



WP11 User's Manual

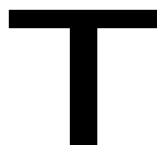
68HC11 Microcontroller Programming Software

Products Described

WP11.EXE Windows Bootloader Program

68HC11 Devices Supported

MC68HC711E20
MC68HC711E9
MC68HC811E2
MC68HC711D3
MC68HC11A1/A8
MC68HC711K4



The Engineers Collaborative, Inc.

Web Site at <http://www.tec-i.com>

Email support@tec-i.com

*** * * IMPORTANT WARRANTY AND LIABILITY INFORMATION * * ***

The Engineers Collaborative, Inc. makes no warranties, expressed or implied for this software. No warranty of fitness for a particular purpose is offered. The user is advised to test the 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 * * ***

US copyright law and international treaty provisions protect this software. 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 cannot 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.

(C) Copyright 1989-2004 The Engineers Collaborative, Inc.
Web Site at www.tec-i.com
Email support@tec-i.com
All Rights Reserved
Printed in the U.S.A.

TABLE OF CONTENTS

1.0 INTRODUCTION 4

2.0 SYSTEM REQUIREMENTS..... 4

3.0 DEVICES SUPPORTED..... 4

4.0 GETTING STARTED..... 4

4.1 Software Installation..... 4

5.0 68HC11 SPECIAL BOOTSTRAP MODE..... 6

6.0 THE WP11.EXE PROGRAM..... 6

6.1 Program Overview..... 6

6.2 WP11 Screen Shot..... 7

6.2.1 WP11 Status Panel 7

6.2.2 Changing the Config Register 8

6.3 WP11 Menus 8

6.3.1 File Menu..... 8

 Load Buffer from File 8

 Load Buffer from Chip..... 9

 Save Buffer 9

 Save Buffer As..... 9

 Exit Program..... 9

6.3.2 Edit Menu..... 9

 Clear Buffer 9

 Fill Buffer 10

 Verify Buffer Against Chip 10

 Edit Buffer Contents..... 10

6.3.3 Select Device Menu..... 11

6.3.4 Communications Menu 12

 Com Port Setup and Test 12

 Com Port Troubleshooting Chart..... 13

6.3.5 Options Menu..... 14

 File Format Options 14

 Load Buffer On Startup..... 14

 Turn Off Hint Boxes..... 14

 Turn Off Status Line Hints 15

 Turn Off Memory Map Checks..... 15

6.3.6 Help Menu 15

 Contents..... 15

 About..... 15

6.4 Chip Operation Buttons..... 15

 Initialize Device..... 15

 Communications Check 16

 Blank Check 16

 Erase EEPROM and Config Reg..... 16

 Program Config Reg 16

 Program EEPROM..... 16

 Program EPROM 17

 Program Entire Device 17

7.0 TUTORIAL EXAMPLE SESSION 17

1.0 INTRODUCTION

This manual provides information on our WP11.EXE program, Windows software used for programming 68HC11 family microcontrollers.

WP11.EXE takes input from object code files created with 68HC11 assemblers or compilers. These object code files may be in Motorola S-record, Intel Hex or binary memory image formats. The software also allows the operator to perform many other useful programming functions such as erasing, blank checking & verifying devices, displaying, editing, exporting and changing the format of the object code files and filling unused memory locations with a user specified "fill" byte. The software was designed to work with our P11 Programming Board but will also work with any hardware that supports the 68HC11 special bootstrap mode of operation.

TECI provides several other development tools for 68HC11 family microcontrollers such as our Cross Assemblers WASM11.EXE and TASM11.EXE and real time in-circuit emulator TECICE-HC11.

2.0 SYSTEM REQUIREMENTS

WP11.EXE requires a minimally configured IBM PC or compatible with at least 64Meg RAM memory, one 3.5" disk drive, one free serial port designated as COM1-COM4 and Windows 3.1, 98, ME, NT, 2000 or XP.

3.0 DEVICES SUPPORTED

Presently WP11 supports the following 68HC11 family members:

MC68HC711E9
MC68HC711E20
MC68HC811E2
MC68HC711D3
MC68HC11A1/A8
MC68HC711K4

4.0 GETTING STARTED

4.1 Software Installation

The software is contained in a single self-extracting .EXE file called wp11setup.exe. This is a standard windows install file and you need to be running windows for it to guide you through the install process. You may have downloaded the wp11setup.exe file from our web site at www.tec-i.com or received the file on a single 3.5" floppy diskette. To start the setup process you can either double click on the file name from windows explorer or use "RUN" from the windows start menu.

