TECICE- HC11

USER'S MANUAL

68HC11 In Circuit Emulator

TECi

The Engineers Collaborative, Inc.
Website www.tec-i.com
E-Mail info@tec-i.com

* * * IMPORTANT WARRANTY AND LIABILITY INFORMATION * * *

This product is guaranteed against defects in material or workmanship for a period of 180 days from the date of shipment. If it should become necessary to return a product for service or repair within the warranty period, contact the factory first for return authorization. The returned item will be repaired or replaced. Except as indicated above, there are no other warranties, expressed or implied. No warranty of fitness for a particular purpose is offered. The user is advised to test the product and software thoroughly before relying on it. The user assumes the entire risk of using the product. The total liability of The Engineers Collaborative, Inc. is limited to the purchase price of the product, and does not cover any lost profits, special, incidental or consequential damages, or any claim against the purchaser by any party.

* * * SOFTWARE LICENSE STATEMENT * * *

The software supplied with this product is protected by US copyright law and international treaty provisions. Therefore, you must treat this software "just like a book", with the following exception. The Engineers Collaborative, Inc. authorizes you to make archival copies of the software for the sole purpose of backing-up our software and protecting your investment from loss. By saying, "just like a book", The Engineers Collaborative, Inc. means that the software can not be used by two different people in two different places at the same time.

* * * OTHER * * *

The information contained in this manual has been carefully checked and is believed to be accurate and complete at the time of printing. However, no responsibility is assumed for errors that might appear. The Engineers Collaborative, Inc. reserves the right to make changes to the product and/or the manual at any time without notice. Furthermore, The Engineers Collaborative, Inc. assumes no liability arising out of the use or application of any of its products. No part of this document may be copied or reproduced in any form or by any means without prior written consent of The Engineers Collaborative, Inc.

TECICE-HC11 USER'S MANUAL

TABLE OF CONTENTS

1. **INTRODUCTION**

Thank you for choosing the TECICE-HC11 emulator development tool. We think that as you become familiar with the product you will feel that you have made a sound investment, one that will pay for itself over and over again on a wide variety of development projects. Please keep in mind that we welcome your feedback regarding your experiences using the product and your suggestions for improvements. Please keep in touch and let us know what your doing with the TECICE-HC11.

1.1 IMPORTANT NOTICES

?? Make sure that you have read the warranty, liability and software license statements on page 2 of this manual. Any use of this product constitutes acceptance of the terms and conditions of those statements. If you do not agree with these terms and conditions return the product at once. The purchase price will be refunded.

?? *FOLLOW THE INSTRUCTIONS IN SECTION 2. GETTING STARTED*. This will take only a few minutes, but it will save you a <u>tremendous</u> amount of time and trouble later on.

?? *USE A COMMON POWER STRIP FOR BOTH THE TECICE AND YOUR TARGET SYSTEM.*

<u>Care must be taken to insure that both the TECICE and your target system are powered on/off together to prevent CMOS latch up, a condition that causes a CMOS chip to draw a large amount of current and possibly damage the chip or other chips connected to it.</u> CMOS latch up occurs when a CMOS chip gets an input voltage that exceeds the power pin voltage of the chip. When either the TECICE or your target system is powered on and the other is off, <u>CMOS latch up can occur.</u>

**THIS CAN BURN OUT CHIPS INSIDE
THE EMULATOR OR ON YOUR TARGET SYSTEM!!!**

Follow this procedure to get the best results. Use a common power strip for both the TECICE and your target system. Turn the power on/off using the switch on the power strip so as to turn everything on/off at once.

IF CMOS LATCH UP OCCURS, TURN ALL POWER OFF IMMEDIATELY!!!

?? The TECICE-HC11 monitor program has no provisions to update the COP watchdog timer. Therefore, if you enable the watchdog timer, the system will reset and lock up. TO GET OUT OF THIS LOCK UP, PRESS THE **MASTER RESET.**

1.2 TECI BUG POLICY

TECI will <u>NOT</u> ship a product that has a known bug that prevents the product from performing its intended function! However, sometimes there are bugs in a product that may be a nuisance to the user, but do not prevent the product from performing its intended function. Furthermore, the risk of introducing new, more serious and unknown bugs into the product by performing a <u>quick fix</u> may not justify an immediate and untested fix. TECI will sometimes ship a product that has these so called "nuisance bugs" if we feel that an acceptable workaround exists and we feel that the benefit to the user far outweighs the nuisance.

TECICE-HC11 KNOWN BUG LIST

1) DISPLAY MEMORY COMMAND

The DISPLAY MEMORY command sometimes fails to stop at the correct place.

For example, the command: **>DM FF0 FFFF<Enter>** will display all 64K of memory.

WORK AROUND

A) Press the emulator "MASTER RESET" push button when the problem occurs.
B) Give the command **>DM FFF0 0000<Enter>** instead.

2) WHEN POWER IS FIRST APPLIED TO THE TECICE-HC11 WHEN THE EMULATOR IS SET UP FOR EXPANDED MULTIPLEXED MODE

When power is first applied to the TEICICE-HC11 emulator, it performs a self test of memory. This also happens when the TST (self TeST) command is performed. If the emulator is set up for expanded multiplexed mode with some memory to be present externally on the target system, the memory test may fail. When this happens, you will get the message after the power-up: "Checking Emulator Memory.. Error - Emulator Memory failed test." This error message is most likely to occur when you have memory mapped I/O devices on your target system.

WORK AROUND

A) Ignore the error message if the emulator is set up for expanded multiplexed mode.
B) If the external memory is ram memory, make sure that the emulation probe is plugged into the target system and that the target system powers up with the emulator. Target system ram is properly tested by the memory test function. The problem occurs with write only or read only memory locations.
C) To test the emulator's internal memory, set up the emulator for single ship mode, J2 A&B in the S/C position. You should not get the error message.

1.3 GENERAL DESCRIPTION

TECICE-HC11 is a real time in circuit emulator designed to emulate most members of the Motorola 68HC11 family of single chip microcontrollers. TECICE-HC11 is an acronym made up from the letters "TECI", "ICE" and "HC11". TECI stands for The Engineers Collaborative, Inc., ICE stands for in circuit emulator and HC11 stands for the Motorola MC68HC11 family of microcontrollers.

TECICE-HC11 can be used with any host computer running terminal emulation software with the ability to download files over a directly connected RS-232 interface at 9600 baud. TECICE-HC11 is controlled by standard ASCII commands and could be used with just a dumb terminal if it weren't for the fact that some means of downloading object code files generated from compilers/assemblers is required by real applications.

The emulator can be configured by the user to emulate most members of the 68HC11 family of microcontrollers. To change the configuration of the emulator the user must install the proper emulation processor and emulation probe, and choose the proper internal jumper options for his/her application.

For those users that will be using the emulator with IBM PCs or compatibles, a cross assembler program and a special terminal emulation program is provided. The cross assembler program is called WASM11.EXE and the terminal emulation program is called WTT11.EXE. These programs are fully described in the "WASM11 & WTT11 Manual". WASM11 is a Windows hosted Text Editor / 68HC11 Macro Cross Assembler. WTT11 is a Windows hosted Emulator Control Program with Source Level Debugging. Both of these programs are part of an integrated set of software and hardware tools provided by TECI to facilitate the development of 68HC11 single chip microcontroller applications.

The example/demo programs assembled in the "WASM11 & WTT11" and the "GETTING STARTED" sections of this manual assume use of the WTT11 program.

1.4 TECICE-HC11 EMULATOR SPECIFICATIONS

- ?? Dimensions: 7.3" Wide X 9.5" Deep X 1.5" High
- ?? Weight: 4 pounds
- ?? Power Requirements: 9-12 Volts AC 50/60 Hz or 9-15 Volts DC at 400 milliamps.
- ?? Interface: RS-232 9600 Baud, 8 data bits, 1 start bit, 1 or 2 stop bits, no parity.

