P11-WD User's Manual

68HC11 Microcontroller Programming Software

Products Described

WP11.EXE – 68HC11 Bootloader Program for Windows
P11.EXE – 68HC11 Bootloader Program for DOS

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.

Web Site at www.tec-i.com
Email support@tec-i.com

Printed in the U.S.A.

TABLE OF CONTENTS

1.0 INTRODUCTION

This manual provides information on our WP11.EXE(for Windows) and P11.EXE(for DOS) programs, software used for programming 68HC11 family microcontrollers.

These programs take input from object code files created with 68HC11 assemblers or compilers and provide a means of programming or burning this data into the memory of 68HC11 family microcontrollers. The 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 PC capable of running Windows 3.1, 95, 98, ME, NT, 2000 or XP. The PC must have one free serial port designated as COM1-COM4.
P11.EXE requires a minimally configured PC capable of running DOS and one free serial port designated as COM1-COM2. P11.EXE will not run under Windows 2000 or Windows XP but will run under the other versions of Windows.

3.0 DEVICES SUPPORTED

Presently, these programs support 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 .ZIP file called P11WD.ZIP. This is a standard ZIP file that can be unzipped with PKUNZIP, WINZIP, ZIPMAGIC and many others. You may have downloaded the P11WD.ZIP file from our web site at www.tec-i.com or received the file on a

single 3.5" floppy diskette. To install P11-WD just unzip the files into the directory of choice on your PC.
The files extracted from P11WD.ZIP are:

P11.EXE — This is the DOS version of the program.

P11.HLP — This is the help file for P11.EXE.

WP11.EXE — This is the Windows program used to control the programming process.

WP11.HLP — This is the help file for WP11.EXE.

P11-WDManual.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.

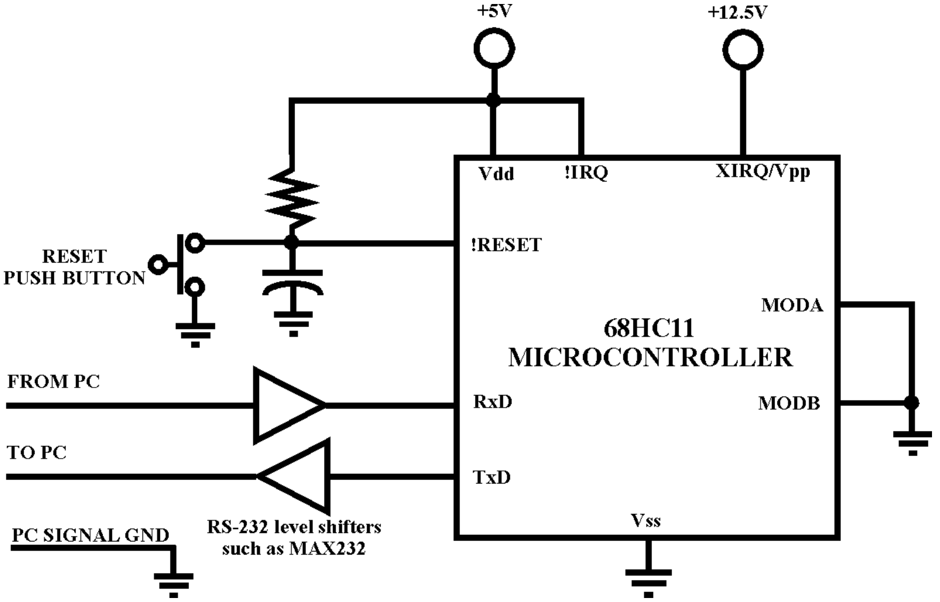
E2_TEST.S19 This is an object code file in Motorola S-Record format. It is used for test programming of MC68HC811E2 devices.

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.EXE and P11.EXE use the 68HC11 special bootstrap mode to program devices. You will need to build or buy the circuitry required to use the special bootstrap mode. Our P11 programmer circuit board is an example of a commercial product that has this capability. 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.