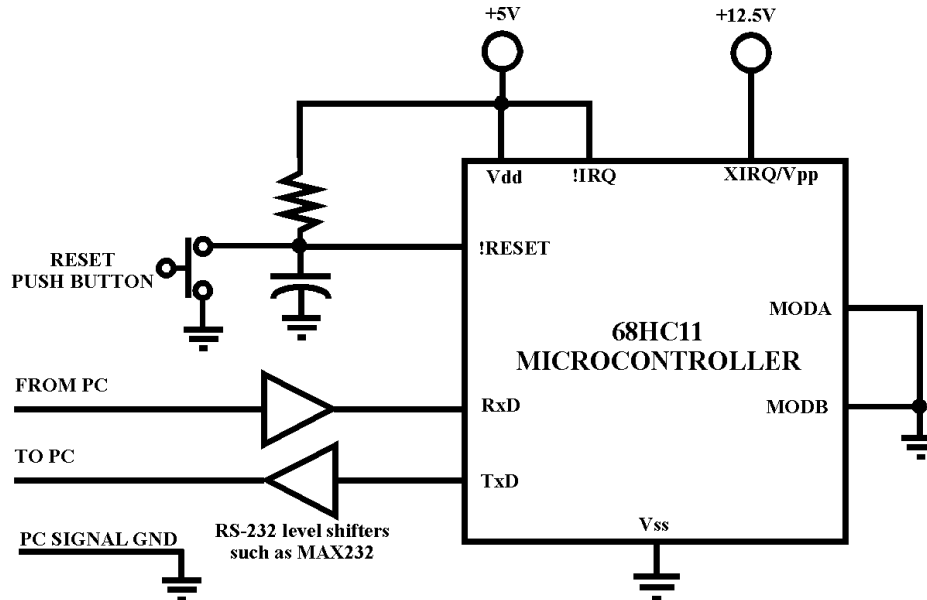
The files extracted from wp11setup.exe are:

WP11.EXE	This is the Windows program used to control the programming process.
WP11.HLP	This is the help file for WP11.EXE.
WP11Man.PDF	This manual in Adobe Acrobat .PDF format. The Adobe Acrobat reader software is available free of charge from a number of online sites such as: www.adobe.com/products/acrobat/readstep.html
P11_A8.MIK	This is an HC11 object code file in Motorola S-Record format that contains the programming code for MC68HC711A1/A8 devices. WP11.EXE downloads this code to the programmer when A1 or A8 devices are being programmed.
P11_D3.MIK	This is an HC11 object code file in Motorola S-Record format that contains the programming code for MC68HC711D3 devices. WP11.EXE downloads this code to the programmer when D3 devices are being programmed.
P11_E2.MIK	This is an HC11 object code file in Motorola S-Record format that contains the programming code for MC68HC811E2 devices. WP11.EXE downloads this code to the programmer when E2 devices are being programmed.
P11_E9.MIK	This is an HC11 object code file in Motorola S-Record format that contains the programming code for MC68HC711E9 devices. WP11.EXE downloads this code to the programmer when E9 devices are being programmed.
P11_E20.MIK	This is an HC11 object code file in Motorola S-Record format that contains the programming code for MC68HC711E20 devices. WP11.EXE downloads this code to the programmer when E20 devices are being programmed.
P11_K4.MIK	This is an HC11 object code file in Motorola S-Record format that contains the programming code for MC68HC711K4 devices. WP11.EXE downloads this code to the programmer when K4 devices are being programmed.
E2_TEST.S19	This is an object code file in Motorola S-Record format. It is used for test programming of MC68HC811E2 devices.
K4EEPROM.S19	This is an object code file in Motorola S-Record format. It is used for test programming of the EEPROM memory area of MC68HC711K4 devices.

EEPROM.S19 This is an object code file in Motorola S-Record format. It is used for test programming of the EEPROM memory area of MC68HC711A1/A8, MC68HC711E9, and MC68HC711E20 devices. Memory locations from \$B600 through \$B7FF are programmed by this file.

5.0 68HC11 SPECIAL BOOTSTRAP MODE

WP11 uses the 68HC11 special bootstrap mode to program devices. You will need to build or buy the circuitry required to do this. The figure below shows the basics of this circuitry.



Special Bootstrap Circuit

Much more information on the special bootstrap mode can be found online by doing a Google search for "68HC11 special bootstrap mode".

6.0 THE WP11.EXE PROGRAM

WP11.EXE is a Windows program that is used to program 68HC11 microcontrollers by controlling hardware that supports the special bootstrap mode of these devices. WP11 uses a serial port to communicate with the programming hardware, which can be the P11 board, or any circuit that uses the HC11 special bootstrap mode.

WP11 is usually started from the "START" menu of your Windows PC or by double clicking on the WP11.EXE file name from within Windows Explorer.