1.5 EQUIPMENT REQUIRED

?? IBM PC BASED SYSTEMS

System Requirements: At least a 486 class PC running Windows NT, Windows 95 or Windows 3.1 with at least 8MB ram, 4 MB of free hard disk space, a 3.5" floppy disk drive and one serial port configured as COM1 or COM2. An SVGA monitor is recommended. Also required is a serial port cable that connects from your COM1 or COM2 serial port and has a male DB-25 connector for mating with the TECICE-HC11. Standard modem cables will work with the TECICE-HC11.

?? OTHER HOST COMPUTER SYSTEMS

Other host computers must have an RS-232 serial port and terminal emulation software that is capable of making a direct connection to the RS-232 serial port at 9600 baud, 8 data bits, one or two stop bits, one stop bit and no parity. This software must send characters that are typed at the keyboard out the serial port and display incoming characters from the emulator on the video screen. The software must also be capable of downloading standard text object code files to the emulator. Most standard modem communication programs will work. Also required is a serial port cable that connects from the serial port and has a male DB-25 connector for mating with the TECICE-HC11. Standard modem cables will work with the TECICE-HC11. Instead of connecting the cable to a modem, it is connected to the emulator.

You may use the WASM11 Editor/Macro Cross Assembler Program or another text editor and 68HC11 cross assembler program. The text editor program is used for entering and editing programs in 68HC11 assembly language. The 68HC11 cross assembler program must create object code files in Motorola S record or Intel HEX format.

2. **GETTING STARTED**

Please read and follow the directions in this section before you go any further!

Before you do anything else, please take just a few moments to familiarize yourself with this section, the WASM11 and WTT11 software and your new emulator.   We specifically designed this section to give you a lot of information in a very brief period of time. Working through the installation and tutorial now will give you a big advantage in understanding how these programs work together to make your emulator a powerful tool!

2.1 SOFTWARE PREPARATION

2.1.1 Installing WASM11 And WTT11

The software is supplied on a 3 1/2 inch floppy diskette. The file supplied on this diskette is named SETUP.EXE. This is a self-installing, Windows executable file that will install all of the necessary files when invoked.

To install the software, place the distribution diskette in a floppy drive and run the program A:SETUP.EXE or B:SETUP.EXE from within Windows and follow the on-screen instructions.

*** Important Notice ***

If you have trouble running SETUP.EXE , try running the program with the /M command line switch as follows: >A:SETUP.EXE   /M<enter>   This is the manual installation mode which will prompt you for the location of needed directories, etc.

After running SETUP.EXE you should find the following files in the installation directory(C:\WASM11 is the default).

> WASM11.EXE - Editor/Macro Assembler executable file.
> WASM11.HLP - Editor/Macro Assembler on-line help file.
> WASM11.INI - Editor/Macro Assembler initialization file.
> WTTDEMO.11P - Demo 68HC11 project file.
> WTTDEMO.ASM - Demo 68HC11 project source code file.
> WTTDEMO.LST - Demo 68HC11isting file.
> ITEST.INC - Demo 68HC11 project include file.
> D68HC11.11P - Project file for showing WASM11 source code examples.
> D68HC11.ASM - File showing WASM11 source code examples.
> UNWISE.EXE - Uninstall executable.
> INSTALL.LOG - Installation log file - required by the uninstall program.
> WTT11.EXE - Emulator control program executable file.

WTT11.HLP - Emulator control program online help file.
WTT11.INI - Emulator control program initialization file.

*** Important Notice ***

FONT SIZE: If the text is not displayed as desired, the font size may be changed from within WASM11 by clicking Options|Font and changing the font size.

2.1.2 Uninstalling WASM11 & WTT11

To uninstall the software, double click on the UNWISE.EXE program in the installation directory. This program will read the INSTALL.LOG file and remove all files that were installed with the SETUP.EXE program.

2.2 HARDWARE PREPARATION

After unpacking the emulator, inspect the emulation probe to make sure none of the pins are bent or shorted together. Make sure that there are no loose components rattling around inside the emulator which would indicate mishandling during shipping. Then, plug the power connector from the wall mounted transformer into the back of the emulator where it is marked 12 VAC, and plug the transformer into a 120 VAC outlet. The Ready lamp on the front panel should come on. If it does not, you should open the emulator housing and make a closer visual inspection for components that may have shaken loose during shipping. The Ready lamp should go off when the Master Reset switch is pressed and come back on when the Master Reset switch is released. Connect the emulator to the host computer via an RS-232 serial cable, the same type of cable used for external modems will work.

For the example session it will not be necessary to connect the emulation probe to any target system.

2.3 WASM11 & WTT11 TUTORIAL

2.3.1 Using WASM11

First, invoke the WASM11 editor/assembler program by clicking on its chip icon. The first time you use the program, it should come up with the WTTDEMO.11P project already loaded into the editor. This is an example project for use with this tutorial. The project name and active editor file name are shown in the title bar of the main program window. If the program doesn't come up with the WTTDEMO.11P project loaded, you can load it with the File|Open Project menu item. At this time the project consists of two files, the WTTDEMO.ASM main source file and one include file named ITEST.INC . You can switch the editor back and forth between these files by clicking on the file tabs. Try It!

To see what each of the speed buttons on the button bar do, move your mouse pointer over each button and hold it there for about a second. A yellow help hint box should appear near the mouse pointer describing the button function. A little more detailed description appears on the status bar at the bottom of the screen. Once you get used to what the speed button functions are, you can turn the help hints and status line help hints off by unchecking the appropriate Option menu items. The operation of the text editor portion of WASM11 is similar to most Windows text editors; so we won't go into how to edit text here. You can use the online help for this purpose or, better yet, just experiment on your own. If you always click on the "Save Project" speed button before exiting the program, WASM11 will come up configured exactly as it was when you last used it.

WASM11 source code files use standard Motorola 6811 op-code mnemonics and source file syntax with a few industry standard enhancements. See the online help for complete documentation. Also, a project named D68HC11.11P was included in the distribution files. This is a project that shows the proper usage of all 68HC11 mnemonics, pseudo ops and source file syntax.

Now let's assemble the WTTDEMO project. To do this click the large "✍6811" speed button on the button bar. Two errors were intentionally left in the project files to show what happens when errors are encountered by the assembler. Note that an error window opened that has a list of the errors. The first error is an "unrecognized opcode mnemonic" error in line 30 of file WTTDEMO.ASM . Also note that the cursor in the WTTDEMO.ASM file was placed at the beginning of line 30, the line that contains the first error. If you click on the "GoTo Next Error" button, the editor cursor goes to the next error, which is in line 7 of the include file ITEST.INC . When errors are encountered, you can find and fix them easily by using the "GoTo Next Error" button and the information in the error window.

Now lets fix the errors and reassemble the project. Click on the "GoTo Next Error" button until the editor cursor is at the line containing the first error: line 30 of file WTTDEMO.ASM . Note that the op code mnemonic in that line is "staaa", an improper 6811 mnemonic. Obviously, the proper 6811 mnemonic is "staa"; so change "staaa" to "staa". Click on the "GoTo Next Error" button. This is another "unrecognized opcode mnemonic" error in line 7 of ITEST.INC . Here, the opcode mnemonic reads ".rts" instead of "rts". Change ".rts" to "rts" and click on the "GoTo Next Error" button. This error just informs the user that the assembler could not complete its second pass because of errors encountered during pass one. We do not have any corrections to make associated with this error.

Click the "✍ 6811" speed button to reassemble the project. Note that this time no error window appears, so no errors were encountered and the project assembled correctly. To view the files created by the assembler, click on the Options menu, then on the Project Options submenu. The Project Options dialog box opens. Check the "Auto Load Output Files" checkbox; then click OK. Click the "✍ 6811" button again and note that after assembly there are now four files loaded into the editor. The two new files were the files created by the assembler and are WTTDEMO.LST, the listing file, and WTTDEMO.S19, the object code file in Motorola S-record format. Click on their tabs to view these files. The WTTDEMO.LST file is used by the WTT11 program to

control the emulator and the WTTDEMO.S19 file is used to program 68HC11 chips when the program is debugged.

We are now ready to use the WTT11 program and the TECICE-HC11 emulator to run and test the WTTDEMO program.

