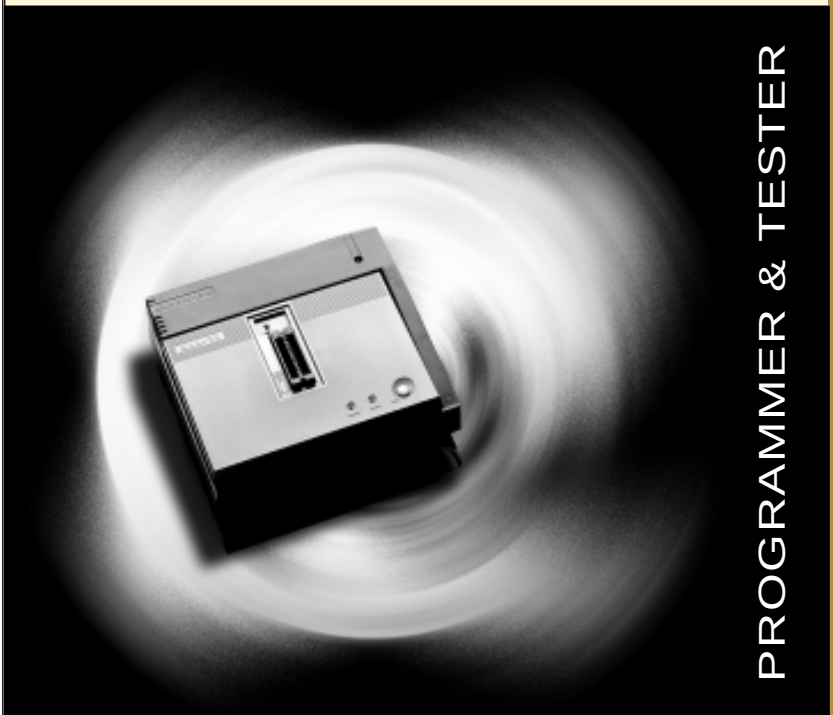
USER'S MANUAL

ALL-11P2



UNIVERSAL PROGRAMMER & TESTER

2000

1999

1998

1997

HI-LO SYSTEMS

Model numbers referred in this manual are:

Programmer: ALL-11P2

CONTENT

4. FUNCTION REFERENCE GUIDE

5. UTILITIES

1. INTRODUCTION

This manual provides instructions for installing and operating the Programmer on an IBM PC or compatible running Windows 95, 98 or Windows NT. Communication with the host PC is through RS232C data transmission (baud rate up to 115,200 bps) or parallel port data transmission. Inside the programmer, there is a high speed processor with a 1Mbit memory buffer as standard. All timing and programming control is done in the programmer to ensure accurate and precise programming waveforms. The 1Mbit buffer supports programming of memory devices and microcontrollers with up to 1Mbit memory as well as most PLD devices. The memory buffer for the serial port is expandable up to 128Mbit by inserting a PCMCIA SRAM CARD. If you do not have an SRAM Card, the software will automatically use the PC memory as the buffer for devices having more than 1M bits. If a serial port is used for programming communication, we recommend you use an SRAM Card to speed up the programming of devices having memory larger than 1M bits. However if a parallel port is used as a communication port, you do not need to use an SRAM Card. The programmer's pin driver circuits are fully programmable to support standard 5V IC's as well as low voltage IC's.

1.1 Programmer and Accessories

Your package should contain the following items:

* DB-9 pin to DB-25 pin adapter.
* 1 meter cable for the serial port with two 9-pin, D-type connectors on the cable ends.
* 1 meter cable for the parallel port with 15- to 25-pin and D-type connector at the end of the cable.
* Diskette(s) with windows control drivers.
* User's manual.
* Programmer
* Options: PCMCIA SRAM CARD (for the serial port only)

2. INSTALLATION

The Programmer is easy to install and use with any IBM 486/Pentium computer or compatible running Windows 95, 98 or Windows NT. You are assumed familiar with the installation of PC add-ons and software in PCs that run the Windows operating systems.

2.1 Host System Requirements

* IBM 486/Pentium or compatible PC (recommended Pentium or more powerful PC)
* Microsoft Window's compatible mouse is recommended.
* Approximately 10 to 20 Mbytes free disk space.
* Operating system: Microsoft Windows 95, 98, Windows NT.
* 2 serial ports: one for a mouse and the other one for the programmer. The 2nd one is definitely required. Or 1 parallel port with EPP, ECP, or bidirectional mode (can not use a unidirectional printer port.)
* 64MB DRAM memory or above is recommended.

2.2 Hardware Installation Procedures

(1) Before installation make sure there are two serial ports (RS-232C ports) in your PC: one for a mouse and the other for the programmer. Some PCs have a separate mouse connector so only 1 serial port is needed. In the case you have only one serial port and no mouse connection you can still operate the programmer software with keyboard commands.

(2) Or 1 parallel port which has ECP, EPP, or the bidirectional mode. A Multi I/O card can also be added to your PC in an extra ISA slot to add an additional serial port(s) or parallel port.

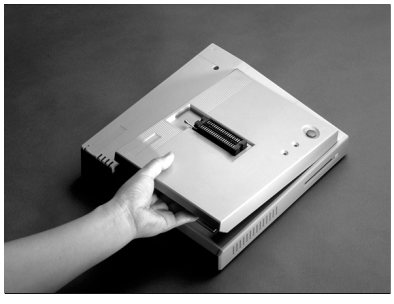
Note: The following guidelines should be followed during installation and during regular operation:

Step1:

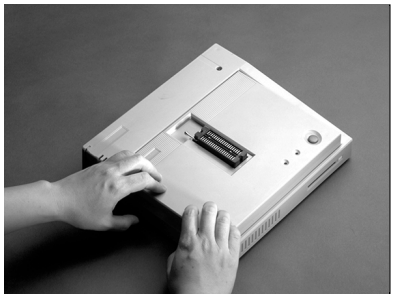
Before the Programmer can operate, the programmer PACK needs to be installed onto the programmer docking station. The docking system power should always

be turned off when inserting the PACK modules. First, switch off your programmer docking system power using the switch in the rear of the unit.

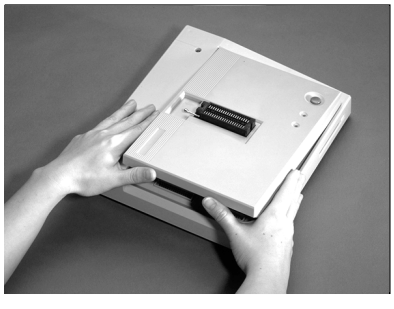
Next, insert the programmer PACK into the programmer docking system slot. The lip that extends from the bottom of the PACK should fit into the slot on the programmer docking system. Press down the PACK to sit firmly as in the following picture.



Insert



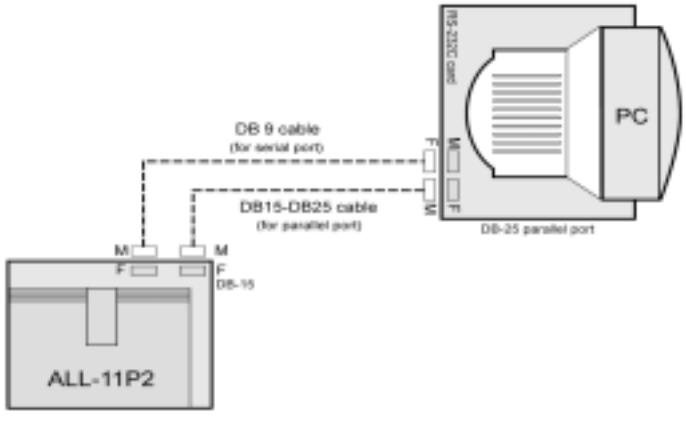
Press



Remove

Step 2:

With the host PC and programmer docking station still turned off connect the programmer and PC as in the following sketch: (using serial port as an example)



a) For the serial port, please connect the male connector of the DB-9 cable to the female connector of the DB-9 connector on the programmer docking system. Connect the female connector of DB-9 cable to RS232C port of your host PC. A DB 9 to DB25 connector is provided for a PC with 25-pin serial port connectors.

b) For the parallel port, please connect the male connector of a DB-15 cable to the female connector of a DB-15 cable on the programmer docking system. Connect the male connector of a DB-25 cable to the parallel port of your host PC.

Step 3:

First, power on the computer, then the programmer. The programmer's on/off switch is located in the back of docking station next to the power line connector. The power supply of the programmer can accept 90 VAC - 270 VAC, 47 Hz -65 Hz power sources.

Second, check LEDs on the programmer.

ON LED must be ON.

BUSY LED must be OFF.

GOOD LED is in random state.

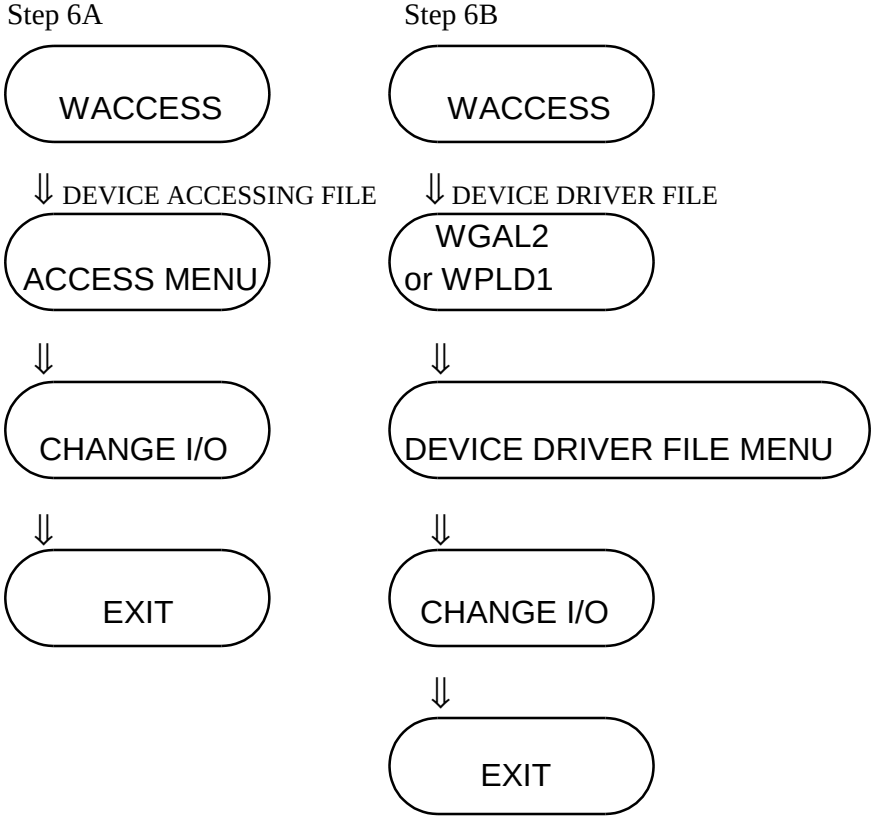
If the LEDs are not in the correct state, turn off the PC and the programmer and check all connections between the serial port and the programmer. If an additional RS-232C or parallel card is being used then check that the card is securely installed in the ISA bus slot in the PC. Then turn on the computer and the programmer to check the LEDs on the programmer again. If the LEDs are still not in the proper state, you may have a conflict with another peripheral on the PC set to the same COM port or parallel port address as the port you are using.

2.3 Software Install Procedure:

Insert programmer install Disk #1 to Floppy Disk Drive or copy all the disks' files to Hard Disk Drive. Execute "SETUP.EXE" by using File Manager or the "RUN" option under STARTUP button of WINDOWS Program. The "SETUP" program will open a window as the following picture so that you can follow the install instruction step by step to install your programmer software into your computer. For WINDOWS NT to install the software, you must be logged on as an Administrator.

The hardware identification test can be performed in two ways:

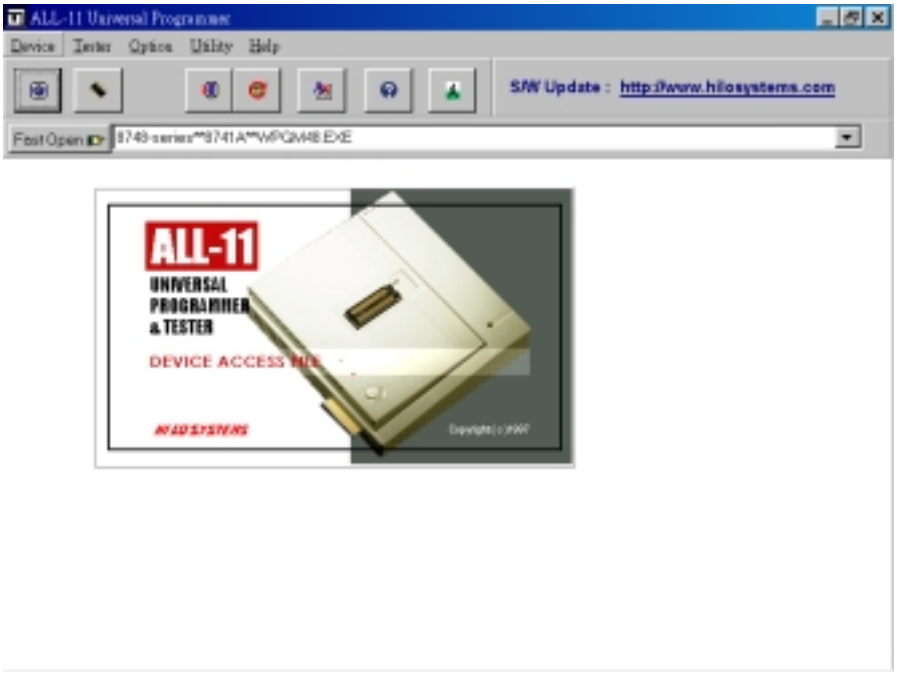
1. Run the WACCESS.EXE file
2. Run a Device Driver file under WACCESS



two ways to check hardware identification status

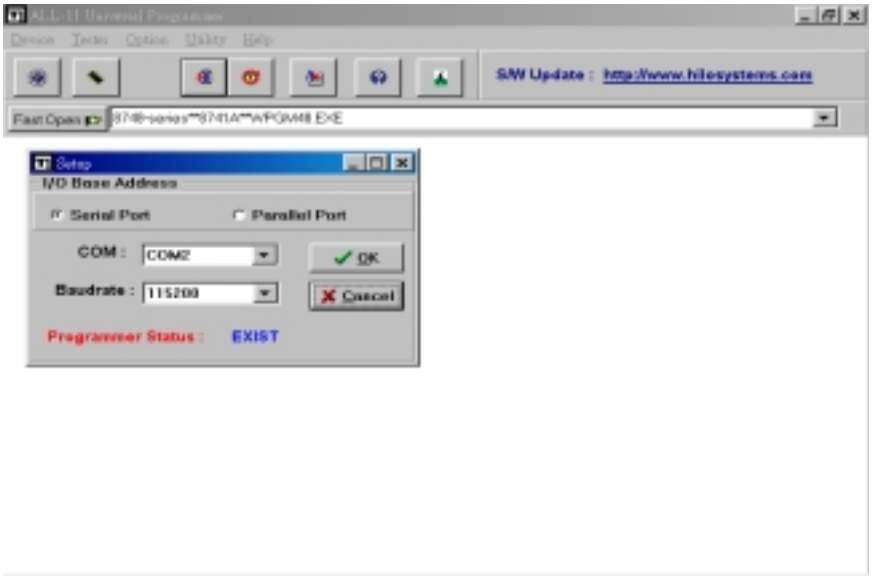
1) Hardware identification from WACCESS.EXE menu.

