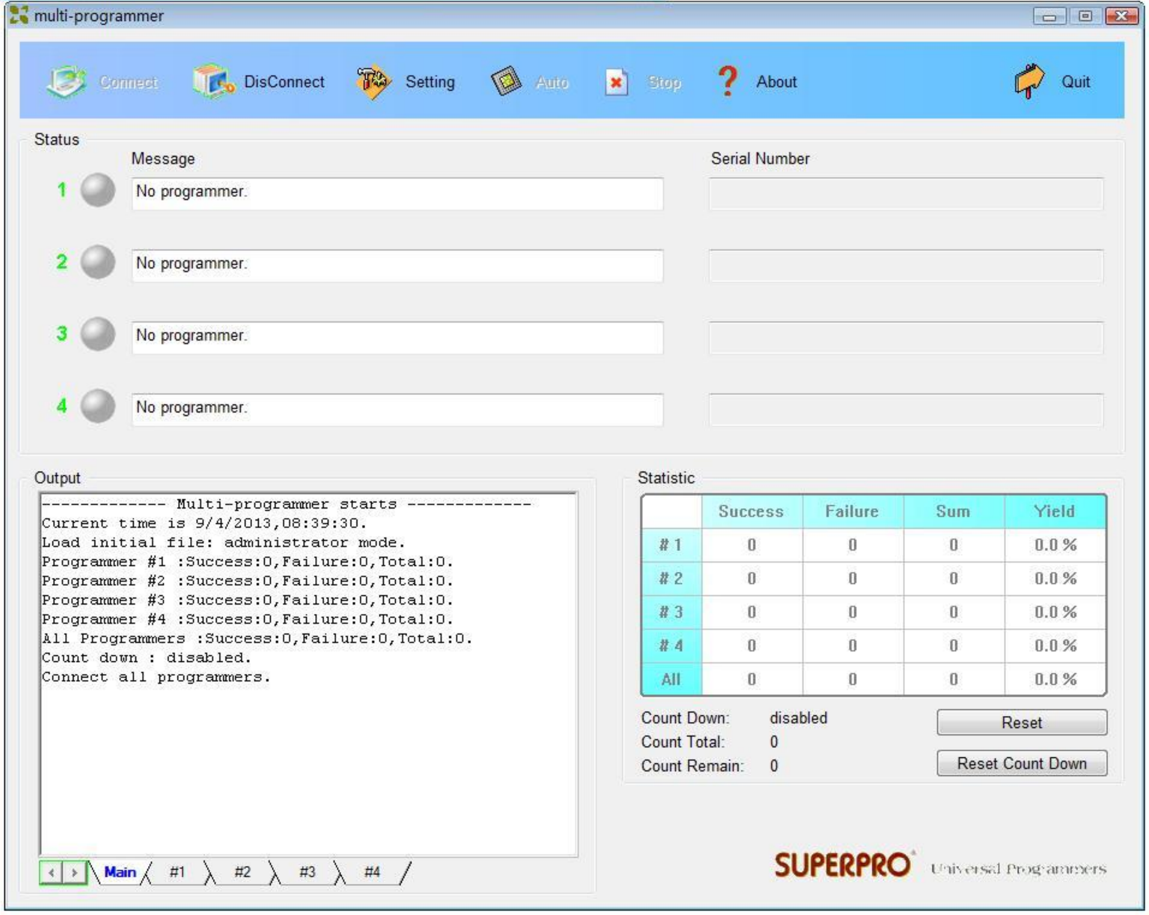


The description above is necessary for programming a chip. If it is the first time to for you use the programmer, please refer to "Detail Explanations on the Functions" in Chapter 4 for more detailed information.

3.3 Muti-programming

This function is only provided by SUPERPRO 6100, SUPERPRO611S. SUPERPRO /610P doesn't support this function (Take SP6100 for example).
SP6100 can be connected to hub, You can connect four SP6100s or one SP6104 by hub.

Find muti_p.exe at the folder "bin" in the directory where your SP6100's software is installed. Run it then following interface shows up.



Status

	Message	Serial Number
1	No programmer.	
2	No programmer.	
3	No programmer.	
4	No programmer.	

Output

```
------------- Multi-programmer starts -------------
Current time is 9/4/2013,08:39:30.
Load initial file: administrator mode.
Programmer #1 :Success:0,Failure:0,Total:0.
Programmer #2 :Success:0,Failure:0,Total:0.
Programmer #3 :Success:0,Failure:0,Total:0.
Programmer #4 :Success:0,Failure:0,Total:0.
All Programmers :Success:0,Failure:0,Total:0.
Count down : disabled.
Connect all programmers.
```

Statistic

	Success	Failure	Sum	Yield
# 1	0	0	0	0.0 %
# 2	0	0	0	0.0 %
# 3	0	0	0	0.0 %
# 4	0	0	0	0.0 %
All	0	0	0	0.0 %

Count Down: disabled
Count Total: 0
Count Remain: 0

“muti_p.exe” can manage four modules（4 SP6100s or 1 SP6104） integratively. The main screen is a management interface that controls the hardware of the programmer by sending commands to the SP6100 software. After the main interface, the process of communicating with the SP6100 software starts automatically.

If the system does not find the programmer or something is wrong with a module, select Disconnect to cancel the connection. Then check the connection between the hardware and computer. Select Connect to reconnect.

“muti_p.exe” has administrator mode and producer mode. In administrator mode, users can set projects and save them, then entering producer mode, users can load these project. In producer mode, users can load different project for different module or load the same project for all the modules. It’s very convenient and flexible.

NOTE: the hub used to connect SP6100s should be specified by XELTEK.

Chapter 4 Detailed Explanations on the Functions

Hotkey: SPACE = “Auto”

This chapter gives detail explanations of the programmer software functions.

- Menu and Toolbar.
- Select the device and device info box.
- Edit in the buffer and file info box.
- Window of function items for operations on devices.
- Operation info window.
- Status bar.
- Statistics window.

4.1 Menu and ToolBar

4.1.1 File

4.1.1.1 Load

There may be one or more data buffers in the device. If more than one data buffer are displayed after the device is selected, the buffers should be understood according to their names and the device data manual.

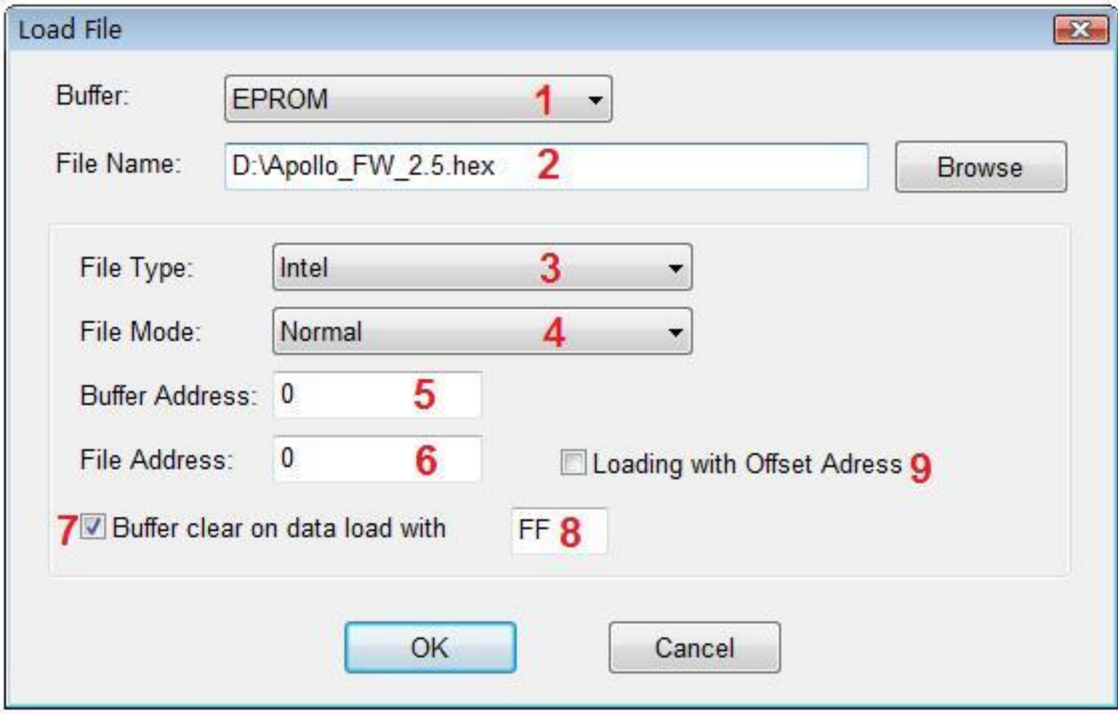
The two data types of the device are Data (HEX/ASCII) and Fuse.

For most EPROM and SCM, the data type is Data (HEX/ASCII). For PLD device, the data type is Fuse. After device is selected, the software will identify the data buffer type automatically. You can open the buffer edit dialog box to see all the buffer data types.

When some file data are loaded into the data buffer, the following rules apply:

- HEX/ASCII data buffer (EPROM, MCU etc.). 8 bits of the data are valid.
- JEDEC buffer (PLD/PAL). The lowest bit (1 bit) of the file data is valid.

The operations are shown below. After the “Load File” in the menu is selected, the Load File box pops up.



Load File

Buffer: EPROM 1

File Name: D:\Apollo_FW_2.5.hex 2

File Type: Intel 3

File Mode: Normal 4

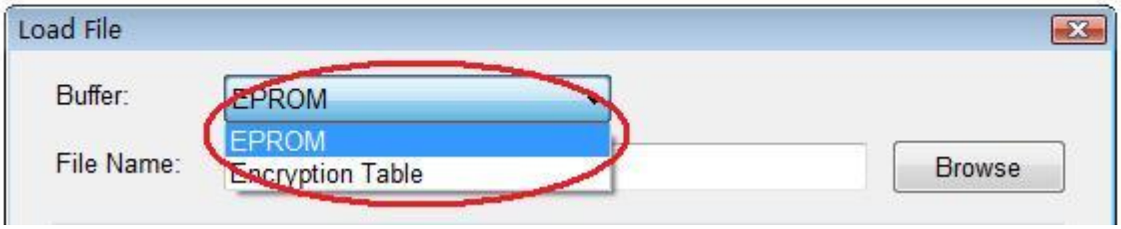
Buffer Address: 0 5

File Address: 0 6

☐ Loading with Offset Adress 9

7 ☑ Buffer clear on data load with FF 8

1. Data buffer names. When the default is used, it is considered to be the programming zone of the device described in the data manual. If there are several buffers, the corresponding programming zones are determined according to their names and the data manual. The file data are loaded into the data buffers according to the data buffer names.

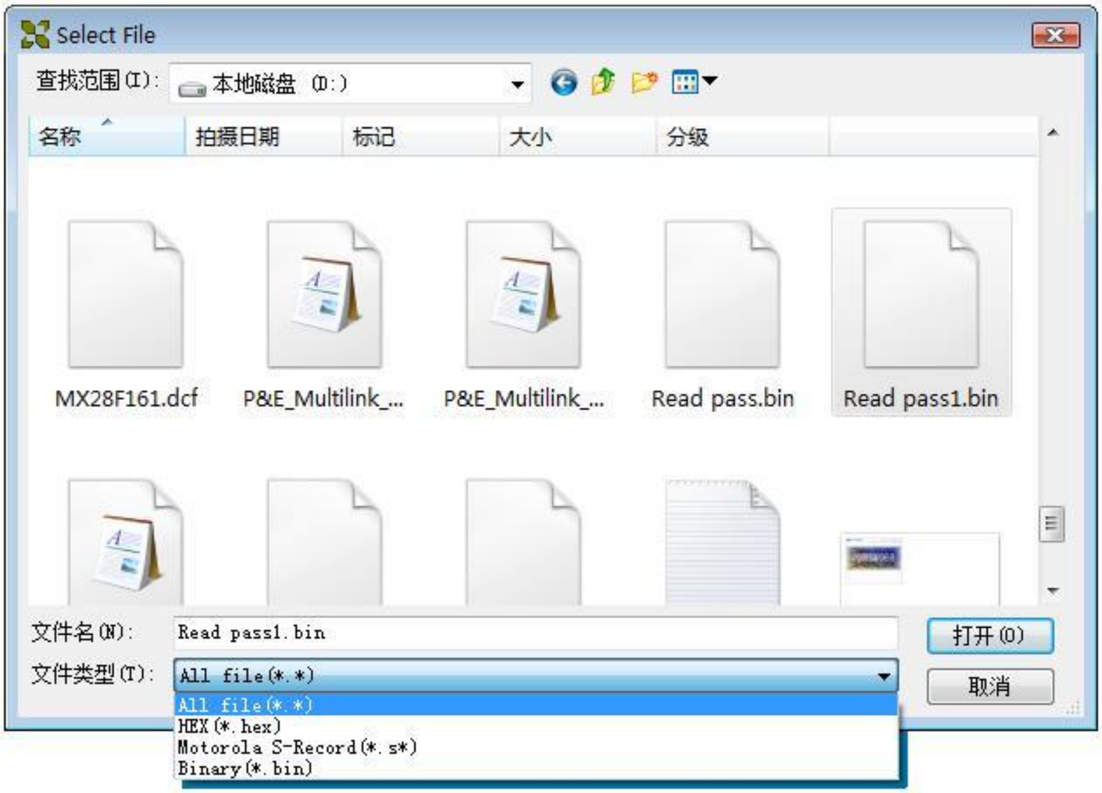


Load File

Buffer: EPROM

File Name:

2. To select a data file to be loaded, the path and the file name should be entered into the name field. You can directly input the file name or click the "Browse" button to select the file in the Select File box.



文件名(N): Read pass1.bin

文件类型(T): All file(*.*)

When the data type is Data (HEX/ASCII data), the files can be further divided into many different types according to their storage forms. Therefore, it can be ensured to load the data correctly after the corresponding file type is selected.

If the data type is Fuse, the data file type is JEDEC and the file extension is “jed”. There is no need to select the data file type for JEDEC file. Altera Corporation uses the file of POF format. If the users need to program the chip made by Altera Corporation, please refer to “Detail Explanations on Common Problems”.

3. File types. They include the binary, Intel Hex (or extended), Motorola S record, Tektronix Hex, etc.

4. File Loading Mode. It can be divided into the following classifications (the unit is byte):

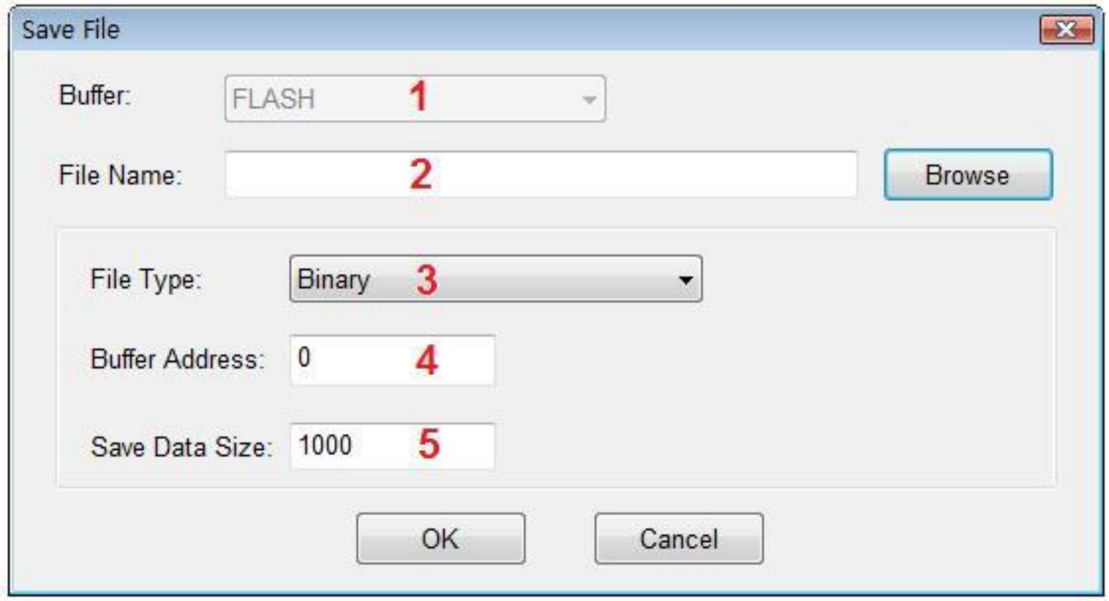
<Normal> : load the whole file.
<Even> : load the first byte of every two bytes and discard the second byte.
<Odd> : load the second byte of every two bytes and discard the first byte.
<1st byte of 4> : load the first byte of every four bytes and discard the other three bytes.
<2nd byte of 4> : load the second byte of every four bytes and discard the other three bytes.
<3rd byte of 4> : load the third byte of every four bytes and discard the other three bytes.
<4th byte of 4> : load the fourth byte of every four bytes and discard the other three bytes.
<1st 2_byte of 4>: load the first two bytes of every four bytes and discard the other two bytes.

<2nd 2_byte of 4>: load the last two bytes of every four bytes and discard the other two bytes.
5. Start Address of the Data Buffer. The data are loaded into the buffer from this buffer address.
6. File Offset Address. For files whose start addresses are non-0 start addresses (data offset addresses), their start addresses may need to be input when the files need to be loaded. Input the correct addresses in the edit boxes. The incorrect file offset address causes the head part of the buffer to be filled with FF (or incorrect data). The incorrect and large offset address causes the data to overflow or the system to fail. In order to avoid any unwanted events, the software will automatically detect the minimum offset address and display it in the information box.
7. Fill the data buffers with the specified data before the data are loaded.
8. Fill out the data input box.

9.If "File Offset Address" is not 0, load the file from non-0 start addresses (data offset addresses).

4.1.1.2 Save

Save the data in the current buffer to disk.



Buffer: FLASH
File Name:
File Type: Binary
Buffer Address: 0
Save Data Size: 1000

1. Data Buffer Name.
2. File Name for saving the Data.
3. File Type.
4. Start Address of Data Buffer. The data in the buffer is saved from this address.
5. Data Size to Be Saved. Save the data of the corresponding amount (the unit is byte).

4.1.1.3 Recent Projects

The full file names (including the full path) of the recent projects have been saved in the sub-menu of the main menu, it is convenient for the user to re-open the projects.

Refer to the “Project” menu for the definition of the project.

4.1.1.4 Exit

This command closes the programmer software and returns you to the control of the operating system.

