Инструкция по эксплуатации программатора USB MiniPro TL866II Plus

1. Programmer Overview

1.1 Performance Introduction

Carefully built low-cost professional programmer, high-density chip production process, a unified user interface, easy to use, functional integrity, application software, reliable operation, code small, faster execution. supported operating system: WINXP WIN2003 WIN2008 VISTA WIN7 WIN8 WIN10 32-bit or 64-bit operating system

• The programming speed is greatly improved. It is suitable for small batch production and supports 4 programmers for one computer.

• A wide range of chip support, the initial version supports the chip 15000+, supports a variety of 24 25 26 27 28 29 37 39 49 50 series of serial and parallel memory chips, microcontroller 51 series, ATMEL AVR ATMEGA AT90 series, MICROCHIP PIC10 PIC12 PIC16 PIC18 series microcontroller, GAL programmable logic device programming, SRAM test, 54/74 and CMOS4000 integrated circuit integrated functional test. Support NAND Flash chip, the function is complete and comprehensive, the capacity supports to 8Gbits.

25 series Nor Flash capacity support to 512Mbits.

- Pin contact check function, poor contact pin can pinpoint
- Ultra-low power consumption: the entire programmer uses ultra-low-power chips, using USB interface, the first truly all chip programming without external power integrated programmer. Portable performance super: the world's smallest volume, size 10CM * 6CM * 2.5CM, only the size of the driver's license.
- Programmable VCC from 1.8V to 6.5V, VPP from 1.8V to 18V chip
- Chip serial number secondary development functions: Programmer built a variety of commonly used automatic numbering function. And can use a custom algorithm to achieve any sequence number through DLL dynamic library. This feature allows you to produce a unique identification information for each chip in mass production. The logo can also be encrypted with a custom algorithm to more effectively protect your product's rights.
- Universal serial programming interface ICSP (to support both high-voltage parallel programming and low-voltage serial programming). Note: ICSP now supports 24-Series 25 Series 93 Series Serial Integrated Circuits, ATMEL89S51,52, AVR ATMEGAxxx Full Series, and MICROCHIP PIC10Fxxx 12Fxxx 16Fxxx 18Fxxx Full Series, and the new series of SYNCMOS SM59Dxx SM59Rxx full range chips for in-circuit serial programming.
- Powerful 54 / 74F / LS / HC CMOS4000 Series Integrated Functionality Tests: The First Development Programmer to Test Common Logic ICs True, Test Errors Locating Logic Gate Circuits, Test Arbitrary Possible input combinations.

1.2 Programmer chip support range

- Variety of 26 27 28 29 37 39 49 50 series of parallel ROM, EPROM, EEPROM Fast, full support. Support TSOP32 TSOP40 TSOP48 chips to 64MBits, will join the TSOP56 pin support.
 (TL866II programming voltage VPP only supports up to 18V, for early 27C series ROM, the chip with 21V 25V VPP voltage no longer supports writing, but can be read. If it is used, the VPP with the same type of programming voltage can be used as 13V or 12V chip replacement, for example: 27C32 21V or13V Vpp chip ,Pay attention to the VPP voltage)
- 24 25 35 45 85 93 95 Series Serial EEPROM Serial Series Chip Support Full, Supports Chip Capacity 25 Series 8 or 16PIN to 512MBits (64M Bytes)
- NAND FLASH chip supports up to 8Gbits
- MCU series, hundreds of models of 51 series of microcomputer
- AVR microcontroller package scratch ATMEGA series ATtiny series AT90XXX series of microcontrollers, all AVR ATMEGAxxx chip also supports the ICSP interface for SPI serial download programming. Support AVR one-to-one soft-key with RC correction byte.
- MICROCHIP PIC10 PIC12 PIC16 PIC18 Series Microcontroller: PIC series of more than 300 models, more than 900 package chip support, currently supports one of the most complete PIC chip integrated programmer. For various types of PIC monolithic, in strict accordance with the manufacturer's manual requirements, the optimal programming, programming speed is excellent. The vast majority of chips support both ISCP online programming and programming through programming sockets.
- GAL programmable logic device programming

SRAM test

Support for common 24 61 62 DS12 series SRAM test, respectively, to data line testing, address line testing, memory unit incremental testing and unit testing.

• Super performance 54 / 74F / LS / HC CMOS4000 series integrated circuit integrated function test:

2

The first development-type programmer to really test common logic ICs to test for errors that can be located to logic gates, and to test any possible combinations of inputs in an integrated circuit when testing.

2.Software installation

2.1.1 Download and install the application

Temporarily shut down antivirus software when installation.

When installing a 64-bit operating system, run the EXE installation package as an administrator.

Download the Xgpro_setup.exe file from http://www.xgecu.com/en and double-click the file to go to the installer interface. As follows: (The installation file contains the contents of this manual.) Click [Install] to start the application installation. The following screen will pop up:

Note: When WIN7 64-bit operating system is installed, right-click the XgproVxxx_setup.exe file and select Run as administrator. Enter the directory you want to install here. Click Install to start the installation of the application.

\bigotimes	欢迎使用鑫工XGecu TL866II多功能通用编程器!	
\square	• 单击[install] 按钮开始安装。[Cancel] 取消安装	
	 如果你以前安装过本编程器软件,现在是升级程序请选择和 以前相同的安装目录。本安装将进复盖安装。如果选择不同的 目录将会保留原版本的程序。 	
	 应用程序安装完毕后,将进行设备驱动程序的安装。 	
		-
	目标文件夹 (0)	_
	D:\Xgpro √ 浏览(W)	
	安装进度	

Click Install until done.

2.1.2 Install USB driver

In the installation directory (D: $\ xgpro$) to find UsbDiverInstall.exe file, you can doubleclick to run.



Pop-up USB driver installation dialog box. as follows: Device Driver Installation Wizard



Next step:

Device Driver Installation \	Nizard	
	Completing th Installation W	ie Device Driver Vizard
	The drivers were succes	ssfully installed on this computer.
	You can now connect y	our device to this computer. If your device
	Driver Name	Status
	V Haikou Xingong Ele	ectron Ready to use
	< 上-	-步(8) 完成 取消

USB driver installation is complete.

2.1.3 Equipment installation

After completing the above installation, then insert the programmer into the USB port, the operating system will find the hardware, as follows:



Select [Automatically install software], click Next, windows will be automatically installed, until the successful completion of the installation.

(Note: In this process, the driver will change the registry. If the firewall software is installed, please release the software, windows will also prompt the driver without a digital visa, to confirm the installation).

If the automatic installation does not attack, you can also choose to install USB driver files in the installation directory (D: $\ xgpro \ drv$). Installation is complete, open the



Plus run programmer software.

You can run Xgpro.exe also in the installation directory (D: \ xgpro).

2.1.4 Firmware Refresh

The first time you use or upgrade the firmware may need to be updated, in the software's main menu, a refresh operation, as shown below:

Select IC IC IC IC System Self-check ACE24AC16 IC Sit Reflash firmware Set Interface Set Interface IC Set Interface	100	P		2	हिंग 🛄
Select IC II: System Self-check ACE24AC16 Set Interface	100	人			
Set Interface					
G 40P adapter C ICSP port □ ICSP_VCC Enable 0 8 Bits C 1 S 8 S 8 Bits C 1 S 8 S 8 S 8 S 8 S 8 S 8 S 8 S 8 S 8 S 8 S 8 S	6 Bi	ts [buffe	r sel	lect —
Address 0 1 2 3 4 5 6 7 8 9 A B	C	D	E	F	ASCII

3.Quick start

3.1 Hardware preparation and select the correct chip type

After you make sure that the universal programmer application and the USB driver are properly installed, connect the USB communication cable. At this time, the red power indicator of the programmer indicates that the connection is normal (lighting state), the yellow indicator is off waiting for programming (if the yellow indicator is blinking, the USB driver is installed incorrectly, or the USB programmer device driver is not installed) Can enter the next step.



- 1. 1 Programmer Connected indicates that the programmer is connected
- 2、 Click the chip select this button, pop up the chip selection dialog box,
- 3. In the search chip, enter the chip type to read and write (in this case W29C020C)
- 4. Click on the correct model Note: @ PLCC32 @ TSOP32 chip package no @ label refers to the DIP package
- 5、 Choose [Select]

3.2 Read the chip and save the file

3 Xgpro v7.03 File(F) Select IC(S) Project(P) Device(D) Tools(V) Help(H) Language(L)		- - X
	P 😼 🐨 🛄	
Select IC W29C02QC Set Interface G 40P adapter C ICS port IC Information (No Project opened) ChipType: EEPROM IC Size: 0x40000 Bytes (262144 Bytes) Set Interface G 40P adapter C ICS port ICSP_VCC Enable G 8 Bits C 16 Bits	buffer select	Code Memo
Address 0 Address 0 6000-0000 F F 6000-0020 E F 6000-0020 B F 60001 B F 60001 B <td>, 3</td> <td>USB Device Info: None device found. W29C020C Memory Size : 0x00040000 TL886HI-Plus V04.2.83 1 Programmer Connected.</td>	, 3	USB Device Info: None device found. W29C020C Memory Size : 0x00040000 TL886HI-Plus V04.2.83 1 Programmer Connected.
Ready Hardware Interface Ver: T	L866II-Plus	0000 0000

Complete the steps in section 3.1 after selecting the chip to finish

- 1. Click the [R] button in the toolbar above to pop up the read dialog box.
- 2. In the read dialog box, click the [Read] button.
- 3、 After reading, click [Cancel] button.
- 4. Click Save File (you can also select <Save File> in the File menu). After entering the file name as shown below, save the file.