Double click the "WACCESS" icon in the Programmer Group window and the main WACCESS window will appear as below. In Windows 95/98/NT you can use the START button and then select "Programs" to get access to all available programs on your PC.



<Access Main Menu>

Click on the "Setup" menu and select the "0. I/O Base Address" function.



<I/O selection menu>

The current I/O base address registered in the software will appear on the screen and the software will automatically check hardware identification circuits in the programmer. The software will display “Programmer Status: Exist” or “Programmer Status: Not Exist”. When “Exists” and “OK” appear in the status window that means the hardware installation is OK.

The "Not Exist" message appears when the hardware I/O address (COM port address or parallel port address) does not match the I/O address selected in software, when the RS-232C hardware COM port or parallel port address or IRQ setting conflicts with other add-on cards inserted in the PC, if the cables and interface connections are not firmly fastened, if the programmer is not turned on, or if the programmer docking system is not properly connected to the programmer.

If the "Not Exist " message appears, please try the following to resolve the problem:

1. Make sure the programmer is well connected to the programmer docking system, the power is turned on and the green ON LED is ON.
2. Check if the RS-232C (or parallel port) connection between docking system and PC is properly connected.
3. If no COM port settings work, please try setting the baud rate to a slower speed. Some PC's serial port hardware cannot support 115K baud transmission speeds. However if no parallel port settings work, please check if the PC's BIOS setup sets the parallel port to ECP, EPP, or Bidirectional mode, not the Normal or Unidirectional mode.
4. Turn the programmer's power off and then on again.

When the software is exited the COM port or parallel port address and baud rate setting will be saved as the default settings for future use.

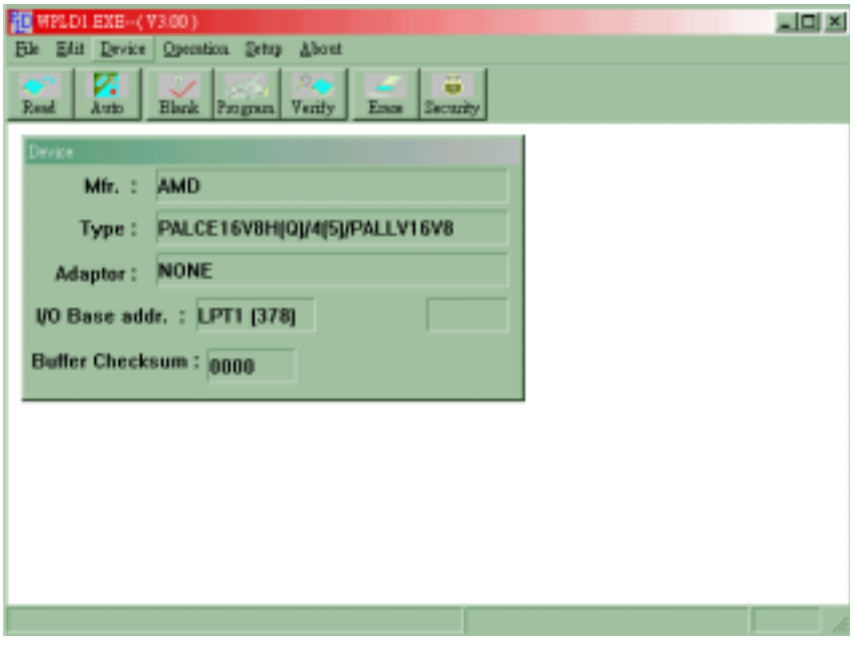
* Refer to the Function Reference Guide for more details about the change I/O address function.

2) Hardware Identification from a device driver file

WPLD1.EXE is one of the GAL device driver files. Device driver files are the files called by WACCESS when a particular device manufacturer and type are selected. These files can also be called separately for execution as in this example.

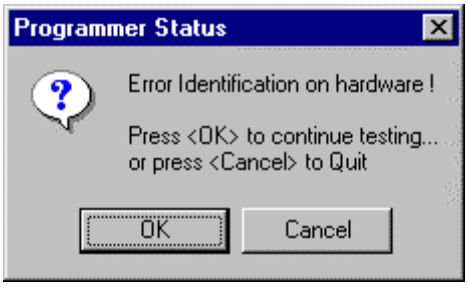
If WPLD1.EXE is executed the first thing it will do is look for the parameter file WPLD1.DAT. If it exists on the default disk drive, the parameters of I/O address (COM port or parallel port address), baud rate selection (for serial port only), Manufacturer, and Type will be loaded as your default state. Otherwise, the default values of COM port 3 (3E8), baud rate (115K) or parallel port, Manufacturer (AMD), Type (PALCE16V8H), will be considered as the default state.

Once a device driver file is entered the hardware identification circuits will be rechecked. If everything tests OK then the driver file main menu will appear and you are ready to program devices. The main menu will look as follows:



<Main Menu>

If the following error message appears on the screen:



then the address of the serial port the programmer is connected to does not match the default software COM port or parallel port address, there are conflicts with other add-on cards or peripherals inserted in the PC, the programmer power is not turned on, the docking system is not properly connected with the universal programmer, or the baud rate selected is too fast for your host PC's serial port hardware. (If you use a parallel port, please ensure the PC's BIOS setup sets the parallel port to ECP, EPP, or Bidirectional mode, not Normal or Unidirectional mode.)

There are three solutions as follows to solve the problem:

1. Press < enter > to enter the main menu of the device driver file, then click on the "Setup" menu and select "I/O Setup" followed by selecting the listed COM port addresses: COM1, COM2, COM3, COM4 sequentially until the "Programmer Status: Exist" message appears. If none of these addresses will work then try selecting a slower baud rate and try selecting the COM port settings again. (If a parallel port is used, please select the "Parallel" from "I/O Setup".)
2. If the first solution fails and you are using an add-on serial port card or parallel card, press "Q" to quit, and then turn off the computer. Check that the IRQ setting and COM port or parallel port setting of the card does not conflict with any other PC peripherals. Change these settings to any of the other choices you believe are not occupied by other peripherals and retry selecting the COM port or parallel port addresses and baud rate (for serial port only) settings in our software again to match the new COM port or parallel port address.
3. Contact your local distributor for assistance.

Once the installation is correct all settings will be remembered for future use when you exit the software.

* Refer to Section 4.4.2 for more details about change I/O address.

NOTE

Most problems are due to poor connections between serial port cards and the PC slot, poor cable connections between the serial port or parallel port and the programmer, or poor connections between the docking system and the universal programmer.

3. USING THE PROGRAMMER

Before starting to use the programmer definitions of some commonly used symbols are explained here for your reference.

3.1 Definition of Key Symbols and Terms Used for Programming

3.1.1 Device Symbols for EPROM, EEPROM, BPROM and MPU

DEVICE: An integrated circuit (IC) that will be programmed or copied.

* **Bit, Nibble, Byte, Word, Address and Buffer Base Address**

Bit : A basic unit of Binary data.

Nibble : A group of 4-bit Binary data. A nibble's value ranges from 0H to FH.

Byte : A group of 8-bit Binary data. A byte's value ranges from 0H to FFH.

Word : A group of 16-bit Binary data. A word's value ranges from 0000H to FFFFH.

Double Word: A group of 32-bit binary data. A double word's value ranges from 00000000H to FFFFFFFFH.

*** Working Buffer**

A block of memory inside the programmer controller allocated by the device driver file. The working buffer is used by the device driver file as an intermediate storage.

When you want to program a data file to a DEVICE you first load the file to the working buffer. The contents in the working buffer are then programmed to the target DEVICE. When you read the contents of a master DEVICE the data is stored in the working buffer. The working buffer can then be edited or saved to disk for future reference.

*** Buffer Start and Buffer End Addresses**

The buffer start and end addresses are offset addresses specified from the base address of the Working Buffer. The data stored between the buffer start and buffer end addresses contains the subset of the programmer memory buffer that is programmed to a target device. This is also the area of the buffer included in checksum calculations.

*** Check Sum**

This is the sum of all data contents between buffer start and end addresses. All bytes are added byte by byte and then the least significant 16 bits (4 HEX characters) are displayed as the check sum. Some bytes in the address range may not be included in the checksum as specified by the manufacturer of the particular DEVICE. Check sums will be calculated during DEVICE reading, file loading, type changing or after buffer editing.

Bit count of the data contents of several DEVICES:
NIBBLE WIDE PROM is 4.
BYTE WIDE PROM is 8.
WORD WIDE PROM is 16.
MPU is usually 8 bits but can be 12, 14, or 16 or some other count depending on the device.

*** Device Start and Device End Addresses**

The DEVICE start and end addresses are offset addresses specified by a particular DEVICE. Data stored at the buffer start address in the working buffer will be programmed to a DEVICE beginning at the Device Start Address location.

NOTE: In order to get correct programming data, some device files forbid users to change the Buffer Start, Buffer End, Device Start, and Device End settings.

* **EVEN and ODD address mapping**
  The address sequence of data contents can be assigned to CONTINUOUS (NORMAL), EVEN or ODD (note: not all devices have EVEN/ODD functions) whenever you want to READ, PROGRAM or VERIFY the EPROM.

For example, in a program function selection the software will give you three selections --- Normal, Even, or Odd.
You may click on Normal (Continuous) to program the data in the buffer to the device as follows:

Buffer start	+ 0	to	Device start	+ 0
	+ 1	to		+ 1
	+ 2	to		+ 2
	...	to		...

You may click on Even to program the data in the **EVEN** addresses to the device as follows:

Buffer address	+ 0	to	Device start	+ 0
	+ 2	to		+ 1
	+ 4	to		+ 2
	...	to		...

You may click on Odd to program the data in the **ODD** addresses to the device as follows:

Buffer address	+ 1 is programmed to Device start	+ 0
	+ 3	+ 1
	+ 5	+ 2
	...	...

Troubleshooting Hint:

Whenever using the even or odd functions, the software will automatically expand your working buffer to twice as large as your target device address space. For example, if your target device is a 27C256 with addresses from 0h to 7FFFh you want the buffer start address to be 0000h and the buffer end address to be FFFFh which is the address space of a 27C512.

In the READ operation, the software will perform these functions in the reverse direction. Even/Odd address mapping is used for 16 bit systems that use 2 eight bit memory devices to store 16 bit program code.

* **I/O Address (the COM port or parallel port)**
  This is the I/O base address of the RS-232C port or parallel port. Each I/O interface card added into a PC slot will occupy one or more I/O addresses.

* **Counter**
  This is the programming address counter. During DEVICE programming, the counter on the screen will scroll to show the current address in the target device that is being programmed. (Note: not all software has this function)

* **MFR & TYPE**
  Every DEVICE has its own manufacturer (MFR) and type number (TYPE). Please refer to Chapter 4 for detailed descriptions.

* **Security Fuses**
  Security fuses are present in many microcontroller devices. When a security fuse is programmed the data stored inside the microcontroller can not be read by a device programmer for reverse engineering purposes. Generally a device will read as blank if the security fuse is blown. The device will operate functionally equivalent with the security fuse intact or blown.

Troubleshooting:

Some UV erasable devices can not be programmed again once the security fuse is programmed. Please check with your data book before programming the security fuse of a UV erasable (windowed) microcontroller.

*** Lock Bits**

Some microcontrollers use lock bits in place of a single security fuse. Usually you have the option of selecting to program 1 lock bit or all lock bits to provide different levels of code protection. The definition of these bits varies from manufacturer to manufacturer. Please consult the data book of your target device for more details.

For some devices like Microchip PIC series devices, if your Lock Bits in the data file are set, the checksum will then change because of the PIC devices' checksum counting rules.

*** Encryption**

Encryption is another form of code protection found in some microcontrollers. If you burn the security fuse of a device then nobody can read the contents of the device. If a device is programmed with an encryption code it can not be read unless the person operating the programmer enters the proper encryption code. Without this code only garbage data will be read.

*** Protection Fuses**

Some FLASH based memory devices have protection fuses. A Protection fuse is used to protect a sector of memory in the FLASH device from being accidentally programmed or modified by the target hardware where the device will be installed. This fuse will need to be reset by a specific series of events before the data in the particular sector can be programmed or changed to new values. Protection fuses are only programmed using the separate "Protection" function or the "Auto" function. Please see the function descriptions section of the manual for more details.

3.1.2 Device Symbols for PLD, PAL, GAL, PEEL, CPLD, EPLD and FPGA

*** Programmable Logic Device (PLD)**

Generally speaking, a device that can be programmed to perform many different logic operations is a PLD. PLDs are generally grouped into following five categories:

PLD : A one time programmable logic device such as a PAL device.
EPLD : A UV erasable PLD such as a EPLD, CPLD, or FPGA device. These devices contain a transparent window on top for UV light exposure.
EEPLD : An electrically erasable PLD such as a GAL, PEEL, CPLD, or FPGA device.
CPLD : A more complex PLD device.
FPGA : Field programmable gate array.

