WinLV
User's Manual

Software Operating Manual for the Following ICE Technology Programmers:

EPMaster LV
EPMaster LV48
Speedmaster 1000+
Speedmaster LV
Speedmaster LV48
Micrmaster 1000+
Micromaster LV
Micromaster LV48
LV40 Portable
Speedmaster GLV32
Matrix EP4
Matrix EP8
Matrix Universal 4 Way
Matrix Universal 8 Way

1. Introduction

WinLV is ICE Technology's Window's based user interface for the 1000+, LV, LV48 and Matrix range of programmers. It is currently compatible with Windows 95, 98 and NT4.0 with Windows 2000 likely to be supported in early 2001.

This manual explains how to use WinLV to get the most from your programmer and is designed with inexperienced users in mind, with simple instructions and tutorials to help you get programming quickly, as well as explanations of the powerful features available to more advanced users to help streamline production and minimise programming errors.

Most of the manual is written on the assumption that you are using one of the single socket range (1000+, LV & LV48), where there are differences from this, additional sections explain the specific features available on the other programmer types. This applies to the following programmers:

Speedmaster GLV32
Matrix Programming System

We hope that you find this manual a useful aid to getting the most out of your programmer. Publishing this on the web allows us to quickly and easily amend and update the information herein, so if you have any comments or observations on this manual please feel free to email these to us at manuals@icetech.com , suggestions will always be welcome.

2. Installation

2.1 Software

All programmers are shipped with WinLV on a CD. It is also available for download from our website at www.icetech.com - this is also the best way to obtain new versions of WinLV after their release.

In order to install WinLV from the CD you can either:

Insert the disk to your CD-ROM Drive, which will autorun if you have this, set and follow the instructions in the browser window that opens.

Or

Explore the CD and run setup from the directory "d:\install\".

If updating WinLV from a downloaded "Icetech.exe" file from the ICE Technology web site, you should copy the download to a temporary directory and run this there. This will expand into a number of smaller files, running "setup.exe" will then install WinLV.

If you have previously installed WinLV, you will be reminded that you are overwriting existing files and asked if you wish to continue, it is recommended that you select the "yes to all" option at this point to ensure a full, clean installation of WinLV.

NB. If you are installing onto a Windows NT4.0 PC, it is vital that you are logged on as an Administrator – or the parallel port driver will not be installed properly!

2.2 Connecting The Programmer

Make sure that your PC has the following minimum requirements otherwise you may experience problems using WinLV and the programmer:-

- IBM PC Compatible – 80386DX, 80486DX, Pentium or higher
- 16 MB RAM
- Approx 8.0 MB Free hard disk space
- Microsoft Windows 95/98 or Windows NT4.0 Operating System
- ECP Parallel Port – with this selection set in the PC BIOS.
- Standard serial/PS2 mouse

Check the contents of your ICE Technology programmer package. Make sure that you have the following:-

1x Device Programmer
1x Copy of programmer software

1x Power supply (UK, Euro, USA or Japanese)
1x Parallel cable (25-pin D Type)
1x User Manual

Connect one end of the parallel cable to the programmer and the other into one of the parallel ports on the PC. Tighten the holding screws to ensure the cable is properly attached to the LPT port

Connect the power supply provided into the back of the programmer and plug into any mains socket.

NOTE:- The single socket programmers do not have a power on switch. It will power up (the red power LED will light up) automatically when it receives a signal from the PC and then power down when not in use, i.e. when the WinLV software is closed.

Run WinLV in Windows 95

Run the “seltest” utility in WinLV (you’ll find it under “programmer” then “diagnostics”) to confirm that your programmer has been connected correctly and the WinLV software is communicating with the programmer.
If Selftest does not return any errors then you can start using your programmer and WinLV software. If you are experiencing any problems contact Technical Support at ICE Technology.

2.3 Connecting a Matrix Programmer

There are three connections available on the back of the Matrix as shown below:

The power connection is of the Universal IEC type. This is a universal input – the onboard power supply can accept any of the standard international Mains AC Voltages or frequencies.

The black D25 parallel connection is the IN port. When using a single Matrix, or if this is the first unit in a daisy-chained set of programmers, this connects directly to the LPT parallel connector on the PC controlling the programmer

via the D25 parallel cable provided. When daisy-chaining multiple units this connects to the out port of the previous programmer.

The blue D25 parallel connector is the OUT port – this will always be connected to the next programmer's IN port in a daisy chaining arrangement, or not at all in the case of a single Matrix.

2.4 Installation FAQs

Although the programmer and software have been designed to work with all PCs and associated hardware, from 286 to Pentium III, there are occasions when the programmer will not work straight away. Should you encounter any problems, please just follow the steps below and hopefully these will resolve any issue you encounter:

Ensure that you are running the very latest software. If you have purchased the programmer through one of our distributors, it is quite possible that the programmer has been held in stock for a period of time. The software version number is printed on the disks. To check if this has been superseded, please check our web site on the Internet, at www.icetech.com. This will show the latest software version number and release date, plus a full list of associated updates, changes and amendments.

Currently there is no support for UNIX or LINUX Operating systems. However, LINUX drivers are under development, and will be available at a later date. Other OS platforms are also being considered. Details of these will be available on our web site.

These Frequently Asked Questions cover almost all problems encountered during installation

Q1. Programmer power is connected, but the power light is not lit.
A1. *The power to the programmer is controlled by the software. This means that power is not wasted if the programmer is being run off batteries. The programmer will also power down automatically during periods of inactivity. Try running the Selftest, or running WinLV, EPROM or PAL. The light should come on as the programmer is initialised.*

Q2. When installing the software in Windows 95/98, the message "*system file already in use*" is reported, and the installation quits.
A2. *This is caused by one of two system files (MFC42.DLL or MSVCRT.DLL) appearing to the set-up file that they are in use, or even newer than that being installed. We provide the latest versions of .these files as released on the Microsoft Developer Network collection. To correct this, restart your PC in MS-DOS mode. Go into the windows system subdirectory (CD C:\WINDOWS\SYSTEM). Rename the two files as MFC42.OLD and MSVCRT.OLD. Restart windows, then install the software. There will then be no problems with installation.*

Q3. When I am in the WinLV software, the READ light is greyed out in the operations menu, and I cannot edit the buffer contents.
A3. *This feature is to protect the contents of the buffer from being modified by accident. To enable READ operations, select FILE/ENABLE EDIT MODE.*

Q4. When running the software (Selftest, WinLV, EPROM or PAL), random communications problems are reported. Printers, and other devices connected to the port have no problems.
A4. *First, check that the programmer is connected properly to the parallel port. Remove any dongles to see if this may be the cause of the problem. Also, check our web page to see if you have the latest software version. The software release notes will indicate if there have been comms problems with any specific device. The programmer works best when connected to an ECP port. On newer PC's, this is selected in the BIOS. Note that on Laptops with an Ir port, selecting IrDA 1.1 could cause ECP mode to become unavailable. Select ECP,* ***NOT*** *ECP/EPP. We have also seen that corrupt .INI files may cause problems. Delete these, and try running the software again. Also, try the programmer on another PC, to see if the programmer, or the PC are the problem, or possibly try separate I/O card if the problem persists.*

Q5. When running WinLV, the software hangs at the splash screen, or when running PAL or EPROM in DOS, the PC crashes. Printers, and other devices connected to the port have no problem.
A5. *Check that all the items in* ***A4*** *have been adhered to. Also, we have seen that adding C:\ICETECH to the PATH statement in the AUTOEXEC.BAT file can resolve this problem, and also ensure that C:\WINDOWS; C:\WINDOWS\SYSTEM and C:\WINDOWS\COMMAND are also in the PATH statement.*

NOTE: In addition to A4 and A5: For certain branded PC's and Notebooks, separate software utilities are available which may solve the port problems. These are required as certain branded machines do not use standard ChipSets for their parallel ports, and also include power saving and low voltage features on the I/O ports. Check their web pages for the utilities:

Compaq: http://www.compaq.com/support/files/index.html
Dell: http://www.dell.com/support/index.htm
Gateway: http://www.gateway.com/support/product/drivers/index.html

If you are still having problems, just call our Technical Support team, who will be happy to help:
tel: +44 (0)1226 767404 fax: +44(0)1226 370434 email:
support@icetech.com

2.5 Easy Start Tutorials

The following tutorials are designed to guide you step by step through the most fundamental programmer operations, to get you programming from scratch as quickly as possible with minimal reference to the rest of this manual.

2.5.1 Tutorial 1: Programming an EPROM from file

This first tutorial is designed to get you operating your programmer and working with WinLV as quickly as possible. It explains, step-by-step, the procedure for programming an EPROM or microcontroller using an existing file. Follow these steps very carefully and soon you will become familiar with the basic principles of the WinLV software and then be able to implement some of the more advanced programming features.

Step 1: Run WinLV

Select the WinLV.EXE file using the Windows 95 Explorer or, alternatively, create a shortcut on the Windows 95 Desktop.

Step 2: Select Device

Choose the device that you are using by selecting it from the ***Select Device*** window. To open the ***Select Device*** window – click the left hand mouse button on ***Programmer*** from the top menu bar then click on ***Select Device***. Use the mouse to browse through the Parts Database and select a device from the Programmable Parts section then click on Accept to confirm your choice.

Step 3: Load File into Buffer

Select the file you would like to programme into the selected EPROM or Micro following these steps:-

- Select the ***File*** menu and then ***Open…***
- Select which file you want to load into the buffer OR type the file name Into the ***File Name*** box
- If any files that you expecting to see aren't there try using the ***Files of Type*** drop down menu to select which file type is displayed in the main window
- Use the ***Open as*** drop down menu to select which Format the file will be opened as and loaded into the buffer.
- Set the Defaults to pre fill the buffer with 0xFF so that all empty address locations are filled with FF.
- Remember - If you want to specify more advanced file open settings this can be set in the lower half of the ***Open*** dialogue window.

- Remember - If the buffer size is not large enough for the file selected then it will truncate the file before loading. To increase the buffer size select ***New Buffer*** from the ***File*** menu and use the slider bar to increase the buffer size and then re-load the file.

Step 4: Place Device in Programmer

Place the device you've selected to programme into the socket on the programmer. Make sure that the device is aligned with the bottom of the ZIF socket unless instructed to do otherwise, e.g.:- on The MMLV and the MM1000+ the Microchip PIC12C508/509 microcontroller must be aligned with the top of the ZIF socket.

Step 5: Programme the Device

Follow these final steps to finish programming the device:

- Select ***Operations*** from the ***Programmer*** menu to view the “Operations for Device” dialogue window.
- Click on the ***Programme*** button in the EPROM ***Operations*** dialogue window.
- The device in the socket will now be erased, programmed and verified.

2.5.2 Tutorial 2: Programming an EPROM from Master Device

This second quickstart tutorial explains how to read data from a master device into the buffer and then programme another device with same data. This particular example assumes that the device being read and the other device being programmed are the same. It is possible, of course, to read data from one EPROM/Flash device and programme it into another similar EPROM/Flash device that isn't exactly the same. To do this you must re-select the device type from the Parts Database between reading and programming.

Follow these steps very carefully and they will give you some idea of the basic Read & programme operations that WinLV is capable of. When you feel comfortable with reading and programming devices using basic WinLV operations experiment with some of the more advanced options.

Step 1: Run WinLV

Select the WinLV.EXE file using the Windows 95 Explorer or, alternatively, create a Shortcut on the Windows 95 Desktop.

Step 2: Select Device

Choose the device that you are using by selecting it from the ***Select Device*** window. To open the ***Select Device*** window – click the left hand mouse button on ***Programmer*** from the top menu bar then click on ***Select Device***. Use the mouse to browse through the Parts Database and select a device from the Programmable Parts section then click on Accept to confirm your choice.

Step 3: Place Master Device in Programmer

Place the device you've selected to programme into the socket on the programmer. Make sure that the device is aligned with the bottom of the ZIF socket unless instructed to do otherwise.

Step 4: Read Master Device

Follow these steps to read data from the master device:-

- Select ***Operations*** from the ***Programmer*** menu to view the Operations For dialogue window.
- Click on the ***Read*** button in the EPROM ***Operations*** dialogue window.
- The device in the socket will now be read and all the data will be loaded into the buffer and displayed on the screen.

At this point you can save the data in the buffer by selecting ***Save*** or ***Save Project*** from the File Menu and also change the device selection in the Parts

Database if you wish to programme a different device with the file you have just read.

Step 5: Programme the Blank Device

Follow these final steps to programme the blank device with the data from the master device:-

- Select ***Operations*** from the ***Programmer*** menu to view the Operations For Device dialogue window.
- Click on the ***Programme*** button in the EPROM ***Operations*** dialogue window.
- The device in the socket will now be programmed and verified.