6.1 Program Overview

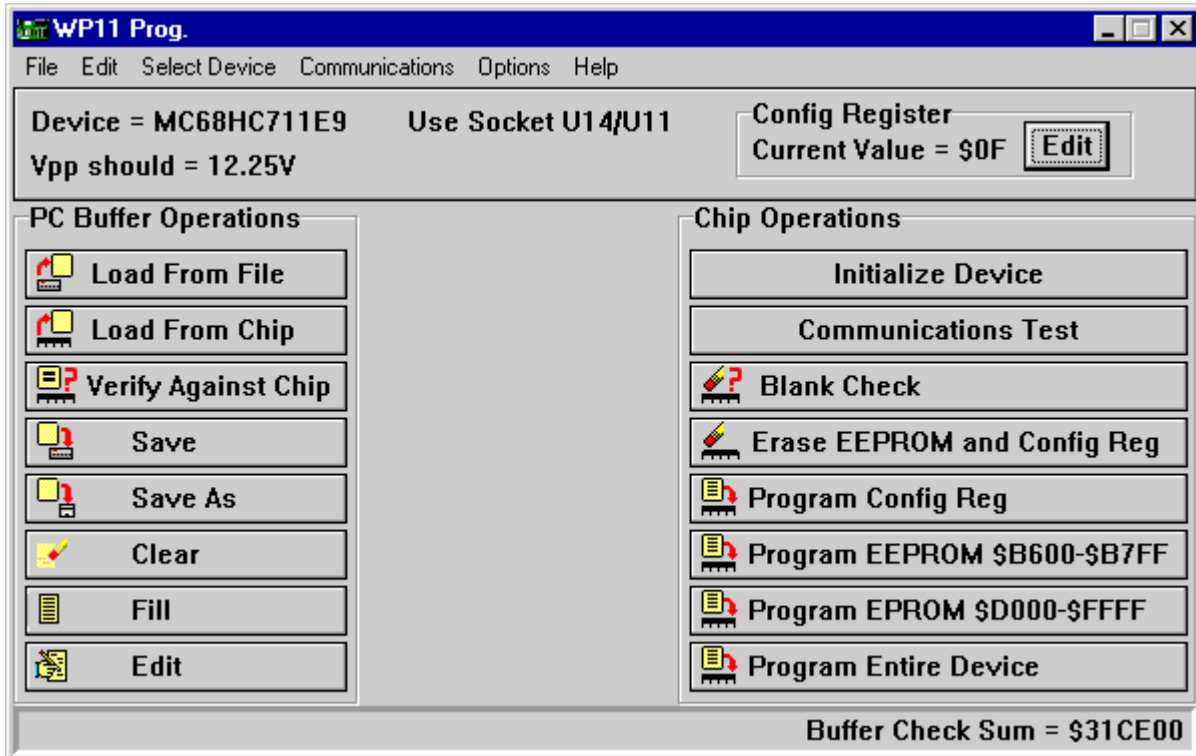
To program a device using WP11.EXE follow these steps:

- 1) Make sure that your PC serial port is properly set up to communicate with the hardware. Use the "Communications" menu item "Com Port Setup and Test" to verify proper operation of your serial port.
- 2) Select the desired HC11 family member using the "Select Device" menu.

- 3) Load the PC Buffer with the data that you want to place in the HC11 device. This data can come from a number of different sources, as we will see.
- 4) Make sure that the programming hardware is powered off and insert a device in the appropriate programming socket. Power up the programming hardware.
- 5) Click the “Initialize Device” button and follow the instructions.
- 6) Click on a programming button to perform the desired operation.

The rest of this section will explain the WP11 menu options.

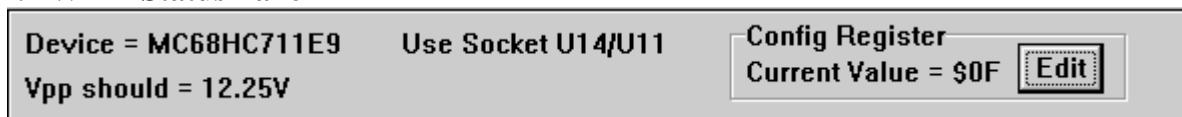
6.2 WP11 Screen Shot



As you can see from the screen shot above WP11 has the usual Windows menu structure above a status panel. Below the status panel is a column of buttons on the left side, which are used to invoke various operations on the PC buffer. Remember, the PC buffer is memory inside your PC that contains the data that will be transferred to the HC11 chip during programming.

The column of buttons on the right side of the screen is used to invoke various chip programming operations.

6.2.1 WP11 Status Panel



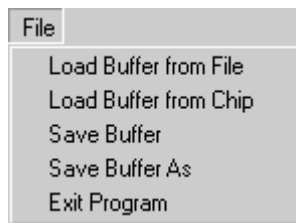
The status panel is used to show information about the currently selected device. It gets updated when a different HC11 device is chosen from the Select Device menu.

6.2.2 Changing the CONFIG Register

The Status Panel contains a means of changing the CONFIG register value. When the 68HC11 family member is changed the CONFIG Register value (in the PC) automatically changes to the erased state of the new device. You may, however, wish to change the CONFIG Register to some other value. Pressing the “EDIT” button will enable you to specify a new value. You must be careful to specify values that are valid for the selected device. Changing the CONFIG Register will not affect the microcontroller until the device is programmed.

6.3 WP11 Menus

6.3.1 File Menu




Load Buffer from File is the same as 




Load Buffer from Chip is the same as 



Save Buffer is the same as 



Save Buffer As is the same as 



Load Buffer from File

Use this selection to load an object code file into the PC buffer.

When this option is selected, a standard Windows “Open” dialog box is displayed and the operator selects the desired object code file. File formats are automatically detected by WP11.EXE, and may be Motorola S-Record, Intel Hex or a binary memory image. Binary files must be 64K bytes in length. If the file name is entered correctly and can be opened and read by WP11.EXE, a memory image of the target chip is created in the buffer. If the file can't be opened, an error message is displayed and the operator is given another chance to enter the file name. This continues until the operator enters a correct file name or presses the “Cancel” button.

Load Buffer from Chip

Use this selection to load the contents of a programmed chip into the PC buffer.