*** JEDEC Fuse Map File of a PLD Device**

The JEDEC fuse map file is the standard format which can be loaded to a programmer for programming PLD's. It contains fuse information (Blown/Intact) and the Function Test Vectors of the PLD (optional). Most PLD assemblers or compilers, such as the PALASM, OPAL, CUPL, ABEL, AMAZE and PDK-1, will produce a JEDEC fuse map file.

*** POF fuse map file for PLD devices**

The POF fuse map file is a format used for programmable logic devices from Altera Corporation. POF files store programming fuse data for larger devices in a more compact way.

*** Fuse Blown and Intact**

A new programmable device generally comes with all fuses in an intact state (i.e. logic 1 or blank state). The designer can only blow an Intact fuse into a Blown state i.e. logic 0 state. The default state of a new device may be opposite to this depending upon the technology used to manufacture the device. For one time programmable (OTP) devices you can not change the state of a fuse back to the unprogrammed (default) state once it is blown (programmed). UV erasable devices have a window on them. When this window is exposed to UV light for some time the fuses will be erased to the default (or unprogrammed) blank state. Other devices are electrically erasable and can be erased by using an ERASE function on the Programmer.

*** Array Fuses & Configuration Fuses**

Array fuses are the main logic fuses in a PLD. Different arrangements (Blown/Intact) will have different logic functions.

Configuration fuses indicate the I/O architecture of the PLD, such as Combinatorial/Registered, Output Feedback/Output Enable. Generally the end user does not have to know the function of these fuses because the logic compiler used to translate your logic statements and equations into a JEDEC fuse map or other format for programming into a device will specify the value of these bits.

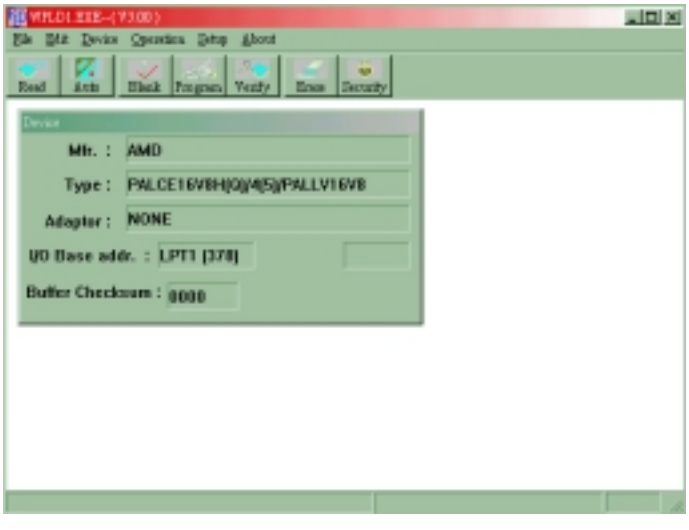
* **Security Fuses**
  Security fuses are present in many PLD/CPLD devices. When a security fuse is programmed the data stored inside the PLD can not be read by a device programmer for reverse engineering purposes. Generally a device will read as blank if the security fuse is blown. The device will operate functionally equivalent with the security fuse intact or blown.

3.2 Viewing the Main Menu of a Device Driver file

Before we begin operation please take a moment to familiarize yourself with the main functions found in the device driver files. Device driver files contain all of the device specific functions. Functions that are not available for certain devices will be faded out and can not be selected.

3.2.1 Main Function Menu

All main programmer functions are listed on the left side of the window. Status information is displayed on the right side of window.



<Main menu >

3.2.2 Status Field

In the upper right side of the window is status information about the target device being used including MFR, TYPE, Adapter, I/O Base address, & Baud rate. This is called the Device Status Field. Below the Device Status Field there is information about the target device being used, such as device start address and device end address, check sum, buffer start address, buffer end address, buffer size. This is called Buffer Status Field.

Whenever the programmer is being used, please make sure that data in the status field matches your requirements. Otherwise the target DEVICE may be damaged or programmed to an unknown state. Many devices have ID codes that the programmer will read to make sure the software algorithm called is the correct algorithm for the target device inserted in the socket. However, not all devices have an ID code so it is best not to rely on this feature.

By now you should be familiar with the main menu screen. A brief overview of most of the available functions is contained in the rest of this section. All relevant functions will be discussed in detail in later chapters.

3.3 Getting Started

The following example will demonstrate how to start and exit the device driver main menu through WACCESS and use functions such as DEVICE, LOAD, BLANK CHECK and PROGRAM. If you do not get the desired results please check that your installation is correct.

In the example, we are going to use an AMD PALCE16V8H GAL device. The procedure for working with other types of devices is similar.

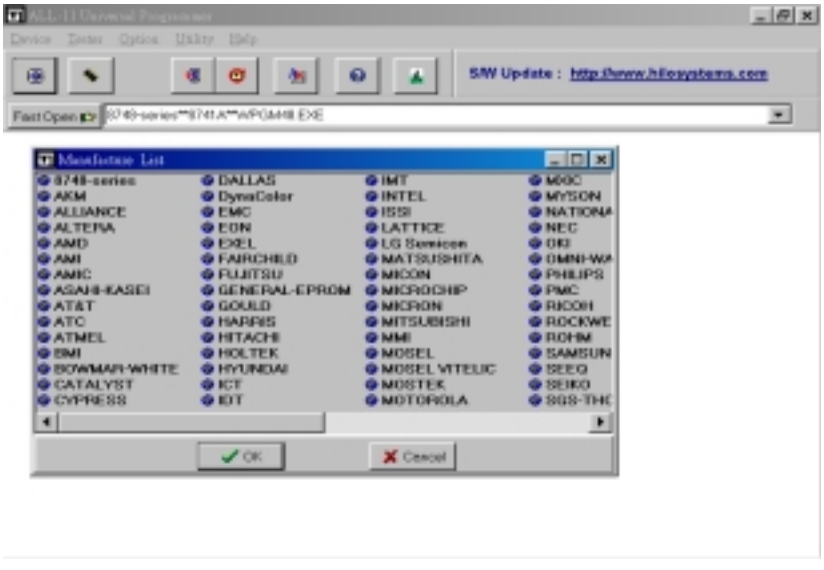
3.3.1 Starting from the WACCESS file

3.3.1-1 Execute the WACCESS program.

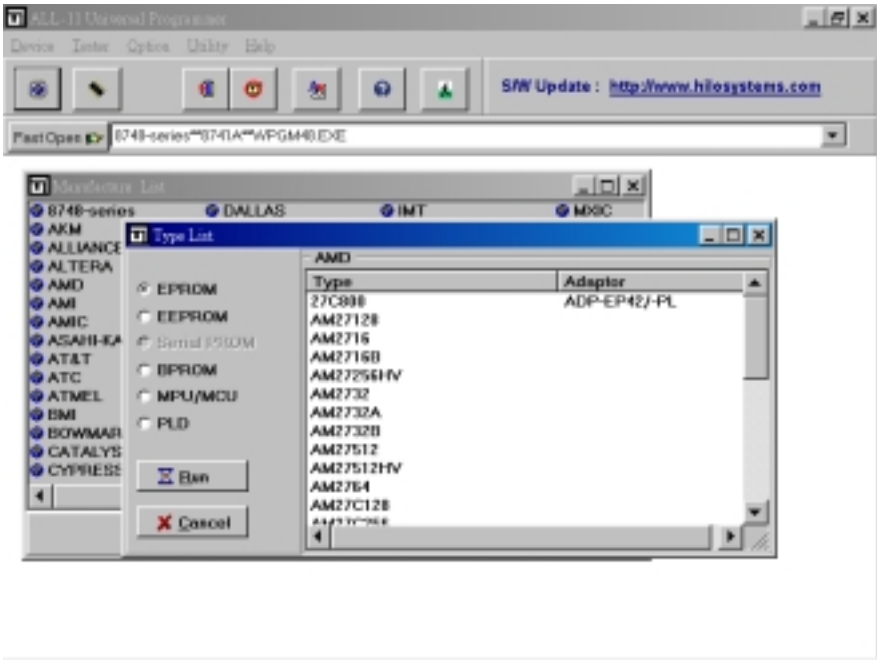
Double clicking the WACCESS icon from the programmer's window will execute WACCESS.EXE. WACCESS will open the following window:

3.3.1-2 Select MFR

Click on the "Device" heading to select the desired manufacturer, in this case AMD, by double clicking on AMD. Alternatively you can click the desired manufacturer name and press the <OK> key. Then the type menu screen will appear.



3.3.1-3 Select Device Group and Type

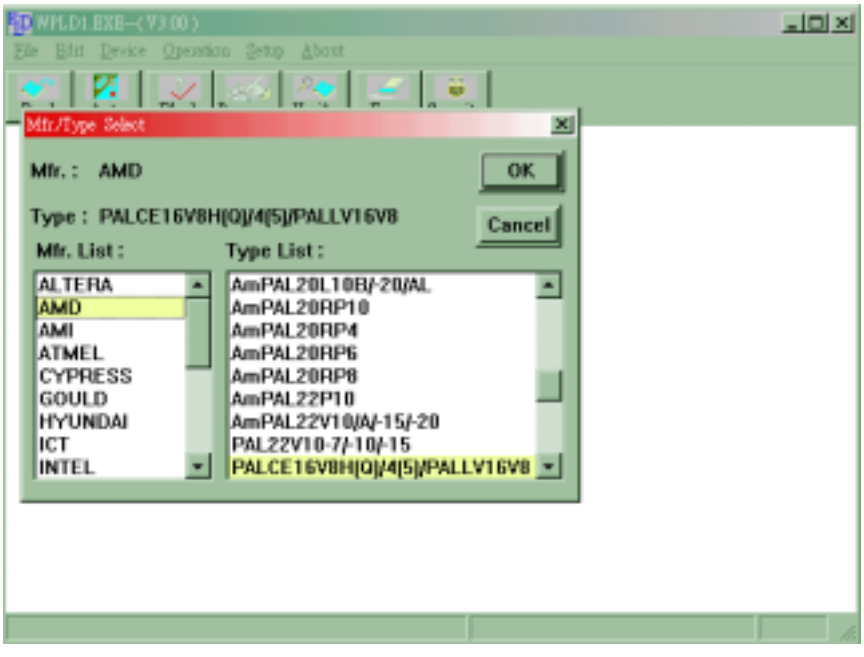


Type Selection

The device type is selected in the same way as the manufacturer. You can use the mouse to click on a particular device. After selecting the correct type, click on the RUN button and the WACCESS program will automatically search for and execute the relevant device driver file for the manufacturer and type selected. In this case the WPLD1.EXE driver file will be executed and the status field of the driver file will indicate the device manufacturer and type that was just selected and the driver file main menu will be displayed. Please note the WACCESS program will record each IC type you select so the next time you can use the Fast Open button to select the desired device quickly. If you wish to clear the devices in Fast Open, you may delete the WACCESS.MRK file.

Software Note:

All driver files can be executed individually. You can also select PLD MANUFACTURER and TYPE directly within individual driver files. Select the "Device" function to get a list of device manufacturers and types supported by the current device driver file. Make a selection by moving the mouse to the desired manufacturer and clicking the left mouse button or by using the < Up > and < Down > arrow keys followed by pressing the < Enter > key when the desired device is highlighted. The WPLD1.EXE manufacturer and type selection window is shown below.

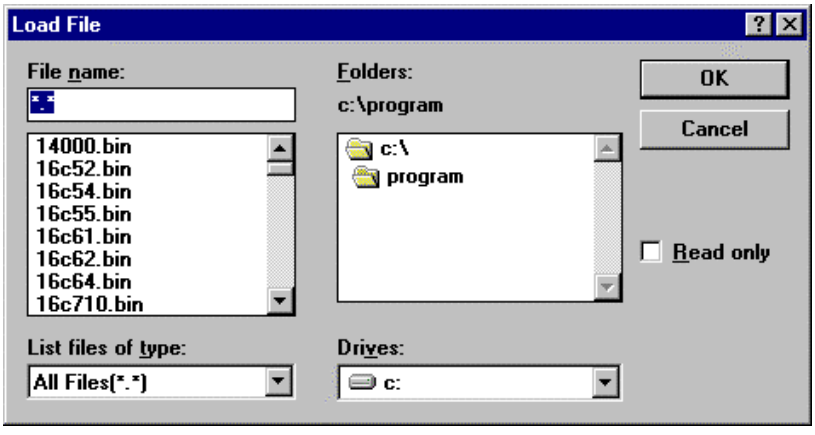


Troubleshooting:

When the manufacturer and type are entered, the data will be updated in the status fields and the relevant driver file will be loaded to the programmer docking station. If you select a device that requires an adapter to be programmed you may also get this message. Adapter driver files are only included with adapters that you purchase to save room on your hard disk and keep your driver files up to date when adapters are purchased. If you select a manufacturer and type from WACCESS program but the wrong manufacturer and type appear after, the reason for the problem may be that your driver is too old. You need to check the driver and select the Device menu to confirm that the desired device's manufacturer and type exist.

3.3.2 Load Disk File into Buffer

After selecting a device manufacturer and type you are ready to begin working with a device. Usually you will want to program a device with a data file you have stored on disk or you will want to read the contents of a master device and copy the contents from the master to another device. We will cover both alternatives in this example. In this example we are working with a PLD/GAL device which requires a JEDEC fuse map file as input. To load a JEDEC file from disk select "Load JEDEC File" from the "File" menu and the following window will open:



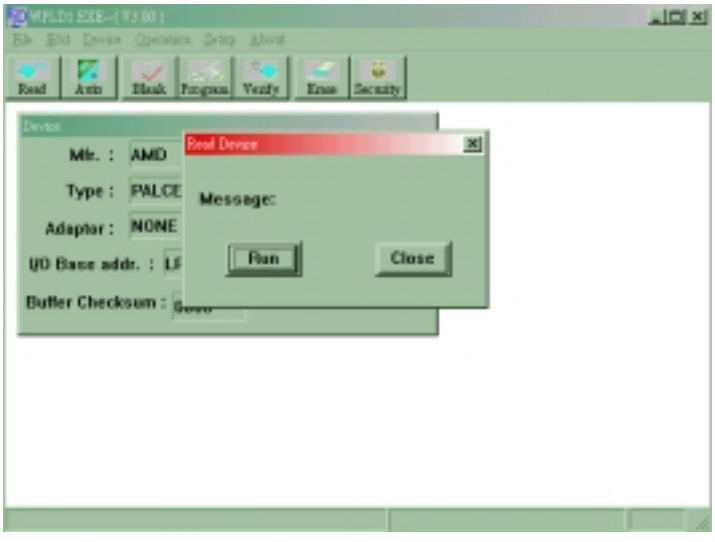
Submenu of file loading

The file loading window is similar to file loading/opening windows in most other windows programs. The desired filename may be typed in the "File: " input box and the indicated file will be transferred to the programmer's working buffer. The drive letter and any path names must be correct to get the desired results. An alternative to the direct typing method is to select a file from the files displayed in the list box directly under the "File Name:" heading. You can mask which files are displayed by clicking a file type mask in the "List Files of Type:" control box. To list files found in a different directory click on directory folders in the Directories list box shown on the right side of the window. To list files found on a different drive click the down arrow next to the "Drives" control box and select a new drive letter (i.e. a:, b:, e:, etc.)