2.5.3 Tutorial 3: Programming a PAL from file

This third tutorial is designed to get you operating your programmer and working with WinLV as quickly as possible. It explains, step-by-step, the procedure for programming a logic device using an existing file. Follow these steps very carefully and soon you will become familiar with the basic principles of the WinLV software and then be able to implement some of the more advanced programming features.

Step 1: Run WinLV

Select the WinLV.EXE file using the Windows 95 Explorer or, alternatively, create a Shortcut on the Windows 95 Desktop.

Step 2: Select Device

Choose the device that you are using by selecting it from the ***Select Device*** window. To open the ***Select Device*** window – click the left hand mouse button on ***Programmer*** from the top menu bar then click on ***Select Device***. Use the mouse to browse through the Parts Database and select a device from the Programmable Parts section then click on Accept to confirm your choice.

Step 3: Load File into Buffer (Fuse Map)

Select the file you would like to programme into the selected PAL, PALCE, EPLD etc. following these steps:-

- Select the ***File*** menu and then ***Open…***
- Select which file you want to load into the buffer OR type the file name into the ***File Name*** box. The buffer will automatically change into Jedec Mode and display the buffer as a Fuse Map.
- If any files that you expecting to see aren't there try using the ***Files of Type*** drop down menu to select which file type is displayed in the main window
- Use the ***Open as*** drop down menu to select which format the file will be opened as and loaded into the buffer.
- Remember - If the buffer size is not large enough for the file selected then it will truncate the file before loading. To increase the buffer size select ***New Buffer*** from the ***File*** menu and use the slider bar to increase the number of fuses in the fuse map and then re-load the file.

Step 4: Place Device in Programmer

Place the device you've selected to programme into the socket on the programmer. Make sure that the device is aligned with the bottom of the ZIF socket unless instructed to do otherwise.

Step 5: Programme the Device

Follow these final steps to finish programming the device:-

- Select ***Operations*** from the ***Programmer*** menu to view the Operations For Device dialogue window.
- Click on the ***Programme*** button in the PAL ***Operations*** dialogue window.
- The device in the ZIF Socket will now be erased, programmed and verified.

2.6 Operating from Batteries

The LV48, LV and 1000+ programmers can be operated from mains or batteries. To operate from batteries use 8 alkaline AA type cells. The battery cover is held in place with screws. When replacing the cover, take care to use only the screws provided.

WARNING : USE ONLY THE SCREWS PROVIDED TO FASTEN THE BATTERY COVER, & DO NOT OVER TIGHTEN THE SCREWS.

NOTE: Certain Bipolar devices require high currents in very short bursts and therefore some batteries may not always be able to handle these devices.

2.6.1 Recharging the Batteries

The programmer can be used as a battery charger. In order to start recharging the batteries the programmer needs to be connected to a PC and connected to the mains using the mains adapter provided.

Please Note: This utility is ONLY available in DOS

To charge the batteries, the command is: `BCHARGER`

usage: `BCHARGER [port] CHARGEMODE [REPORTMODE]`
where:

`port`	parallel port to use (default = 1)
`CHARGEMODE`	`TRICKLE/SLOW/FAST` (default=slow)
`REPORTMODE`	`START/REPORT` (default=both)
`START`	starts the charging and does immediate return
`REPORT`	reports current stage of charging

When the batteries are charging the ‘BUSY’ LED on the programmer will flash red and yellow. Once charging is complete this LED will switch off.

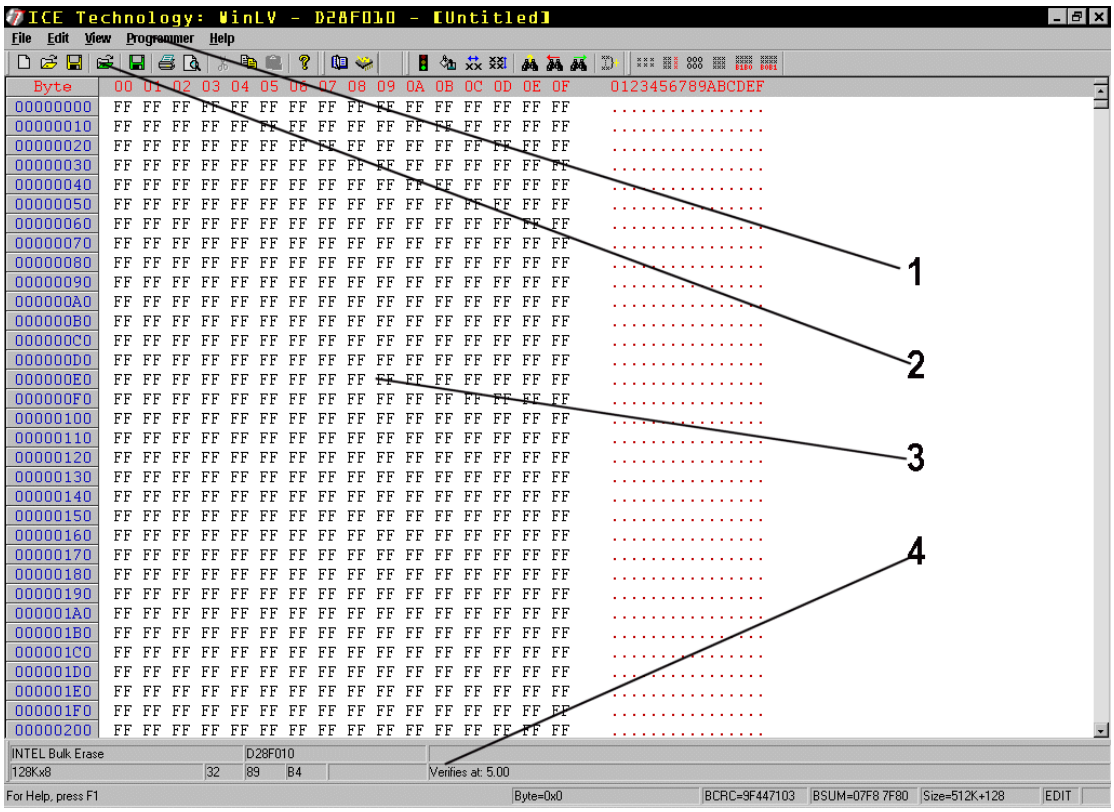
Once recharging has commenced the programmer can be disconnected from the PC. Charging will stop once the batteries have been fully charged. If the PC is switched off battery charging will be interrupted for a few seconds. Recharging will resume automatically.

WARNING : DO NOT ATTEMPT TO CHARGE NON-RECHARGEABLE BATTERIES

3. WinLV Screen Arrangement

The WinLV software provides you with an easy-to-use, menu-driven, front-end for ICE Technology's range of Single Socket Multi-Device Programmers and our Speedmaster GLV-32 Gang Programmer. WinLV offers you a comprehensive range of file handling & editing utilities, data manipulation, ROM/RAM Emulator Options, Chiptest Options and device programming functions all incorporated in one "familiar" looking Windows 95 application.

The basic on-screen elements of the main WinLV window are as follows:



1. WinLV Menu Options

These allow quick access to WinLV's functions and are logically grouped according to their function. Use the mouse or keyboard to select the menus and menu commands. The specific functions available are described in detail in section. To summarise, the main menus are as follows:

File Menu

New Buffer	Opens a window allowing the adjustment of the buffer size
Open	Opens "Open File" Window
Save	Save the file under the present settings/file name
Save As	Saves the file with changed settings or file name
Open Project	Open a previously saved project (*.ice) file
Save Project	Save the current file/device parameters in a project (*.ice) file
Print	Print the current buffer contents
Print Preview	Preview the print of the current buffer contents
Print Setup	Access the printing setup for the current file

Recent Files	Open a list of recently used data files for quick access
Recent Projects	Open a list of recently used project files for quick access
Enable View Only Mode	Limit users access to the data buffer – useful for production
Exit	Exit WinLV

Edit Menu

Copy	Copy the present selection onto the clipboard
Paste	Paste the present clipboard contents into the current location
Goto	Opens a window to define a move to a specific buffer location
Find	Find a specific value in the data buffer
Find Next	Find the next buffer address containing that value
Find Previous	Find the previous buffer address containing that value
Fill	Opens a window to specify a range of address and a fill value
Swap Bytes	Swap
Swap Words	Swap
Checksum Calculator	Opens a window for user to define checksum parameters

View Menu

Toolbar	Show standard toolbar in main window
Operations Toolbar	Tick to have the Programmer Operations window displayed
Edit Toolbar	Tick to have the Edit Toolbar displayed
Display Mode Panel	Tick to have the Display Mode toolbar shown
Information Panel	Tick to have the device information toolbar shown
Split	Splits the buffer window to show two address ranges at once
Synchronise Split	Movements within one split window are reflected in the other
Colours	Opens window for definition of the buffer's colour scheme
Nibble Mode	Explained in section 3.1.3
Byte Mode	" "
12 Bit Mode	" "
Word Mode	" "
Word Mode	" "
8 Bit Octal Mode	" "
Jedec Mode	" "
Vector Mode	" "
Waveform Mode	" "

Programmer Menu

Select Device	Open Device Database/Selection Window
Operation	Open Programmer Operations Window
Match 74/4000 device	Test device for match with 74/4000 series
Diagnostics	
Selftest	Run the programmer selftest programme
Emulator Test*	Run the emulator selftest programme

Help Menu

Help Topics	Open the WinLV help index window
About WinLV	Open the about WinLV window

2. WinLV Toolbars

The toolbars & display panels allow you to view current WinLV information and settings as well as allowing quick access to specific functions as an alternative to using the WinLV menus. Which toolbars are displayed can be defined by the user via the "View" menu, they can also be be dragged and dropped to any position either in the main WinLV window or within the borders on any side of that window.

The Toolbars and Panels are:-

- General Toolbar
- Status Toolbar
- Edit Toolbar
- Display Mode Toolbar

A full description of the functionality of each toolbar can be found in section

3. Programming Buffer Window

The majority of the screen is taken up with displaying the programming buffer. The buffer can be viewed differently by selecting any of the Buffer View options. The buffer can also be edited so that data can be changed before or after it has been programmed into a device.

4. WinLV Information Panel

This Displays information about the device currently selected such as it's type, name, memory size, verify voltages and what adapter to use if it is not a DIP package.

3.1 WinLV Menus

3.1.1 File Menu

The WinLV File Menu contains the standard Windows commands for opening, saving & printing files.

New Buffer (Ctrl+N)

This allows you to open and specify the size of the programming buffer that WinLV is using. The minimum buffer size is 256K and the maximum is 128M. To alter the buffer size, use the mouse or keyboard to move the buffer size selector to increase or decrease the amount of buffer memory.

If you are programming a logic device, e.g. PAL, GAL or EPLD etc. the ***Buffer Size*** window allows you to specify how many Jedec fuses are in the buffer. To alter the number of fuses use the mouse or keyboard to move the Jedec fuses selector to increase or decrease the size of the fuse map.

Open… (Ctrl+O)

This allows you to open previously saved files into the buffer. The ***Open*** dialogue window is set out using the normal, familiar, Windows 95 arrangement.

To open a file:

Select the ***File*** menu and then ***Open…***

Use the mouse or keyboard to select which file you want to load into the buffer. OR type the file name into the ***File Name*** box

Use the ***Files of Type*** drop down menu to select which file type is displayed in the main window

Use the ***Open as*** drop down menu to select which format the file will be opened as and loaded into the buffer.

Set the Defaults to pre fill the buffer with 0xFF if you are opening a new file. You can change the value used to fill the buffer and the length of each value in the ***Set Defaults*** section.

The lower section of the ***Open*** dialogue window also allows you to select other advanced file loading options (Not available when opening *.JED files for logic devices):

The ***Load Every*** option allows you to select if every byte, every 2nd byte, every word, every 2nd word etc. is loaded from the file into the buffer.

Between address allows the lower and upper address to be set for reading a file into the buffer.

Store into every selects if you want to load the file data into every byte, every 2nd byte etc.

Starting at buffer address allows you to specify an address in the buffer where the first address from the file will be loaded.

Swap Bytes (within words) and ***Swap Words (within long)*** allow the MSB and LSB to be swapped when loading a file into the buffer.

Save (Ctrl+S)

Allows you to save currently open files from the buffer in any file format. Use the mouse or keyboard to select a file name and file type.

Save As…

Allows you to save a currently opened file in the buffer with a different file name. The ***Save As…*** dialogue window looks much like the ***Save*** and ***Open*** windows and operates in an identical way.

Open Project

Allows you to open a previously saved project file with a *.ICE extension and load all the information stored in the project into WinLV.