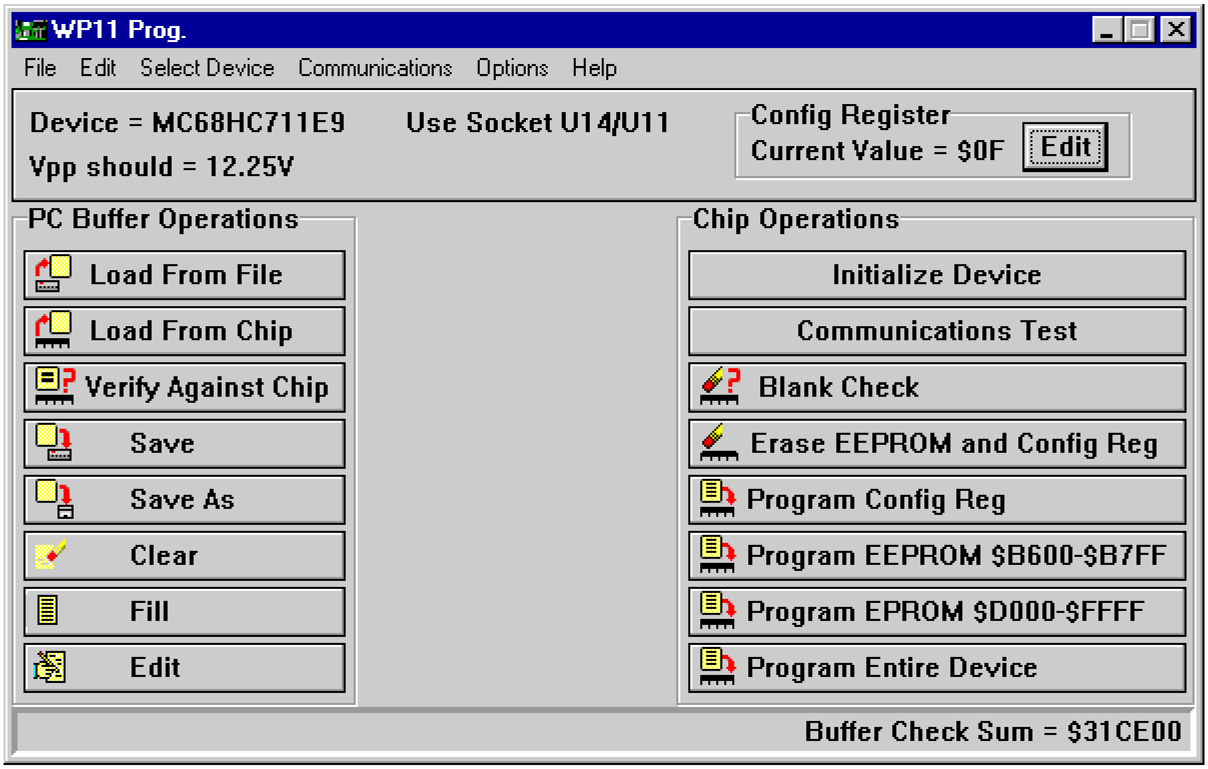
6.1 Program Overview

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



Device = MC68HC711E9 Use Socket U14/U11

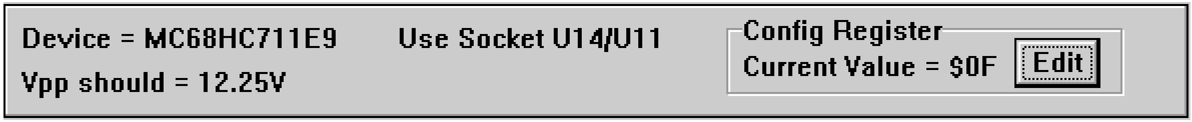
Vpp should = 12.25V

Current Value = $0F

Buffer Check Sum = $31CE00

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



Device = MC68HC711E9 Use Socket U14/U11

Vpp should = 12.25V

Config Register

Current Value = $0F

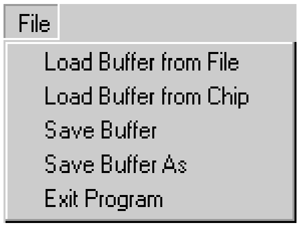
The status panel is used to show information about the currently selected device. It gets updated when a different HC11 device is chosen from the Select Device menu.

6.2.2 Changing the CONFIG Register

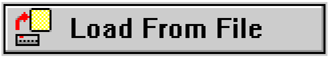
The Status Panel contains a means of changing the CONFIG register value. When the 68HC11 family member is changed the CONFIG Register value (in the PC) automatically changes to the erased state of the new device. You may, however, wish to change the CONFIG Register to some other value. Pressing the “EDIT” 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

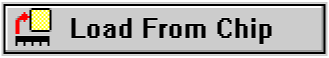
6.3.1 File Menu



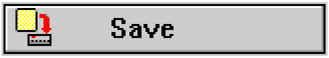
Load Buffer from File is the same as ✍



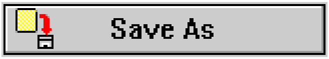
Load Buffer from Chip is the same as ✍



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.