4.1.2 Buffer

The menu manages data in the buffer.

4.1.2.1 Edit

The selection brings up Fuse Buffer Edit window if the device buffer type is Fuse; otherwise, brings up Data Buffer (HEX/ASCII) Edit window. You may edit the buffer data on the screen with the following keys:

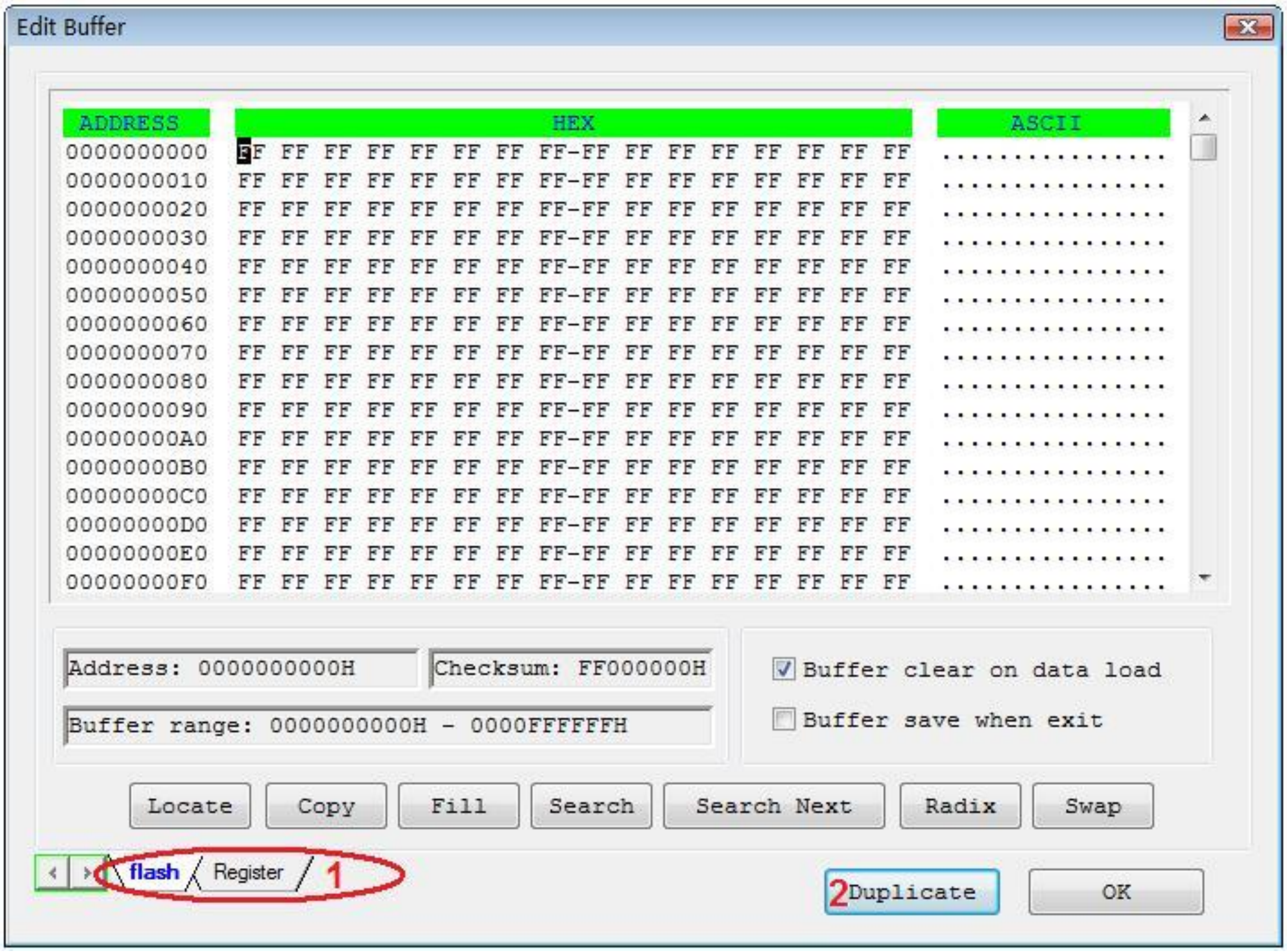
<PageUp> Page up

<PageDown> Page down

<Home> Move cursor to the beginning of the line

<End> Move cursor to the end of the line

An example with two data buffers is given in the following:



```
ADDRESS     HEX                                               ASCII
0000000000  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF  ................
0000000010  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF  ................
0000000020  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF  ................
0000000030  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF  ................
0000000040  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF  ................
0000000050  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF  ................
0000000060  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF  ................
0000000070  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF  ................
0000000080  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF  ................
0000000090  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF  ................
00000000A0  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF  ................
00000000B0  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF  ................
00000000C0  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF  ................
00000000D0  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF  ................
00000000E0  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF  ................
00000000F0  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF  ................
```

Address: 0000000000H

Checksum: FF000000H

Buffer range: 0000000000H - 0000FFFFFFH

☑ Buffer clear on data load

☐ Buffer save when exit

Note: Several data buffers can be opened at the same time by using the toolbar.

1. Can switch from one data buffer to another. The current data buffer name is shown in the highlighted area.

2. Save the data in the buffers in the text form.

The user can save the data in the buffer to the specified file in the text form as follows:

HEX/ASCII data buffer:

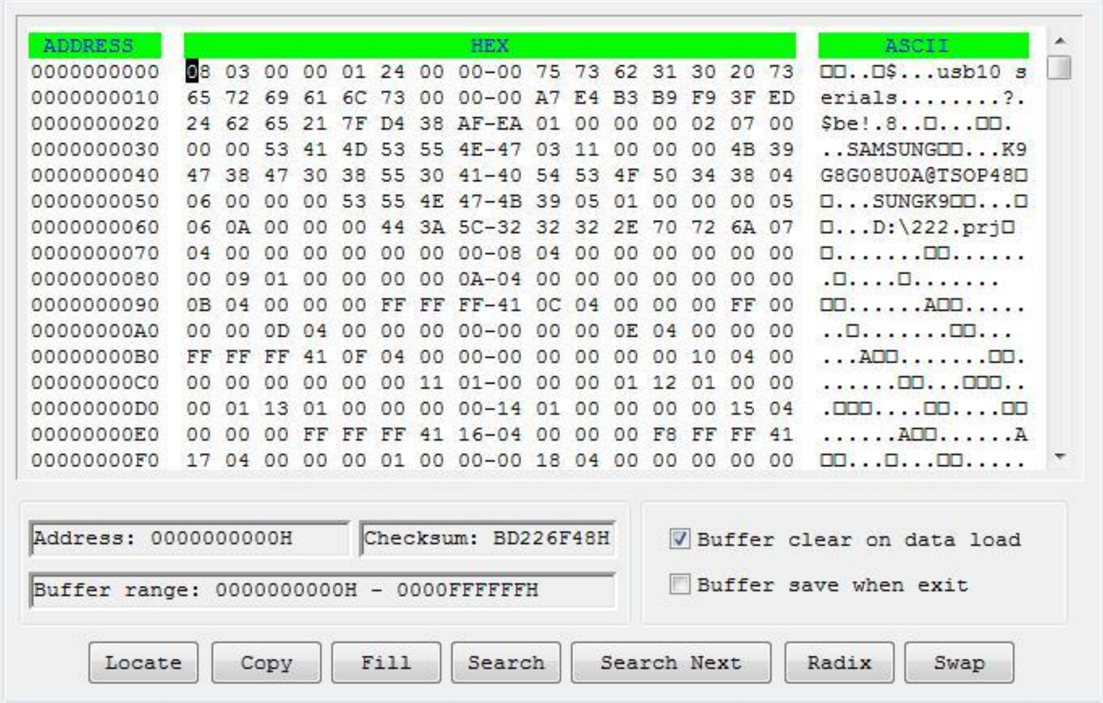
```
00000000  4D 5A 90 00 03 00 00 00-04 00 00 00 FF FF 00 00  MZ.._..._.......
00000010  B8 00 00 00 00 00 00 00-40 00 00 00 00 00 00 00  ........@.......
00000020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
00000030  00 00 00 00 00 00 00 00-00 00 00 00 80 00 00 00  ................
```

Fuse data buffer:

```
00000768  10011010101010101010101010101010
00000800  01101010101010101010101010101010
00000832  10100110101010101010101010101010
00000864  10101010011010101010101010101010
00000896  10101010101001101010101010101010
00000928  10101010101010100110101010101010
00000960  10101010101010101010011010101010
00000992  10101010101010101010101001101010
```

4.1.2.1.1 HEX/ASCII data buffer

The HEX/ASCII data buffer is 8-bit wide. TAB key may be used to switch between HEX and ASCII data for editing. According to the rule, if required to enter start and end addresses, the value of start address must be less than that of the end address.



```
ADDRESS     HEX                                                ASCII
0000000000  08 03 00 00 01 24 00 00-00 75 73 62 31 30 20 73   □□..□$...usb10 s
0000000010  65 72 69 61 6C 73 00 00-00 A7 E4 B3 B9 F9 3F ED   erials........?.
0000000020  24 62 65 21 7F D4 38 AF-EA 01 00 00 00 02 07 00   $be!.8..□...□□.
0000000030  00 00 53 41 4D 53 55 4E-47 03 11 00 00 00 4B 39   ..SAMSUNG□□...K9
0000000040  47 38 47 30 38 55 30 41-40 54 53 4F 50 34 38 04   G8G08U0A@TSOP48□
0000000050  06 00 00 00 53 55 4E 47-4B 39 05 01 00 00 00 05   □...SUNGK9□□...□
0000000060  06 0A 00 00 00 44 3A 5C-32 32 32 2E 70 72 6A 07   □...D:\222.prj□
0000000070  04 00 00 00 00 00 00 00-08 04 00 00 00 00 00 00   □.......□□......
0000000080  00 09 01 00 00 00 00 0A-04 00 00 00 00 00 00 00   .□....□.......
0000000090  0B 04 00 00 00 FF FF FF-41 0C 04 00 00 00 FF 00   □□......A□□.....
00000000A0  00 00 0D 04 00 00 00 00-00 00 00 0E 04 00 00 00   ..□.......□□...
00000000B0  FF FF FF 41 0F 04 00 00-00 00 00 00 00 10 04 00   ...A□□.......□□.
00000000C0  00 00 00 00 00 00 11 01-00 00 00 01 12 01 00 00   ......□□...□□□..
00000000D0  00 01 13 01 00 00 00 00-14 01 00 00 00 00 15 04   .□□□....□□....□□
00000000E0  00 00 00 FF FF FF 41 16-04 00 00 00 F8 FF FF 41   ......A□□......A
00000000F0  17 04 00 00 00 01 00 00-00 18 04 00 00 00 00 00   □□...□...□□.....
```

Address: 0000000000H

Checksum: BD226F48H

Buffer range: 0000000000H - 0000FFFFFFH

☑ Buffer clear on data load

☐ Buffer save when exit

- Locate:

In Locate Buffer（<Locate>） dialog box, enter the address of the data you wish to display and press 'OK'. The cursor will move to the location quickly.

- Fill:

The function will bring up the “Fill Data Into Buffer “dialogue box. It consists of three dialog boxes for the Start Address, End Address, Filling Data the three edit boxes, the ‘OK’ button and the ‘Cancel’ button. Input desired data to be filled into the Fill Data boxes, and specify the start address and end address. For the Fuse Buffer Edit window, the data will be either 1 or 0.

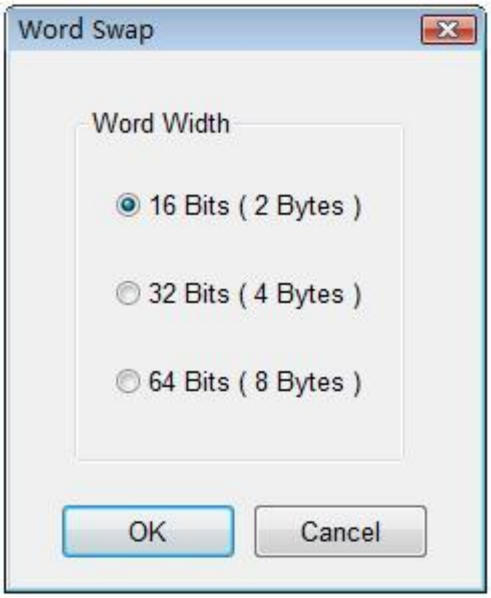
For the Data Buffer (HEX/ASCII data) Edit window, the data will be HEX numbers, for example AA, 55, E4.

- Copy:

This function displays Copy Buffer dialog box. It consists of Start Address, End Address, New Address input lines and OK and Cancel buttons. Data between start address and end address will be copied to the buffer beginning with new address.

- Swap:

Swap high byte and low byte according to the specified word width in the address range.



For example, assuming data buffer addresses 0-10(HEX) is:
12 34 56 78 90 AA BB CC - DD EE FF 11 22 33 44 55
The word width for selecting is:
16 Bits (2 bytes), after swapping the data is:
34 12 78 56 AA 90 CC BB - EE DD 11 FF 33 22 55 44
32 Bits (4 bytes), after swapping the data is:
78 56 34 12 CC BB AA 90 - 11 FF EE DD 55 44 33 22
64 Bits (8 bytes), after swapping the data is:
CC BB AA 90 78 56 34 12 - 55 44 33 22 11 FF EE DD

- Radix:

Toggles between HEX and DEC memory address display.

- Search:

Searching for a specified string.

- Search Next:

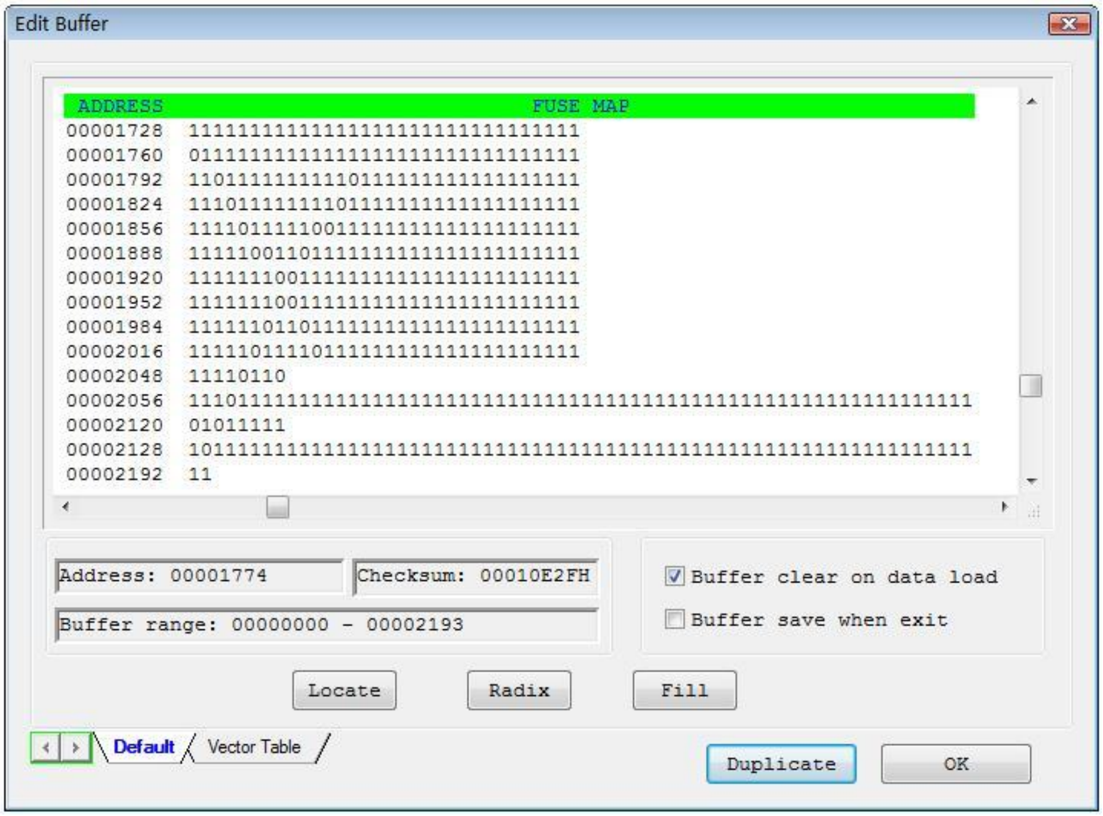
Performing the next search for the search string

4.1.2.1.2 Fuse Buffer

The data for editing is either 0 or 1, which has two explanations according to the different devices.