A chip must be installed in the programming hardware and properly initialized for this menu option to be enabled. When this option is selected, a status Window with a Progress Gauge is displayed to keep the operator aware of the command progress. During the upload, the chip being read computes a checksum of the bytes sent to the PC. This checksum is sent to the PC where it is compared to a checksum of the received bytes computed by the PC. The data is placed in the buffer only if the correct number of bytes was received and the checksums match.

Save Buffer

This selection saves the contents of the buffer in a file on your PC. The file name used is the same as the currently opened file. If no file is currently opened a standard Windows "Save As" dialog box is opened to allow the operator to enter a file name. The file can be in one of three object code formats, Motorola S-Record, Intel HEX, or binary image format. Use the "Options" menu to select the format.

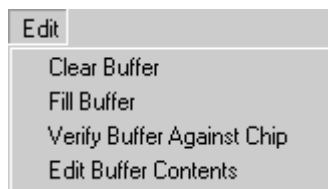
Save Buffer As


This selection is very nearly the same as the "Save Buffer" menu item described above except the "Save As" dialog box is opened immediately so that a new file name can be specified.

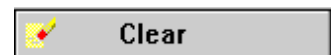
Exit Program


This selection ends WP11.EXE execution. If the buffer was changed since the last save the operator is given the option of saving it. A WP11.INI file is created and saved in the WP11.EXE directory and is used to restore program options and settings the next time the program is used. The WP11.INI file is a standard text file and may be edited to change program options if need be. If WP11.INI cannot be found, WP11.EXE starts up with default settings for the program.

6.3.2 Edit Menu



Clear Buffer is the same as 



Fill Buffer is the same as 



Verify Buffer Against Chip is the same as 



Edit Buffer Contents is the same as 



Clear Buffer

This selection erases the data in the buffer in such a way as to leave each byte of the buffer the same as the corresponding byte in an erased chip. With presently supported devices each byte is returned to \$FF. This function erases the current contents of the buffer.

Fill Buffer

This selection provides a means of filling the entire buffer with a byte of data that you specify. This function overwrites the current buffer contents. As an example of the use of this function consider this:

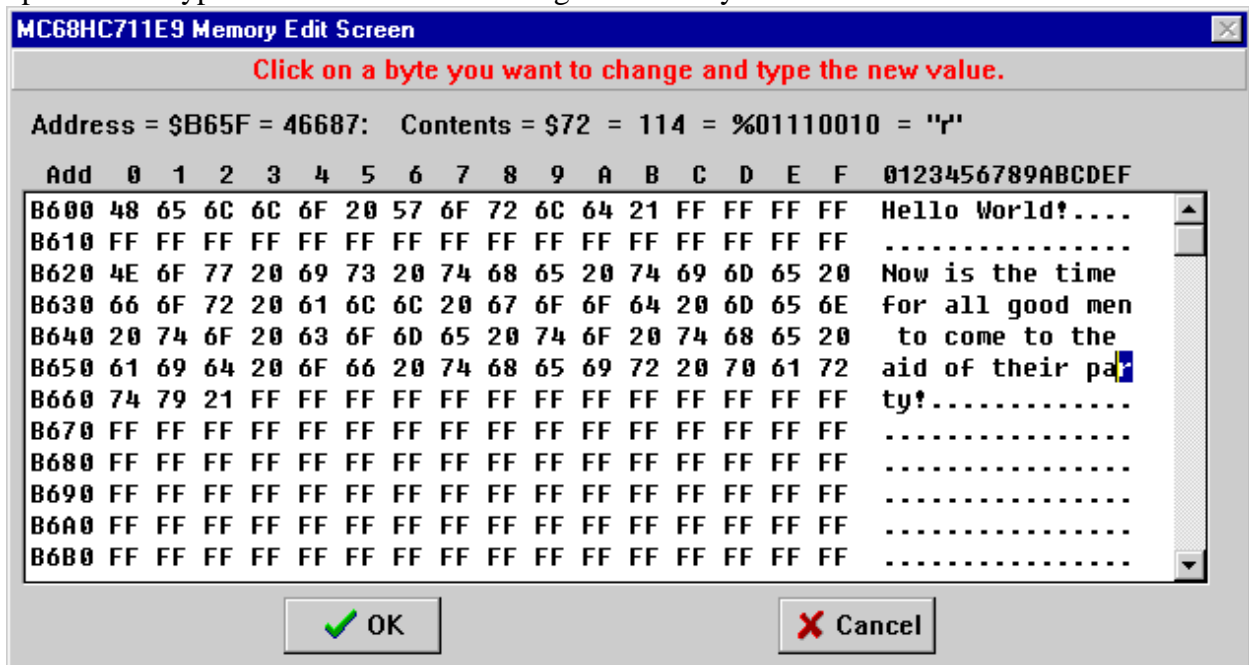
Suppose we want to have all the bytes in a device that are not taken up by our program set to a specific value. Say we want all unused memory locations filled with \$3F, the op-code for the SWI instruction. To do this use the "Fill Buffer" function and specify \$3F as the fill byte. Then use the "Load Buffer from File" function to load your program into the buffer. The buffer is not changed before the "Load Buffer from File" function is executed and the "Load Buffer from File" function does not change any bytes in the buffer that do not have data in the object code file. We started with the buffer completely filled with \$3F and the "Load Buffer from File" function overwrote only those bytes that had data specified in the object code file so we end up with all of the bytes in the buffer that are not used by our program still set to \$3F.