2.3.2 Using WTT11

To invoke WTT11 from WASM11, click on the "D", run debugger, speed button. The "Debugger Command Line" under Options|Project Options menu in WASM11 determines which program runs when this button is pressed. If you installed these programs in the default directories, the Debugger Command Line entry should be "c:\wasm11\wtt11.exe c:\wasm11\wttdemo.lst .

*** Important Notice ***

FONT SIZE: If the text is not displayed as desired, the font size may be changed from within either program by clicking Options|Font and changing the font size.

Connect the emulator to your PC with a standard serial cable such as would be used with an external modem or serial printer. **The emulation probe should <u>NOT</u> be plugged into a target system for this tutorial!**

The first time you run WTT11 you must configure a serial port for WTT11 to use to communicate with the emulator. You do this by entering the "Com Setup" dialog box under the Communications menu. Select a com port and click OK.

To check communications with the emulator, plug the wall transformer that powers the emulator into a power strip with an ON/OFF switch and power the emulator on.   Each time the emulator is powered on the following message should appear in the WTT11 Communications Window:

> **TECICE-HC11 Version x.x**
> **Checking Emulator Memory…Memory is functioning.**

The "Checking Emulator Memory…" self test is only done on a cold start up. So you will have to leave the emulator off for several seconds to get this message. If you get this message, at least one way communications is established.   Press the "Download" speed button; if two way communication is working, you should see activity in the communications window as the object code file is downloaded to the emulator. (You can find the "Download" button by moving the cursor over each button until the help hint box appears.)

Under the WTT11 Options menu, make sure the WASM11, Yellow Hint Boxes, and Status Line Help Hints options are checked.

Click on the "Download" button to download the WTTDEMO program to the emulator. Click on the "Reset & Run" button to run the WTTDEMO program. Note that the emulator "Ready"

light goes OFF, the "Running" light goes ON and the all of the WTT11 speed buttons are disabled except for "STOP". This indicates that the emulator is running a user program. A "Running user program…" message appears in the communication window.

Click on the "STOP" button and note that the emulator "Running" light goes OFF, the "Ready" light goes ON and the WTT11 speed buttons become active again. This indicates that the emulator and the WTT11 program are ready for commands.

Click the "Run" button, then the "STOP" button several times and note that each time the "STOP" button is pressed the Code Window is updated to show where the program stopped. The blue PC arrow graphic points to the next program instruction that will be executed when the program is again allowed to run. Also note that the register display on the left side of the button bar is updated to show the current value of all the CPU registers.

Press the "STEP" button. This sends a single step command to the emulator. Note that WTT11 executes a single program instruction and then updates the screen. Press the "STEP" button several times to step the program through its main loop and one subroutine call.

Press the "ANIMATE" button to see an animation of the program. Animate is a continuous sequence of single steps. Press the "STOP" button.

Highlight the "power_on_reset" line in the code window by clicking on it. Then press the "Reset and Run to Highlighted Line" button. This performs a user reset of the CPU to put all internal registers and I/O ports in a known state and sets a breakpoint at address $F800. After execution, the PC arrow should go to this line, indicating that the operation was successful.

Now highlight the "main_loop" line by clicking on it; then press the "Run to Highlighted Line" button. This runs the program from where the PC was ($F800) to the highlighted line ($F811) without resetting the CPU.

Now highlight the "bra   main_loop" at address $F81B; then press the "Set PC to highlighted value" button. Note that this sets the PC value to $F81B. There are other ways to change the PC value. You can just edit its value in the register display area at the left side of the button bar. If you have entered a valid PC value, the code window will be updated to show the code at the new PC value. You can change all CPU register values by editing their values in the register display edit boxes.

If you want to change one of the condition code flags, the "Z" flag for example, just click on its present value and the value will toggle.

The "Manual Commands" window is for entering emulator commands that don't have a corresponding speed button. For example, to send a display memory(dm) command to the emulator, click inside the Manual Command window and type "dm 0 20 <enter>". A memory dump from address 0 to address $2F will appear in the communication window. A complete list of emulator commands is in the Help File.

Click in the "code" window in the margin to the left of any code-generating line of code; for example, click on the "power_on_reset" line at address $F800. Note that a STOP sign graphic appears in the margin. A breakpoint is now set on this line of code. Set additional breakpoint at addresses $F819 and $F81D. Press the "Reset & Run" button and note that program execution stops at the first breakpoint. Press the "Run" button and note that the program stops at address $F81D, the next breakpoint encountered.   Press the "Run" button again to have the program stop at the address $F819 breakpoint. If one or more of the breakpoints are not shown on the screen, you can cycle through them by pressing the "STOP" button. The "STOP" button stops program execution of the user program if the user program is running, and cycles through the breakpoints if the user program is not running.   Up to 5 breakpoints may be set this way.   To remove breakpoints just click on the stop signs. Remove all breakpoints now.

WTT11 has a neat feature called the "Watch Wizard". The Watch Wizard is an intelligent watch point that is used to display the present value of program variables. The Watch Wizard data appears on the status line at the bottom of the screen. However, status line help hints overwrite the Watch Wizard data; so to make sure that the data is in view, you must not position the cursor over a control that activates a status line help hint. Or, you may disable status line hints by unckecking it in the Options manu.

The Watch Wizard shows the variable address, its name and its present value in hex, decimal and binary. The variable that is displayed depends on which line of code is highlighted in the code window. To change the present value of the Watch Wizard variable, simply edit its value in one of the value edit boxes.

Click on the "inc   count_up" line at address $F814 to highlight that line of code. Note that the Watch Wizard now displays the value of the count_up variable at address $0.   Now, click on the "dec   count_down" line at address $F81D to highlight that line of code. Note that the Watch Wizard now displays the value of the count_down variable at address $1. When you highlight a line of code that references a variable, the current value of that variable is shown by the Watch Wizard.

Click on the "inc   count_up" line at address $F814 again to highlight that line of code and press the "Reset and Run to Highlighted Line" button. The Watch Wizard should show the current value of "count_up" to be zero.   Now, repeatedly press the "Run to Highlighted Line" button and note that the value of "count_up" increases by one after each press of the button.   At each button press, the program starts running at address $F814 and runs completely around the main_loop, stopping back at the "inc   count_up" line at address $F814 where the Watch Wizard displays the changed value of count_up.

If you want to look at more than one Watch Point at a time, use the Watch Window speed button on button bar instead of the Watch Wizard at the bottom of the screen.

You have now completed the WASM11 & WTT11 Tutorial. From this point on, you should be able to utilize both programs successfully with the assistance of the online Help program.

3. **TECICE-HC11 USER'S GUIDE**

3.1 GENERAL CIRCUIT DESCRIPTION

3.1.1 Physical Description

The circuitry of the TECICE-HC11 is contained on one 7" X 9" double sided printed circuit board mounted inside the emulator housing. The board contains a Port Replacement Unit, power supply, RS-232 serial port circuits, clock generator, reset circuit, operator control switches, indicators, control processor, emulation processor, memory, address decoder, breakpoint logic, map switch logic, configuration jumpers and emulation probe connections. All user configurable options are accessible on the extreme front or back of the board.

3.1.2 Electrical Description

Schematic diagrams of the emulator circuit board are included in Appendix A. The emulator consists of one 68HC05C8 control processor and a 68HC11 emulation processor.

Control Processor - Schematic Sheet 2 of 9
The control processor (U3) is a 68HC705C8 single chip microcontroller that controls the emulator by performing all communications with the host computer via the RS-232 port, communicating with the emulation processor via U20 and U22, controlling the breakpoint circuits and interfacing with the front panel controls and indicators. The control processor receives commands from the host computer, checks the commands for errors, executes commands by passing instructions and data to and from the emulation processor or other circuits, and reports the results of the commands back to the host computer.

Emulation Processor - Schematic Sheet 3 of 9
The emulation processor has two functions to perform and two memory maps, one memory map for each function. 1) The emulation processor runs user programs from the user memory map. 2) The emulation processor executes commands issued to it from the control processor. It performs this function from the monitor memory map. The emulator has map switch logic to switch between the two memory maps.

There is one socket on the board for the emulation processor. U34 is a 52 pin plcc socket which will be used to emulate both 52 pin and 48 pin versions of the 68HC11. Configuring the TECICE-HC11 to emulate a specific processor consists, in part, of placing the appropriate emulation processor in the socket.

