# ALL-100 Universal & Gang Programmer User's Manual

Oct. 2003

#### © 2003 HI-LO SYSTEM RESEARCH CO., LTD.

Information provided in this document is proprietary to HI-LO System Research Co., Ltd. (http://www.hi-lo.com.tw)

The information in this publication is believed to be accurate in all respects at the time of publication, but is subject to be changed without notice.

HI-LO Systems assumes no responsibility for any errors or omissions, and disclaims responsibility for any consequence resulting from the use of the information included herein.

#### Trademarks:

HI-LO is a trademark of HI-LO System Research Co., Ltd. Windows 98/Me/2000/XP/Server 2003 are trademarks of Microsoft Corporation. IBM is a trademark of IBM Corporation. AMD is a trademark of Advanced Micro Devices Inc. All other trademarks are the property of their respective owners.

# Contents

1. Intruduction	1
1.1 Programmer and Accessories	1
1.2 PC System Requirements	3
1.3 ALL-100 Specification	4
2. Single ALL-100 Installation	5
2.1 Hardware Installation	5
2.2 USB Driver Installation	9
2.3 Software Installation	14
3. ALL-100 Basic Operation	17
3.1 Getting Started	17
3.1.1 Start with XACCESS	17
3.1.2 Select IC Manufacturer	18
3.1.3 Select IC Protect Group and Product Type	19
3.2 Load file to Programmer buffer	21
3.3 Read contents from Master IC to buffer	
3.4 Program buffer contents to IC	
3.5 ADAPTERs and CONVERTERs	25
3.5.1 ADAPTER and CONVERTER installation	26
4. Multiple ALL-100 Installation	
4.1 Multiple ALL-100 Operation	
4.2 Read/Compare on Multiple ALL-100s	30

5. ALL-100 Gang Option Installation	
5.1 ALL-100 Gang Operation	32
5.2 Read/Compare on Multiple ALL-100 Gangs	33
6. ALL-100 Software Description	
6.1 XACCESS User Interface	
6.2 Utilities	35
6.2.1 Hex to Bin Converter	35
6.2.2 Bin to Hex Converter	36
6.2.3 2-way splitter	37
6.2.4 4-way splitter	38
6.2.5 2-way shuffler	
6.2.6 4-way shuffler	40
6.2.7 Version List	41
6.2.8 Cross Reference	42
6.2.9 Device List	43
6.3 Programming Data /Function Description	44
6.3.1 File	45
6.3.2 Edit	50
6.3.3 Operation	64
6.3.4 USB Info	72
6.3.5 About	73

7. ALL-100 Troubleshooting74	4
8. Glossary7	5
8.1 EPROM, EEPROM, BPROM, and MPU7	5
8.2 PLD, PAL, GAL, PEEL, CPLD, EPLD, and FPGA74	9

### 1. Intruduction

This manual describes how to install and operate ALL-100 Programmer under environment of PC Windows 98/Me/2000/XP/Server 2003. ALL-100 Programmer works with PC through USB 1.1/2.0 (Universal Serial Bus) to perform high speed data transmission. The high speed processor in programmer precisely controls programming timing and flow, this ensures accurate programming waveforms always generated on ALL-100.

ALL-100 is embedded with 4Mbit memory to support the programming capacity for most E(E)PROM, MCU/MPU, and PLD. The software automatically uses PC memory as buffer to support high density memory ICs.

#### 1.1 Programmer and Accessories

Each ALL-100 package contains following standard accessories:

- Base Unit and single socket Programming Module (M1-UN-DP48).
- An AC power cable, 1.8 M in length.
- A USB cable (Type A to Type B), 1.0 M in length.
- A CD-ROM for Driver Files.
- User's Manual.

#### ☆ Option 99

Price deduction for requesting another Programming Module to substitute M1-UN-DP48 as standard accessary.

 $\stackrel{\wedge}{\searrow}$  Optional Accessories:

M8-FLASH-TS48	8 sockets Programming Module dedicated for TSOP 48 PINs FLASH.
ADAPTER	Support high pincount devices from 48 to over 300 pins. See ADAPTER LIST on HI-LO Web site.
CONVERTER	Convert signals from DIP to different package types such as PLCC, QFP, SOP, TSOP, BGA, etc. See CONVERTER LIST on HI-LO Web site.

#### 1.2 PC System Requirements

- PC/Pentium above.
- Microsoft compatible mouse.
- A hard disk with at least 20 Mbyte free space.
- A CD-ROM drive with speed x2 or above.
- At least one USB port (Version 1.1/2.0)
- Equip 64MB memory space or above.
- Operating System: Windows 98/Me/2000/XP/Server 2003.

### 1.3 ALL-100 Specification

Device Support	Pin Count : from 8 pins up to over 300 pins				
	Device Type: EPROM, EEPROM, Serial PROM, FLASH,				
	PLD/CPLD/FPGA, MPU/MCU, etc.				
Device Contact	Default: DIP48,Textool				
	Others: SOP, TSOP, PLCC, QFP, MLF, SDIP etc. through				
	optional CONVERTERs or ADAPTERs				
Max Sockets in parallel	8 sockets on optional GANG Programming Module				
Controller	16 bits high-speed controller with big sized FPGA $\&$				
	CPLD				
Interface Port	1 x USB port				
Data Transfer Rate	USB 1.1 : 12 Mb/s				
	USB 2.0 : 480 Mb/s				
Max Sites in parallel	up to 8 via tiered star USB				
Functions	Load file,Read Master,Program,Verify,Auto,ID Check,				
	Checksum,Blank Check,Erase,Protect/Unprotect,				
	Secure, Edit, Function Configuration, Self Test				
Host Computer	An Intel Pentium or compatible processor with 64MB				
Requirements	of RAM				
	<ul> <li>At least one USB port available (V 1.1/ 2.0)</li> </ul>				
	<ul> <li>20 MB free hard disk space with Windows 98/Me/</li> </ul>				
	2000/XP/Server 2003 operating system				
	CD-ROM Drive				
Power	AC voltage : 100-240 VAC				
	Frequency : 50-60 Hz				
	Power consumption : 50W				
Dimension	L x W x H : 260mm x 150mm x 100mm				
Weight					
Operating Temperature	4 kg				
operating remperature	4 kg 0- 40 ℃ ( 32-105 °F )				

# 2. Single ALL-100 Installation

#### 2.1 Hardware Installation

Before installation, make sure your PC has USB 1.1/2.0 port which can be connected to ALL-100 Programmer through USB cable. Suggest to use USB 2.0 for fast data transmission with ALL-100 Programmer.

USB 1.1 (Full Speed) : Transmission rate 12 Mb/s USB 2.0 (High Speed) : Transmission rate 480 Mb/s

#### Step 1:

Make sure power of Base Unit is in "OFF" state and then positioning and connecting Programming Module onto Base Unit.

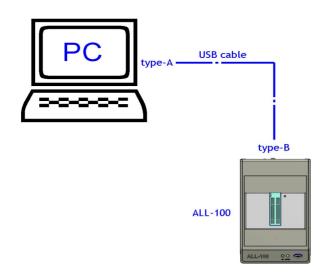
See figures below:



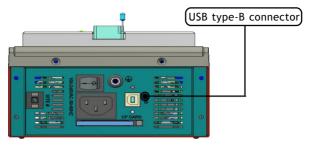


#### Step 2:

Connect programmer and PC as figures below.



Connect the Type B end of USB cable to the USB Type B connector at rear panel of programmer, connect the Type A end of USB cable to the USB Type A connector on the PC.

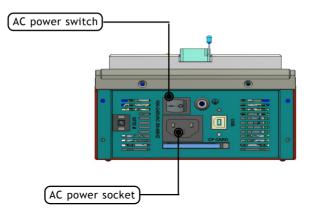


ALL-100 rear panel

Step 3:

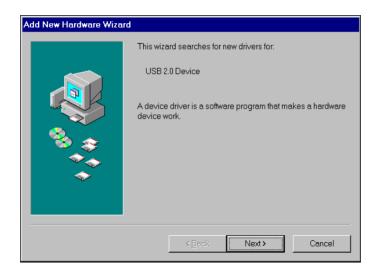
Connect power cable to AC power socket of ALL-100 Programmer and plug in the other end to the outlet of power source (100-240VAC 50-60 Hz).

Power on the ALL-100 from the switch above the AC power socket on the rear panel.



#### 2.2 USB Driver Installation

Insert the Driver Files CD into CD drive, connect USB cable between PC and ALL-100, power on ALL-100 Programmer, the PC will detect the new hardware and a window will open showing "Add New Hardware Wizard" with information "..... USB 2.0 Device .....".



Click "Next" to continue.

 $\stackrel{<}{\sim}$  To do installation under Windows 2000/XP/Server 2003, user needs to change Log-in authority to "Administrator" or "Power-User" in order to install new software/hardware driver.

Select "Search for the best driver for your device"



Click "Next" to continue.

Select "CD-ROM drive" (Insert Driver Files CD into CD drive).

Add New Hardware Wiza	rd
	Windows will search for new drivers in its driver database on your hard drive, and in any of the following selected locations. Click Next to start the search.
<b>~</b>	EX
	В <u>т</u> ожае
	<back next=""> Cancel</back>

Click "Next" to continue.

PC will allocate files named "ALL100.INF" and "ALL100.SYS" for installation.



Click "Next" to continue.

Windows has finished the USB driver installation for ALL-100 Programmer.



Click "Finish" to complete USB driver installation.

#### 2.3 Software Installation

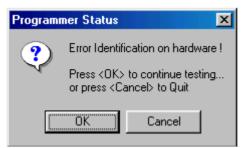
Insert Driver Files CD to CD-ROM drive, go to directory of ALL-100 under File Manager to execute the SETUP.EXE file, or run the SETUP.EXE from START menu of WINDOWS and follow all steps accordingly as follows:



#### Check of installed software:

(1) Check if software is installed properly with File Manager, see if XACCESS.EXE, individual IC programming driver, and Utility files exist under C:\Program Files\ALL-100 directory. XACCESS.EXE is a system file which provides an easy way to select IC Manufacturer, Product Type, and the corresponding programming driver. All the programming drivers can also be executed without running XACCESS.EXE file. Each programming driver usually supports a series of relevant ICs. For instance, driver file W28F.EXE is able to program 27C128 ~ 27C512 EPROMs.

(2) When executing programming driver, software will automatically check if ALL-100 Programmer is properly connected. If software can not recognize the existence of ALL-100, the connection and/or installation might have problem and ALL-100 Programmer might not be able to be accessed.



- (3) Methods to check if ALL-100 Programmer is properly connected/existed:
- 1. Run "USB Info." option under XACCESS menu.
- 2. Run programming driver file under XACCESS menu.