8

■ 閉片	* <u>2</u> %	修改日期	<u>246</u> 开U	大小
2 文档	D.	(PRAMOZ	~	200
□ 洪善下#	111.BIN	2018/2/16 14:22	BIN 文件	135,168
	222.BIN	2018/2/18 12:42	BIN 文件	9
一)首乐	BBT.BIN	2017/12/22 17:31	BIN 文件	8,192
	Test.BIN	2017/12/22 17:31	BIN 文件	8,192
→ 家庭组		美型: BIN 文件 大小: 8.00 MB 修改日期: 2017/12/22 17:31		
	÷ • [
文件名(N):	Myfile1			
保存举型(刀:	BIN Files(*,BIN)			

3.3 Burn the contents of the file to the chip

Complete the steps in section 3.1 after selecting the chip to finish

Xgpro v7.03		X
le(F) Select IC(S) Project(P) Device(D) Tools(V) Help(H) Language(L)		
🛎 🖬 🛃 🖪 📅 🕫 🕸 🗛 📇 📿 🔋	🕑 😼 😇 🛄	
-Select IC W29C020C W29C020C IC Information(No Project opened) ChipType: EEPROM ChkSum: Ox01FE 8484 IC Size: Ox40000 Bytes (262144 Bytes)		Ti Information
Set Interface	_ buffer select	
(* 40F adater Address 0 66868-8866 Ft ○ - → ≪ OS8 (D:) → Xgpro → UserData -	・ ・ チャ 提素 UserData の	
0000-0010 F4 0000-0020 Ef 组织 ▼ 新建文件夹 0000-0030 Ef 组织 ▼	III • 🔟 🕢	W29C020C Memory Size : 0x00040000 TL866II-Plus V04.2.83
80006-0040 BE 文档 名称 ^ 80808-8056 90 計測電下載	修改日期	1 Programmer Connected.
0000-0060 9F	2018/2/27 10:04 BIN 文件	107
0006-0090 37 0008-0090 CE 必家庭组		e O
66666-6686 51 ≡ 66666-66C6 AS ∰ 计算机 66666-66D6 FF 및 计算机		CLOX
8888-88E 8 4 Soft (C:) 8888-88F 8 Soft (C:)	- <u>_</u>	
Options ↓ Pin Detect WIN8PRO (E:)		3
▼ Erase before		
▼ Verify after 文件名(N): test.BIN	✓ All_Files(*.*)	10-
₩ Skip 0xFF	TTT(O) BIOM	
Blank Check		
ady Hardware Interface Ver:	TL866II-Plus	0000 0000

- 1. Click on the figure shown in 1, open the data file, open the file dialog box
- 2. Choose to open the data file (Test.BIN)
- 3. Click [Open], the following pop-up file options dialog box pops up, according to the default settings, click [OK] button.

From File Start Addr(Hex): 000
egion TO Buffer Strat Addr(HEX): 00
efault Clear Buffer when loading the file
ode Memory Clear buffer with default
efault ode Memory

4. Click [OK], then the contents of the file has been transferred to the computer memory buffer.

le(F) Select IC(S) Project(P) De 🛎 🖬 🛱 🛃 🖪 🗊	vice(D) Tools(V) Help(H) Language(L) () 25 田 四 () ?	
Select IC #29C020C	IC Information(No Project opened) ChipType: EEFEOM ChiSum: 0x01FE 8484 IC Size: 0x40000 Bytes (262144 Bytes)	Information
Set Interface Address 0 1 2 3 0000-0000 F3 41 59 BA 0000-0000 F3 41 59 BA 0000-0010 F3 68 55 54 0000-0020 EF 75 44 66 0000-0020 EF 75 44 66 0000-0020 EF 75 44 66 0000-0020 BF 75 44 66 0000-0050 9D 3E 69 71 0000-0050 9D 3E 69 71 0000-0050 9F 76 9C A6 0000-0050 37 44 E6 52 0000-0060 37 44 E6 52 0000-0080 51 BD 20 B8 0000-0080 FF B8 03 37 0000-0080 FF B8 03 5	Chip Program Range W29C020C V CODE Memory Start Adr: 00000000 End Adr: 0003FFFF Programming successful ! 6 D: 0x DA 45OK! Programme CodeTime: 10202ms Verify CodeTime: 203ms Software Data Protection EnableOK! Programm Cancel	Clear 1. 10040000 83 meeted.
 ✓ Verify after	o SH_NUM LL C Sect D -> 0003FFFF	
ady	Hardware Interface Ver: TL866II-Plus	0000 0000

- 5. Click [P] button on the tools bar, chip programming dialog box is pop-up.
- 6. Click [Program], start to burn the chip, there is no error message when burning, that is the burn is completed.
- 7. If you need to burn more chips, put a new chip, click on the "programm" to start a new programming.

4. Programmer features detailed

4.1 Main menu summary

Menu command su	mmary		
File menu	Open, save, cache operation function, exit		
Chip selection	Press the menu list to select the programmed chip, 25 auto recognition		
menu			
Project menu	Open Project, Save Project, Save Project as, Close Project, Project		
	Properties, Project Password Change		
Device menu	Read chip content, chip ID, Verify chip, programming chip, erase chip,		
	Blank check , automatic Serials number, testing , Multi-Programming, NAND		
	bad block check		
Tools menu	Calculator, programmer self-check, Refresh firmware		
Help menu	Help , about, Upgrade online		
Language menu	Choose a language		
Edit buffer	In the cache to find the corresponding hexadecimal value or ASCII string,		
	data, copy, copy, paste, fill, part of the save as TXT file		

4.1.1 File menu

File menu items are as follows:

🐱 Xgp	ro v7.03		
File(F)	Select IC(S)	Project(P)	Device(D)
0	pen(O)		Ctrl+O
Sa	ave(S)		Ctrl+S
C	opy		Ctrl+C
Pa	aste		Ctrl+V
B	ock Save As(S)	txt
D	efine Block		Ctrl+B
Fi	ll Block(F)		
C	ear <mark>current</mark> bu	uffer(C)	
C	ear <mark>a</mark> ll buffer(A)	
Fi	nd		Ctrl+F
Fi	nd Next		F3
G	oto(G)		Ctrl+G
E	cit(X)		

open a file:

Load hexadecimal files or binary files to the specified buffer, GAL chip for loading JED

format files.

NAND FLASH operation, open the file, the file is transferred to the BBT buffer.

Save to file

Save the current cache to HEX file in hexadecimal format or BIN file in binary format, save as JED format file when GAL chip. NAND FLASH operation, save the file, is to save the BBT buffer to file.

Note: For MICROCHIP microcontrollers, HEX files will load into the MiroChip IDE development environment to generate HEX files into the full buffer, save all buffered HEX files and be fully compatible with the MICROCHIP IDE development environment.

For the Bin file to save and open, the default is CodeMemory content + Data Memory content (if it exists)

For MICROCHIP microcontrollers, Bin also stores configuration information (sorted by Microchip address)

Copy, paste, save as block, block definition, block fill, clear the current cache, clear all the cache

These menu functions can be called using the shortcut keys or the right-click menu. See file editing features ---> Edit buffer

12

4.1.2 Chip selection menu

Sele	ect IC(S)	Project(P)	Device(D)
Search and Select IC(S)			
25 Flash Detect			

Click <Search and Select IC>

S Xgpro v7.03		X
File(F) Select IC(S) Project(P) Device(D) Tools(V) Help(H) Language(L		
🛎 🖬 👫 🛃 🖪 🐨 🎔 🤔 强 📇 <u>R</u> 🔋	🕑 🛃 🔨 🛄	
Select IC IC Information (No Project opened) ACE24AC16 IC Information (No Project opened) Set Interface ChipType: EEPROM C ADP adapter Select Device Address Select Device Address Search Device Address Search Device Address Search Device Address Search Device Manufactory W29C020 Type C ALL 0606-0620 C MMFLASH/NVRAM 0606-0620 C ROMFLASH/NVRAM 0606-0620 C RAM/DRAM 0606-0600 C SRAM/DRAM 0606-0600 Seclect a Manufactory 0606-0600 C SRAM/DRAM 0606-0600 C Sect Winbond IC Total: 15118 Winbond IC Total: 15118	800 Enter IC Name and Sea	rch Clear B Device Info: None device fund. 824AC16 Henory Size: 0x00000800 360II-Plus V04.2.83 1 Programmer Connected. ip
1 DIANK Check 0x 0000000 -> 000007FF		
Ready Hardware Interface V	Ver: TL866II-Plus	0000 0000

25 Flash Detect:

automatic recognition of 25-series Nor Flash with 8-pin and 16-pin. Note: Because many chip IDs are the same and all results have more than one, you need to choose an accurate model according to the result of recognition.

function description Only to 8/16 Pins 25 series large capacit Flash Memory. after detecting completely, Pus Select > button to automatic selection	ty sh <
Pin Counts • 8 Pins C 16 Pins	
Model: Manufactory	
Place as right figure, Click <detect> to star</detect>	
Detect Select Canc	

First select the pin number of the chip and click <Detect> to start recognizing

4.1.3 Project menu

Project files are created to better maintain the programming of the chip. It is a file that holds all current buffer data and working environment settings, including information about

the current device, settings for all operating options, and auto-related settings. Its easy to use daily volume production. Project can be set password protection, to a certain extent, can protect the safety of the product .

Project menu items are as follows:

Project(P)	Device(D)	Tools(V)
Open	Project(O)	
Save	Project(B)	
Save	Project As(S)	l.
Close	Project(C)	
Projec	ct attrib(A)	
Modi	fy Password	(K)

Save the project

After setting the required parameters of the chip in the application program, select [Project] - [Save Project] to create a project. And in the window of the information area display project information. Click [Save Project], enter the project name and click OK to pop up the following project dialog box:

asic Information -				
Project Name:	D:\Xgpro	UserData	aaa.mpj	
Device Type:	W	29C020C	\mathbb{Q}	25
Project Desc:	This is Test Project	5		
CRC 32bits:	0x 1911CB13			
Create Time:	2018-02-27 10:45	Modi:	2018-02	-27 10:45
Password length<8:	***			
Confirm password:	*	_		
Prot.	ected	Sa	ave	Cancel

1, the password is blank, the current project does not set password protection.

2, if the project is set to protect mode, the password must be entered. Set to Engineering After the protection mode, the relevant data of the project will not be changed. And some functions of the programmer are disabled.

Open the project

Open an existing project, select [Project] - [Open Project], select the project name in the open window and click OK. If the project is set password, the following dialog box will pop up. (If you do not set the password directly open the project)

Project is Enc	rypted!		
Input	PASS:		
	OK	Cancel	

Note: For the TL866 programmer's older version of the project file, you can also open the call here.

Save the project as:

Save the current project as another project name. Operation Reference 【Save Project】.

Close the current project

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

Project properties

Select [Project] - [Project Properties], pop-up property settings dialog box. Project Properties dialog box can be more change the project description and project protection mode. Enter the project password when saving.

D:\Xgpro\UserData\aaa.mpj					
W	729C020C	e			
This is Test Project		20			
0x 1911CB13					
2018-02-27 10:45	Modi:	2018-02-27 10:45			
	W This is Test Project 0x 1911CB13 2018-02-27 10:45	W29C020C This is Test Project 0x 1911CB13 2018-02-27 10:45 Modi:			

Change the project password

set a new project password

prompt:

1, the exact project description can distinguish a large number of projects. Project mode general use in mass production.

2, the project file has been encrypted to protect the project password to prevent the project file copy. To a certain extent, the data can be protected.

3, 32-bit CRC checksum is saved when the project file is saved or opened, to prevent other uncertain factors such as storage file corruption.

4.1.4 Device menu

This menu function is to chip programming operation, with the content see 4.2 programmer operation function, Menu items as shown below:

De	vice(D) Tools(V) Help(H)	Langu
	Read(R)	
	Read ID(I)	
	Verify (V)	
	Program(P)	
	Erase (E)	
	Blank Check(B)	
	Serials Number	
	Testing	
	Multi Prog.	
	NAND Bad Block Check	

4.1.5 System Tools menu

Tools menu items are as follows:



Calculator: This is a calculator that comes with Windows, which makes it easy to calculate various formats when switching to scientific mode.