Save Project

Allows you to save all of the currently selected WinLV options into one project file (*.ICE)

Print (Ctrl+P)

Allows you to print the currently open buffer to either a local or network printer. Selecting ***Print..*** will open a dialogue window in which you can select:-

Which printer (local or network) and printer properties.

Print range, i.e.: specify a range of pages to be printed by giving the start page number and finish page number.

How many copies

The printed buffer will display addresses and the hexadecimal data stored in the addresses, or if the buffer is set up for a logic device, it will display fuse numbers and the fuse map.

Print Preview

Allows you to preview the currently open buffer before it is printed. The ***Print Preview*** menu option displays the pages of the current buffer as they will appear after they have been printed.

Print Setup…

Allows you to select a default printer for WinLV to use and also define the current printer properties. Other printer options, such as paper size and orientation, can be selected using ***Print Setup…***.

Recent Files

Allows you to re-open a previously saved file. The last, most recent, file names that you have been using with WinLV will be stored here.

Recent Projects

Allows you to re-open a previously saved project. The last, most recent, project file names that you have been using with WinLV will be stored here.

Enable View Mode Only / Enable Edit Mode

This option allows you to toggle between editing and viewing the buffer. The ***View Mode Only*** will prompt you for a password that must be entered when you select to go back into ***Edit Mode***. If you do not wish to enter a password to return to ***Edit Mode*** then click on ***OK*** twice in the password dialogue window and no password will be saved.

In ***View Only Mode*** the user will not be allowed to perform operations or change parameters that affect the contents of the buffer – such as reading, editing or serial numbering. The main window will be hidden in this mode, with only the ***Programmer Operations*** window open.

The setting of this mode can be saved within a Project File, which can be opened by a user by just double-clicking on the project icon. In this case all other programming parameters are also saved so once the project is opened, the programming can commence with no input from the operator other than clicking the “Programme” button or, in hands free mode, replacing the devices.

Exit

Select ***Exit*** to close the WinLV software and power down your programmer.

3.1.2 Edit Menu

The WinLV Edit Menu contains commands for easily manipulating the data displayed in the programming buffer. It Includes utilities for finding specific data in the buffer, as well as byte/word swap and buffer fill and others.

Remember - The functions available in the Edit Menu will sometimes change depending on whether you are programming a device, using the Chiptest Utility or the ROM/RAM Emulator.

Copy (Ctrl+C)

This menu option allows highlighted sections of the programming buffer to be copied to a different location (address) in the buffer.

To copy a selection of addresses from one location in the buffer to another:-

Move the mouse pointer to the start address of selection.

Hold down the left mouse button whilst dragging the pointer across the buffer to include all address to be copied.

Release the left mouse button when the mouse pointer is above the last address in the selection to be copied.

Without touching the mouse buttons or pressing any other keys, select ***Copy*** from the ***Edit*** menu.

Move the mouse pointer and click once on the start address in the buffer where you want to copy the highlighted selection.

Select ***Paste*** from the ***Edit*** menu and the highlighted selection will be pasted into the new location in the buffer.

Paste (Ctrl+V)

This allows you to paste a copied, highlighted selection from one location in the buffer to another. (See ***Copy***)

Goto (Ctrl+G)

This allows you to locate and go to a specific address in the buffer. Enter the hexadecimal byte or word address in the ***Goto*** dialogue window and the buffer view will change to display the new location.

Find (Alt+F3)

This allows you to find a specific hexadecimal, octal or ASCII value in the currently open buffer. Select one the hexadecimal, octal or ASCII radio buttons and enter the value to search for in the ***Find String*** box.

Find next (F3)

After you have specified a search string using the ***Find*** menu option use ***find next*** to search through the buffer and locate the next occurrence of the search string.

Find previous (Shift+F3)

After you have specified a search string using the ***Find*** menu option use ***find previous*** to search back through the buffer, from currently selected address, and locate the previous occurrence of the search string.

Fill (Ctrl+F)

Allows you to fill the currently open buffer with a specified hexadecimal value. The ***Fill*** dialogue window allows you to enter a value (as a Byte or Word) and the start and end address within which all addresses will be filled with the selected value. The ***Fill*** default start and end addresses are set to the size of the currently open buffer.

Swap Bytes

Allows you to swap the MSB and LSB within any word in the currently open buffer. ***The Swap Bytes within Word*** dialogue window allows you to select the start and end addresses of an area of the buffer within which all word bytes will be swapped.

Swap Words

Allows you to swap the Most Significant Word and Least Significant Word within any Long value in the currently open buffer. ***The Swap Words within Long*** dialogue window allows you to select the start and end addresses of an area of the buffer within which all words will be swapped.

Checksum

Allows you to calculate a buffer checksum (using the checksum calculator) or a device checksum. The checksum calculator can calculate a checksum for a user-defined range of addresses in the buffer, or for the complete buffer. Checksums calculated are:

The sum of unsigned bytes with the carry being added to the MSB

1’s Complement (Inverse of the above checksum)
CRC – Cyclic Redundancy Check

3.1.3 View Menu

The WinLV View Menu contains commands for hiding/showing toolbars and changing the view mode of the buffer.

Toolbar

Allows you to show or hide the General Toolbar on the screen by clicking the left mouse button. The checkmark indicates that the toolbar is active and not hidden. To show a hidden toolbar, select the relevant toolbar from the ***View*** menu and click once with left mouse button.

Operations Toolbar

Allows you to show or hide the Programmer Operations Toolbar on the screen by clicking the left mouse button. The checkmark indicates that the toolbar is active and not hidden. To show a hidden toolbar, select the relevant toolbar from the ***View*** menu and click once with left mouse button.

Edit Toolbar

Allows you to show or hide the Edit Toolbar on the screen by clicking the left mouse button. The checkmark indicates that the toolbar is active and not hidden. To show a hidden toolbar, select the relevant toolbar from the ***View*** menu and click once with left mouse button.

Display mode panel

Allows you to show or hide the Display Mode Panelon the screen by clicking the left mouse button. The checkmark indicates that the display panel is active and not hidden. To show a hidden display panel, select the relevant display panel from the ***View*** menu and click once with left mouse button.

Information panel

Allows you to show or hide the Information Panel on the screen by clicking the left mouse button. The checkmark indicates that the display panel is active and not hidden. To show a hidden display panel, select the relevant display panel from the ***View*** menu and click once with left mouse button.

Split

Allows you to view the currently open buffer using two separate windows. Each window can display a different area in the same buffer.

To split the screen:

Select ***Split*** from the ***View*** menu.

Drag the mouse point to position on the screen where you would like the split and press the left mouse button.

Edit / view the two areas of the same buffer independently.

Close the split by moving the mouse pointer over the bar between the two buffer windows, hold down the left mouse button, and dragging the bar up to the top of the buffer window. Alternatively, you can move the mouse pointer over the bar between the two split screens and double-click.

Synchronize split

If you are using the ***Split*** option to view the buffer using two separate windows, the ***Synchronize split*** menu option will allow you to view / edit one of the windows in a different view mode, with any changes to the buffer being displayed in both split windows. The checkmark indicates that the synchronize split option is enabled.

Colours

Allows you to change the default colour settings for WinLV.

Nibble Mode

Only applicable when programming EPROMs and Flash memory etc. ***Nibble Mode*** displays the current buffer using 16 nibbles per line. The nibbles are arranged so that the LSN is on the left hand side and the MSN is on the right hand side of each pair of nibbles.

Byte Mode

Only applicable when programming EPROMs and Flash memory etc. ***Byte Mode*** displays the current buffer using 2 bytes (16 bits) on one line. Therefore, buffer addresses down the left hand side start at 00h and increase in 16 bit intervals, e.i:- 00h 10h 20h 30h 40h…F0h etc. The ASCII equivalent

characters at each address in the buffer are displayed down the right hand side of the buffer.

---

12 Bit Mode

Only applicable when programming EPROMs and Flash memory etc. ***12 Bit Mode*** displays the current buffer using 12 bits (3 nibbles) at buffer address, with 8 address on one line. Therefore, buffer addresses down the left hand side start at 00h and increase in 8 bit intervals, i.e.: 00h 08h 10h 18h 20h…F8h etc.

Word Mode (Even:Odd)

Only applicable when programming EPROMs and Flash memory etc. ***Word Mode (Even:Odd)*** displays the current buffer using 8 2-byte words on each line. Each word is displayed with the even byte on the left-hand side and the odd byte on the right-hand side. Buffer addresses down the left hand side start at 00h and increase in 8 bit intervals, e.i:- 00h 08h 10h 18h 20h…F8h etc.

Word Mode (Odd:Even)

Only applicable when programming EPROMs and Flash memory etc. ***Word Mode (Odd:Even)*** displays the current buffer using 8 2-byte words on each line. Each word is displayed with the odd byte on the left-hand side and the even byte on the right-hand side. Buffer addresses down the left hand side start at 00h and increase in 8 bit intervals, ie. 00h 08h 10h 18h 20h…F8h etc.

---

8 Bit Octal Mode

Only applicable when programming EPROMs and Flash memory etc. ***Octal Mode*** displays the current buffer using buffer addresses and data in octal.

Jedec Mode

Only applicable when programming logic devices e.g.: PALs , GALs etc. ***Jedec Mode*** displays the current buffer as a Jedec compatible fuse map

---

Vector Mode

Only applicable when programming logic devices e.g.: PALs , GALsetc. Selecting ***Vector Mode*** allows you to view and edit the test vectors applicable to the currently selected logic device.

Waveform Mode

Only applicable when programming logic devices e.g.: PALsGALsetc. ***Waveform mode*** displays the currently open fuse map showing the logic waveforms generated at each pin.

To edit the waveforms at each pin:

Select ***Waveform Mode*** from the ***View Menu***

Select a pin number from the left hand side (move the cursor using the mouse)

Use the following keys to edit the wave forms at each pin:

0	Forces a logic zero onto pin
1	Forces a logic one onto pin
L	Expect a logic zero on pin
H	Expect a logic one on pin
C	Clock. Apply all other signals with C=0 and apply C=1 followed by C=0
X	Don’t care
Z	Expect high impedance
N	Power pin (Vcc or GND). Sometimes used as don’t care but the test vector loader will issue a warning and convert all non power pins to X.

3.1.4 Programmer Menu

The WinLV Programmer contains commands for controlling the operations of the programmer, selecting devices and running diagnostic selftests.

Remember:- If you are using the Speedmaster GLV-32 gang programmer with WinLV then the options in the ***Programmer Menu*** will change slightly.

Select Device

The ***Select Device*** menu item allows you to select which device you are currently using with your programmer. All physical devices including PALs, EPROMs & flash etc. can be selected from the ***Select Device*** dialogue window, as well as emulated devices and chip test parts.

Operations

Selecting the ***Operations*** menu item will open the ***Programmer Operations*** window so that you can control exactly what the programmer is doing. There are five different types of ***Programmer Operations*** depending on which type of device you have currently selected to use from the Parts Database and which programmer is connected to your PC, they are:-

Programmer Operations for EPROMs/Flash/Microcontrollers
Programmer Operations for PALS/GALs/EPLDs
Programmer Operations for ROM/RAM Emulator
Programmer Operations for Chiptest
Programmer Operations for Gang programming

Match 74/4000 device

Selecting this menu item will open the Programmer Operations for Chiptest and automatically start to match any device in the ZIF socket to any device included in the Chiptest libraries (CTPars). This option is useful for identifying the functionality of an unknown device.

Diagnostics

Selecting ***Diagnostics*** allows you to run either:-

Programmer Selftest
Emulator Test

3.1.5 Help Menu

The WinLV Help Menu allows you to run the WinLV Help topics and see release information.

Help Topics

Selecting this runs the WinLV Help topics.

About WinLV…

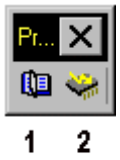
Selecting this will display the WinLV software version details for your copy of WinLV. Also shown is ICE Technology's copyright information.

3.2 WinLV Toolbars

3.2.1 General Toolbar

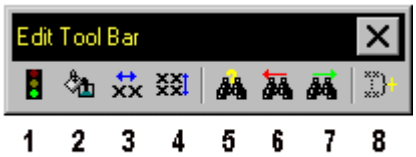
1. **New File** click if you wish to open a completely new buffer
2. **Open File** click to browse for and open an existing file.
3. **Save File** click to save the current file using the existing file settings.
4. **Open Project** click to open an existing project (*.ice) file
5. **Save Project** click to save a project with the exiting file settings and buffer parameters
6. **Print** click to print the current buffer contents.
7. **Print Preview** click to print preview the current buffer contents
8. **Cut** cut the selected area, saving a copy to the clip board
9. **Copy** copy the currently selected area of the buffer to the clipboard
10. **Paste** paste the contents of the clipboard to the currently selected area

3.2.2 Programmer Toolbar



1. **Select Device** click to open the device database viewer for selecting the device type
2. **Operations** opens the Programming Operations window.