ALL-100 Universal Progr	ammer	_ 🗆 ×
<u>D</u> evice <u>T</u> ester <u>U</u> tility U <u>S</u> I	B Info. Help	
• •	) 💇 🌆 🖗 👗 S/W Update : http://www.hilosystems.co	om.tw
Fast Open 🔊 8748-series**87	741A**WMPU2.EXE	•
🖬 Utility	x	
-HEX to Bi	inary Converter	
PATH :	c:	
Input Hex	: File : 🕒 Browse	
Output Bi	in File :	
HEX Form	nat : Intel	
	© 00H © FFH	
Start Add	ress : 00000	
End Addr	ess : FFFFF Z Start Close	
	0%	

#### XACCESS.EXE Utility Dialog Box

ALL-100 Universal Programmer	
Device Tester Utility USB Info. Help	
SW Update : http://www.hilosystems.com.tw	
Fast Open 15 8748-series**8741A**WMPU2.EXE	
About	
Help	
ALL-100 Universal Programmer	
Device ACCESS File Version 1.00	
S/W update : http://www.hilosystems.com.tw	
Tech e-mail:support@hilosystems.com.tw	
Sales e-mail : sales@hilosystems.com.tw	
If you have any questions , please contact us.	
<u> </u>	

XACCESS.EXE Information Box

## 3. ALL-100 Basic Operation

#### 3.1 Getting Started

We will have a brief description of ALL-100 basic operation, introduce how to access the desired IC programming driver through XACCESS, the main system program. We will also introduce functions of Device, Load, Blank check, and Program by taking AMD AM29F200B as example.

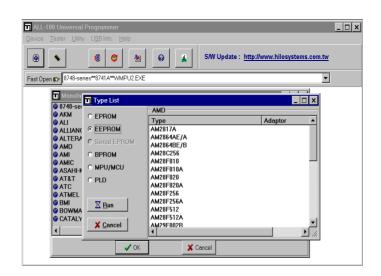
 $\stackrel{\wedge}{\sim}$  For best view of ALL-100 information displayed, user's screen should have resolution 800x600 pixels or above.

#### 3.1.1 Start with XACCESS

Click the icon of XACCESS to activate XACCESS.EXE and get following display on the window.

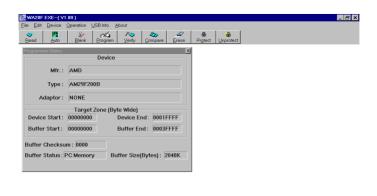
#### 3.1.2 Select IC Manufacturer

Click "Device" to display IC Manufacturer options. Select "AMD" and click "OK" to display Type List of the selected manufacturer.



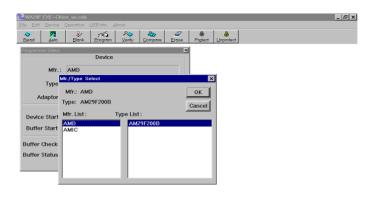
#### 3.1.3 Select IC Product Group and Product Type

Select "EEPROM" in Product Group at left side and select "AH29F200B" in Product Type at right side of Type List Dialogue Box and then click "Run".



Main Menu of the Driver File

Main Menu of the Driver File contains three major parts, the first row for menu of main fuctions, the second row for quick function keys, and the rest provides space for dialogue boxes of IC information like IC Manufacturer, Product Type, Adapter needed... etc. The driver file can be executed without going through XACCESS. In main menu of the driver file, user can reselect IC Manufacturer and Product Type. For example, execute driver file "A28F.EXE" directly and click "Device" to select IC Manufacturer and Product Type to be programmed. See dialogue boxes below.



#### Remark:

When reselecting IC manufacturer and Product Type, information in dialog box will be updated accordingly and the driver file will be downloaded to ALL-100 Programmer. If message "File not found" appears, it means the driver file is not available in PC, check the attached Driver File CD or visit HI-LO web site at http://www.hi-lo.com.tw for S/W download. If a specific ADAPTER is needed, "File not found" might be due to the absence of the required ADAPTER S/W. Try to install the S/W again.

### 3.2 Load file to Programmer buffer

It is ready for IC programming after selecting IC Manufacturer and Product Type. In general, programming code is saved in a file in Bin/Hex format. This code needs to be loaded to programmer buffer and then programmed into Blank IC devices. To load file to programmer buffer, click "File" menu, select "Load file to Programmer buffer" option, following dialogue box will appear:

WA28F.EXE-( V							_
le <u>E</u> dit <u>D</u> evice	Operation						
🎸 🎽 Bead Auto	Blank	erada Brogram	P⊘ & ⊻erify ⊆omp		Protect Unprote	a	
rogrammer Status				Þ	4		
		Device					
Mfr.:	AMD						
Type :	AM29F2	008					
Type.	rime of 2	000					
Adaptor :	NONE	Load File				? ×	1
	Targe	Look in: 🖂	Channels		💌 🗈 💆 🖻		
Device Start :	0000000	Name	Size T	ype	Modified	•	
Buffer Start :	0000000	Chlen_gy		obinet	5/11/98 8:01 PM		
		Chlen_ie		abinet abinet	5/11/98 8:01 PM		
Buffer Checksu	m · 0000	Chlen_jm		abinet	5/11/98 8:01 PM 5/11/98 8:01 PM		
buller offectad	. 0000	Chlen_nz	60KB 0	abinet	5/11/98 8:01 PM		
Buffer Status:	PC Memo	Chlen_ph		abinet	5/11/98 8:01 PM		
		Chlen_sr	34KB 0		5/11/98 8:01 PM		
		Chlen_tt	34KB 0		5/11/98 8:01 PM	-	
		File <u>n</u> ame:				Open	
		Files of type:	All Files(*.*)		•	Cancel	
			□ Open as <u>r</u> eav	d-only	_		

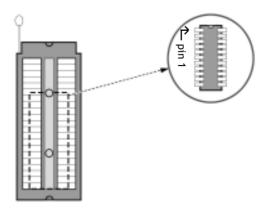
Function of file loading is similar to that under Windows environment. Enter the file name to be downloaded and click "Open", the named file will be loaded to programmer buffer. **Note: Disk drive and file path must be correct.** Select and click the correct drive and folder that the file located to. If it can not be operated by mouse, apply <TAB>, <UP>, <DOWN>, and <ENTER> keys for selecting and confirming.

### 3.3 Read contents from Master IC to buffer

When programming code is stored in a Master IC, insert the Master IC onto socket, click "Read" button on screen or press "R" key on keyboard to read programming code from Master IC to programmer buffer.

WA28F.EXE-Chil File Edit Device	en_au.cab Operation _USB InfoAbou	ut
🛷 р		Prove Anterna
Programmer Status	Device	×
Mfr.:	AMD	
Type :	AM29F200B	
Adaptor :	NONE	Read Device X Counter :
Device Start : Buffer Start :		Message :
Buffer Checksu Buffer Status:		E ID Chack
		If you use even or odd mode,the buffer will automatically expand to 2 times the device size .

When insert Master IC onto socket, make sure Pin 1 orientation and Pin count positioning is correct as diagram indicated.



 $\stackrel{\wedge}{\succsim}$  Caution! Incorrect IC positioning might cause IC damage or be programmed to an unknown state.

#### 3.4 Program buffer contents to IC

Insert IC to be programmed onto socket, click "Auto" button on screen or press "A" key on keyboard, following dialogue box will appear.

	eters atto Sectors/Blocks Scriet No. Par Mfr.: AMD Type: AM29F200B	smeter Serial No. : OFF		ckSum :	
Mfr Typ Adapto	Program Setting	Site # Status:	Со ОК 0	unter : NG 0	
Device Star Buffer Star Buffer Check Buffer Statu	For Frage P Blank Check Program Verity F Protect	Programming 132           2           3           4           5           6           7			
		Total :	0 Rese	0 t Count	

Click "Run" button on screen or "Y" key on keyboard or "YES" on programmer to start programming buffer contents to blank IC.

After programming, system will automatically verify data read from programmed IC with data in programmer buffer. If both data match then the "GOOD" LED will be lit to show a successful programming.

For next IC programming, the "BUSY" LED needs to be off and then insert blank IC onto socket, click "Run" button on screen or "Y" on keyboard or "YES" on programmer to continue programming.

Click "Close" button on screen or <ESC> key on keyboard to go back to main menu.

### 3.5 ADAPTERs and CONVERTERs

Adequate ADAPTERs and CONVERTERs are available to support various IC types and packages in market such as PLCC, SOP, TSOP, QFP, PGA, ... etc.

#### ADAPTER:

Each ADAPTER has 40/48 gold plated pins in DIP layout which can be inserted onto 48 pins ZIF socket on the Programming Module. Each ADAPTER has one (or more) programming file(s) that needs to be loaded to the same directory as XACCESS.

#### ■ CONVERTER:

CONVERTER is used to convert signals from DIP package to others like PLCC, SOP, TSOP, ...etc. No extra programming file is needed to work with CONVERTER.

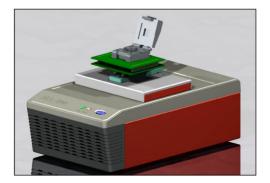
#### 3.5.1 ADAPTER and CONVERTER installation

#### S/W Installation:

Copy the ADAPTER file(s) attached to the same directory as that for XACCESS file(s).

#### H/W Installation:

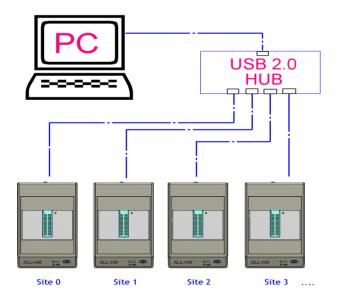
Insert the DIP-layout 40/48 pins of ADAPTER or CONVERTER onto the 48 PIN DIP ZIF socket on the Programming Module and lock it. See diagram below:



## 4. Multiple ALL-100 Installation

USB interface provides functions like "Plug-and-Play", auto detection, high expansibility, and high transmission rate (480 Mb/s for 2.0 version). By using these functions, ALL-100 can be installed up to 8 sets and running simultaneously.

After single ALL-100 installation, more ALL-100 Programmers can be installed through either extra USB ports of your PC or USB 2.0 Hub like following diagram.



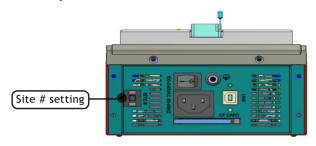
Multiple ALL-100 installation through USB 2.0 Hub

### 4.1 Multiple ALL-100 Operation

Multiple ALL-100 operation can increase programming throughput. With Multi-Thread methodology, each programmer can operate independently. The diagram below shows there are two ALL-100 Programmers (<u>Site #0</u> and <u>Site #1</u>) are running programming.

Edit Deve Auto     Edit Deve Auto     Sectors/Blocks Serrelftor     Gend Auto     Mfr.: AMD     Type: AM29F200B	Parameter	Serial No. : OFF			ckSum :	
Mir Jopp Adapto Device Star Buffer Star		atus: cogramming 13% cogramming 13%	Total:	0K 0	NG 0 0 0 0 0 0 0 0 0 0 0	

Site # setting switch is located on the rear panel of ALL-100 Programmer. Valid Site # for Multiple ALL-100 setup is 0~7 and can not have the same Site # in whole system.



