

XELTEK INC

**XELTEK**

---

# **SuperPro® 3070 Handbook**

Xeltek

12/15/2010

## Contents

Contents .....	0
Outline .....	2
What is SP3070 .....	2
General functions .....	2
Hardware functions .....	2
Software functions .....	2
System requirements .....	2
Standard package of the programmer .....	2
Hardware installation .....	3
Connection of ISP cable with the target system .....	3
Description of connection port .....	3
Definition of several common interfaces .....	5
Examples for connection of target system .....	6
SP3070 requirements for target system .....	7
Attentions on pin connection .....	7
Precautions for power: .....	9
Precautions for the design with reset generator .....	9
Examples of common problems .....	11
Other matters needing attention .....	12
Examples for connection of several common serial communication protocols .....	13
M25P64 .....	13
24C04 .....	13
S9S08AW16 .....	13
Installation of software .....	13
Quick start .....	14
User interface of application software .....	14
Steps for creating a project .....	15
Select device .....	15
Load data into buffer .....	15
Set options .....	16
Save project .....	16
Operate SP3070 on I3070 .....	17
Detailed description of functions .....	19
Browse menu .....	19
Load file .....	19
Save file .....	20
Exit .....	20
Load project .....	21
Save project .....	21
Select device .....	22
XPlayer .....	23
Buffer .....	23
Checksum .....	27
Device information .....	27
Device Configuration Word .....	27
Edit automatic processing command .....	28
Operation options .....	28
Tool bar .....	32
Main toolbar .....	32
Information bar .....	32
Device information .....	32
Buffer information .....	33
File information .....	33
Statistical Information Bar .....	33

## **Outline**

### **What is SP3070?**

SP3070 for Agilent's Medalist i3070 Series 5 is a cost effective programming module for memories, micro-controllers and programmable logic devices.

### **General functions**

- Programming speed is adjustable to be compatible with target board, length of ISP cable, etc.
- Supports online (USB2.0 high-speed) and offline mode simultaneously.
- Supports ISP programming of devices with I2C, SPI, UART, BDM, MON, MW, JTAG, CAN, ICC, RS232 and other serial protocols.
- Over-current and ESD protection.
- Includes stable and reliable design, such as R & D, mass production, in-system programming, etc.

### **Hardware functions**

- ISP cable accommodates I2C, SPI, UART, BDM, MON, MW, JTAG, CAN, ICC, RS232 interfaces.
- Cable driver circuit supports target voltage level of 1.5-5V and is also ESD protected.
- USB 2.0 interface for PC connection and management of project files on SD card.

### **Software functions**

- Supports 5,000+ serial devices.
- Very fast programming rate selectable in three gears: high, medium and low. Adaptable to different cable lengths and load features of user boards.
- Supports dynamic buffer function (each chip burns different contents including: serial number, MAC address, sensor calibration etc.). The interface is of open type.
- File format on SD card: FAT16, FAT32.
- Operates up to 12 programmers through one USB HUB. Chips and files can be different.

### **System requirements**

Minimum requirements for the system are as follows: a computer, desktop or portable computer compatible to Pentium or above; a universal serial bus interface to USB2.0 standard; Windows XP/Vista operating system; CD-ROM; hard disc with at least 1G free space; Agilent Medalist i3070 Series 5 with 08.00p Software Release.

### **Standard package of the programmer**

The standard package includes the following items:

- Programmer
- Connecting cable for USB interface
- Installation software- one CD-ROM
- User registration form

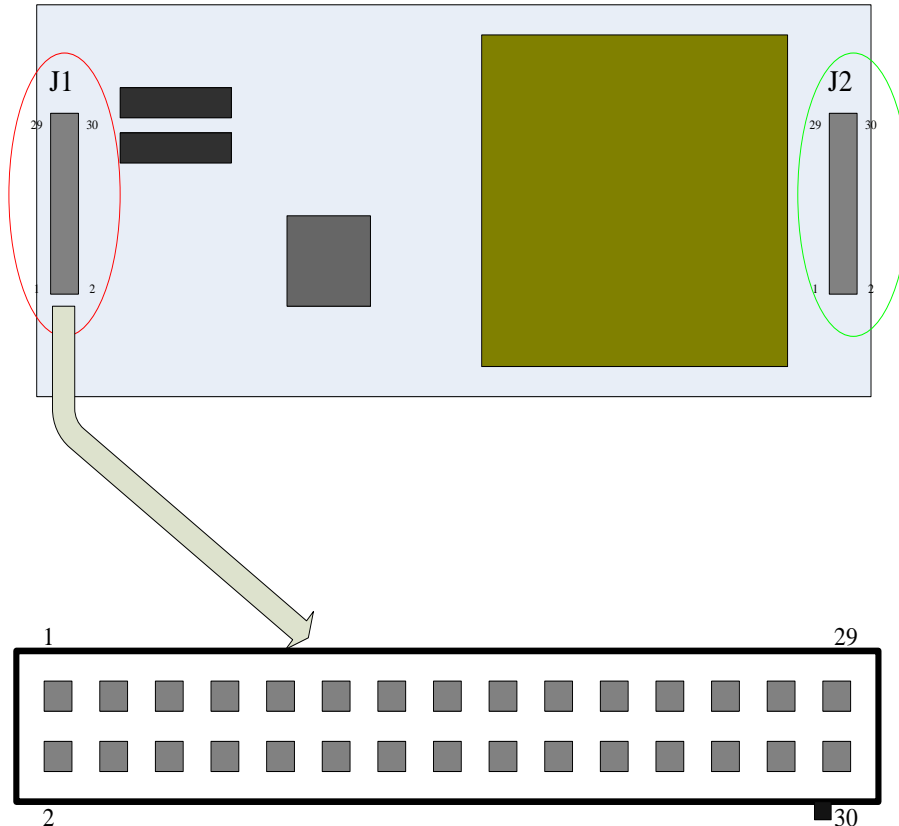
## Hardware installation

### Connection of ISP cable with the target system

#### Description of connection port

ISP connection port contains the signal ends required for programming of the target system. The connection port is divided into group J1 and group J2:

The programming interfaces required for most serial protocols are contained in Port A, e.g. SPI, I2C, JTAG, UART, BDM etc. The pins of this port are defined as follows:



Pin #	Signal Name	Description
1	VIO	Target Chip Voltage Level Input
2	DIO1	Digital input 1
3	DIO2	Digital output 2
4	DIO4	Digital output 4
5	RSV2	Reserved digital output 1
6	DIO3	Digital input 3
7	DIO0	Digital input/output 0
8	DIO7	Digital output 7
9	GND	Ground connect with chip ground
10	TPIN	target system power on detection(input)
11	NC	No Connect

12	DIO5	Digital output 5
13	NC	No Connect
14	RSV3	Reserved digital output 2
15	NC	No Connect
16	GND	Ground connect with chip ground
17	NC	No Connect
18	TPOUT	3.3V voltage output
19	NC	No Connect
20	ETXD	TX(output, RS-232 levels)
21	NC	No Connect
22	ERXD	RX(input, RS-232 levels)
23	GND	Programmer Electrical Ground
24	ECTS	CTS(input, RS-232 levels)
25	GND	Programmer Electrical Ground
26	ERTS	RTS(output, RS-232 levels)
27	+12V	Programmer Power Supply Input(+12V DC)
28	ECANL	CAN Bus difference signal for low
29	+12V	Programmer Power Supply Input(+12V DC)
30	ECANH	CAN Bus difference signal for high

**Notes:**

1. SP3070 has a working mode called “mass production.” Once the programmer detects that the target board is ready and the ISP signal arrives, it immediately starts the functional operation defined by AUTO without keyboard operation. TPIN and TPOUT in the port are provided for this purpose. TPOUT supplies high level and TPIN is level detection input line. Short-circuit them on the target board. Once ISP signal is applied to the test point, TPIN immediately detects high level, and AUTO command sequence can be started.
2. The input value of VIO should keep identical with IO working voltage of the chip. Otherwise, it might not work or could even damage the chip.

The pins of J2 are defined as below:

Pin #	Signal Name	Description
1	Spare(NC)	No Connect
2	USB VBUS	USB VBUS
3	Spare(NC)	No Connect
4	USB GND	USB Ground
5	Spare(NC)	No Connect
6	USB Data +V	USB difference signal for high
7	Spare(NC)	No Connect
8	USB Data -V	USB difference signal for low
9	AGND	Programmer Analog Ground

10	Spare(NC)	No Connect
11	NC	No Connect
12	Spare(NC)	No Connect
13	NC	No Connect
14	Spare(NC)	No Connect
15	AGND	Programmer Analog Ground
16	NC	No Connect
17	AGND	Programmer Analog Ground
18	NC	No Connect
19	NC	No Connect
20	NC	No Connect
21	NC	No Connect
22	NC	No Connect
23	AGND	Programmer Analog Ground
24	NC	No Connect
25	GND	Programmer Electrical Ground
26	NC	No Connect
27	Plug-in card detect	Short with Pin29
28	Universal Clear	Clear signal input for card
29	Plug-in card detect	Short with Pin27
30	Plug-in card reset	Reset signal input for card

J2 pins can be connected according to corresponding relationships of these pins.

### Definition of several common interfaces

In order to facilitate use by customer, we have defined several common interfaces corresponding to the ISP port (this definition only contains communication protocol pins. For the specific wiring method, please refer to the hint on the software of the host computer).The pin correspondence table is as follows:

Pin #/Name	SPI	I2C	JTAG	BDM	UART
VIO					
DIO1					CTS
DIO2	SPIMOSI		TDO		TXD
DIO4	SPICLK	SCL	TCK		RTS
RSV2					
DIO3	SPIMISO		TDI		RXD
DIO0		SDA		BDM	
DIO7			TMS		
GND					
TPIN					
NC					
DIO5			TRST	RESET	

NC					
RSV3					
NC					
GND					
NC					
TPOUT					
NC					
ETXD					
NC					
ERXD					
GND					
ECTS					
GND					
ERTS					
+12V					
ECANL					
+12V					
ECANH					

Note: SDA and SCL are reserved for test and are not provided for the user.

### Examples for connection of target system

Before programming the target device, you must learn the function of each pin of the device, read the instructions on programming the device with ISP programmer, know the hardware programming interfaces on the target board, and confirm compliance of the target board design with ISP requirements (see the section “ISP requirements for the target system” for detail).

The programming operation steps are as follows:

1. Search the type of chip to be operated in the ISP programming software and view the operation hint.
2. Connect the signal lines (including GND) to corresponding interfaces of the ISP programmer with the target board.
3. If the above “mass production” function is to be used, TPIN and TPOUT signal lines should also be connected.
4. It is recommended that the power of the target board is supplied independently for those with high power consumption and multiple power systems. If it is confirmed that the power is supplied by the programmer ISP cable, then VCC (sometimes also need Vpp) should be connected.
5. The independent power supplier of the target board switches on the power.
6. Operation functions for running the programmer chip.