- 1 represents an intact fuse
  0 represents a blown fuse
- 1 represents an blown fuse
  0 represent an intact fuse

User can find the significations of 0 or 1 according to the device data manual of the device and JEDEC files.



```
ADDRESS                       FUSE MAP
00001728  11111111111111111111111111111111
00001760  01111111111111111111111111111111
00001792  11011111111110111111111111111111
00001824  11101111111101111111111111111111
00001856  11110111110011111111111111111111
00001888  11111001101111111111111111111111
00001920  11111110011111111111111111111111
00001952  11111110011111111111111111111111
00001984  11111101101111111111111111111111
00002016  11111011110111111111111111111111
00002048  11110110
00002056  1110111111111111111111111111111111111111111111111111111111111111
00002120  01011111
00002128  1011111111111111111111111111111111111111111111111111111111111111
00002192  11
```

Address: 00001774

Checksum: 00010E2FH

Buffer range: 00000000 - 00002193

☑ Buffer clear on data load

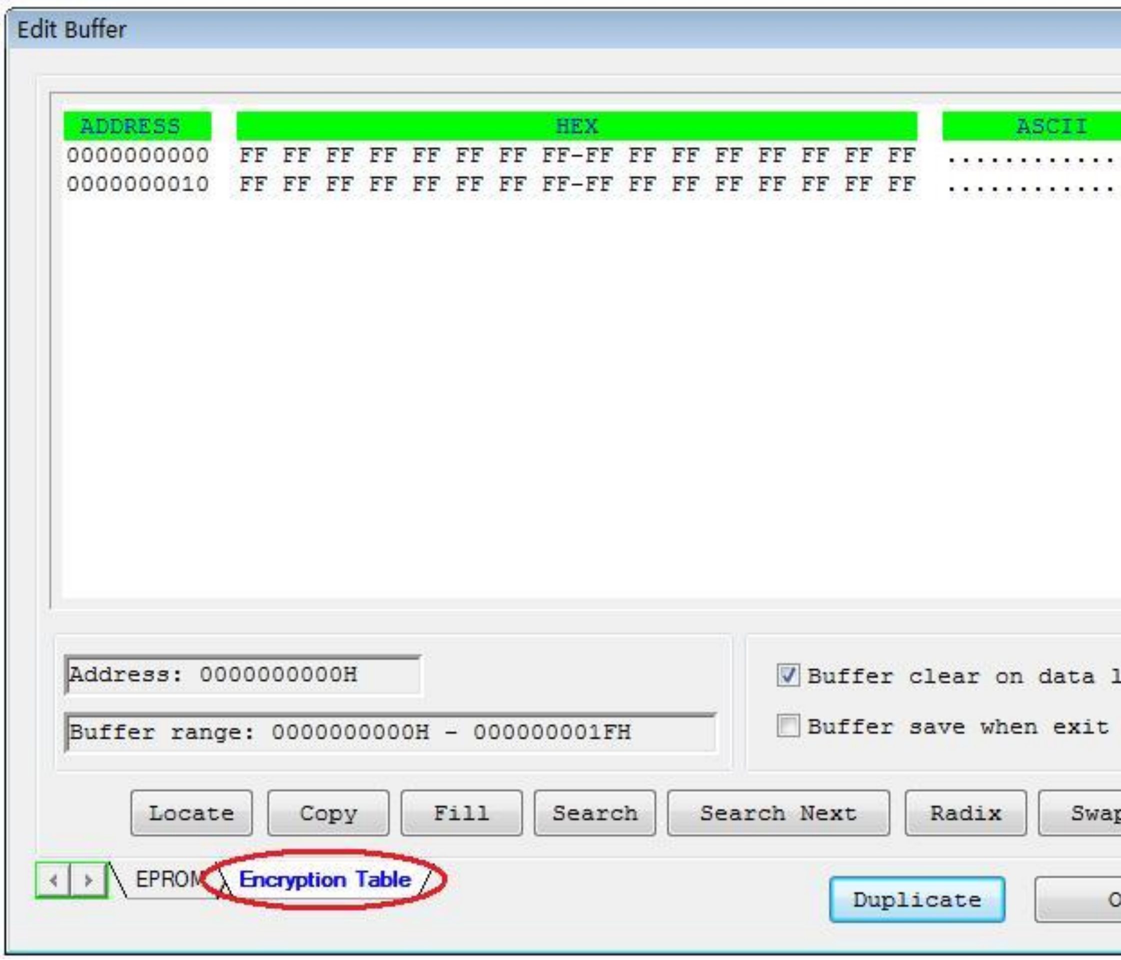
☐ Buffer save when exit

Note the three important options in the Buffer Edit dialogue box:

- Buffer clear at IC change: clear the buffer after device being selected
- Buffer clear on data load: clear the buffer before load file data
- Buffer save when exit: save buffer data when exit. When entering the system next time, load the saved data automatically.

4.1.2.3 Encryption table

The encryption table is a special data area. A data buffer corresponding to the encrypted data is called an encryption table. Whether an encryption table exists or not depends on the data manual.



Edit Buffer

```
ADDRESS     HEX                                                ASCII
0000000000  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF    ............
0000000010  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF    ............
```

Address: 0000000000H

Buffer range: 0000000000H - 000000001FH

☑ Buffer clear on data l

□ Buffer save when exit

The data of the encryption table are written into a chip by using the function item "Encryption" in the window of "Function Items for Operations on Devices". Attention should be paid to the following two points (in most cases):

- The chip can not be checked after the data is written into the encryption table.
- The data of the encryption table can not be read.

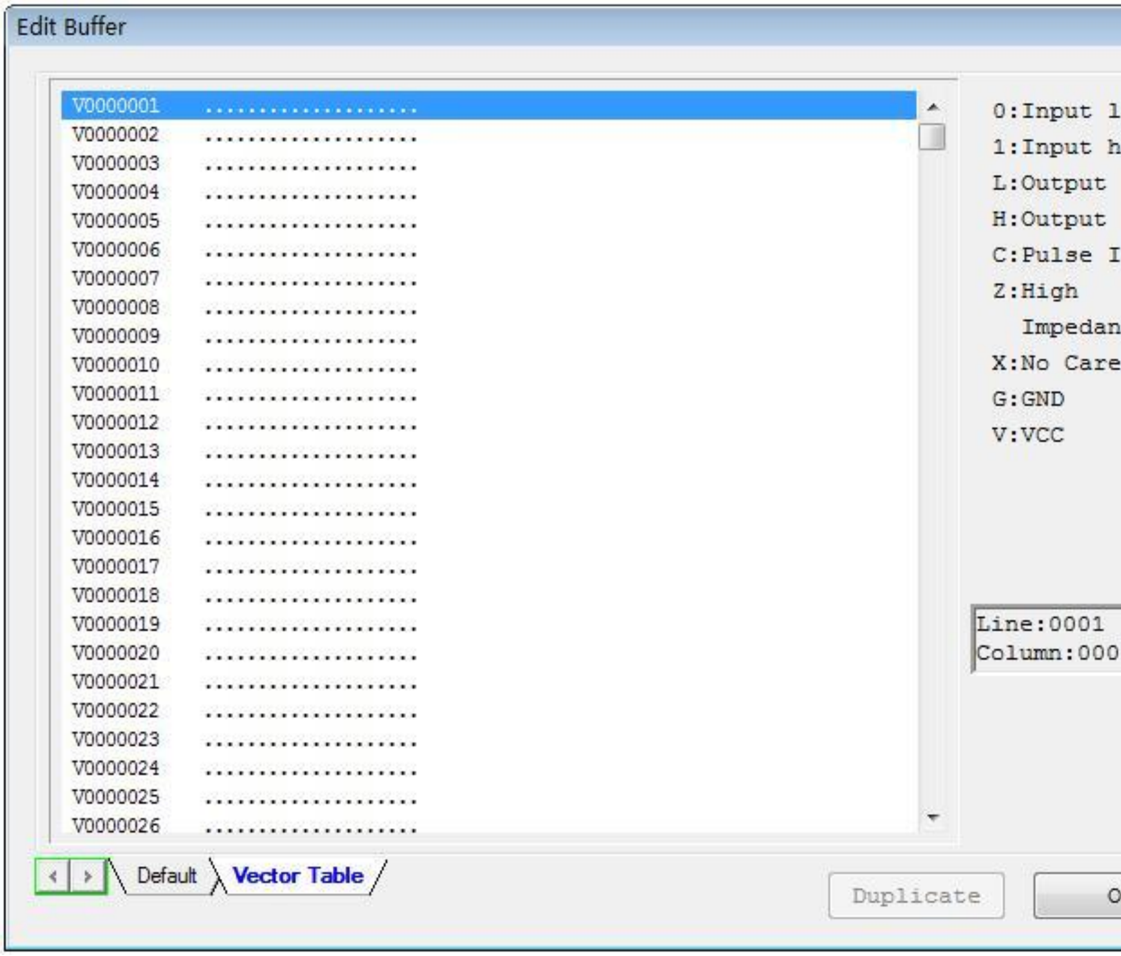
4.1.2.4 E-Fuse

If the data type of the device is Fuse and the JEDEC file has E field, you can open the E Field Fuse Data Edit dialogue box. Please refer to the chapter of Fuse Buffer.

4.1.2.5 Edit the Test Vector Table

The test vector table is a special data area. The data buffer corresponding to the data is called the test vector table.

If a test vector table is included in the JEDEC file, the software will load the test vector table into the buffer when the JEDEC file is loaded. Each line displays one vector and each bit represents one pin's test characteristic. The bits from left to right correspond to the pins from the first to the last respectively.



```
V0000001  ....................
V0000002  ....................
V0000003  ....................
V0000004  ....................
V0000005  ....................
V0000006  ....................
V0000007  ....................
V0000008  ....................
V0000009  ....................
V0000010  ....................
V0000011  ....................
V0000012  ....................
V0000013  ....................
V0000014  ....................
V0000015  ....................
V0000016  ....................
V0000017  ....................
V0000018  ....................
V0000019  ....................
V0000020  ....................
V0000021  ....................
V0000022  ....................
V0000023  ....................
V0000024  ....................
V0000025  ....................
V0000026  ....................
```

```
0:Input l
1:Input h
L:Output
H:Output
C:Pulse I
Z:High
  Impedan
X:No Care
G:GND
V:VCC
```

```
Line:0001
Column:000
```

Z: High impedance state
X: No care state
N: VCC or GND (non-test input pins)
H: Output Logic High (VOH)
L: Output Logic Low (VOL)
C: Clock pin
1: Input Logic High (VIH)
0: Input Logic Low (VIL)

4.1.3 Device

Before any operation, please select the device for programming, so that the programmer could select the proper algorithm according to the name and manufacturer of the devices.

4.1.3.1 Select Device

The function brings up the 'Device Selection' dialogue box. It includes the 'Manufacturer' box, 'Device Name' list box, 'Device Type' button, 'OK' button, 'Cancel' button, and 'Search Edit' box, etc.

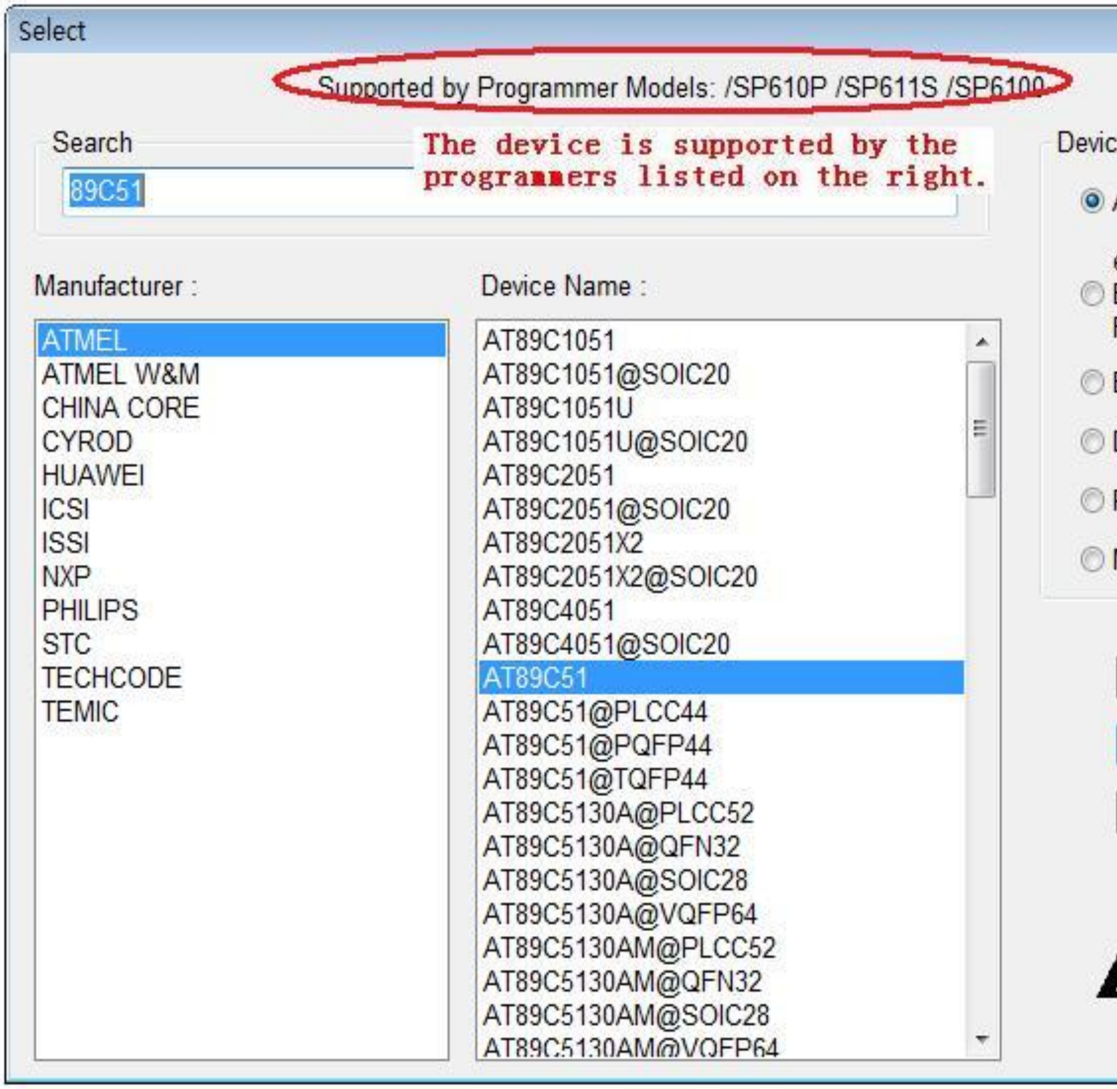
How to select a device?

- Select the device type through the Type Selection button, totally five groups: E/EPROM (including EPROM, EEPROM, FLASH), PLD, B/PROM, DRAM/SRAM, MCU. If not sure, please select All.

- Select the manufacturer in the Manufacturer column, meanwhile you can check whether it's correct or not by the icon shown at the lower right of the box.
- Select the device name through the Device Name column, click OK button.

Some printed name on the chip may include the parameters such as speed, temperature, encapsulation, etc. If the name is different from the name in the ‘Device Selection’ box, the user should ignore the difference or ask for help.

Use of the Search Edit box: Because of the large quantity of the devices and difference among the chips’ names, there may be some trouble in selecting the devices directly. The Search Edit box helps to select. Entering a few key characters, the number of the devices to be selected will decrease greatly to facilitate the selection of the specific device. For example, if the user enters '89' in the Search Edit box to list all the devices of the “89” series, the device name or manufacturer name containing '89' will be listed in the list box. The search function is sensitive to the sequence of the character string. The character strings such as '819', 'DA8S9' will be listed, but '98' will be ignored.



Select

Supported by Programmer Models: /SP610P /SP611S /SP6100

Search: 89C51

The device is supported by the programmers listed on the right.

Manufacturer :

ATMEL
ATMEL W&M
CHINA CORE
CYROD
HUAWEI
ICSI
ISSI
NXP
PHILIPS
STC
TECHCODE
TEMIC

Device Name :

AT89C1051
AT89C1051@SOIC20
AT89C1051U
AT89C1051U@SOIC20
AT89C2051
AT89C2051@SOIC20
AT89C2051X2
AT89C2051X2@SOIC20
AT89C4051
AT89C4051@SOIC20
AT89C51
AT89C51@PLCC44
AT89C51@PQFP44
AT89C51@TQFP44
AT89C5130A@PLCC52
AT89C5130A@QFN32
AT89C5130A@SOIC28
AT89C5130A@VQFP64
AT89C5130AM@PLCC52
AT89C5130AM@QFN32
AT89C5130AM@SOIC28
AT89C5130AM@VQFP64

Users can conveniently reduce the number of devices listed by using Search Edit box, but please make sure only correct character string is entered. Otherwise, you will not be able to find the specific devices. Therefore please double check the character string or make the Search Edit box blank when a device can not be found.

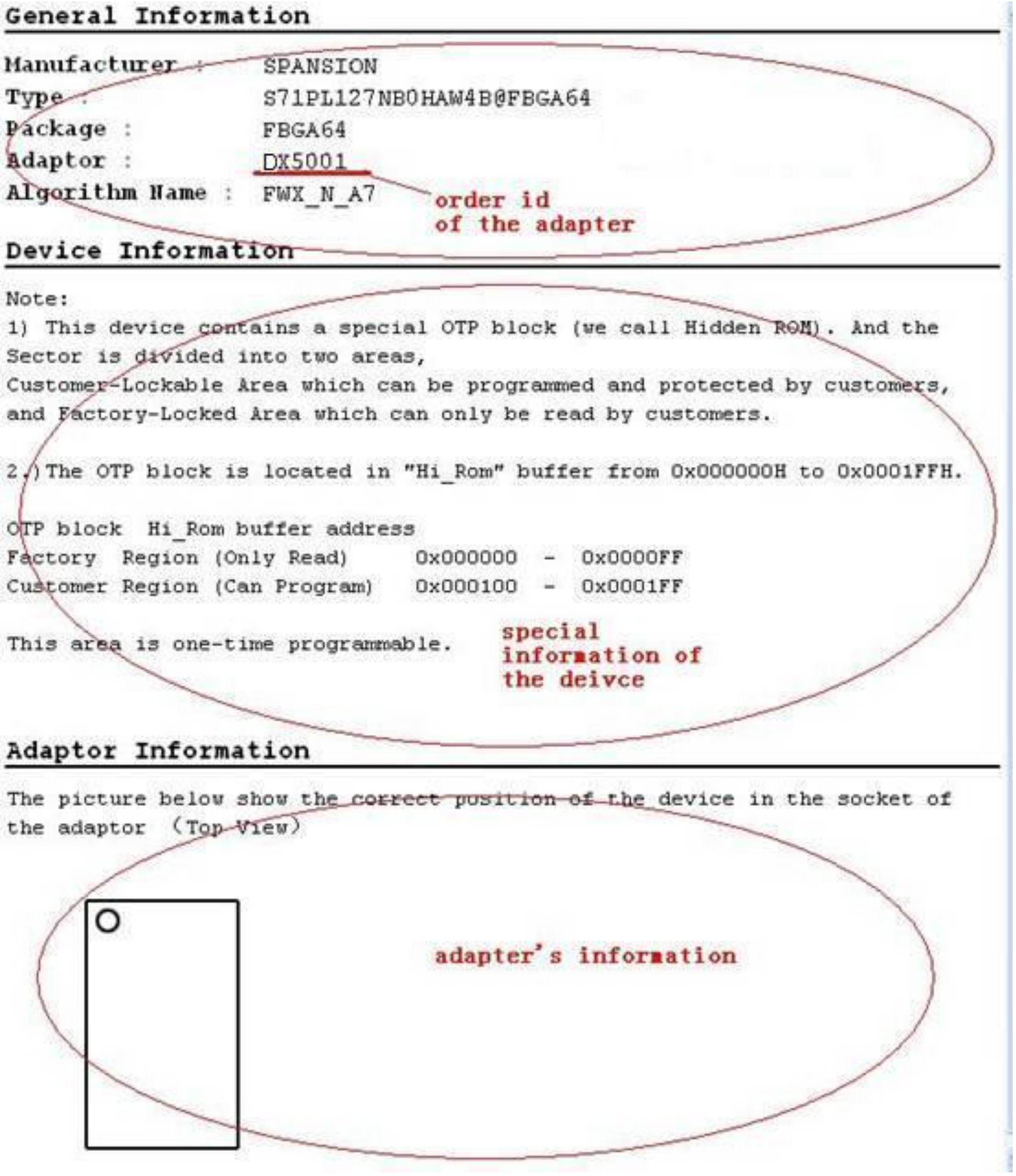
4.1.3.2 Dev. Information

It includes:

- The key information about the devices.
- The requirements of the manufacturers.
- The conventions on the special devices. Besides the basic requirements on the operations of a device, there may be some special requirements (or conventions) on them, including the arrangement of the data buffer content, the explanations on the special operations on the device, etc. the user should read the information carefully, arrange the data and prepare the corresponding settings according to the requirements to ensure the correctness of the programming. This kind of information can be found in the data manual of the device.