When running ALL-100 programming software, system will automatically detect and load necessary driver S/W and also download programming code to ALL-100 through USB port. Operation of multiple ALL-100 is basically the same as that of single ALL-100. There are two operation modes for multiple ALL-100 programming, synchronous operation and asynchronous operation.

After entering programming mode on screen, user can choose either of following two operation modes depending on production needs:

#### Synchronous operation:

Run programming on all the ALL-100s at the same time. Put ICs onto socket of each ALL-100, press "Y" key on PC keyboard. All ALL-100s will start programming simultaneously.

#### Asynchronous operation:

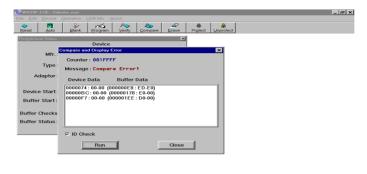
Run programming on each ALL-100 one after another. Put IC on the 1st ALL-100, press <YES> on that programmer to start programming, then put another IC on the 2nd ALL-100, press <YES> on that programmer to start programming, then the 3rd one ... and so on.

#### 4.2 Read/Compare on Multiple ALL-100s

To perform Read/Compare function under Multiple ALL-100 setup, only Site #0 is valid. This means that only IC on **Site #0** can be read/Compared under Multiple ALL-100 setup.

wa28F.EXE-Chi			_ 8 ×
Eile Edit Device			- 61 -
Bead Auto	Blank Brogram Verity Compare	Erase Protect Unpro	
Programmer Status	Device	×	
Mfr. :	AMD		
Type :	AM29F200B		
Adaptor :	NONE Read Device		×
-	Counter : Target Zone (Byte V		
Device Start :			
Buffer Start :	00000000 Buf	⊂ Even ⊂ Odd	
Buffer Checksu			
Buffer Status:	PC Memory Buffer P ID Chec	k .	
	Bun	Close	
	if you use ev will automati device size .	en or odd mode,the buffe cally expand to 2 times the	5

Read Dialogue Box



Compare Dialogue Box

100%

### 5. ALL-100 Gang Option Installation

For further speed up programming for volumes of ICs, 8 sockets Programming Module option is available. Contact your local HI-LO agent for ordering information.

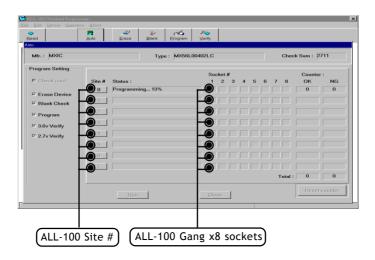
ALL-100 Gang Programmer is an ALL-100 Base Unit with 8 sockets Programming Module put on the top. User can choose either Single ALL-100 Gang operation or Multiple ALL-100 Gang operation for mass programming. The S/W attached with 8 sockets Programming Module is used for ALL-100 Gang as well as Multiple ALL-100 Gang operation. The installation /operation is similar as what is stated above.

 $\stackrel{\wedge}{\sim}$  Due to more information to be displayed on screen for Gang operation, user's screen resolution needs to be 1024x768 pixels (SVGA mode) or above.



# 5.1 ALL-100 Gang Operation

ALL-100 Gang Programmer ties IC pin count in parallel, drive programming and control signals through FPGA to program ICs on all sockets simultaneously. The diagram below shows there is one ALL-100 Gang Programmer (Site #0) is running programming.



Up to 8 sets of ALL-100 Gang Programmer can be connected to PC through USB interface, this is so call "Multiple ALL-100 Gang Setup". Each ALL-100 Gang can program 8 ICs. One PC with 8 sites of ALL-100 Gang setup can program 8x8= 64 ICs.

# 5.2 Read/Compare on Multiple ALL-100 Gangs

To perform Read/Compare function under Multiple ALL-100 Gang operation, only IC on MASTER (Socket #1) of Site #0 Gang programmer is valid. This means that only the 1st IC on MASTER (Socket #1) of the 1st site (Site #0) Gang Programmer can be read/compared under Multiple ALL-100 Gang setup.

Put IC to be read/compared onto **MASTER** (Socket #1) of **Site #0** Gang Programmer and click Read/Compare keys on screen.

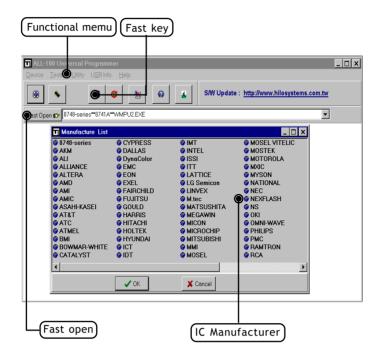
 $\stackrel{\scriptstyle \wedge}{\scriptstyle \sim}$  Refer paragraph 4.1 for setting of Site # for Gang Programmer.

🚰 ALL-100 Universal Programmer	×
File Edit Device Operation About	
-	
Programmer Status	
Device	
Mfr. : MXIC	
Type : MX56L00402LC	
Adaptor : AD8-UN-SD11 Memory Read	×
* Please insert MMC card on ma	ster (#1) socket.
Device Start : 00000000 E Counter : 09E800	
Buffer Start : 00000000 Message : Reading now	
Buller Start . 0000000	
Buffer Checksum : 0200 Burn	Close
Buffer Status : PC Memory Burler Size . 40501	

# 6. ALL-100 Software Description

## 6.1 XACCESS User Interface

System Software "XACCESS" is an interface guiding user to locate the driver software for product type to be programmed. XACCESS also provides File Management Utilities as well as Data Base of programmable IC products.



# 6.2 Utilities:

File Management Utilities includes Hex to Bin Converter, Bin to Hex Converter, 2-way splitter, 4-way splitter, 2-way shifler, and 4-way shuffler, ...etc.

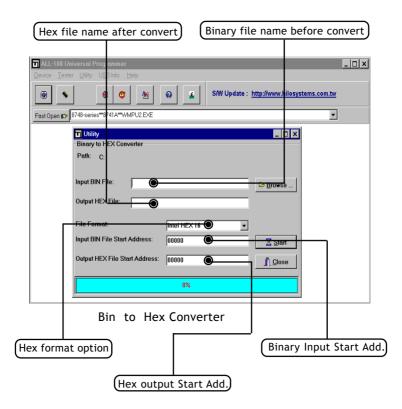
#### 6.2.1 Hex to Bin Converter

Convert data from Hex format to Binary format for programmer Read/ Write.

(Binary file name to convert to) (Hex file name to be converted)
ALL-100 Unive sal Programmer
Device Tester Mility USBInfo. Help
S/W Update : ttp://www.hilosystems.com.tw
Fast Open 😰 874 rseries**8741A**WMFU2.EXE
Ti Utility
HEX to Binary Converter
PATH: C:
nput Hex File :
Dutout Rin File :
HEX Format : Intel  Unused Byte O 00H
Start Address : 100000 •
End Address :
End Address :
0%
Hex to Bin Converter
Hex format option (Start/end address) (Contents of Unused Bytes

#### 6.2.2 Bin to Hex Converter

Convert data from Binary format to Hex format.



# 6.2.3 2-way splitter

Split one file into two output files. One file contains odd-byte data of the original file and the other file contains even-byte data of the original file.

Filenames after split Filename to be splitted	
ALL-100 Universal Programmer	×I
Device Tester Utility USB Info. Help	
S/W Update : http://www.hilosystems.com.tw	
Fast Open 😰 8748-series **8741.A***WMPU2.EXE	
- 2-way splitter	
Input File :	
Browse	
Output File :	
Even:	
Odd:	
Split Data Format	
Split in Byte Wide C Split in Word Wide C Split in Dword Wide	
<u>I</u> Start	
0%	
2-way splitter	
Split format options	

Split Data Format:

Normally the split data is in Byte Wide, however, user can choose Word Wide (two bytes) or Double Word Wide (four bytes) as unit of data split.

## 6.2.4 4-way splitter

Split one file into four output files. The 1st file contains the 1st byte of every 4-byte data segment of the original file. The 2nd, 3rd, and 4th file contains the 2nd, 3rd, and 4th byte of every 4-byte data segment of the original file.

Filenames after split	Filename to be splitted
ALL-100 Universal Programmer	
Device Tester Utility USBInfo. Help	
🖉 💊 🖬 Utility	hilosystems.com.tw
4-way splitter	
Fast Open D	
Output File :	
1st:	
2nd:	
3rd:	
4th:	
Split Data Format	de la Constate Descal Made
© Split in Byte Wide C Split in Word Wi	de C Split in Dword Wide
🔀 <u>S</u> tart	👖 Close
0%	
0%	
4-way	splitter
Split format options	

Split Data Format:

Normally the split data is in Byte Wide, however, user can choose Word Wide (two bytes) or Double Word Wide (four bytes) as unit of data split.

# 6.2.5 2-way shuffler

Combine two files into one. Insert data of Even file into even byte position of the combined file and insert data of Odd file into odd byte position of the combined file.

(Filenames to be shuffled) (Filename after shuffle)
ALL-100 Universal Programmer
Device Tester Utility USB Info. Help
Fast Open 12 8748-series**8741A **WMPU2.EXE
🖬 Utility 📃 🗖 🔀
2-way shuffler
Output File :
Input File :
Even:
Odd:
Shuffle Data Format
<u>I</u> Start <u>I</u> Close
0%
2-way shuffler
Shuffle format options

Shuffle Data Format:

Normally the shuffled data is in Byte Wide, however, user can choose Word Wide (two bytes) or Double Word Wide (four bytes) as unit of data shuffle.

## 6.2.6 4-way shuffler

Combine four files into one. Insert data of 1st file into the 1st byte of every 4-byte data segment of the combined file, insert data of 2nd, 3rd, and 4th file into the 2nd, 3rd, and 4th byte of every 4-byte data segment of the combined file.

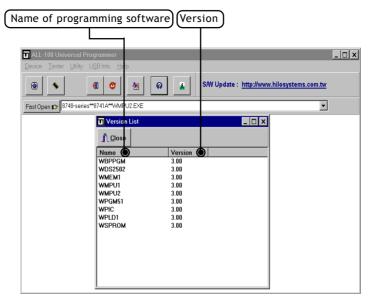
Filena	umes to be shuffled	Filename aft	er shuffle
	v USB Into Help		osystems.com.tw
	Start 0%		
Shuffle forma	4-way sh	ouffler	

Shuffle Data Format:

Normally the shuffled data is in Byte Wide, however, user can choose Word Wide (two bytes) or Double Word Wide (four bytes) as unit of data shuffle.

#### 6.2.7 Version List

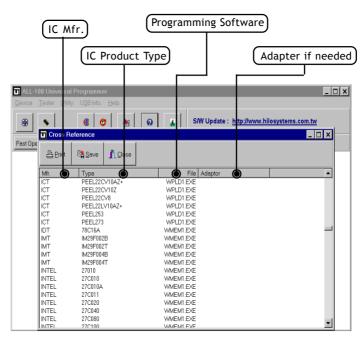
List version number of current programming software.