Take AT Mega48 as an example: According to the operation hint, we should make a line connection in relation to the table below:

ISP Programmer		Target system
Pin #	Signal Name	Connection
1	VIO	3.3V

2	DIO1	No connect
3	DIO2	MOSI
4	DIO4	SCK
5	RSV2	No connect
6	DIO3	MISO
7	DIO0	No connect
8	DIO7	No connect
9	GND	No connect
10	TPIN	No connect
11	NC	No connect
12	DIO5	RESET
13	NC	No connect
14	RSV3	No connect
15	NC	No connect
16	GND	GND
17	NC	No connect
18	TPOUT	No connect
19	NC	No connect
20	ETXD	No connect
21	NC	No connect
22	ERXD	No connect
23	GND	GND
24	ECTS	No connect
25	GND	GND
26	ERTS	No connect
27	+12V	+12 Power Supply
28	ECANL	No connect
29	+12V	+12 Power Supply
30	ECANH	No connect

### **SP3070 requirements for target system**

Take into account the particularity of ISP at the design stage of the target board. This point is often neglected by many people. They think that it is enough as long as the control signal lines are connected to IC's corresponding pins.

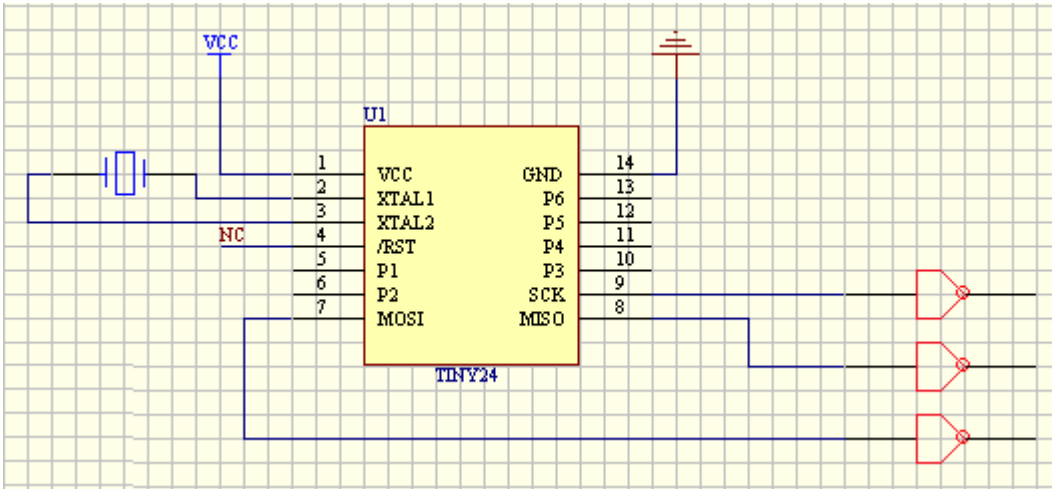
### **Attentions on pin connection**

1. All pins required during the programming should have interface terminals or measuring points in order to ensure that SP3070 can be connected to them conveniently.

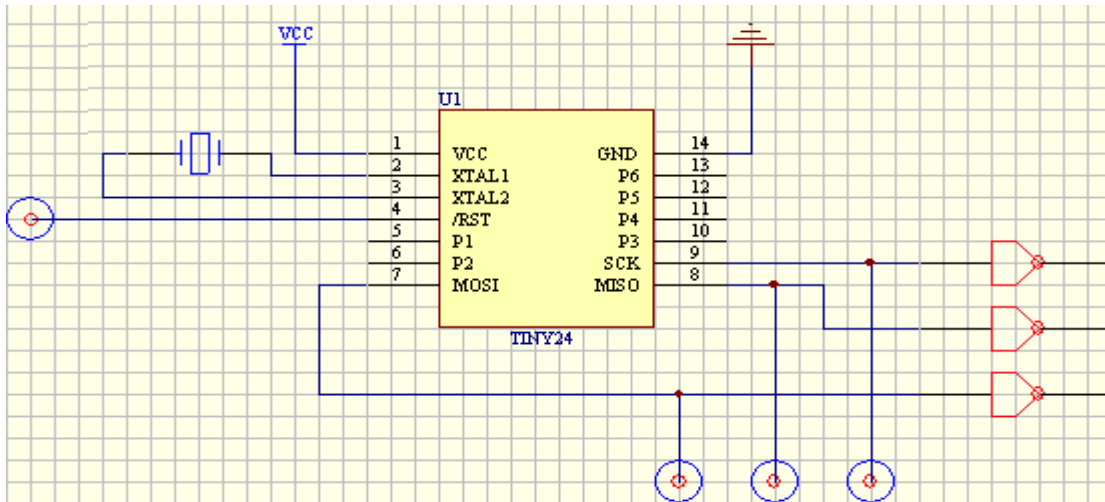


2. In order to prevent the required programming pins from incurring signal jumps during programming, the PCB circuit board where the pins are located must not be connected with other circuits.

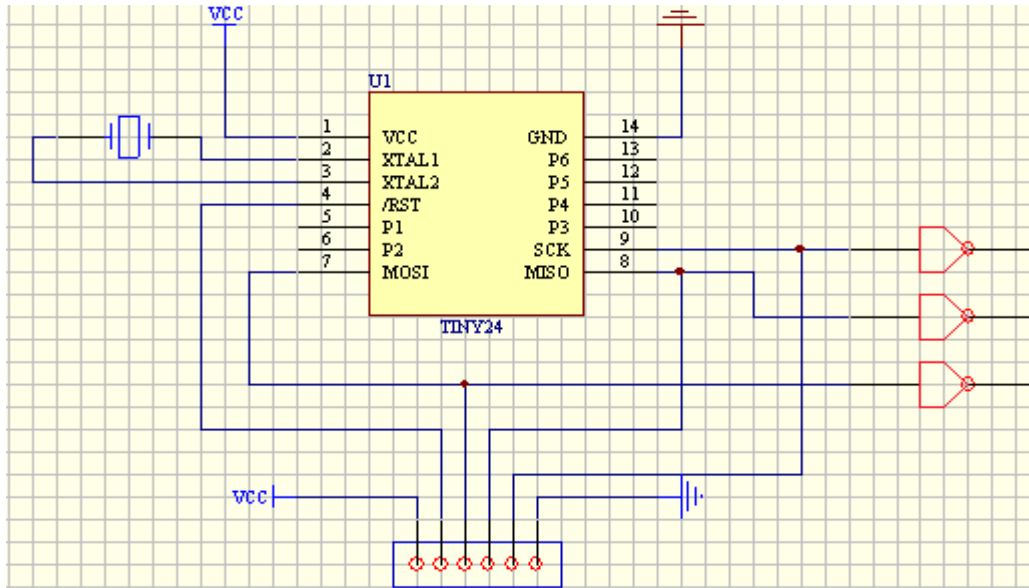
Below are some common circuit connection methods. We will describe their merits and demerits one by one.



The design in the figure above makes the programmer difficult to test the pin signals and to connect with the programming pins.



All the programming pins in the above diagram have a test point. Therefore, it is easy to measure the signal values. But since there are no connecting terminals for programming, it is still not easy to connect with the programmer.



The pins required for programming in the above diagram are all connected to the interface terminals. Hence, it is very easy to test and burn record.

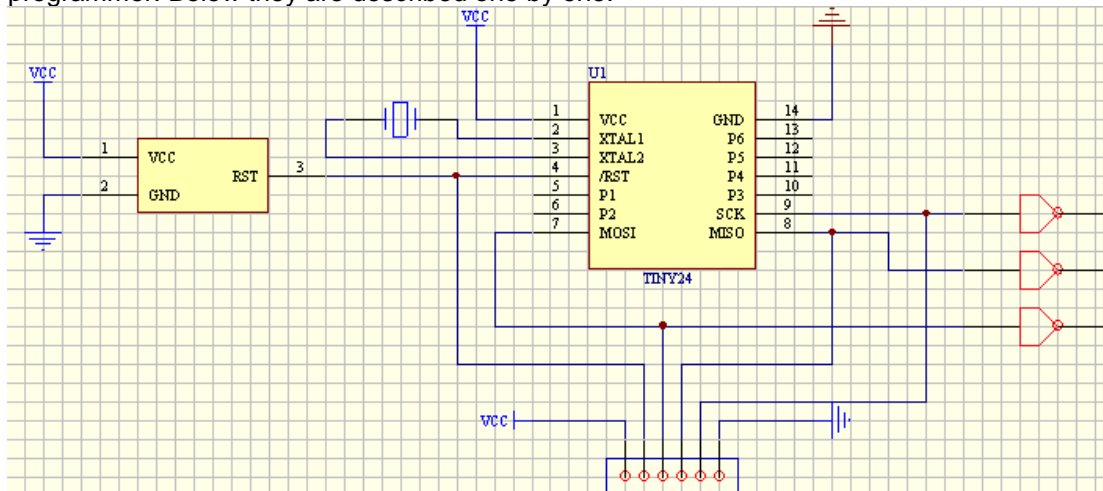
#### Precautions for power:

Though SP3070 can provide programming power (1.8-18V, 5W), it is not sufficient enough to make the target board work normally. Therefore, we recommend using the self-contained power of the target board. If the self-contained power of the target board or working voltage (VCC) and programming voltage (VPROG) is needed during the programming, all target boards must ensure their correct programming voltages. This is determined by the chips that need to be programmed and other circuits. Please refer to the programming manual of this chip for detail.

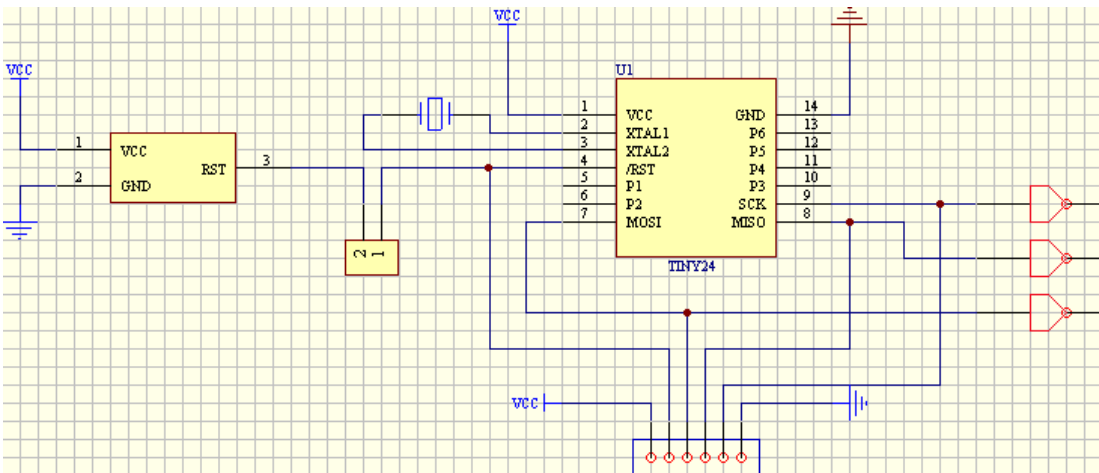
To verify the programming chip with different voltage, the PCB board must be able to provide working voltage with a different range of accuracy.