- The information about the adapter. The configuration of the programmer the user bought comes with a standard pin-driver socket for a DIP encapsulation chip. If you wants to program the chips of the other encapsulations, you need to buy other types of adapters.

The adapter information may include the order number (such as DX5001), the insertion direction of the chip, the size of the chip, etc.



General Information

Manufacturer : SPANSION
Type : S71PL127NB0HAW4B@FBGA64
Package : FBGA64
Adaptor : DX5001
Algorithm Name : FWX_N_A7

order id
of the adapter

Device Information

Note:
1) This device contains a special OTP block (we call Hidden ROM). And the Sector is divided into two areas,
Customer-Lockable Area which can be programmed and protected by customers, and Factory-Locked Area which can only be read by customers.

2.) The OTP block is located in "Hi_Rom" buffer from 0x000000H to 0x0001FFH.

```
OTP block  Hi_Rom buffer address
Factory   Region (Only Read)     0x000000  -  0x0000FF
Customer  Region (Can Program)   0x000100  -  0x0001FF
```

This area is one-time programmable.

special
information of
the deivce

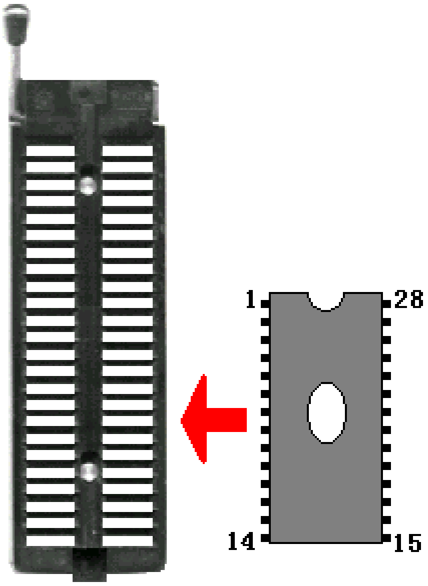
Adaptor Information

The picture below show the correct position of the device in the socket of the adaptor (Top View)

adapter's information

Chip Insertion

- DIP Encapsulation: The chip is inserted with the notch upwards while its bottom line is aligned with the bottom line of the DIP socket. The diagram below shows the process of inserting a 28-pin chip.



- None-DIP Encapsulation: Socket adapters are available for PLCC, QFP, SOIC etc. Insert the chip according to the instructions in the Adapter Information box. The insertion follows the position of pin 1. It is necessary to identify the pin 1 of the chip. Usually there is a solid square or dot on the chip which indicates the first pin of the chip.
- Direct-connect: Connect adapters with 144 pins to programmers directely.
- Indirect-connect: Insert adapters with 48 pins into DIP on the programmer..

Note:

1. Generally, the pins of the chip are downward when they are inserted. However, for the chip with PLCC20 package, its pins should be upward. This insertion is called the "Dead Bug" method.

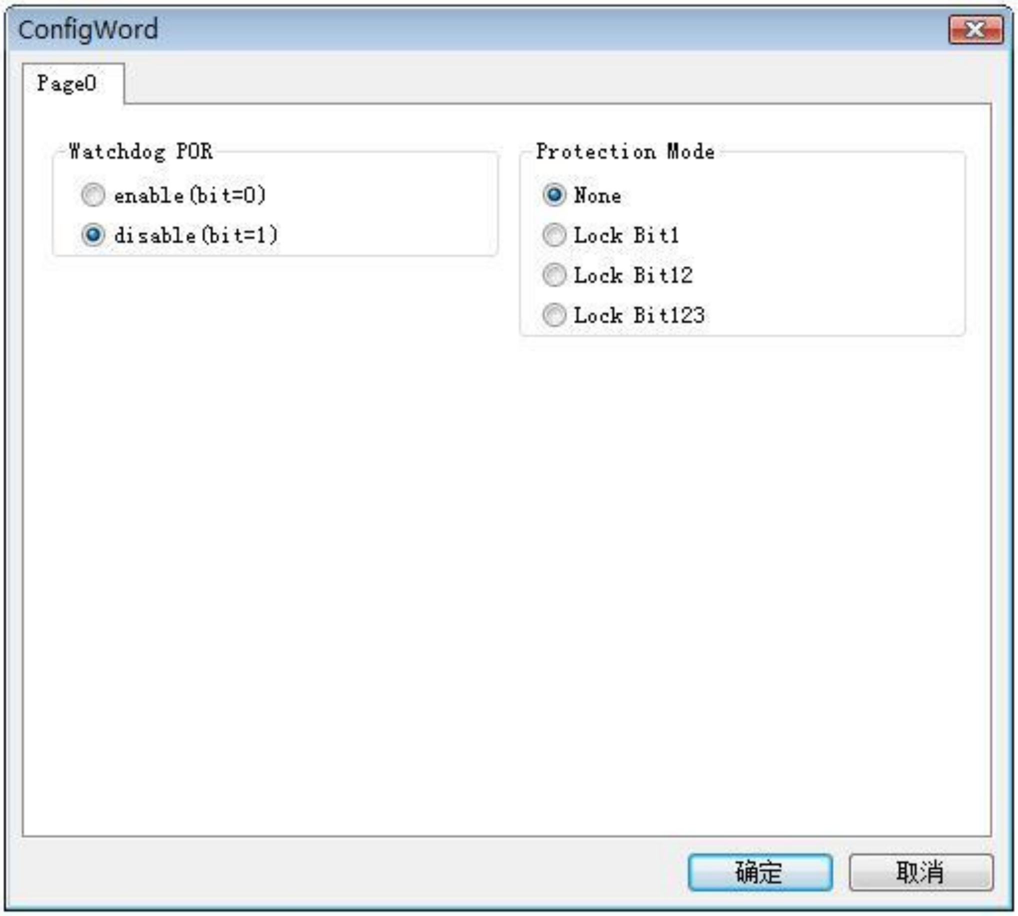
2. SUPERPRO /6100 supports 144 pins while SUPERPRO /611S and SUPERPRO /610P only support 48 pins.

4.1.3.3 Device Configuration Word

In some single-chip microcomputers, some special programming modes are allowed to be set, such as the storage area mapping, the watchdog time, the clock, the encryption, etc. They are all set through the device configuration word. The configuration words of some devices are contained in the user files and are automatically loaded into the configuration word buffer along with the data files when the files are loaded. As for the loading of the configuration words of the other devices, some selections must be made manually.

The configuration word for DALLAS DS89C420 is shown in the below figure. It contains the watchdog and three protection modes.

As for some single-chip microcomputers, the configuration word has many items which may be placed in some separate pages. The items on each page should be set before a chip is programmed.



ConfigWord

Page0

Watchdog POR

○ enable (bit=0)
● disable (bit=1)

Protection Mode

● None
○ Lock Bit1
○ Lock Bit12
○ Lock Bit123

There are several ways to input the configuration word: the 8-bit (byte) edit box input, the 16-bit (word) edit box input, the single choice input (select and deselect, choose one out of two), the multiple choice input (choose one out of many).

The configuration words are classified into two categories when chips are programmed:

1. If the configuration word contains some protected items, then the items can not be verified after the program is prepared. The programmer automatically masks the protected items and writes the other items while programming. The user needs to use the operation function item “Security or Protection” to write the protected items.

Note: If “None” is selected in the protection option, no Security operation is carried out even if you run the operation function “Security or Protection”.

2. For some devices, configuration words can not be written separately. In order to write the configuration word to the chip, the user needs to run the corresponding operation function, such as “Write_Option”.

The device configuration word varies from device to device. The user should refer to the data manual of the devices for the details and set the configuration words according to the requirements of the target system. Otherwise, the device may not work in the circuit although the chip seems to be programmed and verified well.

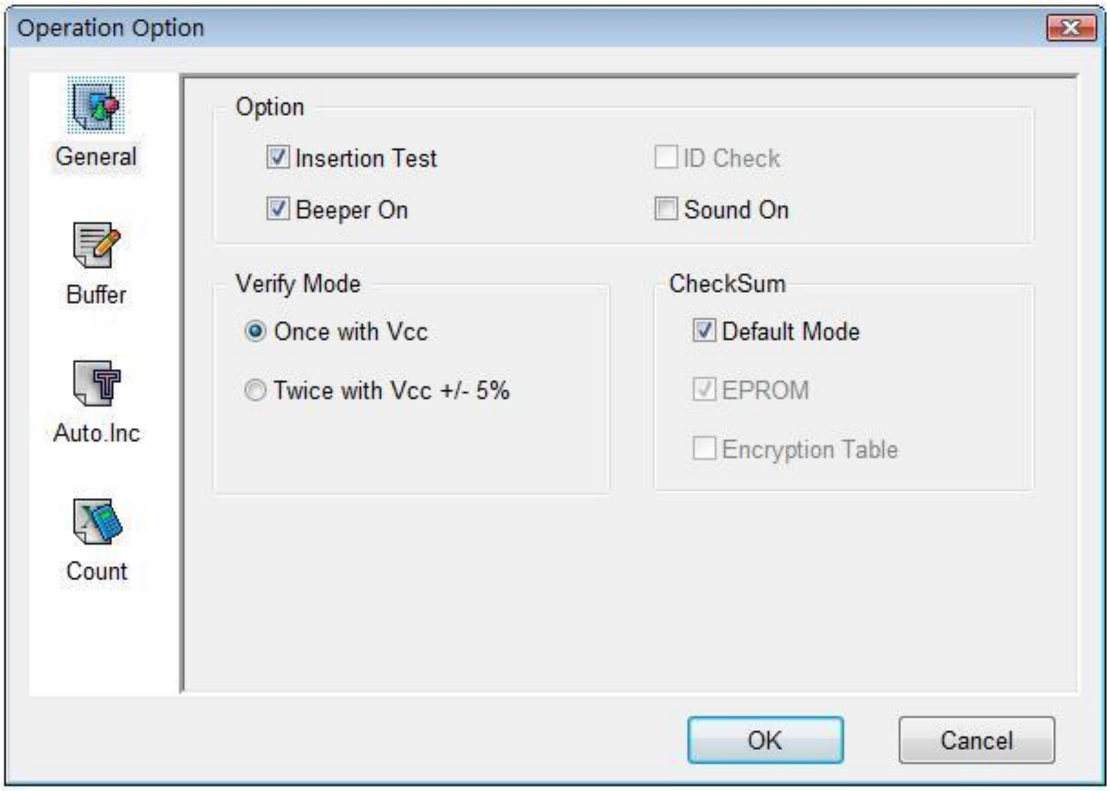
Some FLASH devices also use the configuration word to operate on the operation function item “Protect” and to execute and display the segment protection. Refer to “Window of Function Items for Operations on Devices”.

4.1.4 Options

This function helps users to program in a specific mode to meet the targeted system requirements. The necessary settings must be completed before chip programming.

4.1.4.1 Operation Option

This option includes:



Operation Option

Option
☑ Insertion Test
☐ ID Check
☑ Beeper On
☐ Sound On

Verify Mode
● Once with Vcc
○ Twice with Vcc +/- 5%

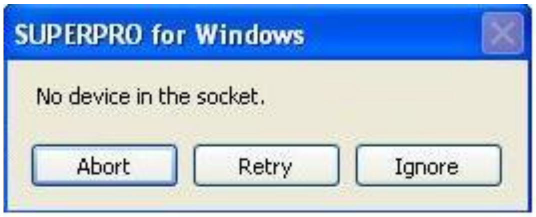
CheckSum
☑ Default Mode
☑ EPROM
☐ Encryption Table

- Insertion Test

Only for chips with 48 pins or less (include the chips with more than 48 pins which are transformed to 48 pins or less by an adapter).

After this option is selected, the programmer will check the pin contact status before programming, including check for poor contact, chip insertion orientation error, etc. The cases include:

Case 1:

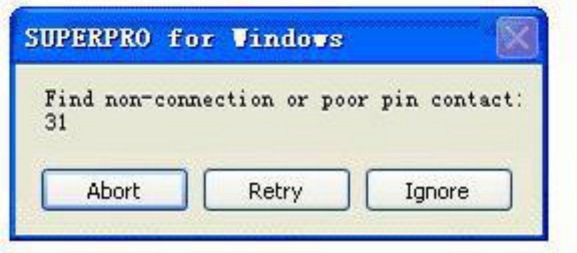


SUPERPRO for Windows

No device in the socket.

No device inserted in the socket. Please insert a chip.

Case 2:



SUPERPRO for Windows

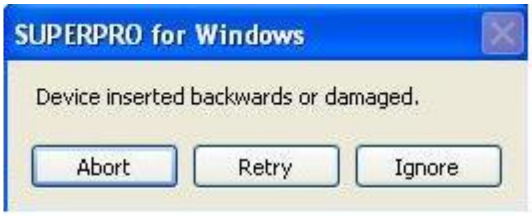
Find non-connection or poor pin contact: 31

If the chip has been inserted, the pin contact is poor or some pin(s) of the chip has been damaged.

In the case of the diagram above, for example, Pin 31 has trouble. The resolution is:

Check the pin contact first. If the contact is nothing wrong, please replace the chip with another to ensure whether the former chip is broken or not. The pin number is referred to DIP chip's pin number. If an adapter is used, it is referred to the adapter's pin number.

Case 3:

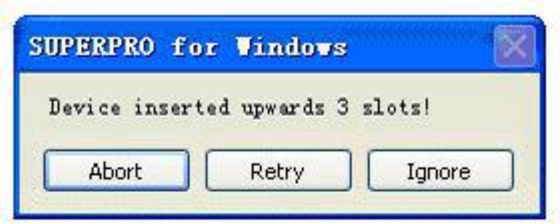


SUPERPRO for Windows

Device inserted backwards or damaged.

The chip has been inserted backwards. Please take out the chip, reverse it and insert again properly. If the problem still exists, the chip may have been damaged.

Case 4:

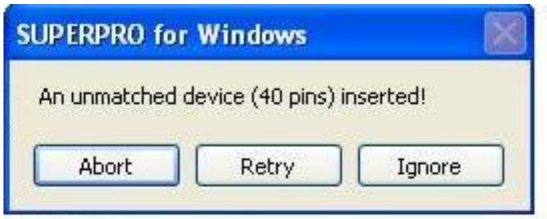


SUPERPRO for Windows

Device inserted upwards 3 slots!

The chip has been inserted improperly. It has been inserted upwards. Please refer to ‘Device Insertion’ in the manual.

Case 5:



SUPERPRO for Windows

An unmatched device (40 pins) inserted!

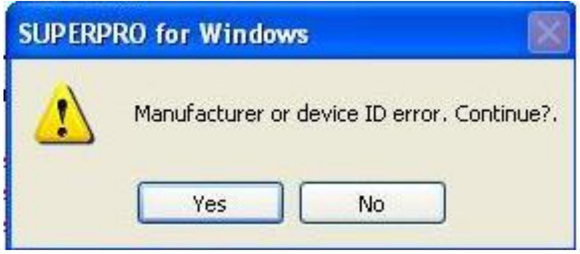
An unmatched device inserted.

If a chip has more than 48 pins and the pin contact errors are found, the serial number of the poor contact pin is displayed. What we need to emphasize here is that the serial number is the serial number of the dip socket of the programmer module which is to be connected with the adapter, not the serial number of a chip pin. It is not necessary for the user to know which pin the poor contact occurs. The information is used as a hint to tell the user to insert the chip properly.

If some errors are found by the 'Insertion Test', the user can click 'Abort' to stop the programming, click 'Retry' to run Insertion Test again, or click 'Ignore' to continue to program ignoring the errors.

- Device ID Check

The Device ID (Electronic Identifier Code) is the code read from the chip to identify the manufacturer, the device type and the programming algorithm code. If the 'Operation Option' box is opened after the device is selected, the option "Check Device ID" is activated. This means the device ID will be checked before programming. If the device ID is unmatched after this option is selected, an error window will popup.



Manufacturer or device ID error. Continue?.

Click Yes to ignore the error and continue programming.

Click No to abort the programming. When the ID is unmatched, the software will auto search for matched devices and display it in the error window for reference.

- Beeper On

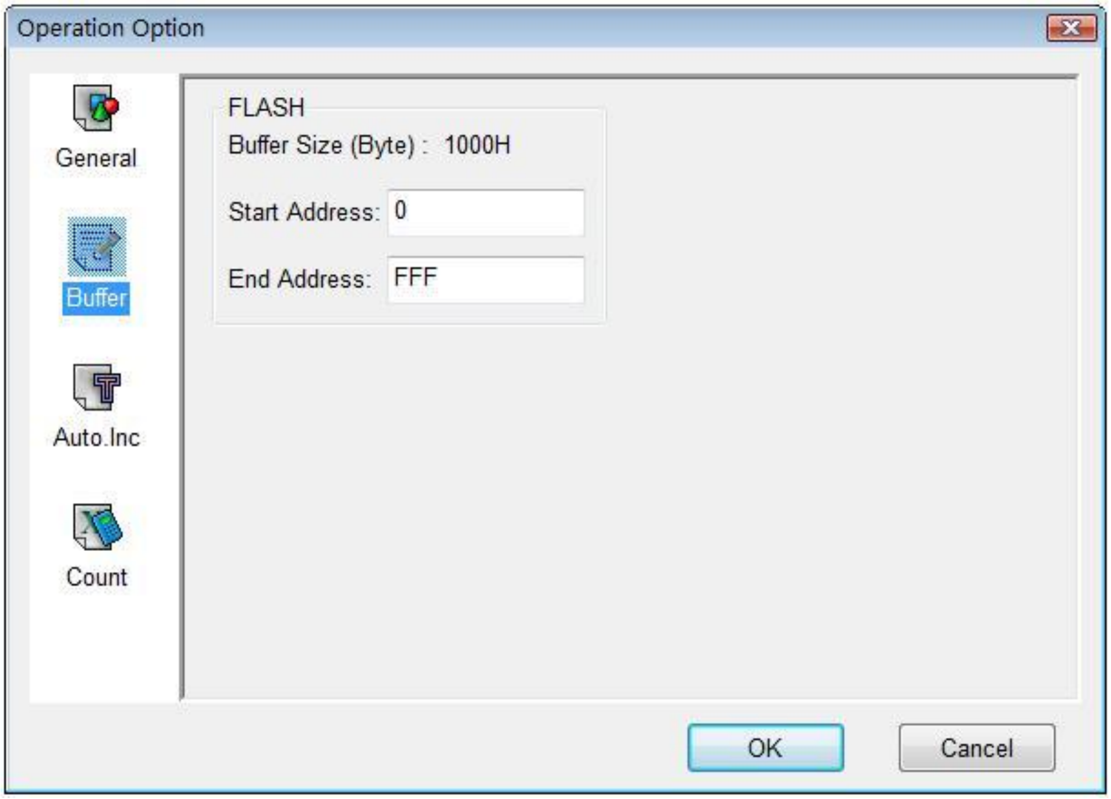
During the operation the beeper in the programmer will sound off to indicate the result of operation, such as Insertion Test error, ID unmatched, programming successfully or failed etc. This option allows user to turn on or off the beeper.