Version List

#### 6.2.8 Cross Reference

List the Cross Reference of IC Mfr., Product Type, and the corresponding programming software.



Cross Reference

\_

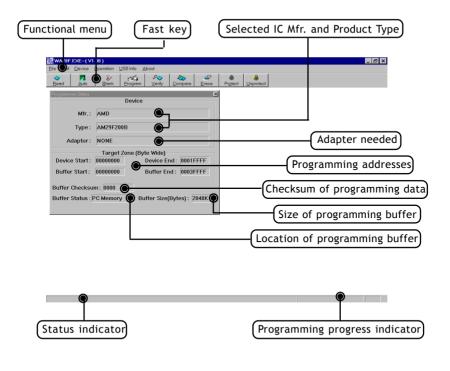
# 6.2.9 Device List

List all device types that are programmable at current version.

Print out the list Save t	the list
ALL 100 Universal Programmer       Device     Tester       O     VSB Info       Help	S/W Update : <u>http://www.hilosystems.com.tw</u>
Device List	
ATC24LC02W         ATC24LC04         ATC24LC04P           ATC24LC04W         ATC24LC08         ATC93C46           ATC93C56 (128*16)         ATC93C56 (128         ATC93C56 (128           ATC93C66 (256*16)         ATC93C56 (128         ATC93C56 (128           ATC93C66 (256*16)         ATC93LC56 (1         ATC93LC56 (1           ATC93LC56 (128*6)         ATC93LC56 (256*16)         ATC93LC56 (256*16)           ATC93LC56 (256*16)         ATC93LC66 (256*16)         ATC93LC66 (256*16)	*8) ATC93C66S
<pre>&lt;&lt; ATMEL &gt;&gt; ### EPROM Device ### AT278V010 AT27BV020 AT27BV040 AT278V512 AT27C010 AT27C0101 AT27C040 AT27C0104 AT27C1024 AT27C128 AT27C2048 AT27C128 AT27C1296 AT27C2512 AT27C512 AT27C4096 AT27C512 AT27C512 AT27C513R AT27C516 AT27C520 AT27HC256R AT27HC64 AT27HC64 AT27HC256R AT27HC64 AT27HC02A AT27LV0300 AT27LV030 AT27LV026 AT27LV056R AT27LV4096</pre>	AT27BV1024 AT27C020 AT27C1024L AT27C256R AT27C513 AT27C1024 AT27LV01024 AT27LV01024 AT27LV0124 AT27LV0124
Device	List

# 6.3 Programming Data/ Function Description

Programming software includes loading of programming driver, utilities of data management, and control of programming function/operation. Each programmable IC needs correct functional setup before programming. User needs to refer IC data sheet and/or application software for proper setup of programming function. See description below by taking AMD AM29F200B IC as example.



## 6.3.1 File

Load/Save of data file and configuration file. See dialogue boxes below.

_							
( F	unctio	onal m	enu)				
<b>%</b>	8F.EXE-( V1						_ 日 ×
_							
		Operation USB	Info. About				
	File to Program File from Prog		v <b>≦a P⊘ &amp;</b> Igram ⊻erity <u>C</u> omp		- Brotect <u>U</u>	lnprotect	
	Programmer C Programmer C		evice	×			
Enab	le Job Functior	ı					
Exit							
	Type :	AM29F200B					
	Adaptor :	NONE					
		Target Zo	ne (Byte Wide)				
Dev	/ice Start :	00000000	Device End: 0	001FFFF			
Buf	ifer Start :	00000000	Buffer End: 0	003FFFF			
Buffe	er Checksu	m : 0000					
		PC Memory	Buffer Size(Bytes	1: 2048K			
		,		,			

(1) Dialogue boxes of load data file:

🥵 WA28F.EXE( V1.0	0)				_ 🗗 X
<u>File Edit Device Op</u>					
Image: Second		P₀         №         ∅           ⊻erify         ⊆ompare         Erase	e Protect Unprotect		
Programmer Status	Device		×		
Mfr.:	AMD				
Type :	AM29F200B				
Adaptor :	NONE Load File			? ×	
Device Start: 0	Targe Look in: Same	Channels Size Type	Modified		
Buffer Start: 0	000000 Chipt_br	60KB Cabinet 48KB Cabinet	5/11/98 8:01 PM 5/11/98 8:01 PM		
Buffer Checksum	Chisv_se	60KB Cabinet 25KB Cabinet 46KB Cabinet	5/11/98 8:01 PM 5/11/98 8:01 PM 5/11/98 8:01 PM		
Buffer Status : PC	Chith_th	100KB Cabinet 63KB Cabinet 48KB Cabinet	5/11/98 8:01 PM 5/11/98 8:01 PM 5/11/98 8:01 PM	_	
	File name:	55KB_Cabinat	5/11/98.9-01 PM	 Qpen	
	Files of type:	All Files(*.*)		Cancel	
		All Files(**) Binary Files(*.bin) HEX Files	•	1.	
C	Data Chart				
l	Data file to	be loaded) (	File format	options	

KA28F.EXE-(V1	0.00) Operation US	Binfo, About				_ @ ×
Contraction Provide Auto	<b>≥</b>	xa 20 X	npare <u>E</u> rase	Rigtect Unprotect		
Programmer Status	-	Device	×	l		
Mfr.:	AMD					
Type :	AM29F200	В				
Adaptor :	NONE					
Device Start :		File Format			×	
Buffer Start :		File Formats : Binary	File St	atus :		
Buffer Checksu	m · 0000	Intel HEX MOTOROLAS Reco		tart: 0000	]	
Buffer Status:		- Unused By es :	File	end: 001FFFFF		
		• Don't Care 🌘	Buffs	tart: 00000000		
		000 C FF	0	K Cancel		
					_	
					(=	
Life form	at opt	ions) (Data	of Unus	ed Bytes	LFile addr	esses setting

\_\_\_\_\_

(2) Dialogue box of save data file:

e Edit Device Operation	USB Info. About					
Sead Auto Blank	Brogram Verity		Erase Protec	t <u>U</u> nprotect		
rogrammer Status			×			
	Device					
Mfr.: AMD						
Type: AM29F2	200B					
Adaptor : NONE	Save File				? ×	
Targe Device Start : 0000000		nels	<b>▼</b> [	9 🛛 🗗 🖾 🖿		
	Ch199	Chlen_bb	Chlen_ph	Chies_co	ac ac	
Buffer Start : 0000000	Childa_dk	Chlen_ca	Chien_ti	Chies_do Chies_do	S ⊂	
Buffer Checksum : 3EB4	Childe_ch	Chlen_gy	Chlen_us	Chles_gt	<u>କ୍</u> କୁତ୍ର କ୍ରୁତ୍ର	
Buffer Status : PC Memo	Childe de O	Schlen_jm Schlen_me	🖄 Chlen_za 🆄 Chles_ar	Chies_hn Chies_mx	୍ରର ଭୁଦ	
	Chlen_au	Chlen_nz	Chles_cl	🖄 Chles_ni	(a) C	
	File name: test	1		Sec	ve	
		Files(*.*)		Can	cel	
		ary Files(*.bin)	Y		6	
					<b>`</b>	

(3) Dialogue boxes of load/save Programming Configuration file: The Programming Configuration file contains all programming related data setup by user. It includes programming driver, data file, programming functional setting, and serial number setting. User can save Programming Configuration file for load it back next time for programming without re-setting of programming function and serial number.

WA28F.EXE-Chi	len_au.cab									_ & ×
le <u>E</u> dit <u>D</u> evice	Operation	USB Info. Abi	out							
Sead Auto	Blank	Program	P         200           ⊻erify         ⊆ompan	e Erase	8 Pr <u>a</u> tect	<mark>.</mark> ⊎nprotect				
Programmer Status		Device		×	1					
Mfr.:	AMD									
Type :	AM29F2	00B								
Adaptor :	NONE		nmer Configuratio					-		
Device Start : Buffer Start : Buffer Checksu Buffer Status :	Targe 0000000 0000000 um : 3EB4			P	• €	<b>3 6</b>	?	-		
		File <u>n</u> ame:	test	_			<u>O</u> pen			
		Files of type:	Macro File (*.cfg)			•	Cancel	1		
								777		
	Prog	gramm	ing Conf	igura	tion f	ile to	o be lo	aded	)	

WA28F.EXE-Chi File Edit Device	on_au cab Operation_USB Info_About	×
ead Auto	V         A         A         B	
Programmer Status	Device	
Mfr.:	AMD	
Type :	AM29F200B	
Adaptor :	NONE Save Programmer Configuration ? X	
Device Start : Buffer Start : Buffer Checksu Buffer Status :	Ψ	
	(Programming Configuration file to be saved)	

=

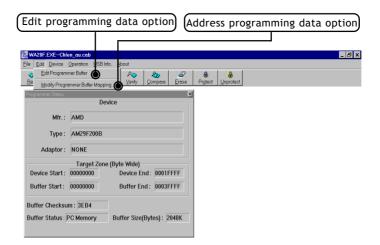
(4) Dialogue box of enable Job Function:

Once user enable Job Function and followed by loading Programming Configuration file, system will enter <AUTO> programming mode and is ready for programming. User needs not re-do the programming functional setting and serial number setting.

WA28F.EXE-Chlen_au.cab			_ & ×
Eile Edit Device Operation USB	Info. About		
	gram ⊻erify <u>C</u> ompare <u>E</u> rase	Image: Second system         Image: Se	
Load Programmer Configuration Save Programmer Configuration	zvice	l	
Enable Job Function			
Exit			
Type: AM29F200B			
Adaptor : NONE			
Target Zor	ne (Byte Wide)		
Device Start: 00000000	Device End: 0001FFFF		
Buffer Start: 00000000	Buffer End: 0003FFFF		
Buffer Checksum : 3EE4			
Buffer Status: PC Memory	Buffer Size(Bytes): 2048K		
		-	
(Enable Job	Function		
Enable Job	runction		

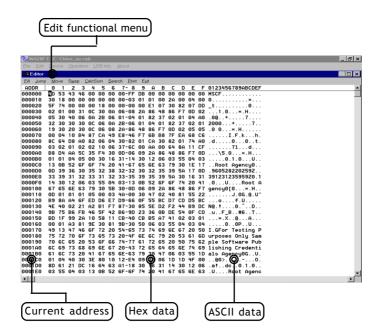
#### 6.3.2 Edit

Includes edit and address of programming data. See dialogue box below.



(1) Edit data:

Provide functions such as Fill, Jump, Swap, Calc, SUM, Search,... etc for user to edit programming data in Hex and ASC II expressions.



(2) Fill:

Specify start and end addresses and data to be filled in, then click "OK", data will be filled into the specified area.