3.2.3 Edit Toolbar

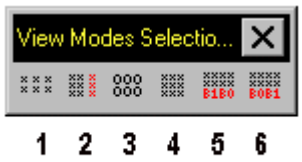


1 **Goto** Opens the got window, allowing the user to specify a specific buffer address to view.

2. **Fill** Opens the Fill window, allowing the user to specify a buffer area and a value with which to fill it.

3. **Swap Bytes** Allows the user swap the MSB and LSB within any word in the currently open buffer.
4. **Swap Words** Allows the user to swap the Most Significant Word and Least Significant Word within any Long value in the currently open buffer.

3.2.4 Display Mode Toolbar



1. **Nibble Mode** Displays the buffer content as nibble (4bit) wide data
2 **Byte Mode** Displays the buffer content as byte wide data
3. **Octal Mode** Displays the buffer content as octal (0-7) data
4. **12 Bit Mode** Displays the buffer content as 12 bit wide data
5. **Word (Byte1:Byte0) mode**
   Displays the buffer content as 16 bit wide data with the MSB on the left hand side
6. **Word (Byte0:Byte1) mode**
   Displays the buffer content as 16 bit wide data with the LSB on the left hand side

3.2.5 Selftest

Running The *Selftest* Utility

The WinLV software contains a programmer *Selftest* utility that enables you to check the following:-

- Checks your programmer has been installed correctly
- Checks for communications between PC and programmer
- Check for any internal hardware faults in your programmer

Use the *Selftest* when you connect up a new programmer, move your programmer to a new PC or if you suspect there may be fault with your programmer.

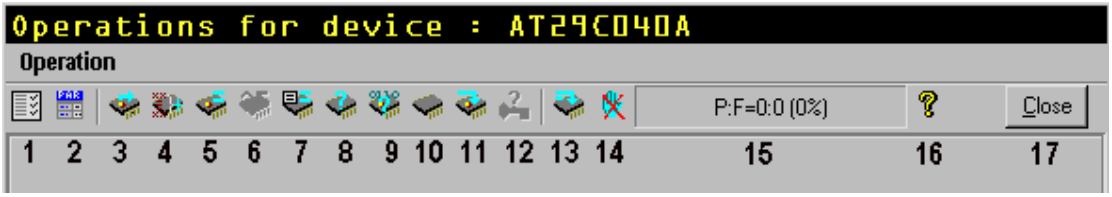
WARNING:- Always ensure that there is no device in the programmer before running *Selftest* . Doing so may lead to damaging the device and incurring false results from the *Selftest* diagnostic

What does the *Selftest* Do ?

- Tests communications between the programmer and PC via the parallel cable
- Tests programmer firmware version
- Tests programmer ROM and reports checksum
- Tests programmer RAM
- Tests programmer DACs
- Tests programmer ZIF socket pin drivers

3.3 Programmer Operations for Memory/Microcontrollers

3.3.1 Programmer Operations Toolbar (Memory/Microcontrollers…)



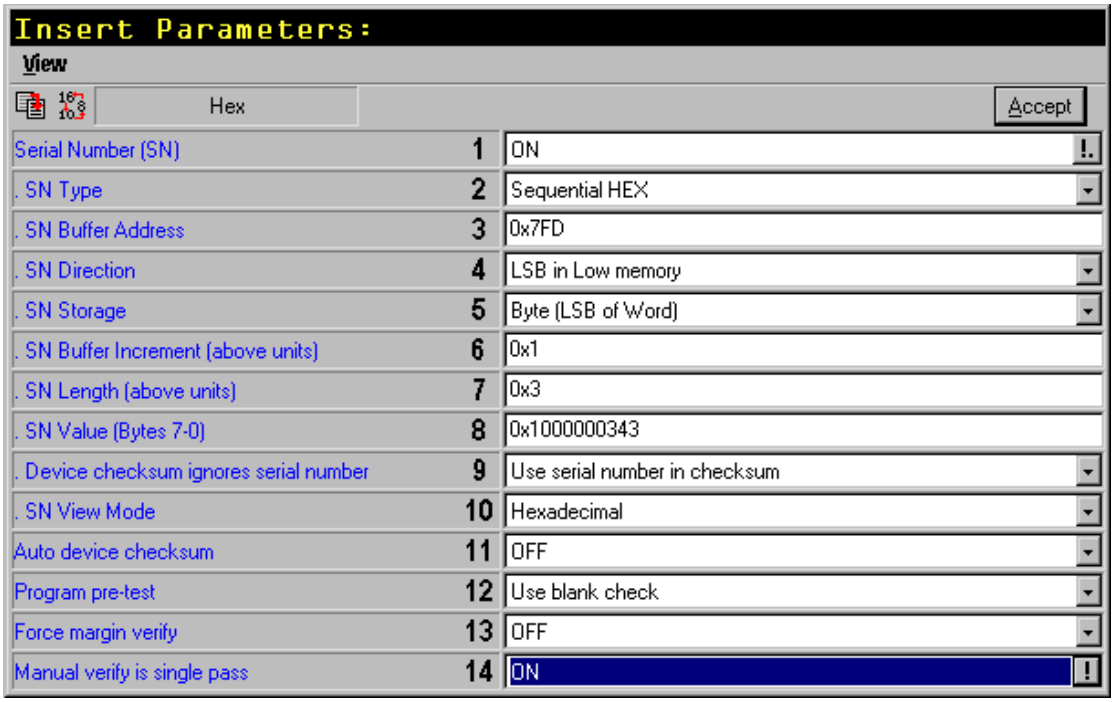
Programmer Operations Icons

1. Displays ***Programming Options*** window. Allows all programming parameters to be changed.
2. Toggles the ***Force Re-entry Of Parameters*** option.
3. Read device. This operation reads back data already programmed into the device in the ZIF socket.
4. Display ***Device Checksum*** window. Allows device checksum parameters to be changed.
5. Verify device. This operation will verify all of the data that is current held in the buffer with data that has been programmed into the device in the ZIF socket.
6. Verify encryption. This operation verifies the section of the buffer that holds the encryption data with the corresponding encryption array in the device.
7. Verify device signature.
8. Blank check device. This operation checks every address on the device and will report back if it is completely blank (empty) or if it has data at any of its addresses.
9. Bit test device. This operation checks that the data held in the buffer can be successfully programmed into the selected device. This is useful when over-programming a partly programmed device.
10. Erase device. This operation will clear/erase data held in the currently selected device. Empty address locations are usually represented with FF. This operation will only work on erasable devices, e.g. Flash memory
11. Over-Program device. Programs & verifies the device with the data held in the buffer without performing the blank check or bit test priory to the programme cycle.
12. Program device. Programs and verifies the currently selected device.
13. Reports back any device security information.
14. “Hands Free Programming” (*Only Available on the 48 pin programmers*). This operation helps you to programme large amounts of the same device. Once selected, follow the on screen instructions.

15. Program counter display. Keeps a count of how many of the currently selected devices have been programmed and gives the programming success rate as a percentage.
16. WinLV Help
17. Close. Select this to close the ***Programmer Operations*** window.

3.3.2 Programming Options / Parameters

Programming Parameters for EPROMs/Flash/Microcontrollers etc.



1. Serial Number (SN). Select serial number option ON or OFF.
2. SN Type. Select what format the serial number should take.
3. SN Buffer Address. Select at which address in the buffer should the serial number be stored.
4. SN Direction. Select LSB or MSB in low memory. E.g. a-byte serial number starting at 0000 at buffer address 00 being incremented by 1 every time a device is programmed. Selecting LSB will increment address 00 to FF with address 01 being the overflow. Selecting MSB means that address 03 is incremented to FF with address 02 being the overflow.
5. SN Storage. Select between nibble, byte or word.
6. SN Buffer Increment. Select the value to increment every serial number.
7. SN Length. Select the number of bytes to be used for storing the serial number.
8. SN Value – enter the staring value for the serial number sequence.
9. Device Checksum Ignores Serial Number – Self explanatory.
10. SN View Mode – Allows the user to select how the serial number should be displayed after programming

11 Auto-device Checksum. Select before or after securing the device

12. Select which pre-program test to carry out.

13. Force Margin Verify. After the programme cycle an automatic verify cycle is performed. This is at the nominal operating voltages as specified by the device manufacturer. The margin verify will verify the currently selected device at +/- 5% of the nominal verify voltage.

3.4 Programmer Operations for PALs/GALs/EPLDs

3.4.1 Programmer Operations Toolbar (PALs/GALs/EPLDs…)

Programmer Operations Icons

1. Read device. This operation reads back data already programmed into the device in the ZIF socket.
2. Verify device. This operation will verify all of the data that is current held in the fuse map with data that has been programmed into the device in the ZIF socket.
3. Blank check device. This operation checks every fuse on the device and will report back if it is completely blank (usually represented as all fuses set to 1) or if it is not blank.
4. Erase device. This operation will clear/erase data held in the currently selected device.
5. Over-Program device. Programs the device with the data held in the fuse map buffer.
6. Vector Test. This operation allows you to apply the test vectors for the currently selected device.
7. Erase, Program, Verify & Vector test. This operation performs a complete erase, program, verify and vector test on the currently selected device.
8. Reports back any device security information.
9. Auto-secure. This operation toggles the auto-secure feature to ON or OFF. If the auto secure is set to ON then the device to be programmed will be automatically secured.
10. “Hands Free Programming” (*Only Available on the 48 pin programmers*). This operation helps you to programme large amounts of the same device. Once selected, follow the on screen instructions.
11. Program counter display. Keeps a count of how many of the currently selected devices have been programmed.
12. WinLV Help.
13. Close. Select this to close the ***Programmer Operations*** window.

4. The ICE Technology Emulator Range

ICE Technology manufactures 4 in circuit ROM/RAM emulators as follows:

Model	**ROM size emulated**
LVEC-EMUL8	128K by 8
LVEC-EMUL8M	512K by 8
LVEC-EMUL16	128K by 16, 256K by 8, 2 times 128K by 8
LVEC-EMUL8	512K by 16,1024K by 8 2 times 512K by 8

All of these models work as internal plug-ins to our range of single socket programmers and can either be bought at the same time as a programmer or added later.

As far as operation of the Emulators under WinLV is concerned, the software detect whether an emulator is present and displays the emulator options in addition to the programmer functionality.

The obvious place where this will be seen will be under the “Programmer”, “Diagnostics” option – the Emulator test option will now be active – and in selecting a device

4.1.1 Tutorial 4: Emulating an 8-bit EPROM

This fourth tutorial helps to explain how the WinLV ROM/RAM emulator options work. This tutorial introduces you to the basic steps required to emulate an EPROM in Circuit. Once you are familiar with these you may want to investigate the more advanced emulation features such as the comms channel and the "modify on the fly" comms option. Follow these steps:

Step 1: Run WinLV

Select the WinLV.EXE file using the Windows 95 Explorer or, alternatively, create a Shortcut on the Windows 95 Desktop.

Step 2: Select Device To Emulate

The WinLV software will automatically detect if there is an ICE Technology ROM/RAM Emulator connected to your programmer and if so it will display the currently available parts to emulate in the Parts Database.

Remember that only the emulator parts that are applicable to your emulator will be displayed in the Parts database. Therefore if your are attempting to emulate a 16-bit EPROM with an 8-bit emulator installed then only 8-bit EPROMs will be displayed.

Choose the 8-bit EPROM emulator part by selecting it from the ***Select Device*** window. To open the ***Select Device*** window – click the left hand Mouse button on ***Programmer*** from the top menu bar then click on ***Select Device***. Use the mouse to browse through the Parts Database and select a device from the Emulator Parts section then click on Accept to confirm your choice.

Step 3: Load File into Buffer

Select the file you would like to download to the emulator following these steps:-

- Select the ***File*** menu and then ***Open…***
- Select which file you want to load into the buffer OR type the file name into the ***File Name*** box
- If any files that you expecting to see aren't there try using the ***Files of Type*** drop down menu to select which file type is displayed in the main window
- Use the ***Open as*** drop down menu to select which format the file will be opened as and loaded into the buffer.
- Set the Defaults to pre fill the buffer with 0xFF so that all empty address locations are filled with FF.
- Remember - If the buffer size is not large enough for the file selected then it will truncate the file before loading. To increase the buffer size

select ***New Buffer*** from the ***File*** menu and use the slider bar to increase the buffer size and then re-load the file.

Step 4: Connect Emulator to Target System

Because this tutorial is illustrating how to emulate an 8-bit EPROM then the IDC Cable must connect to the EMULATOR I socket on the back of the programmer with the 32-pin EPROM header end connected onto the Target System .