#### Precautions for the design with reset generator

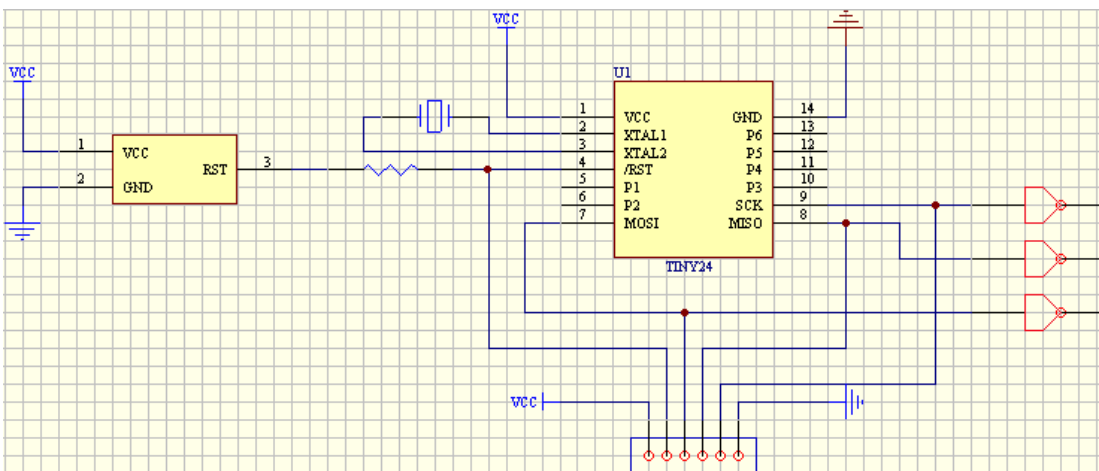
Take caution if SP3070 needs to use the RESET pin, and the RESET generator has already been provided on the target board. The RESET generator may conflict with the actuation signal of the programmer. Below they are described one by one.



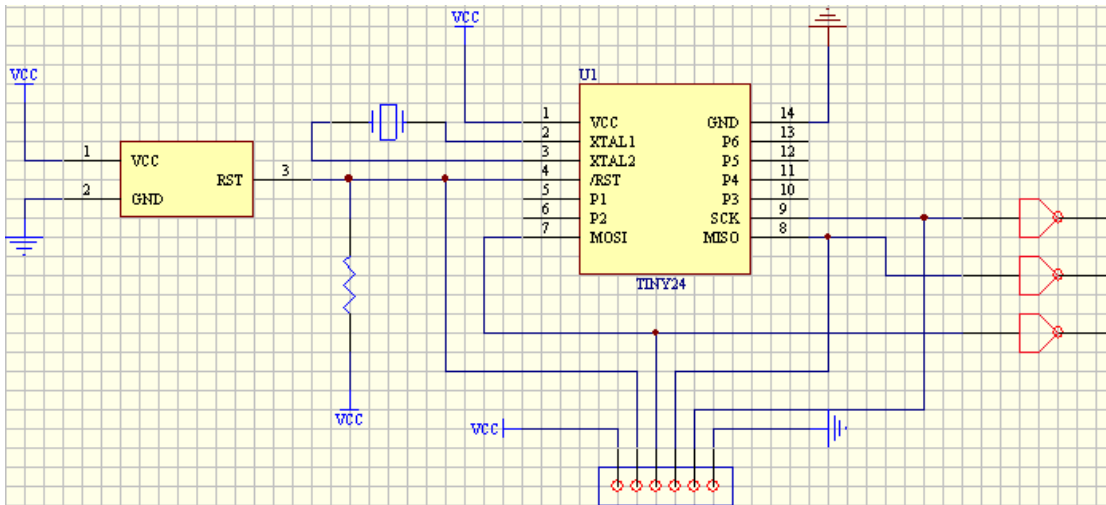
The connection of the reset generator in the diagram above conflicts with the Reset pin for programming. Be careful when the In-System programmer drives Reset pin.



The connection in the diagram above eliminates pin conflicts by adding a jumper, but such an operation is not very convenient.

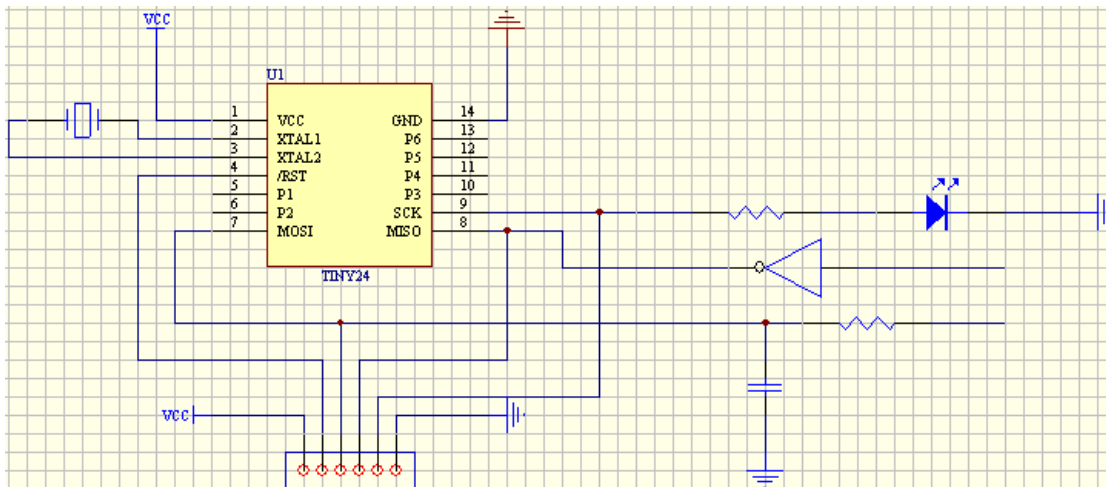


The connection in the diagram above eliminates the problem of pin conflict through adding a resistance of large value. But we still need to pay attention to the effect of pull-up/pull-down resistance on the input level signal of the RESET pin.



The connection in the diagram above is relatively reasonable. It settles the conflict problem through a pull-up resistance and allows the equipment to work well. It is important to note that after detecting an external reset signal, some Resetters will automatically have a delay in order to meet the needs of external equipment.

### Examples of common problems



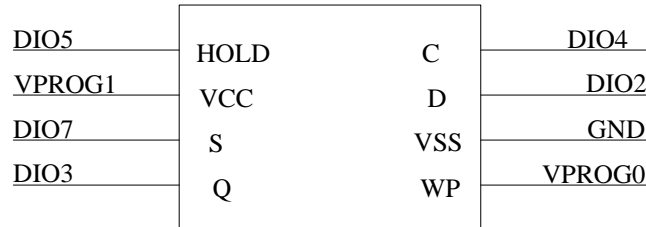
There are three problems in above diagram: ① LED lamp will extract large current from ISP programmer; ② external input signal on MISO pin conflicts with SP3070 programming signal; ③ capacitance on MOSI pin may affect the value of level signal output from this pin.



## Examples for connection of several common serial communication protocols

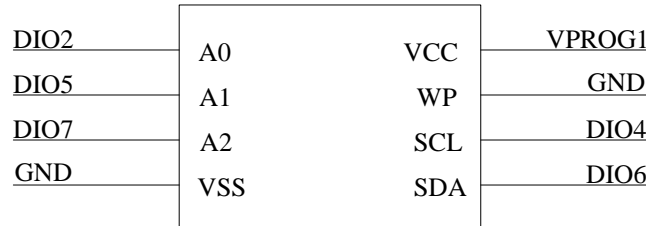
### M25P64

M25P64 is a large-capacity serial ROM programmed through SPI protocol and made by STM Company. In this example, the chip packaging mode adopted is SO16, the schematic connection diagram is shown as below:



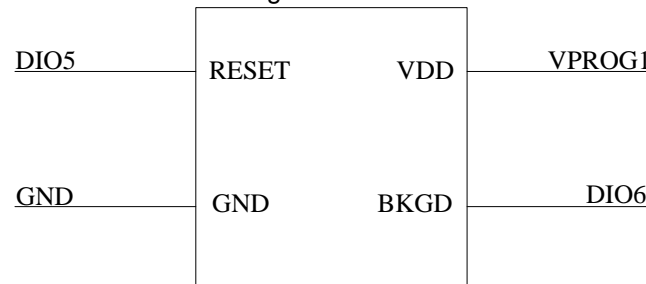
### 24C04

ST 24C04 is a universal serial ROM made by STM company, and programmed through I2C protocol. In this example, the chip packaging mode adopted is SO8. The schematic connection diagram is shown as below:



### S9S08AW16

S9S08AW16 is a HCS08 series low power consumption and high-performance 8-bit controller widely used for consumer electronics manufactured by Freescale Company. This chip is programmed through BDM serial protocol. In this example, the chip packaging mode adopted is LQFP44. The schematic connection diagram is shown as below:



## Installation of software

If you use SP3070 based on the USB port for the first time, the content of this chapter will help you to correctly install the application software for the programmer and connect programmer hardware. USB equipment is a kind of plug & play equipment. In the first time installation, Windows will invoke "add new equipment wizard" and scan all available INF files, attempting to find the appropriate driver.