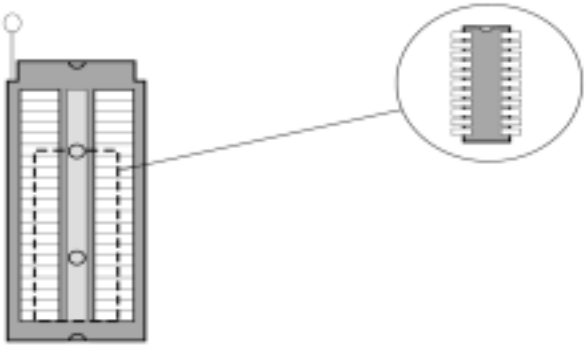
If you do not like to use the mouse you can use the <TAB> key to switch between windows and the <up> and <down> arrow keys followed by <enter> to make selections.

3.3.3 Read Contents from Master PLD

If the PLD data is in a MASTER PLD instead of a disk file, you can transfer the device contents to the working buffer by typing "R" or clicking on the "READ" button and the following window will open:



Insert the MASTER PLD into the socket while the lever on the 40-pin DIP socket is in the up position. Pin 1 should be located on the lever side of the DIP socket. If the device has less pins than the DIP socket then the device should be bottom justified in the socket as shown in the following diagram:



Close the lever. Then click the "OK" button or press 'Y' on the keyboard or press the YES key on the programmer. The data of the MASTER PLD will be transferred to the programmer buffer and the window will display:

Reading now...

OK!

Click the "Cancel" button or press < Esc > to return to the main menu.

3.3.4 Insert a Blank PLD into the Socket

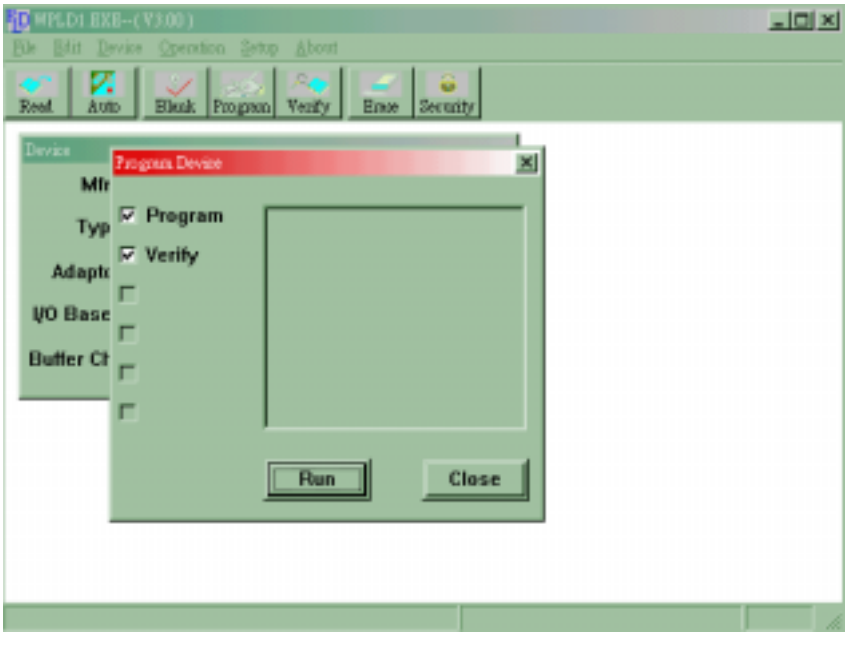
After transferring the data from the disk file or the MASTER PLD to the working buffer you are ready to program a new device. Insert a blank PLD into the DIP socket on the programmer while the pull lever is up. Again, Pin 1 should be located on the lever side of the DIP socket. If the device has less pins than the DIP socket then the device should be bottom justified in the socket. Close the lever. (Please see the previous diagram.) If you are not sure if the device in the socket is BLANK please use the "B" function to test if the device is really blank before programming.

CAUTION

The notch-end of the PLD must face the lever on the programmer DIP socket. If the device has less than 40 pins then the device should be bottom justified so the lower pins on the device (farthest from Pin 1) are closest to the GOOD LED on the programmer. If the IC is inserted incorrectly the PLD may be damaged or programmed to an unknown state. As mentioned earlier, the programmer will read an ID code in the device to make sure it is talking to the correct device and the device is in the correct position but not all of devices have an ID code so it is best not to rely on this feature.

3.3.5 Program Buffer Contents to PLD

After loading the disk file or reading the MASTER PLD data into the working buffer and inserting a blank PLD, the program function is used to copy the contents of the working buffer to a device. Invoke the programming function by clicking the "Program" button or typing "P", the PROGRAM function will open the following window:



<Submenu of Program>

Move the mouse pointer to click on "OK" or press "Y" on the PC keyboard or the "YES" key on the Universal Programmer to start programming the buffer contents into the blank PLD. After programming the device, new contents will be compared against the contents stored in the programmer's working buffer. This is called a VERIFY operation. If the contents match then the GOOD LED will light. For programming other PLD's, wait until the BUSY LED turns off, then replace the PLD in the socket with a new blank device and click "OK" again or press 'Y' on the keyboard or press the "YES" key on the programmer.

If you want to exit the programming process, click "Cancel" or press < ESC > to return to the main menu.

The operation to program a PLD is very simple and similar to the method used to program nearly all other types of devices. Many other functions and operations are available with this Universal Programmer. The details of all functions are described in Chapter 4 FUNCTION REFERENCE GUIDE.

3.4 Expansion Adapter and Converters

The rapid development of Integrated Circuits has lead to a huge variety of device types and packaging techniques. We have developed a wide array of adapters and converters that allow the Universal Programmer user to support virtually any IC type or device packaging, such as PLCC, SOP, TSOP, QFP, PGA, etc. that comes on the market. These adapters are designed with standard DIP footprint so a programmer with a 40-pin or larger DIP socket can program devices in virtually any package and with virtually any pin count. Devices with 8 pins to over 200 pins can be supported on the same programmer.

3.4.1 Adapter and Converter Installation

Adapter –

Each adapter has a 40-pin DIP footprint and can be plugged directly into the 40-pin DIP socket on the programmer. Adapters support devices that have 44-pin or more or devices that require some special programming requirements. Each adapter includes a software driver that should be copied into the same directory as the WACCESS.EXE program.

Converter-

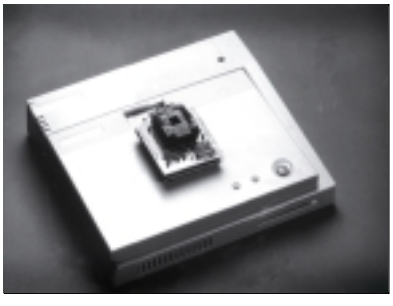
Each converter translates from a DIP footprint of a device to a surface mount footprint (PLCC, SOP, TSOP, etc….). Converters support devices with 8-pin to 44-pin and do not require any additional software.

S/W Installation -

As mentioned earlier, adapter software should be installed under the same directory as the programmer software drivers and WACCESS.EXE.

H/W Installation –

Insert the adapter or converter into the DIP socket of the programmer just like you would install a 40-pin DIP IC into the socket as shown in the following figure. Some converters have a DIP footprint smaller than 40-pin and should be bottom justified just like an IC with fewer than 40 pins. Converters generally have a notched corner on the bottom PCB that indicates the position of the DIP footprint Pin 1. Adapters have a Pin 1 marked on the top PCB or a silk screen drawing of DIP IC with a notch in the drawing to indicate Pin 1 of the DIP footprint. The silk screen is found on the bottom PCB facing upwards.



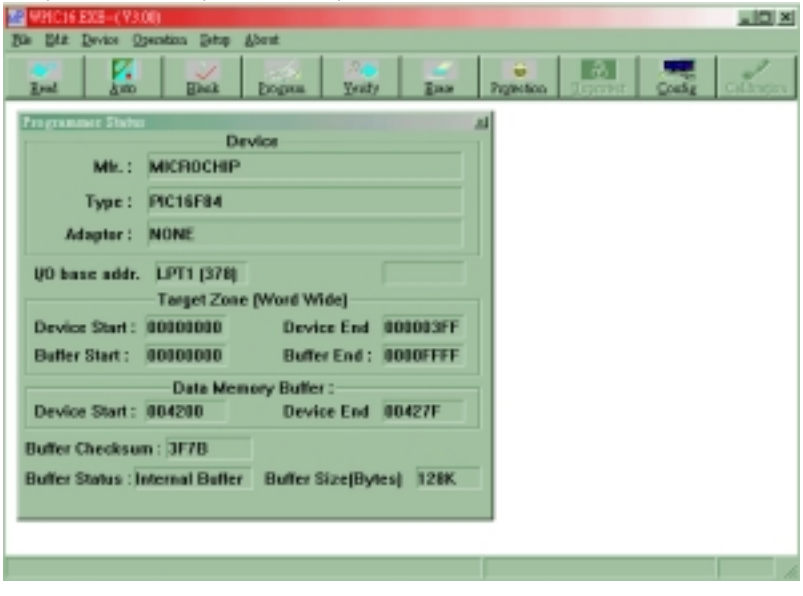
4. FUNCTION REFERENCE GUIDE

The following reference guide describes each of the functions found in the main menu of the Device Driver files. The functions will be described in the order given in the menus. To enter into the desired function, type the first letter of the function name or click the left mouse button when the desired function is pointed to with the mouse pointer.

Software Note:

All software drivers basically contain 1 of 2 screen layouts. Memory devices and microcontrollers use one layout and PLD/PAL/GAL/CPLD and other programmable logic devices use another screen layout. Each layout contains all of the functions relating to those particular family of devices. If a device does not support a particular function then that function will be displayed as faded and can not be selected.

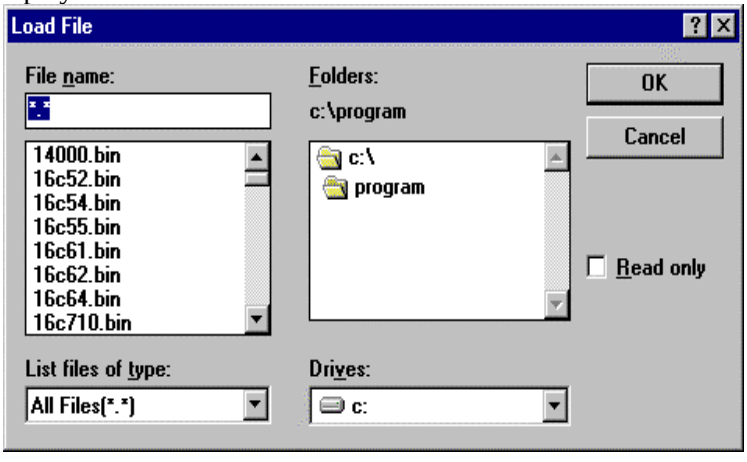
EPROM, EEPROM, BPROM, and MPU Driver File menus



<File Format>

4.1 Load BIN or HEX File to Buffer

Select "Load file to programmer buffer" from the "File" menu. A dialog window will be displayed on the screen as follows:



<Load file>

The file loading window is similar to file loading/opening windows in most other windows programs. The desired filename may be directly entered in the File: input box and the indicated file will be transferred to the programmer's working buffer. The drive letter and any path names must be correct to get the desired results.

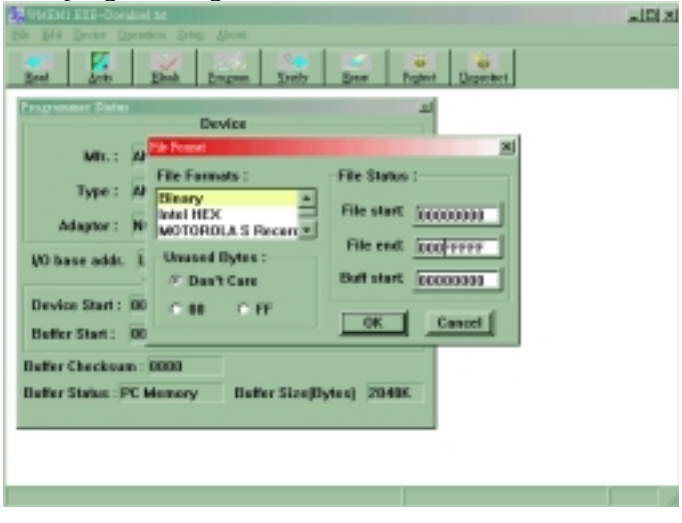
An alternative to the direct typing method is to select a file from the files displayed in the list box directly under the "File Name:" heading. You can mask which files are displayed by clicking a file type mask in the "List Files of Type:" control box. To list files found in a different directory click on directory folders in the Directories list box shown on the right side of the window. To list files found on a different drive click the down arrow next to the "Drives" control box and select a new drive letter (i.e. a:, b:, e:, etc.) If you do not like to use the mouse you can use the <TAB> key to switch between windows and the <up> and <down> arrow keys followed by <enter> to make selections.