16

📓 计算器	*								
编辑(E)	查看(V) 帮	助(日)	18						
	标准型(I)		52				0.	
C +*			进制 〇 二进制	④ 角周		弧度	○ 梯周	₹	
Inv) F6			Backsp	ace	CE		С
Sta	二进制(图) F8	MC	7	8	9	1	Mod	And
Ave	 角度(E) ご(度(P)) 	F2 F3	MR	4	5	6	*	Or	Xor
Stam	梯度(G)	F4	MS	1	2	з	=	Lsh	Not
2	数字分组	(I)	M+	0	+/-	*	+	=	Int
Dat	tan x	2 1/x	pi	A	в	С	D	E	F

Programmer self-test: The programmer hardware self-test operation. Click to pop up the following dialog:

Programmer Self-Check	×	
Self-Check Test Result		
No.29#OK!	~	
No.30#OK!		
No.31#OK!		
No.32#OK!		
No.33#OK!		
No.34#OK!		
No.35#OK!		
No.36#OK!		
No.37#OK!		
No.38#OK!		
No.39#OK!		
No.40#OK!		
Test VCC short protection		
VCCshort protecion OK	100	
Test VPP short protection		
VPPshort protecion OK		
	~	
	5	
If-Check Result		
Test Cancel		

Note: Before starting the programmer's self-test, please remove the IC and ICSP cable from the 40PIN socket. During self-test, the programmer will test VPP, VCC and GND for all pins and perform internal short-circuit overcurrent test Apply VPP VCC and GND to one pin). Inserting the IC may damage the IC inserted in it. After the test is completed, the test of each power supply is displayed in the dialog box.

Firmware FLASH Refresh:

Programmer hardware firmware update process.

When upgrading a new version of the application, such as the need to upgrade the FLASH program to update, the application will prompt for a firmware update when entering, if you do not need to refresh, then the refresh function is disabled. Click <Firmware FLASH Refresh> to pop up the following dialog box:

Attention: Pl computer reli supply, and o hardware upg power failure	ease carry on this upgrade operation on the ably, and guaranteed that computer's power closes all other application software. In the grade process, If the computer to halt or the system restarts that possibly damages the programmer.
APP Version:	V7.03
Hardware Type:	TL866II-Plus
Coursest Einstein VED.	V04.1.00
Current Fileware VER.	

Click the button to refresh, refresh the total time of about 15 seconds. Click <Cancel> to

exit the refresh.

Note when refreshing: Do not use an external USB hub refresh, direct access to the computer USB port, It may not be reset when using external HUB. Turn off all other programs, and ensure that the computer power supply is reliable!

4.1.6 Edit buffer

Programmer software can modify the contents of the buffer, commonly used functions are:

Buffer block definition Buffer block fill copy and paste Save the block as Clear the buffer Find, find next Goto : Locate the address

Buffer block definition

Block definition in two ways: The first method is: in the buffer zone, press the left mouse button, move the mouse, select a block of data. Figure is as follows:

Address	0	1	2	3	4	5	6	7	8	9	Å	B	C	D	E	F	ASCII
0000-0000	FF	EE	CC.	EE	CC	CC	CC	CC	EE	FF							
0000-00D0	FF	FF	FF	FF	FF	FF	FL	(Copy					Ctrl	+C	FF	
0000-00E0	FF	FF	FF	FF	FF	FF	FL	. F	aste					Ctrl	+V	FF	
0000-00F0	FF	FF	FF	FF	FF	FF	FL	Y.		-						FF	
0000-0100	FF	FF	FF	FF	FF	FF	FI		slock	Save	As(S)				txt	FF	
0000-0110	FF	FF	FF	FF	FF	FF	FL		Define	Bloc	k			Ctrl	+B	FF	
0000-0120	FF	FF	FF	FF	FF	FF	FL									FF	
0000-0130	FF	FF	FF	FF	FF	FF	FU	F	ill Blo	ock(F)						FF	
0000-0140	FF	FF	FF	FF	FF	FF	FI	0	lear	curre	nt bu	ffer(C	1			FF	
0000-0150	FF	FF	FF	FF	FF	FF	FL						·			FF	
0000-0160	FF	FF	FF	FF	FF	FF	FL	C	lear	all bu	ffer(A	9				FF	
0000-0170	FF	FF	FF	FF	FF	FF	FL		tind.					Chul		FF	
0000-0180	FF	FF	FF	FF	FF	FF	FL	1	ina					Ctri	+r	FF	
0000-0190	FF	FF	FF	FF	FF	FF	FL	F	ind N	lext					F3	FF	
0000-01A0	FF	FF	FF	FF	FF	FF	FL	(Goto(G)				Ctrl	+G	FF	
0000-0180	FF	FF	FF	FF	FF	FF	EL-			-,						FF	

The second method is as follows: Select [Block Definition] in [Main Menu] or [Right Key Menu] or press the shortcut key "CTRL + B" to pop up the following dialog box, enter the start address and end address, press OK to finish Block definition.

Start Addr	0100	
End Addr	0200	

You can perform subsequent block operations on the defined data blocks: block copy, block fill, block save as TXT file

Haikou Xingong Electronics Co., Ltd. http://www.xgecu.com

FILL Block

Padding buffer refers to the program code buffer in a continuous address of the data are all replaced by a fixed value.

Pop-up mouse [Mouse Right Button menu] select [FILL Block], pop-up the following dialog box

Code Memo	ure f		
· Code Menic	ny		
Stard_Addr: 0x	0000000		
End_Addr: 0x	0003FFFF		
Value(HEX):	FF		
Format			
• BYTE C V	VORD C Tri_byte C DWO	ORD C Random	

- A. Region: Buffer selection, the default value for the current window display buffer.
- B. To fill the start address and end address of the block, the default value is: If a block is defined, define the start to the end of the data block. If no block is defined, then the full buffer size for the current buffer.
- C. Fill value, the longest allow 4 bytes. If the fill-in method is random, the value has nothing to do with it.
- D. Fill-in mode, the default mode: If the chip according to 8-bit programming, the default single-byte mode, if the chip is programmed in 16-bit mode is double-byte mode, you are free to change the way you want to fill in .

сору

After the definition of the data block, you can pop-up mouse right-click menu, select [copy], then the contents of the block has been copied to the WINDOWS clipboard. You can also press CTRL + C keys to complete the copy.

Paste

Function: Fills in the contents of the clipboard to the buffer.

Action: Move the cursor to the start address of the buffer you want to fill in. Press CTRL + V. Paste complete.

19

Effect of copying data from area A to area B

数据地址	0	1	2	3	4	5	6	7	8	9	A	3	C	D	I	7	ASCII	
000000:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF		
888818:	90	16	64	Eő	8E	88	74	70	FB	28	48	Dó	89	20	5D	8F	t[.+H].	
888828:	D7	D8	F5	30	F9	A1	40	53	51	93	CE	89	BF	D9	87	78	0 HSQz	
000030:	DA	83	DB	CB	3F	86	77	45	28	84	88	83	43	CE	28	0B	?.WE*C.*.	
000040:	BA	28	23	51	95	CC	D1	89	49	31	61	ØC	88	80	42	AC.	.(#QI1aBL	
868858:	58	18	-09	98	78	12	77	59	85	FF	FF	FF	FF	FF	FF	FF	[p.wV	
000060:	FF	FF	FF	VE	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF		
888878:	FF	FF	FF	60	YFF	FF	FF	FF	FF	FF	FF	(FF	FF	FF	FF		
000080:	FF	FF	FF	10	FF	FF	FF	FF	FF	FF	FF	10	FF	FF	FF	FF		
888898:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF,	AFF.	FF	FF	FF	FF	**************	
0000A0:	FF	FF	FF	FF	FF	ØC	16	64	Eő	8E	26	74	70	FB	2B	48	t .+H	
000080:	D6	89	20	50	8F	07	D8	F5	30	F.9	A1	4D	53	51	93	CE]ØMSQ	
0000000:	09	BF	D9	87	7A	DA	83	DB	CB.	3F	86	77	45	2A	84	88	Z?.WE*	
0000D0:	83	43	CE	2A	ØB	8A	28	23	51	95	CC	D1	89	49	31	61	.C.*(#QI1a	
0000E0:	90	88	ØC	42	40	58	18	D9	98	78	12	77	59	85	FF	FF	BL[p.wY	
0008F8:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF		*

Note: In order to speed up the pasting speed, the data format of the clipboard is a binary format specially used by this programmer. Therefore, other WINDOW applications can not directly use this program to copy data to the clipboard. If you want to copy data blocks to other applications Use, please use [block as TXT file function.

Block Save as

This function outputs the data block you defined as TXT text file, which is convenient for designers to use when programming. The text format is automatically arranged in 16-bit or 8-bit mode. Operation method: After defining the buffer block, right-click the mouse, pop-up menu, select [Block Save as]

Clear the buffer

Empty the current buffer: Function: Empty the buffer displayed in the application's current window.

Clear All Buffers: Function: Clear all the chip's buffers at default values.

Find, find next

Function: Find the HEX or ASCII string in the current buffer. Operation: Press CTRL + F or click **(**Find**)** in the right front menu to pop up the following dialog box:

FIND:	Find next
Format	Cancel
• HEX C ASCII	

Enter the HEX value or ASCII string to find in the input box, and finally click the Find button in the dialog box. When there is a hexadecimal data or string matching this in the program code buffer, the cursor will be positioned to the address of the first content that matches the content being searched, and the F3 key will be pressed to continue searching for

the next one. Until the buffer all search is completed.

Goto:Locate the address

Function: The cursor moves to the address specified by the current buffer.

Operation: Press the shortcut key CTRL + G, or press the right mouse button to pop up the menu to select [Goto], the following dialog box will pop up:

ioto		×
Goto Address 0x: 00000000		
	OK	Cancel

Enter the address, press the OK button, the cursor will move to the address

4.2 Programmer operation function

Programmer operating fu	nctions
Choose the chip	Press the menu list to select the programmed chip
Read the chip	Read into the contents of the chip to the PC cache (Note: NAND read
S.	directly into the file)
Programming chip	Caches the contents of the buffer (NAND file contents) into the
	chip
Erase the chip	Erase the contents of the chip completely
Blank Check	Check if the current status of the chip is not programmed.
Verify	Check the contents of the chip and the buffer is the same, you can
2 (14	mark different.
Set configuration word	Mainly set the protection bit, fuse bits, encryption bits and so
1	on.
Set Serials number	Add a serial number to the chip
Programmer self-Check	Check the programmer programming port function is normal
ICSP programming	ICSP serial programming function interface, and how to use it
RAM test	Test common static RAM chips
Digital logic test	Test 54 74 series, CMOS4000 / MC14xxx series digital integrated
	circuits

4.2.1 Programming settings

Set the options as shown in the following figure, generally default settings can be used, Not necessary to change.