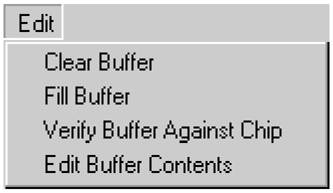
Load Buffer from Chip
Use this selection to load the contents of a programmed chip into the PC buffer.
A chip must be installed in the programming hardware and properly initialized for this menu option to be enabled. When this option is selected, a status Window with a Progress Gauge is displayed to 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.

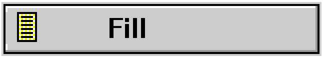
Save Buffer As
This selection is very nearly the same as the “Save Buffer” menu item described above except the “Save As” dialog box is opened immediately so that a new file name can be specified.

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:

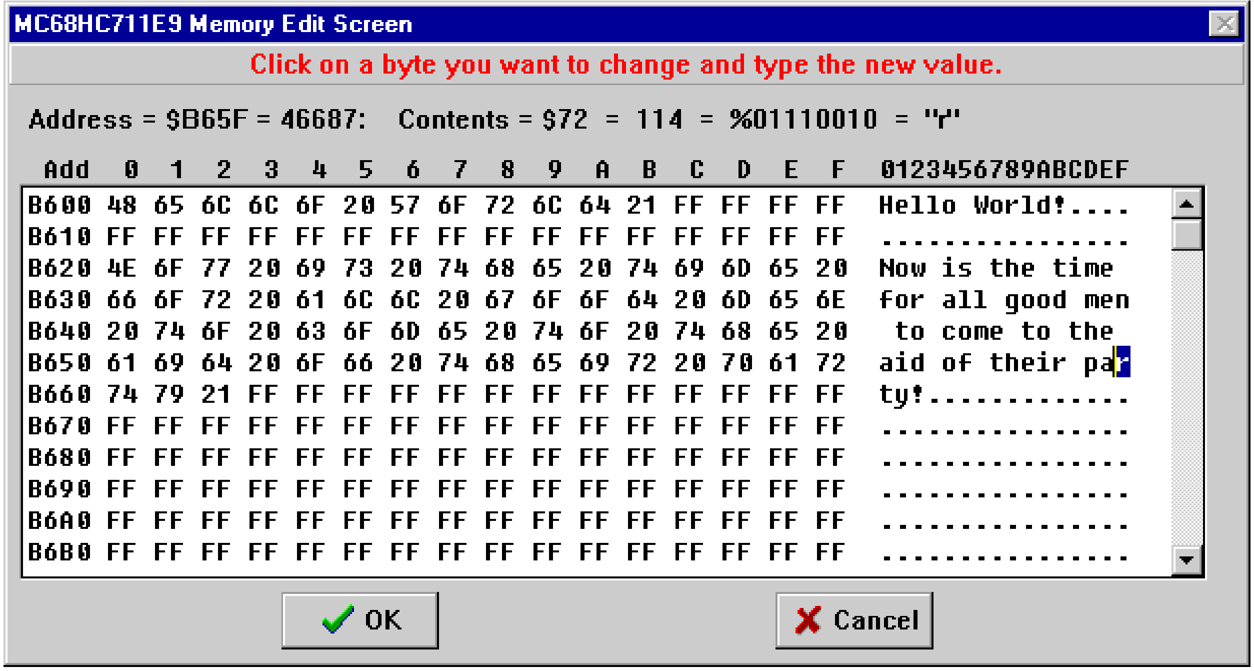
Suppose we want to have all the bytes in a device that are not taken up by our program set to a specific value. Say we want all unused memory locations filled with $3F, the op-code for the SWI instruction. To do this use the “Fill Buffer” function and specify $3F as the fill byte. Then use the “Load 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.



Click on a byte you want to change and type the new value.

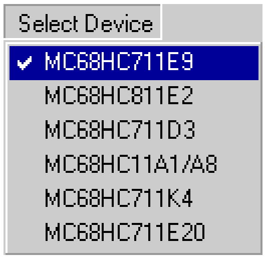
```
Address = $B65F = 46687:   Contents = $72  =  114  =  %01110010  = "r"

 Add   0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
B600  48 65 6C 6C 6F 20 57 6F 72 6C 64 21 FF FF FF FF   Hello World!....
B610  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
B620  4E 6F 77 20 69 73 20 74 68 65 20 74 69 6D 65 20   Now is the time 
B630  66 6F 72 20 61 6C 6C 20 67 6F 6F 64 20 6D 65 6E   for all good men
B640  20 74 6F 20 63 6F 6D 65 20 74 6F 20 74 68 65 20    to come to the 
B650  61 69 64 20 6F 66 20 74 68 65 69 72 20 70 61 72   aid of their par
B660  74 79 21 FF FF FF FF FF FF FF FF FF FF FF FF FF   ty!.............
B670  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
B680  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
B690  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
B6A0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
B6B0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
```