🖕 WA28F																		_ & ×
<u>E</u> lle <u>E</u> dit		rice																
🗱 Editor																		- 🗆 ×
Fill Jump	M	ove	Swe	n a	CalcS	um	Sea	rch <u>P</u> rin	E E	cit								
ADDR	Θ	1	2	3	4	5	6	7-8	9		в	С	D	E	F	0123456789ABCDEF		<b>^</b>
000000		53														MSCF		
000010		18														0		
000020	5F	74	00													t0		
000030	02	01	00	31	0C	30	0A	06-08	2A	86	48	86	F7	ΘD	02	1.0×.H		
000040	05	30	40	06	0A	2B	06	01-04	01	82	37	02	01	04	A0	.00+7		
000050	32	30	30	30	0C	06	0A	2B-06	01	04	01	82	37	02	01	2000+7		
000060	19	30	20	30	0C	06	08	2A-86	48	86	F7	ΘD	02			.0 0×.H		
000070				Β4	87			FA 114		-	-	-				I.F.kh.		
080080		64											×	74	Ĥ0	.d00t.		100
000090		02				STA	er -	00000000	-1	6	OF		i i		CF	d		100
0000A0		D4			- 2	0170		00000000	_		01	·		F7	ΘD	\5.0×.H		
0000B0		01				F	D:	0003FFF	F I				d İ	04		0.1.0U		
000000		0B					· · · ·	00001111			Can	cel		1E		Root Agency0		
0000D0					3	DA	TA:	FE		_			- I	17		.960528220259Z		
0000E0					3	0.4		<u></u>						16		39123123595920.1		
0000F0		30											_			.0URoot A		
000100		65														gency0[0×.H		
000110					6						VE					J.0G.@.U"		
000120								D9-66								of.U		
000130 000140	4E 9B	40														N@. t0.^D .uF B#6T		
000140	BD 8B							CB-40								.uF_B#6I ×.X0A		
000160	00	01						9B-30								×.x@н		
000170																I.GFor Testing P		100
000180																urposes Only Sam		
000190																ple Software Pub		100
000100																lishing Credenti		
000180																als Agencu0GU.		
000100	01																	
000100																.atdc0.1.0		
0001E0																.URoot Agenc		-
<b>_</b>																······································		. Č
•																		

(3) Jump:

Specify the start address to jump to, then click "OK", system will jump to the specified address with data displayed.

28 WA28																	_ & ×
		rvice															
Edito	)r																_ 🗆 ×
Eill Jur	mo N	love.	Sur	in I	Calcs	um	Sec	urch <u>P</u> rin	t E	it.	_	_	_	_	_		
ADDR	6		2	3		5	6	7-8		A	в	С	D	Е	F	0123456789ABCDEF	•
000000		53			00											MSCF	
000010	30	18														0×	
000020	5F	74						00-00									
000030		01	00			30										1.0×.H	
000040	05	30	40	06	0Â	2B	06	01-04	01	82	37	02	01	04	A0	.00+7	
000050																2000+7	
000060																.0 0×.H	
000070		04				-				-	-	- 10			1C6	I.F.kh.	
000080		64				-MP	AUL	RESS						×	A0	.d00t.	
000090		02									1000				CF		
0000A0		D4				STA	BT:	0000000	0	-1		OK			0D	\5.0×.H	
0000B0		01						<u> </u>	_		_				03	0.1.0U	
000000		0B										Cance	et 📗		17	Root Agency0	
000000		39									-	_				.9605282202592	
0000E0		39				•••				**		•	**		31		
0000F0								03-13								.0URoot A	
000100																gency0[0×.H	
000110		01														J.0G.0.U"	
000120								D9-66									
000130																N@. †0.^D .uF_B#6T	
000140								CB-40								×.X0A	
000150		01						9B-30									
000100																I.GFor Testing P	
000180																urposes Only Sam	
000190																ple Software Pub	
000100																lishing Credenti	
000180																als Agency0GU.	
000100																	
000100		61														.atdc0.1.0	
0001E0	03	55														.URoot Agenc	
•																	

(4) Move:

Specify start and end addresses in which data to be moved, also specify the destination address to move to, then click "OK", system will move data between spcified addresses to the destination area.

Έ <b>γ</b> ν,																		_ & ×
Eile	Edit	Dev	vice					1fo.	About									
	ditor																	_ 🗆 ×
	Jump	Mr	ove	Sur	in i	CalcS	Sum	Sec	urch Erin	1 E)	at.	_	_	_	_	_		
ADD		Θ	1	2	3	4	5	6	7-8	9	A	в	С	D	Е	F	0123456789ABCDEF	
0000		4D	53	43	46	00			00-FF	DB				00			MSCF	
0000	10	30	18	00	00	00	00	00	00-03	01	01	00	2A	00	04	00	0×	100
0000	20	5F	74	00	00	00	18	00	00-00	00	E1	07	30	82	07	DD	_t0	
0000	30	02	01	00	31	0C	30	0A	06-08	2A	86	48	86	F7	ΘD	02	1.0×.H	
0000	40	05	30	40	06	0A	2B	06	01-04	01	82	37	02	01	04	A0	.00+7	
0000			30														2000+7	
0000								08	2A-86	48	86	F7	ΘD	02	05		.0 0×.H	
0000					В4	Î	OV/E		50 00		-	-	-		×Î.	C6	I.F.kh.	
0000			64				OVE								ᅳᆘ	A0	.d00t.	
0000			02					- T	00000000	-	Г	-	)K	1	1	CF		
0000			D4				SIA	RE	100000000	_	_ <u>L</u>		л.,		7	ΘD	\5.0×.H	
0000		01			05		E	ND:	0003FFF	F		<u></u>	ncel	1	t t	03	0.1.0U	
0000			0B						-	=1	1	Ca	icei		E	17	Root Agency0	
0000			39				DE	ST:	123400						Ľ	θD	.9605282202592	
0000			39												Ē	31	39123123595920.1	
0000			30												- 2	41	.0URoot A	
0001			65												F	F7	gency0[0×.H	
0001			01 8A						4A-00								J.0G.@.U"	
0001			8H 40						D9-66								of.U N@.t0.^D	
0001																	.uF_B#6T	
0001									CB-40								×.X@A	
0001									9B-30								×.x@H	
0001																	I.GFor Testing P	
0001																	urposes Only Sam	
0001																	ple Software Pub	
0001																	lishing Credenti	
0001		61															als Agency0GU.	
0001																		
0001																	.atdc0.1.0	
0001	Eθ	03															.URoot Agenc	-
																	-	
<u> </u>																		<u>_</u>

(5) Swap - Byte:

Specify start and end addresses in which data to be byte-swaped, then click "OK", system will swap byte for data between specified addresses.

🛵 WA28F	.EXE	-Chl	len_i	au.c	ab											_ 5	P ×
<u>Eile E</u> dit	Dev	lice	Qpe	oratio	n <u>L</u>	ISB I	ıfo.	About									_
Editor																_ 0	I × I
Eill Jump	ı Mi	ive	Sur	in i	CalcS	Sum	Sec	urch Erin	E	at.	_	_	_	_	_		
ADDR	0	1	2	3	4	5	6	7-8	9	Ĥ	в	С	D	Е	E	0123456789ABCDEF	
000000								00-FF								MSCF	
000010		18		00												9 0×	_
000020																) t	1000
000030		01														21.0×.H	1000
000040		30														9.00+7	1000
000050	32	30														2000+	1000
000060	19	30	20			86	08	20-86	48	86	F7	ΩD	62	85	85	5 0 0 × H	1000
000070	00	04			07	~~		FO 110		20	-	-	= ^	68	C6	5I.F.kh.	1005
000080		64			- B	yte S	iwap	1					×			0.d0t.	100
000090		02											-	11	CF		100
0000A0	B8	D4	AA.	50	4	STA	RT :	00000000		1		OK		F7	ΘD	D\5.0×.H	100
000080	01	01	04	05	e									04	03	30.1.0U	100
000000	13	0B	52	6F	e	EN	D:	0003FFF		1	0	ancel		1E	17	7Root Agency0	100
000000	ΘD	39	36	30	3	-		JUUUSPEP				ancei		17	θD	.9605282202592	
0000E0	33	39	31	32	3									16	31	391231235959Z0.1	100
0000F0	14	30	12	06	e									20		I .0URoot A	100
000100	67	65	6E	63										86	F7	7 gency0[0×.H	100
000110		01		01				4A-00								2J.OG.@.U"	100
000120								D9-66									
000130																NQ. t0.^D	100
000140																).uF_B#6T	100
000150	BD	1F						CB-40									100
000160		01						9B-30									100
000170																0 I.GFor Testing P	100
000180																) urposes Only Sam	
000190																2 ple Software Pub	
0001A0																ð lishing Credenti	100
000180																) als AgencyOGU.	100
000100																9@0>0.	100
0001D0		61														5 .atdc0.1.0	100
0001E0	03	55	04	03	13	ΘB	52	6F-6F	74	20	41	67	65	6E	63	3 .URoot Agenc	-
•																<u>&gt;</u>	•

(6) Swap - Nibble:

Specify start and end addresses in which data to be nibble-swaped, then click "OK", system wil swap Nibble for data between specified addresses.

wa28F.	EXE	-Chl	len i	au.ce	nb															_ & ×
Elle Edit	Dev	ice	Ope	eratio	n L	SBI	nto.	About												
Editor								_									_	_	_	- 🗆 ×
Eill Jump	M	ive	Swe	n í	Dates	um.	Sec	urch <u>P</u> rin	E)	cit.	-	-	-	-	-		_	_	_	
ADDR	Θ	1	2	3		5	6	7-8	9	A	в	С	D	Е	E	0123456789ABCDEF				-
000000	40		43		00	00	00									MSCF				
000010	30															0×				
000020								00-00												100
000030	02		00																	1000
000040	05		40		00											.00+7				1000
000050	32	30	30	30	00											2000+7				1000
000060	19	30	20	30												.0 0×.H				100
000070	00	04	10								50	-	= ^	68	C6	I.F.kh.				1000
000080			D8		N	ibble	: Sw	ар					×			.d00t.				1000
000090			01																	100
0000A0	B8	D4	ÂĤ	5C	3	STA	RT :	00000000	1			OK		F7	0D	\5.0×.H				100
000080	01	01	04	05	e						here			04	03	U.1.0.1.0U				1000
000000	13	0B	52	6F	e	EN	JD :	0003FFF		1	~	encel	1	1E	17	Root Agency0				100
000000	ΘD	39	36	30	3			JUUUSPPP	-		- 4	blice		17	ΘD	.960528220259Z				1000
0000E0	33	39	31	32	3									16	31	39123123595920.1				1000
0000F0	14		12		6									20	41	.0URoot A				1000
000100	67	65	6E	63												gency0[0×.H				1000
000110	ΘD	01		01												J.0G.@.U"				100
000120	В9	8A	A4					D9-66								of.U				1000
000130	4E															N@. t0.^D				1000
000140	9B															.uF_B#6T				100
000150	BD	1F	99					CB-40								×.X@A				1000
000160	00		A3					9B-30								00PU				1000
000170	49															I.GFor Testing P				1000
000180	75															urposes Only Sam				1000
000190	70															ple Software Pub				1000
0001A0																lishing Credenti				1000
0001B0	61															als Agency0GU.				1000
000100	01															@0>0.				1000
0001D0	8D		21													.atdc0.1.0				1000
0001E0	03	55	04	03	13	ΘB	52	6F-6F	74	20	41	67	65	6E	63	.URoot Agenc				-
•																				•

(7) CalcSum:

Specify start and end addresses in which data to be calculated for checksum and then click "OK", system will calculate and display result (checksum) on screen.