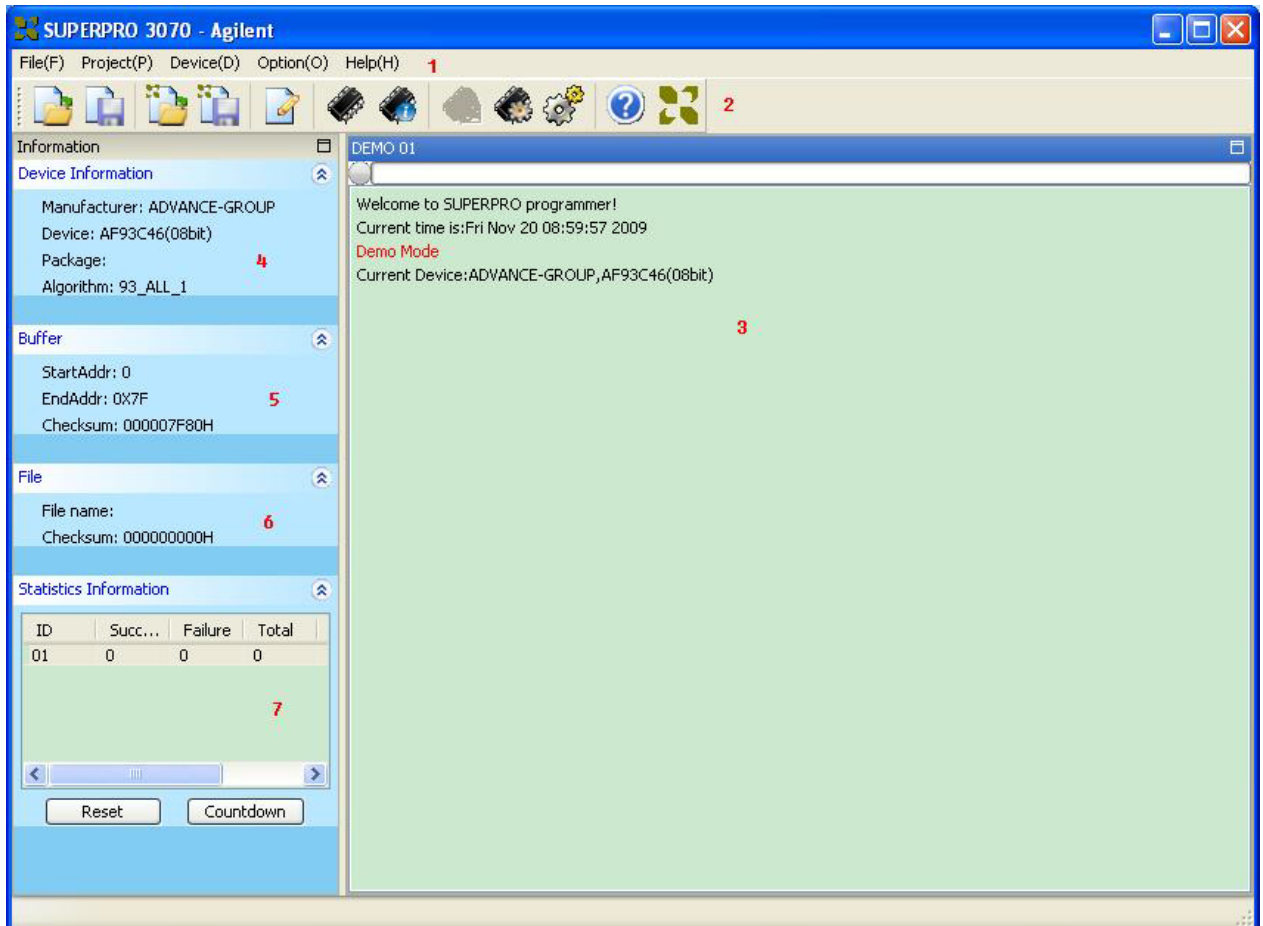
## Quick start

This chapter will help you understand the whole process of chip burn-recording operation from beginning to the end. The content includes:

- User interface of application software
- Steps of create projects

Before you are ready to operate the burn-recording device of the programmer, please make sure that you have properly installed the programmer, and that the computer has successfully communicated with the programmer.

## User interface of application software



1. Menu bar

4. Device information bar

7. Statistic window

2. Main toolbar


5. Buffer information bar

3. Log window

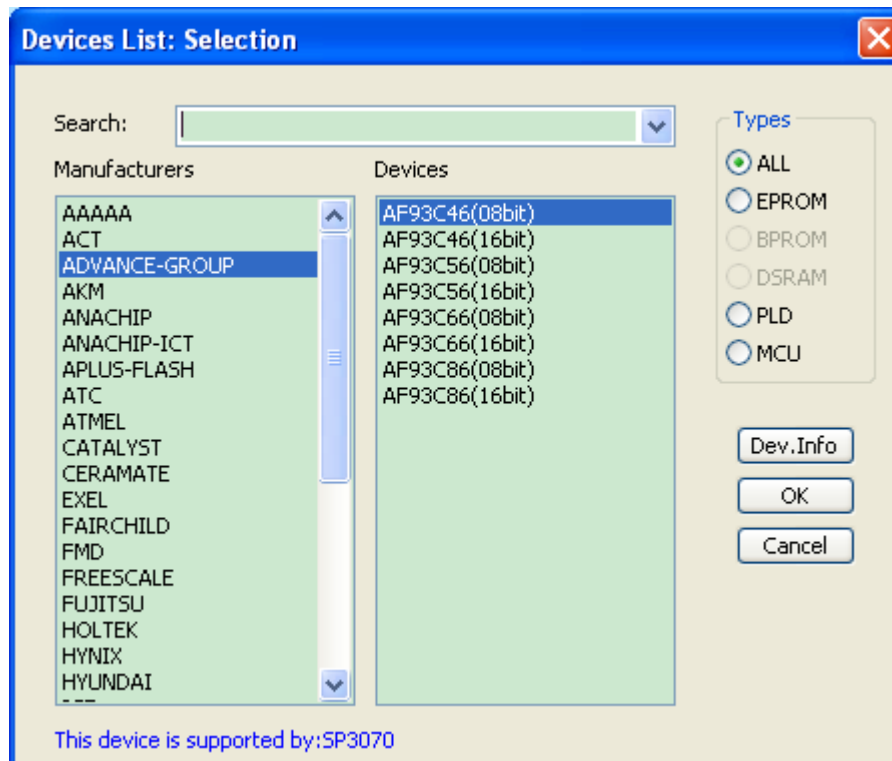
6. File information bar

## Steps for creating a project

### Select device

Click “select device” on the device menu bar or  on the tool bar. A dialogue box for selecting a device pops up.

First, select the device type. For example: E/EPROM, BPROM, DSRAM, PLD or MCU. Then select manufacturer and device name, and click the ‘OK’ button. You can also key in the device name in the Search edit box to narrow down the selections.



Note: SP3070 does not support BPROM, DSRAM.

### Load data into buffer

Programming device is a process that burns the data in the buffer into the storage unit of the chip according to the manufacturer’s requirements. There are two ways to load data into the buffer.

#### 1) Read from file

Select “Load” from “File” on the main menu, select appropriate file, format, and type in “Load File” dialogue box. After confirmation, the data file is loaded. The user can check the data in the buffer edit window.

#### 2) Read data from master slice

After selecting the device, place the master slice, and click “Read” on the operation tool bar. Then, the data in the chip will be copied into the buffer. At this time, the user can enter the buffer edit window to check if the data is correct or not. This data can be stored in a disc for future use.

Attention: Some devices do not have a read-out function, or have been encrypted so that data cannot be read out from the master slice.



## Set options

1) Operation options:

- ID Check: Checks ID of device before chip burning.
- Automatic Increment: when the user selects chip burning, he writes the serial number in the form of accumulative numbers at the designated position so that each chip has a different mark number after burning.
- Change the start and end address of the area of the device that needs to be burn-recorded.
- Verify Mode selection. Select specific voltage value of VCC for verification in order to verify correctness of chip burn-recording.
- 

2) Edit Auto. In the operation tool bar, all devices have a basic batch process operation "Auto". Its function is to execute other operation functions of devices in the sequences that are already edited. Generally, Edit Auto mode is selected for all devices.

Erase

Blank check

Program

Verify


Security or Protect

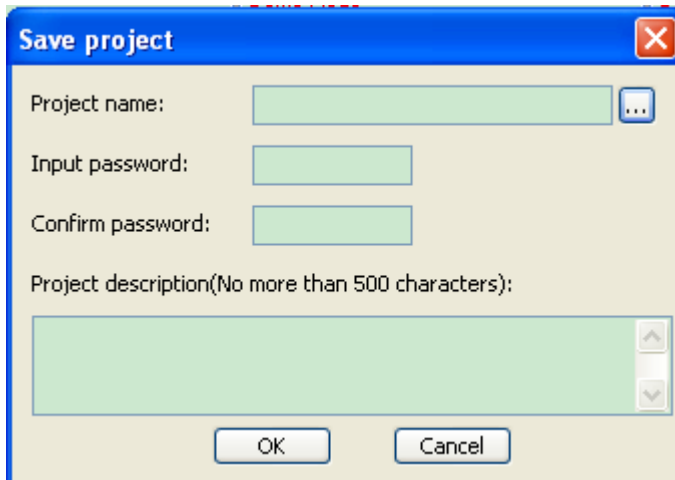
3) Dev. Config. For the chip with configuration words, the words must be set before burning the device to ensure that the burned chip can work in the user's target system. The configuration words of some devices are contained in the user's data file and are automatically filled in the buffer of configuration characters upon import of data files.

4) Dev. Info. After selecting the device, the dialogue box for device information pops up to display the important information of a relevant device. Some devices have special requirements for burning, or the algorithm for burning has special convention. The user should adjust the operation steps or buffer data after carefully reading the requirements or conventions. When the adapter is needed to support burning, the corresponding information is displayed (SP3070 shows how to connect SP3070 programmer and some instructions.)

5) Production mode. The programmer checks the socket status all the time. Once the chip is placed completely, it automatically starts AUTO command without a manual key push to facilitate the user to batch burn the chips (SP3070 production mode can only be used in offline mode).

## Save project

Select "Save Project" on the "Project" menu bar, or click  on the tool bar, then the dialogue box for project loading pops up



Select the project file for saving. Input the password and description if necessary. After clicking “OK”, the project is generated in the specified directory.

### Operate SP3070 on I3070

Please create the project file first (in this example, it's named “24C128.prj”). Write bt-basic command file, and then run this file in bt-basic console, SP3070 will program the device. The command file is as follows:

```
dim Info$[40]

!testhead is 1
!fixture lock

!SETUP
load board
vacuum well a is 2,3
faon
powered
cps

!Start PROGRAMMING
xdload "ISP01"
xdconnect "ISP01"
xdcall "ISP01","Ping", Var, Rtn$;"1" !check isp01
xdcall "ISP01","LoadPrj", Var, Rtn$;"24C128_T.prj"
xdcall "ISP01","Run", Var, Rtn$;"Program"
print "loop start"
loop
  xdcall "ISP01", "GetStatus", Var, Info$
  print " get status:",Info$,"var:",Var
  if Info$ = "set_vpp" then
    VPP = Var/10
    print "msg:setvpp"; VPP !print debug msg
    xdcall "ISP01", "SetVxx", Var, Rtn$
  end if
  if Info$ = "set_vcc" then
    VCC = Var/10
```

```

    xdcall "ISP01", "SetVxx", Var, Rtn$
    print " done:setvcc"; VCC
end if
if Info$ = "on_vpp" then
    sps 1, VPP, 2
    xdcall "ISP01", "OutVxx", Var, Rtn$
    print " done:onvcc"; VPP
end if
if Info$ = "on_vcc" then
    sps 2, VCC, 2
    xdcall "ISP01", "OutVxx", Var, Rtn$
    print " done:onvcc"; VCC
end if
if Info$ = "off_vpp" then
    sps 1, 0, 2
    dps 1
    xdcall "ISP01", "OutVxx", Var, Rtn$
    print " done:offvpp"
end if
if Info$ = "off_vcc" then
    sps 2, 0, 2 ! vcc power supply is enable
    dps 2
    xdcall "ISP01", "OutVxx", Var, Rtn$
    print " done:offvcc"
end if
if Info$ = "algo_ok" then
    print " algo ok"
    goto End_Loop
end if
if Info$ = "algo_fail" then
    print " algo fail"
    goto End_Loop
end if
if Info$ = "scale" then
    !value is percent of scale information
    print "Percent complete = "; Var; " %"
end if
end loop
End_Loop:
print "loop over"
xddisconnect "ISP01"
xdunload "ISP01"
print "all over"

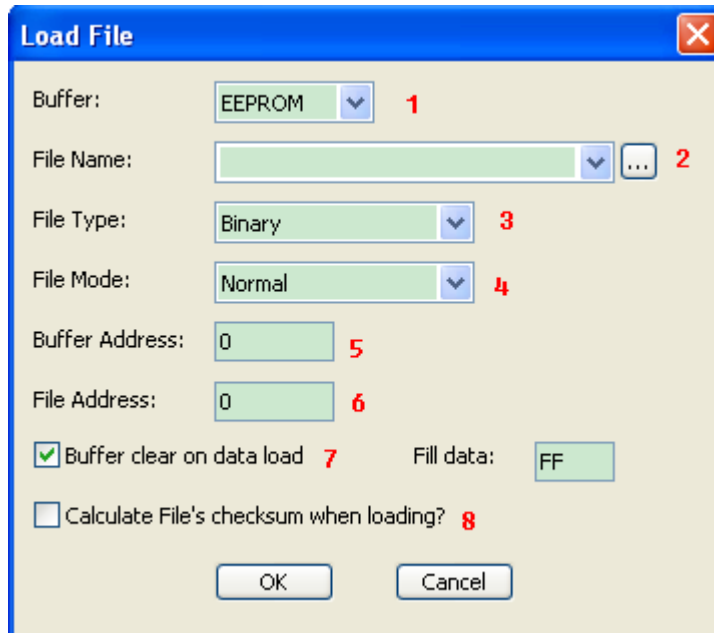
```

## Detailed description of functions

### Browse menu

#### Load file

Select "Load File" on the menu bar, or click  on the tool bar. The dialogue box for file loading pops up.