Note: Whether a programmer has a beeper is related to the programmer type.

- Verify Mode

After programming a chip, it is necessary to verify whether the data are correct. According to the documentation of the manufacturers about the programming, the voltage of the pin VCC may vary when the data are verified:

1. VCC(+/- 5%) or VCC(+/- 10%) is used for the verification. If VCC = 5.00V, select VCC = 5.00V and verify the data once, or select VCC = 4.75V and VCC = 5.25V, and verify the data twice (+/- 5%), or select VCC=4.50V and VCC=5.50V, and verify the data twice (+/- 10%).
2. The MinVcc and MaxVcc are used to verify the data. The values of MinVcc and MaxVcc are provided by the manufacturers.



FLASH

Buffer Size (Byte) : 1000H

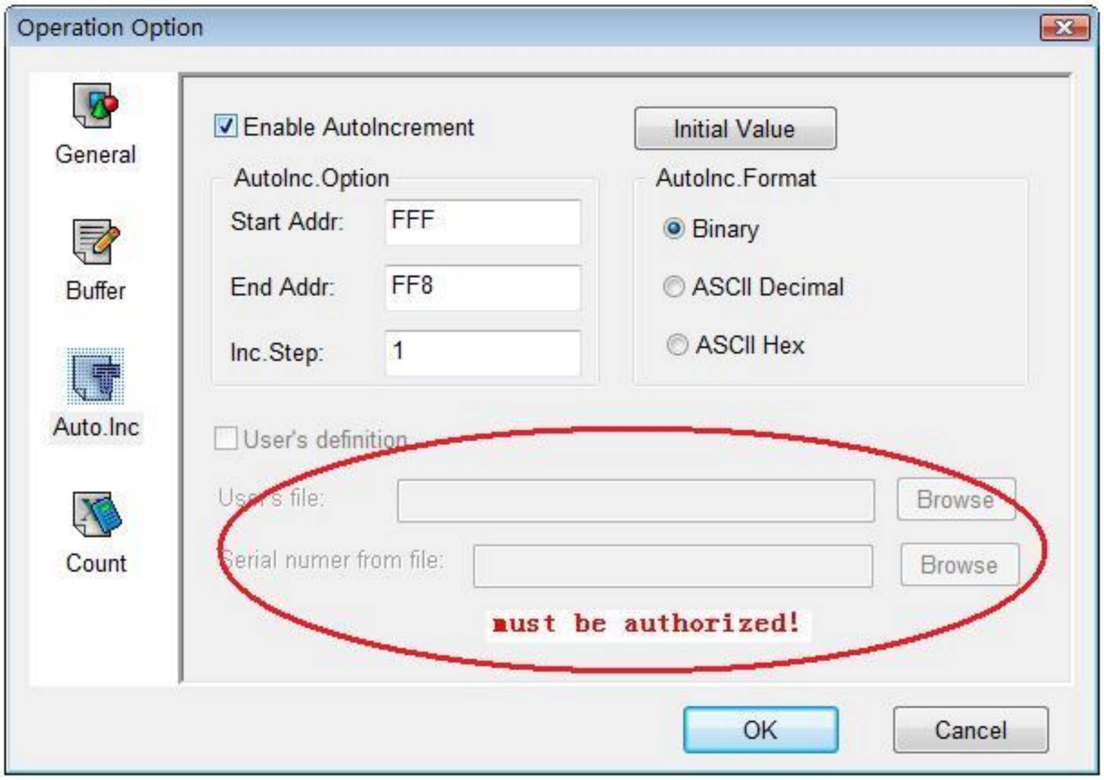
Start Address: 0

End Address: FFF

- Buffers

The user may choose to program only part of the chip. This applies to most E/EPROM (FLASH) devices.

Take the device MACRONIX MX29F200B TSOP48 for example. It is a 16-bit flash device. Its programmable zone starts at the address 0 and ends at the address 1FFFF (hexadecimal system). The buffer size (the unit is byte) is (1FFFF+1) * 2 = 40000 (hexadecimal system). Because it is a 16-bit (2 bytes) device, the buffer size is twice as large as the device size. The default programming method is to program the device from the start address to the end address. If the user wants to program only the second half part of the device, change the start address to 10,000 (hexadecimal system) and leave the end address unchanged. The programming data correspond to the data from the address 10000 * 2 = 20000 (hexadecimal system) to the address 40000 (hexadecimal system) in the buffer.



☑ Enable AutoIncrement

AutoInc.Option

Start Addr: FFF

End Addr: FF8

Inc.Step: 1

AutoInc.Format

● Binary

○ ASCII Decimal

○ ASCII Hex

□ User's definition

Users's file:

Serial numer from file:

must be authorized!

- Auto Increment of the serial number

Some applications require that the different contents (such as the product sequence number, the MAC address, etc.) should be written in a certain area for every chip. The content is called the serial number. This software presents two solutions to generate the serial number. The first solution is to use the auto increment method. The software generates the serial number by adding the step value given by the user to the last serial number. The format is fixed and it meets the general application requirements. The second solution is to use the user customized software to generate the device serial number. It has great flexibility.

The specific implementation is to write the current serial number to the buffer location specified by the user with a default range of 8 bytes (It can not exceed 4K).

As for the standard increment method, the following should be specified before the auto increment method of the serial number is used:

The automatic serial number option: that is, specify a continuous area.

- Set the Automatic Start Address,
- Set the Automatic End Address,
- Set the increment value as a number less than 10,

The address here is referred to the buffer address.

Auto Increment of the Serial Number: define the overflow condition, from the start address, the unit is byte (max decimal number is 255), When does the increment cease and carry to next address?

While a value reaches the overflow value, carrying to next address (+1), this address becomes the start value.

Binary: the start value is 0; the overflow value is 256.

ASCII Decimal: the start value is 30 (30 is a hexadecimal number here, it is the character ‘0’); the overflow value 39+1 (39+1 is a hexadecimal number here and it is greater than the character ‘9’).

ASCII Hex: the start value is 30 (30 is a hexadecimal number here and it is the character ‘0’); the overflow value 46+1 (46+1 is a hexadecimal number here and it is greater than the character ‘F’),
In the above, ‘9’+’1’=’A’.

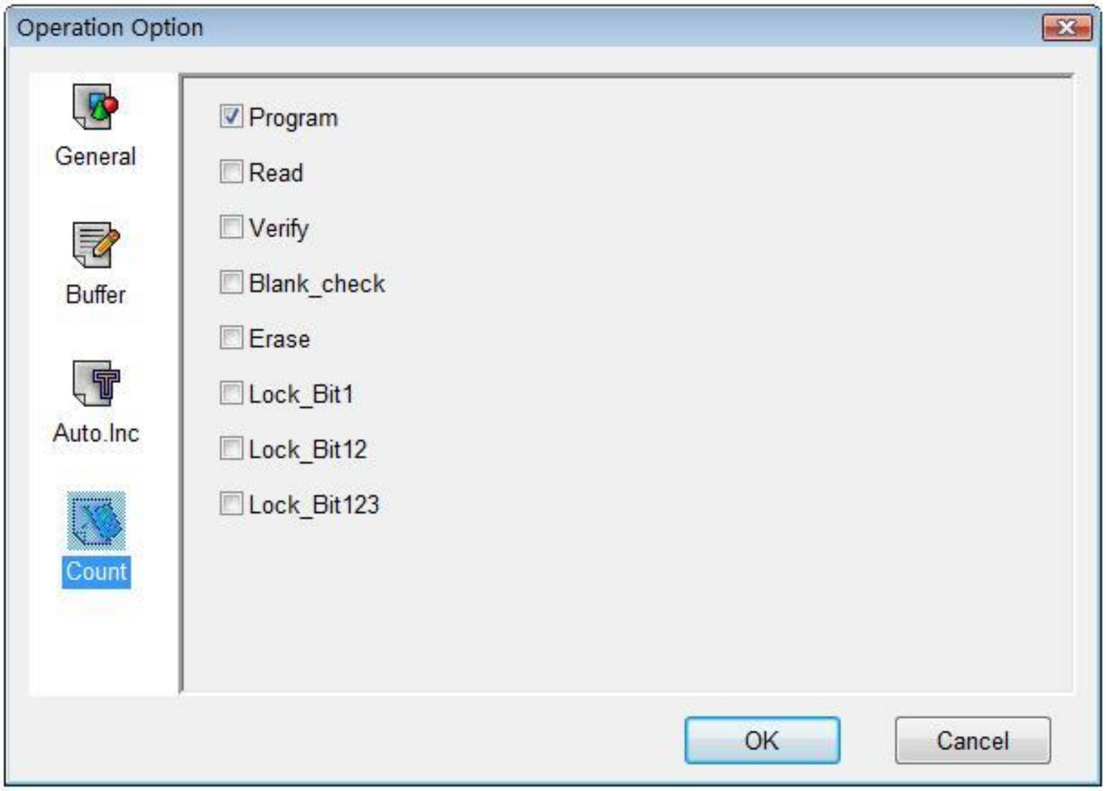
For example, there are 64 chips. Their IDs are from 0001 to 0064, which are in the end address of 89C51 EPROM. The user selects the mode of 4 bytes and the decimal system. The steps are listed as below:
Click the ‘Auto Increment’ option.
Set the Start Address as FFF (Hex) and set the End Address as FFC (Hex). The data start to change at the Start Address. The Start Address must be greater than the End Address.
Set the IncValue as 1.
Set the Format to ASCII Decimal.
Move to the ASCII area in the Buffer Edit window and make the following change:
FFC: 30 FFD: 30 FFE: 30 FFF: 31

During the programming, the following is displayed in the Operation Info window:
Current AutoIncrement Data=0001
Then in the subsequent programming operations, the information in the last 4 bytes of the buffer data is:
FFC: 30 FFD: 30 FFE: 30 FFF: 31

If the programming succeeds, the AutoIncrement increases by 1 and the following is displayed:
Current AutoIncrement Data = 0002
The last 4 bytes of the buffer data is:
FFC: 30 FFD: 30 FFE: 30 FFF: 32

- The user-defined auto increase of the serial number

The user can use this function only after he is authorized. The user can change the data as he wishes in the range of the 4K bytes if he can use the function.



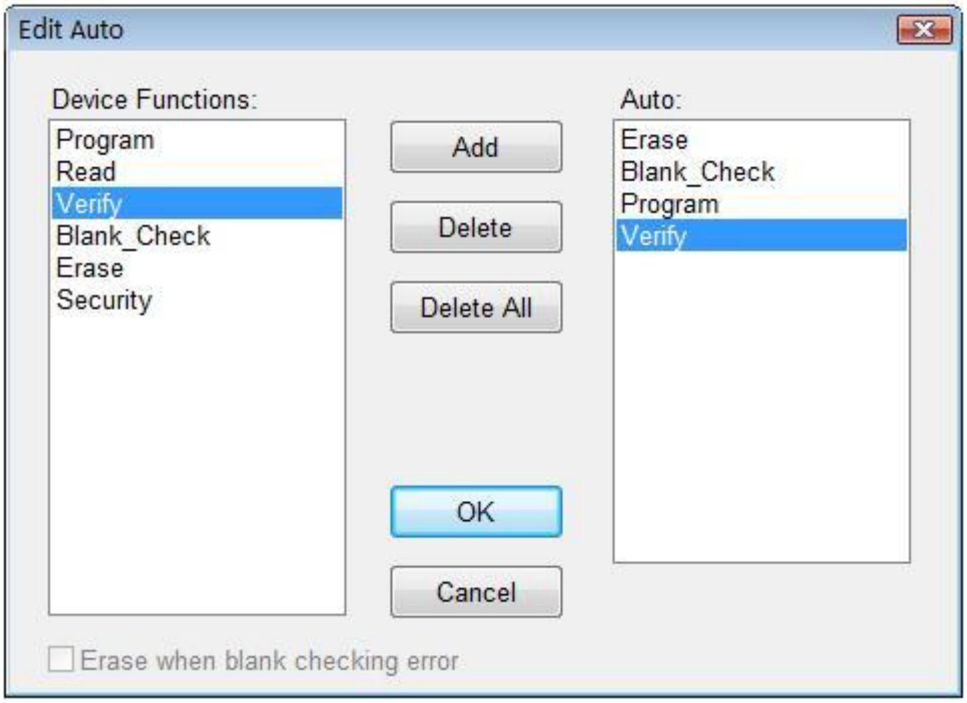
☑ Program
□ Read
□ Verify
□ Blank_check
□ Erase
□ Lock_Bit1
□ Lock_Bit12
□ Lock_Bit123

- When you program the chip, the software will conduct the statistical activities about the programming status automatically or according to the user's setting. It is only effective for the function "program" by default. You can change the default setting in the above window.

Note: Some minor changes should be made for the above description if the devices in the Operation Option are selected in some ways. For example, most single-chip microcomputers do not allow the user to change the start address of the device. The auto serial number function can not be used in the stand alone mode.

4.1.4.2 Edit Auto

In the window of "Function Items for Operations on Devices", all devices have an Auto option. The Auto function is used to organize the different functions of the device into a sequential group and carry out the functions according to the order. It is just similar to a batch command. For example, select ATMEL AT89C51, open Edit Auto dialogue box. On the left of the box is Device Functions column in which all device operations are displayed. On the right of the box is the Auto column in which the operations to be performed and the operation execution order are displayed.



In the diagram above, the operation execution order is Erase, Blank_check, Program, Verify, Lock_Bit123.

- The 'Add' button: add the highlighted function in the left column to the end of the right 'Auto' column.
- The 'Delete' button: delete the highlighted function in the 'Auto' column.
- The 'Delete All' button: remove all the functions from the Auto column.

Double clicking the function in the left 'Device Functions' column is the same as clicking the 'Add' button when the function is highlighted.

Double clicking the function listed in the 'Auto' column is the same as clicking the 'Delete' button when the function is highlighted.

4.1.4.3 Production Mode

This function is applied to program devices in large quantity. With the help of the Insertion Test, the programmer automatically detects the insertion and taking-out of the devices and carries out the whole programming process. In this mode, the user does not need to use mouse or keyboard.

If a non-support information window pops up, that means the selected device does not support production mode.

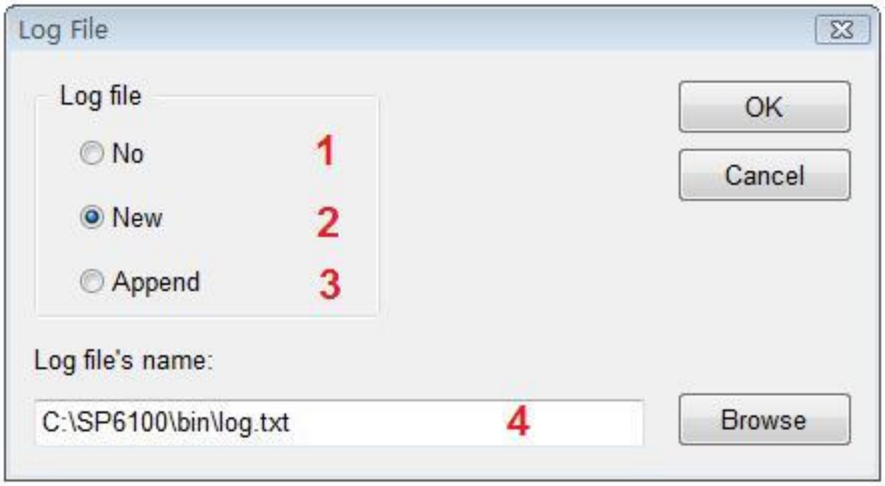
Our advice: First select the device and edit the 'Auto' menu. After the test is conducted, select production mode (tick off the item in the menu), and execute the 'Auto' function according to the info displayed in the 'Operation Info' window.

- The programmer waits for the chip insertion: 'Please, insert a device into the socket' prompts the user to insert the chip into the socket properly. If the chip has been inserted incorrectly, refer to the “Device Pin Insertion Test”. After the chip is placed properly, the software automatically starts the ‘Auto’ operation.

- The programmer waits for the taking-out of the chip. “Please, remove the device from the socket” prompts the user to take out the chip because programming has been finished. At this moment, check the programming result, check whether it has succeeded or failed, according to the info in the ‘Operation Info’ window. To end the production mode, please click ‘Cancel’ button on the right side of status bar.

4.1.4.4 Log File

Save the information in the Operation Information window in a file. In the execution process of the program, some content and data are written to the file when the program exits.



Log File

Log file

○ No
● New
○ Append

Log file's name:

C:\SP6100\bin\log.txt

1. No saving. If the file exists, it will be deleted.
2. Every time the program is started, the old records in the file will be deleted.
3. Every time the program is started, the new records will be added after the old records.
4. The full path name of the log file. It can be edited.

4.1.5 Project

The project file is a file in which all the current working environment is saved. It includes:

- The current device's info, such as the manufacturer name, the device type, the information about the pin, the size, the current buffer data and data file name. Since the buffer data may have been revised, the buffer data may differ from the data of the data file and the revised buffer data prevail.
- All Operation Option settings.
- The content of the ‘Auto’ mode

The project saves all the preparations before the programming. The user can restore the saved working environment.

Note: The project content is related to the programmer software. If the software has upgraded or has been re-installed to another folder, the saved project files become ineffective.