<b>is,</b> wa																		_ & ×
<u>E</u> ile <u>B</u>			ice					1fo.										
👬 Edi	itor																	_ 🗆 ×
Eill J	lump	M	ive	Swe	0 a	CalcS	Sum	Ser	urch <u>P</u> rim	E E	dt							
ADDF		Θ	1	2	3	4	5	6	7-8	9	A	в	с	D	Е	F	0123456789ABCDEF	
00000			35			00			00-FF						00	00	.54d	
00001	0	03	81	00	00	00	00	00	00-30	10	10	00	A2	00	40	00		
00002	20	F5	47	00	00	00	81	00	00-00	00	1E	70	03	28	70	DD	.Gp.(p.	
00003	30	20	10	00	13	CO	03	A0	60-80	A2	68	84	68	7F	DO	20	h.h	
00004		50	03	04	60	A0	B2	60	10-40	10	28	73	20	10	40	0A	P``.@.(s@.	
00005		23	03	03	03	CO	60	A0	B2-60	10	40					10	#`.@.(s .	
00006		91							A2-68					20		50	`h.h PP	
00007			40			7	-	Sur	AF		-	20					.@.Kxd1	
00008			46			4	nerces	Sun	•					×	47	0A	.F(`@.((.G.	
00009			20								a 1				11	FC	0`sF	
00006			4D				STA	RT :	00000000				οк		7F	DO	.MSO`h.h	
00005			10								1			_	40	30	@Pa.A.!`OU@O	
00000			BO				E١	D:	0003FFF	F	1	0	ancel		E1	71	1.%G∪U.6q	
00000			93									_	_		71	DO	c.S#.##.#Sq.	
0000E			93												61		3#3.#3S.Sa.	
0000F			03												02		A.!`OU@01.%G	
00010			56			٤									68	7F	vV.6h.h.	
00011			10						A4-00							22	P.0tU"	
00012							6D		9D-66							CB	Jm~.f.U.).].	
00013		E4	04		12		1B	7F	78-03								××.X/D	
00014		B9	57	6B					6B-D9								.Wk.d.\$k.2cE	
00015			F1						BC-04							10	[z. 0.	
00016			10						B9-03								OU@0@	
00018		94	31						45-56								.1td.'.EV7Gv	
00018			27						02-F4							D6	W'7U75	
00016		07				35			47-77 02-34							26 96		
0001F		C6 16	96						62-34 E6-36								7U.4'UFU.G. 7UU.6t`0U.	
00010		10	40						21-4E							00		
00010									1A-81							60	aF6a.A.!	
0001E									F6-F6								0U001.%GUU.6	
		30	22	70	30	31	80	23	F0-F0	-1	02	14	10	36	E.0	30	00201.2.1400.8	
•																		•

#### (8) Search:

Two options available, search ASC II data or search Binary data.

wa28F.																_ & ×
<u>E</u> ile <u>E</u> dit	Dev	ice														
Editor																
<u>Fill</u> Jump	Mo	we	Swe	φ (	CalcS	um	Sec	arch Brin	t Eg	și.						
ADDR	Θ	1	2	3	4	5	A	ടവ (	9	A	в	С	D	Е	F	0123456789ABCDEF
000000	D4	35	34	64	00	00	B	inary	BD	00	00	00	00			
000010	03	81						00-30								
000020								00-00							DD	.Gp.(p.
000030								60-80							20	
000040								10-40								
000050																#
000060								A2-68								h.h PP
000070								8E-64								
080080								03-28								
000090								73-C6								
000000								D0-60							DO	
000080								13-41								
000000																1.%GU.6q
000000																c.S#.##.#Sq.
0000E0 0000F0																3#3.#3S.Sa. A.!`0U@01.%G
000100																UU.6
000110																P.0tU"
000120																
000120								78-03								
000140																.Wk.d.\$k.2cE
000150	DB							BC-04								
000160								B9-03								
000170		31														.1td
000180																ω'7075
000190								47-77								
0001A0								02-34								
0001B0								E6-36								
000100	10	40	04	03	E3	08	01	21-4E	90	D2	60	D1	D1	F4	00	.etN`
0001D0								1A-81							60	
0001E0	30	55	40	30	31	вθ	25	F6-F6	47	02	14	76	56	E6	36	0U@01.%GUU.6
•																•

Specify start/end addresses and data to be searched, then click "OK", system will search data between specified addresses and list all addresses that match data to be searched.

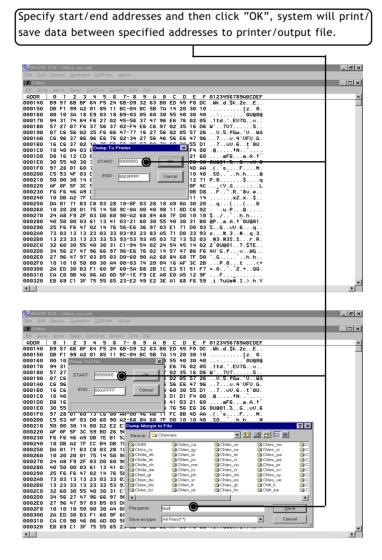
₩A28F.EXE-Chlen_au.cab	- 8 ×
Elle Edit Device Operation USB Info. About	
왕 <mark>문</mark> Editor	_ 🗆 🗙
<u>Fill Jump Move Swap CalcSum Search Print Exit</u>	
ADDR 0 1 2 3 4 5 6 7-8 9 A B C D	E F 0123456789ABCDEF
000000 D4 35 34 64 00 00 00 00-FF BD 00 00 00 00 000010 03 81 00 00 00 00 00 00-30 10 10 00 A2 00	
	40 000. 70 DD .Gp.(p.
000030 20 10 00 13 C0 03 A0 60-80 A2 68 84 68 7F	D0 20h.h
000040 50 03 04 60 A0 B2 60 10-40 10 28 73 20 10 000050 23 03 03 03 C0 60 A0 B2-60 10 40 10 28 73	40 0A P., ., .0.(S.0).
000060 91 03 02 03 C0 60 80 A2-68 84 68 7F D0 20	20 10 #`
000070 00 40 01 4B Terror Accurate	86 6C .@.Kxd1
000000 00 40 00 04 2	47 0A .F(`@.((.G.
0000000 30 20 10 20 2 START: 00000000 OK	11 FC 0`sF 7F D0 .MSO`h.h
000080 10 10 40 50 6	40 30@Pa.A.!`OU@0
0000C0 31 B0 25 F6 F	E1 71 1.%GvU.6q
000000 D0 93 63 03 5 ASCII 0000E0 33 93 13 23 5 Data: EV7G	71 D0c.S#.##.#Sq. 61 13 3#3.#35.Sa.
0000E0 41 03 21 60 3	02 14 A. ! OU@01.%G.
000100 76 56 E6 36 5	68 7F vU.6h.h.
000110 D0 10 10 10 5	55 <u>22</u>
000120 9B A8 4A F6 E V	5D CBJm <sup>~</sup> .f.U.}.]. 9B CD×x.X.=/D
000140 B9 57 68 BE 6	FADC Wild \$k 2c F
000150 DB F1 99 A2 ( 000160 00 10 3A 18 E 000170 94 31 74 64 F6 27 02 45-56 37 47 96 E6 76	30 10[z. 0.
000160 00 10 3A 18 E 000170 94 31 74 64 F6 27 02 45-56 37 47 96 E6 76	30 40
000180 57 27 07 F6 37 56 37 02-F4 E6 C6 97 02 35	16 D6 W'7U75
000190 07 C6 56 02 35 F6 66 47-77 16 27 56 02 05	
0001A0 C6 96 37 86 96 E6 76 02-34 27 56 46 56 E6 0001B0 16 C6 37 02 14 76 56 E6-36 97 03 74 60 30	47 9670.4'UFU.G. 55 D170U.6t`0U.
0001C0 10 40 04 03 E3 08 01 21-4E 90 D2 60 D1 D1	F4 00 .0!N
0001D0 D8 16 12 CD 61 46 36 1A-81 03 61 13 41 03	21 60aF6a.A.t`
0001E0 30 55 40 30 31 B0 25 F6-F6 47 02 14 76 56	E6 36 0U@01.%GvU.6
WA28F.EXE-Chien_au.cab	_ (5) ×
WA28F.EXE-Chien_ou.cob Elle Edit Device Operation USB Info. About	X
Elle Edit Device Operation USB Into. About	_ = = ×
Elle Edit Device Operation USBInto. About Editor Elit Jump Move Swap CalcSum Search Print Exit	X
Bit         Edit         Device         Operation         USB info.         About           Bit         Sump         Move         Skap         CelcSum         Search         Dirth         Edit           ADDR         6         1         2         3         4         5         6         7         8         9         A         B         C         D	E F 0123456789ABCDEF
Bit         Edit         Device         Operation         LISB Mo         About           Bit         Jump         More         Samp         CalcSim         Samch         End         Line           ADDR         0         1         2         3         4         5         6         7         8         9         A         B         C         D           0901140         BS         55         76         B         6         17         24         55         76         50         93         26         80         D         45	E F 0123456789ABCDEF F0 DC . Wk.d. \$k.2cE.
End         End         Conversion         LEB Mode         About           EN         Jamo         Mode         Samo         Samo         Samo           EN         Jamo         Mode         Samo         Samo         Samo         Samo           ADDR         0         1         2         3         4         5         6         7         8         9         A         0         0         0         0         0         1         2         3         4         5         6         7         8         9         A         0 <td>E F 0123456789ABCDEF F0 DCk. d \$k. 2c. E 30 10</td>	E F 0123456789ABCDEF F0 DCk. d \$k. 2c. E 30 10
Print         Dist         Dist <thdist< th="">         Dist         Dist         <t< td=""><td>E F 0123456789ABCDEFF A F0 DC ukk d. Sk. zc. z 30 10</td></t<></thdist<>	E F 0123456789ABCDEFF A F0 DC ukk d. Sk. zc. z 30 10
End         End         Convoc         Operation         LISB Mo         About           End         Line         Move         Samp         CalcSun         Samp         Edit           ADDR         0         1         2         3         4         5         6         7         8         9         A         B         C         D           000146         05         75         68         F         7         8         9         A         B         C         D         000150         DB         F1         94         7         24         61-03         2         63         DE         45         000150         DB         F1         94         20         18         51-03         55         60         54         10         95         18         55         60         55         40         55         40         55         40         56         37         47         62         76         24         55         37         47         62         76         37         56         37         47         62         76         37         56         37         47         62         75         37         47	E F 0123456789ABCDEF F F0 DC .Wk.d.Sk.2c.E 30 10
Prof.         Conc.         Conc. <th< td=""><td>E F 0123456789ABCDEF F0 DCK. d. Sk. 2c. E 30 10</td></th<>	E F 0123456789ABCDEF F0 DCK. d. Sk. 2c. E 30 10
Prof. Come         Common         Common <thcommon< th=""> <thcommon< th=""> <thcommon< td=""><td>E F 0123456789ABCDEF A F0 DC Mk. d. Sk. 2c. E. F. 30 10</td></thcommon<></thcommon<></thcommon<>	E F 0123456789ABCDEF A F0 DC Mk. d. Sk. 2c. E. F. 30 10
Image         Description         Description         Description         Description           ADDM         0         1         2         3         4         5         6         7         8         9         A         B         C         D           ADDM         0         1         2         3         4         5         6         7         8         9         A         B         C         D           000140         B5         75         8         F         4         F         24         6         D         45         D	E F 012345678998CDEF A F0 DC Mk. d. Sk. 2c. E 30 10
Image         Control         Control <thcontrol< th=""> <thcontrol< th=""> <thcon< td=""><td>E F 0123456789NBCDEF A F0 DC .4k. d. 5k. 2c. E 30 10</td></thcon<></thcontrol<></thcontrol<>	E F 0123456789NBCDEF A F0 DC .4k. d. 5k. 2c. E 30 10
Pint         Close         Openeon         Openeon         Description           # Gittori         Data         45         Status         Balance           ADDR         0         1         2         4         5         6         B         C         D           ADDR         0         1         2         4         5         6         F         8         B         C         D           000140         B         9         57         6B         B         C         D         0         1         S         D         B         C         D         0         0         1         S         D         B         C         D         0         D	E F 0123456789ABCDEF A F0 DCK.d. Sk. 2c. E 30 10
Image         Construct         Co	E F 0123456783986CDEF A F0 DC Mk. d. 5k. 2c. E 30 10
Pint         Close         Openeon         Openeon         Description           # Gittori         Data         45         Status         Balance           ADDR         0         1         2         4         5         6         B         C         D           ADDR         0         1         2         4         5         6         F         8         B         C         D           000140         B         9         57         6B         B         C         D         0         1         S         D         B         C         D         0         0         1         S         D         B         C         D         0         D	E F 0123456783986CDEF A F0 DC Mk. d. 5k. 2c. E 30 10
Prof.         Conc.         Conc. <th< td=""><td>E F 01234567839ABCDEF A F0 DC Mk. d. 5k. 2c. E 30 10</td></th<>	E F 01234567839ABCDEF A F0 DC Mk. d. 5k. 2c. E 30 10
Image: Disc.         Concord	E F 012345678980ECDEF A F0 DC 4k. d. Sk. 2c. E 30 10
Diff         Diff         Diff         Diff           # 2000         Barry         More         Dearly         Diff         Barry           ADDR         0         1         2         3         4         5         6         7         8         9         A         B         D           OP0140         99         5         7         68         BF         6         7         8         4         D         0         1         2         4         6         7         8         B         C         D         0         0         1         2         4         6         7         8         B         C         D         0         1         8         1         8         1         8         1         8         1         8         1         8         1         8         1         8         1         8         1         8         1         8         1         8         1         8         1         8         1         1         8         1         8         1         1         8         1         1         1         1         1         1         1         1	E F 01234567839ABCDEF A F0 DC 44k.d.5k.2c. E. 30 10
Diff         Diff         Diff         Diff           # 2003         Bits	E F 0123456789ABCDEF F0 DC 44k.d. 5k.2c. E 30 10
Diff         Diff         Diff         Diff           # 2000         Barry         More         Dearly         Diff         Barry           ADDR         0         1         2         3         4         5         6         7         8         A           ADDR         0         1         2         4         5         7         8         B         C         0           000116         95         7         68         BF         6         7         8         B         C         0           000116         95         7         68         BF         6         7         8         B         5         1         8         C         0         1         1         1         8         1         8         1         8         1         8         1         8         1         8         1         1         1         8         1         8         1         8         1         1         1         8         1         8         1         1         1         1         2         1         1         1         2         1         1         8         1         1	E F 01234567839ABCDEF A F0 DC .Wk.d. 5k. 2c. E 30 10
Image: Construction         Constr	E F 0123456789ABCDEF F0 DC 44k.d. 5k.2c. E 30 10
Image: State	E F 012345678908CCEFF A F0 DC 44K.d 5K.2c.E. 30 10
Diff         Diff         Diff         Diff           # 2003         Bins         Max         Diff         Bins         B	E F 01234567839ABCDEF F0 DC 44k. d, Sk. 2c. E 30 10
Image: Solution         Construction         Construct	E F 01234567839ABCDEF A F0 DC .Wk. d. 5k. 2c. E 30 10
Image         Dots         Description           0001         00001         00000000000         00000000	E F 01234567839ABCDEF F0 DC Mk. d. 5k. 2c. E 30 10
Image: Construction         Construction         Construction         Construction           Image: Construction         Construction         Construction         Construction         Construction           Image: Construction         Construction         Construction         Construction         Construction         Construction         Construction           Image: Construction	E F 01234567890BCDEF F0 DC ukk d. 5k. 2c. E 30 10
Image         Dots         Description           0001         00001         00000000000         00000000	E         F         0123456783986CDEF         A           F0         DC         Mk. d. Sk. 2c. E         30         30         10