Verify Buffer Against Chip

This selection first uploads data from the chip similar to the "Load Buffer from Chip" function. The same dialog box and progress gauge is used during the upload. If uploading of the data is successful then each byte of the uploaded data is compared to the corresponding byte in the buffer. Data in the buffer is not changed. A pass/fail dialog box is displayed to end the function

Edit Buffer Contents

When this function is selected data in the buffer is copied to a special edit area. The contents of this special edit area are then displayed on a Memory Edit Screen as shown below where the operator can type new data over the existing data directly on the screen.

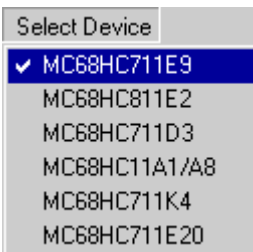


Each line in the edit box above shows 16 bytes of the special edit area data in both hex and ASCII formats with the address of the first of the 16 bytes shown at the beginning of each line.

As an example, in the sixth line shown above, the 'r' in "party" is highlighted. We just clicked on the 'r' to highlight it. The status line at the top of the screen tells us that the 'r' is at address \$B65F. It further tells us that \$B65F = 46687 decimal and that the hex value of the 'r' is \$72 which is 114 decimal which is 01110010 binary. If we wanted to change the 'r' we would simply type the new value, the screen would be updated, and the cursor would move to highlight the next character. We can also edit in the HEX display area by simply clicking on the character we want to change and typing the new value. It's all very easy. Try it!

Clicking the "OK" button copies the data in the special edit area back to the buffer thereby making the edits take effect. Clicking the "Cancel" button discards the data in the special edit area leaving the buffer unchanged.

6.3.3 Select Device Menu



This menu is used to select the HC11 family member that you want to program. The part number is checked when it is the currently selected device. The part number is also displayed on the status panel with the "Device = XXXXXXXXXXXX" label.