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



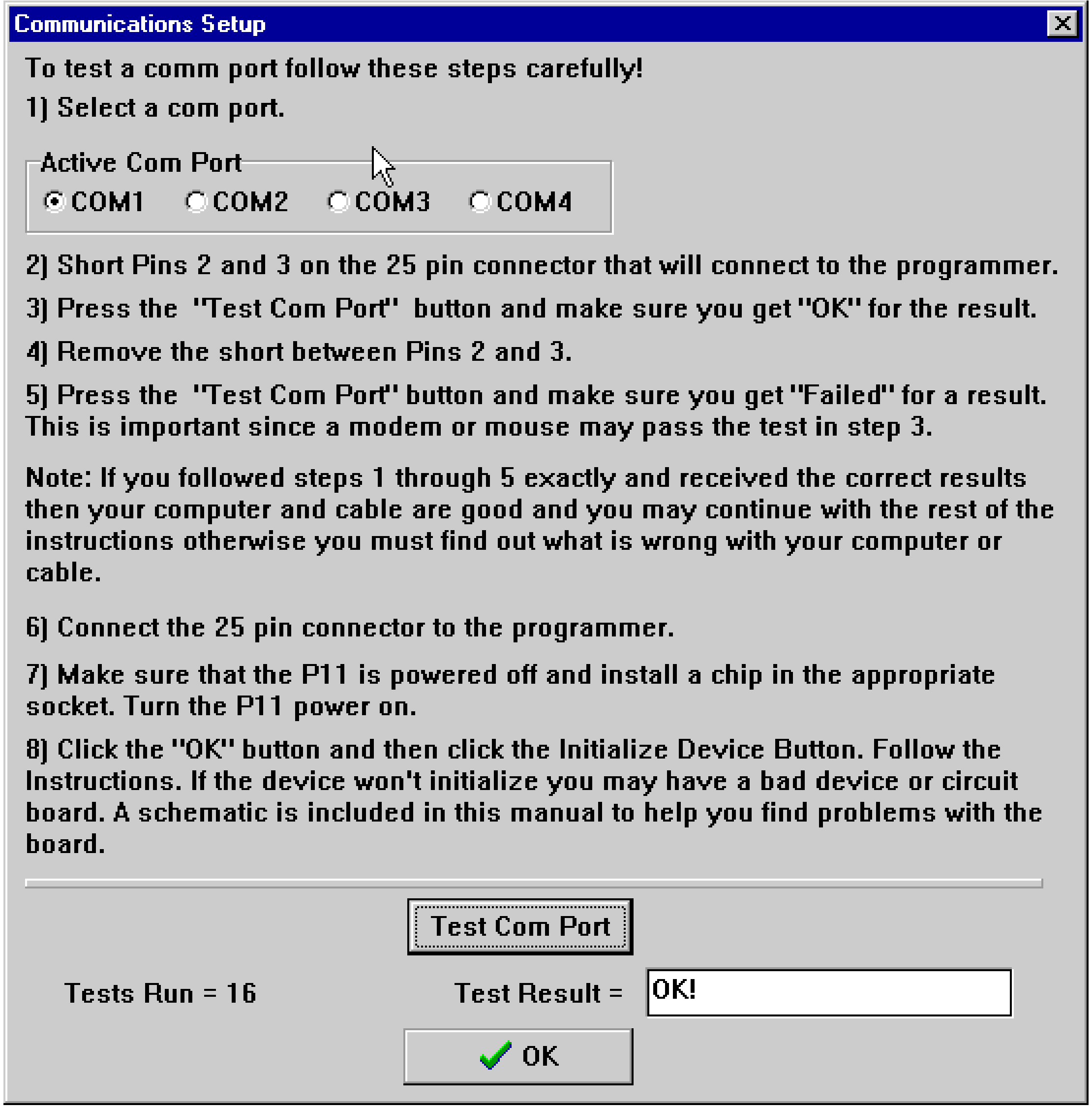
This menu is used to select the HC11 family member that you want to program. The part number is checked when it is the currently selected device. The part number is also displayed on the status panel with the “Device = XXXXXXXXXX” label.

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



Communications Setup

To test a comm port follow these steps carefully!

1) Select a com port.

Active Com Port

● COM1 ○ COM2 ○ COM3 ○ COM4

2) Short Pins 2 and 3 on the 25 pin connector that will connect to the programmer.

3) Press the "Test Com Port" button and make sure you get "OK" for the result.

4) Remove the short between Pins 2 and 3.