The TECICE-HC11 forces the emulation processor into the expanded mode which allows the address, data and control busses to come outside the chip. This allows these signals, and memory to be accessed by the emulator for program debugging purposes. The address and data busses come out of the emulation processor on what would normally be I/O ports B and C. These I/O ports are reconstructed by the Port Replacement Unit (68HC24).

Emulator Memory - Schematic Sheet 5 of 9
U9 is an 8K X 8 eprom that contains the emulation processors' control program. This eprom is located in the monitor memory map. The emulation processor runs this program when it is in the monitor mode(when the emulator is performing commands issued by the host computer). U12 and U18 are 32K X 8 static ram memory chips. These chips are used for user ram memory space and user pseudo rom space. This is where the users' program resides during emulation. U23 is a 2K X 8 static ram memory chip. It is used for scratch pad ram and stack space for the monitor program. The A-B Register(U22), the B-A register(U20), the A-B and B-A flags(U11) are used for communications between the control and emulation processors.

Breakpoint Circuits - Schematic Sheet 7 of 9
The breakpoint control circuit of the emulator consists of a 16 bit address comparator(U35 & U38), a 16 bit pass counter(U32 & U37), and two programmable logic devices (U8 & U6). This circuit allows the emulator, for example, to break on the 65000th pass around a loop.

Map Switch - Schematic Sheet 6 of 9
There are two map switches, a temporary map switch(U40a) and a permanent map switch(U40b). When a user program is running the memory map is "permanently" switched to the user map. When a breakpoint is reached in the user program the memory map is "permanently" switched to the monitor map. During the execution of commands from the host computer it is sometimes necessary for the emulation processor to switch memory maps for only a few cpu cycles. This is called a temporary map switch. A temporary map switch would occur, for example, during the modify memory command. Temporary map switches occur when the emulation processor fetches an instruction from the monitor map, executes the instruction in the user map then returns to the monitor map for the next instruction.

Clock Circuit - Schematic Sheet 2 of 9
The emulation processor clock must be a TTL level signal. Therefore it can not come from the target system if the target system uses a crystal clock. If clock frequency is critical to proper target system operation then the emulation processor clock must be the same as the target system clock would be when there is an actual 68HC11 in the target system instead of the emulation probe. TECICE-HC11 provides four ways to generate the clock signal. 1) The clock can be derived from the emulators 8MHZ oscillator (X2) and divider(U1 & U2). The divider value can be selected by the Clock Freq. rotary switch on the rear panel of the emulator. Using this method, 8Mhz, 4Mhz, 2Mhz, 1Mhz, .5Mhz, .25Mhz and .125Mhz clock frequencies can be selected. 2) The clock source can be from the External Clock input on the rear panel of the emulator. 3) The clock source can be from the target system provided that this clock is a TTL level signal. 4) The emulators clock oscillator(X2) can be replaced by the user to match his specific requirements.

3.2 TARGET SYSTEM CONNECTION

Connection from the TECICE-HC11 emulator to the user target system is made via the emulation probe. The emulation probe replaces the 68HC11 microcontroller chip in the target system. Two types of emulation probes work with TECICE-HC11. 1) 52 pin PLCC and 2) 48 pin DIP.

See the CONFIGURING THE EMULATOR section for more information on this subject.

3.3 LIMITATIONS

In some ways the emulator will not behave exactly like the 68HC11 microcontroller chip that is being emulated. The user must be aware of these differences and allow for them.

No memory limitation checking is done to assure that the user program will fit in the amount of RAM/ROM space available in the device being emulated. The user must know the memory map of the device being emulated and make sure that only valid RAM/ROM memory locations are used.

The TECICE operates in the expanded mode. (see schematic diagram sheet 3) If the user writes to $x03C to put into single chip mode, the TECICE will need to be reset.

The IRQ and XIRQ pins have an internal pull up resistor on each pin to keep false interrupts from occurring when the pod is not connected to a target system.

Breakpoints are implemented by forcing an SWI to occur. This causes several limitations:

1) The SWI puts nine bytes on the user's stack. Space must be available on the stack.
2) All programs must set the Stack Pointer, otherwise the TECICE won't get valid data off the stack. (The Stack Pointer will default to $FF or $F13D on the TECICE.)
3) Only software breakpoints may be used to break any code that is written in the internal RAM. There is no way to force an SWI onto the bus when the opcode is being read from internal memory.This doesn't effect breaks on reads or writes. Because of this limitation, one should always set the illegal opcode interrupt vector.
4) The user's program can't have SWI instructions when using software break points.

Any interrupts that occur while in the monitor map will still be pending when the user's map is entered, except for XIRQ. It is serviced by a do nothing routine in the monitor program. Also the COP monitor interrupt MAY NOT be turned on.

On every Master Reset, the TECICE will make sure that the config register is equal to $xC or $xD. That is no security, no cop monitor, and no ROM. If any of these options are turned on, they will be turned off by Master Reset.

The emulator does not use power from the target system. Therefore target system power consumption will differ under emulation from when an actual microcontroller chip is used.

The emulator may not use the target system clock source when it is a crystal. Therefore, if the emulators clock frequency has not been configured to match that of the target system program timing may be different when emulating.

If EEPROM in the 68HC11 is turned off and pseudo EEPROM (RAM) is used instead. Jumper J5 must be in the NO PROTECT position if the user wants to be able to write to the pseudo EEPROM in the user mode. See the PGM command to program the actual EEPROM.

3.4   CONTROLS, INDICATORS AND EXTERNAL CONNECTIONS

3.4.1   Front Panel
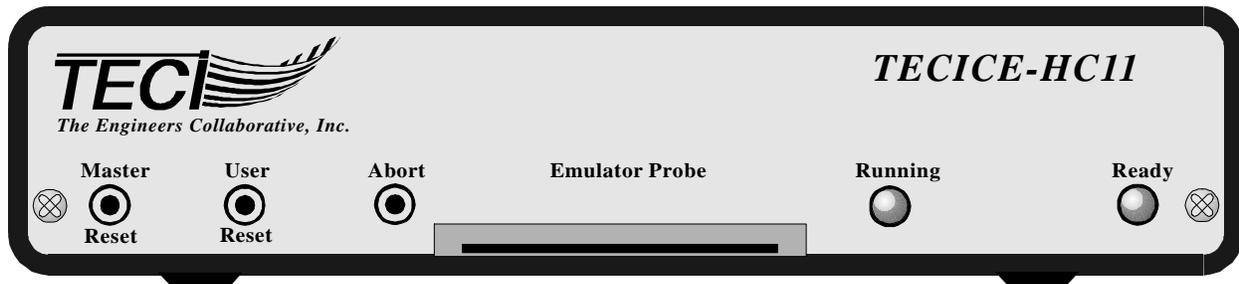
FIGURE 1 - TECICE-HC11 FRONT PANEL

Figure 1 shows the front panel emulator, each of the controls and indicators are described below.

MASTER RESET SWITCH
This is a momentary action pushbutton switch that, when pressed, resets the control and emulation processors, the target system, and user I/O, places the emulator in the monitor mode and sends the monitor prompt to the host computer.

USER RESET SWITCH
This is a momentary action pushbutton switch that, when pressed and released, resets the emulation processor, the target system, and user I/O and forces a map switch to the user memory map, the emulation processor fetches the reset vector from the user map and begins execution of the users' program.

ABORT SWITCH
This is a momentary action pushbutton switch that, when pressed and released, forces a return to the monitor mode from the user mode provided proper operation of user code. The abort switch works like a manual breakpoint. The abort switch does nothing when the emulator is in the monitor mode.

RUNNING LAMP
This is an LED that is ON when the emulator is in the user mode and OFF when the emulator is in the monitor mode. It indicates when the users' program is running.

READY LAMP
This is an LED that is ON when the emulator is in the monitor mode and the control processor is ready to receive a command from the host computer. The READY lamp is OFF when a user program is running or when the control processor is executing a command from the host computer.