File Format Window

Once the desired input file is entered or selected move the mouse to the "OK" button and click the left mouse button. Once the file is entered, the window will automatically switch to the “file Format” window as shown in the next diagram. The following file format choices are available for most devices:

1. BIN file
2. INTEL HEX (for INTEL Hex file)
3. MOTOROLA S HEX (for Motorola S record formats)
4. JEDEC (for PLD only)

You should select 'Binary' if the file you are loading is a binary file, 'Intel HEX' if the file you are loading is an Intel HEX file, or 'Motorola S Record ' if the file is in one of the Motorola S Record formats, and 'JEDEC' if the file is in standard JEDEC fuse map format. If your assembler or compiler has the option to output binary files it is the best to use that option when loading files for memory devices or microcontrollers. Files already in binary format will be loaded faster. All HEX files are actually converted to Binary by the driver file before being loaded to the buffer for programming.



<File Format>

File Status Settings:

For BIN or Hex formats you can key in the file start, file end address, and buffer start address to specify where to load the file. The default for most device types is to load your entire file to the buffer and to locate the data at buffer address 0000h. To do this, do not change any values in the file status area. The following is a description of other possible File Status settings:

File Start:

To load a file starting at the first data location in the file, leave this value at 0000h. If you only want to load data beginning at a certain address, 2000h for example, you would enter 2000h for this value.

Troubleshooting:

If you leave this value at 0000h and load a HEX file you may get the message "Some HEX codes are out of range" and you may get a checksum of 0000h or F800. This means that none or only a portion of your data was loaded to the buffer. HEX files contain address information and may contain an offset. For example, if you load a HEX file with an offset of 1000h to the buffer while leaving the File Start value as 0000h and the Buff Start value as 0000 then buffer locations 0 -> FFFh will be filled with your Unused Bytes value and your file data will appear starting at location 1000h. Please consult your assembler or C compiler linker manual for details on output file formats.

File End:

This is the last address of the file being loaded that will be placed into the programmer's buffer. If you want to load an entire file then leave this value at its

default value (don’t change this value in this case). If you only want to load file data from the file start address up to an address before the end of the file, 1000h for example, you would enter 1000h for this value.

Buffer Start:

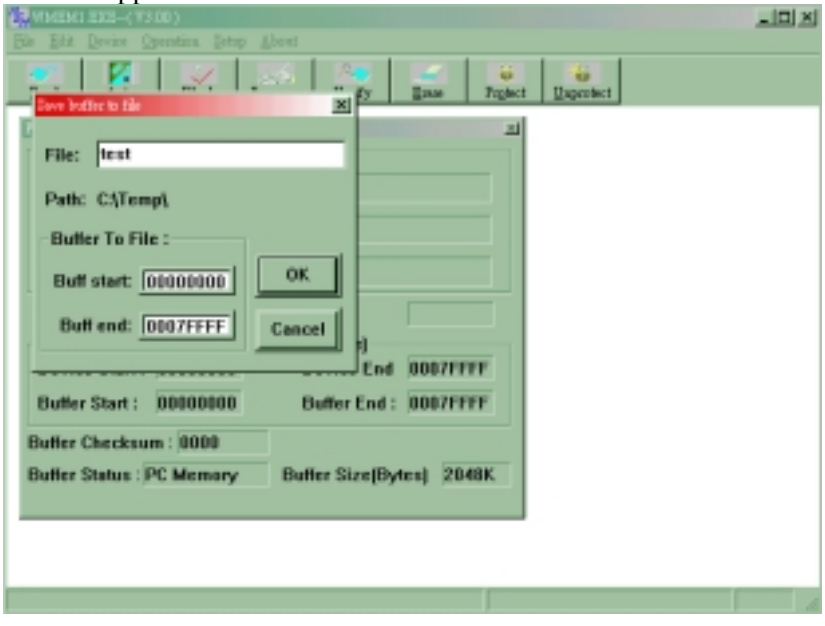
This is the address in the programmer’s buffer where the file data located at the file start address mentioned above will be placed. The default value for this option is 0000h. If you want the file data to be placed in the buffer starting at a different location then enter that address here.

Unused Bytes Settings:

If “Intel HEX” or “Motorola S Record” formats are loaded, you can choose to fill 'Unused bytes' with '00' or 'FF' or 'don't care’. In a typical application when a HEX file is loaded to an EPROM only a portion of the EPROM's available memory is actually filled with your program data. The rest of the unused portions of the EPROM may be filled with all 00's or all FF's depending on your preference. If you would like to load one or more additional HEX files to one DEVICE then you should choose 'Don't care' for the Unused Bytes option. This means that the portions of the buffer not used by the HEX file being loaded to the buffer will not be changed in any way. For example, if you load one HEX file and then load an additional HEX file to the same buffer and select ‘00’ or ‘FF’ for the unused bytes option then all portions of the buffer that are not filled by the new file will be filled with ‘00’ or ‘FF’, including the areas where the first file you loaded is occupying. To get the correct results choose the “Don’t Care” option for the Unused Bytes selection when loading additional HEX files to the buffer.

4.2 Save Programmer Buffer to Disk

Click on "Save File from Programmer buffer" from the "File" menu. A dialog window will appear on the screen as follows:



<Save File>

When prompted, click the desired file name or type in the complete name under which you want the file to be saved including the drive letter and any path names and then press < enter > or click the “OK” button. You can save the entire buffer contents to disk or just a portion of the buffer. Prompts are displayed for the "Start Address" and “End Address”. Please key in a hexadecimal address to respond to these prompts and then all the data in the specified address range will be saved to disk.

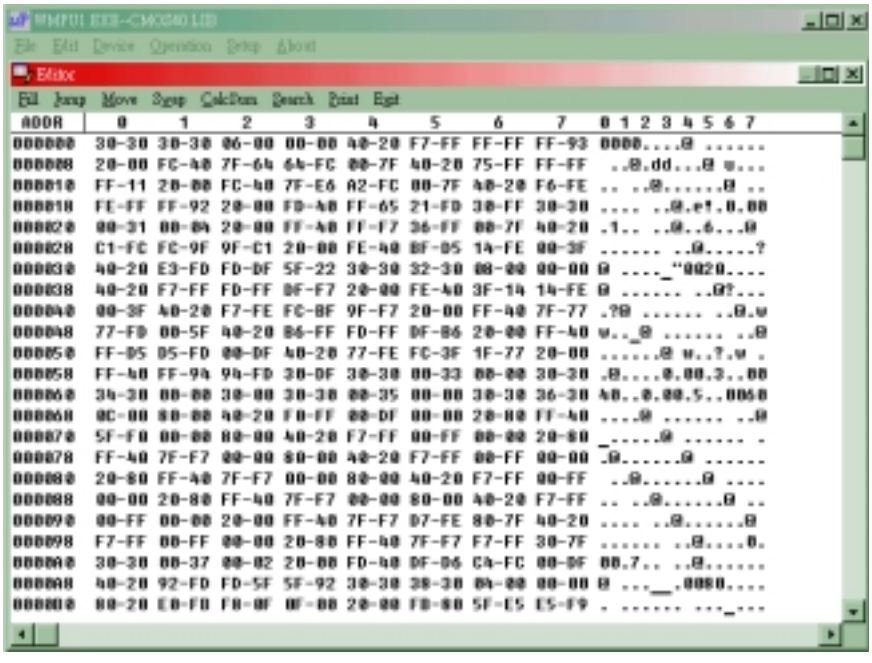
When the OK! message appears the function is completed. Press < ESC > to return to the main menu.

NOTE

Files saved under this function are Binary files with start and end addresses entered as byte addresses.

4.3 Edit Buffer

Click "Edit" and select "Edit programmer buffer" and the following window will be opened:



<Edit Buffer>

The editor is used to view or modify data contained in the programmer buffer. The following functions are available in the editor:

1. Fill : To fill a block of addresses with the same value.
2. Move : Copy data from one block of buffer memory to another block of memory.
3. Swap : Exchange the high and low nibbles of each byte in the buffer.
4. CalcSum : Calculate the check sum for a block of memory.
5. Search : Search the programming buffer for specific ASCII or binary data.
6. Exit : Return to the device driver file menu that called the editor.

All of these functions bring up small parameter windows where appropriate. Parameters include start and end addresses to perform functions on and data values to fill the buffer with and values to search for.

4.4 Setup Function

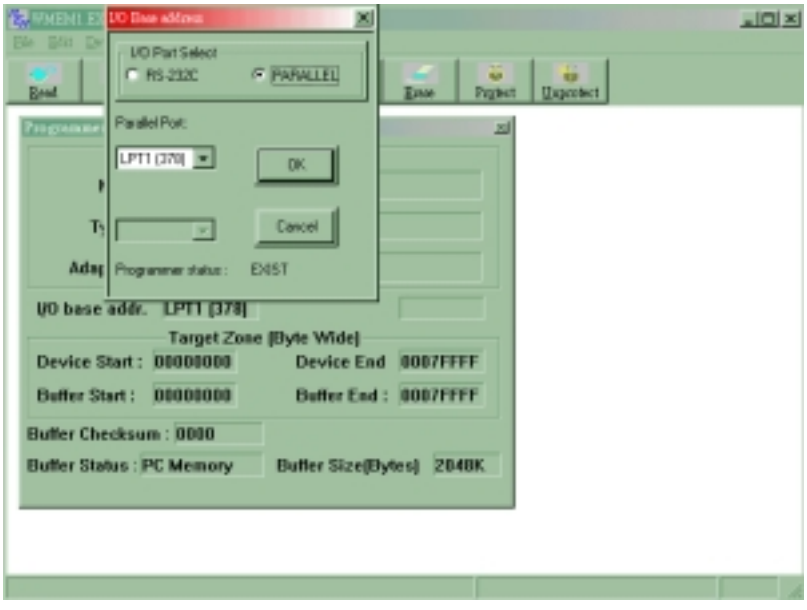
4.4.1 I/O Setup

Before performing any functions the device driver file's I/O address setting must be set to match the address of the serial port or parallel port that the programmer is connected to.

If you install a new serial port or parallel port card you can usually select the I/O address by positioning a jumper on the serial port or parallel port card. The computer power must be switched off when changing this jumper position. Generally for serial port the mouse occupies COM1(3F8), therefore, any one of COM2, COM3 and COM4 can be used for connecting the programmer. The desired parallel port, LPT1, LPT2, or LPT3, can also be used.

For the serial port, the "Baud rate" function is to select the communication speed. It is recommended to choose 115,200 bps (default) for the fastest transmission speeds. If your PC does not have a good UART (serial port IC) or if you use a very long transmission cable you may have to select a slower speed to get the programmer to communicate with the host PC properly.

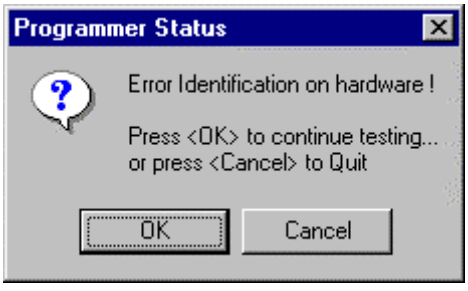
To change the device driver's serial port or parallel port address and baud rate setting click on "Setup" and then select the “I/O Base Setup” function. The following screen will be displayed:



<I/O Setup>

Select the serial port or parallel port that corresponds to the port the programmer is connected to. The software will automatically check the hardware I/O circuits and display the result "Programmer status: Exist" if everything is set up OK. If the "Programmer status: Not Exist" message appears then the software serial port or parallel port address needs to be changed to match the port that the programmer is connected to or the baud rate (for the serial port only) setting needs to be lowered. When you exit the program using the normal exit function the serial port or parallel port address and baud rate setting will be remembered for the next time you use the programmer.

Each time a driver file is executed the software first checks that the software and hardware are set to the same serial port or parallel address and communication is functioning properly. If the address set in software does not match the address set in hardware or if the baud rate is set too fast for the host PC's hardware to handle then the following message will be displayed:

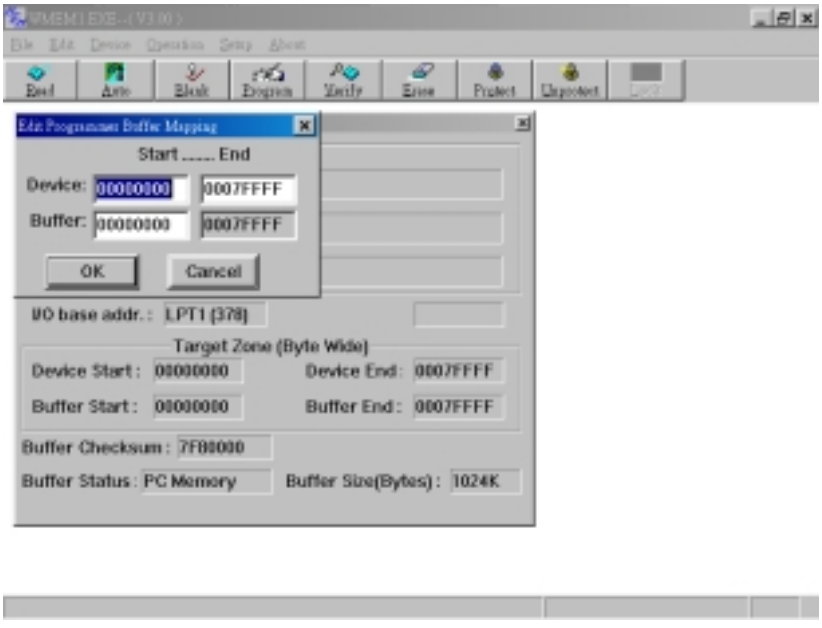


Click "OK" to enter the device driver file main menu. Click on the "Setup" function and try changing the serial port or parallel port settings first. If that does not solve the problem then try selecting a slower baud rate for the serial port only and try selecting the serial or parallel port addresses again. When everything is set up correctly you should see the message “Programmer Status: Exists”.

When you click "Exit" to leave the driver file the new serial or parallel port address and baud rate will be saved in a ????????.DAT parameter file for later reference. The I/O base address serial or parallel port only needs to be set one time and then will be remembered by the software.

4.4.2 Modify Programmer Buffer Mapping

The modify programmer buffer function is used to select what portion of the buffer is programmed to a target device and what portion of the buffer is included in the checksum calculation. Usually this function is used to limit the size of the buffer if a file is small compared to the EPROM size of the target device you will program. Click on “Setup” in any driver file and then click on “Modify Programmer Buffer Mapping” and the following window will appear:



<Modify buffer>

To modify the buffer mapping type new values in for the device start and end addresses and for the buffer start and end addresses. Addresses should be entered as HEX values. The “Device:” start address is the address in the target device where the information starting at the “Buffer:” start address will be programmed to in the target device. The values entered in HEX code will be updated in the STATUS FIELD for reference.

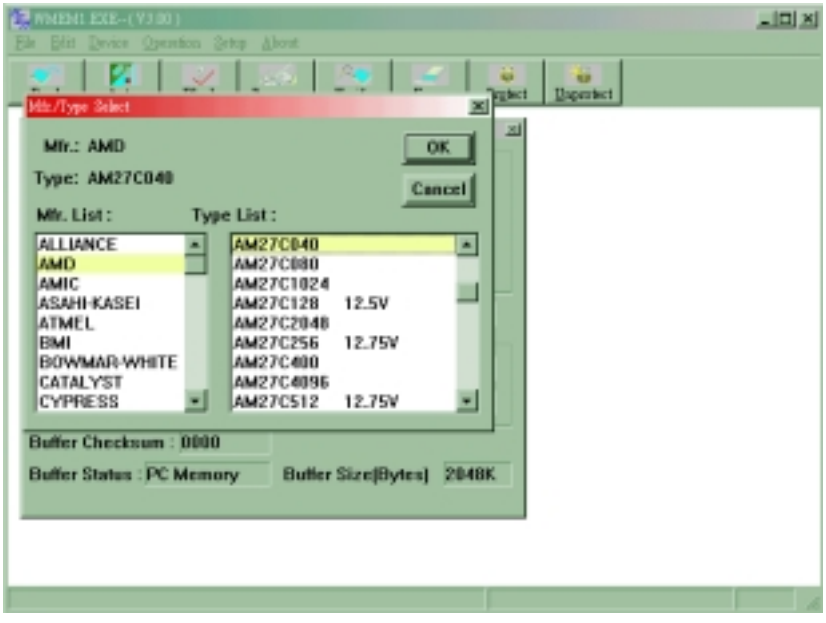
Note:

When the type or manufacturer (Mfr.) selection is changed the buffer mapping will default to the normal mapping specified by the device selected. The normal address range is equal to the entire address range of the target device. This is not supported by all driver software to ensure correct data programming.

Click "Cancel" to return to the main menu.

4.5 Device Select

Click on "Device" and the following screen will appear:



<Type Window>

A list of manufacturers is displayed on the left side of the screen under the “Mfr. List:” heading and a list of device types for each manufacturer that is selected is displayed on the right side of the screen under the “Type List” heading. Once a desired IC manufacturer and device type are selected click the “OK” button and the STATUS FIELD will be updated with the new manufacturer and type selection.

Click < Cancel > to return to the main menu.

Troubleshooting:

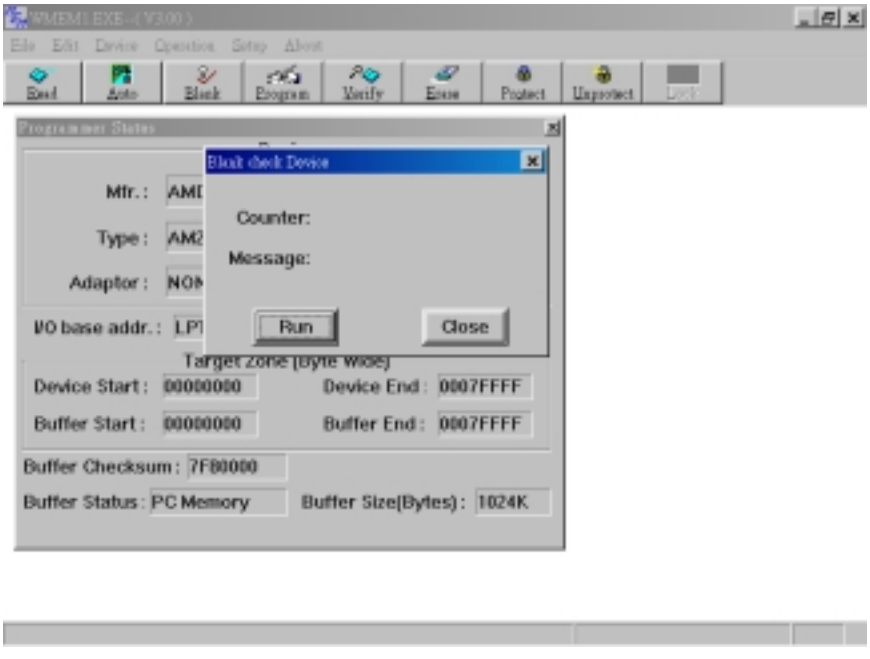
When the manufacturer and type are entered, the data will be updated in the status fields and the relevant driver file will be loaded to the programmer docking station. If the message "BIN not found" appears that means that the software for the selected manufacturer and device type is not installed on your PC. Please check your installation diskettes for the requested file or contact your local distributor. If you select a device that requires an adapter to be programmed you may also get this message. Adapter driver files are only included with adapters that you purchase. This saves room on your hard disk and helps keep your driver files up to date when adapters are purchased.

4.6 Exit

Click on Exit is to leave the current function and return to windows or WACCESS. Before exiting the software will save parameters including manufacturer, type, COM port, and baud rate in the ????????.DAT file associated to the driver file in use.

4.7 Blank Check

Click "Blank check", and the Blank check window will be displayed as shown.

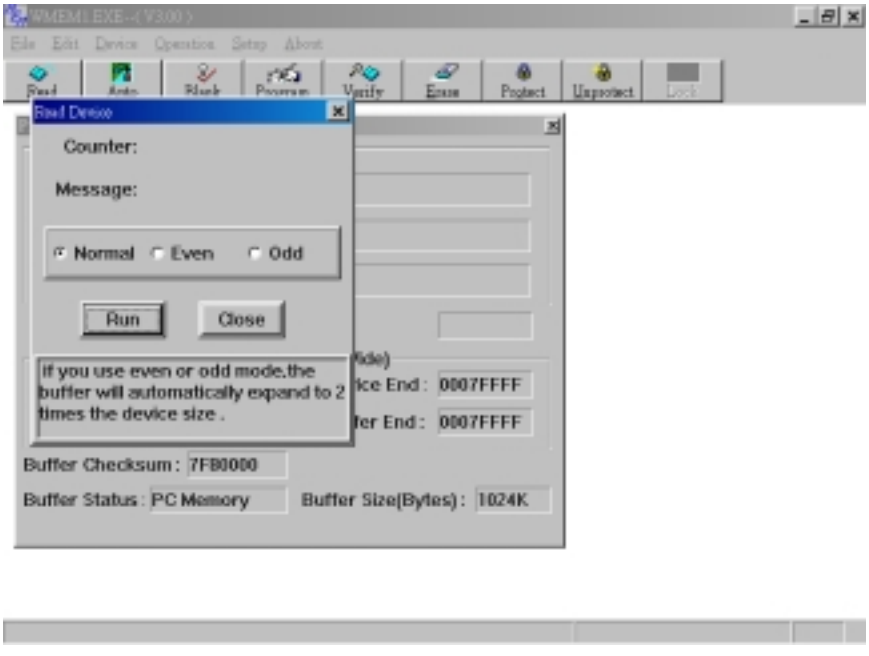


<Submenu of Blank check>

The blank check function is used to check whether a target device is ready to be programmed. All devices, except for EEPROM devices, should be in the blank state before being programmed. Click "OK" or press "Y" on the keyboard or the YES key on the Universal programmer to start the blank check. If the device fails the test, the first address that is not blank will be displayed. Otherwise, the "OK!" message will be displayed. The <Esc> key can be set to interrupt the blank check function.

4.8 Read

Click "Read" and the Read window will be displayed:



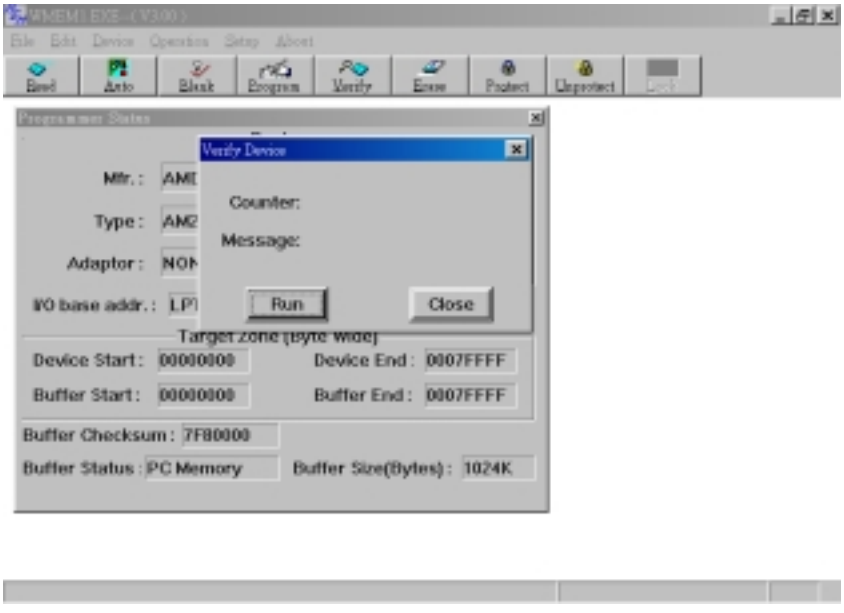
<Submenu of Read>

Click the "OK" button or press " Y " on the keyboard or press the "Yes" key to read data from a device in the socket to the programmer buffer. Date will be read from the device according to the settings in the Programmer Buffer Mapping status window shown in the mapping STATUS FIELD. Click "Cancel" to return to the main menu. The Read Window will display the “Reading now...” message, and when the read is completed, the "OK!" message will be displayed and the GOOD LED under the socket will light.

The check sum of all the data read into the buffer will be automatically calculated after reading a device.

4.9 Verify

After clicking on "Verify", the verify window will be displayed:



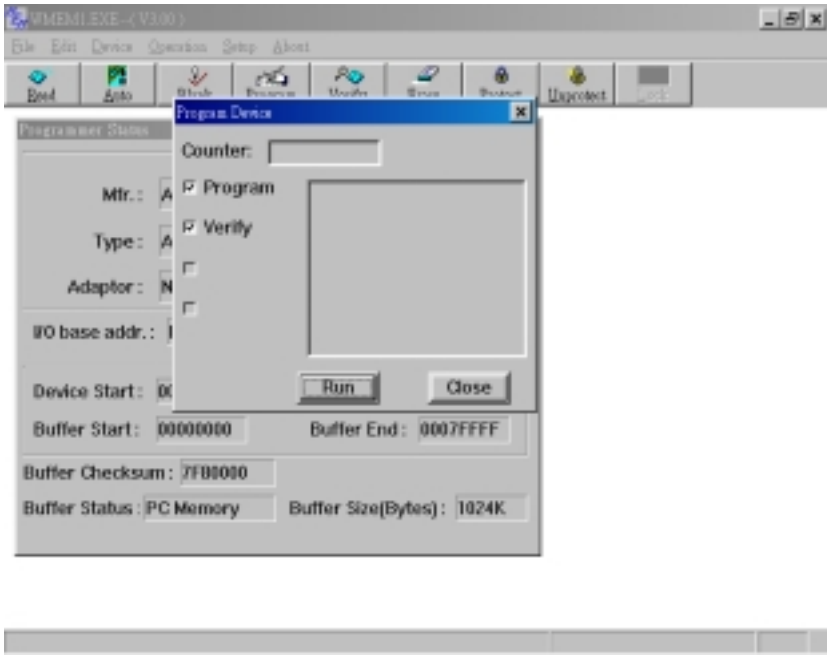
<Submenu of Verify>

Click the "OK" button or press " Y " on the keyboard or press the "Yes" key on the programmer to verify that the target device contents match the contents stored in the programmer buffer at the address range shown in the STATUS FIELD, or click "Cancel" to return to the main menu. The Verify Window will display the "Verifying now" message. When the verify function has been completed, the “OK!” message will be displayed on the screen and the GOOD LED below the socket will come on.

The verify routine will be terminated and an error message will be displayed if the contents of the device under test do not match the contents found in the working buffer.

4.10 Program

Click "Program". The program window will be displayed as shown.

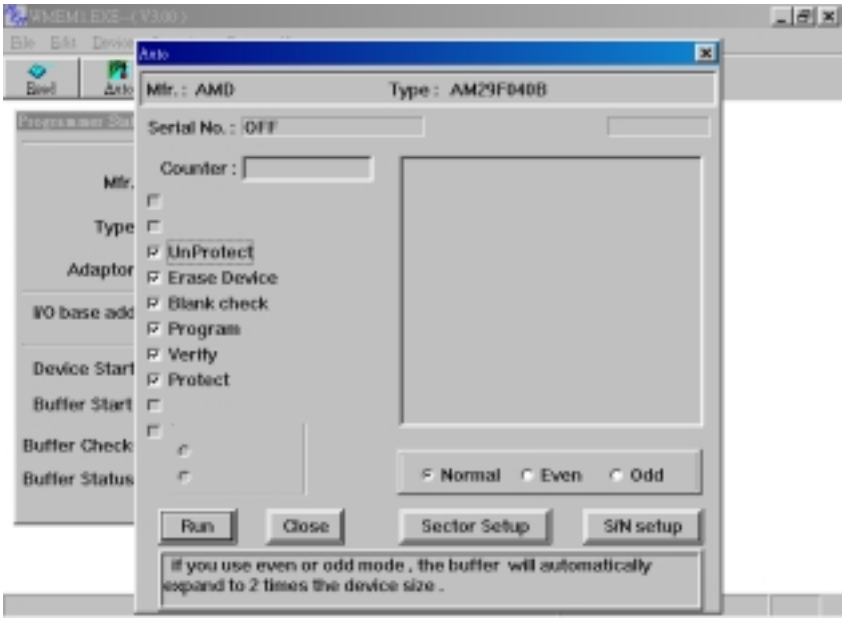


<Submenu of Program>

Click the "OK" button or press " Y " on the keyboard or press the "Yes" key on the programmer to begin transferring the data in the working buffer to the target device, or click "Cancel" to return to the main menu. The “Programming now” message will appear on the screen and the counter in the Status Field will increment until the function is completed. Then, a “Program OK!” message will be displayed on the screen and the GOOD LED will come on if the operation was completed successfully.