1. Data buffer: the data buffer of the device is divided into one or more areas. If there are several data buffers displayed after selecting the device, explain the meaning of the data buffers according to the names of the buffers and the data manual.
2. File name: designate the file with loaded data. The user can select the file by file dialogue box which pops up when the button "..." is clicked, or from file history record which can be browsed when the drop-down arrow is clicked.  
Note: After selecting the file, it is still needed to select the appropriate type of data file. It can then be ensured that the data is loaded correctly.
3. Type of file: include Binary, Intel Hex (extension) etc.
4. Loading mode:
  - <Normal>: the whole file is loaded.
  - <Even>: First byte of every two bytes is taken and the second byte is abandoned.
  - <Odd>: Second byte of every two bytes is taken and the first byte is abandoned.
  - <1st byte of 4>: First byte of every four bytes is taken and the rest three bytes are abandoned.
  - <2nd byte of 4>: Second byte of every four bytes is taken and the rest three bytes are abandoned.
  - <3rd byte of 4>: Third byte of every four bytes is taken and the rest three bytes are abandoned.
  - <4th byte of 4>: The fourth byte of every four bytes is taken and the rest three bytes are abandoned.

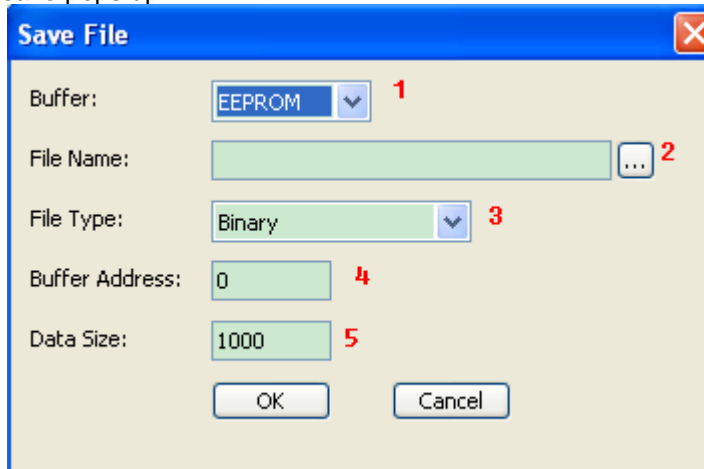
<1st 2\_byte of 4>: The first two bytes of every four bytes are taken and the rest two bytes are abandoned.

<2nd 2\_byte of 4>: The last two bytes of every four bytes are taken and the rest two bytes are abandoned.

5. Buffer address: the start address where data is loaded into the buffer.
6. Offset address of file: some types of files have non-zero start addresses (data offset address). Thus, it may be required to input the file start address when loading the file. Please input correct address data in the Edit box. An incorrect file offset address will result in filling in the buffer head by FF (or incorrect data). An incorrect big offset address will result in data flooding or system failure.
7. Check if the buffer is to be cleared when loading the file. If the user selects "Yes", then he must indicate which value is used to erase the buffer.
8. Check if the check sum is calculated when loading the file.

### Save file

Select "Save File" on the menu bar, or click  on the tool bar, then the dialogue box for file save pops up.



The image shows a "Save File" dialog box with the following fields and controls:


- Buffer:** A dropdown menu set to "EEPROM" with a red "1" next to it.
- File Name:** A text input field with a browse button ("...") and a red "2" next to it.
- File Type:** A dropdown menu set to "Binary" with a red "3" next to it.
- Buffer Address:** A text input field containing "0" with a red "4" next to it.
- Data Size:** A text input field containing "1000" with a red "5" next to it.
- Buttons:** "OK" and "Cancel" buttons at the bottom.

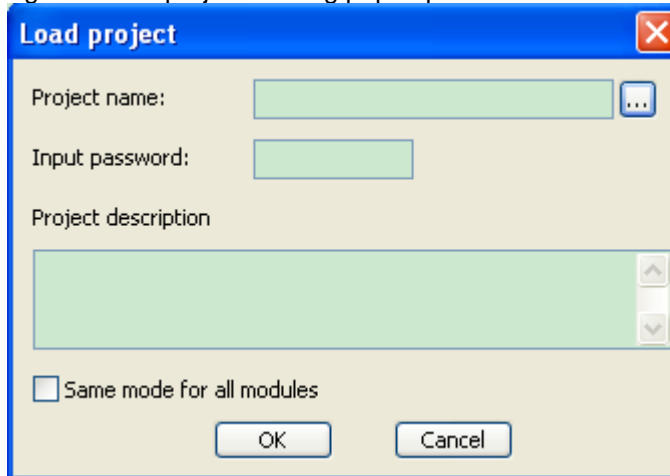
1. Data buffer: select the buffer data to be saved
2. File name: select to save the buffer data into a specific file type.
3. Type of file: include Binary, Intel Hex (extension), Motorola S record, Tektronix Hex, etc.
4. Buffer address: starts to save the data from the start address specified in the buffer.
5. Data size: specify the size of the data to be saved.

### Exit

Close the programmer software and return to the operating system.

## Load project


Select “Load Project” on the “Project” menu bar, or click  on the tool bar, then the dialogue box for project loading pops up.

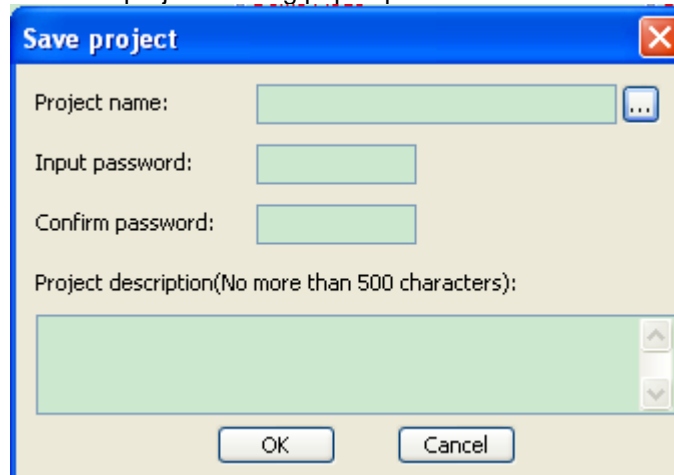


After selecting the project and inputting the correct password, the project is loaded.

If the “all modules adopt the same mode” is chosen, this project will be loaded into all the modules.

## Save project

Select “Save Project” on the “Project” menu bar, or click  on the tool bar, then the dialogue box for project loading pops up.



Project name: specify the name of project to be saved.

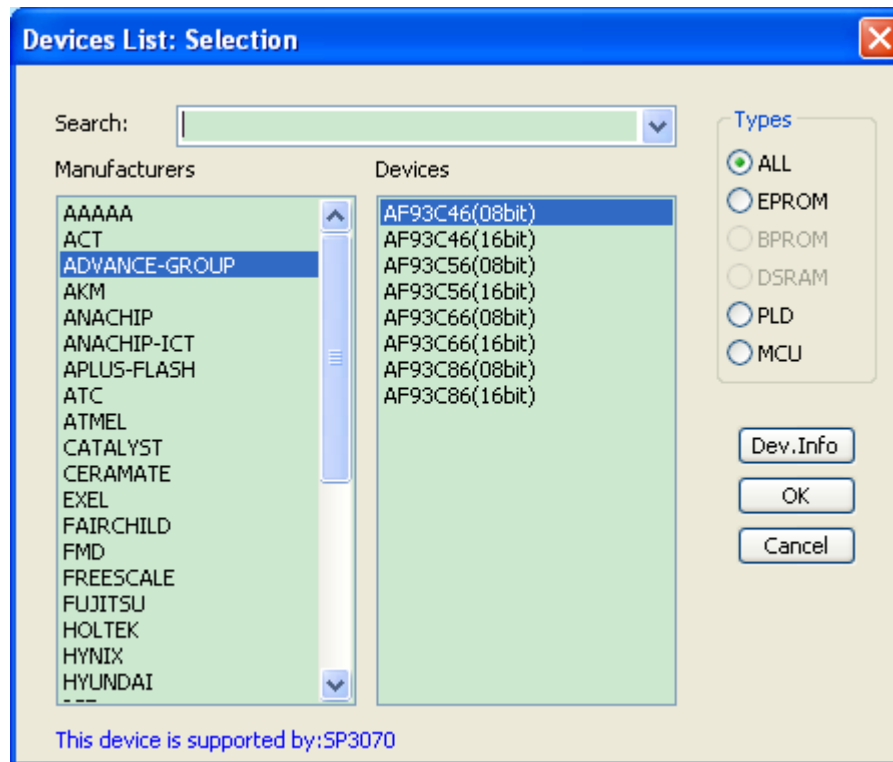
Password: for data security, the user can add a password in the project file.

File description: you can add simple description information for your project (not exceeding 1000 bytes).

Note: the suffix name of project files must be .prj.

## Select device

Select “Device Select” on the menu bar, or click  on the tool bar, then the dialogue box for device selection pops up.



The dialogue box for “Device” includes controls such as the manufacturer browse box and the device name browse box. Type ‘select button’ and OK and Cancel buttons as well as Search Edit box etc. How do you select a device?

- Select device type among the type select buttons. There are five major types: E/EPROM (EPROM, EEPROM, FLASH etc.), PLD, B/PROM, DRAM/SRAM, MCU. If you are not sure, you can select “ALL”.
- Assign a manufacturer name in the manufacturer browse box.
- Assign device name in Device Name browse box and just press the “OK” button.

The name printed on the chip may include parameters such as speed, temperature, package etc. When you select a name that is not identical to the name in the dialogue box of “Device”, you should neglect these parameters or ask for help.

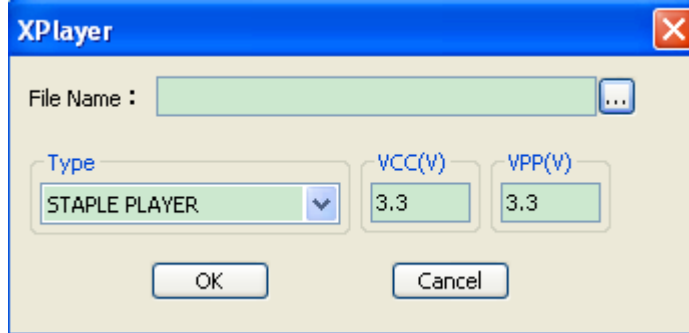
Use of Search edit box: it may be very inconvenient to directly select the device due to a large number of devices and difference in names of chip. In this case, we can use the Search edit box to help the selection. As long as you input several key characters, the number of devices can be greatly narrowed. This facilitates fast search of the device. For example: when you want to find the 89 series single-chip microcomputer, input characters “89” in the Search edit box. All the devices containing characters “89” will be listed. The software is sensitive to the sequence of characters “89” when making a search, but it will neglect presence of other characters in the box. For example: the character strings “819” and “DA8S9” will be listed since they meet the requirement, while characters “98” will be neglected since it does not.

**Note: Some chips cannot be used only after authorization is given, after selecting this chip, the log window will remind “this chip needs authorization!”**

## XPlayer

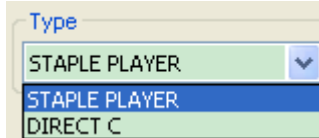
XPlayer is a special programming tool mainly for CPLD series chips produced by ACTEL, ALTERA etc. As long as the customer provides corresponding STP or JAM file (such files can be generated using such design tools as ISE, Quartus II etc.), the corresponding device can be programmed.

Select "XPlayer" on the "Device" menu bar, then the dialogue box pops up.



File name: select the needed STP or JAM file.

Type: 1. STAPLE PLAYER; 2. DIRECT C




The universality of STAPLE PLAYER is comparatively better than DIRECT C, while its speed is lower than the latter.

VCC: working voltage of chip

VPP: programming voltage provided by the programmer

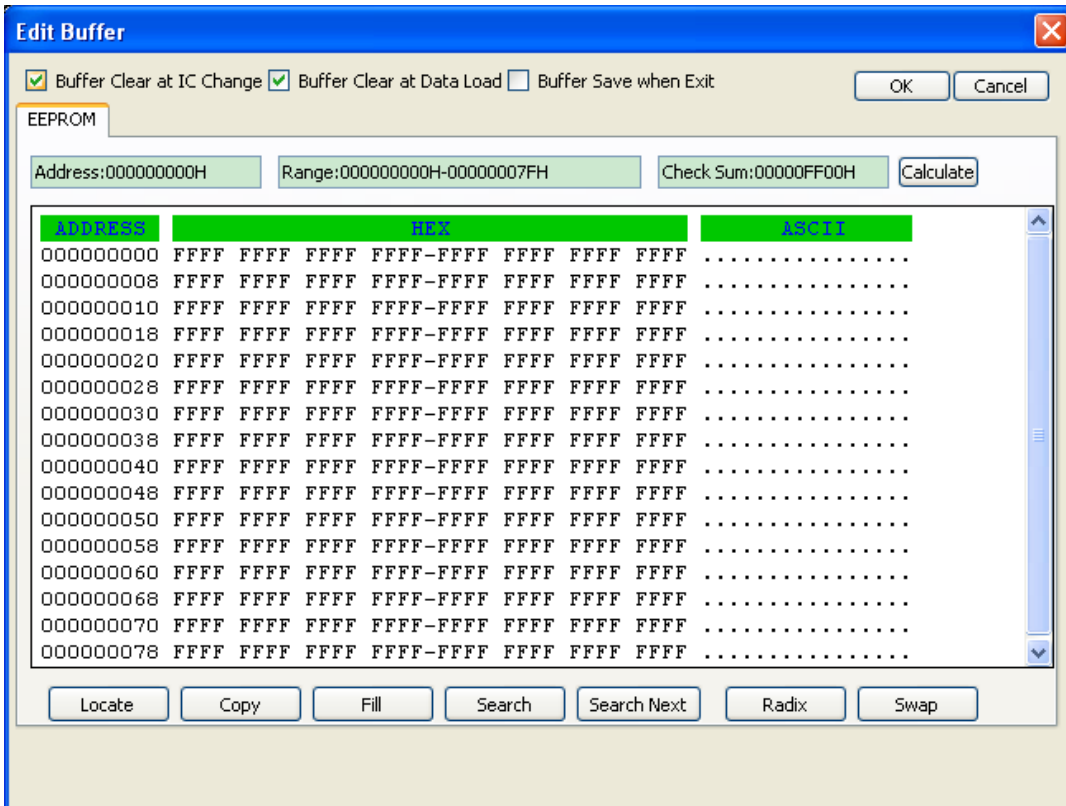
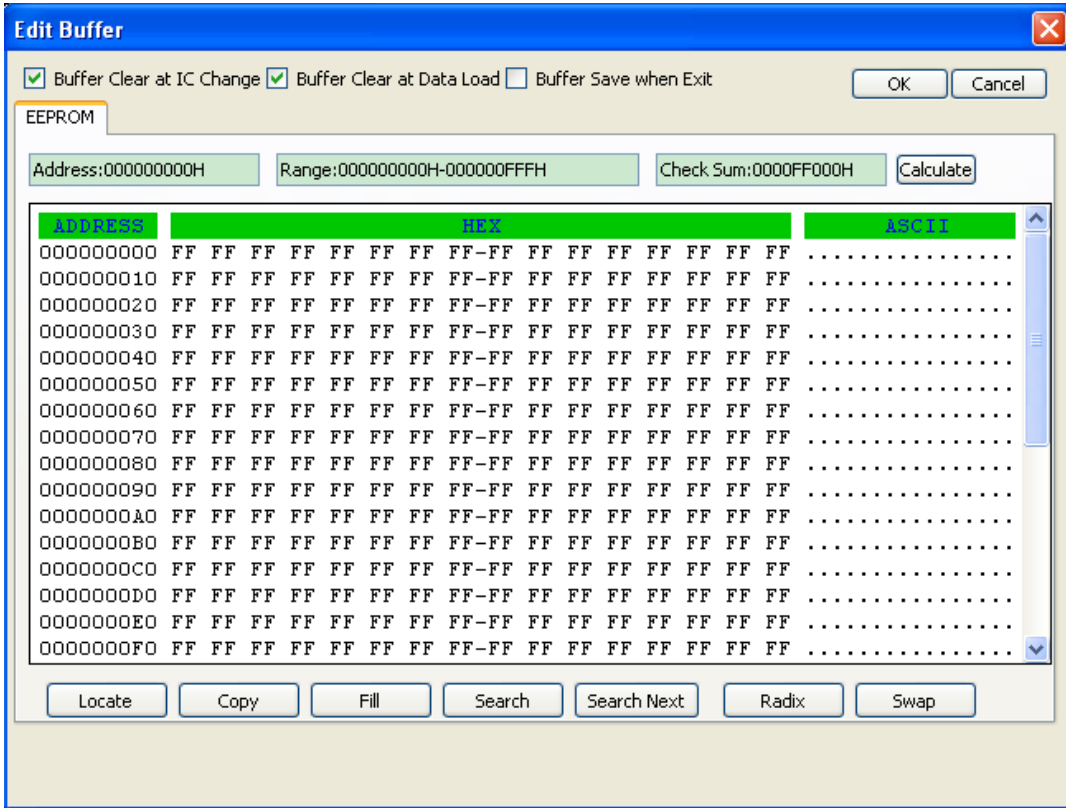
## Buffer

Select "Buffer" on the menu bar of "Device", or click  on the tool bar, then the dialogue box for buffer edit pops up.

HEX/ASCII data buffer

The data width of each unit is 8 bits or 16 bits according to the chip.





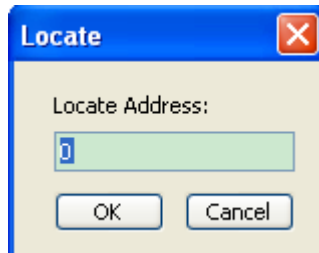
**Range:** displays the address and end address of this buffer (with character width as the unit).

**Address:** displays the position of current cursor Address:000000000H

Note: this address is the address with the word width of chip as the unit.

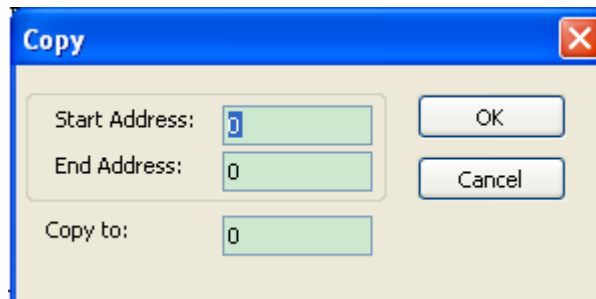
**Checksum:** see “calculation of check sum’ for details.

**Locate:** move the cursor to the designated position of the buffer (with character width as the unit)



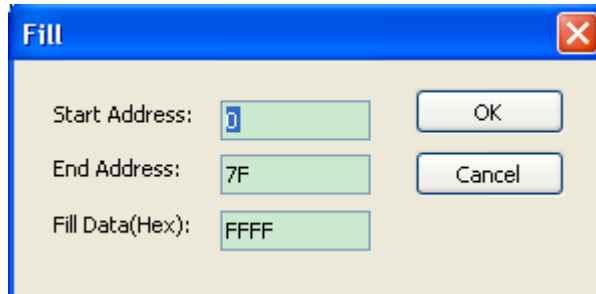
Note: the address to be filled is hex numerical value. Below is the same.

**Copy:**



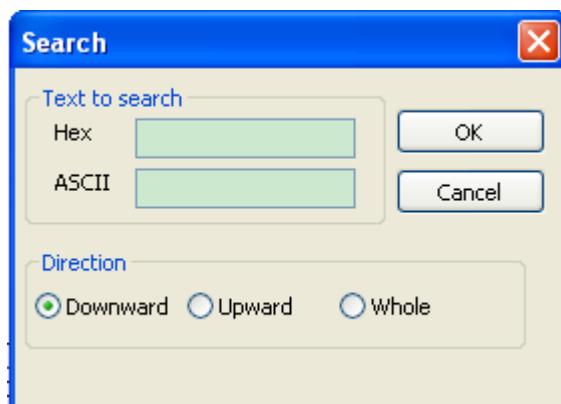
Copy the designated area of buffer to another position (with word width as the unit)

**Fill:**



Fill the designated area of buffer with the filling value (with word width as the unit)

**Search:**

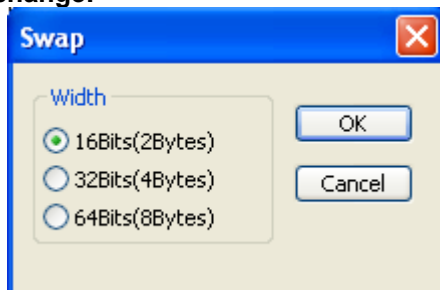


Search the data in the buffer through hex system or ASCII. The user can select the search way (search forward from current position, search backward from current position and search backward from beginning of the buffer).