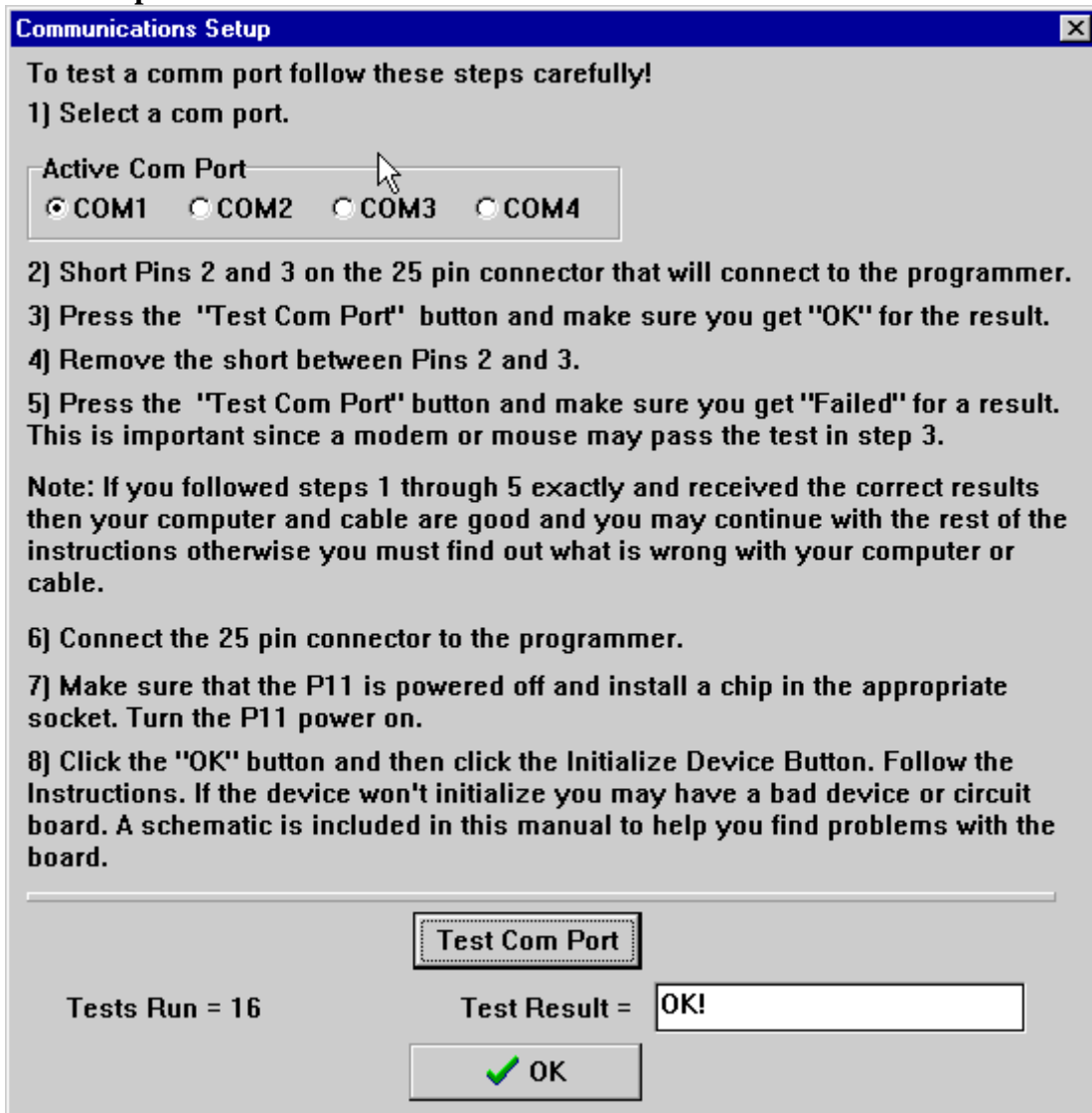
6.3.4 Communications Menu



*** Important Note ***

The instructions on the Communications Setup screen below were written specifically for our P11 programmer that has a 25-pin serial port connector. Your hardware may have a different connector. If your hardware has a standard 9-pin serial port connector you would still short pins 2 and 3 where called for in the instructions. If your hardware has a non-standard serial port connector then you would short **RxD** to **TxD** where the instructions call for shorting pins 2 and 3. The WP11 help file has more information on PC serial port connections.

Com Port Setup and Test



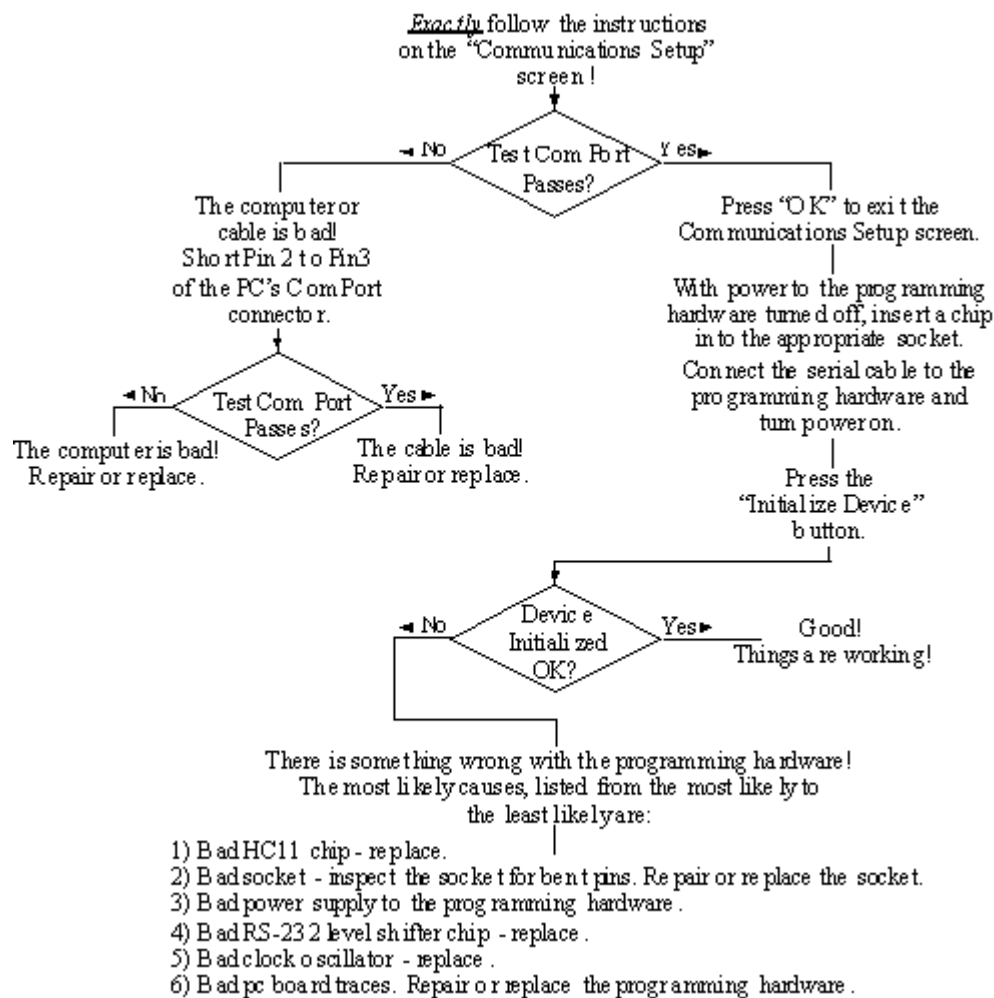
When the Com Port Setup and Test menu item is selected the Communications Setup screen shown above is displayed. The screen has detailed instructions and a test facility for determining whether or not a working serial port and cable are in use.

***** Important Note *****

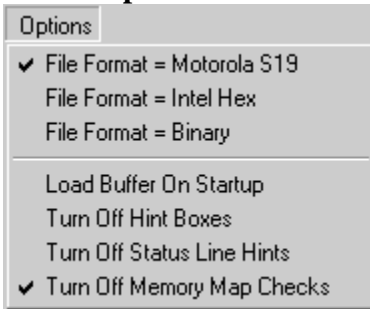
In order to perform any of the chip operations the computer must be able to communicate with the programming hardware. Our experience indicates that the most difficult task faced by a first time user is to establish this communication. This has to do with the non-standard nature of the EIA RS-232 Standard and the many different RS-232 cables that exist. If communications aren't working, the very first thing to do is to determine if the cause lies with the computer and cable or with the programming hardware.