Step 5: Download To The Emulator

Once you have loaded the data file into the buffer (**Step 3**) and connected the emulator to the target system (**Step 4**) follow these steps to download the data to the emulator:-

- Select ***Operations*** from the ***Programmer*** menu to view the Operations For Device dialogue window.
- Click on the ***Download*** button in the emulator ***Operations*** dialogue window

Step 6: Start The EPROM Emulation

Now all the data from the buffer has been downloaded into the emulator hardware the next step is to activate the emulator to start the ROM emulation in the target system.
Do this by clicking on the ***Go*** button in the emulator ***Operations*** dialogue window.

Step 7: Stop The EPROM Emulation

Once the emulator is active you will notice that the Go button changes to a Stop button and other options are made available, e.g.- Generate a reset, "Modify on the fly" comms etc. Once the EPROM has been emulated in the target system and you need to stop the emulator click on the Stop button.

4.1.2 Tutorial 5: Emulating a 16-bit EPROM

This fifth tutorial leads on from the previous tutorial and explains how the WinLV ROM/RAM emulator options work when emulating 16-bit EPROMs. This tutorial introduces you to the basic steps required to emulate a 16-bit EPROM in Circuit. Once you are familiar with these you may want to investigate the more advanced emulation features such as the comms channel and the "modify on the fly" comms option. Follow these steps:-

Step 1: Run WinLV

Select the WinLV.EXE file using the Windows 95 Explorer or, alternatively, create a shortcut on the Windows 95 Desktop.

Step 2: Select Device To Emulate

The WinLV software will automatically detect if there is an ICE Technology ROM/RAM Emulator connected to your programmer and if so it will display the currently available parts to emulate in the Parts database. Remember that only the emulator parts that are applicable to your emulator will be displayed in the Parts database. Therefore if your are attempting to emulate a 16-bit EPROM with an 8-bit emulator installed then only 8-bit EPROMs will be displayed. Choose the 16-bit EPROM emulator part by selecting it from the Select Device window. To open the Select Device window - click the left hand mouse button on Programmer from the top menu bar then click on Select Device. Use the mouse to browse through the Parts Database and select a device from the Emulator Parts section then click on Accept to confirm your choice.

Step 3: Load File into Buffer

Select the file you would like to download to the emulator following these steps:-

1. Select the File menu and then Open...
2. Select which file you want to load into the buffer. OR type the file name into the File Name box
3. If any files that you expecting to see aren't there try using the Files of Type drop down menu to select which file type is displayed in the main window
3. Use the Open as drop down menu to select which format the file will be opened as and loaded into the buffer.
4. Set the Defaults to pre fill the buffer with 0xFF so that all empty address locations are filled with FF.
5. Remember - If the buffer size is not large enough for the file selected then it will truncate the file before loading. To increase the buffer size select New Buffer from the File menu and use the slider bar to increase the buffer size and then re-load the file.

Step 4: Connect Emulator to Target System

Because this tutorial is illustrating how to emulate an 16-bit EPROM then both of the emulator IDC cable must connect to the EMULATOR I and EMULATOR II sockets on the back of the programmer. For 16-bit EPROMS you will need to connect the IDC cable from EMULATOR I to the top of the 16-bit emulator EPROM adapter and the cable from EMULATOR II into the bottom of the adapter. Plug the 16-bit EPROM adapter onto the target system.

Step 5: Download To The Emulator

Once you have loaded the data file into the buffer (Step 3) and connected the emulator to the target system (Step 4) follow these steps to download the data to the emulator:-

1. Select Operations from the Programmer menu to view the Operations For Device dialogue window.
Click on the Download button in the emulator Operations dialogue window

Step 6: Start The EPROM Emulation

Now all the data from the buffer has been downloaded into the emulator hardware the next step is to activate the emulator to start the ROM emulation in the target system.
Do this by clicking on the Go button in the emulator Operations dialogue window.

Step 7: Stop The EPROM Emulation

Once the emulator is active you will notice that the Go button changes to a Stop button and other options are made available, e.g.- Generate a reset, "Modify on the fly" comms etc. Once the EPROM has been emulated in the target system and you need to stop the emulator click on the Stop button

4.1.3 Tutorial 6: Emulating a 8-bit RAM

This sixth tutorial provides step by step instructions to emulating a RAM in circuit. The procedure is very similar to the other emulator tutorials for 8-bit and 16-bit EPROMs.

Step 1: Run WinLV

Select the WinLV.EXE file using the Windows 95 Explorer or, alternatively, create a shortcut on the Windows 95 Desktop.

Step 2: Select Device To Emulate

The WinLV software will automatically detect if there is an ICE Technology ROM/RAM Emulator connected to your programmer and if so it will display the currently available parts to emulate in the Parts database . Remember that only the emulator parts that are applicable to your emulator will be displayed in the Parts database. Therefore if your are attempting to emulate a 1024Kx8 RAM with an emulator that can only emulate up to 256Kx8 RAMs then only RAMs up to 256Kx8 will be displayed. Choose the emulator part by selecting it from the Select Device window. To open the Select Device window – click the left hand mouse button on Programmer from the top menu bar then click on Select Device. Use the mouse to browse through the Parts Database and select a RAM from the Emulator Parts section then click on Accept to confirm your choice.

Step 3: Load File into Buffer

Select the file you would like to download to the emulator following these steps:

- Select the File menu and then Open…
- Select which file you want to load into the buffer OR type the file name into the File Name box
- If any files that you expecting to see aren't there try using the Files of Type drop down menu to select which file type is displayed in the main window
- Use the Open as drop down menu to select which format the file will be opened as and loaded into the buffer.
- Set the Defaults to pre fill the buffer with 0xFF so that all empty address locations are filled with FF.
- Remember - If the buffer size is not large enough for the file selected then it will truncate the file before loading. To increase the buffer size select New Buffer from the File menu and use the slider bar to increase the buffer size and then re-load the file.

Step 4: Connect Emulator to Target System

Because this tutorial is illustrating how to emulate an 8-bit RAM then the emulator IDC cable must connect to the EMULATOR I socket on the back of

the programmer with the 32-pin EPROM header end connected onto the target system .If the RAM you have selected to emulate has less than 32 pins then align the EPROM header with the bottom of the RAM socket on the target system, all unused pins will be ignored.

Step 5: Download To The Emulator

Once you have loaded the data file into the buffer (Step 3) and connected the emulator to the target system (Step 4) follow these steps to download the data to the emulator:-

- Select Operations from the Programmer menu to view the Operations For Device dialogue window.
- Click on the Download button in the emulator Operations dialogue window -

Step 6: Start The RAM Emulation

Now all the data from the buffer has been downloaded into the emulator hardware the next step is to activate the emulator to start the RAM emulation in the target system.

Do this by clicking on the Go button in the emulator Operations dialogue window -

Step 7: Stop The RAM Emulation

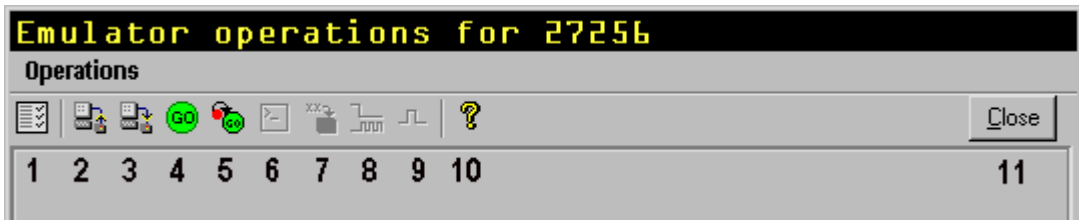
Once the emulator is active you will notice that the Go button changes to a Stop button and other options are made available, e.g.- Generate a reset, “Modify on the fly” comms etc. Once the RAM has been emulated in the target system and you need to stop the emulator click on the Stop button - .

Step 8: Upload From The Emulating RAM to The Emulator

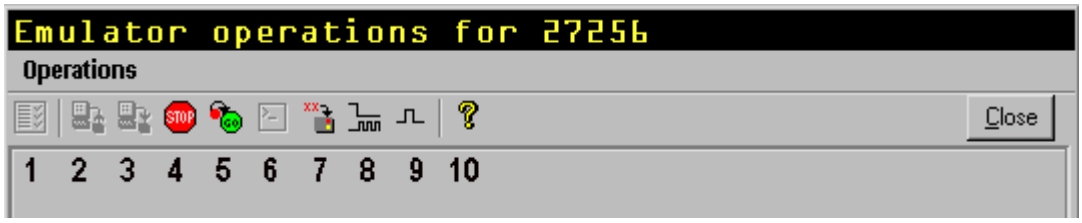
At any point during the RAM emulation after it has stopped you can upload the data from the RAM into the emulator. All the data uploaded will be displayed in the buffer for inspection.
Click on the Upload button in the emulator Operations dialogue window -

4.1.4 Programmer Operations for the ROM/RAM Emulator

Programmer Operations Toolbar (ROM/RAM Emulator)- *Emulator Not Active*



Programmer Operations Toolbar (ROM/RAM Emulator)- *Emulator Active*



Programmer Operations Icons

1. Displays ***Emulator Parameters*** window. Allows all emulator parameters to be changed.
2. Upload. This operation allows you to upload data from the emulating ROM or RAM in the target system - all the data uploaded will be displayed on screen in the programming buffer.
3. Download. This operation is the reverse of previous one. It allows you to download the data currently loaded into the buffer into the emulator hardware.
4. Go / Stop. This will either set the emulator status to active or stopped.
5. Stop, Reload and Restart. This facility allows the user to edit the data file being emulated while emulation is underway. When required the emulation can be stopped, the edited data file reloaded and the emulation recommenced with a single click on this icon.
6. Open Comms.
7. “Modify on the fly”.
8. Generate a reset. This operation will generate a manual reset to the target system. Pins 1 to 23 on the ZIF Socket can be used as an active low reset signal and pins 25 to 43 can be used as an active high reset signal. To generate a reset make sure that you connect any of these ZIF socket pins to the reset line on the target system.
9. Generate an interrupt.
10. WinLV Help.
11. Close. Select this to close the ***Programmer Operations*** window.

4.2 Emulator Test

Running *The Emulator Test* Utility

The WinLV software contains an ***Emulator Test*** utility that enables you to check the following:

Communications between programmer and emulator
Emulator hardware including emulator on-board RAM

Use the *Emulator Test* when you install a new emulator into a new programmer, or move the emulator from one programmer to another or if you suspect there may be fault with your emulator.

4.3 Chiptest Facilities

What Does Chiptest Do?

The WinLV Chiptest utility allows you to test certain devices to confirm that they are exactly what you expect them to be and also to check that the devices are functioning correctly. It does this be applying a combination of test vectors in order to determine exactly what the device is and to test its functionality. Chiptest can check the following device types:-

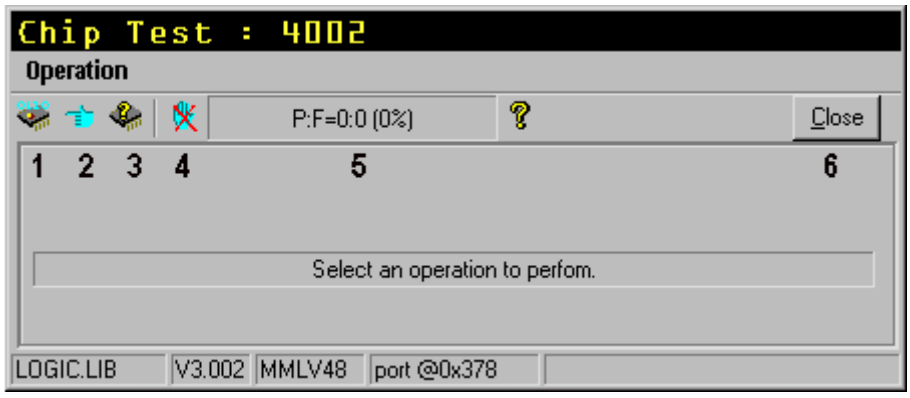
7400 Series Logic Parts
4000 Series Logic Parts
Static RAMs
Dynamic RAMs
Complex Logic Parts
User-Defined Logic Parts

Selecting a Chiptest Part

If you have a specific part to test and you already know its part number etc. you can select its library file using the Parts database. If you have a part that is unknown or it is not exactly clear then you can use the Match 74/4000 Device utility that will attempt to identify the part.

4.3.1 Programmer Operations Toolbar (Chiptest)

Programmer Operations Icons



1. **Test Device**. Click on this test the device in the ZIF socket.
2. **Toggle Hands Free Mode**. This allows you to select either - Test a device "Hands free" or Match 74/4000 Device "Hands free".
3. **Match 74/4000 Device**. This operation will check the "unknown" device currently in the ZIF socket against all of the test vectors available (including any user-defined vectors) to determine what that device actually is.

4. **"Hands Free Testing"** (Only Available on the 48 pin programmers). This operation helps you to test or match large amounts of the same device. Once selected, follow the on screen instructions. Remember that you can toggle between Test and Match using ICON 2.

5. **Chiptest counter display**. Keeps a count of how many of the currently selected devices have been tested using Chiptest.

6. **Close**. Select this to close the Programmer Operations window.

4.3.2 Chiptest Edit Menu/Toolbar

If you select one of the Chiptest parts from the parts database the Edit menu and Edit Toolbar change slightly to offer extra functions specific to Chiptest. These extra functions are included in both the Edit menu and the corresponding Edit Toolbar.

Edit Toolbar (Chiptest)

1. **Goto Vector**. This will move the cursor to a specific test vector.
2. **Fill Buffer**. This is not available during Chiptest
3. **New Device**. This prompts you for a new device name and clears all of the test vectors ready for new ones to be entered.
4. **Add to Library**. This will save the current test vectors as a new device. This allows you to add a new device to the Chiptest parts in the Part database.
5. **Remove from Library**. This will delete a user-defined part from the parts database.
6. **Previous Device**. Selects the previous Chiptest part available from the Parts database.
7. **Next device**. Selects the next Chiptest part available from the Parts database.

5. The Speedmaster GLV32

The WinLV Programmer Menu for Gang programming contains commands for controlling the operations of the GLV-32 programmer, selecting devices and setting up the stand alone mode .

Select Device

The Select Device menu item allows you to select which device you are currently using with your GLV-32 programmer. All physical devices - EPROMs & flash etc. can be selected from the parts database using the Select Device dialogue window.

Operations

Selecting the Operations menu item will open the Programmer Operations window so that you can control exactly what the GLV-32 programmer is doing.

Set Mode

This allows you to select which programming mode the GLV-32 should use. The 3 different modes are:-

Gang Mode:-

Gang mode will set up the GLV-32 to programme all devices with identical data from the buffer, i.e.: 1 set of 8 identical EPROMs.

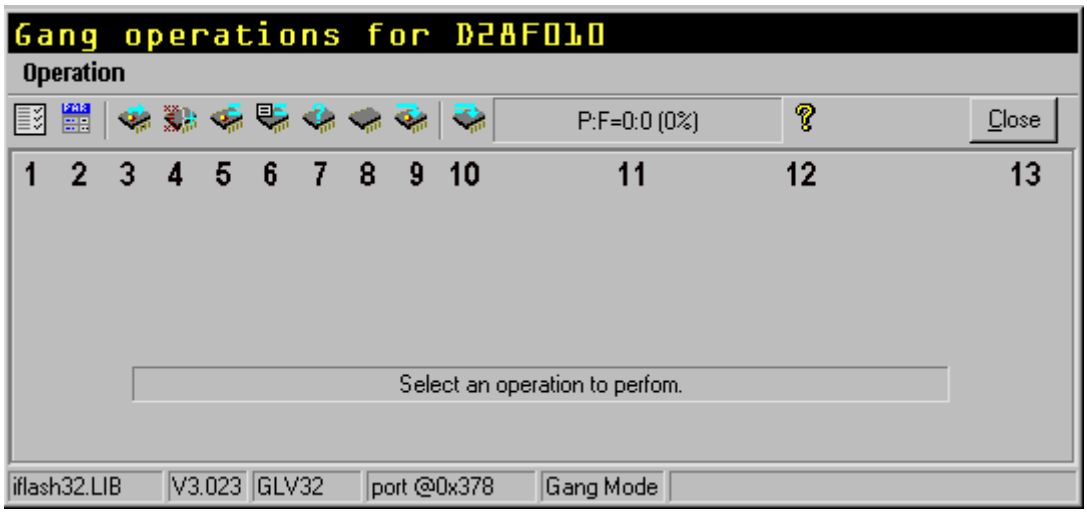
4 sets of 2 (32-bit mode):-

For 32-bit mode the programmer will send information into 4 sets of 2 chips. Sockets 0 and 1 will contain the LSB or the first byte specified from the buffer, and then every fourth byte. This maybe ODD or EVEN addresses, depending on the given Start address. Sockets 2 and 3 will contain the second byte and every fourth byte from there, sockets 4 and 5 will contain the third byte, and every fourth byte from there. Sockets 6 and 7 will contain the MSB or the fourth address, and then every fourth byte in the buffer.

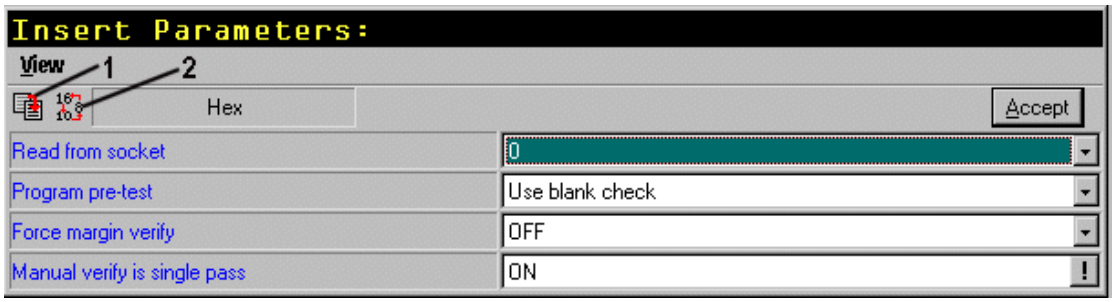
2 sets of 4 (16-bit mode):-

For 16-bit mode the programmer will send information into 2 sets of 4 chips. Sockets 0 to 3 will contain the LSB or the first address specified from the buffer, and then every second byte in the buffer. This maybe ODD or EVEN addresses, depending on the given Start address. Sockets 4 to 7 will contain the MSB or the second specified address, and then every second byte in the buffer.

5.1.1 Gang Programming Options



1. **Programming Parameters** – Opens the Insert Parameters Window (see next section).
2. **Force Re-entry of parameters** – ensures that the Device Operation Parameters window opens on the next operation.
3. **Read** – click to read a device
4. Device Checksum – click to read device and have checksum window open
5. **Verify** – click to compare the buffer contents with those of the device.
6. **Verify Signature** – Compares the device electronic signature with that expected (where appropriate to the device)
7. **Blank Check** – click to check if the device in the ZIF socket is blank.
8. **Erase** – click to erase the device in the ZIF socket (where appropriate)
9. **Over Programme** – programmes and verifies the device without a prior erase cycle
10. **Erase Programme Verify** – as per 9. but with a prior erase cycle.
11. **Program counter display** - Keeps a count of how many of the currently selected devices have been programmed and gives the programming success rate both as a ratio and as a percentage.
12. **Help** - Click to access WinLV help.
13. **Close** – click to close the current window.



Set Up Internal EEPROM

This option is used to configure the internal EEPROM inside the GLV-32 programmer. This is necessary when you are using the GLV-32 in stand-alone mode Selecting this option will configure the GLV-32 to programme devices in stand-alone mode using the device currently selected in the software.

Set Up Master EPROM

This option is used to configure the Master Configuration EPROM in the master socket on the GLV-32. This is necessary when you are using the GLV-32 in stand-alone mode

5.1.2 GLV-32 Command Line Software

WinLV and the DOS software contains a utility that allows the Speedmaster GLV-32 Gang/Set Programmer to be operated using command line software at the DOS prompt. All GLV-32 functions featured in WinLV, such as programme, blankcheck, verify etc., can be invoked directly from the DOS prompt be using simple typed commands. The command line software provides the user with an alternative quick and efficient interface with the GLV-32 programmer which in turn can speed up production of large batches of the same device.

The command line software works by using a previously saved WinLV Project file (*.ice) and a collection of single word commands and command options to operate the GLV-32 hardware.

An application note is available from the website and on the ICE Technology software CD giving instructions on how to use the command line software.

5.2 Stand-Alone Mode

The Speedmaster GLV-32 gang programmer is capable of operating without being connected to a PC. This is known as stand-alone mode. The stand-alone mode allows the GLV-32 to be set up to programme a particular device with a specific data file so that it can be moved from one location to another without the need for a PC to control its actions.

Operating In Stand-Alone Mode

Once the configuration is complete, switch off the programmer. If you are using a Configuration EPROM, place the device in the Master socket. If you are using the Internal EEPROM, ensure that the Master socket is empty.

When the GLV-32 is powered up, it waits to see if there is a message from the PC. If no message is received, it looks for a 27C512 EPROM in the Master socket. If no device is present, it looks in the internal EEPROM. If no data is found, or if there is a problem locating the Configuration EPROM, the Master LED will flash three times in succession and will continue this error message until the machine is reset.

Once the machine has found the correct device parameters, LEDs 0-7 will display a code byte indicating which type of device it is set up for. Once the code has been checked, clear the display by pressing reset.

Programming In Stand-Alone Mode

Place the DATA Master EPROM in the master socket, then place the blank devices in sockets 0-7. Press START to begin programming. While programming is taking place, the indicator LEDs will light for each socket in which there is a chip; the master LED will flash. During the VERIFY stage, the master LED will remain on permanently. If programming is successful, a “walking” sequence will appear on the LEDs (LEDs 0,4 1,5 2,6 3,7).. If unsuccessful, the LEDs will flash to indicate which sockets have failed.

Verifying In Stand-Alone Mode

Devices are automatically verified after programming. If it is wished to verify them separately, press and hold RESET, press START while still keeping RESET held down; then release START, then release RESET. For any device which fails to verify, the associated LED will flash. Press RESET to clear the error.

What are Power-up Codes?

When the Speedmaster GLV-32 is set up for Stand Alone mode it will indicate which device the internal EEPROM is configured for using a power-up code. The power-up code operates using the LEDs numbered 0 to 7 . All device types have a Hexadecimal power-up code which is indicated as a binary sequence using the 8 LEDs. For example, if the internal EEPROM is configured for a 27C010 then the power-up code will be 07h indicated as binary 7 with LEDs 0,1 & 2 on and LEDs 3,4,5,6, & 7 off.

Example Power-up Codes

01	**2716**	**02**	**2732**	**03**	**2764**
04	**27128**	**05**	**27256**	**06**	**27512**
07	**27010**	**08**	**27020**	**09**	**27040**

0A	**27080**	**etc**
11	**2816**	**etc**

LED Programmer Status Codes

During operation, the LEDs provide a lot of information as about the success or otherwise of programming operations. The following list translates the meaning of each message:

Flashing Master LED repeated three times
BAD MASTER CHIP OR INTERNAL EEPROM NOT SET UP CORRECTLY

Flashing Master LED and any other flashing LED
MISALIGNED OR FAULTY DEVICE. IF ALL EIGHT SOCKET LEDs FLASH, THIS IMPLIES THAT ALL SOCKETS ARE FAULTY OR EMPTY.

Flashing Master LED and any other LEDs on
PROGRAMMING IS TAKING PLACE

Master LED on and all other LEDs off
VERIFYING MASTER EPROM

Master LED off and any other LED flashing
VERIFY ERROR

Master LED off and any sequence of LEDs on.
POWER UP CODE FOR DEVICE

6. The Matrix Programming System

The Matrix is a Universal Gang Programming System offering support for nearly 4000 devices including memory, logic and microcontrollers. It is available as either a 4 or 8 site unit. Each unit has a through port facility allowing multiple units to be daisy-chained together - allowing the control of up to 6 Matrix Systems from a single PC, that is up to 48 programming sites controlled by one user.

A “hands free” programming mode is available – allowing the user to instigate each programming operation with a single press of the button on the programmer it self after setting up a programming environment. A separate set of LEDS for each site on the programmer case report the status and result of programming of each device so the operative does not have to refer back to the PC once it is set up.