**Search next:** search next address that meets the condition according to the search way the user selects.

**Switching:** switch over the display format of address in the buffer (hex system or decimal system).

**Exchange:**



Assume the data at address 0-10(hex system) of data buffer is:

12 34 56 78 90 AA BB CC – DD EE FF 11 22 33 44 55

The character width that can be selected is:

16 bits(2 bytes), the data after exchange is as follows:

34 12 78 56 AA 90 CC BB – EE DD 11 FF 33 22 55 44

32 bits(4 bytes), the data after exchange is as follows:

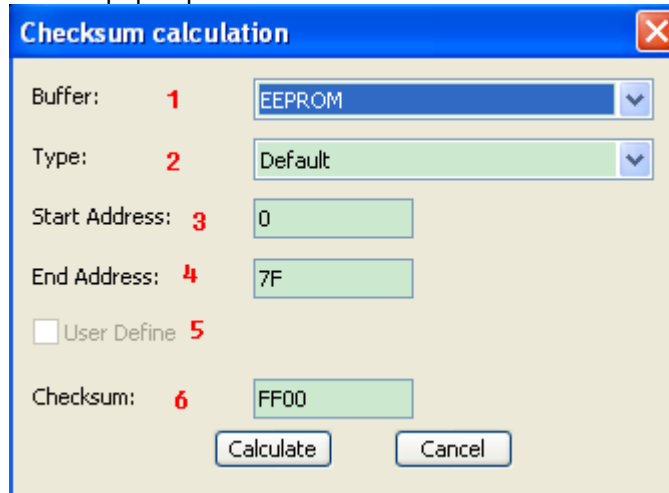
78 56 34 12 BB CC 90 AA – 11 FF EE DD 55 44 33 22

64 bits(8 bytes), the data after exchange is as follows:

CC CC AA 90 78 56 34 12 – 55 44 33 22 11 FF EE DD

### Checksum:


Select “Checksum” in the “device” menu bar. The dialog box of checksum verification and calculation pops up.



When the type of data in data buffer is Data (HEX/ASCII data), the default verification and calculation method is to add up all the data from beginning to end with byte as the unit. To meet the user's special requirements, he can totalize the data in part of the addresses in a specified way.


1. Specify the buffer where the checksum is to be calculated.
2. Way for checksum calculation (sum of bytes by default)
3. The start address of checksum calculation (in bytes)
4. The end address of checksum calculation (in bytes)
5. Special method for checksum calculation, once choosing this, above selection will be disabled.
6. Checksum calculation result

### Device information

Select “device information” on the “Device” menu bar or  on the toolbar. The current device information will be displayed.


If there is no special information, the device information only displays the manufacturer name, device name, package, adapter and algorithm. This information can also be found in the information column on the left. Any additional content will be displayed according to different chip.

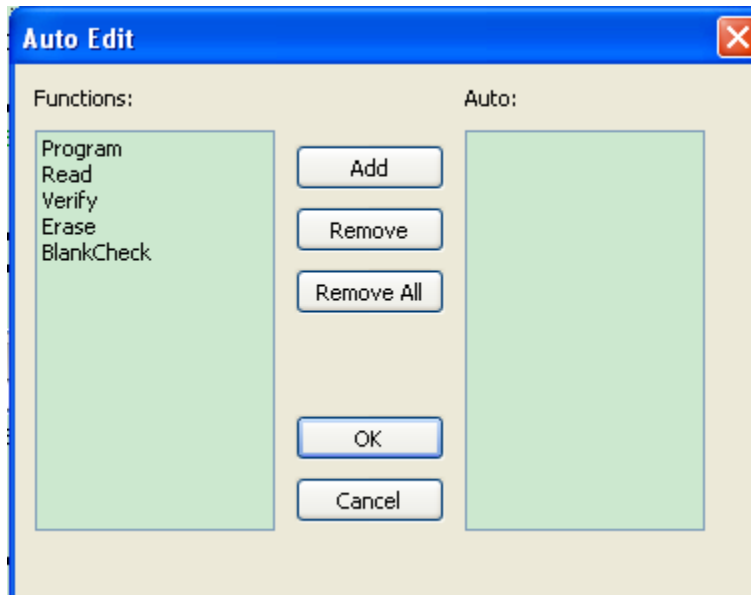
### Device Configuration Word

Select “device configuration word” on the “Device” menu bar or  on the toolbar. The current device information will be displayed.

For the device with a configuration word, the configuration word must be set before burning the device in order to ensure the burned chip can work in the user's target system. The configuration words of some devices are contained in the user's data file and are automatically filled in the buffer of configuration characters upon import of data file. Some devices need manual setting.

## Edit automatic processing command

Select "Edit Auto" in the "Device" menu or  on the main toolbar, the dialogue box pops up:




On the toolbar of device operation, all devices have a basic "Auto" operation. This can perform other operation functions of devices in the window of operation function items in the edited sequence, like processing batch. Open the "Edit Auto" dialogue box. All the operation function items of this device are displayed in the list on the left side of the dialogue box, operation function items, and execution sequence which execute the "Auto" in the "Auto" list box on the right side.

*Add:* Add operation function items from function list to auto list (it can also be added by double clicking this item).

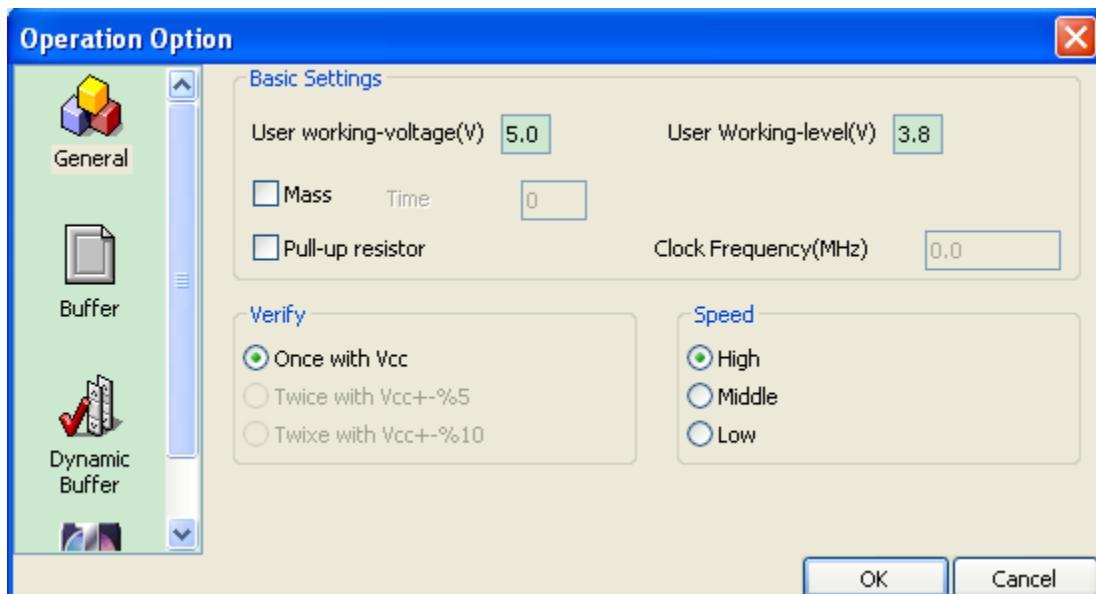
*Delete:* Remove selected function item from auto list (it also can be deleted by double-clicking the item).

*Clear All:* Delete all function items in the auto list.

## Operation options

Select "Operation option" in the "Device" menu or  on the main toolbar, the following dialogue box pops up:

## General options



### *General Options include:*

User's working voltage and working level.

User sets working voltage and level according to the actual condition of target board.

### *Production mode*

If production mode is selected, the user should specify the delay time (required stabilization time after placing the chip).

The production mode is only valid under offline state.

Pull-up resistance

Clock frequency

*Verify mode* (only applicable to the case that the power of target board is supplied by the programmer)

After burning (programming) a chip, it is necessary to verify the burned data. According to the manufacturer's programming data, voltage applied to VCC pin has the following changes when making verification:

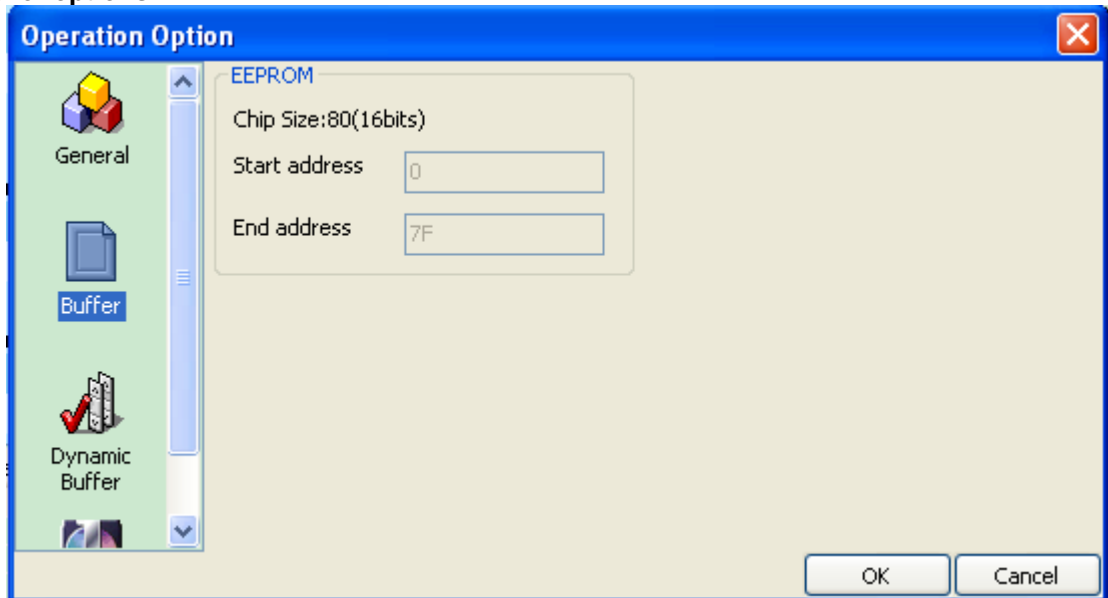
First, verify the data with VCC( $\pm 5\%$ ) or ( $\pm 10\%$ ), for example, for VCC=5.00V, then we can select VCC=5.00V to verify the data once, or use VCC=4.75V and VCC=5.25V to verify the data twice ( $\pm 5\%$ ) or use VCC=4.50V and VCC=5.50V to verify the data twice ( $\pm 10\%$ ).

Second, verify the data using the minimum verification voltage (MinVcc) and the maximum verification voltage (MaxVcc). The two voltage values referred above are supplied by the manufacturer.

### *Speed*

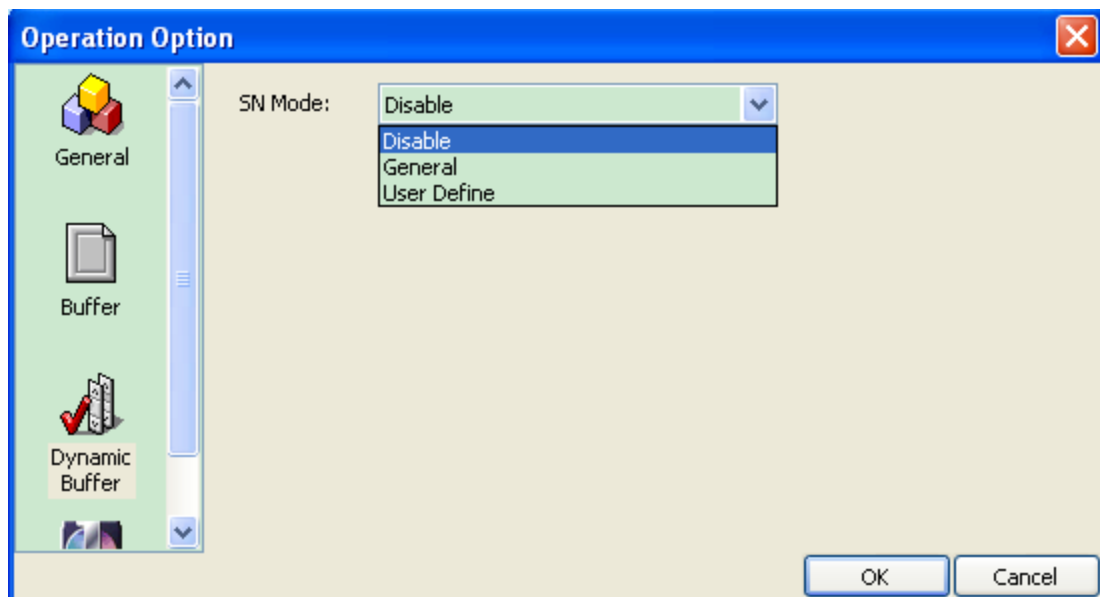
The user can decide the programming speed himself and can choose one of three speeds, i.e. high, medium and low (varies with chip).

## Buffer options

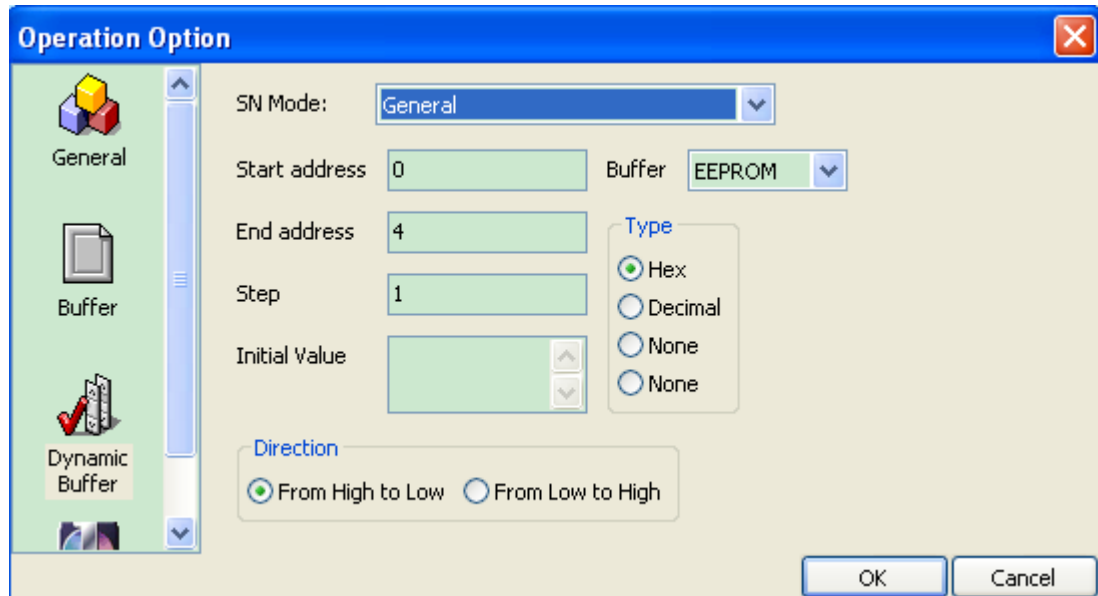


The user can select to burn a part of the chip, which is applicable to most part of E/EPROM (FLASH) devices.

## Serial number setting (dynamic buffer)



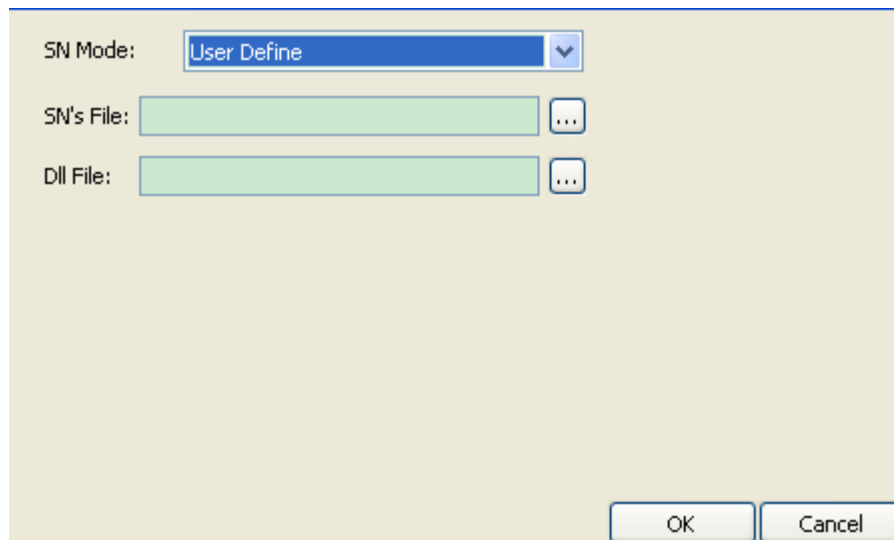
Some applications require writing different content to local areas of the chip such as, the serial number of the product, MAC address etc. These are collectively called 'serial number'. This software has provided two solutions. One is the standard increment method and the other is the user-define method. The software for the former generates the serial number by automatic increment on the previous serial number. The format is fixed and it can meet general requirements. The serial number of the latter is generated by the user-define software, therefore, it has a very big freedom.



The user needs to fill the start and end address, step length and initial value. The user can select which buffer is to be incremented and how to increase (format, step and direction).

Note 1: What is reflected initially is the current value. If the user wants to change the serial number to be burned during the burning process, he should change the initial value.

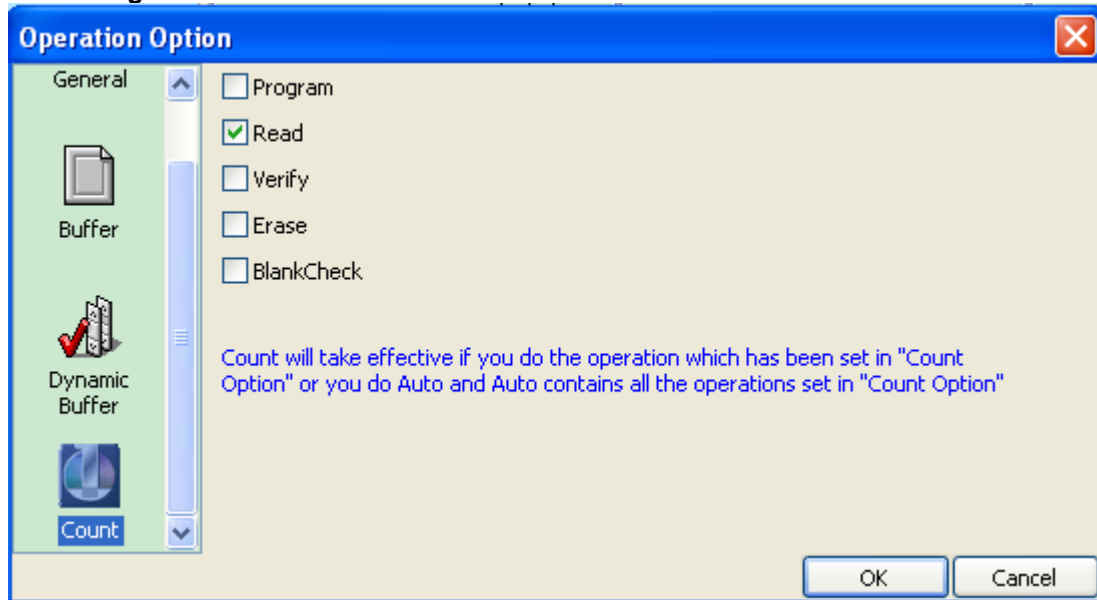
Note 2: The format of initial value must be correct. When the type is non-ASCII, the format of initial value is "XX XX XX". When the type is ASCII, the format of initial value is "XXXXX". If the format is not correct, it will be considered as 0.



User-define mode: The user needs to provide the serial number file and dll called by the software. The serial number file needs to be generated correctly according to a certain format



## Statistic setting



When burning the chip, the software will do statistics of programming according to the user's setting.

Statistical method:

Count will take effect if you do the operation which has been set in "Count Option". Or you can do Auto and Auto, which contains all the operations set in "Count Option"

## Tool bar

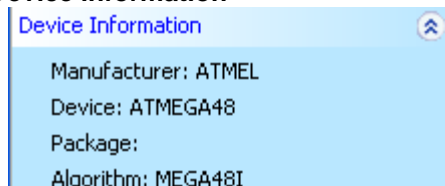
### Main toolbar




- |                       |                              |                 |
|-----------------------|------------------------------|-----------------|
| 1. Open file          | 2. Save file                 | 3. Load project |
| 4. Save project       | 5. Edit buffer               | 6. Select chip  |
| 7. Device information | 8. Device Configuration Word | 9. Edit Auto    |
| 10. Operation Option  | 11. Help                     | 12. About XELEK |

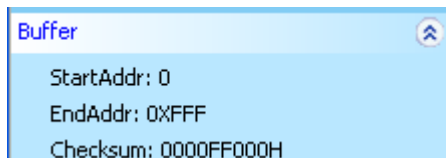
## Information bar

### Device information



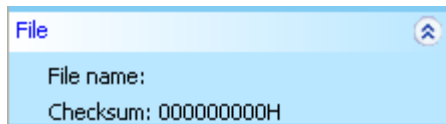
The device information bar shows the basic information of the device. If the device has special information, you need click "Dev.Info" option on "devices" menu bar or  option on tool bar.

### Buffer information



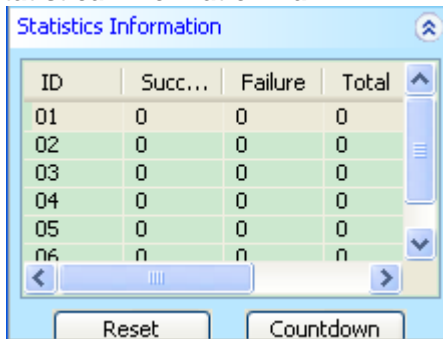
The buffer information bar shows start address, end address and checksum of main chip buffer.

### File information



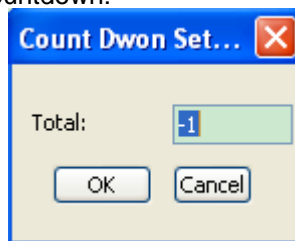
It shows the current loaded file name and file checksum (whether to calculate checksum or not is decided by user himself).

### Statistical Information Bar



Reset: Clear success and failure etc. to zero, and cancel countdown setting (countdown number is -1)

Countdown:



When the countdown reaches -1, it indicates that the countdown setting has been cancelled.

Note: The statistic mode needs to be set in "count" found in "operation option".