The test facility and instructions on this screen provide the tools required to determine if the problem is with the computer and cable or with the programming hardware.

Com Port Troubleshooting Chart



6.3.5 Options Menu



File Format Options

The first three Option Menu choices allow the user to specify the format to use when saving the buffer in an object code file. This format will be used when the “Save” or “Save As” buttons are pressed.

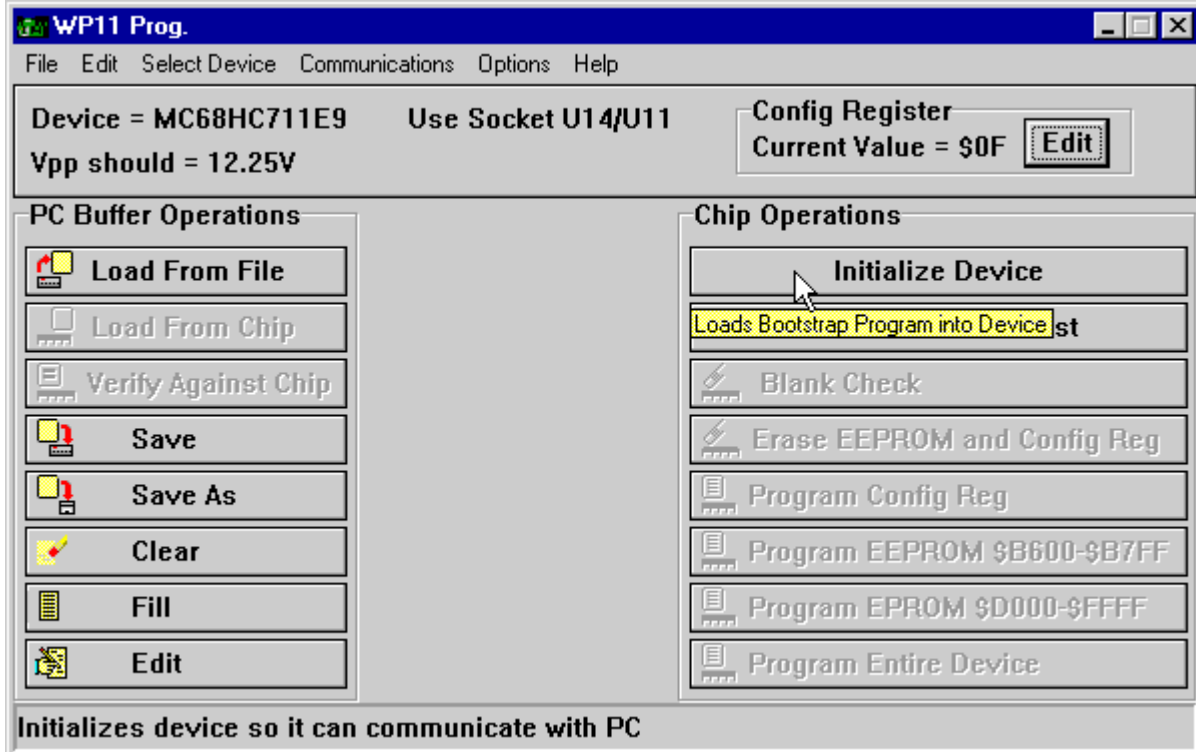
Note: When loading a file into the buffer the file format is automatically detected so these options come into play only when saving the buffer contents to a file.

Load Buffer On Startup

WP11 remembers the file name of the object code file loaded or saved in an “.INI” file. When the “Load Buffer On Startup” option is checked WP11 automatically loads this file back into the buffer.

Turn Off Hint Boxes

Referring to the screen shot below notice that the cursor is located over the “Initialize Device” button. Also notice that just below the cursor is a box containing the text “Loads Bootstrap Program into Device”. This box is called a “hint box” and tells the operator what will happen when the button is pressed. These fly over hint boxes are probably not needed once a user is familiar with the program. The hint boxes will not be displayed when the “Turn off Hint Boxes” option is checked



Turn Off Status Line Hints

Referring to the screen shot above notice that the cursor is located over the “Initialize Device” button and that the panel at the bottom of the WP11 window has the text “Initializes device so it can communicate with PC”. This text is called a “help hint” and gives the user more information about what will happen when the “Initialize Device” button is pressed. These fly over help hints are probably not needed once a user is familiar with the program. The help hints will not be displayed when the “Turn off Status Line Hints” option is checked.

Turn Off Memory Map Checks

WP11 knows the memory maps of supported family members. That is to say that it knows where in memory the EEPROM, EPROM, CONFIG Register and vectors are located. It knows which memory locations are valid for a particular family member and which ones are invalid. Normally, when WP11 is loading the buffer with data in an object code file it checks each byte to verify that it is going into a valid memory location for the currently selected device. If errors are detected they are reported to the operator. Generally, this is just what we want because if, by some mistake, the program doesn't fit in the device memory map the program won't run properly. But, what if all we want to do is load an object code file into the buffer, perhaps edit it, and then save it in a different file or file format? Having this normally nice feature turned on might prevent us from doing this simple thing. To solve this problem, check the “Turn Off Memory Map Checks” option.