= 59 =

# (9) Print:

Two options available, print to printer or ouput to file.

//	005 5	-	01.5													_	
<b>S</b> , WA	_	_	_	_	_	_				-	-	-	-		-	-	_ <u>-</u> ×
<u>File</u>		DRA	108	Obe	ratio	n <u>L</u>	SD II	110.	About								
t <mark>i R</mark> Edi																	
Eill 🛓		Mo	rve		φ (	Calcs	um		rch Prin	E Es	jit –						
ADDF		Θ	1	2	3	4	5	6		To P			С		E		0123456789ABCDEF
00014						64				To Fi				45			
00015									BC-04								
00016									B9-03								
00017									5-56								
00018									02-F4								
00019									47-77								
0001F									02-34 E6-36								
00010									21-4E								
00010									1A-81								
0001E																	0U001.%GVU.6
0001F																	
00020																	.S0h.h@
00021																	P.0\$a
00022	20	AF	0F	5F	зc	59	BD	26	96-B5	90	CD	10	зв	AF	ΘF	4C	
00023	80	F6	F6	46	A9	DB	7E	B1	52-86	27	38	76	DB	65	ΘB	D8	F~.R.'8v.e
00024	ŧ0	10	DB	A2	7F	сс	Β4	DB	78-5A	AB	78	02	C4	24	11	14	xz.x\$
00025	50	DA	01	71	В3	C8	03	28	10-0F	03	28	10	A9	0A	30	20	·q(0
00026									9C-0A								
00027																	\$/`h.h
00028																	@Pa.A.!`OU@01.
00025																	:%G∪U.6q
00026																	s#.3#q.3.
0002E																	.#3.#3S.Sr.R.
00020																	2`0U@01T.\$TE
00020																	40'G.fvWG
0002E																	'.G`h.h
0002F																	:P.0t<+ '×.0`.`Z.+QQ.
00031									5F-1E								
00032																	F
	.0	сø	03	-	ər	13	22	00	23 EZ	79	62	JE	-	00	F B	23	• .1. FUOUH.1. /.II. F
•																	• • • • • • • • • • • • • • • • • • •



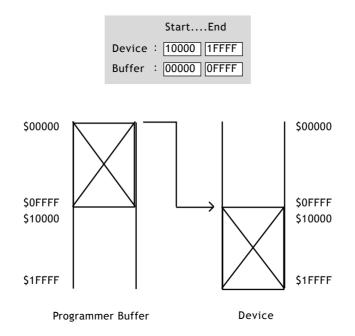
\_\_\_\_\_ 61 =

(10) Modify Programmer Buffer Mapping:

<b>E</b> , WA28F.EXE(V1.	00)			_ @ ×
<u>Eile Edit Device</u>	Operation USB Info. 1			
Read Auto	Blank Brogram	Verify Compare Erase	Brotect	
Programmer Status	Device	2	5	
Mfr.:	AMD			
Type :	AM29F200B	Edit Programmer Buffer Mapping	×	
Adaptor :	NONE	Start E	nd	
Device Start :			1FFFF O	
Buffer Start:	00000000	OK Cance	el	
Buffer Checksu Buffer Status :		iffer Size(Bytes): 2048K		
	et Buffer ddresses	programming	g Set Devi address	ce programming es
<u> </u>				

Example:

Program data from Buffer address \$00000 - \$0FFFF to Device address \$10000 - \$1FFFF.



#### 6.3.3 Operation

Different IC product type might have different options of programming functions. Basically, programming functions include options of Erase, Blank, Program, Verify, Lock, Auto, ...etc.

WA28F.EXE-CH	hlen_au.cab	- 8
ile <u>E</u> dit <u>D</u> evice	Qperation USB Info. About	
Image: Weight of the second	Baad Compare D Check D Check	
Programmer Status	s Uglock.	
Mfr.:	Ersee	
Туре	: Verify Protect	
Adaptor :	Auto he (Byte Wide)	
Device Start:	:: 00000000 Device End: 0001FFFF	
Buffer Start :	: 00000000 Buffer End : 0003FFFF	
Buffer Checks	sum: 001F	
Buffer Status:	PC Memory Buffer Size(Bytes): 2048K	
	Functional options for Device AM29F200B	

User can also click <Fast Key> to excute program functions. See <Fast Key> indicated below:

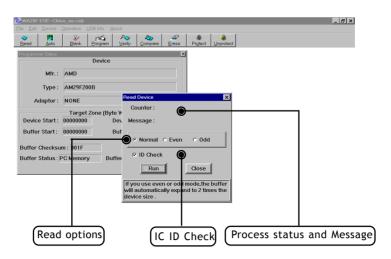
(1) Read:

Read contents in IC memory. Read function is only valid for IC on following position.

- Single ALL-100 setup: The IC on the socket.
- Multiple ALL-100 setup: The IC on socket of Site #0 programmer
- Single ALL-100 Gang setup: The IC on Socket #1 of Gang programmer.

 $\blacksquare$  Multiple ALL-100 Gang setup: The IC on the 1st Socket (Socket

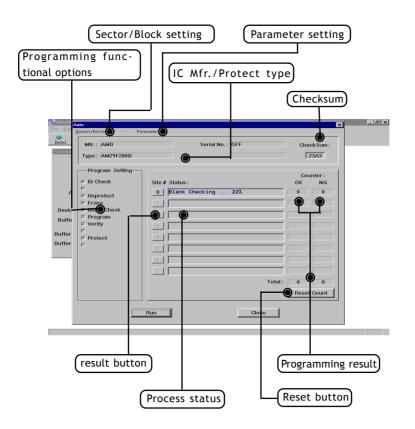
#1) of the 1st Site (Site #0)
Gang programmers.