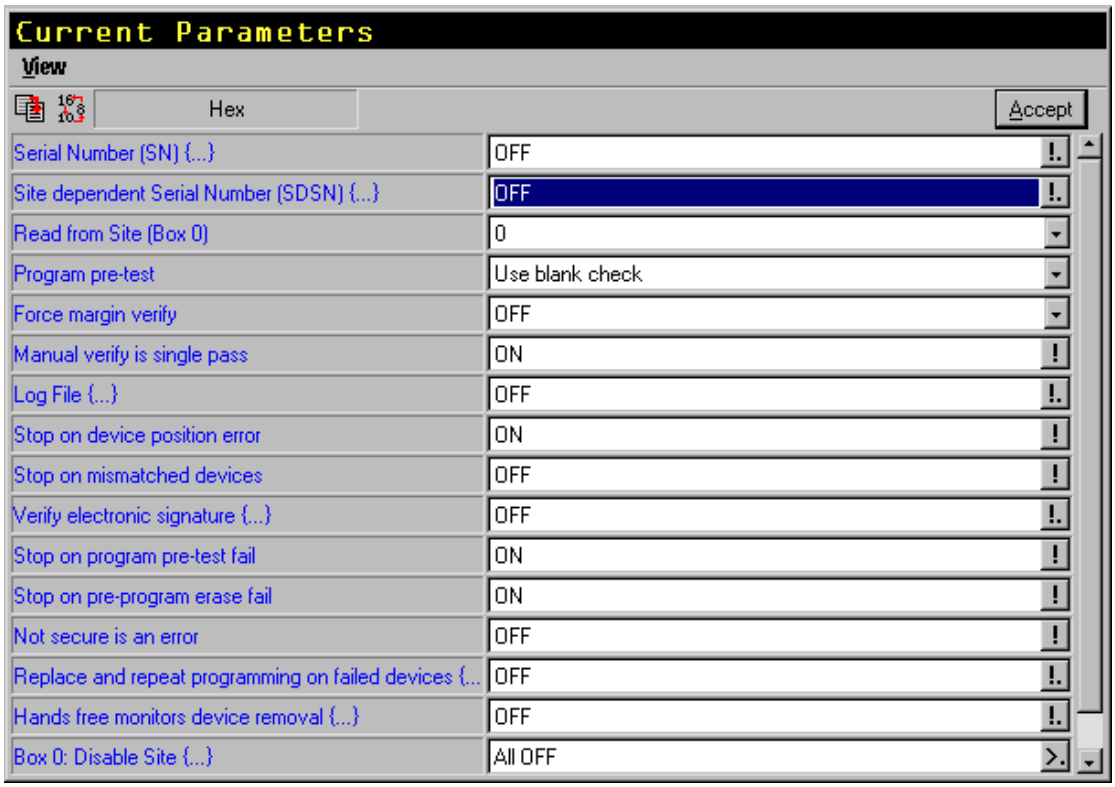
A full range of programming parameters and options are available to the user – allowing the programming environment to be tailored to individual preferences. This is explained in the section Matrix Programming Parameters.

The Matrix uses universal socket modules for all the most popular device packages up to 48 pins – pin counts above this and up to 100 pins are supported in more specialised modules. Modules can be purchased either individually or as sets of 4 or 8. Please contact Ice Technology Sales for more details.

The Matrix uses the same software as our other programmers and installation of its software is identical– with some additional features reflecting the additional facilities and features of this gang programmer. These differences are outlined in Matrix Programming Parameters. Connection is described in “Connecting your Matrix” (Section 2.3).

6.1 Matrix Programming Parameters

The Matrix Programming System offers a number of programming options, you are given the opportunity to select these when you open the Programmer Operations window. The Programming Parameters window will open with options as below:



Serial Number:
This feature allows the user to program a serial number onto the selected devices for batch identification/dating purposes. Switching this option to “on” will make a number of additional serial number options available. See Matrix Serial Numbering Options for more details.

Read From Site:
Allows the user to select the site from which they wish to read the contents of a device into the WinLV buffer.

Program pre-test:
Defines what tests are applied to devices before a programming operation. Three options are available: None (Aways erase FLASH), Use bit test and Use blank check

Force Margin Verify:

Allows the user to define the level of additional margins for verify passes. The available settings are Vcc +5%, Vcc –5% and Vcc ±5%.

Manual Verify is Single Pass:

Defines whether a user instigated verify will perform a single pass. Can be set to ON or OFF.

Stop on Device Position Error:

Allows to user to set the programming to be halted if a device position error is detected. Can be set to ON or OFF.

Stop on device Mismatch:

Allows the user to set the programming to be halted if a mix of device types is detected in the modules. Can be set to ON or OFF.

Verify electronic signature:

Checks the signature of all the devices prior to programming. Can be set to ON or OFF.

Stop on bad signature:

Allows the user to set the programming to be halted if a device signature mismatch is detected. Can be set to ON or OFF.

Stop on program pretest fail:

Allows the user to set the programming to be halted in the event that the any pretest they have set is failed. Can be set to ON or OFF.

Stop on program erase fail:

Allows the user to set the programming to be halted if the pre-programming erase option has been set and fails on any single device. Can be set to ON or OFF.

Not Secure is an error:

Allows the user to set the programmer to test the security of devices after programming and report any unsecured devices as incorrectly programmed. Can be set to ON or OFF.

Replace & Repeat on Failed Devices:

Set to “On”, this allows the user to repeat a programming operation in the event of failures on one or more of a set of devices. When used in conjunction with the Site Dependent Serial Numbering option, this can be used to ensure the continuity of serial numbering – which is useful as a way of avoiding confusion when labelling serial numbered parts.

Maximum Times to Repeat:

This option is available when Replace & Repeat on Failed Devices is set to “ON”. It limits the number of attempts made to re-programme devices in the event of failures.

Hands free monitors device removal:

Allows the user to set the programmer to monitor and report the removal and insertion of devices in each module in hands free mode before allowing the next programming operation. If this option is set to on, a further option box is opened allowing the user to input a number to define how long (in seconds) the programmer will monitor the modules before reverting to its “ready to program” state. The delay can be input as a decimal, hex or binary number – according to the radix set by the button in the top left corner of the parameters window.

Box No. x: Disable Site x

Allows the user to instruct the software to ignore a site in the event that a site is faulty or is being serviced. The boxes should ticked according to which sites are to be ignored. The WinLV software will detect how many Matrix boxes and sites are available on startup.

This feature enhances the fault tolerance of the Matrix system as a fault on a single site will not mean that production is completely halted.

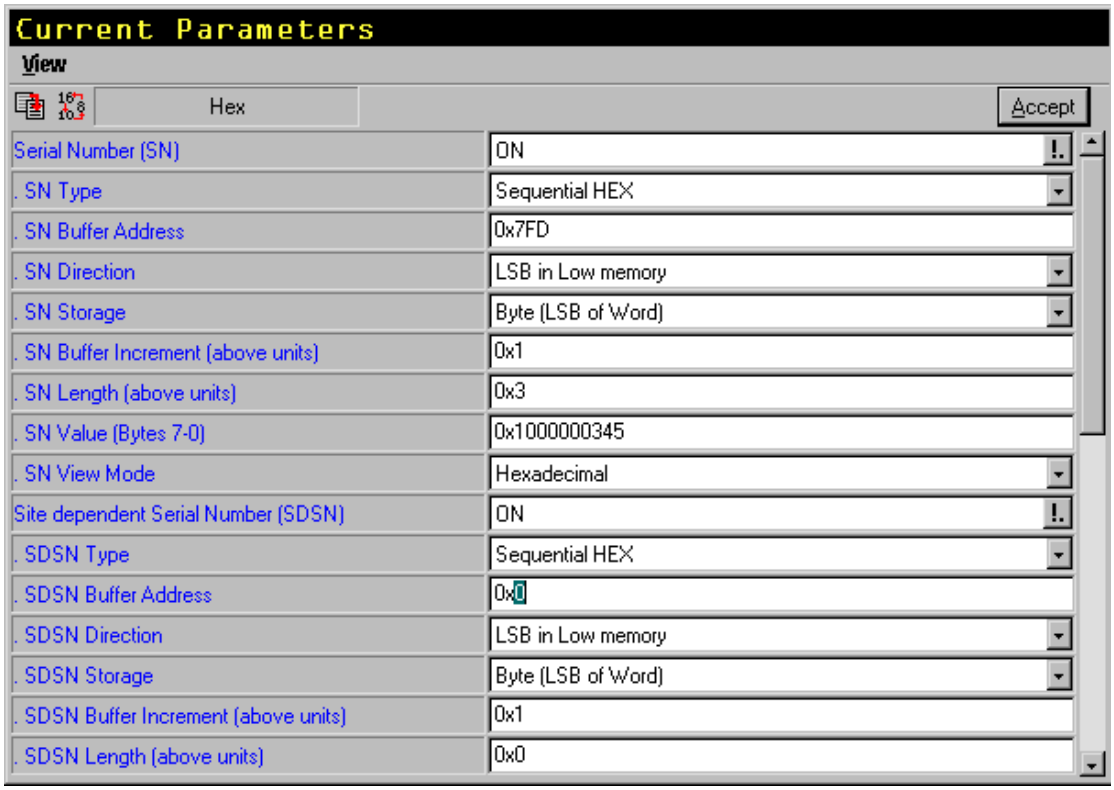
User Privileges and the User Logging Facilities

In a production environment, it can be useful to allow different programmer users different levels of access to the features available within the supporting software. For this reason, the Matrix has an additional facility to allow an administrator to set password protected access rights for each user. This is described in Setting up User Privileges.

It is also possible to define a working directory and set WinLV to create a log file for each user. This is created in a comma-separated format, which can be imported into spreadsheet packages such as Microsoft Excel. This data can then be used for statistical analysis of both operative and programmer performance.

6.1.1 Matrix Serial Numbering Options

Setting the serial numbering option to “ON” will make a number of further serial numbering options available. These are as follows:



Site Dependent Serial Numbering (SDSN):
Allows the user to programme each device with a different serial number in a single programming operation – depending on the site in which it is programmed. When set to “ON” a number of further options become available, details of these can be found under Site Dependent Serial Numbering Options.

SN Type
Allows the user to set the serial number type to one of the following options: Sequential Hex , Random Hex , Hex Date & Time, BCD Date & Time , Sequential BCD and Random BCD.

SN Buffer Address
Allows the user to define the device address at which the serial number should start.

SN Direction
Defines whether the serial number’s least significant byte or most significant byte is stored in low memory.

SN Storage
Defines the serial storage mode – the user can select each of Nibble, Byte and Word wide formats and the appropriate data direction.

SN Buffer Increment
Allows the user to define the serial number buffer increment – the units are based upon the SN Storage selection.

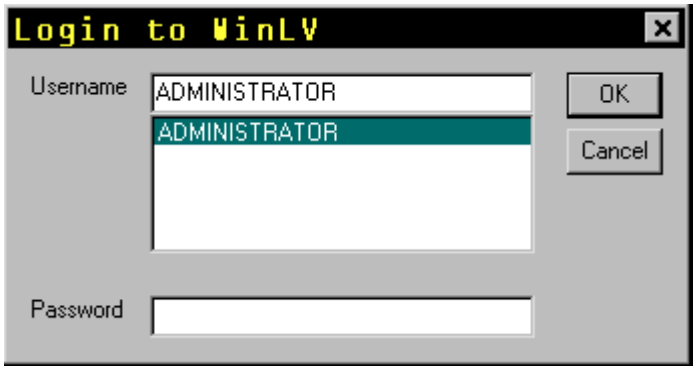
SN Length
Allows the user to define the length of the Serial Number - the units are based upon the SN Storage selection.

SN Bytes 0-3 and SN Bytes 4-7
Allows the user to set a the starting value of the serial number – this is used for sequential serial numbering schemes.

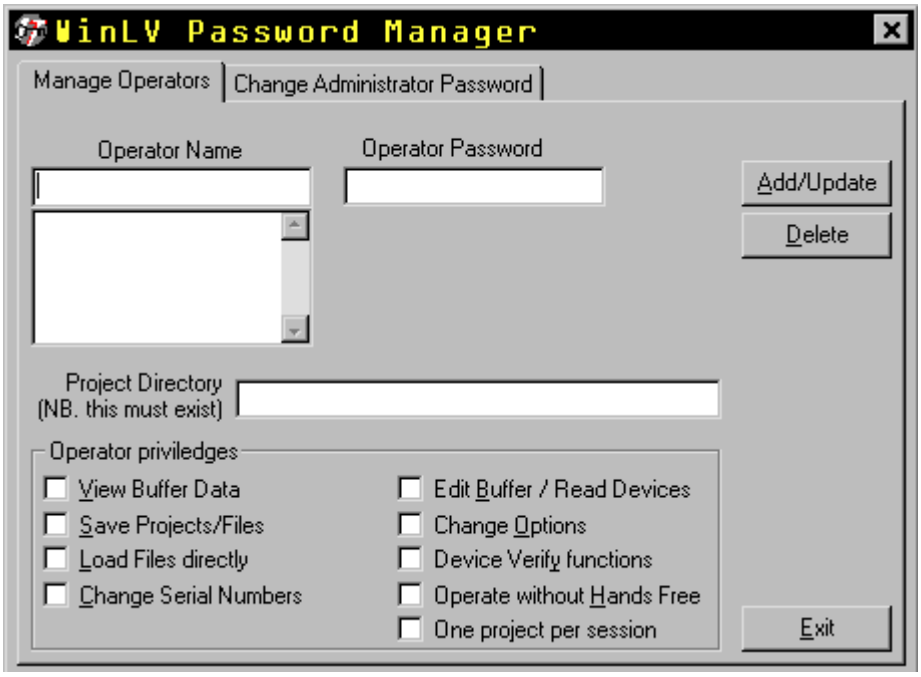
Device Checksum ignores serial numbers
Allows the user to include or not include the serial numbers in any checksum calculations that are performed on the programmer.

6.1.2 Setting Administrator/User Passwords and Privileges

When WinLV is first started, with either a Matrix connected or in Matrix demo mode, you will be asked for an administrator password, in a window as shown in the diagram below. In the first instance, you can either leave this blank or enter the password you intend to use for administration/password purposes in the future.



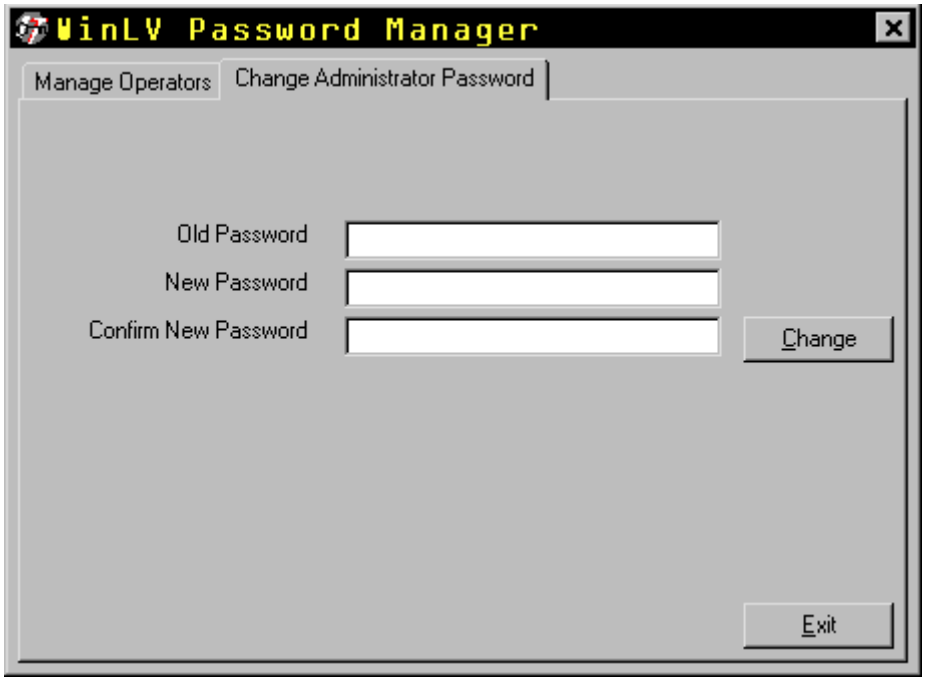
Once WinLV is running you will find an option in the main window, under file, called “Administrator Functions…”. Clicking on this option will start the “WinLV Password Manager” application – with a window as shown below:



The Operator Name field allows you input the names of a user and the Password field must be used to allocate a unique password to that user. The box in the lower left hand side of the window contains descriptively labelled tick boxes that allow each user to be allocated with a very specific level of access. Each box can be ticked to allow the privilege described as desired.

The “Project Directory” field is used to define the path to which the given users log information is to be stored – the directory defined still has to actually exist and will need to be set up in a file management utility such as Windows Explorer.

Clicking on the tab labelled “Change Administrator Password” will open the window shown below:



This allows the administrator to change the administration password – key in the old password, the new password and confirmation of this and click on the “Change” button.

7. ICE Technology Contact Information

UK:

Mail:		**ICE Technology Ltd** Unit 4, Penistone Court Sheffield Road Penistone Sheffield. S36 6HP U.K.
Telephone:		+44 (0)1226 767404
Fax:		+44 (0)1223 370434
Email:	**Tech support:**	support@icetech.com
	Sales:	sales@icetech.com
REG IN ENGLAND No:		2276124
VAT No.		490 5577 19

USA and Canada

Mail:	**ICE Technology Inc.** 430 Peninsula Ave, Suite 4 San Mateo. CA 94401. USA
Telephone	**(toll free US/Canada) 1(888)ICE 2305/423 2305** **(outside US/Canada): 1 (650) 375 0427**
Fax	**1 (650) 375 8666**
Email:	**iceusa@icetech.com**

Web Page:

www.icetech.com

The ICE Technology Home Page on the World Wide Web (WWW) includes up to date information on all of ICE Technology's products and services. For example:

- Technical Specifications on all ICE Technology products.
- Software and Upgrades.
- Current prices.
- Device support lists.
- Device package adapters.
- Other information including ICE Technology News and International distributors.

8. Glossary of Terms

Buffer
This is the memory area into which you can load hex files/fuse map ready to be downloaded and programmed into your selected device. Also, the buffer holds information that has been read back from a device in the programmer. The buffer displays memory addresses and the hexadecimal information stored at each address or, for logic devices, fuse map information.

Buffer Checksum
This checksum takes into account all of the addresses in the buffer, unless a different start and end address are selected using the checksum calculator.

Checkmark
Indicates if a menu item is active (e.g.: if a toolbar is shown (selected))

Device Checksum
This checksum takes into account all of the addresses in the user programmable memory of the currently selected device.

Device Signature
Some devices may have an electronic signature that identifies the manufacturer and the specific device type. This signature can be read back from a device to confirm that the currently selected device matches the actual device in the ZIF socket.

Dongles
A dongle is a small piece of hardware that is used stop software theft. The dongle is connected to the LPT port.

Emulator IDC Cable
This is the cable that connects the emulator to the target system. The IDC cable is a 34 way lead with a standard 32-pin DIL EPROM header together with a red flying lead for interrupt signals and a black flying lead for future use. Always make sure that the bottom of the EPROM header is aligned with the bottom of the EPROM socket on the target system board.

Encryption
Some microcontrollers have the option to be encrypted. This is a form of security that protects the data so that it cannot be read back very easily. Rather than just selecting to secure a device using a normal software data protect feature, the encryption actual holds information in an encryption array on the device which specifies how the data is encrypted.

EPLD (Erasable Programmable Logic Device)
This is similar to the PAL device. EPLD is usually used to describe complex high-density logic devices like the AMD MACHs, Altera MAX5000/7000 and Xilinx EPLDs.

File Type / Format
WinLV can support with many different file formats. The most common being:-

Raw / Binary data	*.BIN / *.DAT
ASCII space hexadecimal	*.HEX
Octal	*.OCT
Intel Hexadecimal	*.HEX
Motorola Hexadecimal	*.HEX / *.S**
Tektronic Hexadecimal	*.HEX
Extended Tektronic Hexadecimal	*.HEX
Mos Tech Hexadecimal	*.HEX
TI Tag Hexadecimal	*.HEX
JEDEC	*.JED / *.JDC / *.VEC

Fuse Map
An array of fuses that can be set (1) or not set (0) and then programmed into a PAL, GAL, PEEL etc. so that specific logical expressions can be implemented in circuit.

GAL (Generic Array Logic)
Has similar functionality as a PAL device although it can be erased. GAL is actually a trade name for PALs manufactured by Lattice Semiconductors.

GLV-32 LED Layout
The LEDs are used for indicating when various GLV-32 operations are active and also for device power-up codes.

Highlighted
By holding down the left mouse button and dragging the pointer across the buffer you can highlight a selection of buffer addresses. Once highlighted, this selection can be copied and pasted into another location in the buffer.

ICE Technology ROM/RAM Emulators

LVEC-EMUL8	up to 128K x 8 ROM	128Kx8 RAM
LVEC-EMUL8M	up to 512K x 8 ROM	512Kx8 RAM

Interrupt
An Interrupt is a unique value assigned to any process or item of hardware in a PC. The interrupt identifies which process or hardware is communicating with the CPU. Problems can occur when two or more pieces of hardware or processes share one interrupt.

JEDEC
JEDEC (Joint Electronics Device Engineering Council) specify standards for implementing and programming Logic file formats.

Keyboard Control
WinLV operates like any other Windows application. You can use the keyboard (if you don't have mouse) to execute any command in the WinLV menus by using the normal Windows keyboard conventions. E.g.:- using <alt> key + <letter> key, <tab> key etc.

LSB
LSB – Least Significant Byte

Master Configuration EPROM
Setting up the Master Configuration EPROM provides you with an alternative to using the internal EEPROM. Once a Master EPROM has been configured it will contain information that tells the GLV-32 which EPROM type has been selected for use in stand-alone mode. If you are using numerous EPROM types (e.g.: AM27C256, M27C64 etc.) in stand-alone mode then you will need a Master Configuration EPROM to set up the GLV-32 for each device type.

Mouse Control
WinLV operates like any other Windows application. Using the simple point and click procedure with the mouse you can select and execute any command in the WinLV menus.

MSB
MSB – Most Significant Byte

Nibble
1 Nibble = 4 bits (1/2 Byte)

PAL (Programmable Array Logic)
A digital electronic device that can accept logical inputs and respond with programmed logical outputs. PALs are one-time programmable (OTP) and implement bi-polar fuses.

Parts Database / Select Device
The Parts Database holds information on all of the currently supported devices for the programmer that you are using with WinLV. The parts can be arranged in alphabetical order or in size order (i.e: memory size for EPROMs & Micros, No. of fuses for PLDs). Use the mouse to select the part manufacturer and then the actual part name from the database list. Remember you only select a device when you have reached the small "chip" icon in the parts database. Click on ***Accept*** to finally select part.

Project
A WinLV project/environment file allows you to save all aspects of a programming project in one convenient file. All project files have the *.ICE extension. Project files save the following information:-

Hex/Data File currently being used

Device currently selected

Any device programming parameters you have specified for the selected device

Radio Buttons
A selection device that allows you to enable or disable a function by clicking the left mouse button.

Security
Some devices have a security option that allows the data stored in them to be protected. There are different types of device security although usually data cannot be read back from a device after it has been secured.

Single Socket Multi-Device Programmers
WinLV can be used with all of ICE Technology's Single Socket Device Programmers:-

EPMaster LV
EPMaster LV48
Speedmaster 1000+
Speedmaster LV
Speedmaster LV48
Micromaster 1000+
Micromaster LV
Micromaster LV48
LV40 Portable

Stand-Alone Mode
The Speedmaster GLV-32 gang programmer is capable of operating without being connected to a PC. This is known as stand-alone mode. The stand-alone mode allows the GLV-32 to be set up to programme a particular device with a specific data file so that it can be moved from one location to another without the need for a PC to control its actions.

Target System
The target system is the system within which the emulating ROM or RAM is situated. It describes any system or circuit board etc. that is currently linked to the emulator and receiving or sending data.

Test Vectors
Test vectors are used to check that the logic implemented in a PLD (Programmable Logic Device) is correct. That is:- the expected logical inputs will incur the expected logical outputs.

Test vector nomenclature:-

0 (zero)	Logic low input
1 (one)	Logic high input
L (low)	Logic low output
H (High)	Logic high output
X	Don't care
C	Clock.
Z	Expect high impedance
N	Power pin (Vcc or GND).

Value Length

Byte	represents data as an 8 bit value
Word	represents data as a 16 bit value
Long	represents data as a 32 bit value

Windows 95 Shortcut
A shortcut will allow you to run the WinLV software by double clicking on the WinLV shortcut icon. See Windows 95 Help for setting up shortcuts.

ZIF Socket
All dual in line devices and device package adapters have to be inserted correctly into the programmer ZIF (Zero Insertion Force) socket before programming / reading etc.

9. Legal Notices

ICE Technology cannot be held responsible for any loss or damage, consequential or otherwise, incurred while using its equipment. While ICE Technology makes every effort to ensure that the programming algorithms are correct, the said algorithms are not guaranteed by the manufacturers and thus ICE Technology cannot make any guarantees regarding the algorithms implemented on its programmers. Users are advised to ensure that devices work correctly in their systems before programming large quantities and to implement regular checks as an integral part of their Quality Assurance procedures. Regularly check the ICE Technology web site for device algorithm updates.

In no event shall ICE Technology be liable to the user or any third party for incidental, special or consequential damages, including without limitation, loss of use, loss or alteration of data, lost profits or lost savings, arising out of the use or inability to use ICE Technology products or software or any combination or portion thereof.

Copyright Notice

WinLV User Manual

ICE Technology has made every attempt to ensure that the information in this document is accurate and complete. ICE Technology assumes no liability for errors, or for any incidental, consequential, indirect or special damages, including, without limitation, loss of use, loss or alteration of data, delays or lost profits or savings, arising from the use of this document, or the product which it accompanies.