-Options		
🔽 Pin Detect	🔽 Check ID	
🔽 Erase before		
🔽 Verify after	🗌 Auto SN_J	NUM
🔽 Skip OxFF	Addr.Range: 📀 ALL	C Sect
🕅 Blank Check	00000000 ->	0003FFFF

4.2.1.1 Pin Detect

Before operating the chip, the software automatically checks the chip pins for good contact and checks, which in most cases avoids programming or reading errors caused by poor pin contact.

Poor contact with the pin the extreme situation:

When the pin is particularly serious oxidation, the individual pins may not be completely disconnected (for example, equivalent to a 3-10K contact resistance), in this case can not determine whether the contact is good, programming errors may occur, so that to old chip, Should carefully handle the pin.

(Soaked in dilute hydrochloric acid for 30 seconds, removed after washing with water to dry, to achieve the best results)

For very few chips, there may be some pins that can not pass the pin contact check. In this case, this option should be temporarily canceled while ensuring that the pins are in good contact. No pin check will be performed until the chip is read or written.

4.2.1.2 Erase before programming

Before programming, the first run chip erase.

Some chips have no erase function, this option will be disabled. For example, most 27Cxxx chips are OTP ROM or UVROM. Is not erasable or no electric erase function. 24 series 25 series EEPROM 93 series chips, this function is also disabled when there is no special erase command, which means that these chips can be rewritten directly without erasing. (If the user must want to Blank the chip content, that is, 0xFF will be written to the chip's entire storage space)

4.2.1.3 Verification after programming

Under normal circumstances, this must be selected, otherwise the data may be wrong after the programming is completed.

4.2.1.4 Skip writing 0xFF

For FF data block will skip the write operation, can greatly improve the programming efficiency. When this option is verify, the verify will also be skipped

4.2.1.5 Check ID

Many chips have a chip inside the Identification (ID), the content is generally composed of two or more bytes, the first byte is usually the manufacturer code (Manufacture ID), followed by the chip type or chip capacity. Different chip, ID will be different, the function of this option is to prevent incorrect chip into. This option checks the chip ID before reading or writing to the chip. If it continues correctly, this function is optional. Operation will be aborted if the ID error. The default option is enable.

Similar chips not supported in the list of the chip, Their programming method may be same, you can choose other manufacturers IC with the same capacity of the chip for burning. Because the ID will be different, before burning, cancel the ID check

4.2.1.6 Address range

You can set the address range of the programming chip, mass production can improve efficiency.

4.2.1.7 Blank Check before programming

In addition to 27 series UVROM, this generally do not need to select,

4.2.1.8 Auto SN_NUM

See Section 4.2.9 Setting Serials Number

4.2.2 Choose the chip

Select the chip to be programmed and click the <Select IC button> in the upper left corner of the software



Select the chip package must pay attention to the choice. Part of the chip package, pin order may be different. For the same type of package IC if not listed in the various packages, for non-DIP package chip as long as the control IC product manual, direct connection to the corresponding pin can be programmed.

4.2.3 Read the chip

Read the chip: Read data to Buffer from the chip . Different chips read the window may be different, the following to read AT89S8253, for example, shows how to read the chip, click the toolbar [R] button, the following dialog box pops up

Read Range ——	AT89S8253	-Location in Scoket
CODE Memory Start Adr: End Adr:	✓ DATA Memory ✓ LOCK Bit ✓ User.Row ✓ Con.FUSE Bit	
Ptug I	C into 40PIN socket,Click- <read>-Button</read>	

First, set the Read Range, including [CODE], [DATA], [FUSE], [LOCK], invalid when reading the start address and ending address. Press **[**read**]** button to start reading chip data, note: read the value as 0xFF if the chip is encrypted. After the reading is completed, in the information field, each buffer read time is displayed.

Note: The read operation is generally not an error, if you want to verify the read data is correct, then use the verification function verification, specific methods of operation, please check the verification instructions, the programmer on the chip before the operation will first check the chip ID value.

4.2.4 Programming chip

Click the <P> button in the toolbar, the programming dialog box is as follows Take W29C020 as an example, if the pin has bad contact, the following shows the bad pin status

Program Range - W29C020C	Location in Scoket
CODE Memory	
0000000	0
Start Adr: 00000000	
End Adr: 0003FFFF	
Pin Detect error!	
Program Cancel	
mal programming as shown below	
n Program	1
p Program	Si Co
p Program Program Range W29C020C	Location in Scoket
Program Range - W29C020C	Location in Scoket
p Program Program Range - W29C020C CODE Memory	Location in Scoket
p Program Program Range W29C020C CODE Memory Start Adr: 00000000	Location in Scoket
p Program Program Range - W29C020C ↓ CODE Memory Start Adr: 00000000	Location in Scoket
p Program Program Range - W29C020C ↓ CODE Memory Start Adr: 00000000 End Adr: 0003FFFF	Location in Scoket
p Program Program Range - W29C020C ↓ CODE Memory Start Adr: 00000000 End Adr: 0003FFFF	Location in Scoket
p Program Program Range - W29C020C ✓ CODE Memory Start Adr: 00000000 End Adr: 0003FFFF Programming successful !	Location in Scoket
p Program Program Range - W29C020C ↓ CODE Memory Start Adr: 00000000 End Adr: 0003FFFF Programming successful !	Location in Scoket
p Program Program Range - W29C020C ↓ CODE Memory Start Adr: 00000000 End Adr: 0003FFFF Programming successful ! D: 0x DA 45 OK1	Location in Scoket
p Program Program Range - W29C020C ↓ CODE Memory Start Adr: 00000000 End Adr: 0003FFFF Programming successful ! D: 0x DA 45OK! trase chip. Time: 203ms	Location in Scoket
p Program Program Range - W29C020C ↓ CODE Memory Start Adr: 00000000 End Adr: 0003FFFF Programming successful ! D: 0x DA 45OK! Irase chip. Time: 203ms oftware Data Protection DisableOK!	Location in Scoket
p Program Program Range W29C020C ✓ CODE Memory Start Adr: 00000000 End Adr: 0003FFFF Programming successful ! D: 0x DA 45OK! Irase chipTime: 203ms oftware Data Protection DisableOK! Programme CodeTime: 10218ms	Location in Scoket
p Program Program Range W29C020C ✓ CODE Memory Start Adr: 00000000 End Adr: 0003FFFF Programming successful ! D: 0x DA 45OK! Programme codeTime: 203ms oftware Data Protection DisableOK! Programme CodeTime: 10218ms Verify CodeTime: 2481ms	Location in Scoket
Program Range - W29C020C ✓ CODE Memory Start Adr: 00000000 End Adr: 0003FFFF Programming successful ! D: 0x DA 45OK! Grase chip. Time: 203ms oftware Data Protection DisableOK! Programme CodeTime: 10218ms Verify CodeTime: 2481ms oftware Data Protection EnableOK!	Location in Scoket
Program Range - W29C020C ✓ CODE Memory Start Adr: 00000000 End Adr: 0003FFFF Programming successful ! D: 0x DA 45OK! Programme codeTime: 203ms oftwase Data Protection DisableOK! Programme CodeTime: 10218ms Verify CodeTime: 2481ms oftware Data Protection EnableOK!	Location in Scoket
P Program Program Range - W29C020C ↓ CODE Memory Start Adr: 00000000 End Adr: 0003FFFF Programming successful ! D: 0x DA 45OK! Programme Successful ! D: 0x DA 45OK! Programme CodeTime: 203ms Software Data Protection DisableOK! Programme CodeTime: 10218ms Verify CodeTime: 2481ms oftware Data Protection EnableOK!	Location in Scoket
P Program Program Range - W29C020C ✓ CODE Memory Start Adr: 00000000 End Adr: 0003FFFF Programming successful ! D: 0x DA 45OK! Programme Software Data Protection DisableOK! Programme CodeTime: 10218ms Verify CodeTime: 2481ms oftware Data Protection EnableOK! Program	Location in Scoket

The information box shows all

Most chips, the the opperation and an and process for automatic verification, if found during programming errors will immediately terminate the programming.

If the user chooses the "Auto SN_NUM" option, the chip number in the buffer will automatically increase according to the setting after chip programming.

In order to ensure the correct programming data, it is recommended that the user always perform verification.

4.2.5 Erase the chip

Chip Erase

MODEL:	W29C020C	0	10
	Erase successful !		e le
D: 0x DA 45OK! Erase chipTime: 203ms			

Click on the [Erase] button on the erasable chip to pop up the following dialog box:

By default, the chip ID is checked before erasing. For a very few MCUs, some chips can not read the chip ID after being encrypted. In this case, cancel the ID check in the main interface, otherwise the ID check is incorrect. Click the [Erase] button in the dialog box, after erase is completed, the erase time is displayed.

4.2.6 Blank Check

By default, the chip ID is checked before erasing, and the area where the chip is to be checked can be selected first if it is empty. The default is all. Click on the [check empty] button in the dialog box, start to check the empty operation, the completion of the display to check the time of the various regions of the time. If you check empty during the chip was found to be nonempty, then check the empty, and display non-empty area address and store value.

4.2.7 Verify the chip

Before Verify, select the area want to be checked. The default is all. Click the Verify Button in the dialog box to start the verify operation. If all the data are the same, the time spent in verifying each area will be displayed after completion.

If you find that the chip data is different during the verification, the following dialog box will pop up:

Verify Range	W29C020C	Location in Scoket
Start Adr:	Verify	
D: 0x DA 45OK!	Memory Area: CODE MEMORY Address: 0x 0000 0000 Buffer Value: 0x D0 Chip Value: 0x FF Continue Marked Cancel	
_		

Click [Continue]: Will continue to compare if there are different then will pop up the same dialog box prompts.

Click **[**marked**]** : will compare the entire contents of the chip, in the process of comparison, if found different will not pop-up prompts, until done, the number of tips are different, and in the buffer zone in red font shows different units (Except NAND), as shown below.

Click [Cancel]: will exit directly, not for later comparison.