4.1.5.1 Load Project

Load the specified project file by selecting the file in the File dialogue box. According to the information in the project file, replace the device and change the buffer data, reset the operation options, etc.

4.1.5.2 Save Project Files

Save the current working environment to a specified project file. To ensure the security of the data, the user can encrypt the project file. Refer to "Security and Password Input".

4.1.5.3 Downloading Library (Standalone)

Under standalone mode, the programmer is directly used to program the chips by using the information saved in its own module with no connecting to a computer. It is applicable for the volume-production. With the help of the insertion contact test, the user does not need to know the detail about the use of the programmer, what he needs to do is just to insert and take out the chip correctly.

Before the standalone operation is started, the corresponding standalone file should be prepared for the programmer with the standalone operation function. Based on the project file, all the information needed for programming the device is saved in the standalone file so that it can be transferred to the storage module of the programmer hardware.

NOTE: SUPERPRO /6100 and SUPERPRO /611S provide this function. SUPERPRO /610P doesn't support offline mode.

How to create a downloading library?

- Create a project file. Under the standalone mode, the programmer can program the devices of the same model according to the different requirements, or program different devices. The maximum number of the programming methods that the programmer supports depends on the storage capacity of the programmer hardware. The selection of the programming method depends on the information saved in the project file.

For example, two Atmel AT89C51 chips, they are going to be programmed with two different files. If the storage capacity is enough, add an AMD AM27C128. The operation steps are as follows:

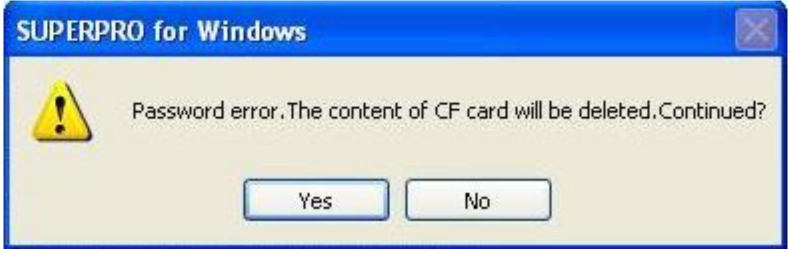
Step One: select Atmel AT89C51, and load the first file for programming. Edit the 'Auto' function, adding the 'Erase', 'Blank_check', 'Program', and 'Verify'. Then execute the 'Auto' function. After the programming operation is finished, check the chip in the target system. Save the current working environment to the specified file 'sample1.prj'.

Step Two: Load the second file for programming. Add 'Lock_Bit123' to the Auto function. Then execute the ‘Auto’ function. After the programmed chip has been tested successfully, save the current working environment to the specified file ‘sample2.prj’.

Step Three: Select AMD AM27C128. Load the file for programming. Edit the ‘Auto’ function, adding the ‘Blank_check’, ‘Program’, and ‘Verify’. Then execute the ‘Auto’ function. After the programmed chip has been tested successfully, save the current working environment to the specified file ‘sample3.prj’.

- Create a downloading library and download.

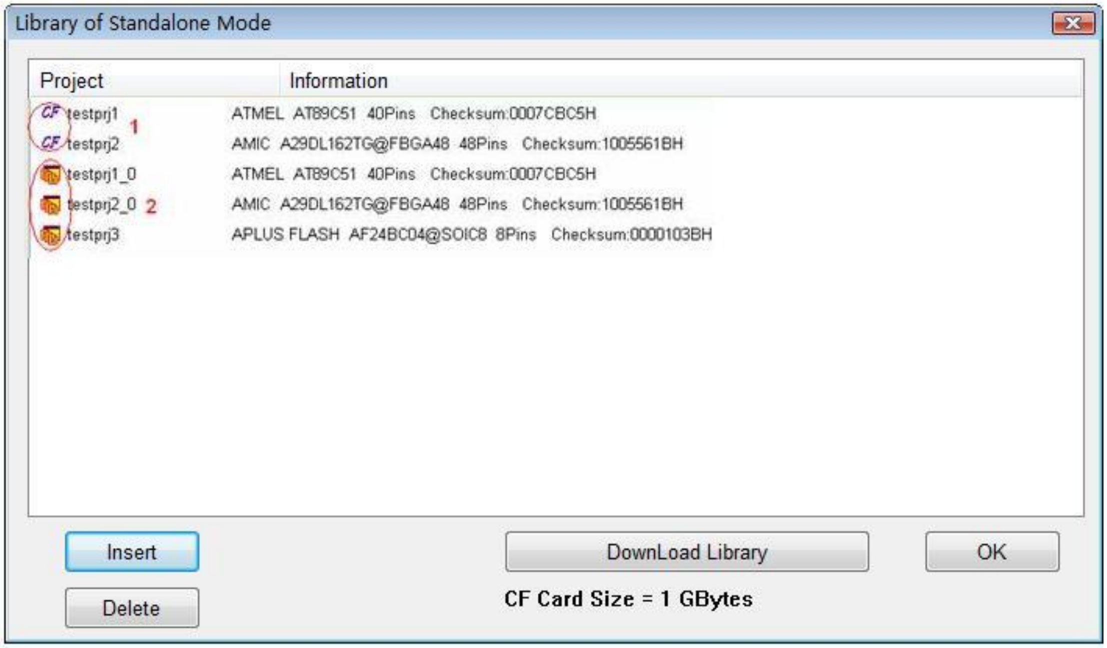
Click “Project” in the menu and select the sub-menu “Standalone”. The program, at first, check whether there are any data in the storage module. If there are some data in the module, it reads the data position allocation table. Meanwhile, it checks the data security and prompts the user to input the password if the password protection is required.



Password error.The content of CF card will be deleted.Continued?

Password error. The data in the storage module (CF card) will be deleted permanently. Select ‘Yes’ to continue.

Then the ‘Library of Standalone Mode’ window pops up.



Project	Information
CF testprj1	ATMEL AT89C51 40Pins Checksum:0007CBC5H
CF testprj2	AMIC A29DL162TG@FBGA48 48Pins Checksum:1005561BH
testprj1_0	ATMEL AT89C51 40Pins Checksum:0007CBC5H
testprj2_0	AMIC A29DL162TG@FBGA48 48Pins Checksum:1005561BH
testprj3	APLUS FLASH AF24BC04@SOIC8 8Pins Checksum:0000103BH

CF Card Size = 1 GBytes

1. The project in the storage module of which the data already exist.

2. The project to which new data are added.

It is necessary to connect the programmer to the PC first if the user wants to use the standalone function because the capacity of the standalone library is related to the storage module (CF card) of the programmer hardware. Before the dialog box appears, the storage capacity of the programmer hardware is displayed on the screen. In the above figure, ‘CF Card Size = 256 Mbytes’ means the storage capacity is 256 MB.

Click the ‘Add’ button to add the three project files to the library. Click the ‘Download Library’ button to start the downloading. In order to ensure the security of the data, the user can set some password to protect the data. Refer to “Security and Password Input”.

Wait until the downloading is finished.

Now the programmer is able to work under a standalone mode. Refer to the “Standalone Operation” for the detail.

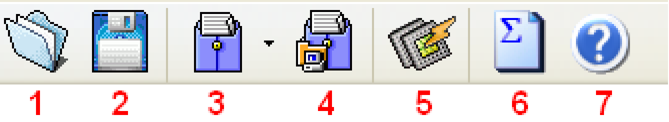
Error messages:

- “Please check connection of the programmer”: The programmer hardware is not connected or an error occurs when the storage capacity is read. In this case, the generated standalone library can not be used.
- “Library size larger than capacity of CF card”: The added project needs a larger storage capacity. Please increase the storage capacity (CF card capacity).
- “Download library OK”: The library has been downloaded into the programmer hardware successfully and now the programmer can be used in the standalone mode.
- “Download library error”: The library has failed to be downloaded into the programmer hardware. Check whether the CF card is normal.

4.1.6 Help

4.2 Tool Bar

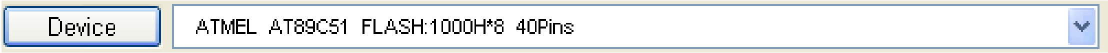
The tool bar provides a quick way to execute the common functions. The corresponding relations are as follows:



1. Load 2. Save
3. Load Project 4. Save Project
5. Factory Mode (refer to “Detail Explanations on Common Problems)
6. Calculating the checksum (refer to “Detail Explanations on Common Problems)
7. Help

4.3 Device Selection and Device Information

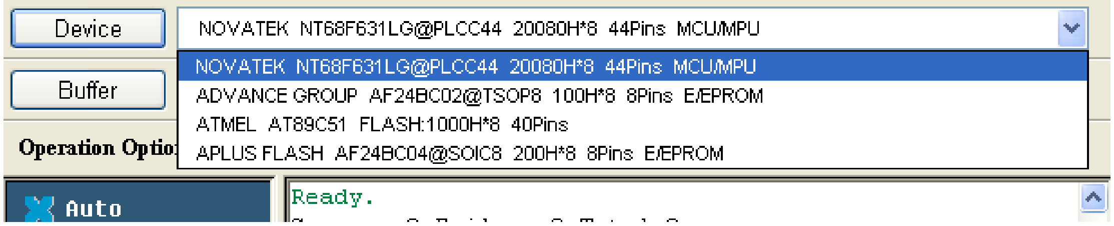
Open the ‘Device Selection’ dialog box quickly. The information about the recently selected devices is displayed (the manufacturer name, the device name, the size, the number of the pins, the device types).



ATMEL AT89C51 FLASH:1000H*8 40Pins

The function of the button “Device” is the same as that of the menu "Device".

The device information displayed in this bar includes the historical records. It is allowed to re-select the device displayed in the bar. By using this bar, users can only change the device selection, including manufacturer, device name and buffer capacity, the number of the pins, the device types etc. It is different from loading project file to change the work status. Loading a project file changes the working environment.



NOVATEK NT68F631LG@PLCC44 20080H*8 44Pins MCU/MPU

NOVATEK NT68F631LG@PLCC44 20080H*8 44Pins MCU/MPU
ADVANCE GROUP AF24BC02@TSOP8 100H*8 8Pins E/EPROM
ATMEL AT89C51 FLASH:1000H*8 40Pins
APLUS FLASH AF24BC04@SOIC8 200H*8 8Pins E/EPROM

Ready.

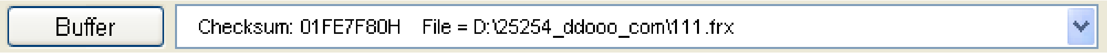
After a selection is made, a dialog box appears and the user should confirm the selection.

- “Select ATMEL AT89C51. Continue?”

Then the user confirms.

4.4 Edit the Buffer and File Info

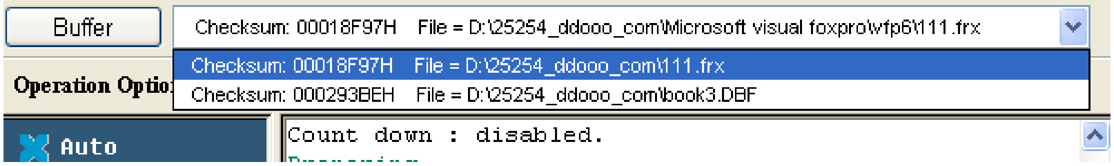
Quickly open the ‘Buffer Edit’ dialog box and display the information about the file loaded recently (the checksum and the full file name (including the full path)).



Checksum: 01FE7F80H File = D:\25254_ddooo_com\111.frx

For the details about the function of the ‘Buffer’ button, refer to sub-menu “Edit” of the menu “Buffer”.

The file information includes the historical records and can be used to re-load the file. Since the selected device may be different when the file is loaded, that may cause the data to be lost, the checksum to be different, the file type to be unmatched, etc.



Checksum: 00018F97H File = D:\25254_ddooo_com\Microsoft visual foxpro\vfp6\111.frx

Checksum: 00018F97H File = D:\25254_ddooo_com\111.frx
Checksum: 000293BEH File = D:\25254_ddooo_com\book3.DBF

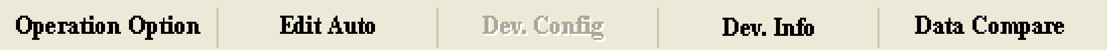
Count down : disabled.

After the selection is made, the following dialog box will pop up.

- “Reload the file d:\25254_ddooo_com\111.frx in the buffer. Continue?”

4.5 Special Device Info and Operation Option Bar

Before the programming, users should obtain the special device info and set the operation options properly according to the targeted system's requirements. The toolbar below will facilitate user to finish this work. The user can also find the corresponding operation in the menu.



- Operation Option: Refer to the menu “Option”.
- Edit Auto: Refer to the menu “Option”.
- Device Configuration Word: Refer to the menu “Option”.
- Special Device Info: Refer to the menu “Device”.

Data Compare: This function is applicable to most memory chips and SCM chips. Data in the buffer and the chip will be compared byte by byte. The different data and the address in the buffer and the chip memory will be recorded and saved in file in the Bin directory. The file name and path will be displayed in the Operation Info screen.

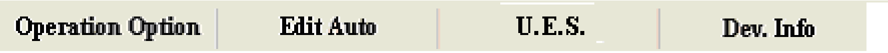
A data comparison example:
Compare result in file:
C:\SP3000U\BIN\27256.CMP
The file format is shown in the following:
Title : SUPERPRO Data Compare Result file.

```
Displacement Buffer address        Chip Data          Buffer Data
------------------------------------------------------------------
000000D0                               04                 12(EPROM)
000000D1                               00                 34(EPROM)
000000D2                               00                 56(EPROM)
000000D3                               00                 78(EPROM)
000000D4                               00                 90(EPROM)
000004A5                               00                 AA(EPROM)
000004A6                               00                 BB(EPROM)
000004A7                               00                 CC(EPROM)
```

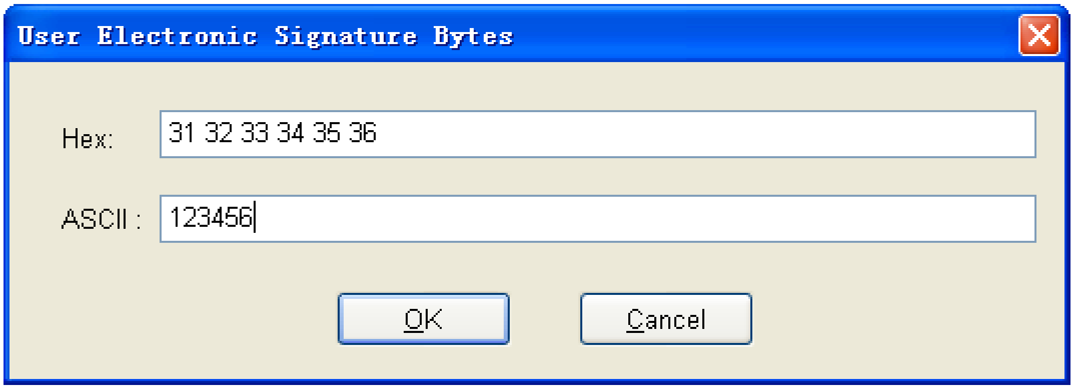
- Displacement Buffer address
- Chip Data
- Buffer Data

Note: If there are many buffers, the buffer names (such as EPROM) will be added. However, most buffers do not have the names.

If a PLD device with a U-Field is selected, the Operation Option Bar will look like the following:



U.E.S.: Edit user electronic signature. This content may be changed during reading the device or loading files. Users can check and edit U.E.S. (two display methods, Hex and ASCII).



User Electronic Signature Bytes

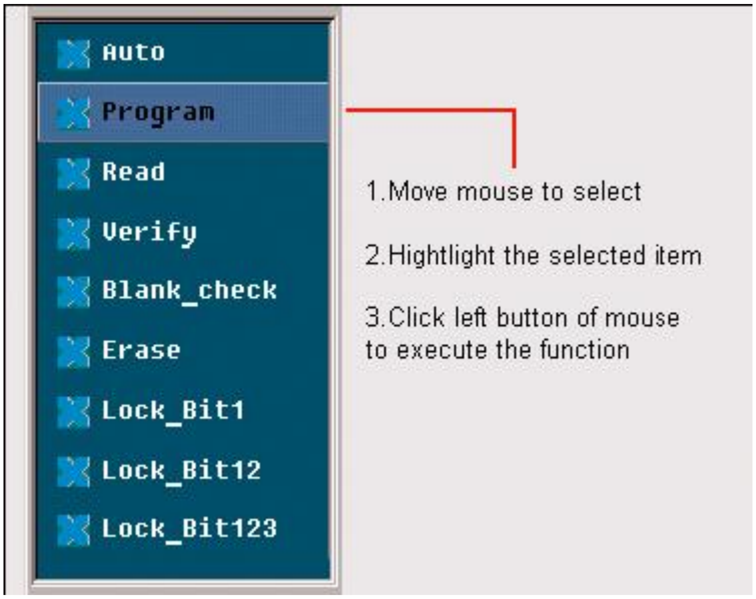
Hex: 31 32 33 34 35 36

ASCII : 123456

4.6 Device Operation Window

According to the references provided by the manufacturer, the operations, which can be executed during the programming, are displayed in this window. After the preparation for the programming is made, users can execute the corresponding operation items according to their needs.

Following is the window for Atmel At89c51.



1.Move mouse to select

2.Hightlight the selected item

3.Click left button of mouse to execute the function

The function displayed may vary from chip to chip. The frequently-used operations are explained below:

- Auto: The automatic programming mode. The function executes a series of operations on the device in a defined order. Please refer to the “Edit Auto” in the menu.
- Program: Programming the chip with the data in the buffer. The function writes the data from the buffer into the chip. The Verify function is performed in two modes. One mode is verifying while programming. This method basically ensures the correctness of programming after programming a chip is finished. If an error occurs, the error message will be displayed and the process will be ceased. The

other method is no verifying during the programming. Even if an error occurs, the process will not be ceased. After the data for the last address are written, the successful programming message will be displayed. Anyway, the successful programming message only means that the programming process is finished, the correctness is dependent on the result of verification.

- Read: The function reads the content of the chip into the buffer. To view the content read, please open the Buffer Edit dialogue box.

Note:

1) For the encrypted chips, their content cannot be read.
2) For the chips with the ConfigWord, the ConfigWord will be read and saved to the ConfigWord buffer.
3) For PLD devices, their test vector isn't in the chip, unable to read it.