6.3.6 Help Menu

Contents



Selecting “Help – Contents” opens the WP11 help file.

About

Selecting “Help – About” opens the WP11 About screen where information about the program version number and our company contact information can be found.

6.4 Chip Operation Buttons

Initialize Device



WP11 uses the HC11 special bootstrap mode to program devices. This requires that a small program be downloaded to the target device RAM for the purpose of controlling the programming process and communicating with the PC. The “Initialize Device” button performs this task. No programming operation on a chip can be done without initializing the chip. This is why all menu items and buttons that talk to the programming hardware are disabled until the chip is initialized.

The small programs that “Initialize Device” downloads to the target RAM ***must*** be in the same directory as the WP11.EXE program so that they can be found when needed. The programs are different for each family member and have names such as “P11_E9.MIK” for the

MC68HC711E9 chip and “P11_K4.MIK” for the MC68HC711K4 chip. If WP11 can't find these files an error message is generated.

When the “Initialize Device” button is pressed a dialog box is displayed that has a progress gauge to show the status of the operation. If the operation is successful the dialog box just goes away and the appropriate menu items and buttons are enabled. If the operation is not successful an error message is generated.

Normally this initialization process has to be performed only once when a target device is inserted into the programming hardware. It does *not* have to be repeated unless power is turned off to the target device or the target device is reset.

Communications Check



This button is used to determine if the target device is properly initialized. The PC sends \$00 to the programming hardware and expects to receive \$AA followed by \$55. If the expected response is received then the appropriate menu items and buttons are enabled else they are disabled.

The “Communications Test” button can be used to enable the chip operations buttons when you know that the target device is properly initialized but the buttons are not enabled such as would be the case if you exited and re-entered the program. Of course, you could use the “Initialize Device” to do the same thing but it would take much longer.

Blank Check



Press this button to test the target device's CONFIG register, EPROM, and EEPROM memory to determine if every location is in the erased state.

Erase EEPROM and Config Reg



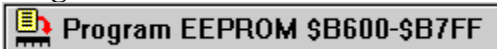
This button returns the EEPROM and CONFIG register in the target device to its erased state.

Program CONFIG Reg



Pressing this button programs the value shown for the CONFIG register in the WP11 status panel into the CONFIG register of the target device then tests to determine if the operation was successful. This button will not be enabled if the currently selected device does not have a CONFIG register.

Program EEPROM



Pressing this button programs the contents of the EEPROM portion of the WP11 buffer into the EEPROM memory of the target device then tests to determine if the operation was successful.

Program EPROM



Pressing this button programs the contents of the EPROM portion of the WP11 buffer into the EPROM memory of the target device then tests to determine if the operation was successful.

Program Entire Device



Pressing this button programs the contents of the EEPROM & EPROM portion of the WP11 buffer and the value shown for the CONFIG register on the WP11 status panel into the target device then tests to determine if the operation was successful.

7.0 TUTORIAL EXAMPLE SESSION

Working through this tutorial will take just a short time and will provide you with an overview of WP11 by actually using it.

It is assumed that you have read the rest of this manual, installed the software in accordance with the *Getting Started* section and are certain that your hardware and serial port connection are working properly. It is further assumed that you have a 68HC11 chip that can be used with this tutorial. We will only program the EEPROM memory so the chip can be erased and re-used for another purpose.

Three demonstration/test 68HC11 programs were included for use with this tutorial. If your chip is a MC68HC711K4 use K4EEPROM.S19 as the test file. If your chip is a MC68HC811E2 use E2_TEST.S19 as the test file. Use EEPROM.S19 as the test file for all other chips.

The steps required to program a chip are as follows:

- 1) Select the desired device type from the *Select Device* menu.
- 2) Insert the target device into the proper socket of your programming hardware. **NOTE:** You must insert and remove devices from the ZIF sockets with power removed from the programming hardware.
- 3) Load the *Buffer* with the data that you want to program into the target device. This can be done in a number of ways, from a file, from another chip, manually etc.
- 4) Initialize the target device for programming.
- 5) Program the target device using one of the available *chip operations* buttons.

Tutorial:

- 1) Load the Buffer with the desired data to be programmed into the target device by pressing the *Load From File* button on the WP11 main screen. From the Open dialog box open the test file for your chip as determined above.
- 2) Press the *Edit* button and use the scroll bars to view the data. Exit the screen by pressing the *Cancel* button.
- 3) Place the target device in the programming socket. **NOTE:** You must insert and remove devices from the programming socket with power removed from the programming board.

- 4) Press the *Initialize Device* button. This downloads a small program to the target device that controls the programming process.
- 5) Press the *Program & Verify Device* button if using an E2 device or *Program & Verify EEPROM* button if using any other device. This programs the contents of the *Buffer* into the target device.
- 6) To prove that we can read the contents of a programmed device, press the *Clear* button to clear the buffer, verify this with the edit button if you are the untrusting type, then press the *Load From Chip* button. Then press the *Edit* button. You should see the data that was originally programmed into the device.

That's it! You have just performed all of the necessary steps to program an HC11 device.

We sincerely hope that after using this tool, you will consider it to be a sound investment. We look forward to your comments and suggestions and to providing any assistance with their use that you may require.