- Verify Range -	_	W2	9020	oc		- Location in Scoket -
CODE M	emory					0
Start Adr:						
End Adr:						
00		414 11 11	54			
	Venity K	esult(differen	icequanti	ty): 6		
ID: 0x DA 45 Verify Code T	OK! me: 5023ms					
, any cost						

4.2.8 Set configuration

Click the [Config] button in the chip buffer selection bar to pop up the following interface (chip type PIC18F4550):

Set Interfac • 40P adapte	e er	C ICSP p	ort	Г	ICSH	_VCC Enabl	e	🕼 8 Bit	5 1	C 16 Bits	- b1	iffer selec Co	et – ode	Memo Data	Mem	Config		
	1	BIT7		BIT6		BIT5	1	BIT4		BIT3		BIT2		BIT1		BITO		
0X300000:	1	-	-	-	1	USBDIV	1	CPUDIV1	7	CPUDIVO	7	PLLDIV2	V	PLLDIV1	~	PLLDIVO		
0X300001:	1	IESO	1	FCMEN		2	1	-	~	FOSC3	Г	FOSC2	1	FOSC1	Г	FOSCO		
0X300002:		-	-		VREGEN		BORV1		Г	BORVO	Г	BOREN1	Г	BORENO	F	/ PWRTEN		
0X300003:		-	_		_		Г	WDTPS3	Г	WDTPS2	Г	WDTPS1	Г	WDTPSO	Г	WDTEN		
0X300004:		-		-		-		-	-			-				~		
0X300005:	Г	MCLRE				-		-		-	1	LPT10SC	Г	PBADEN	Г	CCP2MX		
0X300006:	Г	/DEBUG	~	XINST	V ICPRT		-		-		Г	LVP	5		Г	STVERN		
0X300007:		-	-		-		- 1			-		14	1	~		-		
0X300008:		-		-		-		-		CP3		CP2	Г	CP1		CP0		
0X300009:	Г	CPD	Г	CPB		2		-		-		2		-	10			2
0X30000A:		-		-		_		-		WRT3	Г	WRT2	F	WRT1	Г	WRTO		
0X30000B:	Г	WRTD	Г	WRTB	Г	WRTC		_			5		~	-		-		
0X30000C:		-		-		-			Г	EBTR3	Γ	EBTR2	Г	EBTR1	Г	EBTRO		
DX30000D:		-	Г	EBTRB	1	-		-		6		-		-		-		
ptions						1.1	IC	Config Inf	orm	aton	è							
🗸 Pin Detect			$\overline{\mathbf{v}}$	Check ID				CONFIG. (Ux3	00000-0x50	0001	0)		USERID: (0x2	00000-0x200		
▼ Erase before ▼ Verify after □ Auto SN N			00 05 1F 1F 00 83 85 00 0F C0 0F E0 0F 40					\sim		FF FF F	FF	F FF FF FF						

The content of the configuration information in the lower red circle corresponds to the configuration interface: Click the configuration interface, the value in the configuration is automatically changed.

Different chip configuration information is different, so the user first view the chip's data sheet, and in accordance with the specific needs of the correct configuration of the chip.

Chip configuration word,	need to understand the role of each bit in the configuration												
word													

4.2.9 Setting Serials Number

Serials number setting function is to automatically add a Serials number in t memory area of the chip can be easily used. Add time or a unique serial number, etc., the programmer has a variety of numbering algorithms available to meet a variety of user requirements. Click <Main Menu> --- <Device> --- <Serials Number> to pop up the following dialog box:

AutoInc. mode	- Operations	10	is w	
Default INC	Start Addr:0x	00000000	SN.Length:	08
C Time Mode	original value:		18100073	
C Radom	Inc.Step(10):	1		
C User Define	CODE		-Inc.Format	
Region	(HEX (ASCII	• HEX C	DEC
CODE Memory			Inc.from high a	address
C DATA Memory			-	
C USER.ROW	Record			-
C USERID	Note: Save Serial	s to end of this	file	-

First select the automatic numbering algorithm, in the dialog box there are four algorithms to choose, the following for a variety of numbering algorithm for a separate description.

4.2.9.1 The default incremental algorithm (Default INC)

The setting interface of the algorithm is as shown in the figure above. The function of the algorithm is to add a number of a set length in the address where the set storage area starts, and the number increases by each step. After setting the parameters, you can press the button to test and check the changing rules under various parameter settings.

The initial value and step size can be set in the dialog box.

Set the parameters and test the correct, set the log file, if empty, the log file is invalid, the program will not record the data to the file, if you set the log file, after each programming, put you The set number is added to the end of the log file.

Finally, <Enable Auto SN> checked, and <Save> button.

4.2.9.2 Time Mode algorithm

Time algorithm dialog interface as shown below: At 0x2000, put in a time data

Address	0	1	2	3	4	5 6	7	8	9	A	B	C	D	E	F	ASCI	I	_			_
0000-2000	32	30	31	38	2D :	30 3	2 20	32	37	20	31	34	3A	33 :	33	201	8-0	2-2	7 14	:33	
0000-2010-	FF	FF	FF					-		ee.							92	11-1-1			
0000-2020	FF	FF	FF	Set A	uto Se	rials										-	~				
0000-2030	FF	FF	FF	- A.	stoles e	node	11070	Operatio													
0000-2040	FF	FF	FF	^	stome. i	noue		operation	Jus		2000	_	1000		16	_					
0000-2050	FF	FF	FF	0	Defau	lt INC		Start Ad	dr:0x		2000	_	SN.I	.ength:	10						
0000-2060	FF	FF	FF	6	Time	Mode		original	value:									- (-)			
0000-2070	FF	FF	FF		Padau									Inc St.	en(1)	m.	-				
0000-2080	FF	FF	FF		Radon	n		- Char I	Format						-P(-	-/-1					
0000-2090	FF	FF	FF	0	User I	Define		ΘY	YYY-N	IM-DI	00:00)									
0000-20A0	FF	FF	FF	-				CY	YYY-N	IM-DI) +5 by	tes IN	IC.					- (-)			
0000-20B0	FF	FF	FF	Re	egion —			CY	YYY-N	IM-DI	00:00	+8 by	tes INO	1						/	
0000-2000	FF	FF	FF	6	CODE	E Memo	ny	CY	YYY-N	IM-DI	00:00	+16 1	vtes Ra	dom							
0000-20D0	FF	FF	FF	1	DATA	Memo	ry										-				
0000-20E0	FF	FF	FF	1	TISER	ROW										-					
0000-20F0	FF	FF	FF					Record	:												-
Options				0	USER	Ш		Note: Sa	ve Seria	als to e	end of t	his fil	e						-		
🔽 Pin Detect							₩ Ea	nable Aut	o.SN	(Т	est	ð	Save	1	Cane	el	D: ()	0x2000	00-0x20	000'
🔽 Erase befor	e					-	-				-	-	-		-	\sim	1	FFI	F FF F	F FF FF	FF

There are four time code formats, namely:

- (1) Date + time number Total length: 16 bytes
- When programming add the date and time in the specified location, the total length is 16bytes.
 - (2) Date Increment number of +5 bytes Total length: 16 bytes

The incremental part of the content can be set initial value and step.

(3) Date + time + 8-byte incremental number Total length: 24 bytes

The incremental part of the content can be set initial value and step.

(4) Date + Time + 16 bytes Random Numbering Total length: 32 bytes

Of these, 16 bytes are randomly generated bytes that generate a unique serial number for the product. In this way, the log file must be set so that the generated code is recorded in the file. Otherwise you do not know the product number is what.

The back of the operation steps is same as "default INC"

4.2.9.3 Random algorithm

The algorithm generates a random number of the specified length, as shown below: At the beginning of the address 0x2000, put 16 bytes of random data

Address 0 1 2	3 4 5 6 7 8 9 A B C D E F ASCII
0000-200 <mark>0 77 2D EB</mark>	4A AF 63 EC 48 9A 43 B7 68 21 49 2F 77 wJ.c.H.C.h!1/w
0000-201 0 FF FF FF	
0000-2020 FF FF FF	Set Auto Serials
0000-2030 FF FF FF	- AntoIne mode - Operations
0000-2040 FF FF FF	Automa operations
0000-2050 FF FF FF	C Default INC
0000-2060 FF FF FF	C Time Mode Start Addr:0x 00002000 SN.Length: 16
0000-2070 FF FF FF	G Palan
0000-2080 FF FF FF	(* A200m
0000-2090 FF FF FF	C User Define
0000-20A0 FF FF FF	← HEX C ASCII
0000-2080 FF FF FF	Region
0000-20C0 FF FF FF	© CODE Memory
0000-2000 FF FF FF	C DATA Memory
0000-20E0 FF FF FF	C USER ROW
0000-20F0 FF FF FF	Record:
Options	CUSERID Note: Save Serials to end of this file
🔽 Pin Detect	✓ Enable Auto.SN Test Save Cancel D: (0x200000-0x200
🗸 Erase before	F FF FF FF FF FF FF

4.2.9.4 User-defined algorithm

User-defined algorithm has a strong flexibility, you can set any number of algorithms, the use of detailed look at the directory .. \\ Serialnumber \ source_dll VC ++ source code comments. The interface is as follows:

AutoInc. mode	Operations			
C Default INC	User algorithm	GetSerial_Test18	F4550. dl1	
C Time Mode	Note:Copy the file Algorithm Descr	to \\SERIALNUI	MBER	
 Radom User Define 	[DEVICE] Xingong TL8661	I DLL Test		
Region	PIC18F4550自动 2018-0	≧义序列号测试 □1-11		
C DATA Memory	original value:	18	100073	
C USER.ROW	Record:			
C USERID	Note: Save Serials to	o end of this file		-

AVR microcontroller RC correction byte encryption method, fully use here to call DLL dynamic library,

.. \ TAMEGA_LED directory contains all the source code and test methods.

After making the DLL file, you just copy it to the subdirectory .. \\ Serialnumber in the application directory. Set in the user algorithm file list, select your algorithm file. The following steps are the same as the other algorithms.

important hint:

1. After setting the above, the programmer will not automatically add the number you set to the chip during the programming operation. To add the number automatically, you must also

select <Auto SN_NUM> in the main interface.

- 2. The main interface can be selected in the condition of <Auto SN_NUM> is, you must first set the numbering algorithm, and in the algorithm set <Enable Auto. SN>.
- 3. In the main interface, select <Auto SN_NUM>, if you re-set the numbering algorithm, the main interface <Auto SN_NUM> button must be re-selected.
- 4. After changing the programming chip type, you must reset the number setting algorithm.
- 5. In Project mode, the algorithm is saved in the project file, can open the project file directly, and automatically transferred to the automatic numbering algorithm. Therefore, mass production is generally used in Project mode, simple and convenient.

4.2.10 ICSP programming

about serial programmable chips, the programmer can programming chips through a ICSP interface on the target board .First select ICSP port in the main interface of the "Set Interface options bar", , then [ICSP_VCC_ENABLE] To allow the state, the default [ICSP_VCC_ENABLE] is selected, that means programmer provide VCC power to the target board . Note that the programmer VCC power supply can provide a maximum current of 120ma, if the target board requires more power, please use the target board from the external power supply.