4.11 Auto

Click "Auto". The Auto window will be displayed:



<Submenu of Auto>

Click the "OK" button or press " Y " on the keyboard or press the "Yes" key on the programmer to start the Auto function, or click "Cancel" to return to the main menu. There are many choices to be enabled or disabled in the auto function. The Auto function window displays all available functions for a given device manufacturer and type. Next to each function is a box that can be enabled (checked) or disabled (left blank) by using the mouse. Boxes filled in with a solid gray color can not be selected. This means that the particular device that is selected does not support the displayed function.

4.12 Security, Lock Bits & Encryption Code

Some microcontrollers have special facilities such as Lock Bits, Security Bits, or Encryption Codes to prevent unauthorized copying of proprietary code. The user can easily operate these functions by selecting "L", "S", or "E" under the main menu, or use the "Auto" function to program the Lock Bits, Security Bits, or Encryption Code. Refer to each individual microcontroller's manual for the definition of these special options as their definition may change slightly from manufacturer to manufacturer and even from type to type from the same manufacturer.

4.13 External YES Key, LEDs and Side Power Switch

The only external key on the programmer module is the "YES" key. It functions exactly the same as the "Y" key on the PC keyboard or as a click on the "OK" button. When a window asks you to click "OK", you may press the "YES" key on the programmer or press the “Y" key on the PC keyboard.

The three LEDs on the programmer module are labeled ON, BUSY, and GOOD.

The ON LED will light up after turning on both the computer power switch and the programmer's side power switch. The BUSY LED will light up during device programming, reading, and verifying. The GOOD LED will light up when the result of an operation is positive.

NOTE

1. The programmer's side power switch must absolutely be turned on before running the programmer file on the hard disk or device driver file on floppy disk.
2. When the programmer is not being used, the user can switch off the side power switch which will turn off the +5V, +12V, & -12V power supplied to the programmer which can help to prolong the life of the equipment.

About PLD (PAL, GAL, PEEL, & EPLD)

The functions used for PLD programming are almost the same as those used to program memory devices and microcontrollers. The following section describes the additional functions.

4.14 Additional or Alternate Functions for PLD Programming

Load JEDEC File:

To program a PLD the user needs to load a JEDEC Fuse Map file instead of a Binary file. This is a standard file format produced by nearly all logic compilers such as ABEL, CUPL, PALASM II, SLICE, PDK-1, PLAN II. Some PLD/CPLD's require proprietary formats which are also supported by the device driver files where appropriate.

Save JEDEC File:

Under PLD Device Driver files, data is saved in JEDEC file format instead of binary format. Files are saved in other formats where are appropriate.

Edit Buffer:

Though the Edit Buffer function is included, we recommend you use it only for viewing. To edit the buffer directly the user must first master the PLD JEDEC Fuse Map assignment. We advise to only edit with a PLD compiler.

Display Buffer, Display Device:

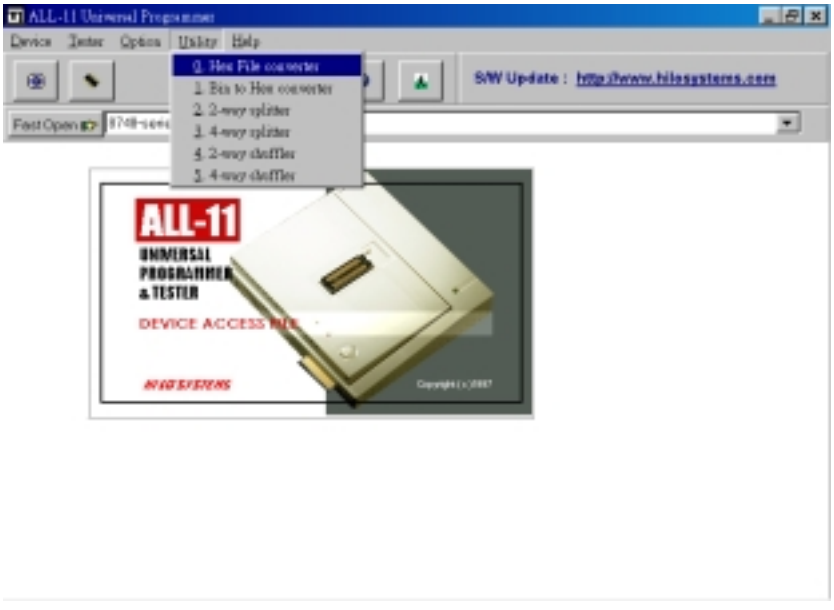
Under PLD Device File, both display functions display files in the JEDEC format.

Security Fuse Blow:
Click "Security" to blow the security fuse of a PLD device. This is the final step in PLD programming. Blowing the security fuses can prevent any further access to the PLD either for "modification" or for "reading". Blowing the fuses prevents anyone from making unauthorized copies of your PLD.

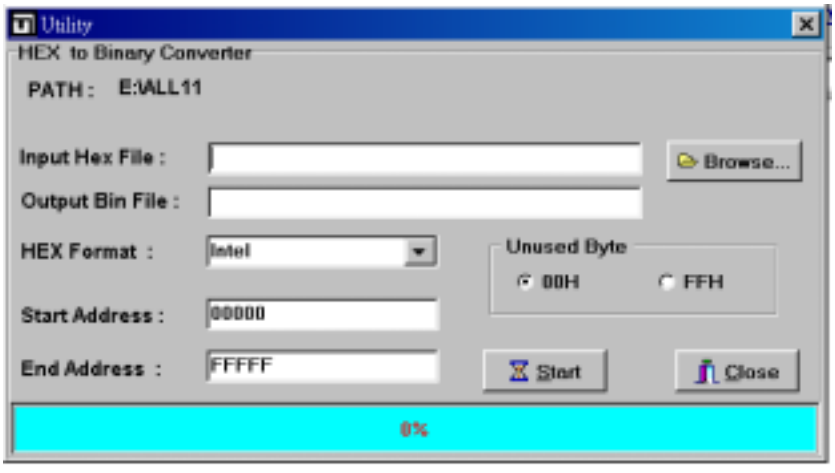
NOTE: After the security fuse is blown the device data can not be read or verified.

5. UTILITIES

Some functions such as Hex to Binary file converter, Binary to Hex File converter, …etc. are provided in the Utility function. They are described as follows:



5.1 Hex File converter --- Hex to Binary file converter



(1) Input HEX File: Type the directory/name of your HEX format file which you want to convert to a Binary file. You can also use the “Browse” button to select the input file.

(2) Output Bin File: Type the directory/name of your output binary file to have it converted and saved to a BIN file.

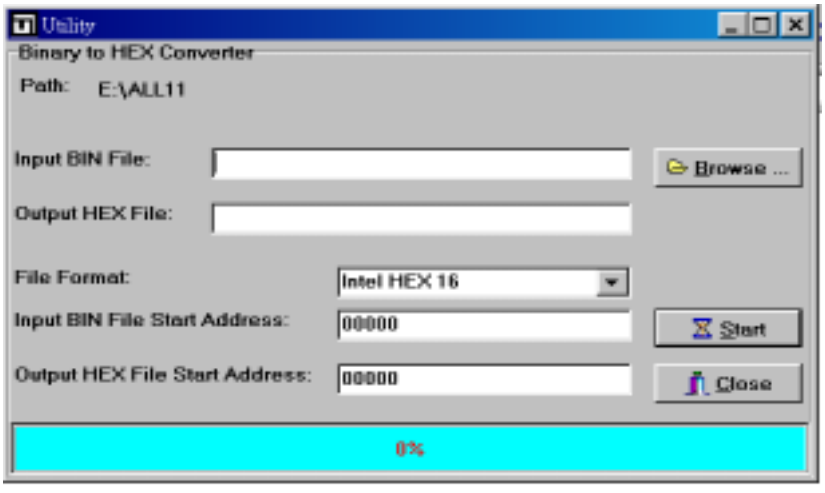
(3) File Format: Select the desired HEX format. You can choose from Intel HEX 16, Intel HEX 32, Motorola HEX, etc.

(4) Start Address: Type the beginning address of the HEX file you want to convert.

(5) End Address: Type the end address of the HEX file you want to convert.

(6) Unused Byte: Some HEX format files may not have consecutive data so you will need to select the unused data (00H or FFH) to fill in the inconsecutive area when you convert the data to Binary data.

5.2 Bin to Hex converter --- Binary to Hex file converter



(1) Input Bin File: Type the directory/name of your binary file which you want to convert to a HEX file. You can also use the “Browse” button to select the input file.

(2) Output HEX File: Type the directory/name of your output HEX file to have it converted and saved to the HEX file.

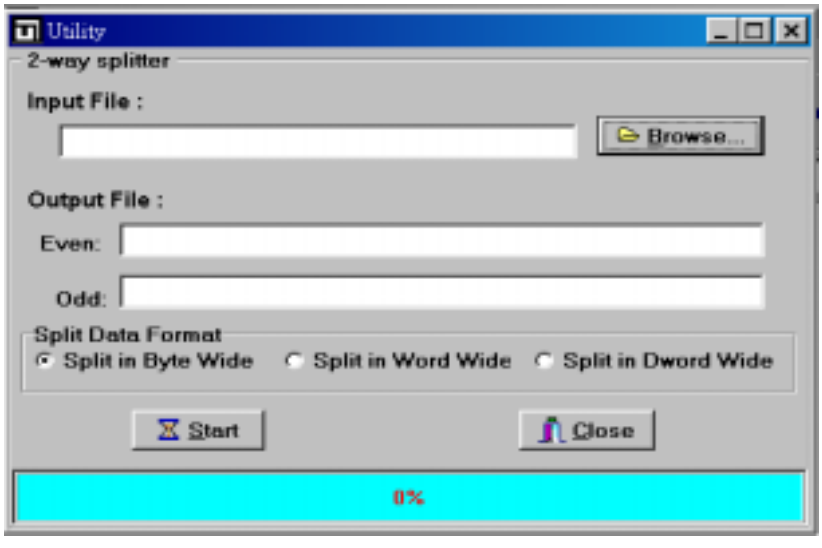
(3) File Format: Select the desired HEX format. You can choose from Intel HEX 16, Intel HEX 32, Motorola S1, Motorola S2, and Motorola S3.

(4) Input BIN File Start Address: Type the beginning address of the BIN file you want to convert.

(5) Output HEX File Start Address: Type the beginning address of the HEX file that you want the BIN data transferred to.

5.3 Two-Way Splitter

This function allows users to split one file into two output files. One file contains odd-byte data from the original file and the other file contains even-byte data from the original file.



(1) Input File: Type the directory/name of the binary file which you want to split. You can also use the "Browse" button to select the Input File.

(2) Output File:

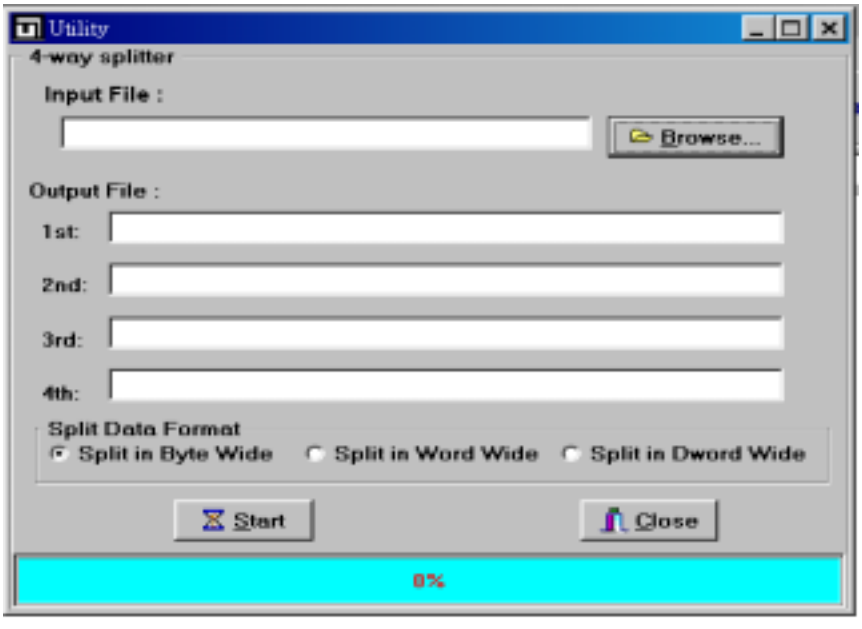
a. Even: Type the directory/name of the Even Data Output File you wish to save the even data to.

b. Odd: Type the directory/name of the Odd Data Output File you wish to save the odd data to.

(3) Split Data Format: Normally the split data is in the Byte Wide mode. That is using one byte as a unit of the basic split data. However users can choose Word Wide (two bytes) or Double Word Wide (four bytes) for a unit of the basic split data.

5.4 Four-Way Splitter

This function allows users to split one file into four output files. The first file contains the first byte of every 4-byte data segment of the original file. The second, third, and fourth files contain the second, third, and fourth bytes of every 4-byte data segment file.



(1) Input File: Type the directory/name of the binary file which you want to split. You can also use the "Browse" button to select the input file.

(2) Output File:

a. 1st: Type the directory/name of the First-Byte Output Data File you wish to save.

b. 2nd: Type the directory/name of the Second-Byte Output Data File you wish to save.