- Verify: The function compares the content of the buffer with that of the chip. If there is any discrepancy, verification will cease and failure message will be displayed. For most memory chips and SCM chips, the failure message includes address and data. For PLD devices, the message displayed depends on the programming reference. Only the chips that pass verification can be recognized as the chips being programmed correctly.

Some chips do not provide the cell to cell check function, but only provide the accumulation check function similar to the VerifyCRC. Few chips even do not provide the accumulation check function and the check is carried out automatically in the programming process.

- Blank_check: This function checks if the device is in blank state, which is dependent on the data manual. If filling the blank data buffer is selected when the devices are selected, the programmer will fill all the HEX/ASCII buffer data with FF or 00 (Hex), and the fuse buffer with 1 or 0 after the device is selected. An encrypted chip may be displayed as a blank chip. EEPROM (the chip can be overwritten) doesn't need Bank_check.

How to make a chip blank?

1) For the chip that can be erased electrically, execute the 'Erase' operation.
2) For the chip that can be erased with ultraviolet ray, use the ultraviolet ray eraser for 3-30 minutes.
3) For OTP, once written it is unable to be written over back to blank again.

- Erase: Erase the chip and make it blank. This operation can only be executed on the chip that can be erased electrically. EEPROM doesn't need to execute this operation. If you need a chip of blank state you can directly write the blank data to the chip. Some chips that can be erased electrically are able to become an OTP by configuring the ConfigWord. Please refer to the device data manual.
- Protect:

1) Encrypt the data to make them unreadable. Whether the protection state is readable or not, affects the editing of the operation function. When the protection state is unreadable, the 'Protect' operation will be added to the operation function window. As long as the 'Protect' function is executed, the chip will be protected. If the protection state is readable, the state 'None' and

all the ‘Protect’ options will be listed for user to select. The ‘Protect’ operation will be executed according to the ConfigWord. If the user selects ‘None’, the ‘Protect’ operation is equivalent to no operation.

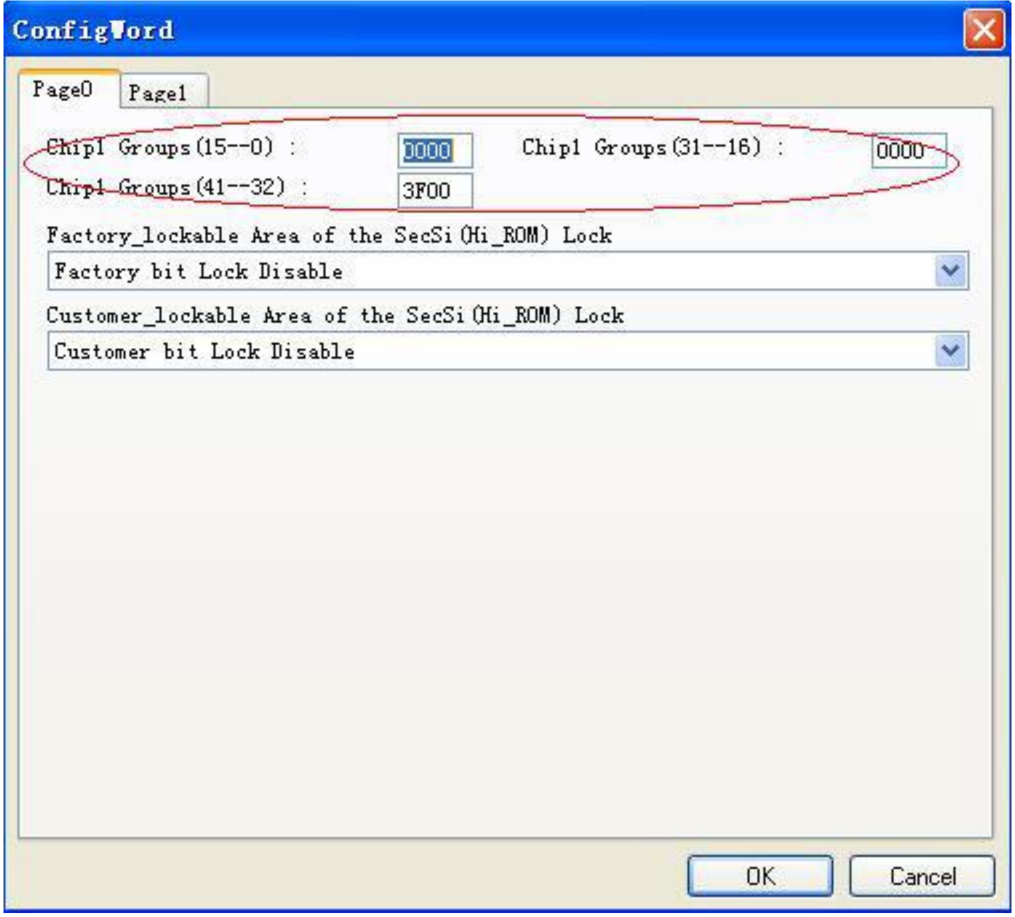
Note: If the chips made by some manufacturers are fully protected, the data can not be erased even if the ultraviolet ray eraser is used and therefore they can not be re-used.

2) For FLASH chips, this function is to protect the data in the chips. The protected chip is not re-writable before the protection removed. The Erase operation can remove the protection.

This operation is required to be used with Dev. Config. Users need to identify the Sector to be protected. There are two ways to select the Sector:

1. Users enter the address of several (max 8) sectors to decide the sectors to be protected. After the ‘Protect’ operation is performed, the sectors whose addresses are entered will be protected.

2. The chip has been divided into several sector groups in advance, so users can only choose these groups to be protected.



Chip1 Groups(15--0) : 0000

Chip1 Groups(31--16) : 0000

Chip1 Groups(41--32) : 3F00

Factory_lockable Area of the SecSi(Hi_ROM) Lock: Factory bit Lock Disable

Customer_lockable Area of the SecSi(Hi_ROM) Lock: Customer bit Lock Disable

- Security: The same as ‘Protect’, to encrypt the chip so that the content in it can not be read.
- Erease_All: The same as ‘Erase’.

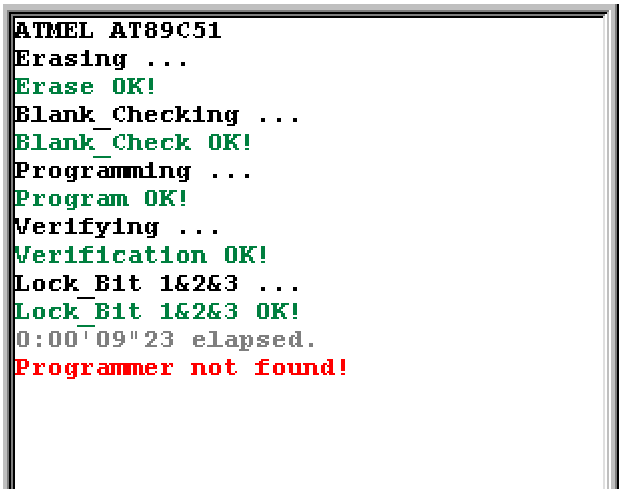
- Protect_All: The same as ‘Protect’.
- Lock_Bit: A protection method. Please refer to the Device Data Manual for the detailed information about the operations. Take Atmel At89c51 as an example, it has three operation items: Lock_Bit1, Lock_Bit12, Lock_Bit123. Their meanings can be found in the data manual.

The above are commonly used functions. For some devices, which have special requirements and functions, the operation function window may be different. For detailed information, the user could refer to:

- The manual of the device data.
- The Dev. Information.
- The technical support of XELTEK.

4.7 Operation Info Window

This is for the interaction between the software and the user. The operation process, the operation result and the historical records of information, etc. will be displayed in this window.



```
ATMEL AT89C51
Erasing ...
Erase OK!
Blank_Checking ...
Blank_Check OK!
Programming ...
Program OK!
Verifying ...
Verification OK!
Lock_Bit 1&2&3 ...
Lock_Bit 1&2&3 OK!
0:00'09"23 elapsed.
Programmer not found!
```

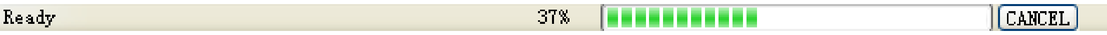
The above is an example. The example appears after the device is selected and the operation "Auto” is performed. The meanings are explained as follows:

- ATMEL AT89C51: The device selection is correct. If any error occurs, please refer to “Correction according the error information”.
- Erasing.....: is erasing…..
- Erase OK!: Erased successfully.
- Blank_check......: Blank check is being performed.
- Blank_check OK!: Successful Blank_check.
- Program.....: is programming.
- Program OK!: Program successfully!.
- Verifying......: is verifying.
- Verifying OK!: Verify successfully.
- Lock_Bit 1&2&3....: protecting. Refer to the Data Manual for AT89C51, the functions of Lock_Bit123 can be found. They are as follows:
- MOVC instructions executed.

- Verify is disabled.
- External execution is disabled.
- Executing one function will finish several operations. The user should select the function according to the manual in order to avoid misoperations.
- Lock_Bit 1&2&3 OK!: Successful protection.
- 0:00'09"23 elapsed: Executing Auto takes 9.23 seconds.
- Programmer not found: the red words mean that errors have occurred. Read the error information.

4.8 Status Bar

This bar displays the menu item information, the progress of operation, and a Cancel button.

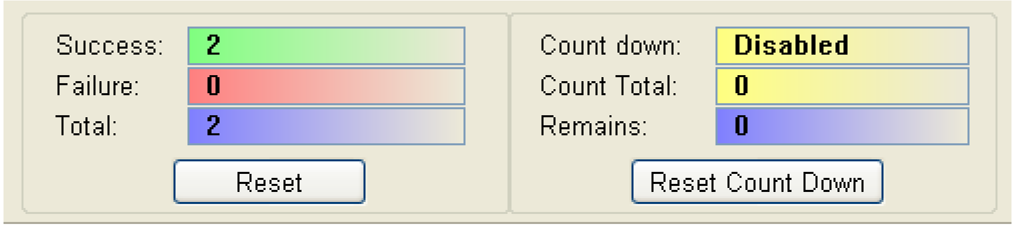


If the 'Cancel' button is clicked, the operation which is being performed will be stopped and the following dialog box will pop up:

- Cancel process. Continue?

Before the user confirms the 'Cancel' operation, the operation continues. The 'Cancel' operation may destroy the chip, please use it carefully.

4.9 Statistic Window



Success: 2
Failure: 0
Total: 2

Count down: Disabled
Count Total: 0
Remains: 0

- Statistical Information

The times of successful and failed programming are recorded. The default counting rules:

1. When "Program" is executed, if it succeeds the success count increases by 1, otherwise the failure count increases by 1.

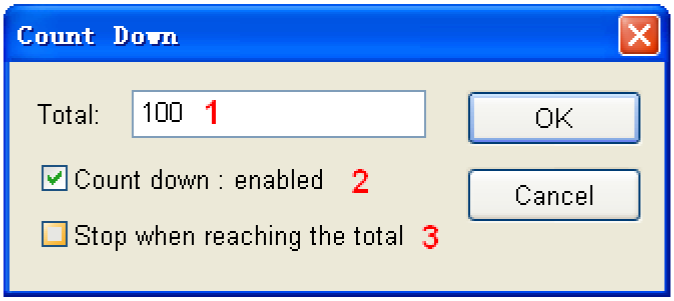
2. When "Auto" is executed, if the "Auto" contains "Program" and the "Program" succeeds the success count increases by 1, otherwise the failure count increases by 1.

Except for the operations, all the operations are not related to the special status bar. Refer to "Operation Option".

In the above figure, the success count is 2 and the failure count is 0. If the "Reset" button is clicked, the success count and failure count are all set to 0.

- Count Down

It can be set as follows:



Count Down

Total: 100

☑ Count down : enabled

□ Stop when reaching the total

1. The count down number (greater than 0). The number decreases by 1 every time a chip is programmed successfully. It continues till the number becomes 0.
2. The count down function is enabled.
3. The action to be conducted when the count down operation ends. When it is disabled, it is not allowed to proceed to write a program until the user re-sets the option.

Chapter 5 Detailed Explanations on Common Problems

Using Operation Info window and message window, the user is able to monitor what the programmer is doing, what it will do, or what problem it had so that the user can stop the operation. This chapter will help the user to deal with some common troubles according to the information mentioned above.

5.1 Dealing with Data Files

The detail usage of the “file type” dialogue box is described here.

5.1.1 Intel Hex, Motorola and Tektronix formats

The data for programming are usually stored in the data file. Commonly used formats are Intel Hex, Motorola and Tektronix. Refer to the menu “Loading files”.

The files of these formats save data in text mode, which includes data info and address info. The offset address is not always zero. If the user loads the file directly, some errors may occur. For example, the location of data in buffer may be incorrect (this can be viewed in “Edit Buffer” window) or even the file can not be loaded and some message like the following is displayed:

- “Invalid file type, or all file data overflow”

Solutions:

1) Confirm the file type and ensure the buffer being able to load all the files.

2) Find the offset address of the file. If it’s difficult for the user, he can select the option “Display the minimum start address after the file is loaded” to show the start address of the file after the file is loaded. Thus the start address of file will be displayed after the file is loaded. For example:

File Offset Address (min): 0x0000E0000 (Hex)

This means the offset address of the file is not 0. The user needs to move the data at the address to the address 0 of the buffer. To do that, the user should reload the file and enter E000 in the file start address edit line in the File Type dialogue box.

5.1.2 File Data Need to Be Separated for Programming

By using the File Load Mode in the File Type dialogue box, users can select the data of a file and write the data to several chips of the same type. The operation mode is to load the file multiple times according to the needs.

The first method: Based on byte (8bit), divide the file into two parts by the odd and even address, and program the two parts into two chips respectively. When the file is loaded, the user selects "Even (taking the first byte out of every two bytes)" in the File Load Mode, the data in the address 0,2,4,6.... will be written to one chip. The user selects "Odd (taking the second byte out of every two bytes)", the data at the address 1.3.5.7.... will be written to the other chip.

The second method: Based on the byte (8-bit), take the first byte, the second byte, the third byte and the fourth byte out of every four bytes, write the file data to four chips respectively. When the file is loaded, the user selects the first byte out of every four bytes in the File Load Mode box, the data at addresses 0, 4, 8, 12 (it is C in HEX).... will be written to one chip. The rest can be reasoned out by analogy.

The third method: Based on the word (16-bit), write the former two bytes and the latter two bytes of every four bytes to two chips respectively.

5.1.3 Program Two or More Files to One Chip

In this mode, you will need to load several files into the buffer first, and then write them to the chip. The following example shows you how to program three files to a chip.

In this example, some data from the files Sample1, Sample2 and Sample3 are written to one chip. The data from the address 200 (Hex) of the file Sample1 will be written to the address 0 of the chip. The data from the address 0 (Hex) of the file Sample2 will be written to the address 3000 (Hex) of the chip. The data from the address 4000 (Hex) of the file Sample3 will be written to the address 4000 (Hex) of the chip.

Step one: Open 'Buffer Edit' dialogue box, leave the option 'Clear Buffer During File Loading' unselected.

Step two: Click "Load file", Select file Sample1. In the "File Type" dialog box, enter 0 in the buffer start address line and 200 in the file start address line. Load the file "Sample1". Select the file Sample2. Enter 3000 in the buffer start address line and 0 in the file start address line. Load the file "Sample2". Then select the file Sample3. Enter 4000 in the buffer start address line and 4000 in the file start address line. Load the file 'Sample3'.

Step three: Program files

The user can also save the data in buffer to a file Sample4, then directly load Sample4 and program next time if there is no change about requirement of this file and content of the three files.

5.1.4 File of POF Format

When the PLD devices made by ALTERA are programmed, the data are saved in the files of the POF format. When a file is loaded, a ‘File Type’ box will pop up to prompt the user to select.

1) The POF file, which is created by a compiler, will be transformed into the Fuse data and be checked when loaded to buffer.

- Unmatched file to this chip: this means this POF file does not match this chip. The POF file is related to the device when it is compiled. The error indicates that this POF file is not for this device. Please replace it with a matched one.
- The POF file has errors:
- Load POF file dll error (not found): An auxiliary file has errors; please contact technical support.

2) Files of other formats are mainly the JED files. The data in the buffer will be directly saved in a JED file after the chip is read. This mode can be used when the file is loaded next time.

5.2 Adapter Selection

The programmer has a standard configuration. It supports the DIP devices with 48 or less pins. If the user wants his programmer to support the device which is non-DIP or has more than 48 pins, an adapter is needed to convert the pins. There are two types of adapters, universal and specific. Universal adapters are applicable for all devices of same encapsulation and the same number of pins. Specific adapters are applicable for certain devices only.

Why are specific adapters required?

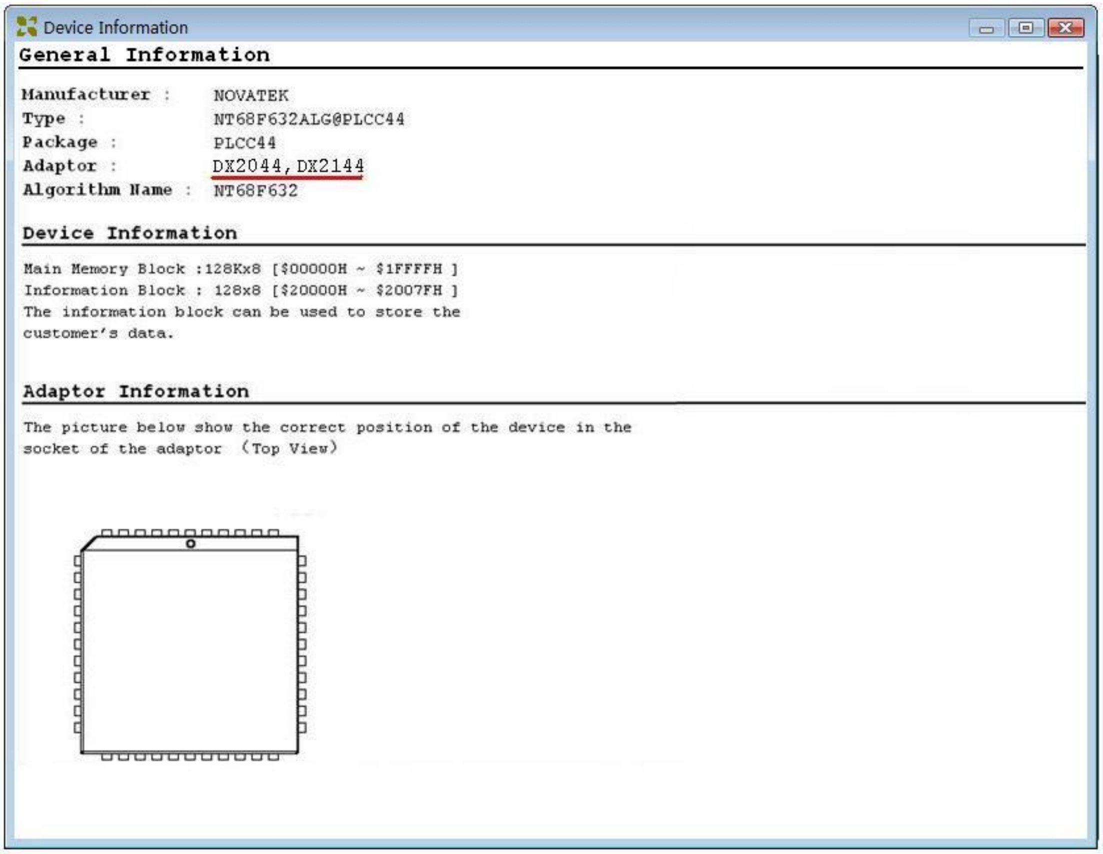
- If the number of the pins for an economical programmer hardware is not enough, some specific adapters are used to solve the “insufficient number of the pins” problem so that the devices with more than 48 pins can be supported by the economical programmer.
- For the devices with pins between 48 and 144, the user has two choices. One is the universal adapter and the other is the specific adapter.
- For the devices with more than 144 pins, the specific adapters must be used.