If you do not need the programmer to provide power, do not check the [ICSP_VCC_ENABLE] button, so that when programming the VCC pin will not have VCC power supply. As shown below:

C 40P adapter	🕫 ICSP port 🛛 🔽	ICSP_VCC Enable	C 8 Bits C 16 Bit	For select Code Memo Data	Memo Config	
Address	0 1 2 3 4	5 6 7 8	9 A B C	D E F ASCII	📕 🕹 🛛	B Device Inf
0000-2000	Chip Read				and the second se	
8888-2828 8888-2838 8888-2848 8888-2848 8888-2868 8888-2868 8888-2878 8888-2888	Read Range CODE Memory Start Adr: End Adr:	PIC18F	4550 Nory Bit	Location in Scoket		
0000-2090 0000-20A0	✓ Supply Vcc Power (MA Connect <ic< td=""><td>X 120mA) SP> to target board,Cl</td><td>ick-<read>-Button</read></td><td></td><td>123456</td><td></td></ic<>	X 120mA) SP> to target board,Cl	ick- <read>-Button</read>		123456	
0000-2000 0000-2000 0000-2000 0000-2000	2			6 — 5 — 4 — 3 — 6	Not Connect PGC PGD GND	
Options Pin Detect				2		

When the next operation, reading, erasing, Verify, programming and other operations, it will be operated from the ICSP port. When operation, the 40PIN universal socket can not be inserted into the chip at the same time. Click the [Read] button on the toolbar to pop up the following dialogue Box: (Note that the dialog box for reading, erasing, checking, etc. is the same as this one). ICSP port on the right shows the wiring diagram, a more detailed wiring diagram can click [View ICSPConnection]

ICSP chip support:

- 1、 24 Series 25 Series 93 Series,
- 2、 ATMEL89S51, 52, AVR ATMEGA full range, Note: High-voltage parallel programming, low-voltage ISP programming
- 3、 MICROCHIP PIC10Fxxx 12Fxxx 16Fxxx 18Fxxx full range,
- 4、 New Mao SYNCMOS SM59Dxx SM59Rxx full range of chips

4.2.11 Multi-machine programming

Software supports up to 4 programmers at the same time programming (NAND FLASH not to hold multi-machine programming).

USB Expansion Please choose to use a good quality USB hub with power supply,

Multi-machine programming interface is as follows: Multi-machine programming automatic Serials numbering function can be used normally.

ulti Programming	Server and Server 1 - 1		9
	Multi.Prog. Please use	HUB with Power	2
Programming Option ↓ CODE Memory ↓ DATA Memory ↓ Con.FVSE Bit	₩ Vser ID	Jes. IL	
IC: PIC18F4550 Inter	face: ICSP		
Device 1	Device 2	Device 3	Device 4
Prog. [Z]	Prog. [X]	Prog. [C]	Prog. [V]

Multi-machine programming can use shortcut keys [Z] [X] [C] [V], start the corresponding programmer.

4.2.12 RAM test

Select	the	corresponding model	of th	e RAM	chip,	click	the	toolbar	【test】	button,	the f	ollowing
dialog	box	pops up:										

Test Range —		61512	1	Location in Scoket
Start Adr:		Data Bus	□ Increase ☑ Unit	
End Adr:	Plug IC into	o 40PIN socket,Click- <te< td=""><td>est>-Button</td><td></td></te<>	est>-Button	

RAM test There are 4 ways, through these four methods of testing, basically can test the RAM unit function is normal.

General programmer only read and write to the RAM unit is not tested out of RAM is good or bad. For example, if two adjacent address lines are short-circuited, you are simply reading and writing data to the RAM cell and it is impossible to detect the problem. So just one way to test is incomplete test.

35

Insert the chip, click [test] button, you can test the RAM.

4.2.13 Digital logic test

The programmer may be able to really test the 74 series CMOS4000 series integrated circuit, the test can locate a gate fault.Select the corresponding digital logic integrated chip, click the [test] button in the toolbar, the following dialog box will pop up:

Test Result	-Location in Scoket	
Logic Unit 1 Test Error!		
Logic Unit 3 Test Error!		
Logic Unit 4 Test Error!		
Test Result: Error!		
74HC(LS)00		

The above is the 74HC00 test results, under normal circumstances shows that each logic unit test is normal, the message box shows a variety of input combinations tested. If a gate of the integrated circuit is wrong, for example, the 74HC00 PIN1 is bent and broken. The test result is as follows: The chip is wrong.

Depending on the chip, the number of logic cells is different. The logic unit number is the same as the IC manual.

5.NAND Flash programming instructions

5.1 NAND FLASH features:

- A. NOR Flash Fully random access to memory-mapped and dedicated interface (such as EPROM) address and data lines. NAND flash memory addressless line. It is through the 8/16-bit wide interface bus to send commands, addresses and data to the internal registers, so many master control provides a more flexible configuration, NAND flash structure, emphasizing lower cost per bit, higher performance , And like the disk can easily upgrade through the interface, NAND lower cost, larger capacity.
- B. Nand Flash allows bad blocks to exist. Due to the NAND production process, the chip will randomly appear in the factory bad blocks. The bad blocks have been initialized at the factory and marked as unusable in the special area. If bad blocks appear during use, they also need to be marked.
- C. Easy to bit reversal. NAND FLASH more easy to bit reversal, if bit reversal appears in the key files, will cause the system to hang up. Therefore, in the use of NAND FLASH at the same time, the need to use ECC / EDC and other algorithms to ensure reliability.
- D. Spare zone exists. Because NAND FLASH has the above two special places, Spare area plays the role of deposit bad block mark, ECC value and chip information and file information.
- E. NAND FLASH must be bad block processing

NAND FLASH bad blocks deal with a lot of different companies or system providers will choose different bad block handling methods to meet the needs of product development. TL866II programmer defines the commonly used three methods for the user to choose to use. And the combination of user-defined ECC algorithm for flexible applications.

5.2 Main interface:

Select a chip, the software interface as shown in Figure 1-1:



- 1、 Click this button <Select IC>
- 2、 Custom <BBT> Bad Block Table button

TL866II programmer, set an optional block size custom table, the table can generally be used as a dedicated bad block table BBT, when the contents in the file is burn finished, followed by the contents of this table can be written to and location that user set in the configuration interface. However, the block index number written by BBT parameter settings need to ensure that the address blocks do not overlap.

- (1) The data in the BBT can be changed in the DLL custom function according to the actual burning result of the chip.
- (2) The BBT can be loaded with initial values: The file can be opened in the File menu and the data is pre-loaded into the BBT table.
- (3) This table is ignored if the option Enable BBT Programming to Block is not selected in the programming configuration.
- 3. Before reading the chip: you need to first set the file name of the archive When programming a chip: Set the source data file name to be written to the chip. Note: In NAND programming, the data is written directly to the chip from the set <Source file>. Read the contents of the chip is saved directly to the file. This operation is different from other chips, other non-NAND chip operations: programming, the buffer data (in memory) is written to the chip. Read the chip data is read into the computer buffer. When saving a file, the data in the cache is saved to a disk file. Open the file is that the disk file data transferred to the computer memory.

Figure 9: Array Organization - MT29F4G08 (x8)

In NAND, <save file> and <open file> are operations on the BBT table's memory buffer.

This document MT29F1G08ABAEA, for example, describes the TL866II programmer NAND FLASH burning method

The MT29F1G08 consists of 1024 blocks, the TL866II programmer operates on a block basis with index numbers 0-1023. Each block has 64 pages (pages), each block is processed in the order of pages written to the chip.

Each page size: 2048 bytes (Page size) + 64 bytes (Spare size) = 2112 bytes

The total chip capacity is: 2112 * 64 (pages) * 1024 (Block Block) = 138,412,032 bytes.



5.3 Configuration interface

As shown below, click <Config> button to enter the programming Configration interface. The exact configuration of the programming parameters is the key chip can be programming normally.

Figure 1-2:

Salect IC IC Information(No Project opened) MI29FIG08ABAEA @TSOP48 IC Information(No Project opened) Set Interface C Information(No Project opened) G 40P adapter C ICSP port ICSP.VCC Enable G 8 Bits C 16 Bits BBT Config Spare Area In File Partition EN Start of BLK(Addr) End of BLK (Addr) Counts.R/W(DCC) None device found. MI29FIG08ABAEA With Start of BLK(Addr) End of BLK (Addr) Counts.R/W(DCC) None device found. Miss Block Fast of BLK(Addr) Counts.R/W(DCC) None device found. G 1nclude spare area Fast of BLK(Addr) Counts.R/W(DCC) None device found. G 1nclude spare area Fast of BLK(Addr) Counts.R/W(DCC) None device found. G 1nclude spare area Fast of BLK(Addr) Counts.R/W(DCC) None device found. G 1nclude spare area Fast of BLK(Addr) Counts.R/W(DCC) None device found. G 1nclude spare area Fast of BLK(Addr) Counts.R/W(DCC) None device found. G 1nclude spare area G 0x00000000 0x000000550 Ox000000550 Fast of Block G 1nclude spare area G 0x00000000 0x00000000 Ox000000000 Fast of B	; 🖬 👫 j 🛃 🔠 💷 🕫 🧏) 🕀 📇 🗶 💡 🔰 📔 📴 🛄	
Spare Area In File Partition © Include spare area EN Start of BLK (Addr) End of BLK (Addr) Counts. R/W (DEC) Washington V <1>2 0x000080000 19 0x004FF83F Bad Block V <1>2 0x00000000 50 0x00001F83F 0 V <2> 0 0x00000000 0 0x0000000 0 0x0000000 V <2> 0 0x00000000 0 0x0000000 0 0x0000000 V <2> 0 0x00000000 0 0x00000000 0 0x00000000 V <2> 0 0x00000000 0 0x00000000 0 0x00000000 V <1> 0 0x00000000 0 0x00000000 0 Mark BB when verifi.ERR Verifi.ERR Verifi.ERR Verifi.ERR Verifi.ERR Verifi.ERR Config Informaton V Enable EB bB T Programming to Block: 0 (0x00000000) Vox00000000) Vox00000000 Verify after Anto SN_NWM Select File Select File Select File Verify after Anto SN_NWM </th <th>eleot IC</th> <th>ormation(No Project opened) we: EEPROM ChkSum: 0x020D F000 e: 0x8400000 Bytes (138412032 Bytes) SP_VCC Enable</th> <th>Clear</th>	eleot IC	ormation(No Project opened) we: EEPROM ChkSum: 0x020D F000 e: 0x8400000 Bytes (138412032 Bytes) SP_VCC Enable	Clear
Wark DB will VeIII.LEAR User Define [Process ECC and BBI via DLL Function] Image: Permissible Err bits Image: Programming to Block: 0 (0x00000000) Image: Process ECC and BBI via DLL Function] Image: Process ECC and BBI via DLL Function] Image: Process ECC and BBI via DLL Function] Image: Process ECC and BBI via DLL Function] Image: Process ECC and BBI via DLL Function] Image: Process ECC and BBI via DLL Function] Image: Process ECC and BBI via DLL Function] Image: Process ECC and BBI via DLL Function] Image: Process ECC and BBI via DLL Function] Image: Process ECC and BBI via DLL Function] Image: Process ECC and BBI via DLL Function] Image: Process ECC and BBI via DLL Function] Image: Process ECC and BBI via DLL Function] Image: Process ECC and BBI via DLL Function] Image: Process ECC and BBI via DLL Function] Image: Process ECC and BBI via DLL Function] Image: Process ECC and BBI via DLL Function] Image: Process ECC and BBI via DLL Function] Image: Process ECC and BBI via DLL Function Image: Process ECC and BBI via DLL Function] Image: Process ECC and BBI via DLL Function Image: Process ECC and BBI via DLL Function] Image: Process ECC and BBI via DLL Function Image: Process ECC and BBI via DLL Function] Image: Process ECC and BBI via DLL Function Image: Process ECC and BBI via	Spare Area In File Pail © Include spare area EN C none V Bad Block Handle Mode V C Skip Bad Block V C Hard Copy BB to Block(Addr) 900 0x0E100000	Counts.R/W(DEC) Start of BLK(Addr) End of BLK (Addr) Counts.R/W(DEC) <1> 2 0x00080000 19 0x004FF83F 5 <2> 20 0x000500000 50 0x000EF83F 10 <3> 0 0x00000000 0 0x0003F83F 0 <4> 0 0x0000000 0 0x0003F83F 0 <5> 0 0x0000000 0 0x0003F83F 0	USB Device Info: None device found. MT29FIG08AB&EA @TS0F48 PageSize : 2048 SpareSize : 64 Pages Per Block : 64 BlockSize : 1024 Config Button
Options IC Config Information IC Pin Detect IC Config Information IC Ense before Source File D:\Xgpro\UserData\222. BIN Select File Source File D:\Xgpro\UserData\111. BIN IC Skip OxFF Addr. Range: @ ALL C Sect Save to File D:\Xgpro\UserData\111. BIN Select File	Mark DD when vermi.ERR Usi	er Define (Process ECC and BBT via DLL Function) Enable BBT Programming to Block: 0 (0x00000000) Enbale User Algorithm file: ECC-256Bx2-SMALL.dll	
Skip OxFF Addr. Range: C ALL C Sect Save to File D:\Xgpro\UserData\111.BIN Select File	ptions F Fin Detect	IC Config Informaton Source File D:\Xgpro\UserData\222.BIN	Select File
Diald check $O_X = 0000000 \rightarrow 033FFFF$	F Skip OxFF Addr.Range: @ ALL C Blank Check Ox 00000000 -> 0	Sect Save to File D:\Xgpro\UserData\111.BIN	Select File

5.4 Configuration instructions

Spare Area In file

Include spare Area

Due to the special nature of NAND FLASH, the data in the user file contains the Spare Area data according to the actual situation.

If not include:

Spread Area (redundant area 64 bytes) data will not be saved in the saved file when reading the chip contents.

When programming (writing) a chip, the programmer only reads PageSize (1024) bytes of data from the file, and the next 64 bytes of data (SpareSIze) will default to blank data (OxFF). This situation, the general need to use user-defined algorithm files, the data in the redundant area according to user requirements correctly filled in.

Bad Block Handle Mode

There are three ways to deal with bad blocks:

Skip Bad Block:

In the process of reading and writing chips if found bad blocks, TL866II programmer will skip the bad block directly to the next block of data to read or write. Write Flow as shown below Table 1-1:



Hard Copy: Forces the data to be read or written regardless of whether or not the block has a bad block mark, and stops programming if an error occurs during programming. This method is usually technical maintenance, the use of memory chips replacement method.

BB to Block: Found bad blocks written to the beginning of a particular block, so the advantage is to make the system logically feel the memory is still a continuous area, usually used in conjunction with the partition (Partition). In the following example configuration details. Enter the number in the edit box (Example: 900) is the replacement block index (address) actually written to flash when a bad block is found

Mark BB When Verify Error :

When a parity error occurs after programming the data, this block is automatically marked as bad and the data is written to the next block (Skip Bad Block), or the data is written to a specific block . (BB to Block alternative address)

If not, the programming will be terminated when the verification error occurs.

Permissible ERR bits

None: does not allow an error to occur

1bits / 256bytes: One bit error per 256 bytes. Bit flipping may occur when reading NAND FLASH chip and will not be considered as an error if a bit error (256 consecutive bytes) is detected during verification. However, spare area in the data is still not allowed to appear any bit error.

Partition

Divide the file contents into NAND FLASH. Or according to the requirements of the partition

Haikou Xingong Electronics Co., Ltd. <u>http://www.xgecu.com</u>

read into the file.

The software allows the file content is set to 5 partitions, programming respectively.

 $E\!N$ Options: Select whether to allow the partition.

Start of BLK: partition start block index number

End of BLK: partition end block index number

Counts of R / W: The number of blocks the programming software actually reads or programming to this partition.

Partition parameter setting requirements:

- A. All the block index number, can not be greater than NAND FLASH actually contains the number of blocks.
- B. Start of BLK \leq End of Block, Counts of R / W \leq (End of Block s Start of BLK + 1) The number of blocks actually written during partitioning (Counts of R / W) is preferably smaller than the block size of several blocks. Make it have some margin
- C. The Start of BLK for the next partition should be larger than the End of Block for the previous partition

If you use the **BB To Block** option, the Bad Block replacement address should not be included in any of these partitions. If you use a BBT table, the BBT tables should not be included in any of these partitions.

Enable BBT Programming to Block xx

The data in the BBT table is programmed to xx BLOCK

When this option is enabled, the programmer finally writes the contents of the BBT table to the specified xx block after all partitions (or all file data) have been written to FLASH.

Enable User Algorithm file

Enable user-defined algorithm file

In the custom algorithm file, the user can process any data in the block before the data processing or automatic serial number calculation, as well as ECC algorithm and so on. BBT table data can also be automatically filled according to the bad blocks in the programming process, so that the user can verify that the application reads the contents correctly.

Specific methods of operation, please refer to DLL dynamic function of the original code comments.

5.5 Configuration Example 1 (MT29F1G08)

- 1) Programming the flash with 2 partition table.
- 2) Bad block processing mode: Skip bad block.
- 3) File data: include spare area (generally the original data file needs to be handled Spare area)
- 4) Use 2 partitions

1st partition: from 0 # -199 # A total of 200 BLOCK, the actual data length is 10 BLOCK The second partition: from 1000 # - 1023 # A total of 24 BLOCKs, the actual write data length is 5 BLOCKs

The written data file is: 222.BIN (Note: NAND programming can only use binary format files)

Through the above settings, the programming operation can be performed.

If the file is not long enough, then the programming is finished.

If the length of the file exceeds the length of the BLOCK (15 blocks) data to be written, programming is completed after the set number of BLOCKs have been written.

Configuration interface as shown below: Figure 1-3:

🐸 🖬 👘 j 🖾 🖪 🗐 🎙) ※ H 📇 🗷 🕴 🖉 🔝 🛄	
Select IC	C Information(No Project opened) ipType: EEPROM ChkSum: 0x020D F000	
MIZSFIGORADAEA BISUI40	C Size: 0x8400000 Bytes (138412032 Bytes)	3 <u></u> t 11101117
• 40P adapter C ICSP port	ICSP_VCC Enable C 8 Bits C 16 Bits butter select BBT BBT	Config
Spare Area In File	Partition	USB Device Info:
← Include spare area	EN Start of BLK(Addr) End of BLK (Addr) Counts.R/W	J(DEC) WT29F1C08ABAFA @TS0P48
C none	▼ <1> 1 0x00040000 199 0x031FF83F 10	- PageSize : 2048
Bad Block Handle Mode	▼ <2> 1000 0x0FA00000 1023 0x0FFFF83F 5	- Pages Per Block : 64 BlockSize : 1024
• Skip Bad Block	C <3> 0 0x0000000 0 0x0003F83F 0	
C Hard Copy		-
BB to Block(Addr)		
Had DD that well CDD		
Disable C Enable	User Define (Process ECC and BBT via DLL Function)	
Permissible Frr bits	Enable BBT Programming to Block: O (0x0000000)	
• None C 1bit/256byte	Enbale User Algorithm file: ECC-256Bx2-SMALL.dll	
Options	IC Config Informaton	11
V Pin Detect V Check	ID	
✓ Erase before	Source File D:\Xgpro\UserData\222.BIN	Selec
Verify after Auto	N_NUM	
♥ Skip Uxff Addr. Range: ⓒ ALL	C Sect Save to File U: \Xgpro\UserUata\111.BIN	Selec
Blank Check Ox 00000000	> 083FFFFF	

0# block 1th partition start 1# Block
- 1# Block
2#7# block
_
- 8# Bad Block skipped
– 9# block
- 10#
11# 10 blocks end
·····.
5
1000# Block 2th Partition
start
1001# bad Block skipped
~ 1002#
1003#
1004#

The operation results are shown in Table 1-2:

note:

In this configuration example, read or verify the chip, always by partition. When reading:

- 1. Read 10 blocks saved to the file from Partition 1.
- 2. Read 5 blocks saved to the file from Partition 2.
- 3、 Read completed.

With setting up partitions flexibility, you can read or write a few or a single BLOCK of data individually,

5.6 Configuration Example 2 (K9F1208U0C)

To K9F1208U0C example chip capacity: (512 +16) * 32 * 4096 blocks K9F1208U0C parameter:: Page Size: 512 bytes Spare size: 16 bytes Pages Per Block : 32 pages Block Size : 4096 1) Use (Partition) partition table NAND FLASH into the file write operation Bad block processing mode: BB to Block 4000, found a bad block, write block 4000 # 2) start position File data: not include spare area (Spare area in the DLL processing content) 3) 4) Using 2 partitions, 1ST partition: from 0 # -1000 # A total of 1001 BLOCK, the actual length of 100 BLOCK. 2ST partition:from 2000 # - 3999 # A total of 1000 BLOCK, the actual length of 3 BLOCK. 5) The written data file is: 222.BIN (Note: NAND programming can only use binary format files) 6) Enable BBT Programming to Block 0. The contents of the BBT table are assigned to 0 # Block. 7) Enable User Algorithm file, enable the algorithm file, ECC 256Bx2-SMALL.dll, (the DLL instance only calculates the ECC value into the spare area, the user can modify the DLL function according to the actual need, modify the contents of the BBT table) 8) Proramming Flow shown in Table 1-3. Figure 1-4:

🖻 🖬 👫; 😓 🖪 柳 🦻	D 3D 🔃 📇 🗶 🤋 🔰 🔛 🛄	
Select IC	IC Information(No Project opened)	
K9F1208U0C @TS0P48	ChipType: EEPROM ChkSum: 0x0041 BE00 IC Size: 0x4200000 Bytes (69206016 Bytes)	🛄 Information
Set Interface • 40P adapter C ICSP port	LICSP_VCC Enable C 8 Bits C 16 Bits BBT Config	Clear
Spare Area In File C Include spare area Read Block Handle Mode C Skip Bad Block Hard Copy BB to Block (Addr) 4000 0x03E80000 Mark BB when verifi.ERR C Disable C Enable Permissible Err bits C None C 1bit/256byte	Partition EN Start of BLK(Addr) End of BLK (Addr) Counts.R/W(DEC) V 1 0x00004000 1000 0x00FA3EOF 100 V 2000 0x01F40000 3999 0x03E7FEOF 3 <3> 0 0x0000000 0 0x00003E0F 0 <4> 0 0x0000000 0 0x00003E0F 0 <5> 0 0x0000000 0 0x00000000 0 Vser Define (Process ECC and BBT via DLL Function) Image: Algorithm file: ECC-256Bx2-SMALL.dll Image: Algorithm file:	: F48 : 512 : 16 : 32 : 4096
Options Options Fin Detect Fin Detect	SN_NUM C Sect -> 041FFFFF	Select File
a adv	Hardware Interface Ver: No Device!	0000 0000

45

Haikou Xingong Electronics Co., Ltd. http://www.xgecu.com

Table 1-3: Programming Flow:



Note: 16 bytes Spare Area Data get from DLL function or all are OxFF

5.7 Save the project file

All the configuration parameters and BBT table contents can be "saved project" after the setting is completed. When it is used again, it can be directly programmed after opening the project file and is suitable for mass production.

6.NAND Flash Copy Method

6.1 Attention

In the process of repairing the equipment, we often need to read data from a mother chip, copied to a new chip, and then welded to the circuit board to run, in order to ensure the chip can operate normally, the user needs to pay attention to the following point:

- Configuration: with TL866II programmer software default settings. The following figure options, you must use Hard Copy, and the file data must include: include Spare area, all other options are default. When read in this configured mode, the programmer software can read the entire contents of the chip (including the contents of the bad block) to the file.
- 2) Some NAND FLASH chips have a unique ID, embedded system applications may read the unique ID, and encryption operations in the program, because the unique ID is the chip factory has been set when the factory, Can not be changed, can not be copied. In this case, even if you copy the entire contents of the chip correctly (Include the contents of the OTP Area described later), the new chip will not work properly.

Example: MT29F4G08ABA has 16 + 16 bytes unique ID, can only read, can not be rewritten. Whether the application uses this method for encryption requires the user to determine whether the chip has a unique ID according to the chip's datasheet. If yes, and if Unique ID encryption is used, this chips can not be replaced unless you can modify the embedded system's software.

3) OTP Area (one-time programmed memory area)

OTP Area is only part of the Nand Flash chip exists, specifically whether or not to see the relevant chip data sheet

Example: The MT29F4G08ABA has 30 pages (30Pages * 2112 bytes) of OTP data. The TL866II does not yet have the time to program and read the OTP Area of some NAND FLASH chips. If the OTP Area is used in the application, The OTP data can not be copied.

(If you need to copy OTP, you can contact us)

6.2 Copy steps

6.2.1 Step 1: Choose files and settings

Select or enter the file name, <save to file> and <Source file> have the same file name Spare Area In File: Include Spare Area

Bad Block Handle Mode: Hard Copy

All other options are default settings. Figure 1-5 shows the configuration.

MT29F1G08ABAEA @TS0P48	LL information(No Froject opened) ChipType: EEFROM ChiSun: 0x020D F000 IC Size: 0x8400000 Bytes (138412032 Bytes)	La Information
Set Interface • 40P adapter C ICSP port	ICSP_VCC Enable C 8 Bits C 16 Bits buffer select	Clear
Spare Area In File Include spare area Include spare area Bad Block Handle Mode C Skin Bad Block Hard Copy BB to Block (Addr) Include (MagEE00000)	Partition EN Start of BLK(Addr) End of BLK (Addr) Counts.R/W(DEC) <1> 0 0x00000000 7 0x001FF83F 1 <2> 8 0x00200000 12 0x0003F83F 2 <3> 0 0x00000000 0 0x0003F83F 0 <4> 0 0x0000000 0 0x0003F83F 0 <6> 0x00000000 0 0x0003F83F 0	USB Device Info: None device found. MT29FIG08ABAEA @TS0P48 FageSize : 2048 SpareSize : 64 Pages Per Block : 64 ElockSize : 1024
Mark BB when verifi.ERR Disable C Enable Permissible Err bits None C 1bit/256byte Options	User Define (Process ECC and BBT via DLL Function) Enable BBT Programming to Block: 0 (0x00000000) Enbale User Algorithm file: ECC-256Bx2-SMALL.dll IC Config Informaton	
♥ Fin Detect ♥ Check ♥ Erase before ♥ ♥ Verify after □ Auto ♥ Skip 0xFF Addr.Range: @ ALL □ Blank Check 0x	: ID SN_NUM . C Sect -> O83FFFFF	Select File Same Select File
Ready	Hardware Interface Ver: No Device!	0000 0000

6.2.2 Step 2 : Read the chip

Read Range ——	MT29F1G	08ABAEA	Location	ı in Scoket	
CODE Memory				n	
Start Adr:				—řeel	
End Adr:					
Use	Adapter=>TSO	P48_NANDx8			
Phig	IC into 40PIN socket,Cl	ick- <read>-Button</read>			
			5		
				TEODAS NAMO	10

Click the $\langle R \rangle$ button on the toolbar and use a NANDO8-specific adapter to place the chip. The following dialog box will pop up

Note: This program has a pin contact check function, if there is poor contact with the use of the pin, there will be bad tips, a small number of pins on individual chips may not be able to use the pin to check, to ensure good contact can be Temporarily disable the pin check function (in the lower left corner of the main interface programming options).

Until finished reading. After the reading is completed, you can click < Verify> in the main menu to perform a data verification to ensure the correctness of data read in. (The verification speed may be much faster than reading Speed is a normal, because TL866II programmer software will skipped blank block automatically)

6.2.3 Step 3: Programming new chip

编程范围——	MT29F1G08ABAEA	放置芯片位置	
CODE Memory	y	0	
开始地址: 000	00000		
结束地址: 083	FFFFF		
专,	用适配器=>TSOP48_NANDx8		
	正在编程 Code Memory		
ID: 0x 2C F1 80 95 . 删除清空芯片耗时	OK! 1: 4446ms		
0.7%			

Replace the new chip, click the <P> button on the toolbar, the following interface pops up: 芯片编程

Note:

- A. Writing chips may be faster than the reading , not surprising, because the software automatically skip the blank area in programming .
- B. In reading, deleting, burning process, if there is a bad block , Bad Block Messages will appear in the right message box , but it does not affect reading and writting.
- C. No necessary to check the bad block information of source chip and the new chip before copy , as long as the programming is correct, there is no error to stop. the new chip will be available. if there is a bad block in the new chip, the software found the Data is empty data here, the software can automatically skip. If the new chip has a bad block, the source chip has data that needs to be written to

this bad block. . When copying, it can not be written to this bad block. The programming is terminated. Please switch to another new chip to programming.

Because of maintenance technicians, generally do not know how the chip data structure, how to deal with bad blocks. So only hard copy directly, the TL866II programmer software process blank data by skipping, so to avoid the problem that bad blocks of new chips can not be written.

D. Programming and verification in NAND Flash programming is done in blocks, as long as the option "Verify after programming" is selected in the lower left corner of <Programming Options>. After the programming is completed, the verification is completed at the same time, and the verification does not need to be done again.

Haikou Xingong Electronics Co., Ltd. http://www.xgecu.com



6.3 File address calculation method

Calculated in two cases:

File contains Spare Area
 Data Location in File = nBlock *(Page Size + Spare Size) * Number of Pages per Block

51

2) File does not contain Spare Area Location of the data in the file = nBlock * Page Size * Pages per block

7.Appendix

7.1 Disclaimer

- Products that have been repaired, manipulated or altered by any organization or individual not duly authorized by us are not covered under the warranty.
- (2) Locks and other products in the man-made damage, not under the responsibility of repair.
- (3) Due to hardware failures or software defects caused by the expansion of the joint responsibility.

Special

Due to the programmer, many kinds of chip support, complex software system, errors or defects are inevitable. In mass production, special OTP one-time programming chip, you must confirm that the programming is correct, in order to mass production. The company is not responsible for the joint expansion losses caused by hardware failure or software defects. If you find a problem or have a good opinion of our products, please contact us.

7.2 Warranty

Within the normal range of use, from the date of purchase of the product within one year, if the product fails can be free warranty.

7.3 Contact us

With the software update, the contents of this manual are for reference only, product application software shall prevail.

You can download the latest application at <u>http://www.xgecu.com/en</u>.

Name: Haikou Xingong Electronics Co., Ltd. Address: Haikou City, Hainan Province, 80 QiuHaiDaDao, JInRen Hotel on the 4th floor Zip code: 570311 EMIAL: <u>532007590@QQ.com</u> TEL: 0898-68681816

7.4 FAQ (frequently asked questions)

1. Device unavailable or error

Usually the USB driver is not installed, or the driver is prohibited by anti-virus software. At this moment, the yellow light on the USB programmer blinks rapidly. Solution: Refer to section 2.1.2, reinstall the USB device driver.

2. There was an error updating the firmware refresh, unable to refresh

Usually use the USB hub, do not use an external hub when refreshing, the device plugged directly into the computer's USB port, upgrade refresh. If you do not still solve the problem, you try to refresh it on another computer.

3. Pin Detect Errors

Case 1: the old chip, there will often lead to poor contact tips, carefully handle the pin.

Case 2: If using an adapter, use a multimeter to measure if the adapter pins are normal.

Case 3: Individual pins of individual chips may not support the pin contact check. In this case, the pin check function may be temporarily canceled in the lower left corner of the main interface.

4. ID error detected

Many chips have a chip inside the Identification (ID), the content is generally composed of two or more bytes, the first byte is usually the manufacturer code (Manufacture ID), followed by the chip type or chip capacity. Different chip, ID will be different, the function of this option is to prevent incorrect chip into. This option checks the chip ID before reading or writing to the chip. If it continues correctly, this function is optional. Operation will be aborted if the ID error. The default option is enable.

Similar chips not supported in the list of the chip, Their programming method may be same, you can choose other manufacturers IC with the same capacity of the chip for burning. Because the ID will be different, before burning, cancel the ID check

For a very small part of the microcontroller, ID can not be read after Encrypting the chip . At this time, the ID check should be canceled in the main interface, otherwise, the ID check is incorrect.

5. Programming about 27C Series while VPP voltage is 21V or 25V

TL866II programming voltage VPP is only supported up to 18V, for the early 27C series ROM with 21V or 25V VPP voltage no longer be supported writing, but can be read. If you want to use this kind of IC, you can read form the IC(Vpp is 21V or 25V). and buy a new IC (VPP is 13V) to replace it. 27Cxx Serials Chip can be replaced directly. Do not use the external power supply to enhance the programming voltage for forced programming, when the chip is damaged, 21V / 25V high voltage may go back to the programmer's internal, will damage the programmer hardware.