 $\stackrel{\wedge}{\succ}$  After Read operation, suggest user to double-check to confirm checksum and the data read are all correct.

(2) Auto:

Enter programming mode with all programming functions activated.



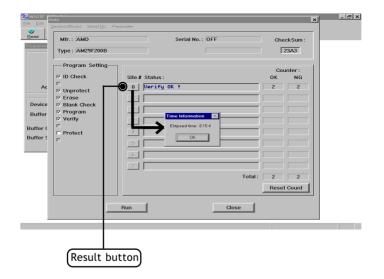
☆ Different IC product type might have different programming functional setting and parameter setting. User needs to refer IC data sheet for proper settings of programming functions and programming parameters. ■ Sector/Block: Specify Sector/Block to be protected if needed.

Mfr.: /	AMD tor Setup for Protect	Seria	al No. : OFF	CheckSum ·
ур 🔽	000000-001FFF			E
	002000-002FFF	Г	Г	Г
-P 🔽	003000-003FFF	Г	Г	Г
II 🔽	004000-007FFF	Г	Г	F
5	008000-00FFFF	<b>F</b>	F	F
υĒ	010000-017FFF		Г	F
E	018000-01FFFF		Г	
в 🗖				
Р		Г		
vΞ		Г	п	
Р		Г	E	F
		Г		F
E		Г	Г	F
		E	Г	F
			П	F
	Previous (S	et / Clear) all block	Next	OK Cancel

Parameter: Special setting for programming if needed.

Eile Edit Auto		a n USB Irlo. About Jo. Parameter	_ & ×
	Mfr.: AMD Type: AM29F	Serial No. : OFF CheckSun 200B 23A3	1:
De p Bu P Buff P Buff	7 Unprotect 7 Erase 7 Blank Chec 7 Program 7 Verify Protect	Site # Status :     OK     Counter :       "anguananiar Parameter     X     2       Progran / Verlfy < Normal	

## Result button: Button to display programming result and elapsed time.



- (3) Erase : Enter programming mode with Erase function activated.
  - Blank : Enter programming mode with Blank check activated.
  - Program : Enter programming mode with Program function activated.
  - Verify : Enter programming mode with Verify function activated.
  - Protect : Enter programming mode with Protect function activated.

Bead	Mfr.: AMD	Serial No. :	DFF	CheckS	
Ar Device Buffer Buffer	Type : KM29F200B	Site # Statue;           0           1           2           3           4           5           6           7	Total :	Counte OK 2 2 2 2 2 2 Reset Co	r: NG 2
_		Run	Close		

ad Igrammi	Mfr.: AMD Type: AM29F200E	Serial No. : OFF	CheckSum :
Ad Device Buffer uffer C uffer S	Program Setting P (D) Check F F 7 7 Program 7 Verity F F	Oite J- Ototus :           0	Counter: OK NG 2 2 2 2 7 7 0 7 0 7 0 7 0 7 0 7 7 0 7 7 7 7
		Run Glos	se

### 6.3.4 USB Info.

Display current ALL-100 connection status through USB interface.

WA28F.EXE-Chi	ien_au.cab	5 ×
<u>File Edit Device</u>	Operation USB Info. About	
🛷 隆 <u>R</u> ead <u>A</u> uto	Image: Weight of the state         Page         Image: Weight of the state         Image: Weight of the	
Programmer Status	Device	
Mfr.:	AMD	
Type :	AM29F200B	
Adaptor :	NONE Programmer Status	
Device Start : Buffer Start :	Target Zone (Byte	
Buffer Checksu	OK	
Buffer Status:	PC Memory Buffer Size(Bytes): 2048K	
	ALL-100 connection status	

## 6.3.5 About

Display model number of current programmer and version number of programming driver.

By WA2EF EXE-Chiler, quicab Elle Edit Device Operation _USB Into _About Φ   P   ↓ ↓ _AΔ   AΔ   Δ   ↓   Δ   ↓   ↓   ↓   ↓   ↓   ↓
Bead Auto Blank Program Verify Compare Erase Protect Unprotect
Programmer Status
Mfr.: AMD
Type: AM29F200B
Adaptor: NONE
Target Zone (Byte         About ALL-100         ×           Device Start:         00000000         De         Universal Programmer
Buffer Start: 00000000 Bu MODEL: ALL-100 (C) HI-LO
1.00 Buffer Checksum: 001F WA28FEXE V1.00
Buffer Status: PC Memory Buffe Driver: AM29200B.100 V1.00
(Version number of programmer and driver)

# 7. ALL-100 Troubleshooting

After power on, ALL-100 will run self diagnostics and USB connection check. Green/Red LED will display test result. Table listed below summarize test result with possible causes/dispositions for troubleshooting.

Condition	Possible cause/Dispositions
After power on, Green and Red LED off.	Poor contact / operation abnormal: 1. Check power cable connection. 2. Need troubleshooting. Contact your local dealer for service.
After power on, Green and Red LED flash.	<ul><li>Fail self diagnostic test:</li><li>1. Check if there is IC on socket (Should be removed).</li><li>2. Need troubleshooting. Contact your local dealer for service.</li></ul>
After power on, Green LED off but Red LED flash.	Fail ALL-100 memory diagnostic test: 1. Try to power on again. 2. Need troubleshooting. Contact your local dealer for service.
After power on or during programming, Green LED flash but Red LED off.	<ul> <li>Fail USB connection check:</li> <li>1. Check USB cable connection.</li> <li>2. Check PC BIOS setup to see if USB is disabled (Should be enabled).</li> <li>3. Need troubleshooting. Contact your local dealer for service.</li> </ul>
During programming, Greend LED on but Red LED flash.	USB communication error: 1. Check USB cable connection. 2. Need troubleshooting. Contact your local dealer for service.
During programming, Green LED on but Red LED become orange color.	USB data trasmission interrupted: 1. Check USB cable connection. 2. Check if PC is interrupted. 3. Need troubleshooting. Contact your local dealer for service.

## 8. Glossary

#### 8.1 EPROM, EEPROM, BPROM, and MPU

Programmable device: An integrated circuit (IC) that can be programmed.

#### ■ Bit, Nibble, Byte, Word, Double Word

Bit : A basic unit of binary data.

- Nibble : A groug of 4-bit binary data. A nibble ranges from 0H to FH.
- Byte : A group of 8-bit binary data. A byte ranges from 0H to FFH.
- Word : A group of 16-bit binary data. A word ranges from 0H to FFFFH.
- Double word : A group of 32-bit binary data. A double word ranges from 0H to FFFFFFFH.

#### Buffer

There is 4 Mbit memory buffer in ALL-100 Programmer. IC driver file can automatically allocate/arrange these 4Mbit memory space for programming depending on IC memory size and read/write needs.

When data needs to be programmed to IC, data needs to be loaded to programmer buffer first and then program to IC. When data is read from Master IC, the data is also stored in programmer buffer, it can then be edited or saved to disk for future use.

#### Buffer Start and Buffer End Address

It specifies the start and end addresses in programmer buffer in which data is to be programmed to IC in sequence. This is also the area that data is used for Checksum calculation.

#### CheckSum

This is the SUM of all data contents between buffer start and buffer end addresses. All data are added and the least significant 16 bits (4 HEX) are displayed as the Checksum. (Some data in some ICs might not be covered in Checksum calculation.) Checksum will be calculated after IC reading, file loading, type changing, or buffer editing.

#### Bit Count of data

A NIBBLE contains 4-bit data. A BYTE contains 8-bit data. A WORD contains 16-bit data.

MPU is normally in 8 or 16 bit width, but still have some in 12 or 14 bit width.

#### Device Start and Device End address

It specifies the start and end addresses inside IC device. During IC programming, data stored in programmer buffer will be written to this specified area.

#### USB interface

USB, Universal Serial Bus, is a high speed data transmission bus initiated by Intel and then supported by NEC, IBM, MicroSoft, Compaq,...etc. It is now a data transmission standard between PC and peripheral devices.

V1.0 / 1.1 USB Low-Speed : 1.5 Mb/s V1.0 / 1.1 USB Full-Speed : 12 Mb/s V2.0 USB High-Speed : 480 Mb/s

#### Security fuse

Security fuse is available in most of programmable ICs. Once the Security fuse has been blown, the data stored in IC can not be read out correctly and IC can not be programmed either. However, IC can still operate functionally no matter the Security fuse has been blown or not.

Note: Once the Security fuse has been blown. IC data can no longer be read out or programmed correctly, please double-check before programming Security fuse.

#### Lock bits

Some MCU/MPU use Lock bits to protect data programmed. Normally user has options to select individual Lock bit to protect different area of memory data. Please refer IC data sheet for definition of Lock bits.

#### Encryption

Some MCU/MPU use Encryption code for data protection. If an IC has been programmed with Encryption code, then a correct decryption code must be given to read the correct data.

#### Protection Fuse

Some FLASH memory use Protection fuse for data protection. It can prevent data change from accident programming. The Protection fuse must be reset to Unprotection state, if the programmed data need to be changed. The default state of Protection fuse is Unprotection.

### 8.2 PLD, PAL, GAL, PEEL, CPLD, EPLD, and FPGA

#### Programmable Logic Device (PLD)

PLDs are usually grouped into following five categories:

- PLD : A one time Programmable Logic Device such as PAL.
- EPLD : A UV Erasable PLD such as EPLD, CPLD, and FPGA.These devices have transparent window on top of package for UV light exposure.
- EEPLD : An Electrically Erasable PLD such as GAL, PEEL, CPLD, and FPGA.
- CPLD : A more complex PLD device.
- FPGA : Field Programmable Gate Array.
- JEDEC fuse map file of PLD

JEDEC fuse map file is a standard format used for PLD programming. It contains fuse information and functional test vectors of PLD to be programmed. Most PLD assemblers or compilers such as PALASM, OPAL, CUPL, ABEL, AMAZE, and PDK-1, can create JEDEC fuse map file.

#### ■ POF fuse map file of PLD

POF fuse map file is a format used for ALTERA PLD programming. POF file can store more programming data than JEDEC file.

#### Fuse blown and intact

Most of unprogrammed (blank) PLD have fuses in intact (connect) state. After programming, PLD fuses are blown to open state. For one time programmable PLD, once fuses are blown (opened), they can not be changed back to intact (connect) state. However, the UV erasable PLD can be erased to change fuses back to intact (connect) state by UV light exposure and the electrically erasable PLD can be electrically erased to change fuses back to intact (connect) state by using Erase function on this Programmer.

#### ■ Array fuse, Configuration fuse

Array fuses are the main logic fuses in a PLD. Different types of PLD have different logic function arrangement. Configuration fuses define the I/O architecture of a PLD such as Combinatorial/Registered, Output feedback/Output enable, and so on. Generally, user do not have to understand the details of these fuses because logic compiler will automatically translate logic statements and equations into JEDEC format file.

#### Security fuse

Most of PLD have Security fuse. Once Security fuse is blown, data in PLD can no longer be read out correctly. Generally, the PLD will be read as blank if Security fuse is blown. Note: