

LOGICAL Help

Contents

Front Panel Features

- [Main Menu](#)
- [Message Window](#)
- [Module/Socket Status Display](#)
- [Operation Progress Bar](#)
- [Programmer Settings Buttons](#)
- [Programmer Operations Buttons](#)
- [Device Status Display](#)
- [File/Editor Options Status](#)
- [Device Counter Display](#)
- [Auto-Options Status Display](#)
- [Front Panel Views](#)
 - [Normal View](#)
 - [Info View](#)
 - [Status View](#)
 - [Minimal View](#)

Operations on Data Files

- [Loading Data Files](#)
- [Saving Data Files](#)
- [Splitting Data Files to Multiple Devices](#)

Programmer Options

- [Device Address Range](#)
- [Auto-Programming/Verify Options](#)
- [Device Specific Options](#)
- [Device Counter](#)
- [Serializer Options](#)
 - [Control Options \(legacy\)](#)
 - [Storage Options \(legacy\)](#)
 - [Generator Options \(legacy\)](#)
 - [Preview Tool \(legacy\)](#)
 - [Log File](#)

Programmer Setup

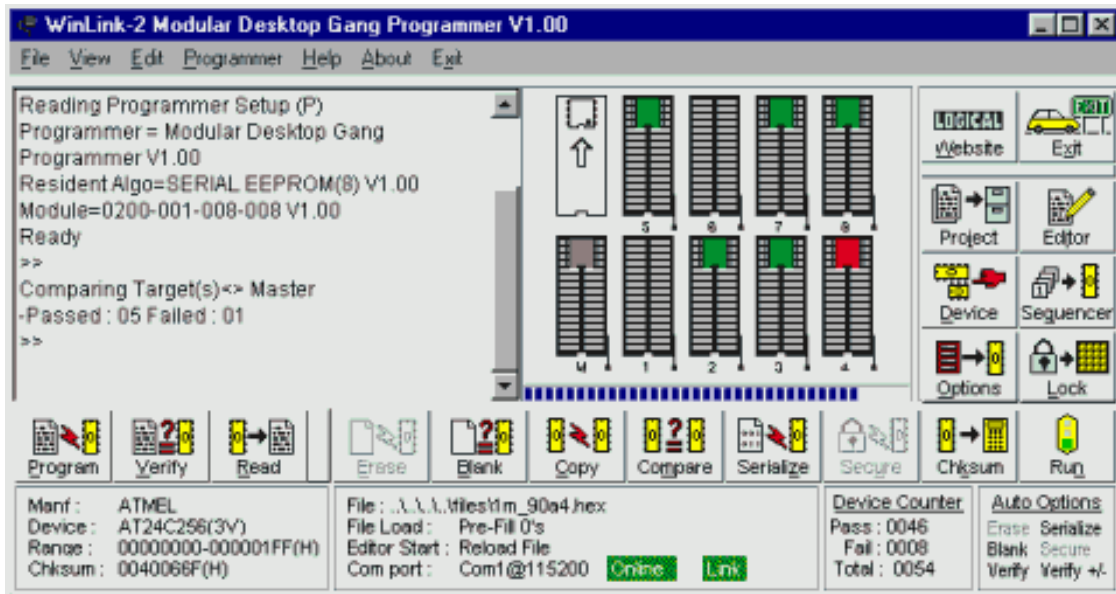
- [Comm Port](#)
- [Update Programmer Base](#)
- [Update Programmer Module](#)
- [Module Algo File Names](#)
- [Panel Lock](#)
- [Device Selection](#)
- [Project Manager](#)
- [Sequencer](#)

Programmer Enhancements

- [Module Combo](#)
- [Command Line](#)

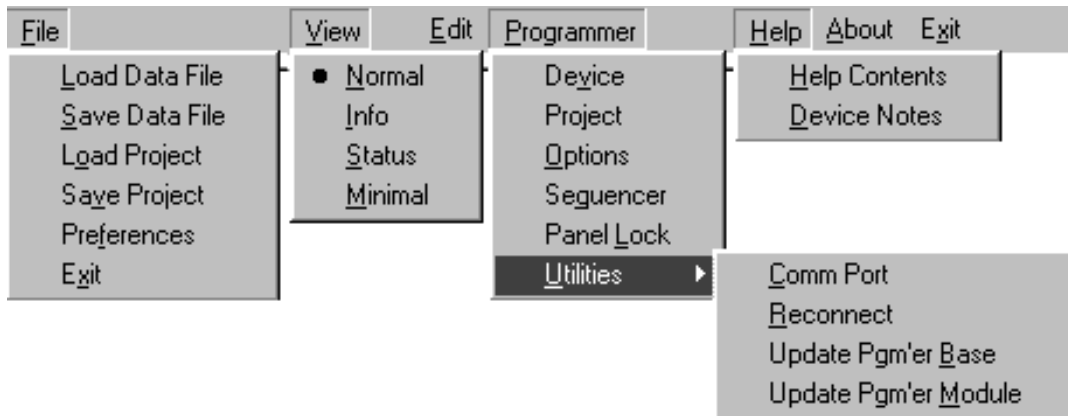
LOGICAL Help

FRONT PANEL



Click on a section to view help on that section

Main Menu



Click on a menu item to view help on that item

Composite view of Main Menu showing expanded sub-menus.

Menu Items may be disabled (grayed-out) depending on the integrity of the software in the programmer, whether the programmer is connected, whether the installed programmer software matches the installed module, whether a device is selected, which options are configured, and the state of the Panel Lock mechanism.

Message Window

The message window displays information regarding the running operation. The PASS/FAIL status of the operation is displayed on completion.

When connection to the programmer is first established, information on the currently installed module and algorithm is displayed.

```

Reading Programmer Setup (P)
Programmer = Modular Desktop Gang
Programmer V1.00
Resident Algo=SERIAL EEPROM(8) V1.00
Module=0200-001-008-008 V1.00
Ready
>>
Comparing Target(s)<> Master
-Passed : 05 Failed : 01
>>
  
```

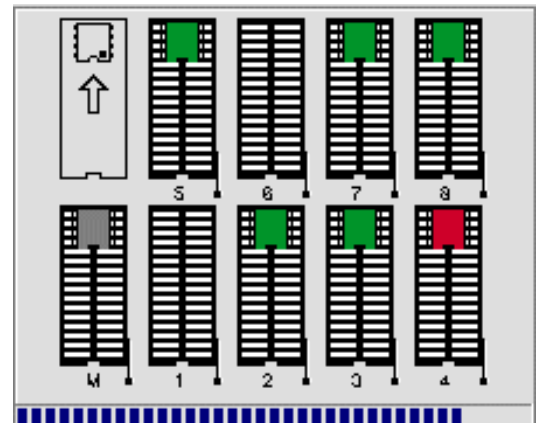
Module Status Display

The Module Status Display is a graphic representation of the state of the devices currently inserted in the module. The display matches the tri-color LEDs on the programmer panel. The socket graphics are dynamic and change to match the installed module. A PLCC socket module will show PLCC sockets, a DIP module will show DIP sockets and so forth, with the socket positions matching the module sockets.

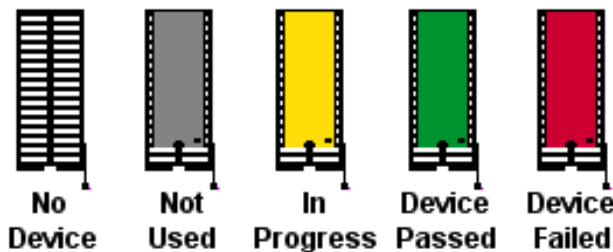
An Alignment graphic indicates proper insertion of the currently selected device.

Sockets containing devices actually show the device in the proper position. The color of each device graphic indicates the state of that particular device both during and after a given operation.

After completion of an operation, the final status display will remain active until the start of a new operation so that the user can determine the status of each device while removing it.



Clicking on a Socket will display status information after operation is completed



Socket Status Color Legend

DO NOT remove devices while status is 'In Progress' (Yellow).

If device(s) are inserted improperly, all sockets will indicate failure.

Progress Bar



The Progress Bar indicates the percentage of completion of an operation. During operations involving transfer of data to/from the WinLink editor buffer (Program, Verify, Read) the Progress Bar will increment smoothly.

During operations where Winlink is essentially monitoring a programmer operation (Copy, Compare, Chksum, Blank), The Progress Bar is slaved to the LCD Progress Bar on the programmer and will increment along with the programmer bar in a series of jumps.

Programmer Settings

This button group controls various programmer settings.

Project - Save/Load programmer panel settings

Device - Select a Device

Options - Setup Address Range, Auto-Options, Device Options, Serializer

Editor - Edit hex data in the buffer

Sequencer - Setup operations to be performed with the RUN button

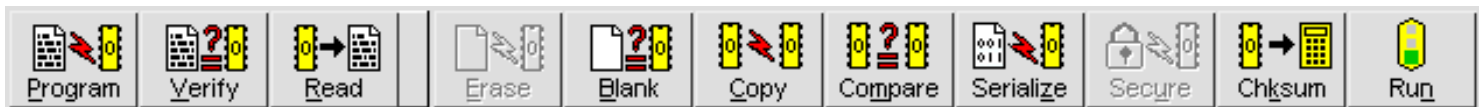
Lock - Setup Panel Lock levels



Click a button for details

These buttons have identical Main Menu functions.

Device Operations



Click a button for details

This button group initiates operations on devices.

Program Program Editor Buffer data into TARGET devices

Verify Verify TARGET devices against Editor Buffer data

Read Read MASTER or TARGET device data into Editor Buffer

Erase Erase entire device to blank state (if available)

Blank Check entire device for blank state

Copy Copy MASTER device data to TARGET devices

Compare Compare TARGET devices against MASTER device

Serialize Program Serialization data into TARGET devices

Secure Program TARGET device options (if available)

Chksum Calculate 32-bit checksum of MASTER or TARGET device data

Run Execute the operations stored in the Sequencer

Device Status

```
Manf :   ATMEL
Device : AT24C256(3V)
Range  : 00000000-000001FF(H)
Chksum : 0040066F(H)
```

Manf Manufacturer of currently selected device

Device Part Number of currently selected device

Addr Range Active operation address range

Chksum 32-bit checksum of Editor Buffer data within selected Address Range

Click on Chksum to recalculate the Editor Buffer Chksum.

File/Editor Startup & ComPort Status

File Filename of most recently loaded data file.

FileLoad Action performed on Editor Buffer prior to each file load.
Go to MAIN MENU FILE | WINLINK PREFERENCES to modify.

Editor Start Action performed on Editor Buffer when WinLink starts.
Go to MAIN MENU FILE | WINLINK PREFERENCES to modify.

Com Port Current com port, baud rate and connection status

Click on Com port: to go directly to the Connection Manager in order to select another ComPort or change the baud rate..

The Online (green) indicator shows when Winlink has control of the currently selected ComPort. Offline (red) indicates Winlink is NOT controlling the ComPort. Click on Online (or Offline) to Open/Close the ComPort currently in use. This allows another program to use the port without closing Winlink.

The Link indicator (green) indicates Winlink is properly connected to the programmer. The Link indicator (red) indicates the programmer is NOT connected, power is off, etc.

```
File : ..\..\..\files\1m_90a4.hex
File Load : Pre-Fill 0's
Editor Start : Reload File
Com port : Com1@115200  Online  Link
```

Device Counter

The Device Counter allows the user to track the number of devices programmed. The counter is only updated for Program and Copy operations. Devices are counted as passed only if all operations in the operation chain are successful. For example, if Post-pgm verify is enabled and Auto-serialize is enabled, the device must pass both of these to be counted as passed. Likewise when using the sequencer, the device must pass all operations to be counted as passed.

The Counter must be enabled to operate.

Device Counter
Pass : 0046
Fail : 0008
Total : 0054

Pass Number of devices successfully programmed.

Fail Number of devices that failed to program.

Total Total devices attempted to program

Auto Options

This panel displays the current Auto-Option settings. These are operations that will be automatically performed before or after Copy or Program. Using these, the user can initiate a chain of operations by pressing a single button. If a device fails an operation in the chain, it is removed from subsequent operations.

Auto Options
Erase Serialize
Blank Secure
Verify Verify +/-

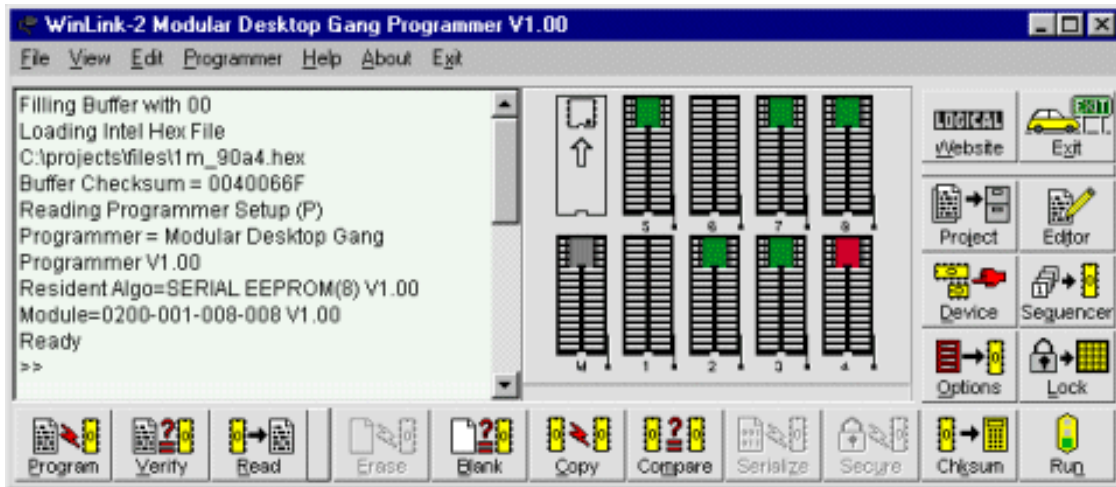
The Verify +/- option applies to the isolated Verify and Compare operations as well as Verify or Compare in chained operations.