Why a type of adapters has several order numbers?

- The sockets are from different manufacturers, and
- There is minor difference in size.

How to get a correct adapter?

Refer to the “Dev. Info”. Take PLCC44 as an example. The adapter order number is DX2044 or DX2144.



Device Information

General Information

Manufacturer : NOVATEK
Type : NT68F632ALG@PLCC44
Package : PLCC44
Adaptor : DX2044,DX2144
Algorithm Name : NT68F632

Device Information

Main Memory Block :128Kx8 [$00000H ~ $1FFFFH]
Information Block : 128x8 [$20000H ~ $2007FH]
The information block can be used to store the customer's data.

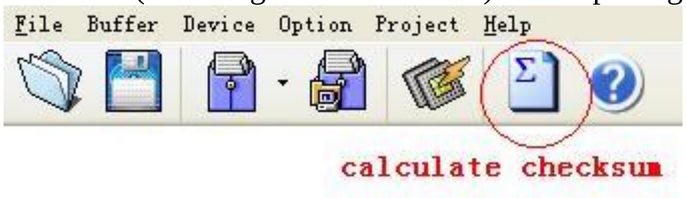
Adaptor Information

The picture below show the correct position of the device in the socket of the adaptor (Top View)

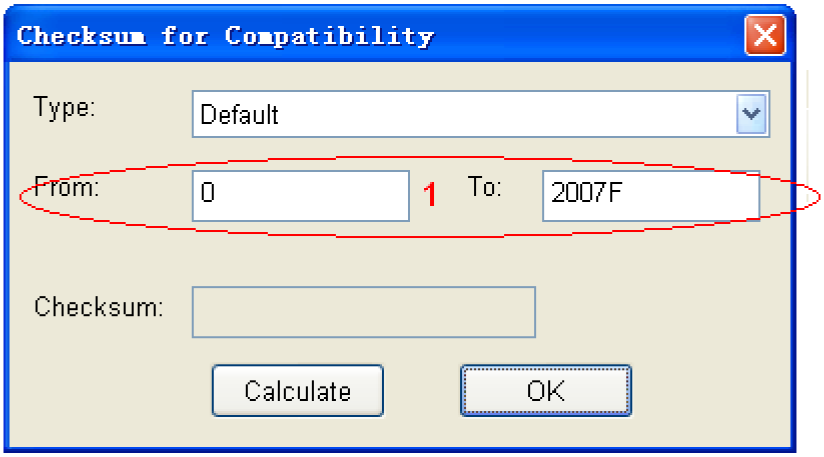
Note: Please do not use self-made adapters, because the success rate of making an adapter for the programmer by the user is very low. It is recommended to buy the adapter made by Xeltek.

5.3 Checksum

When the data type of the data buffer is 'Data' (HEX/ASCII data), the default calculation for the checksum is to put together all the data from Address 0 to the maximum address byte by byte. To meet the user's requirements, the data at some addresses (including all the addresses) can be put together in a specified way.



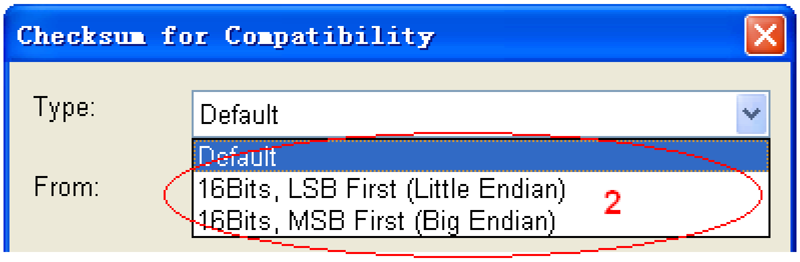
Select the other checksum calculation method:



Type: Default

From: 0 To: 2007F

Checksum:



Type: Default

From:

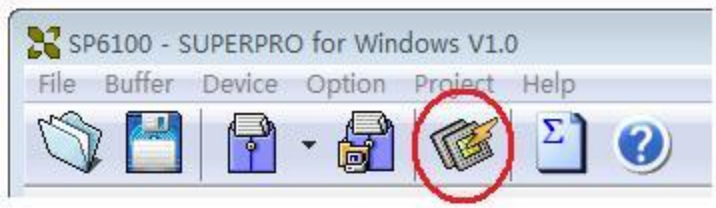
- Default
- 16Bits, LSB First (Little Endian)
- 16Bits, MSB First (Big Endian)

1. Change the calculation range
2. Select the accumulation method

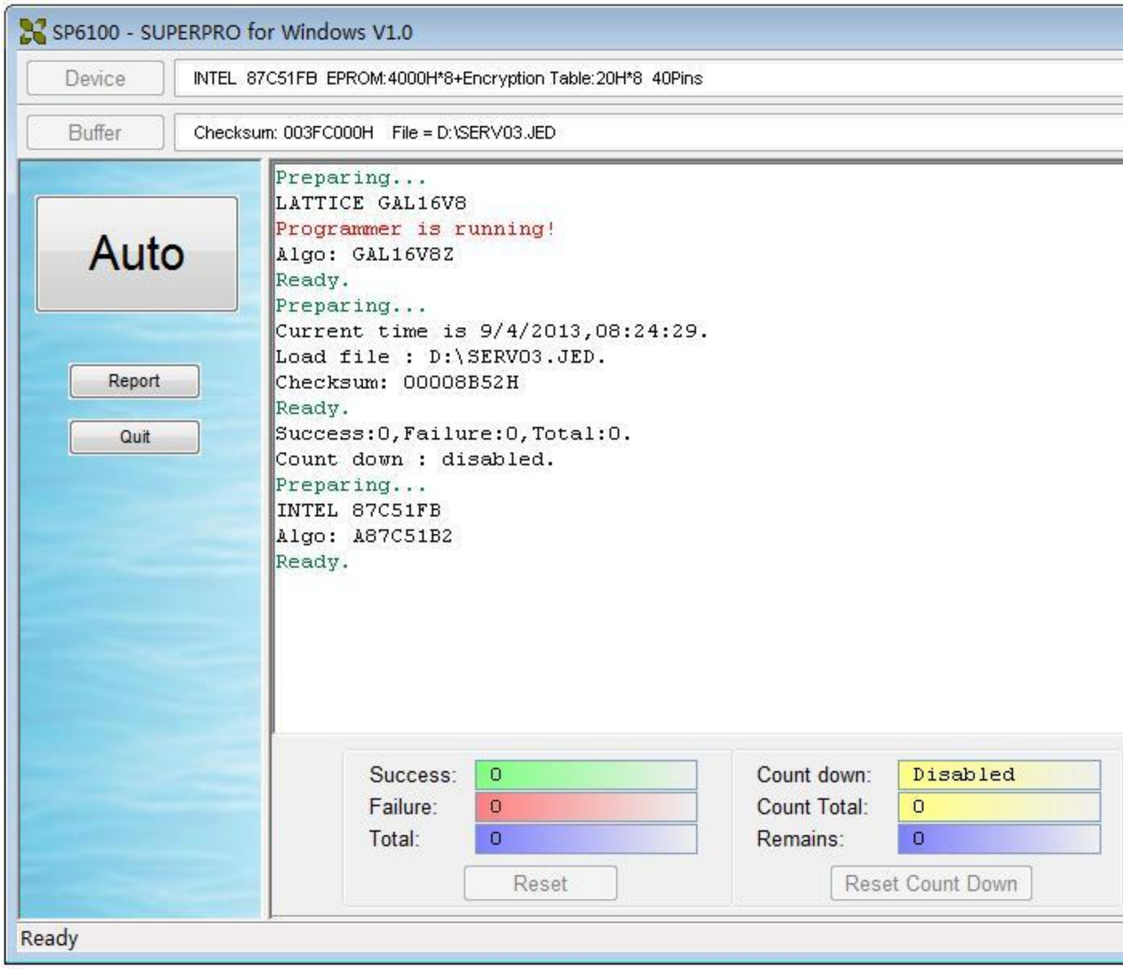
5.4 Factory Mode

This mode is designed for the volume production in the factory. After entering this mode, only a few functions such as "Auto", etc. are allowed by the program to prevent some misoperations from destroying the chips or prevent wrong data from being written to the chip.

The administrator can set a password for entering the system to ensure the data security. If the password has been set, the password also needs to be entered when the user exits the system.



Factory Mode



SP6100 - SUPERPRO for Windows V1.0

Device: INTEL 87C51FB EPROM:4000H*8+Encryption Table:20H*8 40Pins

Buffer: Checksum: 003FC000H File = D:\SERV03.JED

```
Preparing...
LATTICE GAL16V8
Programmer is running!
Algo: GAL16V8Z
Ready.
Preparing...
Current time is 9/4/2013,08:24:29.
Load file : D:\SERV03.JED.
Checksum: 00008B52H
Ready.
Success:0,Failure:0,Total:0.
Count down : disabled.
Preparing...
INTEL 87C51FB
Algo: A87C51B2
Ready.
```

Success: 0
Failure: 0
Total: 0

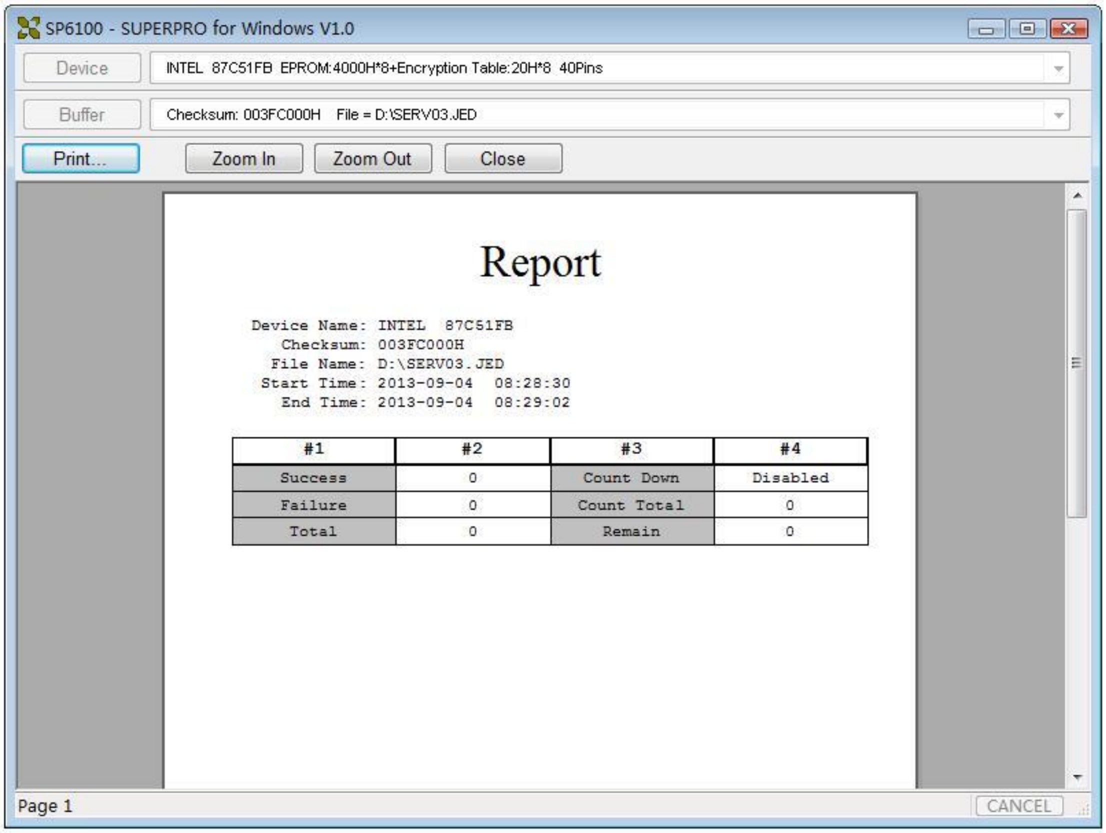
Count down: Disabled
Count Total: 0
Remains: 0

Ready

As shown in the above figure, only three buttons are active. Except the three buttons, no other operations are allowed by the program.

Auto: Execute the “Auto" programming operation.

Report: Generate the statistical report and display it (it can be printed on the paper of Size A4).



Device: INTEL 87C51FB EPROM:4000H*8+Encryption Table:20H*8 40Pins

Buffer: Checksum: 003FC000H File = D:\SERV03.JED

Report

```
Device Name: INTEL  87C51FB
   Checksum: 003FC000H
  File Name: D:\SERV03.JED
 Start Time: 2013-09-04  08:28:30
   End Time: 2013-09-04  08:29:02
```

#1	#2	#3	#4
Success	0	Count Down	Disabled
Failure	0	Count Total	0
Total	0	Remain	0

Quit: Exit.

5.5 Other Messages

Please edit ‘Auto’ first: the Auto consequence is blank; please edit Auto.

Enter a string to search for: The user needs to search for a specified string or some ASCII codes in the ‘Buffer Edit’ dialog box. But the ‘Hex-Edit Search’ dialogue box is blank. Please enter a string or ASCII code to search for.

The string for search is blank. Input it in Search dialog: use the search function in the ‘Buffer Edit’ dialogue box. “The first time search”, “Continue to search”, Search Next” can be used.

Search pattern not found!: no matched string or ASCII code is found.

User chip is MfgID = 0089, DevilD = 0051: The chip ID of the user. The chip has the ID detection function. After it is selected, if the detected chip ID is different from the specified ID in the manual, the detected ID will be displayed in the ‘Operation Info’ window.

ID check error. Ignore: The user ignores the detected ID error and continue the operation.

ID check error: The operation stops because of the ID check error.

The user can ignore the ID error and continue the operation. However, he must have a good knowledge of this chip; otherwise the chip may be damaged.

Pin check error. Ignore: error occurred during the pin check. Ignore it and continue.

Pin check error: An error occurred during the pin check, and the operation ceased.

Programmer not found: the programmer is not connected to the PC or connected improperly. Please connect it and refer to "Installation".

Programmer not ready: shut down the power switch and turn it on again after a few seconds.
Programmer is running: The programmer hardware is operating. Please wait until the operation is finished.

File open error: The 'open' file operation has failed.
Out of Memory: The memory overflows.

Production mode isn't available for this chip: Since the production mode is dependent on the pin check, the chips without pin check function are not applicable for production mode.

Cancel production mode: Cancel the production mode.

Too long file name: A too long file name is used.

User cancel: The user has used the 'cancel' operation.
Can't cancel: The 'cancel' operation has failed.
Not a project file: The selected file is not a project file. After the software is updated the software may not be able to identify the file created previously. Now the user needs to create the file again.

Appendix

XELTEK software only needs the minimum technical support. The program is handed over to the user along with a user manual. If the user can not find a solution to to his problem, he can contact the local sales agent, the distributor or XELTEK for technical help. If the local sales agent or distributor can not solve your problem, you can ask XELTEK for technical help through the telephone in the normal working hours (8:30 --- 17:30, except for the holidays and festivals). You should get your product serial number ready before you make a telephone call, otherwise we may not be able to answer your question.
XELTEK periodically publishes the upgraded software at the website of the corporation. The legal user can download it and upgrade his software. The non-user can download it too, but just can only be used for assessment. The software can be obtained free through the internet.

Xeltek Inc.
1296 Kifer Rd. Unit 605
Sunnyvale, CA 94086

Tel: +1 408-530-8080
Fax: +1 408-530-0096
Website: www.xeltek.com

To get better service, please browse the following contents before calling us:
1. If you think the program is difficult to understand, did you ever carefully study the user manual?
2. If the software reported an error, can you repeat this phenomenon?
3. If any error prompt appears, please write down the error message.
4. Obtain your product serial number before you ask for help through the telephone, fax or email.
5. Understand the configuration of your computer. We hope to know your computer brand, the free memory size before the software is started, the video adapter and the OS.
6. When you make a phone call to the Technical Department for technical help, you should be beside your computer so that our engineer can guide you to solve your problem quickly.

License Agreement

The copyright of the program and the ‘User's Guide’ remain the property of XELTEK.

You may:
1. Copy the program for back-up purposes ONLY in support of its use on a single computer.
2. Transfer the program and license to another party if the other party agrees to accept the terms and conditions of this agreement.

You may not:
1. Use this product in a computer system or network, which allows the program to be operated by more than one user at the same time.
2. Modify, copy, or transfer the User's Guide, other documentation or any copy.
3. Decompile or disassemble any program modules or encrypted devices.

Warranty

XELTEK has a strict product quality assurance system. If the software or hardware has any defects, it can be repaired or replaced free according to the specific situation within one year period after the user buys the product.
The warranty is based on the condition that the software is installed correctly and

used in the specified working environment.

XELTEK is not responsible for the following cases:

1. The products that have been repaired, used or changed by any organizations or persons who are not authorized by XELTEK.
2. The products that have been damaged because of misuse or careless use, or the products whose serial number has been changed by the user or some other person.
3. The program errors caused by the physical damage of the disk.