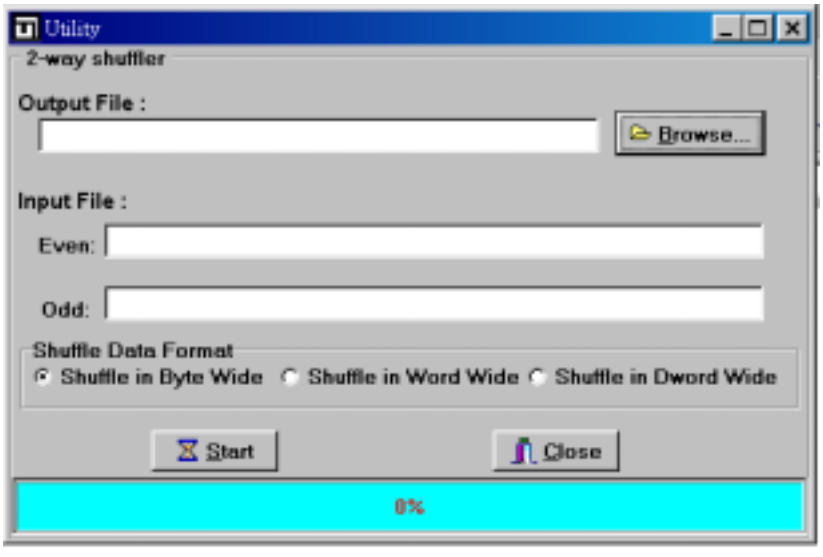
c. 3rd: Type the directory/name of the Third-Byte Output Data File you wish to save.

d. 4th: Type the directory/name of the Fourth-Byte Output Data File you wish to save.

(3) Split Data Format: Normally the split data is in the Byte Wide mode. That is using one byte as a unit of the basic split data. However users can choose Word Wide (two bytes) or Double Word Wide (four bytes) for a unit of the basic split data.

5.5 Two-Way Shuffler

This function allows users to combine two files into one file. There are two input files. One is an odd-byte data file and the other is an even-byte data file. The two files will be combined into one file.



(1) Output File: Type the directory/name of the binary file you wish to save the combined data to. You may also use the "Browse" button to select an output file.

(2) Input File:

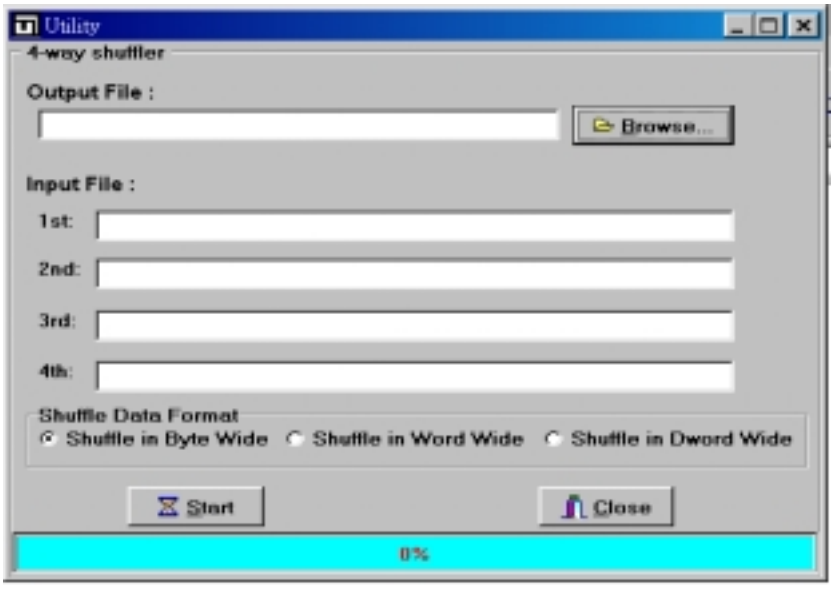
a. Even: Type the directory/name of the Even Data Input File you wish to be combined.

b. Odd: Type the directory/name of the Odd Data Input File you wish to be combined.

(3) Shuffle Data Format: Normally the shuffled data is in the Byte Wide mode. That is using one byte as a unit of the basic shuffled data. However users can choose Word Wide (two bytes) or Double Word Wide (four bytes) for a unit of the basic shuffled data.

5.6 Four-Way Shuffler

This function allows users to combine four files into one file. There are four input files. The first file contains the first byte of every 4-byte data segment. The second, third, and fourth files contain the second, third, and fourth bytes of every 4-byte data segments. The four files will be combined into one file.



(1) Output File: Type the directory/name of the binary file which you want to combine the four files into. You may also use the "Browse" button to select an output file.

(2) Input File:

a. 1st: Type the directory/name of the First-Byte Input Data File you wish to be combined.

b. 2nd: Type the directory/name of the Second-Byte Input Data File you wish to be combined.

c. 3rd: Type the directory/name of the Third-Byte Input Data File you wish to be combined.

d. 4th: Type the directory/name of the Fourth-Byte Input Data File you wish to be combined.

(3) Shuffle Data Format: Normally the shuffled data is in the Byte Wide mode. That is using one byte as a unit of the basic shuffled data. However users can choose Word Wide (two bytes) or Double Word Wide (four bytes) for a unit of the basic shuffled data.

6. JOB FUNCTION:

The Programmer driver contains three special functions under the “File” dropdown menu; Load Programmer Configuration, Save Programmer Configuration, and Enable Job Function.

a) Save Programmer Configuration

This feature allows you to save all settings selected in the software driver, including:

1. Mfr. & Type selections
2. Load File setting
3. Target Zone settings including Device Start, Device End, Buffer Start, & Buffer End
4. Other programming settings
5. Auto function settings

After all settings have been entered and the Auto Function section completed, select the Save Programmer Configuration under the “File” dropdown menu and type the filename you wish to save it under.

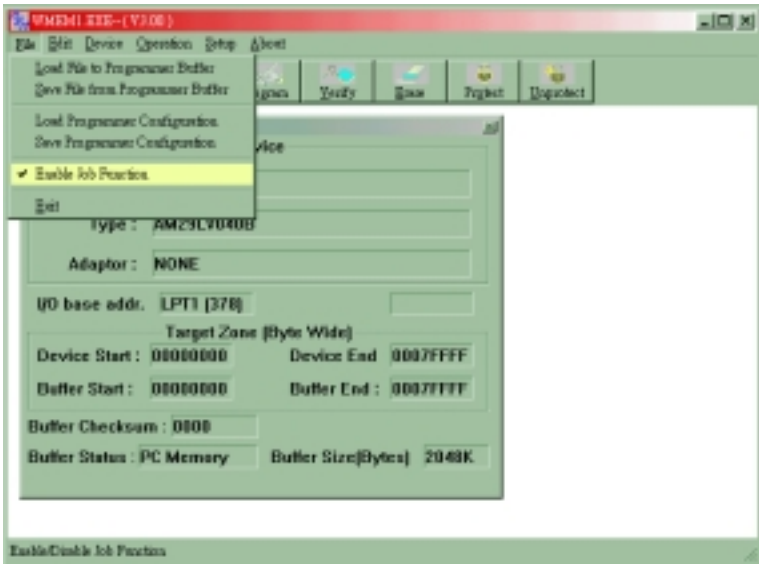
b) Load Programmer Configuration

After saving the configuration information settings using the Save Programmer Configuration command, you may then use the Load Programmer Configuration to load the desired configuration into the software driver. You select the Load Programmer Configuration under the “File” dropdown menu and then choose the filename of the configuration data you wish to use. The software will automatically load the programming data information (note: not all software contains Target Zone data). The “Auto” function will also reinstall all the settings

from the previously programmed device. After the Load Programmer Configuration window has been completed, users only need to select the "Auto" function to begin programming.

c) Enable Job Function

This function allows users to enable or disable the Job Function selection. After the Enable Job Function is set to on by clicking it so that a check mark appears by its side, any previously saved programmer configuration can be easily installed. The next time the program software driver is executed, a file selection window will automatically appear so that you can select the programmer configuration file you wish to use. This function will then automatically open the Auto function window for your convenience. Users may also install a short cut to the programmer driver software in the Startup directory of Windows so that the driver will automatically start when your PC is turned on and Windows is started.



7. SERIAL NUMBER FUNCTION (SERIALIZATION)

When using the Serial Number function on the ALL-11P2, you must be aware of its functions and limitations, which are as follows:

1. Press S/N Setup button in the “AUTO” window to setup the Serial Number function

2. Serial Number On/Off:
This function can be selected to enable or disable the Serial Number function (The Serial Number function is only available in the “AUTO” mode.)

3. Length:
This function sets the length of the serial number. It can be up to 8 bytes (16 digits) long.

4. Display Format:
This function selects the display format for the serial number, either HEX (Hexadecimal) or BCD (Binary Coded Decimal).

5. S/N Start Address:
This function is used to set the starting write location on the ROM.

6. Start Serial No.:
This function is used to set the starting value of the serial number.

7. Direction:

This function is used to specify the sorting order of the serial numbers either from low to high (Low Byte) or high to low (High Byte).

8. After programming the data, the serial number will automatically increase itself.

9. Not all the programmer software can use the Serial Number function (ex: PLD software can not support this function.)

10. The serial number setting for Microchip PIC series devices is different from other devices. Its high byte serial number location's data will be fixed (12 Bits device is 08H. 14 Bits device is 37H, 16 Bits device is B6H).

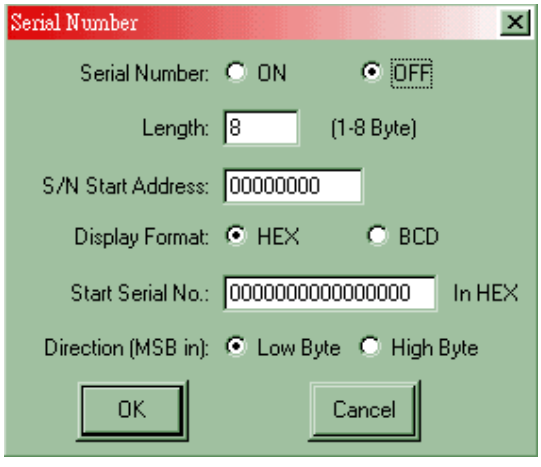
For example, if the user selected serial number is "1234ABCD" then data will be programmed from the beginning of the serial number start address and its data will be formed as follows:

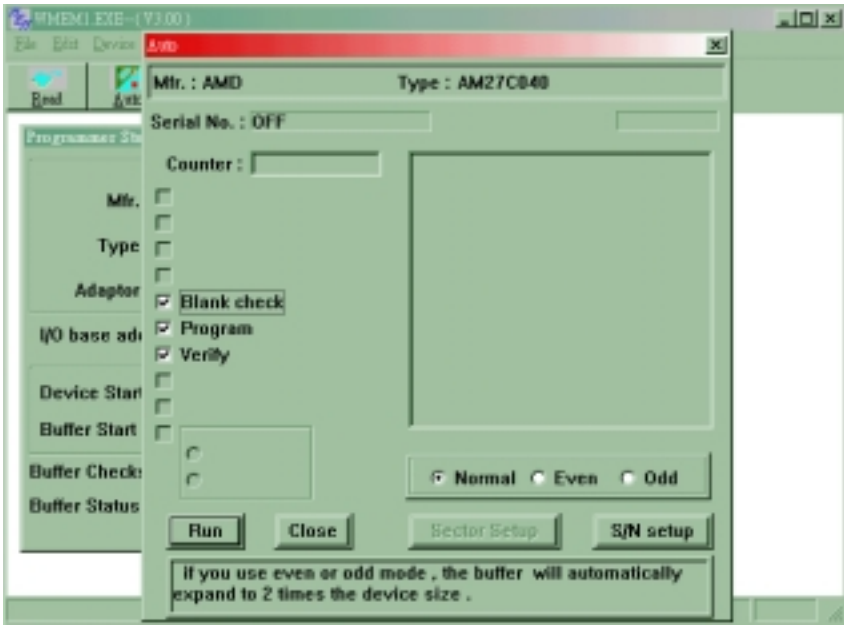
Programmed data to PIC16C54 (12 Bits) is 0812H 0834H 08ABH 08CDH

Programmed data to PIC16C62 (14 Bits) is 3712H 3734H 37ABH 37CDH

Programmed data to PIC17C42 (16 Bits) is B612H B634H B6ABH B6CDH

11. The following is an example of the Serial Number Function:


Serial Number
Serial Number: ON OFF
Length: 8 (1-8 Byte)
S/N Start Address: 00000000
Display Format: HEX BCD
Start Serial No.: 0000000000000000 In HEX
Direction (MSB in): Low Byte High Byte
OK
Cancel


Mfr. : AMD
Type : AM27C040
Serial No. : OFF
Counter :
Blank check
Program
Verify
Normal Even Odd
Run
Close
Sector Setup
S/N setup
If you use even or odd mode , the buffer will automatically expand to 2 times the device size .

APPENDIX

TROUBLESHOOTING

For problems with individual functions please refer to the specific function description under the Function Reference Guide.

PROBLEM 1

When I try to run a driver file, I get communication error messages.

RESOLUTION

1-1 You may not have the serial port or parallel port set correctly for your programmer. Double check the I/O port assignment.

1-2 There may be a bad connection between the RS232 or parallel port and the programmer or you may have a faulty cable. Double check the cable connection.

1-3 You may have the baud rate set too high for your PC's serial port hardware.

1-4 For the parallel port, please make sure PC's BIOS Setup sets the parallel port to ECP, EPP, or Bidirectional mode, not Normal or Unidirectional mode. If you have several usable modes, you can change it to another usable mode and test the unit again.

PROBLEM 2

You have two or more programming failures in a row.

RESOLUTION

2-1 Check with your distributor to make sure you have the latest driver file versions.

2-2 Check if the device is inserted in the ZIF socket correctly.

PROBLEM 3

My operating system is WINDOWS NT. I can use the programmer with the serial port but not with the parallel port. I have set my parallel port in ECP mode.

RESOLUTION

Please note when you install the software in WINDOWS NT4.0 or above, you must be logged on as an Administrator first.

Eight things to do Before Calling Your Dealer:

1. Reboot the computer and try again.

2. Repeat all the steps, following the instructions in this manual.

3. Close all other programs running under windows.

4. See if your problem is listed in the Trouble-Shooting section.

5. Try it on another system.

6. Compare system requirements with your configuration.

7. Ask your in-house specialist (every office has one).

8. Consult the person who installed the product if available.

GENERAL TROUBLESHOOTING CHECKLIST

If your problem is not described above, check the followings:

1. Is the serial port interface card fully seated in its slot?
2. Are all cable connections securely fastened?
3. Is the setting on the RS-232C or parallel port correct?

WEB SUPPORT

If you wish to get updated software, please download the EXE and WPROG.DEV files from our web site, http://www.hilosystems.com.tw. You then have to copy them to the desired directory.

MAUALL-11P2. C.05/2000