Front Panel Views

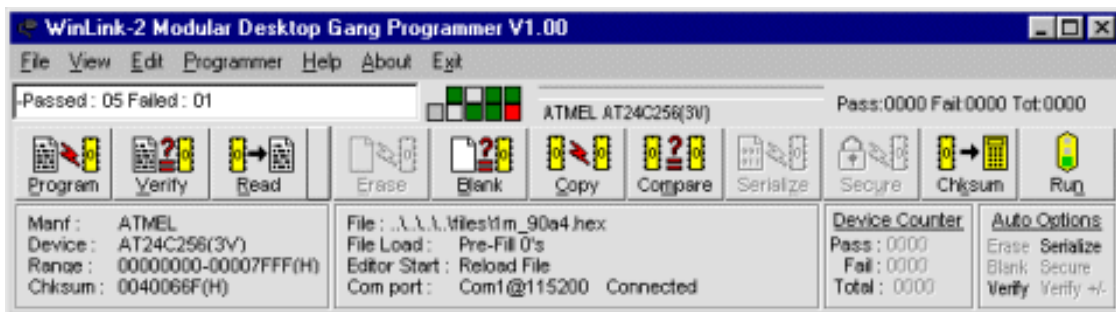
There are several available front panel views designed to suit the needs of different users. Views can be switched 'on-the-fly' without loss of information. Select a view by clicking the desired item under "VIEW" on the main menu. The current view is indicated by a dot next to the name.

Normal View

Info View



Status View



Minimal View



LOGICAL Help

Updating the Programmer Operating Code

Update Programmer Base

You can update the programmers Operating System by clicking
MAINMENU: PROGRAMMER: UTILITIES: Update Pgm'er Base.

Click START, and follow the on screen directions.

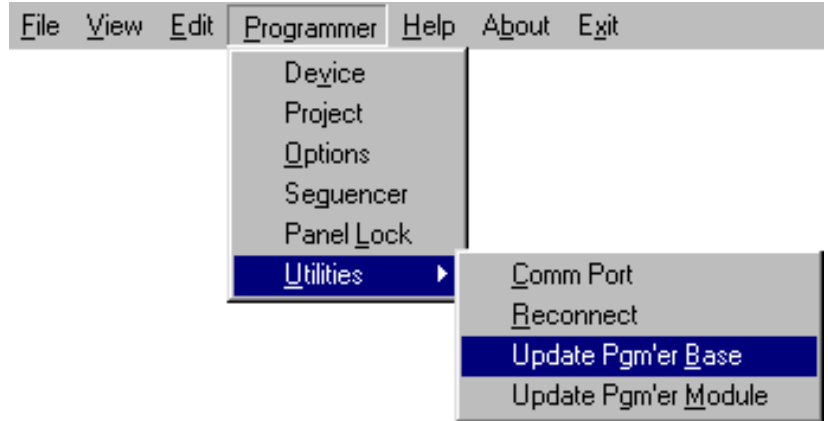
If you are updating a Gangpro-8XP or a ChipKopier unit, choose the following file

kernel_gpx_vxxx.upd

If you are updating a Shooter-XP, choose the following file

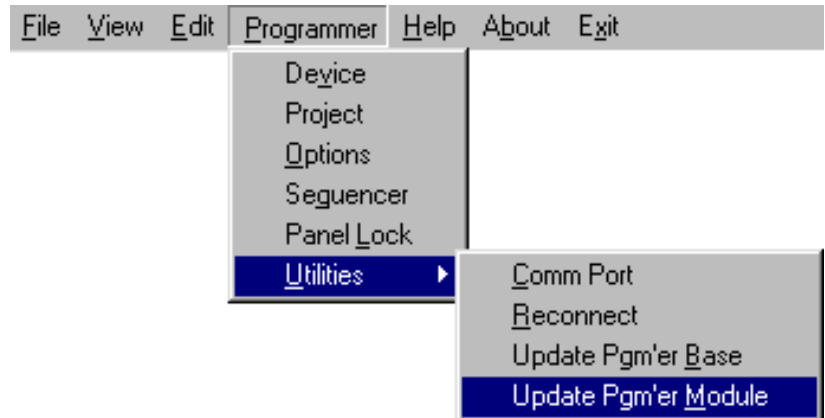
kernel_sxp_vxxx.upd

In either case vxxx = the version number and will vary. Choose the higher version if multiple versions are present.

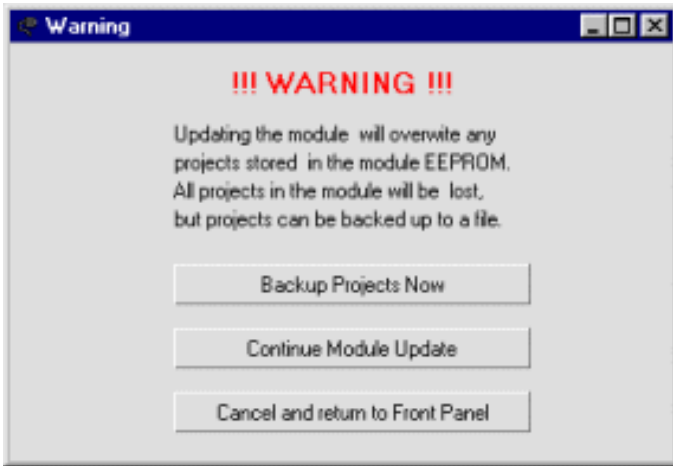


Update Programmer Module

You can update the programmer module's Device Support Program Code by clicking
MAINMENU: PROGRAMMER: UTILITIES: Update Pgm'er Module.



The following warning appears.

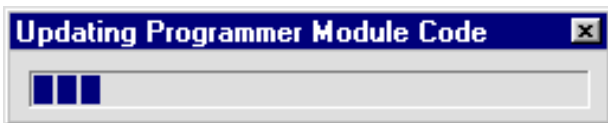
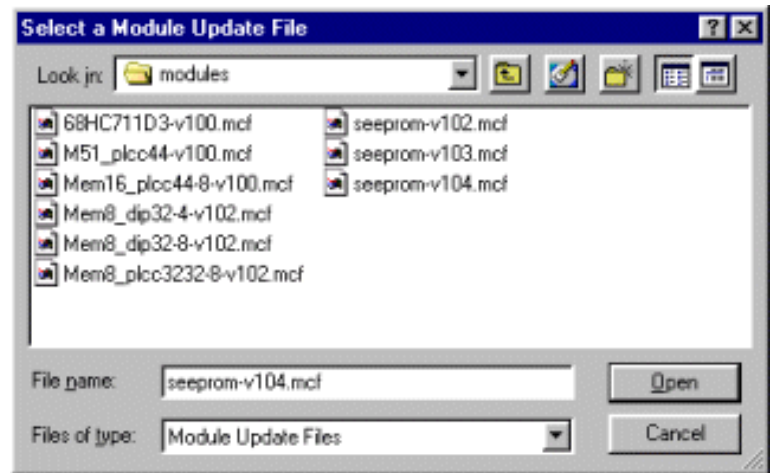


The programmer projects, selected device and options are stored in the module memory and will be erased during the update. You can back up this information by clicking the first option, otherwise click CONTINUE MODULE UPDATE.

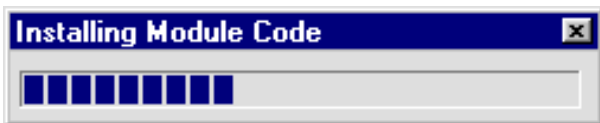
[Details on Saving Projects](#)

Navigate to the location of the module update file. Note that module updates have the file extension .MCF. If you just upgraded the entire Winlink Application, the File Open dialog will automatically show the proper folder. If you are updating just the module file after downloading, from diskette or email attachment, navigate to the folder in which you saved the provided update file.

Choose the proper [file](#) to update the module and click OPEN. The update process will begin automatically.



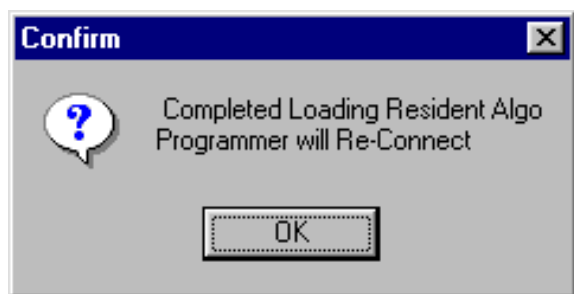
The Update progress bar appears while the data is written to the module EEPROM.



The Install progress bar appears while the data is copied from the module EEPROM to the programmer's FLASH.

The entire update and install process should take less than 45 seconds depending on the size of the update code.

The following message indicates successful module update.



Click OK and the programmer will reconnect using the new module code.

Verify the proper module and version is displayed in the Winlink Message Window.

Example.....Resident Algo=MEMORY 8-bit(32) V1.02

If you previously saved your Programmer Projects it is safe to restore them now.

[Details on Restoring Projects](#)

Module Algorithm File Names

Use the table below to determine the device support algorithm filename required by your module. These files are located by default in c:\Program Files\Logical Devices\Winlink2\modules. The 'Vxxxx' refers to the version number and the filenames installed on your system will contain the actual version number. Use the highest version number available.

Module	Pkg	Number of Target Sockets		
		1 (Chip Kopier/Shooter-XP)	4 (Shooter-4C)	8/16/32 (Gangpro 8/16/32XP)
8-bit Memory	DIP-32	MEM-8 DIP-32-1-Vxxxx.mcf	MEM-8 DIP-32-4-Vxxxx.mcf	MEM-8 DIP-32-8-Vxxxx.mcf
8-bit Memory	PLCC-32/28	MEM-8 PLCC-3228-1-Vxxxx.mcf	MEM-8 PLCC-3228-4-Vxxxx.mcf	MEM-8 PLCC-3228-8-Vxxxx.mcf
8-bit Memory	PLCC-32/32	MEM-8 PLCC-3232-1-Vxxxx.mcf	MEM-8 PLCC-3232-4-Vxxxx.mcf	MEM-8 PLCC-3232-8-Vxxxx.mcf
16-bit Memory (Jedec)	DIP-40	MEM-16 DIP-40-J-1-Vxxxx.mcf	MEM-16 DIP-40-J-4-Vxxxx.mcf	MEM-16 DIP-40-J-8-Vxxxx.mcf
16-bit Memory (Jedec)	PLCC-44	MEM-16 PLCC-44-J-1-Vxxxx.mcf	MEM-16 PLCC-44-J-4-Vxxxx.mcf	MEM-16 PLCC-44-J-8-Vxxxx.mcf
16-bit Memory (MROM)	DIP-48	MEM-16 DIP-48-M-1-Vxxxx.mcf	MEM-16 DIP-48-M-4-Vxxxx.mcf	MEM-16 DIP-48-M-8-Vxxxx.mcf
8051	PLCC-44	M51 PLCC-44-1-Vxxxx.mcf	M51 PLCC-44-J-4-Vxxxx.mcf	M51 PLCC-44-J-8-Vxxxx.mcf
68HC705C8/C9A	PLCC-44	hc705Cx PLCC-44-1-Vxxxx.mcf	hc705Cx PLCC-44-4-Vxxxx.mcf	hc705Cx PLCC-44-8-Vxxxx.mcf
68HC705P6A	SOIC-28	hc705p6 SOIC-28-1-Vxxxx.mcf	hc705p6 SOIC-28-4-Vxxxx.mcf	hc705p6 SOIC-28-8-Vxxxx.mcf
68HC711D3	PLCC-44	hc711D3 PLCC-44-1-Vxxxx.mcf	hc711D3 PLCC-44-4-Vxxxx.mcf	hc711D3 PLCC-44-8-Vxxxx.mcf
68HC711Exx	PLCC-52	hc711Exx PLCC-52-1-Vxxxx.mcf	hc711Exx PLCC-52-4-Vxxxx.mcf	hc711Exx PLCC-52-8-Vxxxx.mcf
68HC908GP32	SDIP-42	hc908 SDIP-42-1-Vxxxx.mcf	hc908 SDIP-42-4-Vxxxx.mcf	hc908 SDIP-42-8-Vxxxx.mcf
Serial E/EPROM	DIP-8	seeprom DIP-32-1-Vxxxx.mcf	seeprom DIP-32-4-Vxxxx.mcf	seeprom DIP-32-8-Vxxxx.mcf

PIC-01 (12F6xx/16F6xx)	DIP-8/14/20	Pic01 DIP-32-1-Vxxxx.mcf	Pic01 DIP-32-4-Vxxxx.mcf	Pic01 DIP-32-8-Vxxxx.mcf
PIC-03 (16F87xA,16F72,16C62B)	SOIC-28	Pic03 SOIC-28-1-Vxxxx.mcf	Pic03 SOIC-28-4-Vxxxx.mcf	Pic03 SOIC-28-8-Vxxxx.mcf
PIC-04 (16F57,16C55A)	SOIC-28	Pic04 SOIC-28-1-Vxxxx.mcf	Pic04 SOIC-28-4-Vxxxx.mcf	Pic04 SOIC-28-8-Vxxxx.mcf
PIC-05 (16C54/A/B/C)	DIP-18/32	Pic05 DIP-32-1-Vxxxx.mcf	Pic05 DIP-32-4-Vxxxx.mcf	Pic05 DIP-32-8-Vxxxx.mcf

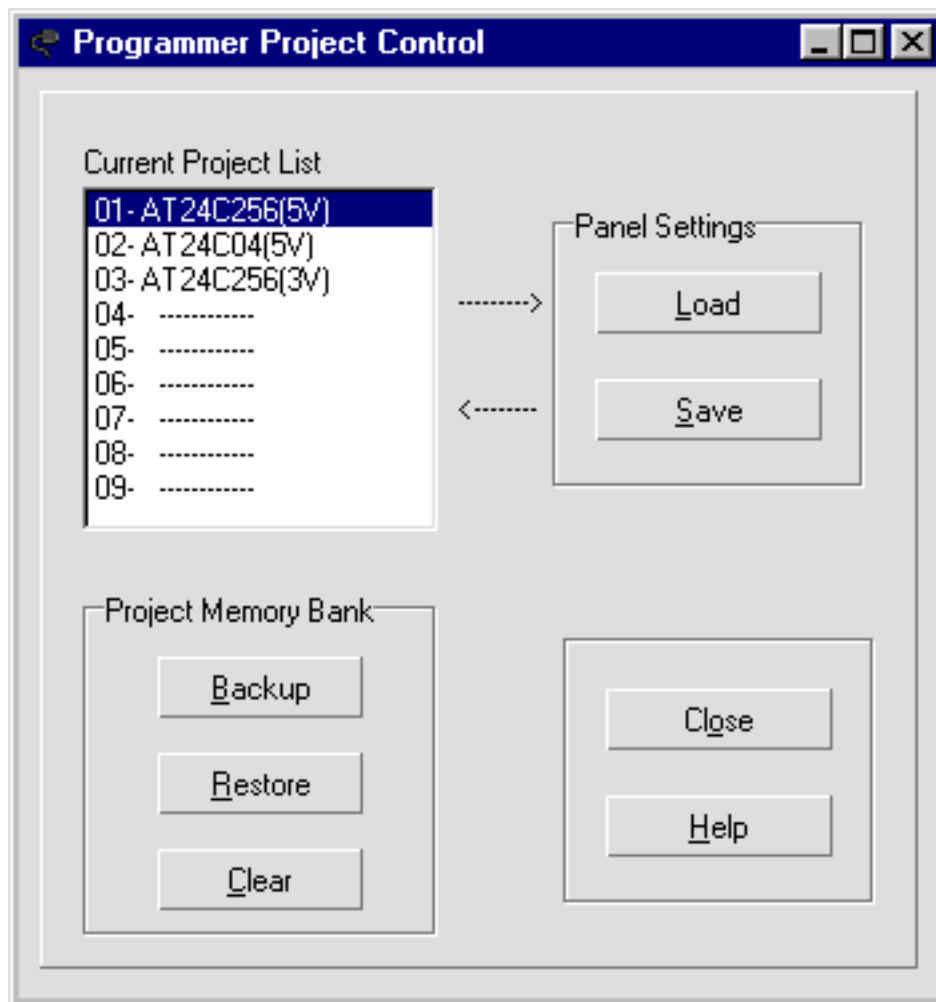
LOGICAL Help

PROGRAMMER PROJECTS

The PROJECT manager stores up to 9 instances of the complete PROGRAMMER settings including the SERIALIZER options. These can be instantly recalled to completely reconfigure the PROGRAMMER.

The PROJECTs are resident in the PROGRAMMER MODULE and can be recalled without connection to a PC. Each MODULE can store the current state of the PROGRAMMER and up to 9 PROJECTS so switching to another module allows access to a new PROJECT bank specific to that MODULE.

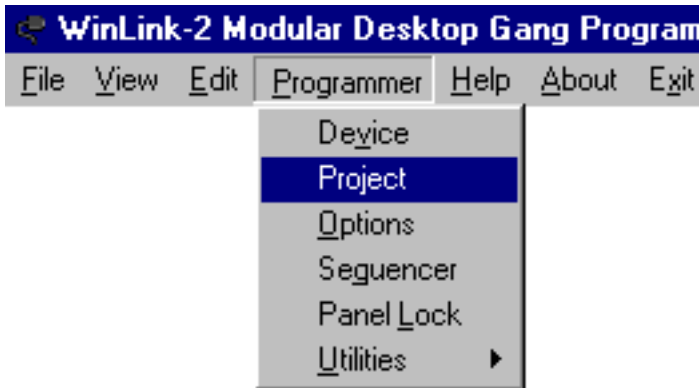
Additionally, the entire PROJECT memory bank for any MODULE can be saved to/restored from a file (if connected to a PC).



Accessing



You can access the PROJECT manager by pressing the PROJECT button.



You can access the PROJECT manager by clicking
MAINMENU: PROGRAMMER: PROJECT.

Changing

Storing a Project

In the PROJECT LIST window, click on and highlight the PROJECT to which the PROGRAMMER settings will be stored.

Click the STORE button.

Recalling a Project

In the PROJECT LIST window, click on and highlight the PROJECT from which the PROGRAMMER settings will be recalled.

Click the RECALL button.

Backing up the Project List

Click the BACKUP button...

Select a Filename for the backup file...

Click SAVE in the file window.

If you use multiple modules, it is suggested that you choose a filename so that you can remember the association of the file to the MODULE.

Restoring the Project List

Click the RESTORE button...

Select a backup file name...

Click RESTORE in the file window.

The file will be checked to determine if it is a backup of the CURRENT MODULE type. You cannot restore a backup for MODULE type other than the installed MODULE.

Clearing the Project List

Click the "CLEAR" button. The entire list will be cleared and cannot be recovered except from a backup file.

LOGICAL Help

WinLink features

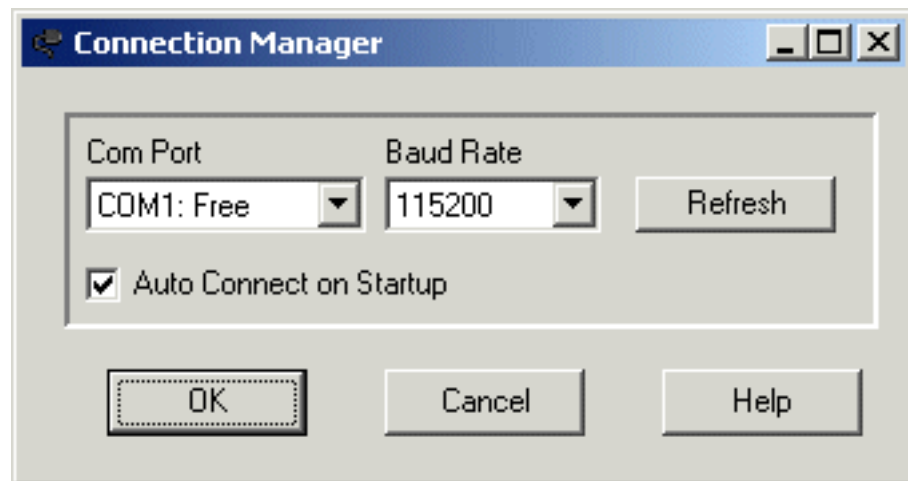
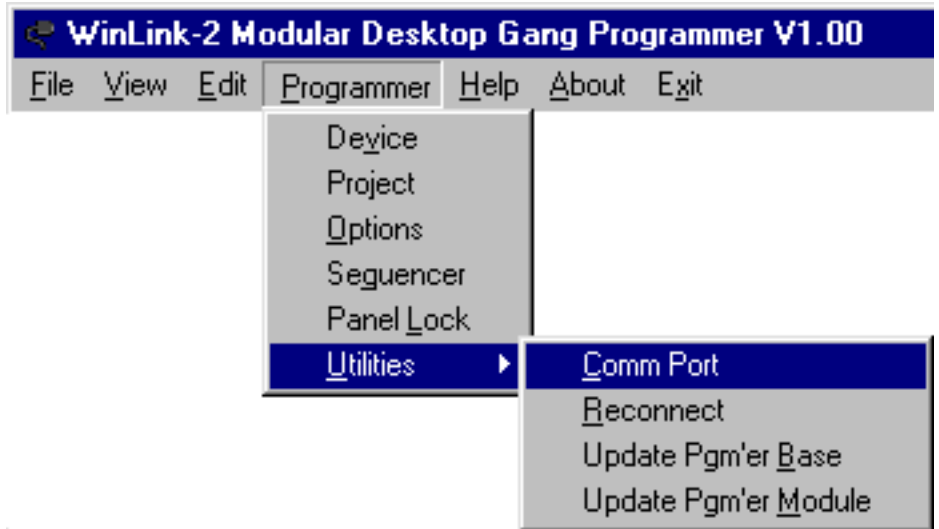
Comm Port Connection Manager

You can setup the COMM PORT by clicking...

MAINMENU: PROGRAMMER:
UTILITIES: Comm Port.

You can also click directly on the 'Com Port:' label on the Front Panel.

When the Connection Manager opens, the PC is scanned for free Com Ports and these listed are in the Com Port drop down list. Com Ports numbers from 1 to 256 are scanned and only existing physical ports are listed. This includes USB/SERIAL ADAPTORS and other devices (MODEMS) that respond correctly to to serial port OPEN commands.



Com Port

Pull down the drop down list and select the desired Comm Port.

Choose a port that will not conflict with other devices in your PC. Even though only existing ComPorts are listed, a given port may be a modem or other serial device that is not currently open.

Baud Rate Pull down the drop down list and select the the desired Baud Rate.

The PROGRAMMER will auto-detect Winlink's BAUD RATE so no configuration of the PROGRAMMER is required.

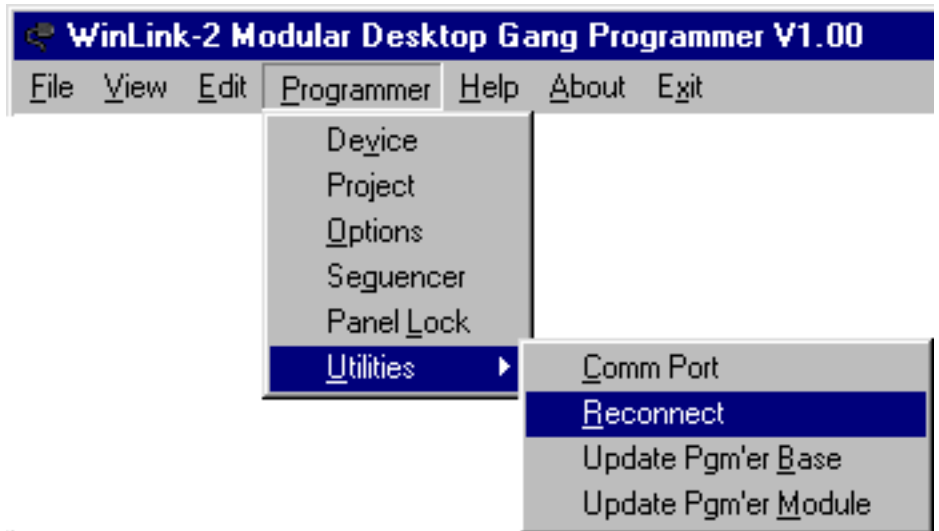
Normally, you should leave this set at 115200 for faster operation.

Refresh Click REFRESH to scan the PC for free serial ports and update the Com Port list.

AutoConnect on Startup Uncheck Autoconnect to prevent Winlink from automatically connecting to the programmer on startup.

Programmer Reconnect

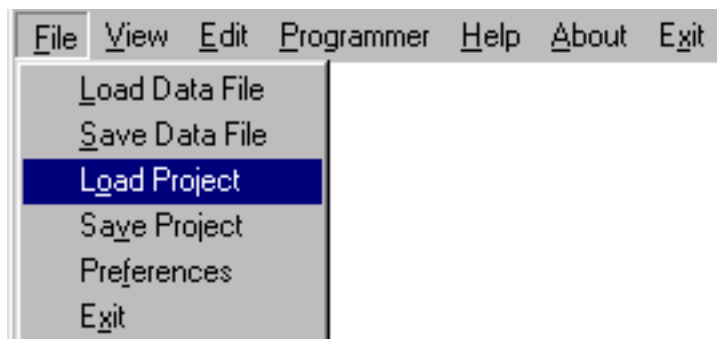
You can force reconnection and retrieval of current settings from the programmer by clicking



MAINMENU: PROGRAMMER: UTILITIES: Reconnect.

Winlink Projects

You can save/restore the WinLink configuration to/from a file.



Save Project

Click...MAINMENU: FILE: Save Project.

Choose a filename and click "SAVE"..

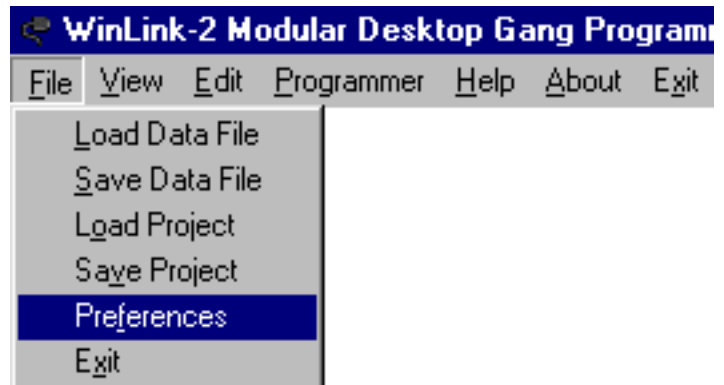
Load Project

Click...MAINMENU: FILE: Load Project.

Choose a filename and click "OPEN"..

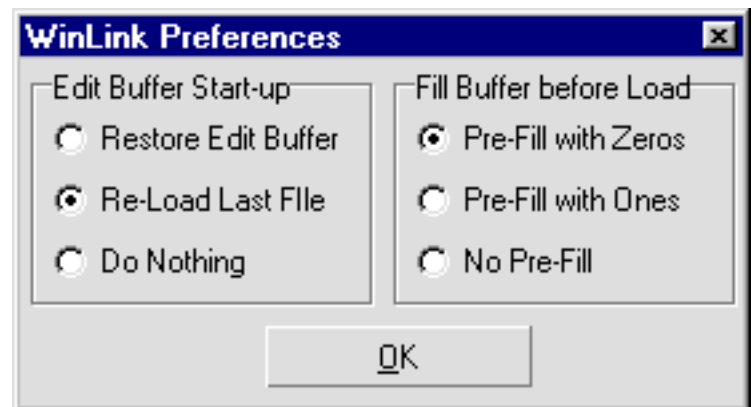
Winlink Preferences

You can set start-up options for the Editor buffer and also select a buffer action which occurs just prior to loading a data file.



Buffer Start Click the button corresponding to the the desired Buffer Startup Action.

- Restore Editor - restore the Editor from the previous session.
- Reload last File - reload the last file previously loaded.
- Do Nothing - take no action.



Buffer Pre-load Fill

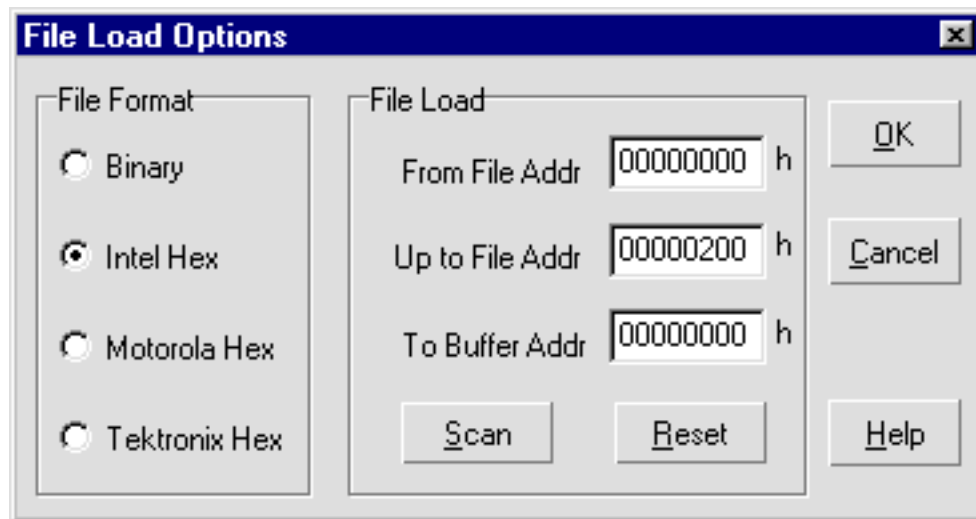
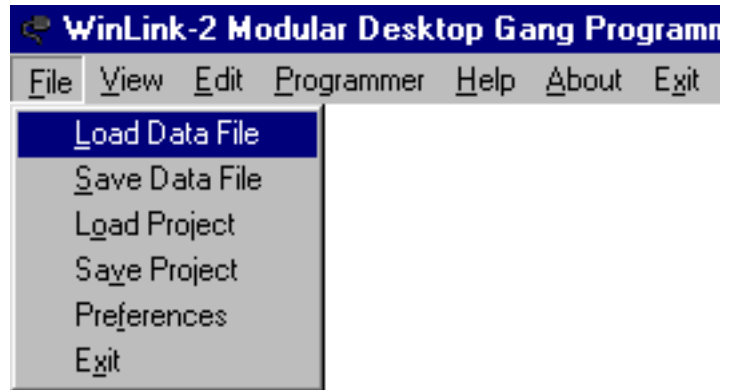
Click the button corresponding to the the desired Pre-fill Action.

- Fill with zeros - Fill buffer with zeros before loading data file.
- Fill with ones - Fill buffer with ones before loading data file.
- No pre-fill - take no action.

Load Data File

Click...MAINMENU: FILE:Load Data File.

Choose a filename and click "OPEN"..



File Format

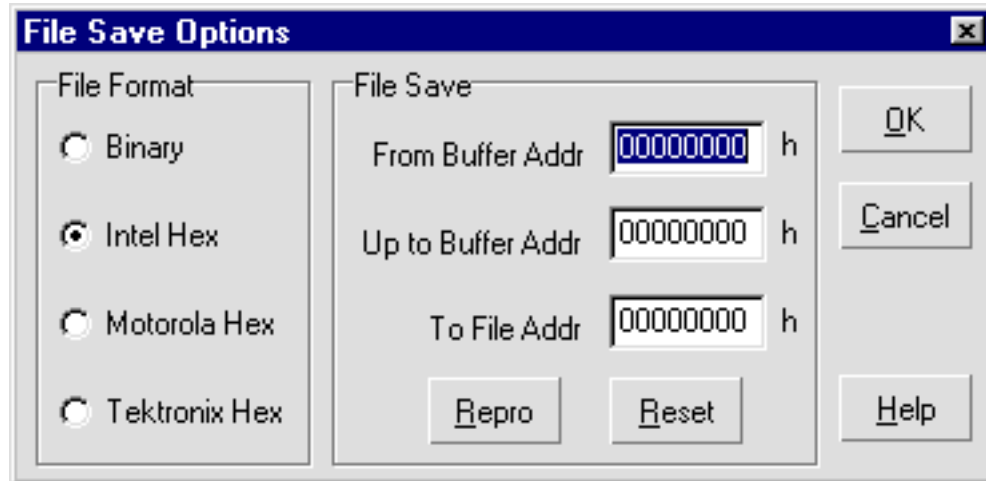
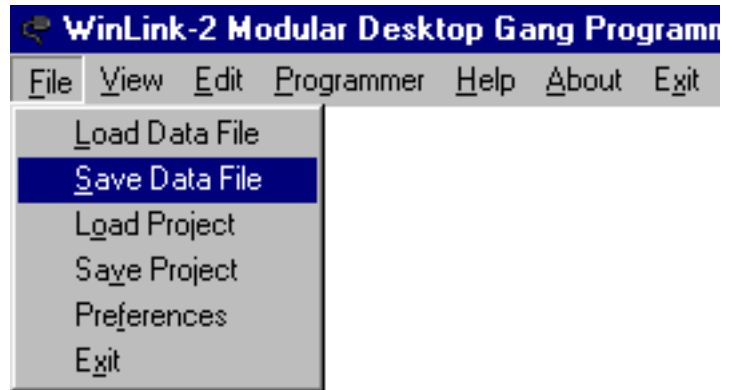
Winlink will detect the file format. If you want to override this, click the desired file format type.

- From File Address** This sets the start address of the file filter and indicates the file address at which data will begin to fill the Editor buffer.
In conjunction with File up to Address, useful for extracting a contiguous section of a file.
- File up to Address** This sets the end address of the file filter and indicates the file address at which data will stop filling the Editor buffer.
In conjunction with From File Address, useful for extracting a contiguous section of a file.
- NOTE : If using File Split, this value MUST be large enough for the ENTIRE file (i.e. \geq the size of the device times the number of devices). Use the RESET button to automatically set this value based on the number of split devices and the device size or the SCAN button to detect the actual size of the file.
- To Buffer Address** This sets the start address of the Editor buffer fill from the file data.
- Scan** Examine the file and make best guess at where file data should be located in the buffer (and therefore the TARGET device(s)).
- Reset** Reset the File Filter and buffer addresses to their default values for the selected device.
This is normally the entire address range for the selected device.
- Note : If using File Split, RESET will setup the addresses based on device size and the number of split devices.

Save Data File

Click...MAINMENU: FILE: Save Data File.

Choose a filename and click "SAVE"..



File Format

Winlink will use the previously used file format. If you want to override this, click the desired file format type.

From Buffer Address

This sets the start address of the buffer filter and indicates the buffer address at which data will be retrieved.

Up to Buffer Address

This sets the end address of the buffer filter and indicates the buffer address at which data retrieval will stop.

To File Address

This sets the address which is added to each file record.

Repro

Set the File Save Options to produce a file with the same characteristics as the original loaded file.

Reset

Reset the Buffer Filter and File addresses to their default values.

LOGICAL Help

PANEL LOCK

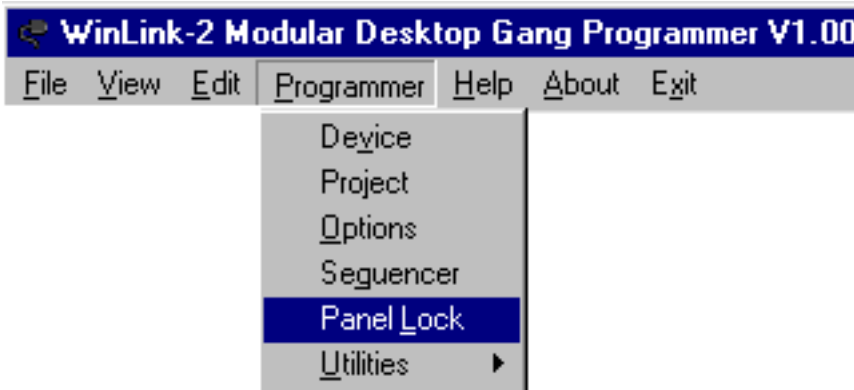
The PANEL LOCK feature allows selective lock-out of WinLink/Programmer features. This allows inexperienced users to operate the programmer with a minimum of training. At the highest level, only a single button (RUN) is operative and this executes a preset series of operations.

Accessing

Regardless of the current LOCK settings, you can always access LOCK through the MAIN MENU.



You can access the PANEL LOCK by pressing the LOCK button.



You can access the PANEL LOCK by clicking MAINMENU: PROGRAMMER: PANEL LOCK.

Password

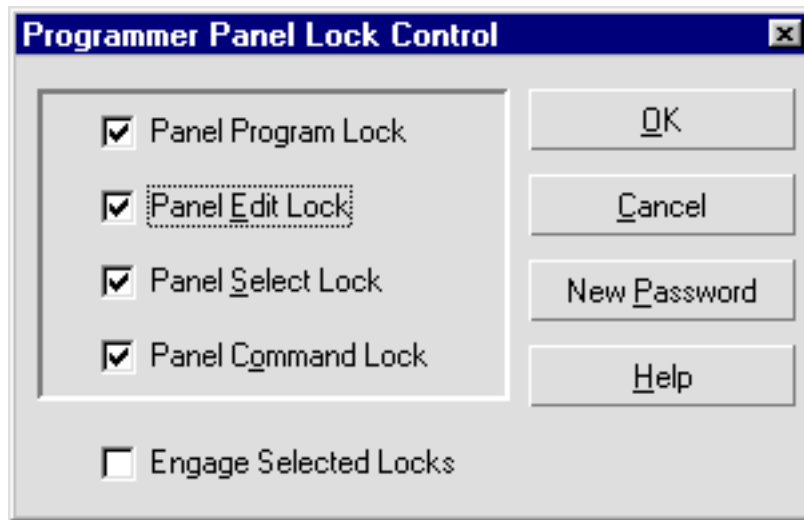
You MUST know the current PASSWORD to access the LOCK mechanism. Enter the current PASSWORD (default= 00000000).



Changing

The PANEL LOCK settings are stored within the PROGRAMMER and any modifications made will remain in effect until you change them again.

Levels



Click the checkbox for the LOCK LEVEL you want to set

Engage

To engage the LOCK LEVELS selected..

Check the "Engage Selected Locks" box.

Click "OK".

Disengage

To disengage the LOCK LEVELS selected..

Uncheck the "Engage Selected Locks" box.

Click "OK".

Password

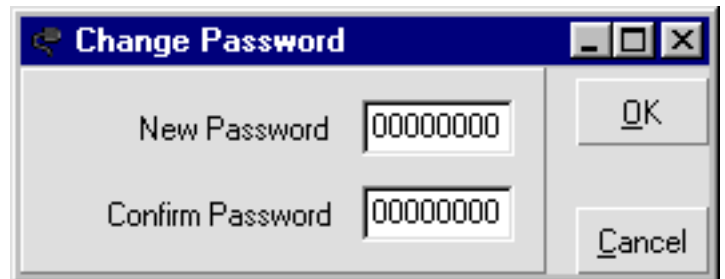
To change the current PASSWORD...

Click "New Password".

Enter the new PASSWORD in the "New Password" box.

Enter the new PASSWORD in the "Confirm Password" box.

Click "OK".



Panel Lock Chart

This chart shows which functions and operations are affected by each LOCK LEVEL.

Function	Program Lock	Edit Lock	Select Lock	Command Lock
Program	INHIBIT	X	X	INHIBIT
Verify	X	X	X	INHIBIT
Read	X	X	X	INHIBIT
Erase	X	X	X	INHIBIT*
Blank	X	X	X	INHIBIT*
Copy	X	X	X	INHIBIT*
Compare	X	X	X	INHIBIT*
Serialize	X	X	X	INHIBIT*
Secure	X	X	X	INHIBIT*
Run	X	X	X	X
Project	X	X	INHIBIT	X
Device	X	X	INHIBIT	X
Options	X	INHIBIT	X	X
Editor	X	INHIBIT	X	X

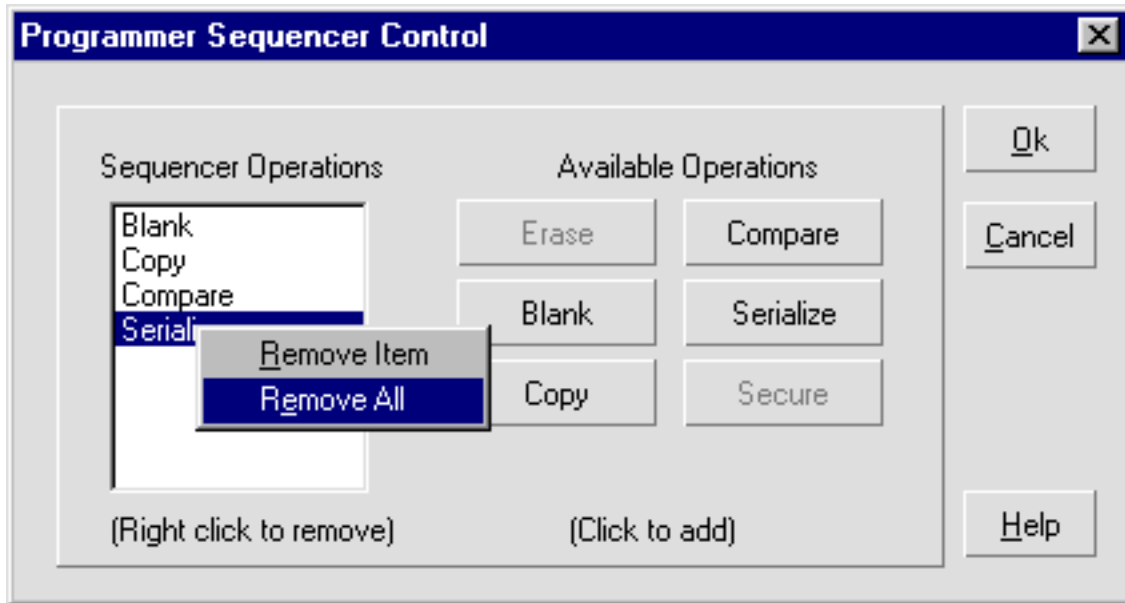
X=no effect.

*=still allowed in SEQUENCER RUN.

LOGICAL Help

SEQUENCER

The SEQUENCER feature allows a series of PROGRAMMER operations to execute by pressing a single button. In conjunction with PANEL LOCK, this allows inexperienced users to operate the programmer with a minimum of training.

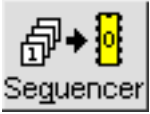


The SEQUENCER is intended primarily for stand alone use (not connected to PC) of the programmer by inexperienced operators.

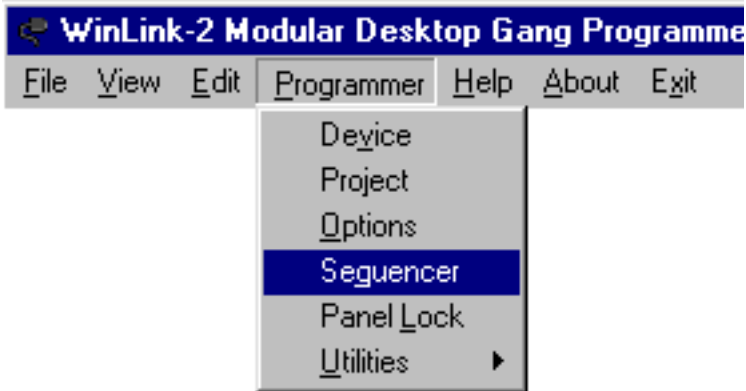
Only these operations are allowed..

- ERASE
- BLANK
- COPY
- COMPARE
- SERIALIZE
- SECURE

Accessing



You can access the SEQUENCER by pressing the SEQUENCER button.



You can access the SEQUENCER by clicking
MAINMENU: PROGRAMMER: SEQUENCER.

Changing

The SEQUENCER settings are stored within the PROGRAMMER and any modifications made will remain in effect until you change them again.

Adding Operations

Click the button for the OPERATION you want to add to the SEQUENCER.

If ERASE or SECURE is not a valid operations for the currently selected device, these buttons will be grayed and the OPERATIONS not available.

If SERIALIZE is added to the SEQUENCER OPERATIONS, it will not execute unless the SERIALIZER is configured and enabled.

Removing Operations

Right Click the the OPERATION you want to remove in the SEQUENCER OPERATIONS window. Click "Remove Item" in the pop-up menu.

Clearing

Right Click the the SEQUENCER OPERATIONS window. Click "Remove All" in the pop-up menu.

Move

To move the position of a SEQUENCER OPERATION...

In the SEQUENCER OPERATIONS window, click and drag the OPERATION to the desired position.

Using

Running

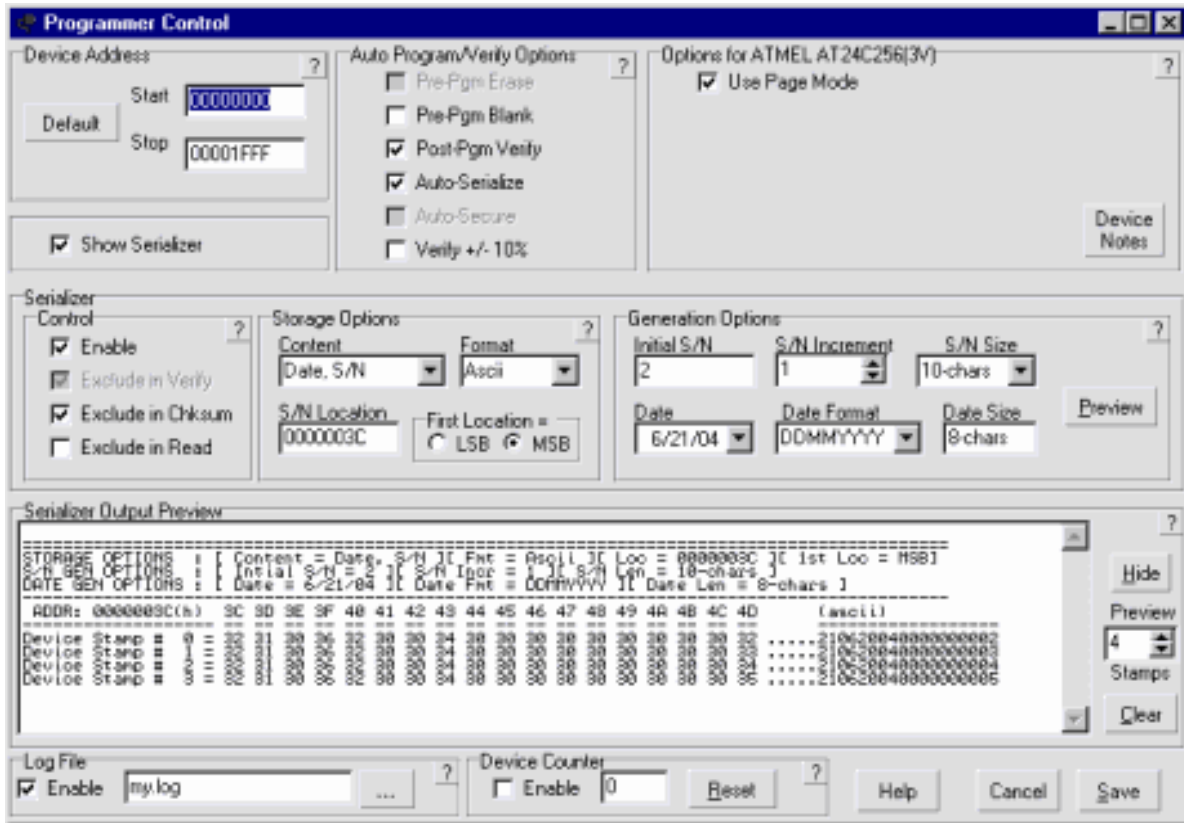
Click the RUN button to execute the OPERATIONS stored in the SEQUENCER.

Note that for the COPY OPERATION, the PROGRAMMING OPTIONS do NOT apply inside the SEQUENCER. ONLY the actual programming component of the COPY OPERATION is performed by the SEQUENCER.

For the SERIALIZE OPERATION to execute, the SERIALIZER must be configured and enabled. If not enabled, the SERIALZE OPERATION will be skipped.

LOGICAL Help

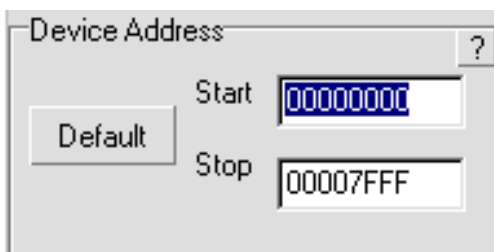
PROGRAMMER OPTIONS



Click on a section to view help on that section.

Address Range

The ADDRESS RANGE defines the valid device addresses for most operations. SERIALIZATION has it's own separate address space and SECURE normally uses specific defined addresses within the device. The ADDRESS RANGE is automatically set to the selected device's first and last address when a new device is selected. Loading a PROJECT will set ADDRESS range to the range stored in the PROJECT. Installing a module will set ADDRESS range to it's value when that module was last used.



For normal operation, it is not necessary to modify the ADDRESS RANGE.

If START and STOP are set to the same value, operations will affect a single byte in the TARGET devices. Addresses are entered in HEXADECIMAL.

Start

[default= 1st device addr]

The starting device address for all operations. It must be greater than or equal to the first address in the device (typically 0) and less than or equal to the STOP ADDRESS.

Stop

[default=last device addr]

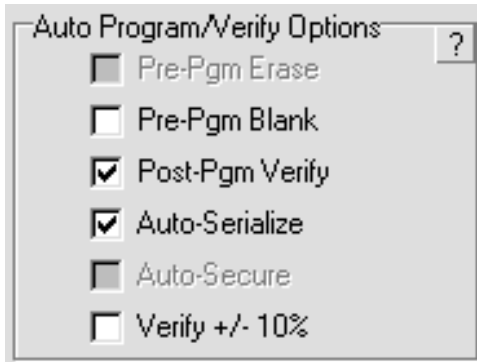
The ending device address for all operations. It must be greater than or equal to the START ADDRESS, and less than or equal to the last (highest) address in the device.

Default

This button will reset START and STOP to their default values.

Auto-Programming/Verify Options

These options control execution of the indicated operation(s) before and after the actual device programming step inside the PROGRAM/COPY operations. They allow the user to perform a chain of common operations by pressing a single button. For example, checking PRE-



PGM BLANK will cause the TARGET devices to be BLANK checked before programming each time PROGRAM or COPY is pressed. The exception is VERIFY +/- which causes the VERIFY or COMPARE operations to execute twice, first with the TARGET device VCC voltage at -10% of the nominal value and then again at VCC + 10% of nominal.

If a device fails any operation in the chain, it will be removed from further operations in the chain, but the failing status is available by clicking on the SOCKET.

If particular options are not available for the currently selected device, the option will be grayed out and inaccessible.

Pre-Pgm Erase

[default=per device]

If available and enabled, this operation automatically erases the TARGET device(s) to their BLANK STATE prior to any other operations in the chain.

Pre-Pgm Blank

[default=per device]

This operation if enabled, automatically compares data at each location in the TARGET device(s) with their BLANK STATE. It will execute after ERASE (if enabled) and prior to the programming operation.

Post-Pgm Verify

[default=per device]

This operation if enabled, automatically compares data at each location in the TARGET device(s) with the data that was programmed into that location. It will execute after programming and prior to any remaining operations in the chain. Note that for COPY, the programming data originated in the MASTER device and for PROGRAM, the data originated in the WinLink Editor buffer.

Auto-Serialization (Legacy < v1.40)

[default= off]

If enabled, this operation will automatically program the TARGET device(s) with the output from the SERIALIZER, essentially electronically stamping each device with a unique code. Note that the SERIALIZER itself must be enabled and setup properly before this option is accessible. In order to receive the electronic stamp, the device must have passed all prior operations in the chain. If a device fails the action of stamping it, the device status will be set to FAIL, the device will be removed from the chain and the same electronic stamp will then be applied to the next device.

Auto-Secure

[default= off]

If available and enabled, this operation will automatically burn any enabled device options into the TARGET device(s). The specific options are entirely device dependent and are dynamically displayed for the currently selected device.

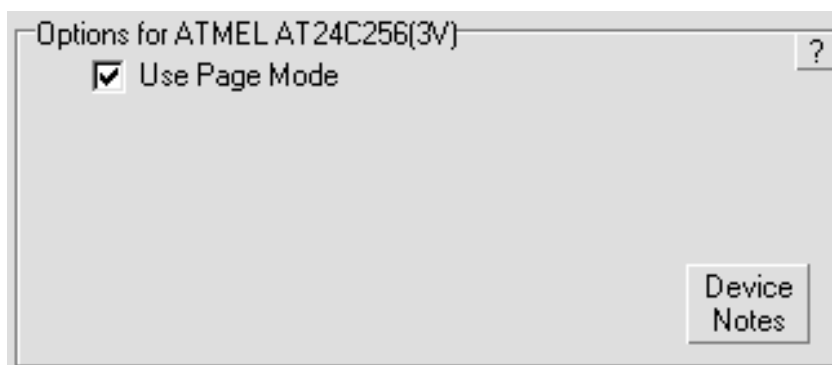
Verify +/-

[default= off]

If enabled, this option causes COMPARE or VERIFY to execute in two passes, first with device VCC= -10% of the nominal value, then with VCC=+10% of the nominal value. If the device fails either pass, it will be marked FAIL and removed from the chain. Note that this option will also modify the behaviour of COMPARE/VERIFY even if initiated as an isolated operation outside COPY/PROGRAM.

Device Specific Options

These options are completely device dependent and are dynamically obtained from the programmer. They will change depending on the selected device and zero to sixteen checkboxes labeled with appropriate captions may appear in this section.



In general, the DEVICE SPECIFIC OPTIONS take one of two forms. First, a DEVICE SPECIFIC OPTION may affect how the programmer carries out some operation. Devices such as SERIAL EEPROMs may have the capability of programming using page mode and this selection is available to the user as a device option.

Secondly, devices such as FLASH memory may contain sector

protection or write protect mechanisms and these are available as device options. Such mechanisms require specific programming steps to set them properly and often render the device unusable for further programming. This form of DEVICE SPECIFIC OPTION has no effect until the SECURE operation is performed either as an isolated operation, in the SEQUENCER, or as AUTO-SECURE during PROGRAM/COPY.

Note that many devices such as UV EPROMs have no DEVICE SPECIFIC OPTIONS whatsoever and the DEVICE SPECIFIC OPTIONS section will be blank for these devices.

Device Notes

This button opens the user's default HTML browser and displays a DEVICE NOTES document containing information specific to the currently selected device or device family. Any DEVICE SPECIFIC OPTIONS are explained in detail in this document. Each DEVICE NOTES doc contains a header section which applies to the entire family of devices (e.g. 24C, 27C, etc) of which the currently selected device is a member. If the your device has unique pertinent information, you are taken directly to the entry for the device. If no unique information is available for your device, you are shown the device family section.

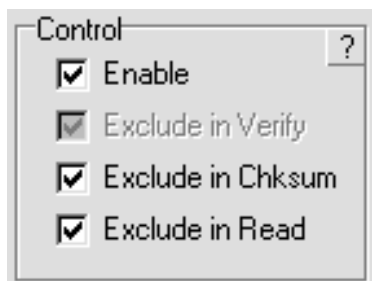
SERIALIZER

Due to a lack of user interest and to provide expanded features, the Serializer functionality has been removed in kernel V1.33 and above. The following remains for historical purposes only.

Serializer Control (prior to kernel V1.40 only)

The SERIALIZER panel is normally hidden. Check 'Show Serializer' to make the panel visible. This setting will now remain in effect until changed again.

The SERIALIZER options are divided into CONTROL, STORAGE, and GENERATION sections although there is some interaction and overlap between the sections. The CONTROL section selects if and under which conditions the SERIALIZER is active. If disabled, the serializer has no effect upon any operation even if the SERIALIZE operation is included in the SEQUENCER.



When a new device is selected, the SERIALIZER options are set to their default values. Recalling a PROJECT or installing another module will restore the SERIALIZER options to the values in the PROJECT or the values with which the new module was last used.

This allows the user to maintain different independent serialization schemes and to select between them.

Enable

[default=off]

This option **MUST** be checked for the SERIALIZER to operate. Note that the AUTO_SERIALIZE option will be disabled and unchecked if the SERIALIZER is disabled.

Exclude in Verify

[default= on]

In Winlink V1.00, this option is always on. When checked, ignore the DEVICE STAMP memory locations in VERIFY/COMPARE operations.

Exclude in Chksum

[default= on]


When checked, ignore the DEVICE STAMP memory locations in DEVICE CHKSUM operations. The DEVICE STAMP data will NOT contribute to the checksum. This assures that stamped devices will return the same chksum. When unchecked, the data in DEVICE STAMP locations will contribute to the checksum and stamped devices with identical data outside the DEVICE STAMP range will return different checksums.

Exclude in Read

[default= on]

When checked, ignore the DEVICE STAMP memory locations in READ operations. Addresses within the DEVICE STAMP range will return 0's so that a data file created from reading a stamped device will have a reference checksum that does not include the DEVICE STAMP data. When unchecked, a READ operation will return all data within the current ADDRESS RANGE including the DEVICE STAMP data. This allows you to verify the DEVICE STAMP for a given device.

Serializer Storage Options (prior to kernel V1.40 only)



The image shows a dialog box titled "Storage Options" with a question mark icon in the top right corner. It contains the following controls:

- Content:** A dropdown menu with "Date, S/N" selected.
- Format:** A dropdown menu with "Ascii" selected.
- S/N Location:** A text input field containing "0000003C".
- First Location =:** Two radio buttons: "LSB" (unselected) and "MSB" (selected).

The STORAGE options control the content, format, and location of the DEVICE STAMP data in the TARGET device(s).

Content

[default=S/N only]

This option controls what will be included in the DEVICE STAMP. To change the selection, pull down the combo box by clicking on the arrow, then click an option.

You can select from the following:

- S/N only ...DEVICE STAMP=Serial number only
- Date only ...DEVICE STAMP=Date only
- S/N,Date ...DEVICE STAMP=Serial number, followed by Date
- Date,S/N ...DEVICE STAMP=Date followed by Serial Number

Format

[default= Binary]

This option controls which format is used to represent the DEVICE stamp. To change the selection, pull down the combo box by clicking on the arrow, then click an option.

You can select from the following:

- Binary..DEVICE STAMP=binary number
- BCD.....DEVICE STAMP=BCD digits
- ASCII...DEVICE STAMP=ASCII characters

Location

[default= 00000000h]

This option controls the DEVICE STAMP start address location in the device. Enter a hexadecimal address somewhere within the device.

(The ADDRESS RANGE option is ignored by the SERIALIZER).

1st Location=

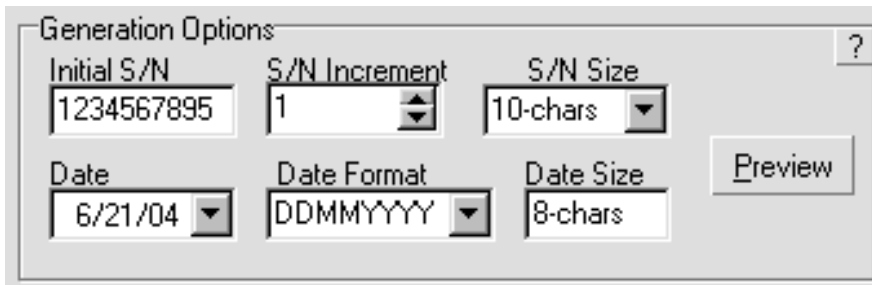
[default=LSB]

This option controls the storage order of the DEVICE STAMP data. The storage format of your target application determines this setting.

You can select from the following:

- MSB...DEVICE STAMP is stored with the most significant data first (Big Endian)
- LSB...DEVICE STAMP is stored with the least significant data first(Little Endian)

Serializer Generator Options (prior to kernel V1.40 only)



The GENERATOR options control the initial values of the S/N and DATE as well as the size of each. The S/N INCREMENT is also set in this section.

Initial S/N
[default=0]

This sets the NEXT S/N that will be generated during the DEVICE STAMPING process. Enter a decimal number from 0 to 4294967295 $((2^{32})-1)$.

S/N Increment
[default= 1]

This is the value by which the S/N is incremented after a successful DEVICE STAMP. An unsuccessful DEVICE STAMPing will not increment the S/N so that S/Ns are not wasted on failing devices. To change the INCREMENT, click on the up/dn arrows. Values can range from -32768 to 32767.

S/N Size
[default=per Format]

This is size of the S/N portion of the DEVICE STAMP. The possible values are dynamically changed to match the FORMAT option.

Format Min Size Max Size

Binary	8 bits	32 bits
BCD	1 digit	10 digits
ASCII	1 char	10 chars

If the formatted representation of the current S/N is larger than the selected size, The generated S/N will be truncated to the size specified. Example...the FORMAT is ASCII and the next S/N is 1234567890 but you have selected a size of only 5 chars. The S/N generator will produce ...67890.

Date
[default=6/21/04]

This is the value for the date portion of the DEVICE STAMP. To change it, click the arrow to open the calendar and navigate to the desired year/month, then click the day. Since there is no real-time clock in the programmer, the date must be manually set.

Date Format
[default=MMDDYY]

This option selects the date format. To change the selection, pull down the combo box by clicking on the arrow, then click an option.

You can select from the following:

- MMDDYY
- DDMMYY
- MMDDYYYY
- DDMMYYYY

Date Size
[default=per Format]

This is a read only display of the size of the DATE portion of the DEVICE STAMP. It is dynamically calculated from the STORAGE FORMAT and DATE FORMAT.

Format YY YYYY

Binary 24 bits 32 bits

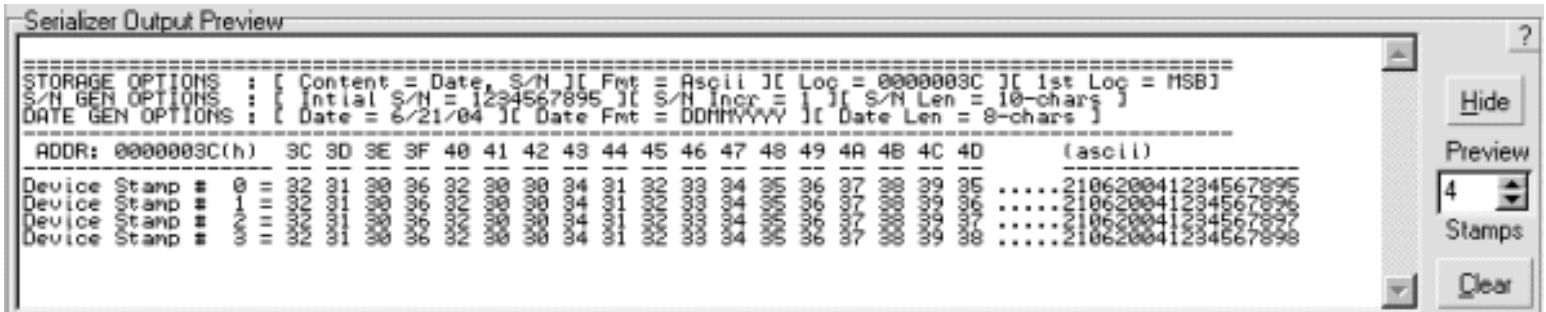
BCD 6 digits 8 digits

ASCII 6 chars 8 chars

Preview

If the PREVIEW panel is closed, clicking this button opens it. See below for details on PREVIEW.

Serializer Preview (prior to kernel V1.40 only)



This is a tool for setup of the SERIALIZER OPTIONS. A selectable number of DEVICE STAMPS are generated from the current SERIALIZER settings and displayed along with a header describing the current settings. Use PREVIEW to setup and adjust the SERIALIZER settings until the desired output is obtained. The Preview does not program or affect devices in any way, it is simply a display.

PREVIEW shows each DEVICE STAMP as both hex and ASCII data as well as the device address for each byte in the DEVICE STAMP.

Preview Stamps
[default=4]

Click the up/dn arrows to adjust the number of DEVICE STAMPS displayed.

Hide

Click this button to hide the PREVIEW panel.

Clear

Click this button to clear the PREVIEW window.

Preview

Once the PREVIEW panel is open, clicking this button displays the selected number of DEVICE STAMPS using the current SERIALIZER settings.

Log File

The Log File is a text file that records the DEVICE STAMP upon successful programming of the DEVICE STAMP into a device. One entry is recorded for each device stamped.

Enable

[default= off]

Check this to enable the Log File capture.

...

[default= default.log]

Click this button to change the Log file name or location. If the named file cannot be opened but logging is enabled, a file named default.log in the same folder as Winlink.exe will be used.

Device Counter

The DEVICE COUNTER tracks the total number of devices programmed as well as the number of passing and failing devices. The DEVICE COUNTER status is displayed on the Winlink PANEL in both NORMAL and MINIMAL views.

Enable

[default= off]

Check this to enable the Device Counter.
Uncheck to diable the counter.

Reset

[default= default.log]

Click this button to reset the counter to zero.

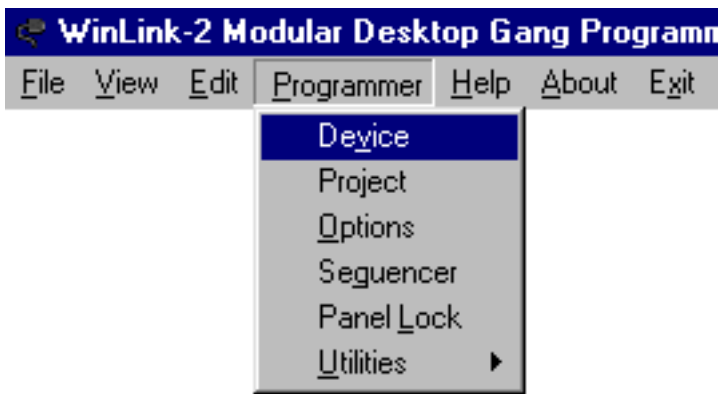
LOGICAL Help

Device Select

Accessing



You can access DEVICE SELECT by pressing the DEVICE button.



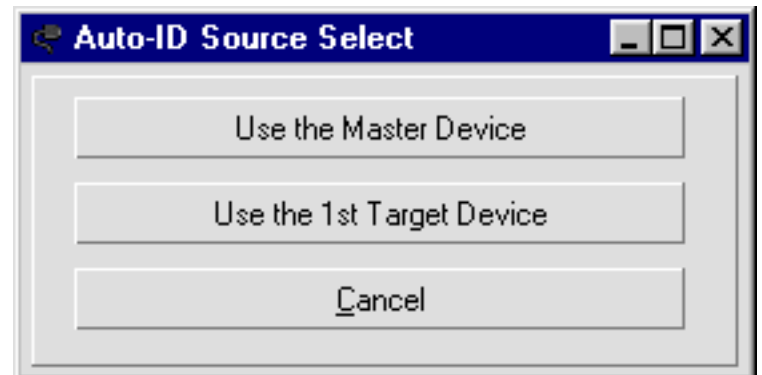
You can access DEVICE SELECT by clicking MAINMENU: PROGRAMMER: Device.

Selecting

If the installed MODULE supports the automatic selection of devices by reading the internal JEDEC manufacturer and device IDs, you can choose this method or choose to select the device from the resident DEVICE LIBRARY. If the MODULE does not support automatic selection, this window is skipped.



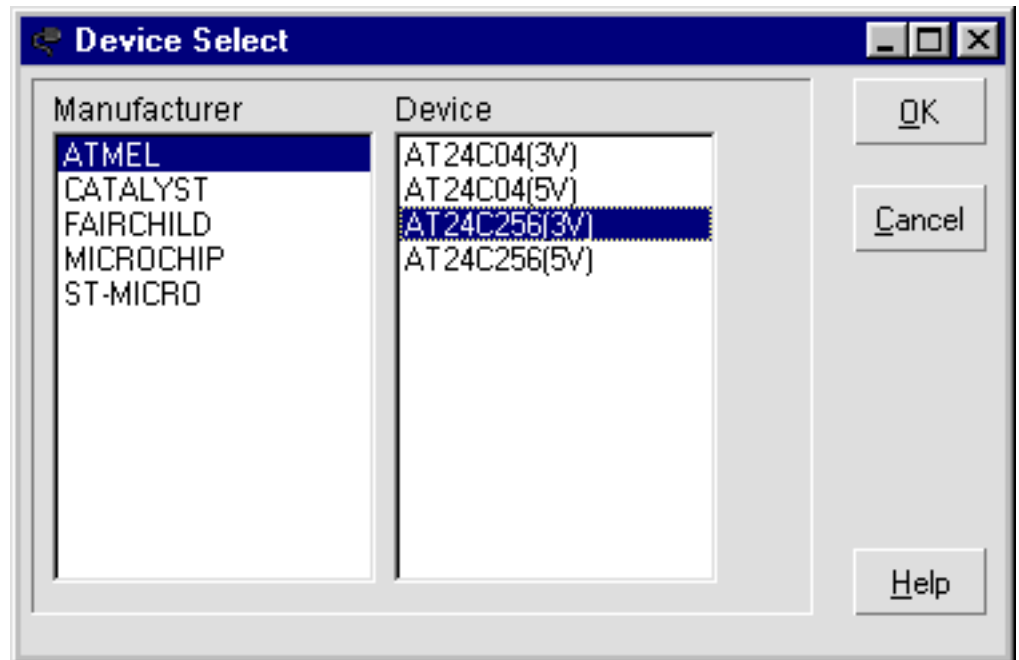
If you chose Auto-Select and the PROGRAMMER detects both a MASTER and one or more TARGET devices, this window appears and you can choose which device to use for Auto-select.



If you chose Auto-Select and the PROGRAMMER finds multiple matches in the resident database, this window appears so you can choose from the matching devices.



If not available or you chose not to use Auto-Select, this window appears so you can choose from the resident library manufacturer and device lists.

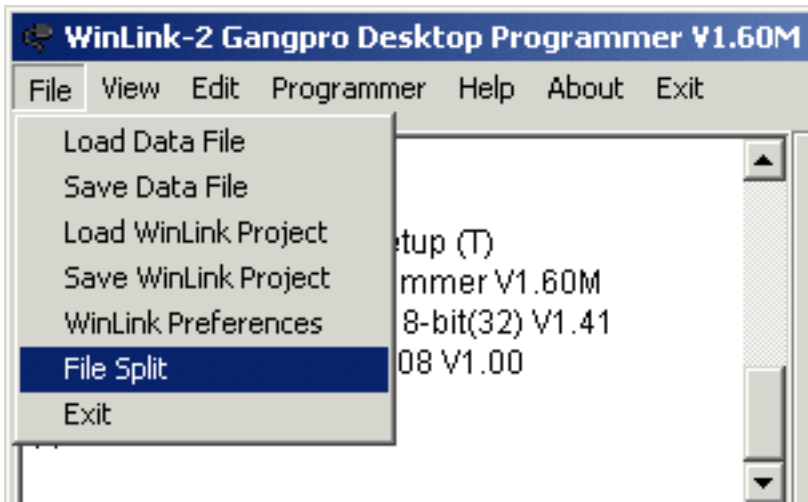


LOGICAL Help

File Split to Multiple Devices

The FILE SPLIT feature allows the EDITOR data to be interpreted as 16 or 32 bit data and split into multiple 8-bit memory devices. The file to be split is only loaded into the EDITOR once. The desired 'split' is selected 'on the fly' and programmed into all devices inserted in the platter.

Setting Up File Split Mode

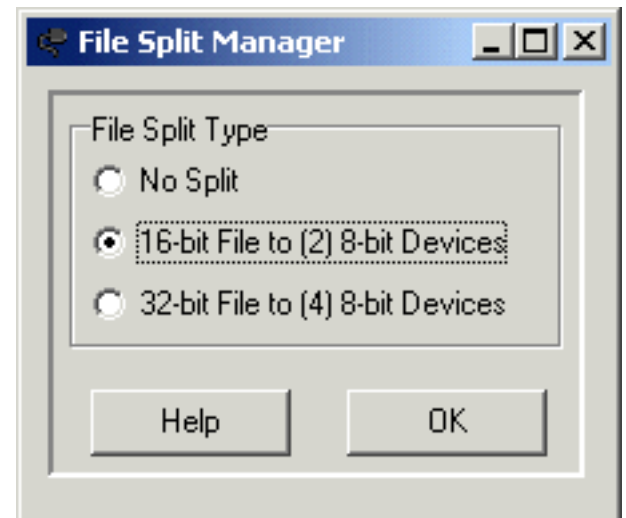


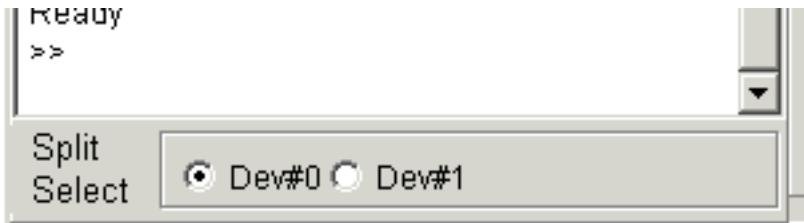
You can access the File Split Manager by clicking MAIN MENU | FILE | File Split .

This opens the File Split Manager as shown.

Click the desired File Split Option.

Click OK.





When the 16-bit or 32-bit File Split is selected, an additional panel is added below the Programmer Message Window.

The panel contains the radio buttons used to select the device from the

split.

For a 16-bit split (as shown) there will be two DEV buttons and for a 32-bit split there will be four DEV buttons.

Clicking a DEV button 'filters' or extracts the EDITOR data that is sent to the programmer.

This applies to all operations where data moves between the PROGRAMMER and the EDITOR.

(i.e. PROGRAM, VERIFY, and READ)

For a 16-bit split...

DEV#0 extracts only the data at EVEN EDITOR addresses.

DEV#1 extracts only the data at ODD EDITOR addresses.

For a 32-bit split...

DEV#0 extracts only the data where EDITOR address AND'ed with 0x03 == 0 .

DEV#1 extracts only the data where EDITOR address AND'ed with 0x03 == 1 .

DEV#2 extracts only the data where EDITOR address AND'ed with 0x03 == 2 .

DEV#3 extracts only the data where EDITOR address AND'ed with 0x03 == 3 .

16-bit example...

EDITOR 000000 = 0011223344556677

DEV#0 0000 = 00224466

DEV#1 0000 = 11335577

32-bit example...

EDITOR 000000 = 00112233445566778899AABBCCDDEEFF

DEV#0 0000 = 004488CC

DEV#1 0000 = 115599DD

DEV#2 0000 = 2266AAEE

DEV#3 0000 = 3377BBFF

First Time Step-by-Step Example

- 1) **Select Device** [...howto](#)
- 2) **Setup File Split**^{1,2} [...howto](#)
- 3) **Load Data File**³ [...howto](#)
- 4) **Select/Check the desired Split** [...howto](#)
- 5) **Insert blank devices**
- 6) **Program Devices with Current Split Data** [...howto](#)
- 7) **Remove Devices**
- 8a) **Program More Devices with Same Split...repeat steps 5-7**
- 8b) **Program Devices with Next Split...repeat steps 4-7**

Notes :

¹ You can setup or disable File Split at any time but doing it prior to loading data file makes address setup easier.

² Each time you exit the File Split Manager, DEV#0 is automatically selected .

³ Make sure the Up To File Address value is large enough for the file. Clicking the RESET button automatically sets this value properly.

File Split Effects in Other Areas

Editor Checksum

When File Split is active, the EDITOR checksum calculation reflects only the currently selected split. The Device CHKSUM button still calculates the checksum of the data in the device.

Note : If you add the EDITOR checksums generated by selecting each split, this sum should equal the original FILE's checksum.

Data File Load

When File Split is active, the calculation performed by the RESET button is modified so that the FILE FILTER end address accomodates the increased file size required by multiple devices.

When File Split is active, the chksums for each split as well as the FILE chksum is displayed in the message window after the file loads.

Editor End Address

The EDITOR size is increased to accomodate larger data size required by multiple devices when File Split is active.

Front Panel

The Split Select Panel appears below the programmer message window when File Split is active.

Byte Ordering

All File Types are by nature 8-bit and have no intrinsic byte ordering for words larger than 8-bits. However the compiler/assembler/linker that generated the file implicitly assumes a byte ordering scheme. The target application hardware also assumes an ordering scheme. There is no way to determine (from the file) which ordering scheme is assumed in the FILE. It is up to YOU to make sure the proper data gets into the proper device.

Notice that by default the byte ordering is LIL-ENDIAN (the LS byte is located at the lower EDITOR address.

LIL-ENDIAN

16-bit

EDITOR 0000 = bits[7..0] -> DEV#0

EDITOR 0001 = bits[15..8] -> DEV#1

32-bit

EDITOR 0000 = bits[7..0] -> DEV#0

EDITOR 0001 = bits[15..8] -> DEV#1

EDITOR 0002 = bits[23..16] -> DEV#2

EDITOR 0003 = bits[31..24] -> DEV#3

This is typical for Intel based architectures.

BIG-ENDIAN

16-bit

EDITOR 0000 = bits[15..8] -> DEV#0

EDITOR 0001 = bits[7..0] -> DEV#1

32-bit

EDITOR 0000 = bits[31..24] -> DEV#0

EDITOR 0001 = bits[23..16] -> DEV#1

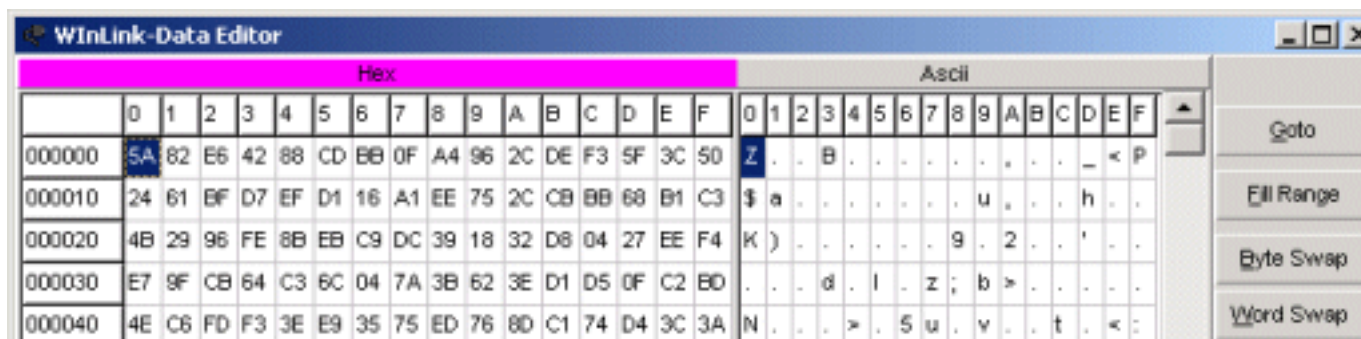
EDITOR 0002 = bits[15..8] -> DEV#2

EDITOR 0003 = bits[7..0] -> DEV#3

This is typical for Motorola based architectures.

Should it be necessary to interpret the EDITOR data as BIG-ENDIAN (the MS byte is located at the lower EDITOR address, use the BYTE SWAP and WORD SWAP(if 32-bit) buttons in the EDITOR to massage the byte order after loading the file. (*You could also simply physically re-arrange the devices in the target app*)

CONVERT BYTE-ORDER in the EDITOR



Use **BYTE SWAP** to convert 16-bit LIL-ENDIAN <-> BIG-ENDIAN

example...

EDITOR 000000 = 00112233 after **BYTE SWAP** EDITOR 000000 = 11003322

Use **BYTE SWAP** and **WORD SWAP** to convert 32-bit LIL-ENDIAN <-> BIG-ENDIAN

example...

EDITOR 000000 = 00112233 after **BYTE SWAP** EDITOR 000000 = 11003322

EDITOR 000000 = 11003322 after **WORD SWAP** EDITOR 000000 = 33221100

Tips & Tricks

Programming devices from a file (via the Winlink2 EDITOR) is considerably slower than copying devices from a MASTER. If your programming task requires large quantities of split file devices, it is recommended that you produce a single set of MASTER devices (one for each split) and then use the COPY operation to produce the desired quantity of each split (from the appropriate MASTER).

LOGICAL Help

Combo Mode

General

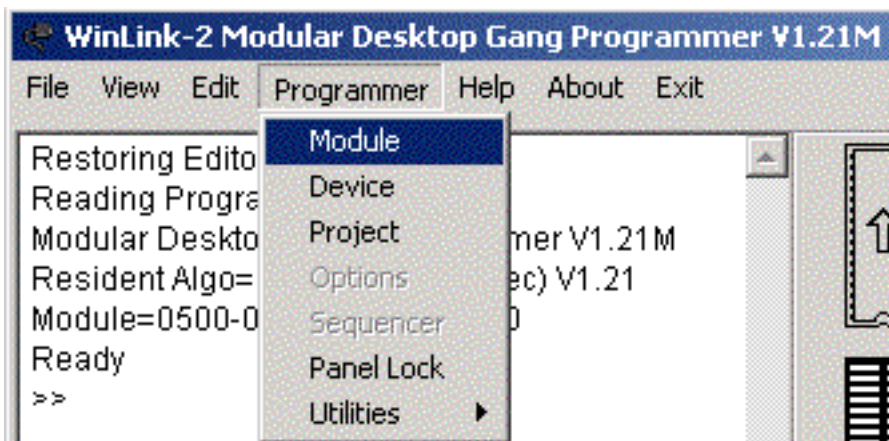
If your GangPro-8XP is configured with more than one device support module, it automatically operates in a special Combo (or Mixed) Mode. Several additional features are available in this mode.

Even though multiple modules are installed, only one will operate at a time. You must select which of the available modules will operate.

All options and settings of any module will be remembered if you select another one. The options and settings of the original module will be restored if you re-select it.

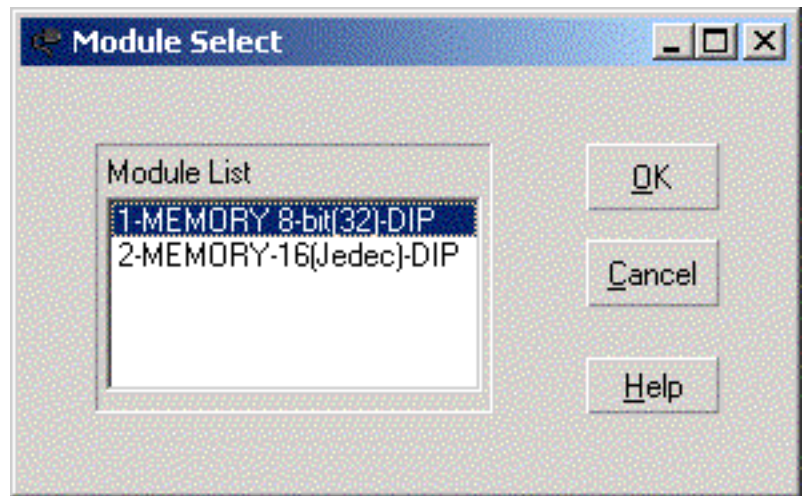
The device support program code is stored on the module itself. When you switch to a module that requires different device support code, the code must be loaded into the microcontroller that operates the programmer. This can take up to a minute depending on the size of the device support code.

Selecting Module from Winlink



You can access MODULE SELECT by clicking MAINMENU: PROGRAMMER: Module.

The available modules are listed in the window. Select the desired module and click OK.



If the module selected requires a code change, the programmer will temporarily disconnect and you will see the message shown at the right.

The LCD on the programmer will show the progress of the code change. When the code change is completed, the programmer will automatically reconnect using the newly selected module.

If the selected module does not require a code change, the switch will be instantaneous.



Selecting Module from Programmer

You can access MODULE SELECT on the programmer by pressing [UTIL].

Then press [PREV] or [NEXT] until the LCD shows [4-Module Select].

Press [ENTER].

The LCD shows the first available module.

[Select Module 0]
[Name of Module]

The LEDs on each module will light as you switch between them.

Press [PREV] or [NEXT] until the LCD shows the desired module or the LEDs are lighted.

Press [ENTER].

If the module selected requires a code change, the LCD on the programmer will show the progress of the code change.

If the selected module does not require a code change, the switch will be instantaneous.

The LCD will show the selected device (if any) of the newly selected module.

LOGICAL Help

Winlink2 Command Line Mode (WCL)

The Command Line feature allows Winlink2/Gangpro-8XP to be controlled via command line arguments. Winlink Command Line (WCL) is part of Winlink2 and is automatically enabled when Winlink2 is executed with command line arguments.

This feature is useful for embedding the programmer into another program or batch file. Arguments can be supplied directly on the command line or inserted into a text file.

Winlink2 writes to two status files that provide ready/busy indication and the results of programmer operations. A host program can utilize these files to interface with WCL.

Example batch files and a simple utility to exercise the command line mode are installed into the Winlink2\cmdline folder.

The utility (wlcUtil.exe) is useful for examining the status of programmer operations in command line mode.

In the remainder of this document WCL refers to calling Winlink2.exe with command line arguments.

Hardware Requirements

The command line mode was added to Winlink2 beginning with V1.30. You can check Winlink2's version by running Winlink2 and clicking 'About'. If you are reading this from within Winlink, you already have V1.30+. If you need to upgrade Winlink2, install Winlink2 from the CD-ROM or download from the [Logical Website](#).

The command line mode was added to the Gangpro-8XP beginning with kernel V1.30M. You can check your '8XP's version by running Winlink2 and connecting the '8XP. On startup you will see...

```
Modular Desktop Gang Programmer Vx.xxM  
Resident Algo= 'name of algo' Vx.xx
```

If the Modular Desktop Gang Programmer Vx.xxM is less than V1.30, you will need to update the kernel base.

Click [here](#) for Base Update instructions.

You also should check the Resident Algo version. It should be > V1.20 although earlier versions will work.

Click [here](#) for Module Update instructions.

WCL Environment

By default, Winlink2 is installed into c:\Program Files\Logical Devices\Winlink2. You may want add the path to the Winlink2 installation for convenience but this is not necessary and Winlink2 can locate it's internal files without adding the path.

Note that if you do choose to add the Winlink2 path to autoexec.bat, it must be in the DOS filename format like this.... `PATH= %PATH%;c:\Progra~ 1\Logica~ 1\Winlink2`

For simplicity, Winlink2 Command Line (WCL) uses 2 files to interface with the outside world.

STATUS.WCL contains the ready/busy status of WCL. The name of this file cannot be changed.

RESULT.WCL contains the result of programmer operations. The name of this result file can be changed with a WCL argument so that a host program can log and store results of operations in different files.

By default, both these files will reside in the CURRENT DIRECTORY in effect at the time WCL is called. The path of both these files can be changed from the default with a WCL argument.

WCL Input Syntax

The syntax is basically....

```
<path> Winlink2<.exe> <@ filename> <cmd1> <arg to cmd1> <cmd2> <arg to cmd2>  
<cmdn> <arg to cmdn>
```

There must be at least one command for Winlink2 to enter Command Line Mode.

Some commands must have an argument.

Command arguments (paths and filenames) must be surrounded by quotes(" ") if the argument contains spaces.

```
Winlink2 -f "c:\My Documents\test files\test.hex" load program verify
```

Windows long pathnames or filenames can be specified with DOS equivalents. (i.e. Program Files -> Progra~1).

Commands and command arguments MUST be separated by spaces.

Commands are NOT case sensitive.

Each command has a 1 or 2 char abbreviation as an alternate.

Command Files

Commands may be placed into a standard text file created with Notepad, Wordpad or your favorite text editor. Enter only one command or command argument per line.

Comments must be on separate lines with '#' as the 1st character of the comment line.

Example cmds.txt:

```
#TEST COMMAND FILE (this is a comment )  
OUTPATH  
c:\my_out_path\  
FILE  
c:\files\test.hex  
LOAD  
BLANK  
PROGRAM  
VERIFY
```

In this case you would call WCL like this... <path>Winlink2 @ <path>cmds.txt

WCL would

- 1) change the output file path to c:\my_out_path\
2) set the Data Filename to c:\files\test.hex
3) Load c:\files\test.hex into the editor.
4) Blank Check the target device(s)
5) Program the target device(s) with the editor data
6) Verify the target device(s) against the editor data.
7) generate c:\my_out_path\result.wcl containing the final result of all the programmer commands.

If cmds.txt is located in a different folder, then you must supply an absolute path or a path relative to the current directory .

Note that by default if you supply the path to the WCL executable when invoking it or put the executable path in autoexec.bat, then the STATUS.WCL,RESULT.WCL, and cmds.txt can all be in one working folder (the current folder when WCL is executed)

What WCL Does Not Control

Currently, WCL does not allow the following....

Module Selection where multiple modules are installed (WCL uses the current module).

Device Selection (this must be performed manually. WCL uses the current selection).

Device Options (these must be setup manually. WCL uses the current options).

Program Options (WCL ignores program options. COPY/PROGRAM will not pre-erase,pre-blank or post-compare/verify).

Winlink Preferences (WCL uses the current Preferences).

Command Reference

(Equivalent shortcuts are enclosed in parentheses.)

Programmer Commands

- ERASE (E)
Erase the TARGET device(s) in the programmer.
- BLANK (B)

Blank Check the TARGET device(s) in the programmer.

- PROGRAM (P)
Program the TARGET device(s) in the programmer with Editor data.
- VERIFY (V)
Verify the TARGET device(s) in the programmer against Editor data.
- COPY (C)
Program the MASTER device data into the TARGET device(s).
- COMPARE (M)
Compare the the TARGET device(s) against the MASTER device.
- CHKSUM (K)
Checksum the MASTER or 1st TARGET device.
- SECURE (U)
Secure the TARGET device(s) in the programmer.
- READ (R)
READ the MASTER or 1st TARGET device into the Editor.

Source Device Commands

- MAST
If both a MASTER and TARGET device(s) are inserted, use the MASTER device in subsequent CHKSUM or READ commands.
If only one or the other (MASTER or TARGETs) is inserted then that device is used.
- TARG (default)
If both a MASTER and TARGET device(s) are inserted, use the 1st TARGET device in subsequent CHKSUM or READ commands.
If only one or the other (MASTER or TARGETs) is inserted then that device is used.

Data File Commands

- FILE (F) [Path\Filename]
Set the Filename for subsequent FILE LOAD or FILE SAVE commands.
Filename must be set before LOAD or SAVE.
Path must be an absolute path or relative to the Winlink2 executable path.
If the Path\Filename contains spaces, then enclose it in quotes(" ");
- SAVE (S)
Save the Editor to Filename.
- LOAD (L)
Load the Editor from Filename

Data File Format Commands

- INTEL (default)
Set the SAVE file format to Intel Hex.
- MOTO
Set the SAVE file format to Motorola Hex.
- TEK

Set the SAVE file format to Tektronix Hex.

- BIN
Set the SAVE file format to Binary.
Note that for LOAD, the file format is autodetected.

Path/File Related Commands

- OUTPATH (OP) [Pathname]
Set the path for RESULT.WCL and STATUS.WCL.
If not set, the PATH in effect when WCL is executed will be used to locate STATUS.WCL.
You must include the trailing backslash '\'.
If the Pathname contains spaces, then enclose it in quotes(" ");
- RESULT (RF) [Filename] or [Pathname\Filename]
Set a new filename for the result file. If the new filename includes a path ('\' char), it overrides the OUTPATH setting for the result file. If the new filename does not include a path , the new filename is appended to OUTPATH to determine the result file location.
If the Pathname contains spaces, then enclose it in quotes(" ");
RESULT only applies to result.wcl and does NOT affect the location of status.wcl.

example:

```
OUTPATH
c:\my_outpath\
RESULT
c:\my_result\new_result.wcl
```

here the status file would be c:\my_outpath\status.wcl and the result file
c:\my_result\new_result.wcl

example:

```
OUTPATH
c:\my_outpath\
RESULT
new_result.wcl
```

here the status file would be c:\my_outpath\status.wcl and the result file
c:\my_outpath\new_result.wcl

Misc Commands

- DEBUG (D)
Do not close WCL after all commands have executed. Remain open and enter normal manual operation mode. WCL will remain open so you can see what occurred. The commands that executed are shown in the Info window on the Winlink2 front panel.

WCL uses STATUS.WCL to indicate it's busy status.

When Winlink2 enters Command Line Mode, it writes the string 'BUSY' to STATUS.WCL. When WCL exits, it it writes the string 'READY' to STATUS.WCL.

A host program can call WCL and then read STATUS.WCL to determine when WCL has finished executing commands.

WCL uses RESULT.WCL to indicate the result of programmer operations. RESULT.WCL is a standard text file. It always contains 10 lines.

Here are some examples of RESULT.WCL files for various programmer commands.

CMD	BLANK	COPY	COMPARE	PROGRAM	VERIFY	CHKSUM	READ	ERASE
MAST	-	M	M	-	-	-	M	-
TARG#1	F	-	-	-	-	-	-	X
TARG#2	P	P	P	P	F	0001FEE2	-	X
TARG#3	P	P	-	P	P	-	-	X
TARG#4	P	-	-	P	-	-	-	X
TARG#5	F	F	F	P	P	-	-	X
TARG#6	-	-	-	-	-	-	-	X
TARG#7	F	-	F	-	-	-	-	X
TARG#8	F	-	-	F	-	-	-	X

Note that unknown commands, file format commands or path commands will not generate a result file. Only programmer commands generate the result file.

Result File Format

Line 1 always denotes the command that generated the result.

Possible values are...

- ERASE
- BLANK
- PROGRAM
- VERIFY
- COPY
- COMPARE
- SECURE
- CHKSUM
- READ

LOAD
SAVE

Line 2 denotes the MASTER device.

Possible values are...

'-' The MASTER device did not participate in the command.

'M' The MASTER device was used as a data source in the command.

'X' THE MASTER device was required by the command but not present or inserted incorrectly.

'NNNNNNNN' is an 8-digit hex number representing the CHKSUM of the MASTER device.

Lines 3 to 10 denote the TARGET device(s). Each line represents one target socket starting with target #1.

Possible values are...

'-' The TARGET device did not participate in the command.

'P' The TARGET device passed the command requirements.

'F' The TARGET device failed the command requirements.

'X' The TARGET device was required by the command but not present or inserted incorrectly.

'NNNNNNNN' is an 8-digit hex number representing the CHKSUM of the TARGET device.

Chained Results

If WCL is executed with more than one command that generates a result, only the final command result is contained in RESULT.WCF. The result of each individual command is chained to the next and a device must pass ALL commands in the chain to be PASSED in the final result. If a device fails ANY command in the chain, it is FAILED in the final result.

Logging Results

If you desire to see the individual command results, then execute WCL with only one command at a time. You can change the name of each result.wcf file by using the RESULT command. This allows a host program to log the results of each operation.

Embedding WCL in other programs

How to call WCL

The Windows API provides several methods for spawning (calling) other programs from inside an application. These include ShellExecute, ShellExecuteEx and CreateProcess. Each varies in complexity and functionality. The simplest ShellExecute is discussed here.

Here is the format of the call (for more details see Help for your compiler or the Windows SDK).

```
HINSTANCE ShellExecute(  
    HWND hwnd, // handle to parent window  
    LPCTSTR lpOperation, // pointer to string that specifies operation to perform  
    LPCTSTR lpFile, // pointer to filename or folder name string  
    LPCTSTR lpParameters, // pointer to string that specifies executable-file parameters  
    LPCTSTR lpDirectory, // pointer to string that specifies default directory  
    INT nShowCmd // whether file is shown when opened  
);
```

Your compiler probably encapsulates ShellExecute() in some form.

Here is the Delphi version from wclUtil.

```
ShellAPI.ShellExecute(0, 'open', pchar(exepath+ 'winlink2.exe'), pchar(edit1.text), nil,  
SW_SHOWNORMAL);
```

here...

exepath is just a string with the path to the Winlink2 installation

edit1.text is a string containing the commands.

pchar() converts Delphi to 'C' strings.

Here is a MS link for Visual Basic...

http://msdn.microsoft.com/archive/default.asp?url=/archive/en-us/dnarvb4/html/msdn_shelexec.asp

Here is a Delphi link....

<http://delphi.about.com/od/windowshellapi/l/aa082499.htm>

Here is a VC++ link....

<http://www.codeproject.com/system/newbiespawn.asp>

How to detect when WCL has terminated

You must consider the fact that your Windows program will not wait for WCL to terminate after it is called with ShellExecute. Reliably determining when a spawned program terminates can

get complicated and is beyond the scope of this document. This is the primary reason I included the STATUS.WCL file method to determine when WCL has terminated . It is certainly inelegant but relatively simple.

If you are proficient with the Windows API and want to detect WCL termination via the Windows API, I suggest you see these links for more info

<http://www.festra.com/eng/mtut01.htm>.

<http://www.codeproject.com/system/newbiespawn.asp>.

You can also Google 'shellexecute termination'.

Using STATUS.WCL

Let us examine what you need to do to use STATUS.WCL to detect if WCL is running.

Here is pseudocode...

```
ShellExecute (...); //Start WCL with some command(s)
while (!ready) //wait for WCL to terminate
{
    wait (10-50 ms); //give WCL a chance to open STATUS.WCL
    opened = fopen(STATUS.WCL); //may fail if WCL is writing to STATUS.WCL,
    if (opened) //handle error gracefully
    {
        status_str = fread(STATUS.WCL); //read the 1st line of STATUS.WCL
        fclose(STATUS.WCL); //close STATUS.WCL
        ready = strcmp(status_str,"READY"); //ready is true if 1st line = "READY"
    }
}
continue:
    now open and read RESULT.WCL
```

Essentially, you spawn WCL, then wait in a loop until STATUS.WCL's first line is 'READY'. You may want to limit the loop iterations in case something goes wrong. If your code has STATUS.WCL open when WCL attempts to write to it, WCL will retry 10K times until giving up with an error. Never write to STATUS.WCL

Reading RESULT.WCL

Once you have determined that WCL has terminated, you can then open and parse RESULT.WCL. Do not try to open RESULT.WCL until WCL has exited (STATUS.WCL contains 'READY'). Never write to RESULT.WCL

Minimizing the WCL panel

When embedding WCL, it is not possible to completely eliminate the panel display. You can however, minimize it down to a narrow bar. Winlink will also load faster using this view.

Run Winlink in normal mode, click 'View' on the MAIN MENU and click 'Minimal'.

Using WCLUtil

WclUtil is a simple utility that may help you to embed WCL in your program. It is in the cmdline folder under the Winlink2 Installation folder.

It essentially implements the STATUS.WCL loop described above to provide an easy method to exercise WCL with different arguments and examine the results.

Just click START to start the loop running.

wclUtil expects STATUS.WCL and RESULT.WCL to reside in the folder wclUtil runs from and Winlink2 to reside in the parent folder.

wclUtil waits for a BUSY to READY STATUS.WCL transition and then displays the contents of RESULT.WCL. You can type WCL commands into the edit control at the bottom and click RUN to execute WCL. You can also execute the example .bat files

It also has a reset button in case you hang up WCL somehow with STATUS.WCL still open and need to close it without a reboot.

Several batch files are also provided as examples. You can execute these while wclUtil is running and look at the results.

Be careful when running the batch files with inserted OTP devices. Some of the batch files load a test file and then program the target devices. If you cannot erase your devices, you may render them unusable by programming them with test data.