3.4.2 Rear Panel

FIGURE 2 - EMULATOR REAR PANEL

CLOCK FREQUENCY SELECTOR SWITCH
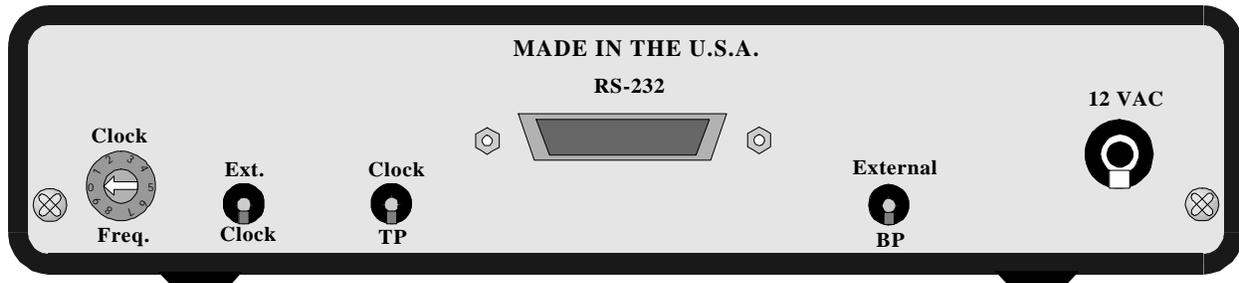The clock frequency selector switch is a ten position rotary switch which selects the emulation processor clock rate. For instructions on how to select a given clock rate, please see Section 3.5.6 (Changing the Clock Frequency) later in this manual.

EXTERNAL CLOCK
The external clock input provides the clock input connection when the clock frequency selector switch is in the external position(switch position 7).

CLOCK TP
This is a clock test point. It provides a convenient place to attach an oscilloscope or frequency counter in case the user needs to verify what clock rate has been selected.

RS-232
This is a DB-25 female connector that connects the serial port of the host computer to the emulator. The schematic diagram Sheet 2 of 9 shows how this connector is wired. For IBM PCs, the same serial cable that is used with external modems can be used. Also, 25 conductor flat ribbon cables with no cross overs or loopbacks work. The cable should have a connector that mates with the serial port of the host computer on one end and a DB-25 male connector on the other end for mating with the emulator.

EXTERNAL BP
This is the external breakpoint input. The user can have external circuitry cause a breakpoint when emulating by pulling this TTL input low. The breakpoint will occur on the first LIR(load instruction register) cpu cycle after the external bp input goes low. This input must remain low until the LIR cycle (about 26 clock cycles worst case) and must go back high before the emulator reenters the user mode after the breakpoint has occurred.

12 VAC
This is the power input. It normally has 12 volts AC at 60/50 cycles from the wall mounted transformer supplied with the emulator. 9-15 volts DC can also be used. The connector inside the emulator is a 2.5mm mono phone jack that accepts standard 2.5mm mono phone plugs.

3.5 CONFIGURING THE EMULATOR

3.5.1 Factory Default Configuration

TECICE-HC11 is shipped from the factory in one of the following configurations depending on the emulation processor ordered.

CONFIGURATION OPTION	68HC11Ax EMUL PROC	68HC11Ex EMUL PROC
Emulation Probe	52 PIN PLCC or 48 PIN CIP	52 PIN PLCC or 48 PIN CIP
Clock Frequency	8.0 Mhz	8.0 Mhz
J1 Jumper Position	VCC	VCC
J2 Jumper Position	S/C	S/C
J3 Jumper Position	NOT EXTAL	NOT EXTAL
J4 Jumper Position	256	512
J5 Jumper Position	PROTECT	PROTECT

3.5.2 Changing The Configuration

Changing some of the configuration options requires removal of the housing cover to provide access to the emulator. To do this, remove the two screws from the rear panel, remove the rear panel and bezel then slide the housing cover toward the rear until it can be detached from the extruded aluminum base. Figure 3 shows component placement on the emulator board with important configuration components noted.

Clock Speed
Selector

Jumper J1

Jumpers
J1 - J5

Emulation
Probe
Connectors

FIGURE 3 - COMPONENT PLACEMENT

3.5.3 Changing Emulation Processors

There is a socket for the emulation processor (U34). To change the emulation processor, 1) remove the present processor and install the new one in the socket, 2) make sure that the appropriate emulation probe is installed, 3) check the option jumpers described below to insure that they are in the correct position, and 4) make sure that the clock speed setting and clock source are correct.

Although there are both 52 pin plcc and 48 pin dip versions of the 68HC11, only a 52 pin plcc can be used as the emulation processor. To emulate a 48 pin dip version, the only change needed is to remove the 52 pin probe and put on the 48 pin probe.

3.5.4 Changing Between Expanded/Multiplexed Mode And Single Chip Mode

To change modes:

1)   Remove the two screws that hold the front panel of the emulator in place, remove the aluminum front panel and bezel and set them aside.
2)   Slide the housing cover toward the front until it is free of the aluminum base of the housing and set the housing cover aside.
3)   Disconnect the emulation cables from the emulator board by unplugging them from connectors JP1 and JP3, or JP2 and JP4.
4)   Plug shorter cable into JP3 for expanded mode, or JP4 for Single Chip mode.

5)   Plug longer cable into JP1 for expanded mode, or JP2 for Single Chip mode.
6)   Select mode with jumper J2. Exp = expanded mode, S/C = Single chip mode.
7)   Configure the "Jump for Target Memory" jumpers per instructions below.
8)   Replace the housing cover. Replace the bezel with the cables going through its center. Screw on aluminum front panel.

When the emulator is used in the expanded multiplexed mode, memory may now be allocated in 8K byte segments as being internal to the emulator or external on the target system. This is particularly useful when you have memory mapped I/O on your target system but still want o use the emulator's memory for ram or pseudo rom. To use this feature, in addition to the instructions above, you must:

1) MOVE BOTH J2 jumpers A & B together when configuring the emulator for single chip (S/C) or expanded multiplexed (EXP) mode.
2) For each 8K byte memory segment that you want to be external on the target system, place a jumper on the header posts marked for that segment.
3) Memory segments can be internal to the emulator or external on the target system in any combination.

BACK

FRONT

TECICE-HC11 CIRCUIT BOARD
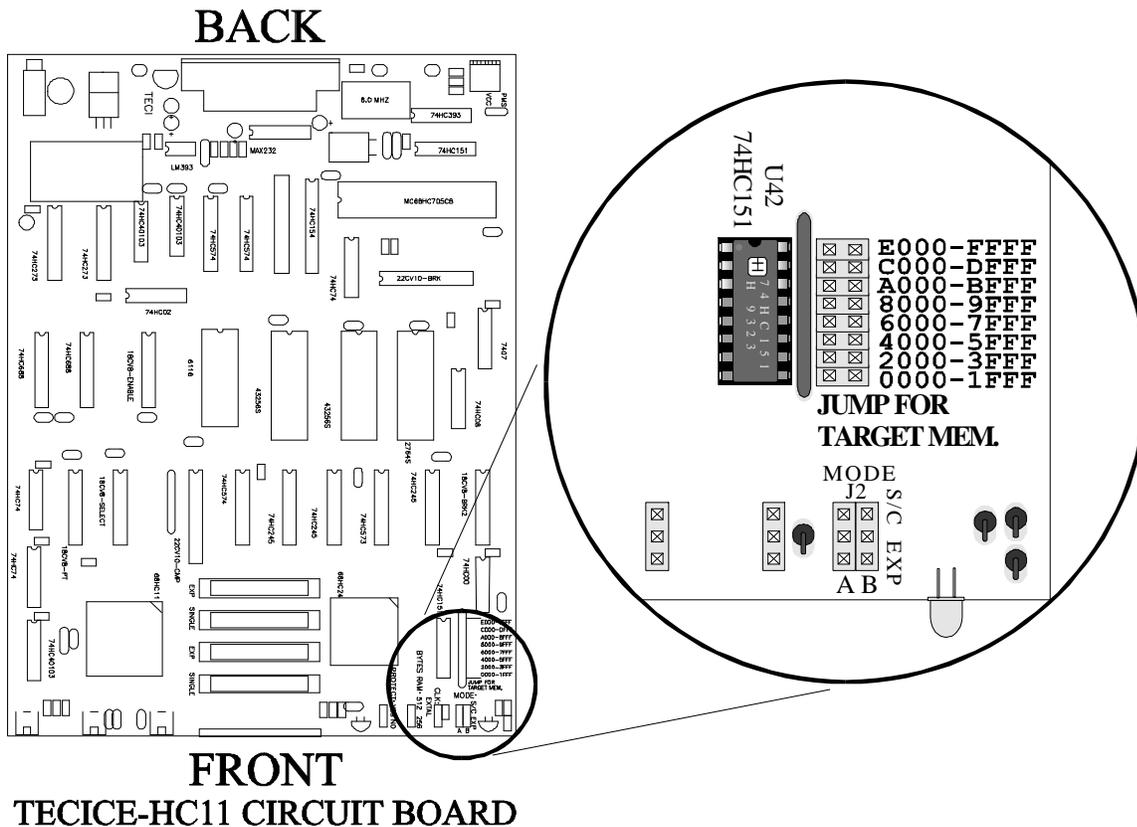
For example, if we had 2 jumpers in place as shown, at 2000-3FFF and C000-DFFF, then the 8K byte memory segments from $2000 to $3FFF and from $C000 to $DFFF would be external on the target system and all other memory would be internal to the emulator. This means that all memory writes would go to both internal and target system memory, but memory reads to locations $2000-$3FFF and $C000-$DFFF would be from the target system while all other reads would come from internal emulator memory.

