

68HC05

Development System

**Integrated Text Editor Environment, In Circuit Simulator,
Emulator, Cross Assembler, Disassembler and Programmer Board**

User's Manual

68HC805C4
68HC705C4A
68HC705C8
68HC705C8A
68HC705C9
68HC705C9A
68HC705D9
68HC705D9A
68HC705J1A
68HC705J2
68HC705K1
68HC705P6
68HC705P6A
68HC705P9
68HC705P9A
68HC705SR3
68HC705SU3
*68HC705P3
*68705U3
*68705R3
*68HC705B5
*68HC705B16
*68HC705E1

04/1997



A.C.N. 061 527 571

Oztechnics Pty Ltd
PO Box 38, Illawong, NSW 2234 Australia
Phone: (+61-2) 9541 0310 Fax: (+61-2) 9541 0734

Email: 68hc05@oztechnics.com.au
WWW: <http://www.oztechnics.com.au/>

Registered Software License Agreement

Oztechnics Pty Ltd provides this software package for use on the following terms and conditions,

You may:

- Use this software package on a single computer system. You may transfer this package from one computer system to another provided you only use it on one computer system at a time.
- Make copies of diskettes supplied with the software package for backup purposes provided all copies are labeled with the name of the software and Oztechnics copyright notice.
- Use the software package to create your own software programs.
- Transfer the software package and licence to a third party provided that the third party agrees to the terms and conditions of this licence and that all copies of the software package are transferred to the third party or destroyed. The third party must advise Oztechnics that they have accepted the software and terms and conditions of the licence.

You may NOT:

- Sell, lend, give away or in any way transfer copies of this software package to any other person or entity, nor allow any other person to make copies of the software package.
- Incorporate any part of this software package in your own programs.

Un-Registered Software License Agreement

The software package may be freely distributed for product evaluation purposes provided that the end user licence register file (oztech.lic) is not included and the package is distributed in its complete form without modification.

The software package assumes a software evaluation mode without the registered licence file present. Some features are disabled in this evaluation mode.

Table of Contents

1. Introduction	1
1.1. Motorola Micro Controllers	1
1.2. 68HC05 Development System Requirements	3
1.3. 68HC05 Development System Software Installation	3
1.4. Adobe Acrobat Requirements (optional)	3
1.5. Adobe Acrobat Installation (optional)	3
2. 68ED: Integrated Text Editor Environment (ITEE)	4
2.1. The ITEE Windows User Interface	5
2.1.1. Pull Down Menus	5
2.1.2. Keyboard Menu Selection	5
2.1.3. Mouse Menu Selection	5
2.1.4. Menu Hot Keys	6
2.1.6. Windows	6
2.1.7. Status Bar	7
2.2. The Editor	8
2.2.1. Auto Indent	8
2.2.2. Keyboard Commands	8
2.2.3. Book Mark	9
2.2.4. Insert Mode	9
2.2.5. Block Commands	10
2.3. = System Menu	11
2.3.1. ASCII Chart Display	11
2.3.2. Calculator	12
2.3.3. Simulator	13
2.3.4. Programmer	13
2.3.5. Disassembler	13
2.3.6. Diagnostics	13
2.4. File Menu	14
2.4.1. Open	14
2.4.2. New	15
2.4.3. Save	15
2.4.4. Save As	15
2.4.5. DOS Shell	15
2.4.6. Print File	15
2.4.7. Repaginate	15
2.4.8. Statistics	15
2.4.9. Close File	16
2.4.10. Exit	16
2.5. Edit	17
2.5.1. Undo	17
2.5.2. Cut	17
2.5.3. Copy	17
2.5.4. Paste	17
2.5.5. Mark/Unmark	18
2.6. Search	19
2.6.1. Find	19
2.6.2. Replace	19
2.6.3. Search Again	20
2.6.4. Go to Line Number	20
2.7. Assemble	21
2.7.1. Assemble	21
2.7.2. View .lst file	23
2.8. Window	24
2.8.1. Close File	24

2.8.2. Editor	24
2.8.3. Error Log	24
2.8.4. File History	24
2.8.5. View History	25
2.8.6. Zoom Editor	25
2.8.7. List all..	25
2.9. Options	26
2.9.1. Screen Setup	26
2.9.2. Micro Controller Type	27
2.10. Help	28
2.10.1. Contents	29
3.0 Software Simulation	30
3.1. In Circuit Simulation	31
3.1.1. ICS Operation	32
3.2. In Circuit Emulation	33
3.3. 68SIM05: In Circuit Software Simulator	34
3.3.1. Simulator Screen Layout	35
3.3.2. Memory Display Window	36
3.3.3. I/O Ports Area	37
3.3.4. CCR Area	37
3.3.5. Registers	37
3.3.6. Status Line	37
3.3.7. Command Line	38
3.3.8. Trace Area	38
3.3.9. Micro Controller Type	38
3.4. Display Modes	39
3.5. Log Files	39
3.6. 68SIM05 Commands	40
 Source Code Display	40
 Set Breakpoint	40
 Command Summary	40
 Toggle External INT2 Pin	41
 Toggle External Timer Input Pin	41
 Toggle TCAP Input Pin	41
 Display Data Format	41
 Go	42
 Step Over Subroutine	42
 Halt Simulation	42
 Toggle External Pin	42
 Reset Cycle Counting	42
 Reset Interrupt Cycle Counting	43
 Load a .S19 file	43
 Modify Memory	43
 SCI - Set RDRF (C8 Controller)	43
 Disable all Breakpoints	43
 Toggle TCAP2 Input Pin	44
 Quit Simulator	44
 Modify Registers	44
 Step One Instruction	44
 Step Backwards One Instruction	44
 Toggle Display Mode	45
 Log File	45
 View Registers and Vectors	45
 Configure Communications Parameters	45
 Reset Micro Controller	45
 Simulation Speed	45
 In Circuit Simulation (ICS)	46
 Auto Load PortA	46
 Auto Load PortB	46
 Auto Load PortC	46
 Auto Load PortD	46

5	K1 Controller External Interrupt: PA0	47
6	K1 Controller External Interrupt: PA1	47
7	K1 Controller External Interrupt: PA2	47
8	K1 Controller External Interrupt: PA3	47
9	Auto Load A-D Converter	47
0	Auto Load SCI Data Register	48
-	Auto Load SPI Data Register	48
Ctrl A	Set Accumulator Break Value	48
Ctrl X	Set X Register Break Value	48
Alt A	Modify Accumulator	48
Alt C	Modify CCR Register	49
Alt G	Go to Line Number in Source Code	49
Alt P	Modify Program Counter	49
Alt X	Modify X Register	49
Alt 0	Toggle Port A Bit 0 - Keyboard Interrupt (SR3 controller only)	49
Alt 1	Toggle Port A Bit 1 - Keyboard Interrupt (SR3 controller only)	49
Alt 2	Toggle Port A Bit 2 - Keyboard Interrupt (SR3 controller only)	50
Alt 3	Toggle Port A Bit 3 - Keyboard Interrupt (SR3 controller only)	50
Alt 4	Toggle Port A Bit 4 - Keyboard Interrupt (SR3 controller only)	50
Alt 5	Toggle Port A Bit 5 - Keyboard Interrupt (SR3 controller only)	50
Alt 6	Toggle Port A Bit 6 - Keyboard Interrupt (SR3 controller only)	50
Alt 7	Toggle Port A Bit 7 - Keyboard Interrupt (SR3 controller only)	51
F1	Command Summary	51
F2	Randomise Controller	51
F9	Jump 50 lines forward	51
F10	Jump 50 lines backward	51
3.7. Entering Values into the Simulator		52
4. 68PROG05: EPROM Programmer		53
4.1. Option 1 - Configure Programmer		54
4.1.1.	Option 1 - Select LPT1 or LPT2	55
4.1.2.	Option 2 - Select COM1 - COM4	55
4.1.3	Option 3 - Select programmer board hardware version.	55
4.1.4.	Option X - Exit to main menu	55
4.2. Option 2 - Display Hex Dump of PROM Image		55
4.3. Option 3 - Program and Verify PROM		56
4.4. Option 4 - Verify PROM Contents vs .S19 File		56
4.5. Option 5 - Verify Blank PROM		56
4.6. Option 6 - C8 Controller Options		57
4.6.1.	Option 1 - Secure PROM and Verify	57
4.6.2.	Option 2 - Secure PROM and DUMP (SCI PORT)	57
4.6.3.	Option 3 - Load Program into RAM / Execute / Monitor	57
4.6.4.	Option 4 - Execute Program in PROM (Monitor Mode)	58
4.6.5.	Option 5 - Dump PROM contents (SCI port)	58
4.6.6.	Option X - Exit to Main Menu	58
4.7. Option X - Exit		58
4.8. Monitor Mode / Real Time ICE		59
4.9. Error Messages		60
5.0 The Programmer Board		61
6. Programmer Board Diagnostics		63
7. 68DIS05: Disassembler		64
8. Tutorial		65

Appendix I - 68SIM05 Quick Reference Guide	67
68HC705C8,C8A,C9 Specific Commands	68
68HC705D9 Specific Commands	68
68HC705B5,B16 Specific Commands	68
68HC705K1 Specific Commands	68
68HC705J1A, J2 Specific Commands	69
68HC705SR3 Specific Commands	69
68705R3 Specific Commands	69
68705U3 Specific Commands	69
68705P3 Specific Commands	70
68HC705P6, P9 Specific Commands	70
68HC705E1 Specific Commands	70
Appendix II - Supported Motorola Micro Controllers	71
Appendix III - Schematic Diagrams V3.6 Programmer Board	73
Sheet 1	74
Sheet 2	75
Sheet 3	76
Sheet 4	77
Sheet 5	78
Sheet 6	79
Sheet 7	80
Sheet 8	81
Sheet 9	82
Sheet 10	83
Appendix IV - ICS Adaptor Boards Schematics	84
Appendix V - Programmer Adaptor Boards Schematics	88

List of Figures

Figure 1 - ITEE Screen	4
Figure 2 - System Menu	11
Figure 3 - ASCII look up table	11
Figure 4 - Programmer's Calculator	12
Figure 5 - File Menu	14
Figure 6 - File Open Selection Box	14
Figure 7 - File Statistics Window	15
Figure 8 - Edit Command	17
Figure 9 - Search Menu	19
Figure 10 - Search for text window	19
Figure 11 - Search and Replace	19
Figure 12 - Move cursor to a line in the text editor	20
Figure 13 - Assemble Menu	21
Figure 14 - Assembler	22
Figure 15 - Auto line position	22
Figure 16 - Window Menu	24
Figure 17 - Options Menu	26
Figure 18 - Screen Setup	27
Figure 19 - Help Menu	28
Figure 20 - Help Main Index	29
Figure 21 - Simulator Screen (Color)	35
Figure 23 - 68PROG05 Main Menu configured for the 68HC705C8 micro controller	54
Figure 24 - Screen capture of Configure Programmer Options	54
Figure 25 - Screen capture of PROM Memory Contents Dump	56
Figure 26 - Screen capture of C8 Controller Options	57
Figure 27 - Screen capture of Monitor Mode	59

List of Tables

Table 1 - Menu System Actions	5
Table 2 - Menu Commands	6
Table 3 - Miscellaneous ITEE Commands	8
Table 4 - Cursor Movement Commands	9
Table 5 - Insert & Delete Commands	9
Table 6 - Block Commands	10
Table 7 - Calculator Operators	12
Table 8 - Command line arguments for supported Micro Controllers	34
Table 9 - Memory Map Colour Codes	36
Table 10 - EPROM programmer board supported controllers	53
Table 11 - Programming Voltages (Vpp)	61
Table 12 - 68SIM05 Quick Reference Guide	67
Table 13 - Supported Micro Controllers	71

Preamble

Disclaimer of liability: Oztechnics Pty Ltd makes no warranties, expressed or implied, as to this product, including but not limited to implied warranties of compatibility with a particular system, suitability for a particular purpose, or as to liability for any damages that may result from its use, such as lost profits or damaged equipment, among other possibilities.

Oztechnics Pty Ltd

A.C.N. 061 527 571

Oztechnics Pty Ltd
PO Box 38
Illawong NSW 2234
Australia

Phone	(+61-2) 9541 0310
Fax	(+61-2) 9541 0734
Email	68hc05@oztechnics.com.au
WWW	http://www.oztechnics.com.au/

This page left intentionally blank

1. Introduction

Developing software for single chip micro controllers requires special techniques. Since the program, memory, registers, I/O, address bus, data bus and control circuitry are completely internal, debugging and running a program is nearly impossible without the correct software and hardware tools.

The Oztechnics 68HC05 Development System provides all the necessary tools to develop applications for these single chip micro controllers in the form of a hardware programmer board, sophisticated Integrated Text Editor Environment, In Circuit Software Simulator, in line cross-assembler, disassembler, programmer board diagnostics utility, programming software, monitor and demonstration programs.

This manual provides information on both the hardware and software components of the Oztechnics 68HC05 Development System.

1.1. Motorola Micro Controllers

Motorola have developed a wide range of micro controllers to suit many different applications. The main features of the 68HC05 series of micro controllers supported by the Oztechnics 68HC05 Development System are listed in Appendix II. Based on these tables, a suitable micro controller to suit your 8 bit design requirements can be determined.

Motorola use a number of standard designations to distinguish between the various micro controllers.

The 68705 designation is an older HMOS device. It is strongly recommended that all new designs are based around the HC05 series as opposed to the older HMOS devices.

The 'HC' designations of micro controllers in the 68HC705 series signify that the micro controller is a high speed, low power CMOS version.

The '7' designation in the 68HC705 series indicates that the device contains Programmable Read Only Memory (PROM). These devices have to be programmed using the internal boot-strap mode of the micro controller. These devices can be erased using a suitable Ultra Violet light source if the 'S' package (windowed) is used.

The '8' designation in the 68HC805 series indicates that the micro controller contains Electrically Erasable Read Only Memory (EEPROM). The Oztechnics Development System can erase these devices from the programming socket of the programmer board prior to burning a program into the controller.

The 'P' designation at the end of a part number indicates that the controller is a One Time Programmable device (OTPROM). OTPROM devices are used after a program has been finalised, such as in a production run.

The 'S' designation at the end of the part number indicates that the controller contains a quartz window for EPROM erasing. To erase, the EPROM is exposed to an UV light source.

The '68HC05' designation indicates that the controller contains a factory masked Read Only Memory (ROM). These devices are manufactured with the program already burnt into the ROM. These devices are typically used for high volume low cost applications.

1.2. 68HC05 Development System Requirements

To use the Oztechnics 68HC05 Development System you will require a PC computer as follows:-

- AT or greater;
- EGA graphics adaptor or better;
- 500Kb free memory
- 1.44Mb floppy disc;
- 4Mb free hard disc space;
- Parallel printer port (LPT1 or LPT2);
- Serial port (COM1 or COM2).

In addition to the above PC hardware you will require:-

- 12 Vdc 150mA plug pack.
- parallel (1-1) and serial cables (DCE straight through type)

1.3. 68HC05 Development System Software Installation

To install the software package:-

1. Insert the supplied floppy disc into drive A: or B:
2. Change to the appropriate drive by typing `A:` or `B:`
3. Type `INSTALL` to install all of the files to the C:\68HC705 directory or alternatively type `INSTALL D:` to install all of the files to the D:\68HC705 directory or `INSTALL E:` to install all of the files to the E:\68HC705 directory.

1.4 Adobe Acrobat Requirements (optional)

- 386, 486, or Pentium-based personal computer
- Microsoft Windows 3.1 or greater
- 4 MB application RAM
- 3 MB hard disk space
- 3.5" floppy disk drive

1.5 Adobe Acrobat Installation (optional)

To install Acrobat Reader 2.1 on your hard drive:

1. From the Windows Program Manager, choose Run from the File menu and select the floppy drive
2. Follow the instructions on your screen. Installation requires approximately 3 MB of free hard disk space.

2. 68ED: Integrated Text Editor Environment (ITEE)

The Integrated Text Editor Environment (ITEE) forms the main shell for assembly language software development. From the ITEE, source code can be written and edited in the text editor, assembled and debugged using the in-line cross assembler and simulator and then finally programmed into a micro controller without ever having to revert to the DOS prompt. Many time saving features are incorporated into the ITEE to assist with software development.

To start the ITEE, simply type **68ed**  at the DOS prompt. An initial window will appear displaying the Copyright and software version details. Click the OK button with the mouse or press the  key to enter the editor environment. Once the editor environment is entered a screen will be displayed similar to the one as shown in *Figure 1* below.



Figure 1 - ITEE Screen

The initial screen is broken up into three areas which from the top, contain the menu bar, the text editor and the status bar. A separate message window is also displayed at the bottom of the screen for displaying file history and assembler out put messages when required.

Note that the text editor automatically loads the last file opened, if the Load Last file option is enabled in the **Options** menu.

The ITEE uses a windows like interface to provide a text screen based user interface with multiple windows and pull down menus.

The ITEE can optionally take a single command line argument which is the name of the text file to open. For example: **68ed test.asm** 

2.1. The ITEE Windows User Interface

The ITEE windows user interface provides a powerful text based user interface which can be used either by keyboard alone, or with a combination of keyboard and mouse. For new users most operations will be simpler with the mouse but with experience, hot key sequences for the most commonly used functions will speed the operation of the ITEE along.

2.1.1. Pull Down Menus

The ITEE includes a system of pull down menus which are based around a menu bar across the top of the screen. The menu bar is broken into a series of words or symbols, each of which is the title of a single pull down menu.

The menu system can be used with the keyboard, mouse or a combination of mouse and keyboard actions. The following keyboard and mouse actions are supported.

ACTION	KEY	MOUSE
Open Menu		Press the left button in the menu bar over the appropriate title
Escape from Menu		Press the left button outside the menu system
Next Menu		Select next title with mouse and press the left button
Next Item in Menu		Select title with mouse and press left button
Previous Item in Menu		Select title with mouse and press left button

Table 1 - Menu System Actions

2.1.2. Keyboard Menu Selection

Selecting a menu item by keyboard is accomplished by pressing the  key to open the menu system then using the arrow keys to move the desired menu and highlight the item required. When the item required is highlighted it can be selected by pressing the  key.

2.1.3. Mouse Menu Selection

The menu system is opened by moving the pointer to the title of the menu which is desired and pressing the left mouse button. Once the pull down menu has been operated move the mouse pointer over the required item in the drop down menu and press the left mouse button to select the item.

2.1.4. Menu Hot Keys

When browsing through the menu system you will notice that some menu items have hot key sequences displayed. For example, the ITEE menu item **File Open** has  as a hot key while **Find** has a hot key sequence of  .

When a menu item has a key equivalent, it can be selected directly by pressing that key without opening the menu system. For example to activate the **File** menu,  . The following menu hot keys are supported.

KEY	FUNCTION
	Help
	Save
	Open
 	Close active file
	Run Simulator
	Run Programmer
	Toggle Edit / Output Window
	ASCII Table
	Calculator
	Assemble
	Menu

Table 2 - Menu Commands

2.1.6. Windows

The size of the text editor window is fixed to one of two sizes. When the message output window is displayed, the editor window is reduced in size to fit the message output window on the bottom of the display.

When the message output window is closed the editor window occupies all the available screen space.

All edit windows have a close button displayed on the top left hand side of the window, a scroll bar down the right side of the window and a window title.

The mouse can be used to select windows, operate the scroll bars, menu system and close windows using the windows close button.

Note the mouse cursor cannot be used to position the editor cursor in the edit window or select the title options from the bottom status bar.

2.1.7. Status Bar

The status bar, located at the bottom of the screen displays the cursor co-ordinates, selected micro controller type and important Hot Keys. The mouse has no effect on the status bar.

2.2. The Editor

The text editor is based in its own window, known as the edit window. The text editor can simultaneously edit up to 5 text files at once. Only one text file can be displayed on the screen at any one time. The **Window/List All** menu option is used to swap between open text files.

The text editor uses a disc virtual buffer cache technique which allows very long text files to be edited. Text files can be up to 4Mb in length. Only part of a large text file is loaded into memory for editing at any one time. As required different parts of the text file are automatically shuffled into and out of memory.

2.2.1. Auto Indent

The auto indent feature ensures that every time a new line is added the cursor will be aligned under the first non-space character in the preceding line.

2.2.2. Keyboard Commands

The editor accepts a number of keyboard commands, broken up into the following categories, Cursor Movement, Insert/Delete, Search, File commands and Block/Clip Board operations.

Table 3 provides an overview of the available **keyboard commands**, *Table 4* provides **cursor movement** commands while *Table 5* lists **Insert** and **Delete** commands and their mappings.

KEY	FUNCTION
 	Switch to File 1
 	Switch to File 2
 	Switch to File 3
 	Switch to File 4
 	Switch to File 5
 	Store Book Mark
 	Recall Book Mark
 	Repeat last Search or Replace
 	Start Search
 	Start Search with Replace
 	Bring up File menu
 	Assemble File
  ,  	Bring up Edit menu
 	Bring up Option menu
 	Bring up Window menu
 	Bring up Help menu
 	Go to Line Number

Table 3 - Miscellaneous ITEE Commands

KEY	FUNCTION
 or 	Character left
 or 	Character right
 or 	Word left
 or 	Word right
 or 	Line up
 or 	Line down
	Scroll up
	Scroll down
 or 	Page up
 or 	Page down
	Move cursor to start of line
	Move cursor to end of line
	Go to start of file
	Go to bottom of file
	Go to top of current buffer
	Go to bottom of current buffer

Table 4 - Cursor Movement Commands

KEY	FUNCTION
 or 	Insert mode on/off
	Insert line
	Delete line
	Delete to end of line
 or 	Delete character left
 or 	Delete character
	Delete word right

Table 5 - Insert & Delete Commands

2.2.3. Book Mark

The book mark command, as listed in *Table 3* stores the current cursor line position,  so that the screen can be recalled, . This is very useful when moving around in a program and having to continually return to the same line in a program.

2.2.4. Insert Mode

The Insert key, toggles the cursor mode operation. When Insert Mode is selected, the cursor becomes a block cursor and overwrites any existing text at the cursor. Normally the cursor shifts any text at the cursor while typing.

2.2.5. Block Commands

The editor supports block operations and clip board operations.

The clip board is implemented as an invisible buffer. The contents of the clip board cannot be viewed, but can be pasted into an existing text file. The block operations are described in *Table 6* below.

KEY	FUNCTION
	Toggle Mark mode on/off
	Copy marked text to buffer
	Cut marked text to buffer
	Paste text
	Move cursor up 1 line and Mark
	Move cursor down 1 line and Mark

Table 6 - Block Commands

The block mark operation is performed by moving the cursor to the appropriate position in the text with the cursor keys and selecting  to toggle block mode on. Move the cursor to the end of the text required for block operation and select either Copy or Cut with the  or  keys. Copy as implied, copies the marked text in to the clip board buffer while Cut removes the highlighted text from the editor and stores it in the clip board buffer.

The contents of the clip board buffer can be pasted into the text by moving the cursor to the appropriate position and using the  sequence.

2.3. = System Menu

The (=) System menu contains system features such as an ASCII chart display, Calculator and access to the simulator, programmer disassembler and diagnostic software packages.

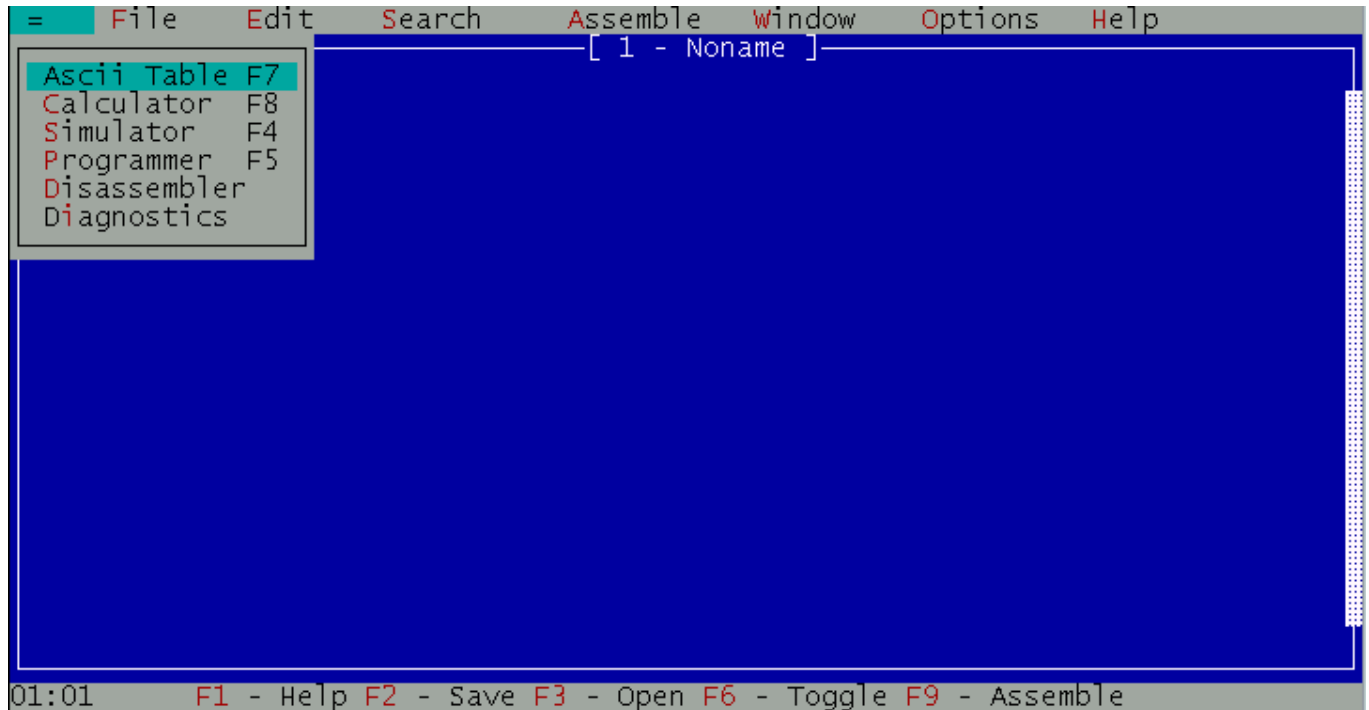


Figure 2 - System Menu

2.3.1. ASCII Chart Display

This option displays a window which contains an ASCII look up table. The cursor keys are used to move the cursor to the required ASCII character. The ASCII character is displayed in binary, decimal and hexadecimal equivalents. The  key exits the ASCII chart window.



Figure 3 - ASCII look up table

2.3.2. Calculator

This option selects the software programmers calculator. This is a multi display integer calculator capable of performing calculations in bases 2 (binary), 10 (decimal) and 16 (hexadecimal).

The results of each calculation are simultaneously displayed in all three bases. Operation is just like a real calculator, just press the buttons on the screen using the mouse or keyboard. The operators have the following functions as listed in *Table 7*.

OPERATOR	FUNCTION
&	Logic AND operator
	Logic OR operator
^	2's complement
~	1's complement
<	Shift Left
>	Shift Right

Table 7 - Calculator Operators

The calculator window can be moved with the mouse by positioning the mouse cursor over the top of the window, dragging the mouse and holding the left button.

The calculator window can be closed by clicking the EXIT button, pressing the  key or pressing the  key.

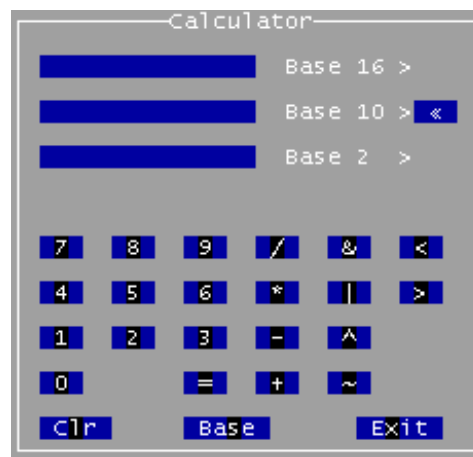


Figure 4 - Programmer's Calculator

2.3.3. Simulator

This option runs the simulator which is a separate software package. The simulator will only be loaded if the current text file has been assigned a micro controller type. Micro controller types can be selected from the **Options** menu.

The current text file (.ASM) must also have been successfully assembled and a .S19 file created for the simulator to operate. Refer to *Section 3* for details on the simulator operation.

2.3.4. Programmer

This option runs the EPROM programmer which is a separate software package. As with the simulator a micro controller type must be selected and .S19 file must be available for loading into the programmer. Refer to *Section 4* for details on the programmer operation.

2.3.5 Disassembler

This option runs the external disassembler program. The disassembler takes a standard .S19 file and creates three new output files.

The three files include a full listing of the source code (.dis extension) with embedded register equates, a binary image (.bin) and a text memory dump (.txt) of the file.

The disassembler requires a .S19 file and chip type to be passed to it from the ITEE.

2.3.6. Diagnostics

The Diagnostics Utility is an external program which verifies correct operation of the programmer board and PC serial and parallel ports.

2.4. File Menu

The File menu contains commands to open, save, repaginate, print and display statistics on files.



Figure 5 - File Menu

2.4.1. Open

This option loads a file into the editor. A file selection box is opened as in *Figure 6* which allows files to be selected. The file selection box defaults to the .ASM extension when loading files. The mouse or keyboard cursor keys are used to select the file to open.

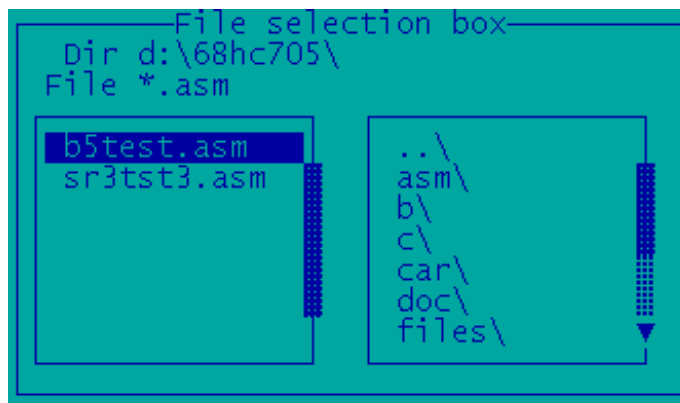


Figure 6 - File Open Selection Box

2.4.2. New

The New command creates a new edit file.

2.4.3. Save

Save the current edit file.

2.4.4. Save As

This command is similar to the Save command except that the file selection box is opened and a new file name is requested.

2.4.5. DOS Shell

This command clears the screen and shells the ITEE to the DOS command prompt line where DOS commands can be issued to the operating system. The DOS **exit** command is used to restore the ITEE.

2.4.6. Print File

This command prints the active file to the LPT1 or LPT2 printer.

2.4.7. Repaginate

This command rebuilds the virtual text file and splits it into manageable sizes for loading into memory. From time to time it is necessary to repaginate a file especially when adding text to a large file. If the text editor buffer is filled then repaginate the file to add more text.

2.4.8. Statistics

This option displays statistical information about the editor and open text file.

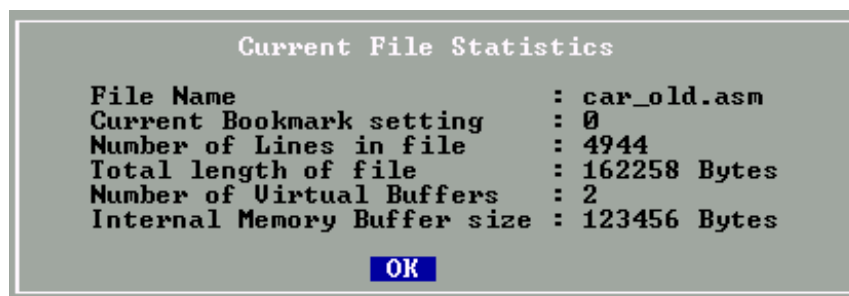


Figure 7 - File Statistics Window

2.4.9. Close File

The Close File option is used to close the current text file.

If a recent copy of the file has not been saved to disc, then the ITEE will prompt you if you want to replace the existing file on the disc with the new session. If you select NO then the file is saved with a .NO extension otherwise it writes over the original file opened.

2.4.10. Exit

The Exit command closes all open files and exits the editor to DOS. If any of the open files have been modified since the last save, a prompt to save changes is displayed.

2.5. Edit

The Edit menu contains text editing commands as displayed in *Figure 8* below.

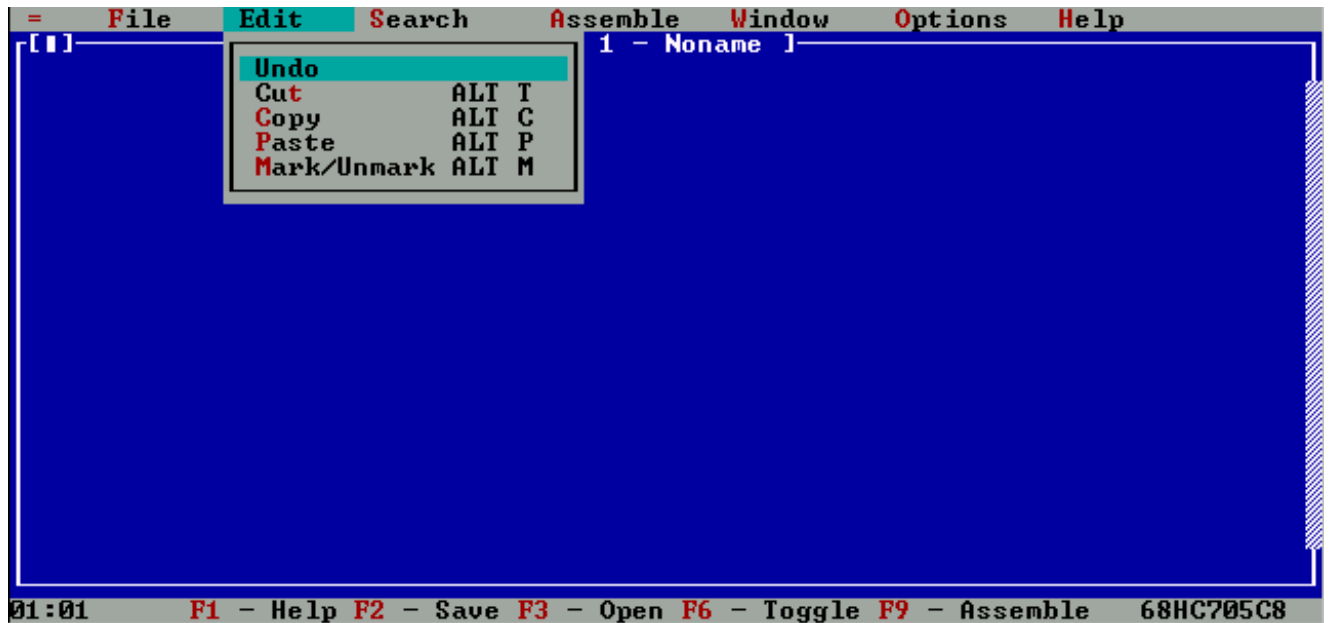


Figure 8 - Edit Command

2.5.1. Undo

This command will restore the text back to the original format after an edit command has been performed. Only the last edit command is stored in the Undo buffer.

2.5.2. Cut

This option copies the high lighted text into the clipboard buffer and deletes it from the editor window. Previous contents of the clip board buffer are lost.

2.5.3. Copy

This command copies the highlighted text into the buffer and deletes it from the editor window. The highlighted text is restored. Previous contents of the clip board buffer are lost.

2.5.4. Paste

This command pastes the contents of the clip board at the current cursor position. The contents of the clipboard are not lost.

2.5.5. Mark/Unmark

The block Mark/Unmark operation is performed by moving the text editor cursor to the appropriate position in the text with the cursor keys and selecting   to toggle block mode on. Move the cursor to the end of the text required for block operation and select either Copy or Cut with the   or   keys. If no marking is required the   key sequence can be used to toggle the Block marking off.

2.6. Search

The Search menu is used to find text, replace text and move the cursor to any position within the editor.

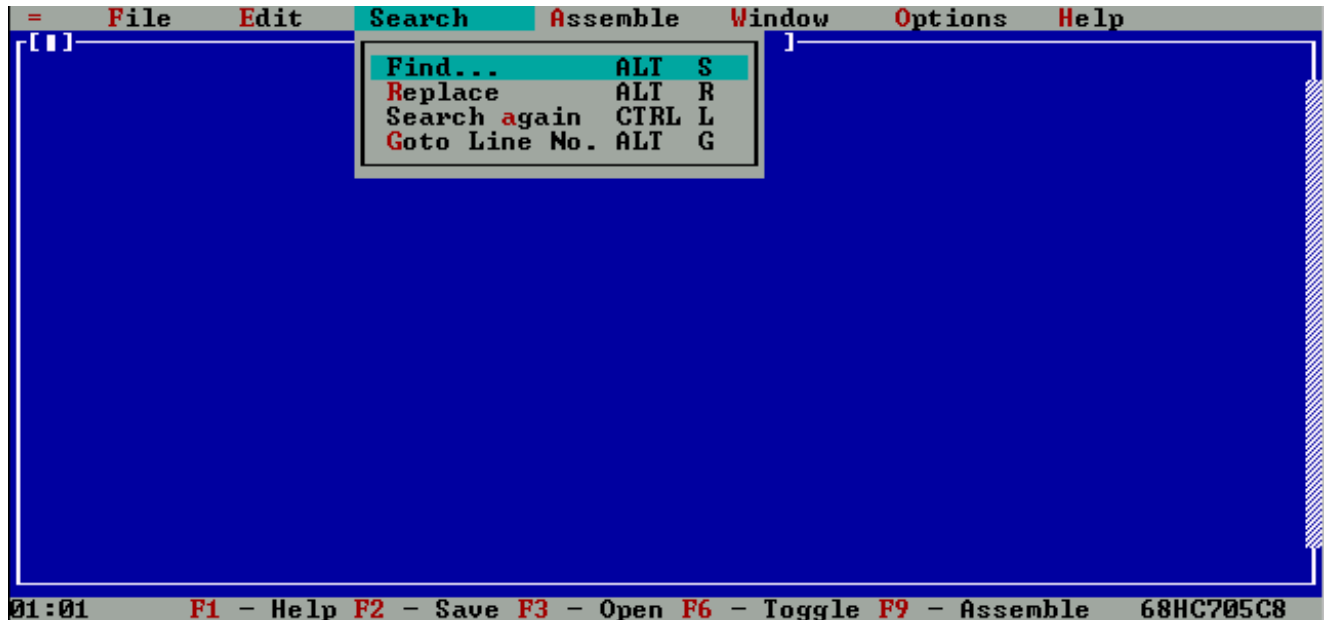


Figure 9 - Search Menu

2.6.1. Find

This option searches the editor for a text match as displayed in Figure 10 below.

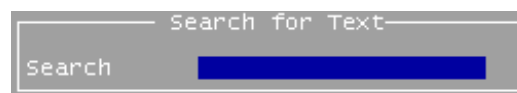


Figure 10 - Search for text window

2.6.2. Replace

This option performs a Search and Replace text function as displayed in Figure 11 below.

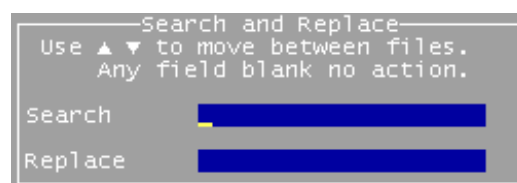


Figure 11 - Search and Replace

2.6.3. Search Again

This option continues the search from the current text cursor position until another match is found or the end of the file is reached.

2.6.4. Go to Line Number

This option moves the cursor to any line number within the editor as displayed in *Figure 12* below.

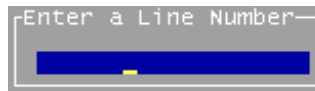


Figure 12 - Move cursor to a line in the text editor

2.7. Assemble

The assemble menu is used to assemble and view the output .LST file of the cross assembler.

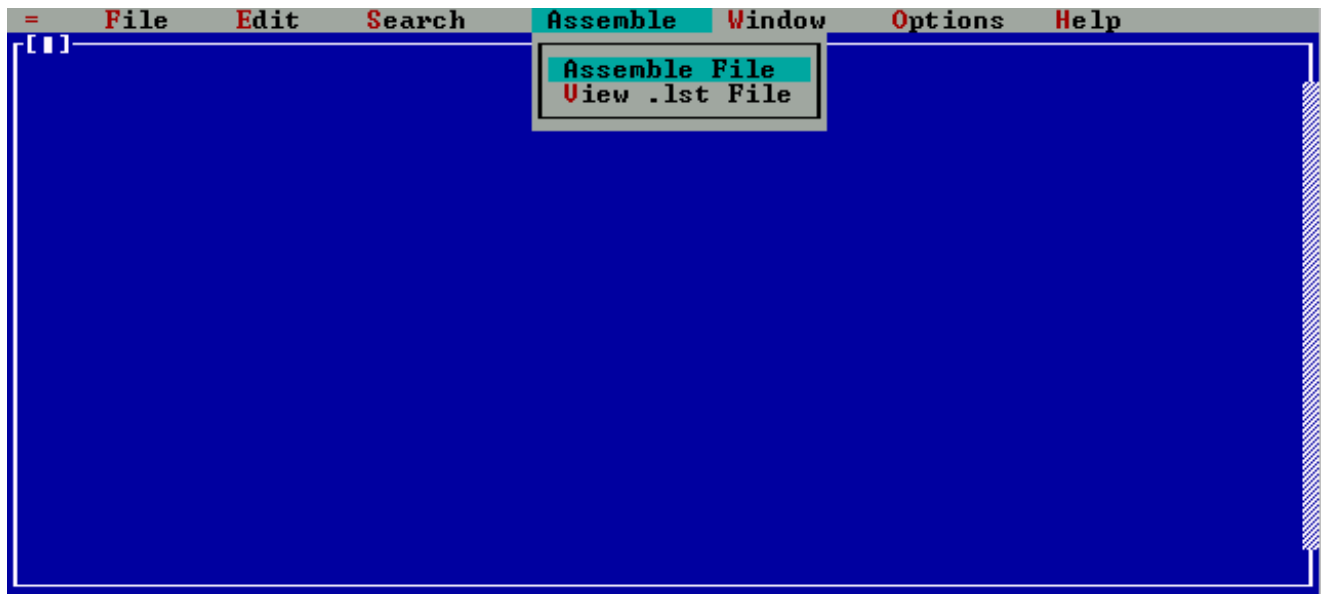


Figure 13 - Assemble Menu

2.7.1. Assemble

The Assemble menu runs the cross assembler program as displayed in *Figure 14*. From this window the .ASM text file (source code) is assembled into 68HC05 code by selecting the **Begin Assembly** button using either the mouse or keyboard.

A cross assembler is a program that takes assembly language source program statements written and edited on one computer (the host) and translates them into executable code for another computer (the target). In this case the host computer is a PC and the target computer is a 68HC05 micro controller.

A .S19 file is a standardised Motorola data format which encodes program and data object modules into a printable (ASCII) format. This allows viewing of the object file with standard tools and allows display of the module while transferring from one computer to the next or during loads between a host and target.

A List file contains the original source code (text) with the assembled machine code instructions. An optional List file can also be created by selecting the **Produce List File** command.

If assembly is successful a complete .S19 file is written to the disc under the same director as the .ASM file with the extension of .S19.

The output of the cross assembler is captured and stored in the file: data, under the root directory.

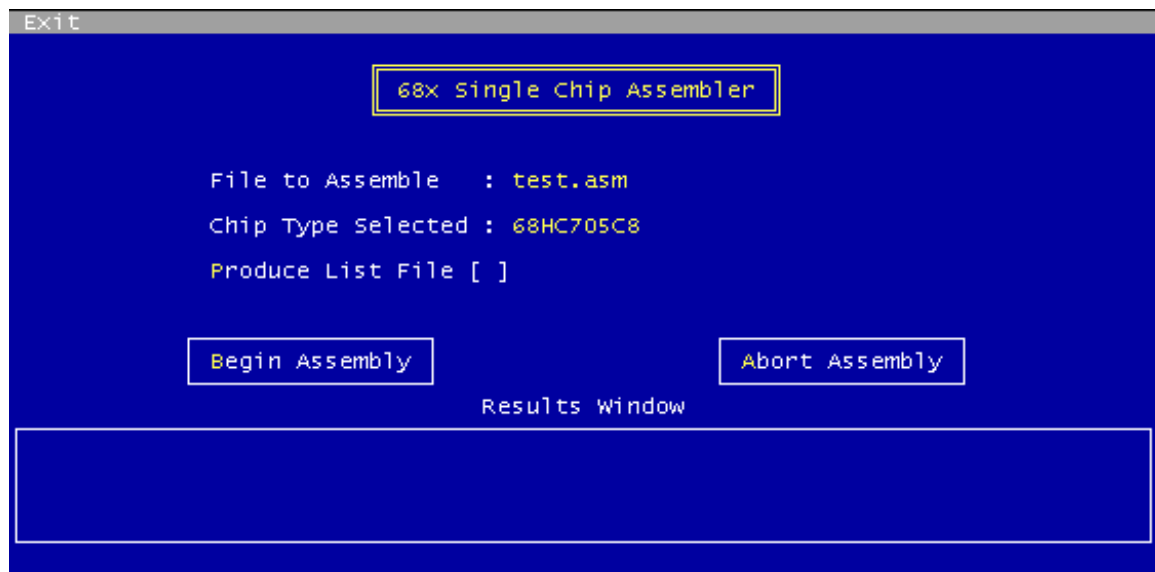


Figure 14 - Assembler

After a source text file (.ASM) has been assembled the ITEE is re-loaded and an Output Window is opened as shown in Figure 14. Any messages the cross assembler produced are displayed in this window.

Assembly errors can be corrected in the editor source code by highlighting the line to edit in the Output Window and pressing the key to automatically position the cursor in the editor ready for correction.

The mouse can be used to select between the text editor window and the output window or the key can be used to toggle between the two windows as required.

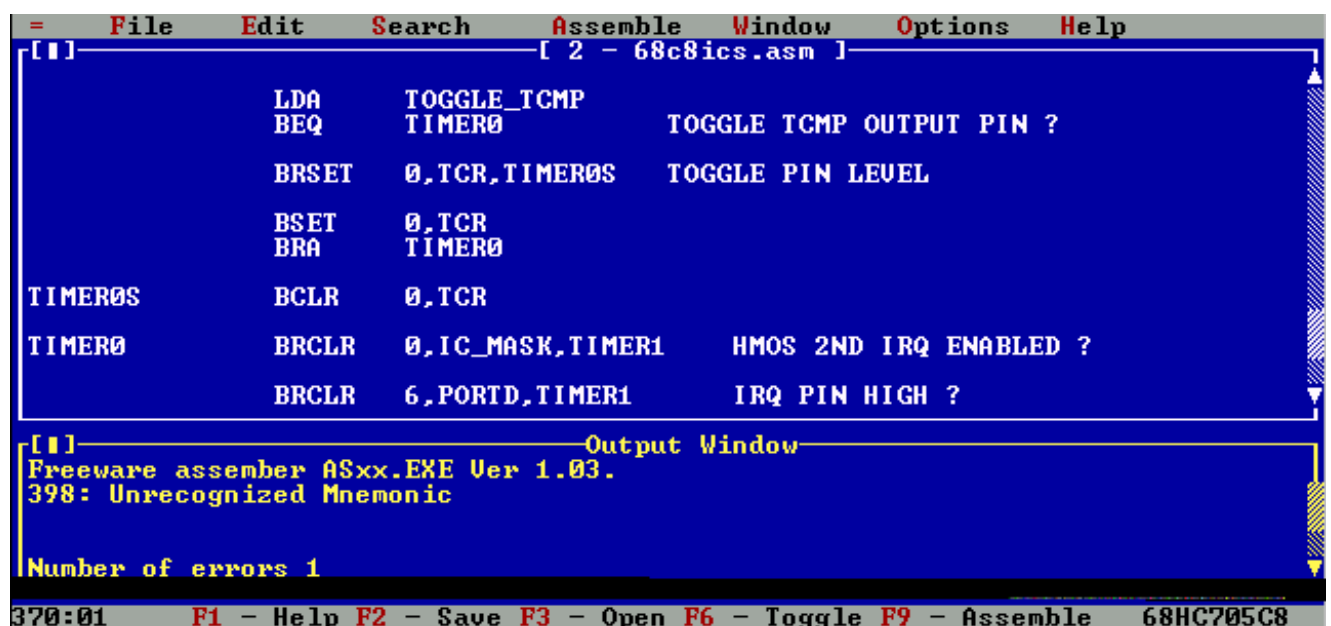


Figure 15 - Auto line position

2.7.2. View .lst file

The view list file command is used to display the .lst file that the cross assembler has produced.

The .lst file consist of the original source code (.asm file) with the compiled assembly instructions and line numbers embedded into the margin.

To produce a .lst file, first assemble the source code to check that there are no errors, then select the produce list file option.

The .lst file is displayed in the standard text editor buffer provided that no more than 5 files are open at any one time.

2.8. Window

The Window menu as displayed in *Figure 16* is used to select a window to display.

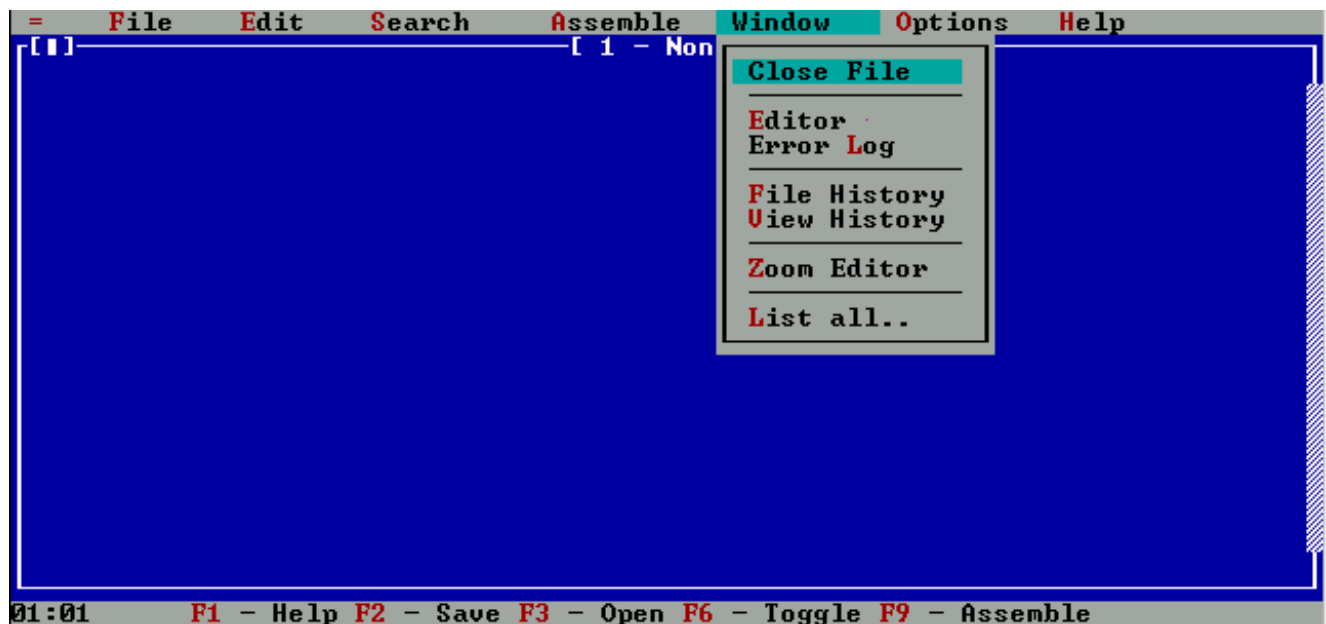


Figure 16 - Window Menu

2.8.1. Close File

The close file option closes the active window. This option has the same effect as the close file command under the File menu.

2.8.2. Editor

Selecting **Editor** moves the cursor into the editor window if opened.

2.8.3. Error Log

Selecting **Error log** moves the cursor into the Output Window if opened.

2.8.4. File History

The file history option calculates and displays the total of all the session times and writes these details in a *report.log* file. This file also has details on the total edit time and number of times assembled.

Note that the history file (.his) is only written to disc when the current edit session is closed. Therefore a new file, must first be saved and closed before a report can be produced.

2.8.5. View History

The view history option displays the report file which contains information on session times, total edit time and number of times a files has been assembled.

2.8.6. Zoom Editor

Zoom editor toggles between full screen editor display and split screen editor and message Output Window display.

2.8.7. List all..

The **List all** option activates a list box which is used to select an already open text file for editing. Up to 5 text files can be simultaneously opened.

2.9. Options

The options menu as displayed in *Figure 17* is used to configure the ITEE, select the micro controller type and display File History details.

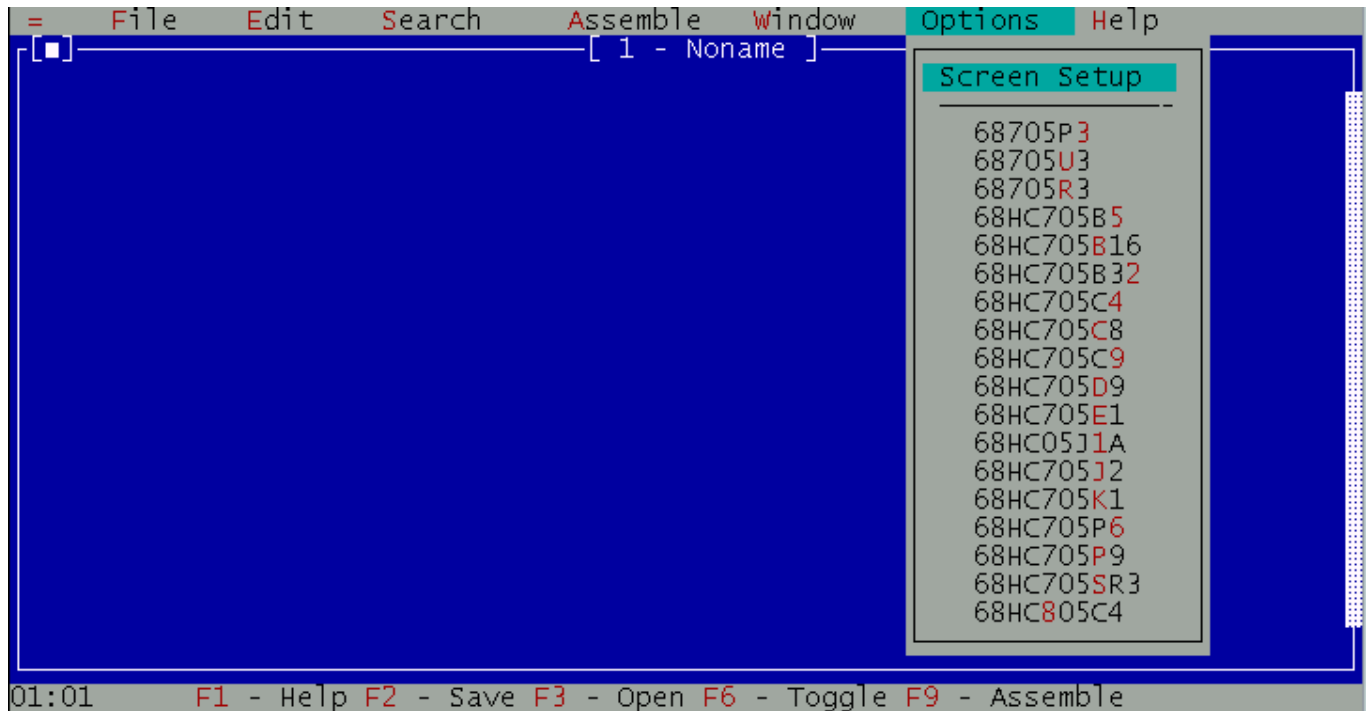


Figure 17 - Options Menu

2.9.1. Screen Setup

The screen setup option as displayed in *Figure 18* is used to configure the ITEE screen text resolution, color attributes, horizontal scroll bar,  key spacing and Load Last file options. Use the mouse or keyboard to select the appropriate item to change.

Note for correct disassembler formatting (.dis) the TAB spacing must be set to 6 positions or higher.

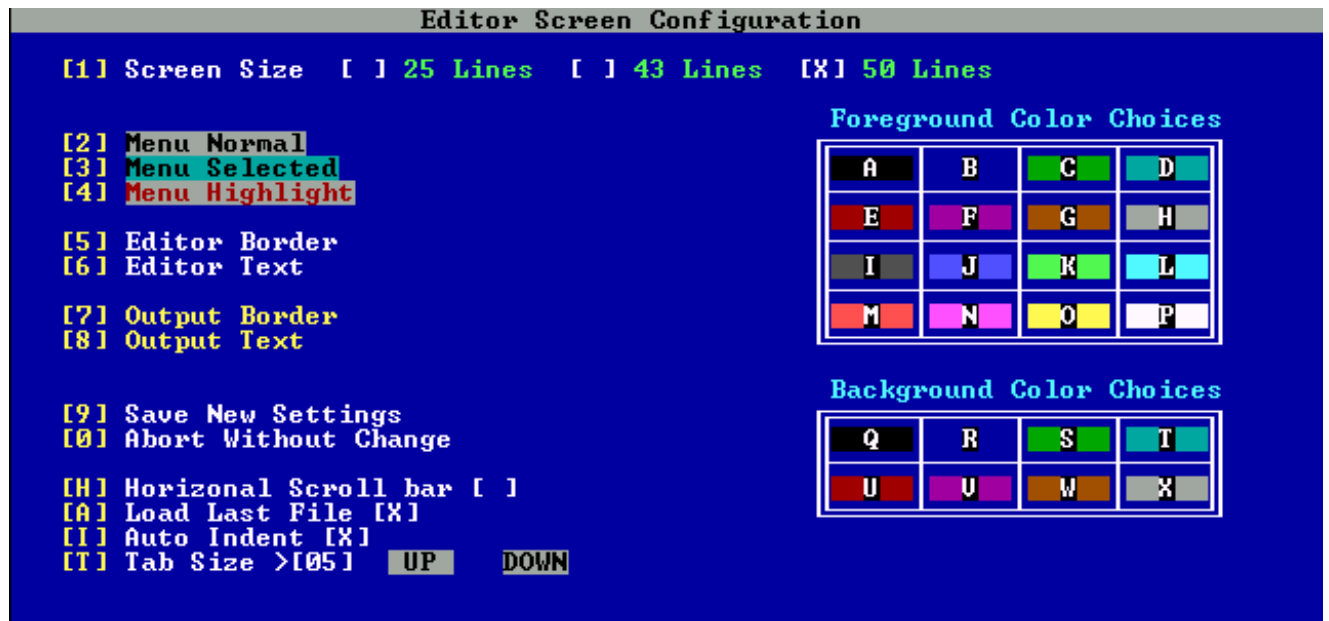


Figure 18 - Screen Setup

2.9.2. Micro Controller Type

The micro controller type is selected from the **Options** window by selecting the micro controller with the mouse or keyboard. The selected controller type is displayed in the bottom right hand side of the status bar.

2.10. Help

The Help menu as displayed in *Figure 19* is used to display the Help system, software version and registered user details. Selecting **Contents** brings up the Help System where Micro Controller Technical Data, assembly instructions and general features of the ITEE can be quickly referenced.



Figure 19 - Help Menu

2.10.1. Contents

The Help Contents option as displayed in *Figure 20*, is the main index to the Help system. From this window, the highlighted topics can be selected with either the keyboard or mouse.

The cursor keys are used to move the cursor and the  key is used to select a topic. The  key exits the help window while the  (backspace) key is used to navigate back to the previous Help topic.

The mouse can be used to select a topic by simply moving the mouse cursor over the topic to display and pressing the left button.

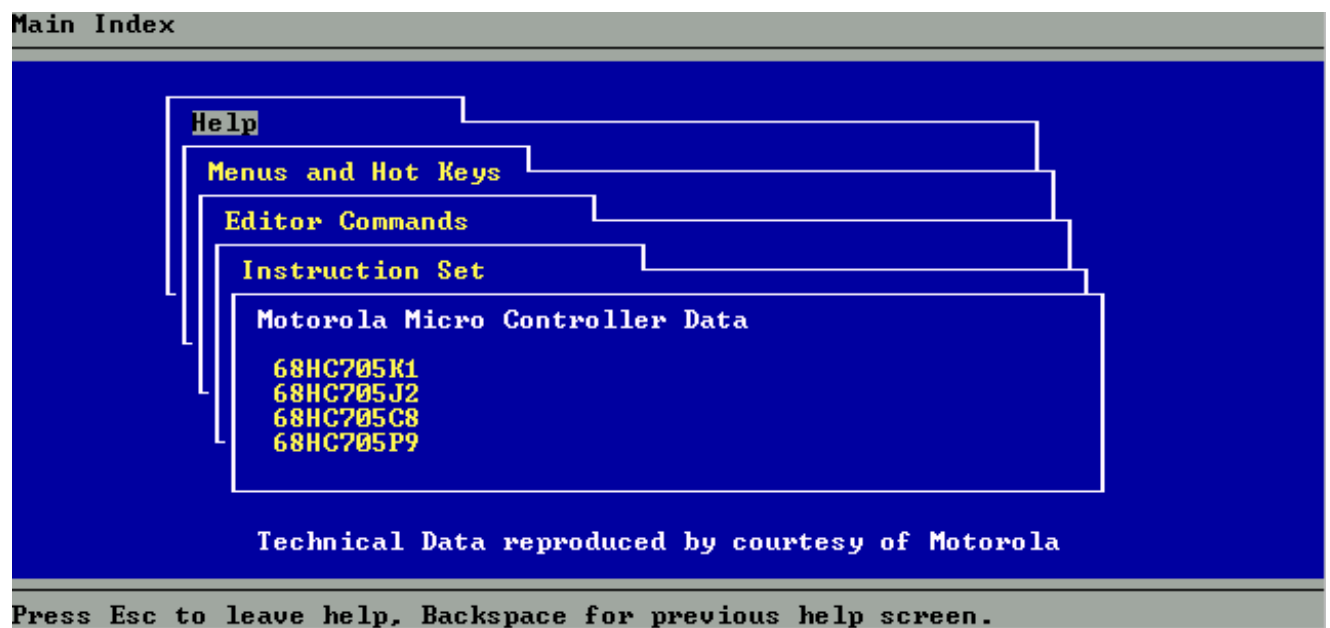


Figure 20 - Help Main Index

3.0 Software Simulation

Simulators are important software development tools for testing and debugging programs as they make it easier to trace errors which are difficult to assign to hardware or software.

The Oztechnics simulator (*68SIM05.EXE*) allows you to test and debug your programs on a PC before programming them into a micro controller.

The simulator allows you to run a program line by line (step) or to a specific address (breakpoint). All of the various registers, memory locations and I/O ports can be viewed and modified.

The simulator allows the source code of a program to be viewed on the screen together with the simulator trace screen. Display values can be formatted in either hexadecimal, decimal or ASCII.

Special features of the simulator include undoing previously executed instructions (step backwards), trapping illegal instructions/addresses, stack over flows, writing to RAM locations used by the stack and writing to illegal addresses such as PROM or read only registers. All of these features greatly assist in software testing and debugging.

3.1. In Circuit Simulation

In Circuit Simulation (ICS) is similar in concept to a Software Simulator except that all I/O processing is performed by an external device to the PC which maintains port values In Circuit (hardware). What is displayed on the simulator screen is directly reflected in hardware.

When the simulator requires I/O operations a data request is made through the serial port to a 68HC705C8 micro controller on the programmer board. The controller running a special monitor program responds to this request and sends data back to the PC via the serial port if required. This arrangement allows the simulator to interface with the real world.

With **68SIM05**, a special In Circuit Simulation mode (ICS) can be entered using the  command. While in this mode all input and output processing of port values is performed by a C8 controller on the programmer board.

In Circuit Simulation allows any supported controller to be software simulated on the PC and hardware emulated by a C8 controller executing a monitor program (68/CSC8.S19) that communicates with the simulator package via the serial port. The I/O pins of the C8 controller are extended via the In Circuit Emulator (ICE) socket to the target hardware.

This ICE socket can be converted by use of a suitable cable and adaptor board so that other micro controller pin outs can be emulated by the C8 controller. Adaptor boards are available for the K1, J1/J2, P6/P9 and U3/R3/SR3 micro controllers.

ICS allows real hardware simulation to be performed.

Advantages

- Excellent for debugging code. The simulator has many features which make debugging code easy.
- Excellent for stepping through code and exercising hardware.
- Low cost compared to a real time hardware In Circuit Emulator .
- Many micro controllers can be supported from one low cost platform.

Disadvantages

- Some hardware limitations such as no A-D converter for the SR3 and no SCI or SPI hardware emulation for the C8 controller.
- Slow. Simulation speed depends on the speed of the PC.
- C8 controller I/O electrical characteristics may not be identical to other micro controller types being simulated.
- Some hardware and software faults can only be trapped in real time.

3.1.1. ICS Operation

1. Connect power, serial & parallel cables to PC
2. Place a C8 controller in the C8 ZIF socket on the programmer board
3. Burn 68/CSC8.S19 into the C8 controller
4. Run simulator and load user .s19 file
5. Select the  command. Simulator initialises programmer board for ICS operation. ICS displayed on screen if successful. Run simulator as normal.

3.2. In Circuit Emulation

In Circuit Emulation (C8/C9/D9 controllers) involves executing a user program in a micro controller in Real Time.

The programmer board has a In Circuit Emulation socket for the 68HC705C8 which is used to plug into a target system under development.

In Circuit Emulation (ICE) for the C8 micro controller is achieved by adding an interrupt driven monitor routine (*68C8MON.ASM*) to the source code under development. This monitor routine communicates to the PC via the serial port.

When a BREAK command is sent from the PC (via the serial port) or a break point is reached (SWI instruction), the C8 controller executes the interrupt monitor routine, causing the contents of RAM, registers and I/O status of the controller to be transmitted back to the PC for evaluation.

Advantages

- Excellent for real time evaluation of both hardware and software.
- Fast evaluation of routines (Real Time).
- Some hardware / software faults can only be trapped in real time.
- Low cost compared to complex hardware In Circuit Emulator's.

Disadvantages

- The SWI instruction has to be inserted into the program at the appropriate locations (break points) before programming the C8 controller PROM.
- The serial port (SCI) of the C8 controller cannot be used for the development application as it is used exclusively for monitoring.
- Programs cannot be stepped line by line. Programs under evaluation have to be run in real time to a break point or stopped by issuing a break command from the keyboard of the PC.
- Monitor Interrupt routine requires a small amount of PROM and RAM.
- Not as flexible as a true hardware In Circuit Emulator.

3.3. 68SIM05: In Circuit Software Simulator

To run the Oztechnics In Circuit Simulator type **68SIM05**  from the directory where the programs were installed or select the simulator from the ITEE system (=) menu.

The full parameters of this command are:-

68SIM05 [.s19 file name [micro controller type]]

A title screen will appear which displays the simulator version and user registration details if no .S19 file and micro controller type are specified.

The supported micro controller families are listed in *Table 8*.

Micro Controller	DOS Command line argument	Shortcut Key Definition
68705P3	P3	3
68705U3	U3	U
68705R3	R3	R
68HC705C8	C8	C
68HC705C8A	C8A	A
68HC705J1A	J1A	1
68HC705J2	J2	J
68HC705K1	K1	K
68HC705B5	B5	5
68HC705B16	B16	B
68HC705C9/C9A	C9	9
68HC705D9/D9A	D9	D
68HC705E1	E1	E
68HC705P6/P6A	P6	6
68HC705P9/P9A	P9	P
68HC705SR3	SR3	S

Table 8 - Command line arguments for supported Micro Controllers

For example *68SIM05 DEMO1 C8* will automatically load the *DEMO1.S19* file and specify the C8 micro controller. Note that the .S19 file extension is not necessary.

After successfully loading a .S19 file the first instruction to be executed after a hardware reset will be displayed at the bottom of the simulator screen.

If the Reset Vector points outside the User EPROM, then a error message is displayed indicating that the Reset Vector is incorrectly defined.

If an invalid instruction is displayed or no instruction is displayed then the reset vectors have been incorrectly defined, the wrong micro controller type has been specified or the .S19 file loaded is invalid.

3.3.2. Memory Display Window

This area is displayed with a blue background and a white border. It displays a memory dump of the controller. The address locations are displayed down the left hand side of this area with address offsets across the top. Memory locations are only displayed in hexadecimal format.

For memory addresses in the range \$00-->\$ff0 the address locations down the left side of the memory map are displayed in Blue, address locations > \$ff0 are displayed in Red.

Normally this area is used to view the RAM contents of the controller. The window can be shifted through the complete range of addresses of the controller by using the         keys.

The memory display window can only be shifted to display a new range of addresses if the view source code command  is not in use as both use the same keys for positioning the display.

The memory display window defaults to displaying the RAM when a file is loaded.

When the RAM is written too, the byte being written is highlighted in RED. Any RAM byte which is currently used by the stack is highlighted in BLACK+CYAN.

RAM bytes previously used by the stack are displayed in CYAN. This is useful for determining how many bytes need to be reserved for the stack.

Different memory types are colour coded for easy identification. The colours used to identify the different areas in the memory map are listed in *Table 9* below.

MEMORY TYPE	COLOUR CODE
I/O Area / Registers	RED
RAM	WHITE
RAM (undefined)	GREEN
current stack RAM	BLACK+CYAN
previous stack RAM	CYAN
EPROM	YELLOW
EEPROM	GRAY
MOR	MAGENTA
Boot ROM	LIGHT BLUE
EPROM vectors	GREEN
Unused	BLACK

Table 9 - Memory Map Colour Codes

The contents of the window can be displayed in HEXadecimal, DECimal or ASCII formats using the  command.

Note that only the RAM locations are updated in real time. If viewing the I/O ports area, the memory map display can be updated by the use of the         keys.

3.3.3. I/O Ports Area

The I/O ports area is displayed with a cyan background and consists of two areas. The top right hand corner of the screen shows the I/O ports in a binary format.

The bits displayed for the I/O ports are colour coded where BLUE represents an output bit and RED indicates an input bit (as defined in the Data Direction Registers).

The second area shows the value of the I/O ports in hexadecimal, decimal or ASCII format depending on which display format has been selected using the Format command, .

3.3.4. CCR Area

The Condition Code Register (CCR) area has a red background and shows the status of the flags in the 68HC05.

The flags that make up the 68HC05 CCR include:-

H	Half carry (from BIT 3)
I	Interrupt Mask
N	Negative
Z	Zero
C	Carry

3.3.5. Registers

The Registers area has a green background and shows the contents of the X Register, Accumulator, Stack Pointer and Program Counter.

This area is also used to display the Timer Data Register contents for HMOS devices or the Timer Counter and Timer Compare registers for the HCMOS devices.

3.3.6. Status Line

The Status Line (red background) located below the memory dump area shows the current status of the simulator. Typical Status Line information includes:-

- program running / halted;
- trace / display on / off;
- log file on /off;
- display format HEXadecimal / DECimal or ASCII;
- breakpoints [1-3]; and
- <file name> currently loaded (.LST / .ASM / .S19).

3.3.7. Command Line

The Command Line (white background) located below the Status Line, is where you enter instructions and values into the simulator. Messages are also displayed on this line.

3.3.8. Trace Area

The trace is located in the bottom half of screen with a blue background and is where each line of code being executed is displayed.

The second line from the bottom of the display represents the current instruction being executed while the line at the bottom of the screen, highlighted in CYAN, represents the next instruction to be executed. As each instruction is executed, the display scrolls up.

This portion of the screen can also be used to display the source text file (.LST) if the display source code command, **[A]** is invoked. When the source text file is displayed the bottom three lines of the screen are used for the trace (blue background) with the remaining portion of the screen used for displaying the source text file (cyan background).

The simulator trace is broken up into 10 columns. A typical screen display is represented below:-

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
195	26	fa		↑	BNE	\$191		872	138
191	ff				STX	,X	086<=86	876	138
192	5C				INCX		X_REG=>86 X_REG<=87	879	138
193	A3	00			CPX	#\$00	194=>00	881	138
195	26	fa		↑	BNE	\$191		884	138

Columns

- C1 Address of instruction
- C2 Opcode (hex)
- C3-4 Operand (hex)
- C5 Jump, branch to new address indicator (up or down in memory relative to current Program Counter)
- C6-7 Assembly Mnemonics
- C8 Instruction results
- C9 Total Clock Cycle Count (decimal format)
- C10 Total Clock Cycle Count for Interrupt Services (decimal for mat)

3.3.9. Micro Controller Type

This area (top right hand corner) displays the selected Micro Controller simulation model. Each micro controller requires a different simulator configuration so it is important to select the correct controller type.

3.4. Display Modes

The simulator has three display modes. To change the display mode use the toggle command,  key to toggle between the different display modes.

For the fastest simulation, the text trace, screen display and log text trace to disc option can be turned off. This mode is useful when you want to run a program to a breakpoint quickly.

The next mode of operation is slower but displays a full text trace of the simulator. Using this option, a text trace or LOG file can be recorded to disc while the simulator is running.

The last mode (default) offers the most detail of the activities of the simulator. Both the text trace and display of the 68HC05 are enabled. The effect of each instruction executed is automatically updated on the display. This is the preferred mode to use when debugging a program.

3.5. Log Files

The simulator log file is an ASCII text file of the simulator trace screen and registers. The default log filename is *68SIM05.LOG* if a name is not specified at the prompt.

Log files are useful for obtaining a hard copy of a program under simulation.

Every time a log file is created, if the filename already exists the old log file is over written.

Exiting the simulator automatically closes the log file.

Note that very large text files can be created on your hard disc when the log file is enabled.

3.6. 68SIM05 Commands

All commands entered into the simulator require only one key stroke. Commands can be entered in either upper or lower case.



Source Code Display

This command is used to display the .LST or .ASM text source file on the bottom half of the simulator screen. When the source code display is enabled the bottom three lines of the screen are used to display a trace of the program.

The         keys are used to position the source code text display.

When source code display is selected the simulator will first search for the .LST file. If this file cannot be found the simulator will look for the original .ASM source file. If both the .LST and .ASM files cannot be found the simulator will report an error.



Set Breakpoint

This command sets a breakpoint. A breakpoint is an address in a program that you want the simulator to stop at. Breakpoints are very useful for debugging code. Up to three breakpoints can be defined at any one time.

When a breakpoint is defined, the address of the breakpoint is displayed in square brackets on the status line.

Breakpoints can be cleared using the clear break point command, .

Do not set a breakpoint for the first line of an Interrupt service routine or a subroutine as the simulator will skip these lines.



Command Summary

This command displays a brief summary of the available commands. The commands available will depend on which micro controller has been selected. When finished viewing the summary, press any key to return to the simulator. This command has the same effect as the  key.



Toggle External INT2 Pin

This command is used to toggle the logic state of the External Interrupt2 Pin of the U3 and R3 micro controllers.

If the INT2 pin is at the correct triggering level and provided that interrupts are enabled (I Flag in CCR = 0) and no other interrupts are being serviced the current instruction is completed then the external interrupt service routine is executed.

If this command is used while servicing another interrupt, it is latched and executed when an RTI occurs.



Toggle External Timer Input Pin

This command toggles the logic state of the External Timer's Input Pin for the P3, U3 and R3 controllers.

The effect of changing the state of the timer's external input depends on the mode set by bits 4 and 5 (TIE and TIN) in the Timer Control Register (TCR).



Toggle TCAP Input Pin

This command toggles the logic state of the TCAP (Timer Capture) input pin for the C8 controller.

If the Input Capture Interrupt Enable Flag (ICIE) [Bit 7 in the Timer Control Register (TCR) is set, a timer interrupt will be generated on the appropriate transition of the TCAP pin.



Display Data Format

This command toggles the display data format on the screen from HEXadecimal to DECimal to ASCII.

Note that the registers (X, Accumulator, SP, PC) are displayed in HEXadecimal format even when ASCII format is selected.



Go

This command starts simulation. The 'Halted' label on the status line changes to 'Running' and simulation commences at the current Program Counter.

Simulation will halt when the Program Counter equals a breakpoint address on an op-code fetch, or if the Halt () or Step () commands are issued. The simulator will also halt if an error is detected.



Step Over Subroutine

This command starts simulation and is used to run through to the end of a subroutine. The 'Halted' label on the status line changes to 'Running' and simulation commences at the current Program Counter.

Simulation will halt when the Program Counter equals a breakpoint address on an op-code fetch, an RTS is execute, or if the Halt () or Step () commands are issued. The simulator will also halt if an error is detected.



Halt Simulation

This command Halts simulation and updates the screen display (registers, RAM, I/O etc). The 'Running' label on the status line changes to 'Halted'.



Toggle External Pin

This command is used to toggle the logic level on the External Interrupt Pin of a micro controller.

Provided that the INT pin is at the correct triggering level and interrupts are enabled (I Flag in CCR = 0) and that no other interrupts are being serviced, the current instruction is completed then the external interrupt service routine is executed.



Reset Cycle Counting

This command resets Cycle Counting to 0. This is useful for measuring the number of clock cycles a particular routines takes to execute.



Reset Interrupt Cycle Counting

This command resets Interrupt Cycle Counting to 0. This is useful for measuring the number of clock cycles a particular interrupt routines takes to execute.



Load a .S19 file

This command is used to load a Motorola .S19 file into the simulator. A file name can be specified at the command prompt or the  key can be pressed to bring up a file list box.

Do not try to load a file format other than a .S19 format. Note that the .S19 file extension is optional.

Once the .S19 file has been successfully loaded into the simulator, the micro controller type must be defined. The supported micro controllers are listed in *Table 8*.



Modify Memory

This command is used to modify the contents of the micro controller's memory and registers. After selecting an address to modify, the value of the address is displayed on the command line. If  is pressed, the next location and its contents are displayed. If a valid value is entered from the keyboard the value is written to the displayed location and the next location is displayed. Press the  key to exit this command.



SCI - Set RDRF (C8 Controller)

This command is used to simulate receiving a byte in the Serial Communications Data Register by setting the Receive Data Register Flag (RDRF) in the Serial Communications Status Register (SCSR).

If the Receive Interrupt Enable Flag (RIE) is set in the Serial Communications Control Register2 (SCCR2), an SCI interrupt will be generated.



Disable all Breakpoints

This command is used to disable all Breakpoints.



Toggle TCAP2 Input Pin

This command toggles the logic state of the TCAP2 (Timer Capture 2) input pin for the B5 controller.

If the Input Capture Interrupt Enable Flag (ICIE) [Bit 7 in the Timer Control Register (TCR)] is set, a timer interrupt will be generated on the appropriate transition of the TCAP pin.



Quit Simulator

This command is used to quit the simulator and return to DOS or the ITEE.



Modify Registers

This command is used to modify the contents of the Registers. Type in the first letter to modify the **P**rogram Counter, **A**ccumulator, **X** register, **S**tack Pointer and **C**ondition Code Register.



Step One Instruction

This command is used to simulate one instruction. This command is similar to the Go command except simulation is automatically halted after one instruction has been simulated.



Step Backwards One Instruction

This command is used to Undo a previously executed instruction. As each instruction is executed with the Go or Step command, the executed instruction and its consequences are recorded in a trace buffer. The trace buffer records the last 550 instructions executed.



Toggle Display Mode

This command is used to toggle the display modes of the simulator. The display modes are (fastest to slowest):-

.	.	.	Trace off, Display off, Log file off
T	.	.	Trace on
T	D	.	Trace on and Display on
T	.	L	Trace on and Log file on
T	D	L	Trace on, Display on, Log file on



Log File

This command toggles the log file on/off. The log file captures the on screen text trace and registers, writing a log file to the disc under the filename *68SIM05.LOG*.



View Registers and Vectors

This command is used to view all the registers and EPROM vectors of the micro controller. A window is created over the memory area to display this information. A HEXadecimal, DECimal and BINary value is displayed for each register.



Configure Communications Parameters

This command is used to configure Parallel and Serial communications parameters between the PC and the programmer board. The default setting are LPT1 and COM1. The configuration is recorded in the file, *68OZTEC.CFG*.



Reset Micro Controller

This command simulates a hardware RESET in a micro controller.



Simulation Speed

This command is used to adjust the speed of simulation. A time delay from 0 to 500mS between each simulated instruction can be used to slow simulation. Default is 0mS delay.



In Circuit Simulation (ICS)

This command toggles between Software Simulation only and In Circuit Simulation / Hardware Emulation. When ICS mode is enabled, a message 'ICS' is displayed above the I/O ports area to indicate that all I/O port operations are being performed by hardware. Refer to *Section 3.1.* for further details on ICS.

Note that Auto Loading an I/O port disables In Circuit Simulation for that port only.



Auto Load PortA

This command can be used at any time to load a value directly into the PortA Data Register without the simulator having to stop and ask for a value from the keyboard.

Note that when enabled, the I/O port blinks on screen to indicate auto load is enabled.



Auto Load PortB

This command can be used at any time to load a value directly into the PortB Data Register without the simulator having to stop and ask for a value from the keyboard.

Note that when enabled, the I/O port blinks on screen to indicate auto load is enabled.



Auto Load PortC

This command can be used at any time to load a value directly into the PortC Data Register without the simulator having to stop and ask for a value from the keyboard.

Note that when enabled, the I/O port blinks on screen to indicate auto load is enabled.



Auto Load PortD

This command can be used at any time to load a value directly into the PortD Data Register without the simulator having to stop and ask for a value from the keyboard.

Note that when enabled the I/O port blinks on screen to indicate auto load is enabled.

5

K1 Controller External Interrupt: PA0

This command is used to latch an external interrupt on I/O Bit0, PortA of the K1 controller.

If the PortA Interrupt Request bit (PIRQ) is set in the Mask Option Register (MOR) an external interrupt will be generated.

6

K1 Controller External Interrupt: PA1

This command is used to latch an external interrupt on I/O Bit1, PortA of the K1 controller.

If the PortA Interrupt Request bit (PIRQ) is set in the Mask Option Register (MOR) an external interrupt will be generated.

7

K1 Controller External Interrupt: PA2

This command is used to latch an external interrupt on I/O Bit2, PortA of the K1 controller.

If the PortA Interrupt Request bit (PIRQ) is set in the Mask Option Register (MOR) an external interrupt will be generated.

8

K1 Controller External Interrupt: PA3

This command is used to latch an external interrupt on I/O Bit3, PortA of the K1 controller.

If the PortA Interrupt Request bit (PIRQ) is set in the Mask Option Register (MOR) an external interrupt will be generated.

9

Auto Load A-D Converter

This command is used to Auto Load a value directly into the A-D converter data register. A separate value can be assigned to each channel of the A-D converter.

The channels that are enabled to auto load are represented by the channel number blinking on the screen under the I/O ports area.

This command only works with controllers that have an A-D converter.



Auto Load SCI Data Register

This command can be used at any time to Auto Load a value directly into the Serial Communications Data Register (SCDAT).

When SCI auto load is enabled, a blinking message 'SCI' is displayed on the command line.

This command only works with controllers that have a Serial Communication Interface.



Auto Load SPI Data Register

This command can be used at any time to Auto Load a value directly into the Serial Peripheral Data Register (SPDR).

When SPI auto load is enabled, a blinking message 'SPI' is displayed on the command line.

This command only works with controllers that have a Serial Peripheral Interface.



Set Accumulator Break Value

This command is used to stop the simulator when the value in the Accumulator equals the value set as the break value. The Accumulator Break value is displayed on the status bar.



Set X Register Break Value

This command is used to stop the simulator when the value in the X Register equals the value set as the break value. The X Register Break value is displayed on the status bar.



Modify Accumulator

This command is used to modify the Accumulator. It is a hot key to replace selecting the Register Command then selecting the Accumulator.



Modify CCR Register

This command is used to modify the CCR Register. It is a hot key to replace selecting the Register Command then selecting the CCR Register.



Go to Line Number in Source Code

This command is used to quickly move to a line number with in the source code. This command only works if the source code is displayed.



Modify Program Counter

This command is used to modify the Program Counter. It is a hot key to replace selecting the Register Command then selecting the Program Counter.



Modify X Register

This command is used to modify the X Register. It is a hot key to replace selecting the Register Command then selecting the X register.



Toggle Port A Bit 0 - Keyboard Interrupt (SR3 controller only)

This command is used to toggle Bit 0 of Port A if the corresponding bit in the Keyboard Interrupt Mask Register (KBIM) is set. If the Keyboard Interrupt is enabled (KBIE=1, bit 7 set in the Miscellaneous Control Register), and the I flag in the CCR is clear, then a Keyboard Interrupt will occur on the negative transition (1 to 0) of the bit.



Toggle Port A Bit 1 - Keyboard Interrupt (SR3 controller only)

This command is used to toggle Bit 1 of Port A if the corresponding bit in the Keyboard Interrupt Mask Register (KBIM) is set. If the Keyboard Interrupt is enabled (KBIE=1, bit 7 set in the Miscellaneous Control Register), and the I flag in the CCR is clear, then a Keyboard Interrupt will occur on the negative transition (1 to 0) of the bit.



Toggle Port A Bit 2 - Keyboard Interrupt (SR3 controller only)

This command is used to toggle Bit 2 of Port A if the corresponding bit in the Keyboard Interrupt Mask Register (KBIM) is set. If the Keyboard Interrupt is enabled (KBIE=1, bit 7 set in the Miscellaneous Control Register), and the I flag in the CCR is clear, then a Keyboard Interrupt will occur on the negative transition (1 to 0) of the bit.



Toggle Port A Bit 3 - Keyboard Interrupt (SR3 controller only)

This command is used to toggle Bit 3 of Port A if the corresponding bit in the Keyboard Interrupt Mask Register (KBIM) is set. If the Keyboard Interrupt is enabled (KBIE=1, bit 7 set in the Miscellaneous Control Register), and the I flag in the CCR is clear, then a Keyboard Interrupt will occur on the negative transition (1 to 0) of the bit.



Toggle Port A Bit 4 - Keyboard Interrupt (SR3 controller only)

This command is used to toggle Bit 4 of Port A if the corresponding bit in the Keyboard Interrupt Mask Register (KBIM) is set. If the Keyboard Interrupt is enabled (KBIE=1, bit 7 set in the Miscellaneous Control Register), and the I flag in the CCR is clear, then a Keyboard Interrupt will occur on the negative transition (1 to 0) of the bit.



Toggle Port A Bit 5 - Keyboard Interrupt (SR3 controller only)

This command is used to toggle Bit 0 of Port A if the corresponding bit in the Keyboard Interrupt Mask Register (KBIM) is set. If the Keyboard Interrupt is enabled (KBIE=1, bit 7 set in the Miscellaneous Control Register), and the I flag in the CCR is clear, then a Keyboard Interrupt will occur on the negative transition (1 to 0) of the bit.



Toggle Port A Bit 6 - Keyboard Interrupt (SR3 controller only)

This command is used to toggle Bit 6 of Port A if the corresponding bit in the Keyboard Interrupt Mask Register (KBIM) is set. If the Keyboard Interrupt is enabled (KBIE=1, bit 7 set in the Miscellaneous Control Register), and the I flag in the CCR is clear, then a Keyboard Interrupt will occur on the negative transition (1 to 0) of the bit.



Toggle Port A Bit 7 - Keyboard Interrupt (SR3 controller only)

This command is used to toggle Bit 7 of Port A if the corresponding bit in the Keyboard Interrupt Mask Register (KBIM) is set. If the Keyboard Interrupt is enabled (KBIE=1, bit 7 set in the Miscellaneous Control Register), and the I flag in the CCR is clear, then a Keyboard Interrupt will occur on the negative transition (1 to 0) of the bit.



Command Summary

This command is identical to the command summary function, .



Randomise Controller

This command is used to scramble the RAM, X register and Accumulator of the micro controller to simulate a power on, undefined state .

The RAM locations that has been randomised to simulate an undefined state are displayed in Green. As each RAM location is written, it is displayed in WHITE to indicate that the location has been initialised.



Jump 50 lines forward

This command is used to move the source code display 50 lines forward. The source code display must be enabled () to be effective.



Jump 50 lines backward

This command is used to move the source code display 50 lines backward. The source code display must be enabled () to be effective.

3.7. Entering Values into the Simulator

When entering a value into the simulator, the input value can be entered in either HEXadecimal (default) DECimal or ASCII.

To enter a value in hexadecimal simply type any 2 character hexadecimal value at the input prompt.

To enter a value in DECimal use the extension ',d' after entering the value (ie 255,d )

To enter a value in ASCII, at the prompt use the  key then select the ASCII key equivalent that you require (ie ? ,1 - will enter the equivalent ASCII value of \$31).

The commands recognised by the simulator at the input prompt (?) when entering a value include:-

(value),d	- decimal input
(value),h	- hexadecimal input (default)
(value),s	- stop simulator
(value),a	- enable auto load
.(ascii key)	- use ascii character value

Note that if the simulator requires an input and only the  key is pressed the default value loaded is 0.

If the simulator is 'running' and a value is required from the keyboard for an I/O operation, the step command () can be selected whilst entering the input to switch the simulator into step mode. This is useful when a program is stuck in a tight input loop and constantly requests inputs from the keyboard with no opportunity to issue additional commands.

Stopping the simulator can also be achieved by pressing the  key when a value is required to be entered from the keyboard.

The enable auto load switch (,a) bypasses the keyboard input next time the register is accessed. It is a short cut to the auto enable commands.

4. 68PROG05: EPROM Programmer

To run the Oztechnics micro controller EPROM programmer type,

`68PROG05 [user_filename.s19 or user_filename.bin [micro controller type]]` 

or select the Programmer option from the ITEE System (=) menu.

A title screen will appear which displays the simulator version and user registration details if no .S19 file and micro controller type are specified.

Both the .S19 file and controller type must be specified to enter the programming menu.

The supported micro controller families are listed in *Table 10* below.

MICRO CONTROLLER	COMMAND LINE ARGUMENT	HARDWARE REQUIREMENTS
68HC805C4	8C4	On board support
68HC705C4	C4	On board support* - use C8
68HC705C8	C8	On board support
68HC705C8A	C8A	On board support
68HC705J1A	J1A	On board support*
68HC705J2	J2	On board support
68HC705K1	K1	On board support
68HC705C9/C9A	C9	On board support
68HC705D9/D9A	D9	On board support
68HC705P6/P6A	P6	On board support
68HC705P9/P9A	P9	On board support
68HC705B5	B5	Requires external adaptor board
68HC705B16	B16	Requires external adaptor board
68HC705SR3	SR3	On board support
68705P3	P3	Requires external adaptor board
68705U3	U3	Requires external adaptor board
68705R3	R3	Requires external adaptor board

Table 10 - EPROM programmer board supported controllers

*Currently not implemented in software

This program can also be used to convert from Motorola .S19 file formats to EPROM binary images, .BIN and visa versa.

After loading a .S19 file, a binary file which represents the .S19 file is written to the disc with the filename *68IMAGE.BIN*.

If loading a raw binary image of a controller EPROM, an equivalent .S19 file is created with the filename, *68IMAGE.S19*.

Once the program is run a main menu will appear as in *Figure 23*. The menu options available change according to the type of micro controller selected.



Figure 23 - 68PROG05 Main Menu configured for the 68HC705C8 micro controller

4.1. Option 1 - Configure Programmer

This option is used to test and configure the programmer board and communications ports.



Figure 24 - Screen capture of Configure Programmer Options

4.1.1. Option 1 - Select LPT1 or LPT2

Use this option if you want to change the default printer port setting and address from the default of LPT1 (378) to LPT1 (3BC) or LPT2 (278).

4.1.2. Option 2 - Select COM1 - COM4

This option is used to select the COM port the programmer is connected to. The default port is COM1.

4.1.3 Option 3 - Select programmer board hardware version.

4.1.4. Option X - Exit to main menu

This option will return to the main menu. The  key can also be used.

4.2. Option 2 - Display Hex Dump of PROM Image

This option will display a memory dump of the program to be burnt into PROM. It also displays information on the number of PROM bytes used by the .S19 program, number of PROM bytes unused and percentage of used PROM. To scroll through the memory pages press any key or use the         keys.

The screen dump is broken up into various coloured zones as with the simulator. These coloured zones correspond to different functions of the 68HC05 memory map.

RED	- I/O Ports and Registers
WHITE	- User RAM
YELLOW	- User EPROM
PURPLE	- Mask Option Register EPROM byte
BLUE	- Boot strap ROM
GREEN	- EPROM Vector Table
BLACK	- Unused

The Display Command () can be used to toggle the display format between HEXadecimal, DECimal and ASCII.

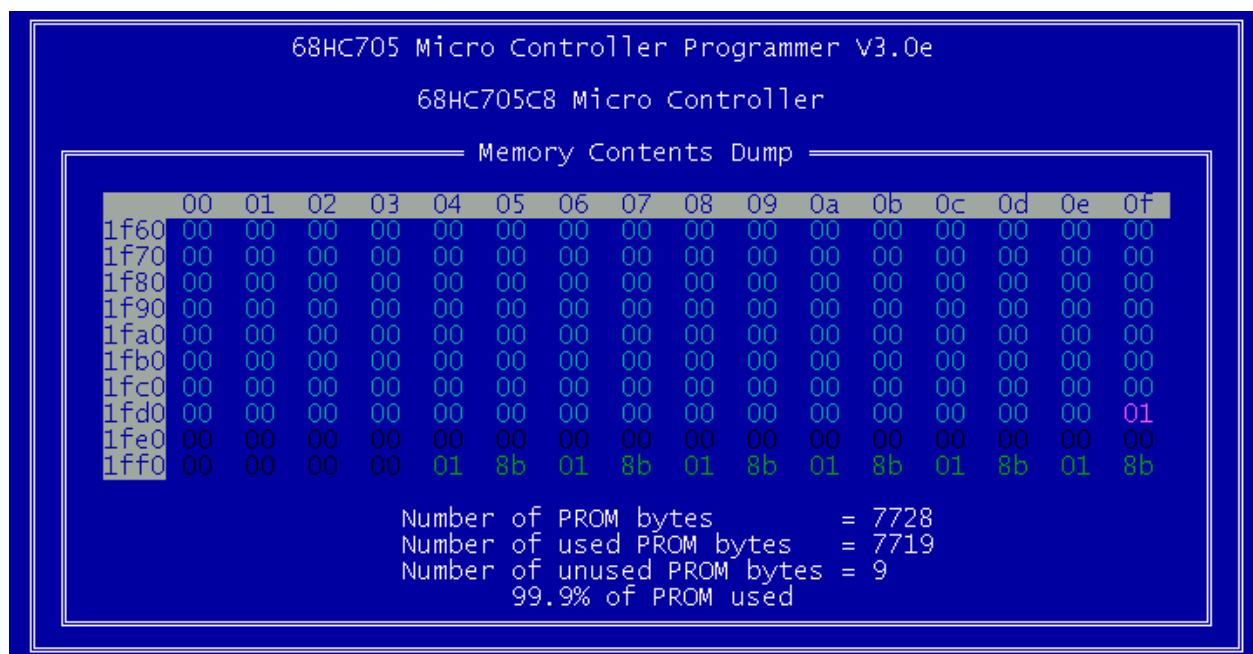


Figure 25 - Screen capture of PROM Memory Contents Dump

4.3. Option 3 - Program and Verify PROM

This option will program the 68HC05 PROM. A time bar and estimated time to program the controller is displayed on the screen. Note that before burning a controller it must first be erased.

4.4. Option 4 - Verify PROM Contents vs .S19 File

This option compares the contents of a 68HC05 PROM to the .S19 file loaded.

4.5. Option 5 - Verify Blank PROM

This option tests whether a 68HC05 PROM is erased.. All memory locations of an erased PROM (68HC705) contain the value \$00 and EEPROM (68HC805) contain \$FF. Note the 68HC705SR3 erased state is \$ff.

4.6. Option 6 - C8 Controller Options



Figure 26 - Screen capture of C8 Controller Options

4.6.1. Option 1 - Secure PROM and Verify

In this mode the C8 controller first verifies its internal PROM versus the .S19 file and if successful blows the security bit so that the internal PROM cannot be read.

4.6.2. Option 2 - Secure PROM and DUMP (SCI PORT)

In this mode the C8 controller blows the security bit so that the internal PROM cannot be read and then attempts to read its internal PROM to test the blown security bit. No read should take place. Press the  key to exit this routine.

4.6.3. Option 3 - Load Program into RAM / Execute / Monitor

In this mode the .S19 file is downloaded into the RAM of the C8 controller and then executed. The PC then enters monitor mode. If the micro controller has the monitor program (68c8mon.asm) in PROM and is enabled, then the C8 controller can be interrupted from the PC keyboard.

4.6.4. Option 4 - Execute Program in PROM (Monitor Mode)

In this mode, the C8 controller is powered up and taken out of RESET as with Option 3. Monitor mode is entered and the PC monitors the serial port for data coming back from the C8 controller. At any time the program running in the C8 controller can be Broken into by a serial interrupt request which is sent from the PC keyboard using the Break command. Once the C8 controller is in monitor mode, all of the I/O, RAM and registers are sent back to the PC. To exit monitor mode use the Go command, .

4.6.5. Option 5 - Dump PROM contents (SCI port)

This mode will read the PROM contents of the C8 controller provided the security bit has not been blown. The data dump is stored in the file *68C8DUMP.S19*.

4.6.6. Option X - Exit to Main Menu

This option is used to return to the main menu. The  key can also be used.

4.7. Option X - Exit

This option is used to return to DOS. The  key can also be used.

4.8. Monitor Mode / Real Time ICE

The Monitor mode for *68PROG05* is used with the C8 controller and the interrupt source routine found in the file *68C8MON.ASM* to perform In Circuit Emulation. During this mode the  command can be used to BREAK into the C8 controller and read back the status of the program under test. The contents of the RAM and registers are then displayed on the screen as displayed in *Figure 26*.

To perform real time In Circuit Emulation add *68C8MON.ASM* to your program under development or build your program around this file. Insert the Software Interrupt Instruction (SWI) into your program where you require the controller to break, assemble the program and burn into a C8 controller. Make sure that the SCI interrupt is enabled.

When a SWI instruction is executed by the C8 controller or a byte is received from the PC via the serial port, the monitor interrupt routine is executed. Once the C8 controller is in monitor mode all of the contents of the RAM and registers are transmitted back to the PC for display. The  command takes the C8 controller out of the interrupt routine (RTI) and continues with execution of the program under test. To quit monitor mode use the  command or the  key.

In addition to the monitor Software Interrupt routine, *68C8MON.ASM* contains equates for the C8 controller and a demonstration routine which changes the values of the I/O ports and RAM.

Remember that in monitor mode the micro controller has executed an interrupt routine and hence the stack contains the values of the CCR, Accumulator, X Register, Program Counter before the interrupt monitor routine was executed.

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
[0000]	FF	FF	FF	80	FF	FF	FF	CC	00	CC	04	00	00	30	00	2C	
[0010]	00	00	00	20	02	02	96	1E	AD	E2	BA	12	00	CC	00	00	
[0020]	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
[0030]	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
[0040]	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
[0050]	11	22	E4	04	00	00	06	57	00	59	5A	5B	5C	5D	5E	5F	
[0060]	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	
[0070]	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F	
[0080]	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	
[0090]	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	
[00A0]	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	
[00B0]	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF	
[00C0]	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	
[00D0]	D0	D1	D2	D3	D4	D5	D6	D7	D8	DA	DB	DC	DD	DE	DF		
[00E0]	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	
[00F0]	F0	F1	F2	F3	F4	F5	F6	02	78	02	06	E8	11	22	01	87	
[0100]	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
[0110]	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
[0120]	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
[0130]	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
[0140]	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
[0150]	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

CCR	HINZC
\$E4:	11100100
Accumulator	
\$11:	00010001
Index Register	
\$22:	00100010
PCH	
PCL	
Program Halted	
(G)o / Continue	
(Q)uit Monitor	
Command ?	

Figure 27 - Screen capture of Monitor Mode

4.9. Error Messages

68PROG05 will report two types of error messages when a micro controller error is detected.

The most commonly reported error message is 'Micro Controller Not Blank'. This error message usually indicates that the 68HC05 PROM is not erased (that is all internal bits are not zero) prior to burning the PROM.

Another possible cause of this error message occurs when there is a problem between the interface of the 'verified' line and the PC such as a faulty buffer circuit, cable or printer port.

The second error message 'Micro Controller Not Responding' indicates that the 'programmed' line of the micro controller did not toggle low. This indicates that boot-strap mode has not been initialised. Possible causes include:-

- Micro controller not sitting correctly in ZIF socket;
- Micro controller not entering boot strap mode;
- Problem with interface between micro controller 'programmed' line and PC;
- Faulty printer port; or
- Faulty micro controller.

If the micro controller is not entering the boot strap mode, check all voltages, the reset line and the crystal. Refer to the diagnostics chapter for fault finding the programmer board.

5.0 The Programmer Board

The programmer board requires a connection to the PC parallel port for normal operation. The parallel cable is of a 1-1 straight through type.

The programmer board requires a 9-12Vdc @ 250mA power source connected to the 3.5mm power jack.

The serial port connection is optional and is only required for In Circuit Simulation (ICS) and reading the contents of a C8 micro controller.

The programmer board contains a DC-DC switcher for Vpp (programming voltage) generation, RAM, address decoding, data latches, control circuitry and 8 programming sockets.

At the bottom left corner of the board is a Ground connection and 4 voltage monitoring points. The board is labeled with these monitoring points and are described below,

- The '+5' point is used to measure the permanent +5Vdc rail.
- The 'Vpp' point is used to measure the programming voltage which is adjustable using VRES1-VRES5.
- The 'SW' point is used to measure the switched +5Vdc rail.
- The 'Boot' point is used to measure the boot strap voltage required for boot strap mode.

The Vpp voltages are factory set. Table 11 summarises this information.

Voltage	Trim Pot	Vpp	Controllers
1	VRES1	13.5Vdc	SR3, SU3, SP3
2	VRES2	15.5Vdc	None
3	VRES3	16.5Vdc	C8, C8A, C9, C9A, D9, D9A, J2, J1A, P6, P6A, P9, P9A, B5, B16
4	VRES4	17.5Vdc	K1
5	VRES5	19.75Vdc	68HC805C4
6	Fixed	21Vdc	68705P3, U3, R3

Table 11 - Programming Voltages (Vpp)

ZIF1, is a 40 way DIL socket used for programming the 68HC705C4, C4A, C8, C8A and 68HC805C4 micro controllers. It is also used to run the monitor program (68ICSC8.S19) in a C8/C8A controller for performing ICS and diagnostics. A 44 pin ZIF PLCC adaptor board is also plugged into this DIL socket for PLCC package support.

ZIF2, is a 40 way DIL socket used for programming the 68HC705C9, C9A, D9 and D9A controllers. A 44 pin ZIF PLCC adaptor board is also plugged into this DIL socket for PLCC package support. The B5-B16-B32 programming adaptor board also plugs into this socket. This adaptor board consists of a 52 way PLCC ZIF socket and support circuitry.

The **ICS** socket is used as a header connection to connect the C8/A controller in the ZIF1 socket to an external circuit for emulation. A monitor program running in a C8/A is required in ZIF1. Various adaptor boards are available for converting the C8 pin out foot print to other micro controllers such as the P6/P9, R3/SR3/U3/SU3, P3/SP3, J1A/J2, K1 etc.

ZIF3, is a 40 way DIL socket used for programming the 68HC705SR3 and SU3 controllers. An adaptor board plugs into this socket for programming the 68HC705SP3 controller. An adaptor board plugs into this socket for programming the HMOS 68705P3, U3 and R3 controllers.

ZIF4, is a 28 way DIL socket used for programming the 68HC705P6, P6A, P9 and P9A controllers. An adaptor board plugs into this socket for SOIC package support. An adaptor board plugs into this socket for 68HC705P18 SOIC package support. An adaptor board plugs into this socket for 68HC705MC4 SOIC package programming support.

ZIF5, is a 28 way DIL socket wired for 20 way DIL operation used for programming the 68HC705J2 controller. This socket is upside down compared to the other programming sockets.

ZIF6, is a 28 way DIL socket wired for 20 way DIL operation for the 68HC705J1A controller.

ZIF7, is a 28 way DIL socket wired for 16 way DIL operation for the 68HC705K1 controller. This socket is upside down compared to the other programming sockets.

The Aries 40 way DIL ZIF socket is moved between the various programming sockets as required. As this ZIF socket handles .3 and .6 inch width chips all DIL footprints are covered with this one socket.

In the case of the 28, 20 and 16 pin packages ensure that the controller is placed at the TOP of the socket.

Remember that the TOP of the socket for the J2 and K1 is at the bottom of the board. This is marked on the board.

The adaptor boards plug directly into the on board sockets. They do not plug into the ZIF socket.

6. Programmer Board Diagnostics

The *68diag05.exe* utility is used to verify correct programmer board, PC communication ports and controller operation.

The Diagnostic program can be run from the ITEE System Menu or from the DOS prompt.

Please note that this diagnostics program is only suitable for V3.x programmer boards.

The diagnostics program performs many tests on the programmer board to verify correct operation. These tests include checking correct operation of data latches, Vpp generator, RS232 translator, RAM, address decoder, crystal, status lines and the RESET circuitry.

A special monitor program (*68icsc8.s19*) is required in a C8/A controller to perform all of the above tests. Note some of the tests can be performed without the controller. The software will prompt you when it requires the monitor program inserted into the ZIF1 programming socket.

To perform a complete check of the programmer board you will need to connect the programmer board to the parallel and serial ports of the PC and configure the ports using the configure board option,  if not already configured.

Whenever the programming software, diagnostic software or hardware configuration menus are selected the programmer board LED indicators, ZIF Power, Boot-strap and RUN are momentary flashed in succession to indicate correct I/O port and programmer board operation.

The Vpp, Boot Mode Voltage, Gnd and two 5 Volt rails can be measured from the test points on the lower left hand side of the programmer board.

The best way to check the programmer board operation is to select the Auto Test option,  which steps through the nine tests available and presents a results summary at the end of the testing.

Individual Tests (1-9) can be selected from the Main Menu if required.

The *Programmer Board Diagnostics Menu* is used for trouble shooting the address and latch decoders on the programmer board. A CRO or Logic Probe is required to measure the logic levels on these components.

The *Controller Diagnostics Menu* is used to Test the correct operation of a C8/A controller. The first test verifies the internal RAM is functioning and the second test outputs data on the I/O ports of the controller. Also the serial port is tested and the results are continually displayed on the PC (Test message 'the quick brown fox jumps over the lazy dog' is sent from the PC and received back from the controller). The *68icsc8.s19* file must first be burnt into the controller before any of these tests can be performed. It is always handy to have one C8/A micro with this program ready to go.

7. 68DIS05: Disassembler

This utility program takes a .S19 file and disassembles it creating three new output files.

The full parameters of this command are:-

68DIS05 [.s19 file name [micro controller type]]

:- .S19 file to be disassembled

:- [micro controller type] P3, U3, R3, J1A, J2, K1, C8, C9, D9, E1, P6, P9, B5, B16 and SR3.

For example *68SIM05 DEMO1 C8* will load the *DEMO1.S19* file and specify the C8 micro controller memory map for disassembly. Note that the .S19 file extension is not necessary.

The three output file extensions are explained below,

.DIS -->	This is an ASCII text file containing the disassembled .S19 file.
.TXT-->	This is an ASCII text memory dump file of the .S19 file.
.BIN -->	This is a binary file representation of the memory map of the .S19 file.

8. Tutorial

This tutorial describes the procedure to burn a C8/A controller with the 68icsc8.s19 monitor and diagnostics file. A brief run down on the editor, programming, diagnostics and simulator software is also described.

1. Connect power to the programmer board. The POWER LED should light. Do NOT have any controllers in the ZIF sockets at this stage.
2. Connect the programmer board to the PC using the parallel and serial cables. Identify which ports (LPT1, LPT2, COM1, COM2) the programmer board is connected too.
3. Run *68ed.exe* and OK out of the start up information window.
4. Load the *68icsc8.asm* file. This file can be found in the 68HC05\ICS directory.  will bring up the File menu or  will highlight the menu bar and using the cursor keys and the  key select the File\Open option. The FILE menu can be selected with the mouse by left mouse clicking on the File Menu bar. Note that the last file opened is automatically loaded on startup.
5. Now select the micro controller type by opening the Options Menu () and selecting the 68HC705C8 micro controller.
6. Once loaded, for practice, run the assembler,  or  or use the mouse to select the Assemble option. The editor will then shell to the cross-assembler program where the file can be assembled using the Begin option (). Once assembled the editor is re-loaded and a report window is opened at the bottom of the screen displaying the results of the cross assembler. 0 Errors should be reported.
7. Now run the programmer by selecting the  key or using the mouse to select the System menu (=) then the programmer option.
8. Configure the PC ports using Option . Once the correct parallel port is selected the ZIF, Boot Strap and RUN LEDS will flash in succession. The ports only need to be configured once. The port values are saved in a configuration file (*oztech.cfg*).
9. Place an erased C8/A controller in the ZIF1 socket and perform a BLANK check, Option .
10. Perform a Verify Test, Option . This should FAIL as the controller is blank. This step is not normally necessary but is a good indication that the board and controller are working correctly.
11. Now burn the controller using option . After 16 seconds the controller should pass.
12. Exit the programmer using the Exit Option,  key.
13. Run the Diagnostics Program from the System menu.
14. Open the ZIF socket and select Auto Test, Option . Follow through the nine tests as prompted by the diagnostic software. To measure the voltages on the board you will need a meter.
15. Exit the diagnostic software using the  key. We have now established correct programmer board and PC operation.
16. Load the file *tutorial.asm* from the 68HC05\FILES directory and assemble as in Step 5.
17. Open the Options Menu and select the 68HC705C8 micro controller.
18. Move the cursor to line 436 and insert an error into the code by placing an X in front of the LDA instruction.   to move the cursor to the start of the file.

-
19. Run the cross assembler again as in Step 5. This time an error will be reported at line 436. Use the **F6** key to focus on the bottom window and move the cursor over the error using the keyboard arrow keys if required. Press the **Enter** key and the editor will automatically move the cursor to the error in the source code. Press the **F6** key to focus on the editor to correct the error or use the mouse. Assemble again as in step 5 but this time produce a List File (**P**). We have now practiced editing errors out of the source code using the auto line position feature.
 20. Run the simulator, **F4** or select it from the systems menu using the mouse.
 21. Place the programmed controller (68/csc8) into ZIF1. Enter the **Z** command to initiate In Circuit Simulation. If successful ICS will be displayed on the screen in the ports area.
 22. Select the Go command, **G** to run the simulator. Note that the port values on screen are the same as on the ICS socket. Use a CRO or logic probe to monitor this.
 23. Experiment with placing different logic levels on PORTD pins and monitoring the effects in the simulator when the IRQ pin is exercised (**I**). An IRQ is generated on the negative transition of this pin. You will note that the logic levels on PORTD are reflected on the screen.
 24. Set a breakpoint at location \$01A7 using the **B** key. Now the simulator will stop when the Program Counter hits this address. You can experiment with setting different breakpoints on addresses, X Register or Accumulator values. Try setting a breakpoint on the SCI Interrupt routine at location \$01e9. Set a X register break point at a value of \$70. Use the Step command, **S** to trace through the IRQ code.
 25. Use the **A** key to view the source code. The cursor keys are used to move the display.
 26. Use the **F1** or **C** keys to display a command summary of the simulator.
 27. Disable the ICS mode by selecting the **Z** key. Note that a value will have to be entered via the keyboard on an I/O port read now.
 28. Use the Format command, **F** to change the format of the screen values. You will notice the ASCII text message in the RAM area.
 29. Use the **E** key to toggle the Timer Input Capture Pin. On the negative edge the IRQ routine is triggered and the IRQ service routine is serviced.
 30. Experiment with the all the commands in the command summary.
 31. End of tutorial

Appendix I - 68SIM05 Quick Reference Guide

KEY	COMMAND
	Source code display (.LST FILE)
	Set Breakpoint [1-3]
	Command Summary
	Display Data Format
	Go
	Step over subroutine
	Halt
	Toggle External Interrupt Pin
	Reset Cycle Counting
	Reset Interrupts Cycle Counting
	Load a .S19 file
	Modify Memory (RAM & Registers only)
	Clear all Breakpoints
	Quit Simulator
	Modify Registers
	Step One Instruction
	Step Backwards One Instruction (Undo)
	Toggle Display Mode
	Log File (save instruction trace to disc)
	View Registers and Vectors
	Configure Communications Parameters
	Reset Micro Controller
	Simulation Speed
	In Circuit Simulation (ICS)
	Auto Load Port A
	Auto Load Port B
	Set Accumulator Break Value
	Set X Register Break Value
	Modify Accumulator
	Modify CCR Register
	Go to Line Number in source code
	Modify Program Counter
	Modify X Register
	Command Summary
	Randomise Controller
	Jump forward 50 lines of source code
	Jump backward 50 lines of source code

Table 12 - 68SIM05 Quick Reference Guide

68HC705C8,C8A,C9 Specific Commands

	Toggle TCAP Input Pin
	SCI - Set RDRF
	Auto Load Port C
	Auto Load Port D
	Auto Load SCI Data Register
	Auto Load SPI Data Register

68HC705D9 Specific Commands

	Toggle TCAP Input Pin
	SCI - Set RDRF
	Auto Load Port C
	Auto Load Port D
	Auto Load SCI Data Register

68HC705B5,B16 Specific Commands

	Toggle TCAP Input Pin
	SCI - Set RDRF
	Toggle TCAP2 Input Pin
	Auto Load Port C
	Auto Load Port D
	Auto Load SCI Data Register
	Auto Load A-D Converter
	Auto Load SPI Data Register

68HC705K1 Specific Commands

	K1 Controller External Interrupt: PA0
	K1 Controller External Interrupt: PA1
	K1 Controller External Interrupt: PA2
	K1 Controller External Interrupt: PA3

68HC705J1A, J2 Specific Commands

None

68HC705SR3 Specific Commands

	Toggle External Int2 Pin
	Toggle External Timer Input Pin
	Auto Load Port C
	Auto Load Port D
	Auto Load A-D Converter
 	Toggle Port A Bit 0 - Keyboard Interrupt
 	Toggle Port A Bit 1 - Keyboard Interrupt
 	Toggle Port A Bit 2 - Keyboard Interrupt
 	Toggle Port A Bit 3 - Keyboard Interrupt
 	Toggle Port A Bit 4 - Keyboard Interrupt
 	Toggle Port A Bit 5 - Keyboard Interrupt
 	Toggle Port A Bit 6 - Keyboard Interrupt
 	Toggle Port A Bit 7 - Keyboard Interrupt

68705R3 Specific Commands

	Auto Load Port C
	Auto Load Port D
	Auto Load A-D Converter
	Toggle external INT2 pin
	Toggle External Timer Input Pin

68705U3 Specific Commands

	Toggle external INT2 pin
	Toggle External Timer Input Pin
	Auto Load Port C
	Auto Load Port D

68705P3 Specific Commands

	toggle External Timer input pin
	Auto Load Port C

68HC705P6, P9 Specific Commands

	Toggle external timer input pin
	Auto Load Port C
	Auto Load Port D
	Auto Load A-D Converter

68HC705E1 Specific Commands

	Auto Load Port C
---	------------------

Appendix II - Supported Motorola Micro Controllers

FAMILY LIST OF FEATURES	MC68HC705C8/A	MC68HC805C4	MC68HC705J1A/J2	MC68HC705K1	MC68HC705SR3
Technology	HCMOS	HCMOS	HCMOS	HCMOS	HCMOS
Number of Pins (DIL)	40	40	20	16	40
On Chip RAM (bytes)	176-304	176	64/112	32	128
On Chip EPROM (bytes)	7600-7744	4156 EEPROM	1240/2064	504	3840
Personality EPROM	0	0	0	64 (bits)	0
Bi directional Lines	24	24	14	10	32
Uni directional lines	8	8	0	0	0
Other I/O features	serial communication interface (SCI), serial peripheral interface (SPI), 16 bit multi functional timer with capture/compare	serial communication interface (SCI), serial peripheral interface (SPI), 16 bit multi functional timer with capture/compare	15 bit multi functional timer with real time interrupt circuit	15 bit multi functional timer with real timer interrupt circuit	4 channel A/D converter, 8 bit programmable counter driven by a 7 bit programmable pre scaler
External Interrupt Inputs	1	1	1	5	2
Power Saving Stop, Wait & Data Retention Modes	yes	yes	yes	yes	yes
Computer Operating Properly (COP)	yes	yes	yes	yes	no
Software programmable interrupt sensitivity	yes	yes	yes	yes	no
EPROM Secure mode	yes	no	no	no	no

FAMILY LIST OF FEATURES	MC68HC705B5/B16	MC68HC705C9/A	MC68HC705D9/A	MC68HC705E1	MC68HC705P6/P9
Technology	HCMOS	HCMOS	HCMOS	HCMOS	HCMOS
Number of Pins (DIL)	56	40	40	28	28
On Chip RAM (bytes)	176/352	352	352	368	176/128
On Chip EPROM (bytes)	6208/15104	15744	15744	4112	4672/2112
Personality EPROM	0	0	0	0	0
Bi directional Lines	24	31	31	20	20
Uni directional lines	8	0	0	0	1
Other I/O features	8 channel A/D, SCI, 16 bit multi functional timer with real time interrupt, input capture and compare, Pulse Length D/A converter (PLM)	16 bit multi functional timer with real time interrupt, input capture and compare, SPI, SCI	16 bit multi functional timer with real time interrupt, input capture and compare, Pulse Width Modulation channels high current led drive output	Phase Locked Loop Frequency Synthesiser, 15 stage Multi function timer	16 bit multi functional timer with real time interrupt, input capture and compare
External Interrupt Inputs	1	1	1	1	1
Power Saving Stop, Wait & Data Retention Modes	yes	yes	yes	yes	yes
Computer Operating Properly (COP)	yes	yes	yes	yes	yes
Software programmable interrupt sensitivity	yes	yes	yes	yes	yes
EPROM Secure mode	yes	yes	yes	no	no

Table 13 - Supported Micro Controllers

Appendix III-V not included