JUMP FOR
TARGET MEM.
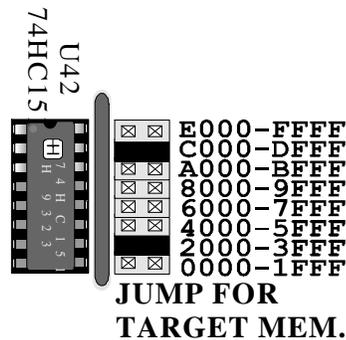
3.5.5 Changing Emulator Probes

Currently two types of emulation probes are supported by TECICE-HC11 they are 1) 52 PIN PLCC and 2) 48 PIN DIP. To change an emulation probe just remove one probe adapter and replace it with the other probe adapter.

?? 52 PIN PLCC EMULATION PROBE
This probe consists of two 26 conductor ribbon cables with connectors on one end that attaches to JP1 or JP2(the longer cable) and JP3 or JP4(the shorter cable) on the emulator board. On the target system end of these cables there are connectors that attach to a 52 PIN PLCC EMULATION PROBE adapter that, in turn, plugs into a 52 pin plcc socket on the target system.

?? 48 PIN DIP EMULATION PROBE
You can change from a 52 pin plcc probe to a 48 pin dip probe by changing the adapter on the target system end of the two 26 conductor ribbon cables without opening the emulator housing. The 48 pin dip probe consists of the same two 26 conductor ribbon cables as the 52 pin plcc probe and they attach to the emulator board at the same points. Only the adapter is different.

3.5.6 Changing The Clock Frequency

Changing the clock frequency usually consists of setting the clock freq. selector switch to the desired divider value. The TECICE-HC11 is shipped from the factory with an 8 Mhz oscillator installed in the X2 socket on the board. X2 may be changed by the user to obtain other clock rate as shown in the chart below.
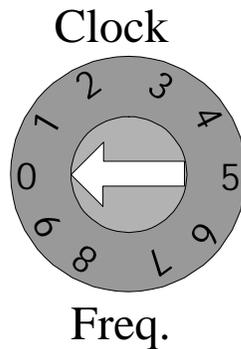
Clock

Freq.

FIGURE 4 - CLOCK FREQUENCY SELECTOR SWITCH

The clock frequency selector switch is a ten position rotary switch which selects the emulation processor clock rate. To select a given clock rate, find the desired rate in the chart below and rotate the switch until the desired number is next to the alignment dot. Note that X2 is the clock oscillator that is located in the upper left hand corner of the board. As shipped from the TECI factory, X2 is an 8.0 Mhz oscillator from which the clock rates shown in the second column of the chart are derived. X2 is a standard type of clock oscillator that can be obtained in many frequencies. If the standard frequencies do not meet user requirements X2 can be replaced with a version that has a more appropriate frequency. Alternately, switch position 7 can be selected and the correct clock rate applied to the external clock input on the rear panel.

CLOCK FREQ. SWITCH NUM	AS SHIPPED X2 FREQ. 8.0 Mhz	X2 FREQ. USER X2
0	8.0 Mhz	USER X2
1	4.0 Mhz	USER X2 / 2
2	2.0 Mhz	USER X2 / 4
3	1.0 Mhz	USER X2 / 8
4	0.5 Mhz	USER X2 / 16
5	0.25 Mhz	USER X2 / 32
6	0.125 Mhz	USER X2 / 64
7	EXTERNAL	EXTERNAL
8	UNUSED	UNUSED
9	UNUSED	UNUSED

### 3.5.7 Setting The Option Jumpers

?? MONITOR MAP CLOCK SELECTOR (J1)

The J1 jumper determines whether or not the fastest internal clock will be used as the clock for the emulation processor while in the monitor map. To illustrate its use, say the user wanted to emulate at a low frequency. The emulator would respond slowly to commands because of the slow clock. If the user selected an oscillator that was a multiple of the desired frequency, then the rotary switch could be used to select the frequency. J1 would be set to "PMS". Now the emulator would emulate at the low frequency, but use the full speed clock for the monitor map to allow quick responses. Unfortunately, if you need to get your slow clock from the EXTAL pin (jumper J3) both maps will have the same slow clock.

This jumper is shown on schematic diagram sheet 2 of 9. This jumper should be in the "VCC" position unless you are emulating at low clock frequencies. The diagram below shows J1 in the "VCC" position.

```
        PMS    ●

        J1     ●
               |
        VCC    ●
```
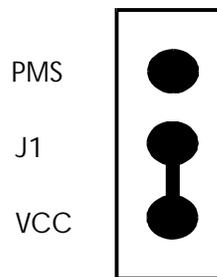
?? MODE SELECT JUMPER (J2)

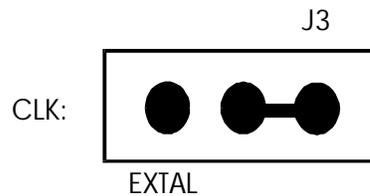The Mode Select jumper (J2) determines whether or not Port Replacement Unit is to be used. This jumper is shown on schematic diagram sheet 3 of 9. This jumper should be in the "S/C" position when emulating a 68HC11 in Single Chip mode, or in EXP when emulating the expanded mode. The diagram below shows J2 in the "S/C" position.

```
MODE:    S/C        EXP

J2      ●━━●   ●

        ●━━●   ●
```
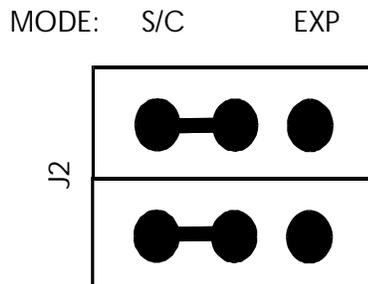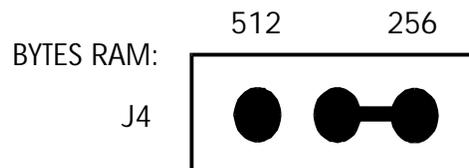
?? EXTAL JUMPER (J3)

The EXTAL jumper (J3) determines whether the emulation clock source is from the emulators internal clock generator or from the target system via the emulation probe. This jumper is shown on schematic diagram sheet 3 of 9. If the clock comes from the target system it must be a square wave at TTL signal levels. The diagram below shows the jumper in the position that selects the emulators internal clock generator.

```
                                    J3

                            ┌──────────────────┐
           CLK:             │  ●    ●━━━━●      │
                            └──────────────────┘
                                  EXTAL
```

?? RAM JUMPER (J4)

This jumper must be set to indicate the amount of internal ram the current emulation processor has. Some 68HC11 processors have 256 bytes of internal RAM and some have 512 bytes of internal RAM. This jumper is shown on schematic diagram sheet 4 of 9. The diagram below shows the jumper in the 256 byte position.

```
                         512          256

        BYTES RAM:   ┌──────────────────┐
                     │  ●    ●━━━━●      │
           J4        └──────────────────┘
```

?? RAM PROTECT JUMPER (J5)

The RAM PROTECT jumper J5 will prevent a user's program from changing the contents of the pseudo ROM and pseudo EEPROM (implemented with RAM). There are cases when a user will not want this protection. An example is when the user is emulating in the expanded mode and on the target system there is additional RAM for use as a stack. Not being able to write to the corresponding RAM would be a problem.
This jumper is shown on schematic diagram sheet 4 of 9. The diagram below shows the jumper in the protected position.

```
              PROTECT:

                     YES          NO

                 ┌──────────────────┐
         J5      │  ●    ●━━━━●      │
                 └──────────────────┘
```
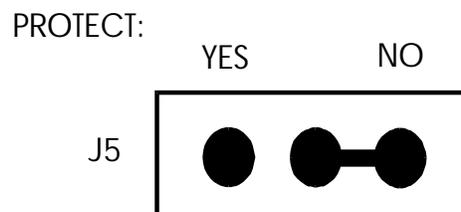
3.6 EMULATOR SELF TESTS

The TECICE-HC11 has a memory test. In addition to this test, the emulator maintains checksums on communications with the host computer and between its two processors. The test and the checksums are provided to give the user means to determine if the emulator is functioning properly and isolate any malfunction so it can be repaired.

The memory test is performed whenever the emulator is powered up, and when the TeST command is used. The memory test only tests the internal RAM (pseudo ROM/EEPROM) used by the TECICE-HC11.

See **Error Messages** section for an explanation of test errors.

?? **Special Test Program**

In Appendix B there is a test program listing, HC11TEST.PRN.   This test program gives the input/output ports a good workout. It is to be run with the emulation probe connected to a test board shown on the Test Board schematic. The test board is designed to be general purpose. It can be useful anytime viewing the contents of the ports and providing inputs in real time are desired.

When the board is used with the test program, JP1, JP2, and JP3 should be shorted, although, not to each other. See the program listings and schematics for more details.

?? **Warnings**

Only one of the two processors on the Test Board schematics may be in place at one time.
A switch on D port MUST be opened when using that pin as an output, otherwise a high output may be shorted to ground. In the test program this means all switches must be open.

3.7 EEPROM PROGRAMMING

Object code files downloaded to the TECICE-HC11 may contain data to be programmed into the EEPROM area of the emulation processor inside the TECICE-HC11 with the following notes and restrictions.

1) For downloads to the EEPROM to be successful the register bank must be at $1000 (the default location and the BPROT register must be cleared). If your user program has relocated the register bank or set the bits in the BPROT register, you must press the MASTER RESET switch on the TECICE-HC11 before attempting the download.

2) If the EEPROM area is not erased before the download is attempted and the new data is different from the present data, each byte will be erased on a byte by byte basis during the download…this takes time (especially at slow clock speeds). To speed things up you can bulk erase the entire EEPROM area with the PGM command before the download. To do this, type at the emulator ready prompt:

> **>PGM 0<enter>**
> then type
> **>PGM byte<enter>**
> then press the MASTER RESET switch on the emulator.

Where byte is the value you want programmed into the config register.

Valid values for the config register are $0C or $0D for the HC11Ax and HC11E9 and $1F and $FF for the HC11E2; any other values for the E2 will leave the EEPROM turned off and out of the memory map.

4. **WASM11 & WTT11 USER'S GUIDE**

4.1 INSTALLING WASM11 & WTT11

The software is supplied on a 3 1/2 inch floppy diskette. The file supplied on this diskette is named SETUP.EXE. This is a self-installing, Windows executable file that will install all of the necessary files when invoked.

To install the software, place the distribution diskette in a floppy drive and run the program A:SETUP.EXE or B:SETUP.EXE from within Windows and follow the on-screen instructions.

*** Important Notice ***

If you have trouble running SETUP.EXE , try running the program with the /M command line switch as follows: >A:SETUP.EXE   /M<enter>   This is the manual installation mode which will prompt you for the location of needed directories, etc.

After running SETUP.EXE you should find the following files in the installation directory(C:\WASM11 is the default).

> WASM11.EXE - Editor/Macro Assembler executable file.
> WASM11.HLP - Editor/Macro Assembler on-line help file.
> WASM11.INI - Editor/Macro Assembler initialization file.
> WTTDEMO.11P - Demo 68HC11 project file.
> WTTDEMO.ASM - Demo 68HC11 project source code file.
> WTTDEMO.LST - Demo 68HC11isting file.
> ITEST.INC - Demo 68HC11 project include file.
> D68HC11.11P - Project file for showing WASM11 source code examples.
> D68HC11.ASM - File showing WASM11 source code examples.
> UNWISE.EXE - Uninstall executable.
> INSTALL.LOG - Installation log file - required by the uninstall program.
> WTT11.EXE - Emulator control program executable file.
> WTT11.HLP - Emulator control program online help file.
> WTT11.INI - Emulator control program initialization file.

*** Important Notice ***

FONT SIZE: If the text is not displayed as desired, the font size may be changed from within WASM11 by clicking Options|Font and changing the font size.

4.2   UNINSTALLING WASM11 & WTT11

To uninstall the software, double click on the UNWISE.EXE program in the installation directory. This program will read the INSTALL.LOG file and remove all files that were installed with the SETUP.EXE program.

4.3 WASM11 PROGRAM OVERVIEW

WASM11 is a windows hosted text editor and 68HC11 macro cross assembler. The program is a general purpose text editor that has special enhancements for managing assembly language programming projects. WASM11 can edit 255 files simultaneously in a tabbed editor environment. To switch between files just click on the tab with the name of the file that you want to edit. Each of the 255 files can be up to 16MB in size and several files can be grouped together and associated with one another. Thus all of the files required for a programming project can be loaded and/or saved with a single command.

WASM11 is a project oriented program. Each project is specified by a unique project file name. Project files have an extension of .11P which stands for HC11Project. When a project file is opened, all of the files that comprise that project are loaded into the editor at once. Each file is assigned its own unique editor tab and the name of the file appears on that tab. A file is selected for editing by clicking on its tab. When a project is saved, all of the files that are open in the editor are saved as part of the project. Among others, the program has speed buttons for opening and saving projects, and adding or removing files from the current project so that managing projects is very easy and natural. However…

*** Important Notice ***

There are two project management details that the **user is responsible** for getting correct: (1) Projects <u>must</u> be saved with the .11P file extension; and (2) the main source code file of the project <u>must</u> have the same file name as the project file with an extension of .ASM. This is how the assembler knows which file to assemble first.

Example: If the project file name is **DEMO.11P**, then the main source code file must be named **DEMO.ASM**. All other project files may have any name/extension.

The figure below shows some of the more commonly used controls and features of the WASM11 program. Most of the speed buttons perform normal text editing functions. Refer to the online help file for complete descriptions of these speed buttons. Three of the speed buttons are specifically for program development work. The "✍ 6811" speed button assembles the project files and creates an object code file, a listing file and a map file. The "D" speed button calls your favorite debugging program. We recommend the WTT11.EXE emulator control program, another TECI 68HC11 development tool. The "P" speed button calls your programmer software. Which programs the "D" and "P" buttons invoke depends on what you specify under the Options/Project Options dialog box. You can set up these buttons to call any program you like by entering the command line parameters for your program in the edit box provided.

WASM11 MAIN SCREEN

Open Project
Save Project
Add File to Project
Remove File from Project
Create Blank Editor Page
Save Current File
Print Current File

Cut
Copy
Paste
Undo
Redo
Find
Find Next

Find and Replace
Exit
Assemble Project Files
Run Debugger
Run Programmer

```
WASM11 - MCPM3B.11P - [C:\RBJPAS\TECIASM\MCPM3B.ASM]        _ □ ×
File  Edit  Search  Options  Window  Help

temp_accum      rmb     1                       ;
;
;Note: The last ram location, not counting the stack is at $BF.
;
                org     rom             ;start of program memory
power_on_reset  lda     #%01001010      ;reset
                sta     port_a          ;init port a
                clr     port_b          ;
                lda     #%11001111      ;init port c
                sta     port_c          ;
                lda     #%11111111      ;
                sta     port_a_ddr      ;
                sta     port_b_ddr      ;
                sta     port_c_ddr      ;
                ldx     #ram            ;
                lda     #176            ;
clear_ram       clr     ,x              ;zero ram
                incx                    ;
```

6811   D   P

Line:176   Col:16   Insert   File Lines: 2506   File Bytes: 134197   1:49:10 PM

\MCPM3B.ASM /MCPM3B.LST /MCPM3B.S19 /MCPM3B.MAP /

File Modified
Column Number
Line Number
Insert/Overwrite

File Lines
File Size
Current Time
File Tabs - Click to change file.

4.4   WTT11 PROGRAM OVERVIEW

WTT11 USER'S MANUAL
ADDENDUM

NOTES ON WTT11 .INI FILE

When WTT11 is exited, setup information is saved in an .INI file so that when the program is invoked again it can start up with the same settings as were used last. When invoked, WTT11 first looks for this .INI file in the same directory as the WTT11.EXE executable file. If it doesn't find the .INI file in this directory, it looks for the file in the directory that contains the listing file which is specified on the command line as the second parameter.

EXAMPLE:

> C:\TECI\WTT11.EXE   D:\MYPROJ\MYFILE.LST

> WTT11 will first look in C:\TECI for the WTT11.INI file. If it doesn't find it there, it will look in D:\MYPROJ .

This is important for client server networks where the clients don't have permission to write to certain directories on the server.

On client server networks where the clients don't have permission to write to the directory where WTT11.EXE will be stored, you should delete the WTT11.INI file from this directory. This will force WTT11.EXE to look for and write the .INI file in the user's directory.

WTT11 is a Windows emulator control program for TECI's TECICE-HC11 in circuit emulators. It takes the place of the older DOS TT program. The graphic below shows several of the most salient features of WTT11. Some major enhancements over the older emulator control program are as follows:

1) A dynamic listing file window that follows code execution.

2) The ability to set or remove breakpoints simply by clicking on a listing file line.

3) The ability to change register or memory values in the emulator simply by changing them in Windows style edit boxes.

4) A Watch Wizard to quickly show you the current value of any program variable.

5) Speed buttons to automate common tasks.

6) Much closer integration with the editor/assembler.

7) and more.

*** Important Notice ***

FONT SIZE: If the text is not displayed as desired, the font size may be changed from within either program by clicking Options|Font and changing the font size.

WTT11 MAIN SCREEN

Run prog from current location.

Run prog from current location
to highlighted line of code.

Single Step

Reset MPU and run prog.

Click to open a
watch window.

Reset MPU and run prog
to highlighted line of code.

Click to set the PC
value to that of the
highlighted line of
code.

Click to stop
running user
program.

Click to download code.

Animate

CPU register display -
A and X registers are
shown in Hex, Dec.
and Binary. To change
a register value, just
edit the registers
current value.

Condition Code Reg.
Click on a flag to
toggle its value.

Click here to set
or clear breakpoints.

Current program
counter location.
Code window
follows code
execution.

Code window.
Automatically
updated - follows
code execution.

Communications
window.

Manual command
window.

```
WTT11 - C:\HC11DEMO\WTTDEMO.LST
File  Options  Communications  Help
PC=F80B IX=FFFF D=FFFF A=FF=255=11111111 SXHIN ZVC
SP=00FF IY=0000       B=FF=255=11111111 11111111

                ;
    F800                      org    program         ;start of program memory
    F800 8E00FF  power_on_reset lds  #$ff              ;load stack pointer
    F803 86FF                  ldaa   #%11111111      ;program starts here aft
    F805 B71003               staa   port_c           ;initialize port A with
    F808 B71007               staa   port_c_ddr       ;set port A to an output
    F80B 7F0000               clr    count_up          ;clear
    F80E 7F0001               clr    count_down        ;the ram variables
    F811 731003  main_loop    com    port_c            ;compliment port A
    F814 7C0000               inc    count_up          ;increment count up
    F817 8D04                 bsr    decrement_byte    ;increment a word
    F819 4C                   inca                     ;
    F81A 5A                   decb                     ;
    F81B 20F4                 bra    main_loop         ;loop forever
                ;

PC = F80B    A = FF    B = FF    D = FFFF    X = FFFF    Y = 0000
SP = 00FF    CC = SXHINZVC    Target RAM = OFF    Step mode = OVER Interrupt
>DM 0000
  0000= 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  : ...............

Manual Commands

WatchWizard   Add= 0000    Name= count_up      Value=00 =0 =00000000
```

Watch Wizzard - Just click on any line of code
and the current value of the variable referred to
in that line is displayed in hex, decimal and
binary.

5. APPENDIX

5.1 TUTORIAL LISTING FILE

```
;*****************************************************************************
;**                      WTTDEMO.ASM   4/30/97   RBJ                       **
;**                                                                        **
;** A WASM11 cross assembler and WTT11 emulator demonstration program.     **
;**                                                                        **
;**              TECI - The Engineers Collaborative, Inc.                  **
;**              RR#3 Box 8c                                               **
;**              Barton, VT 05822                                          **
;**              Ph (802)525-3458  Fax (802)525-3451                       **
;**              email bobj@tec-i.com                                      **
;*****************************************************************************
;
irq_vectors     equ     $ffc0
;
;
ram             equ     $0                      ;start of ram area
program         equ     $F800                   ;start of program area
;
registers       equ     $1000
port_c          equ     registers + 3           ;
port_c_ddr      equ     registers + 7           ;
;
                org     ram                     ;ram variables
count_up        rmb     1                       ;
count_down      rmb     1                       ;
;
                org     program                 ;start of program memory
power_on_reset  lds     #$ff                    ;load stack pointer
                ldaa    #%11111111              ;program starts here after reset
                staaa   port_c                  ;initialize port A with 1's
                staa    port_c_ddr              ;set port A to an output port
                clr     count_up                ;clear
                clr     count_down              ;the ram variables
main_loop       com     port_c                  ;compliment port A
                inc     count_up                ;increment count up
                bsr     decrement_byte          ;increment a word
                inca                            ;
                decb                            ;
                bra     main_loop               ;loop forever
;
                include itest.inc               ;an include file
;
dummy_isr       rts                             ;should never be called
;
                org     irq_vectors             ;interrupt service routine vectors
                dw      dummy_isr       ;reserved
                dw      dummy_isr       ;reserved 1
                dw      dummy_isr       ;reserved 2
                dw      dummy_isr       ;reserved 3
                dw      dummy_isr       ;reserved 4
                dw      dummy_isr       ;reserved 5
                dw      dummy_isr       ;reserved 6
                dw      dummy_isr       ;reserved 7
                dw      dummy_isr       ;reserved 8
                dw      dummy_isr       ;reserved 9
                dw      dummy_isr       ;reserved 10
                dw      dummy_isr       ;sci serial system
                dw      dummy_isr       ;spi serial transfer complete
                dw      dummy_isr       ;pulse accumulator input edge
                dw      dummy_isr       ;pulse accumulator overflow
                dw      dummy_isr       ;timer overflow
                dw      dummy_isr       ;timer output compare 5
```

```
        dw    dummy_isr            ;timer output compare 4
        dw    dummy_isr            ;timer output compare 3
        dw    dummy_isr            ;timer output compare 2
        dw    dummy_isr            ;timer output compare 1
        dw    dummy_isr            ;timer input capture 3
        dw    dummy_isr            ;timer input capture 2
        dw    dummy_isr            ;timer input capture 1
        dw    dummy_isr            ;real time interrupt
        dw    dummy_isr            ;!IRQ (external pin or parallel I/O)
        dw    dummy_isr            ;!xirq pin
        dw    dummy_isr            ;SWI
        dw    dummy_isr            ;illegal opcode trap
        dw    dummy_isr            ;COP failure (reset)
        dw    dummy_isr            ;COP clock monitor fail
```

5.2 TECICE-HC11 SCHEMATICS