5) Press the "Test Com Port" button and make sure you get "Failed" for a result. This is important since a modem or mouse may pass the test in step 3.

Note: If you followed steps 1 through 5 exactly and received the correct results then your computer and cable are good and you may continue with the rest of the instructions otherwise you must find out what is wrong with your computer or cable.

6) Connect the 25 pin connector to the programmer.

7) Make sure that the P11 is powered off and install a chip in the appropriate socket. Turn the P11 power on.

8) Click the "OK" button and then click the Initialize Device Button. Follow the Instructions. If the device won't initialize you may have a bad device or circuit board. A schematic is included in this manual to help you find problems with the board.

Tests Run = 16

Test Result = OK!

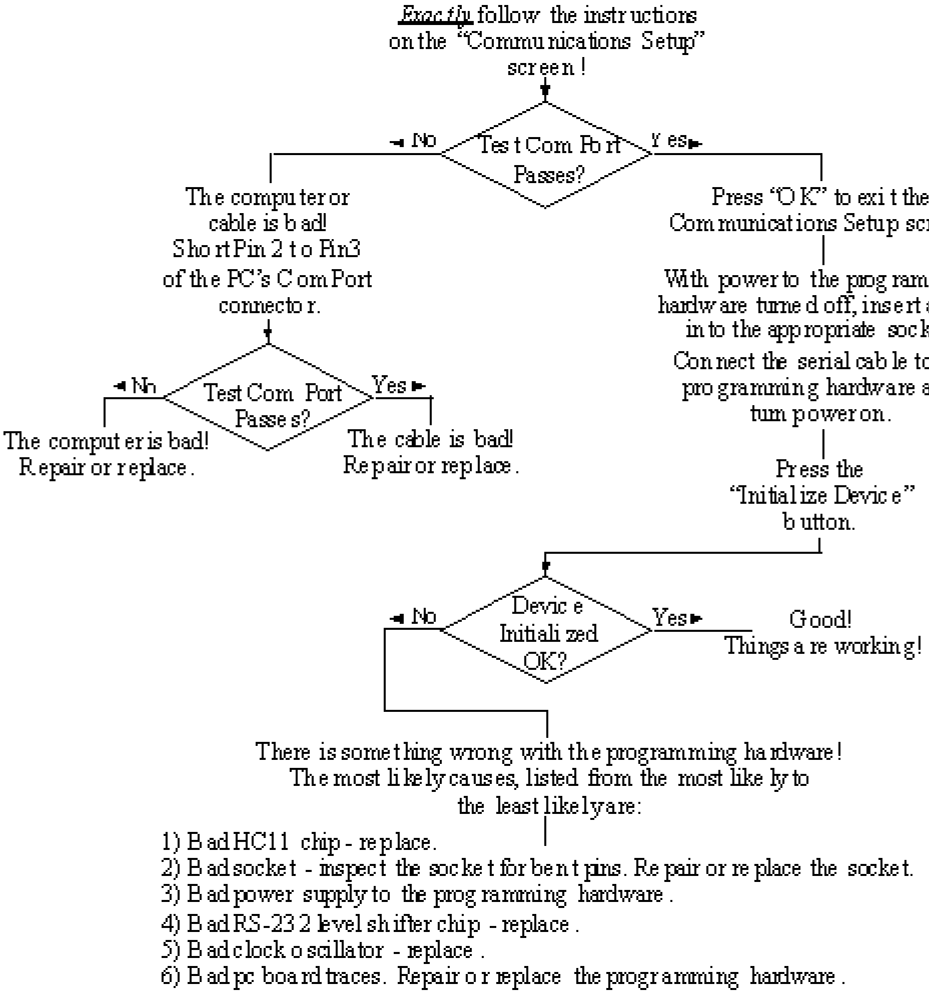
When the Com Port Setup and Test menu item is selected the Communications Setup screen shown above is displayed. The screen has detailed instructions and a test facility for determining whether or not a working serial port and cable are in use.

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

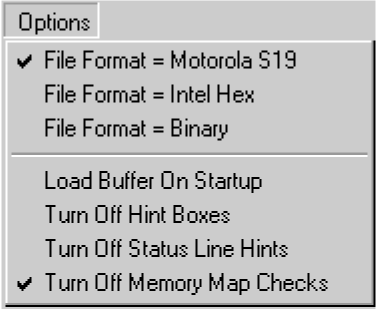
In order to perform any of the chip operations the computer must be able to communicate with the programming hardware. Our experience indicates that the most difficult task faced by a first time user is to establish this communication. This has to do with the non-standard nature of the EIA RS-232 Standard 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


Exactly follow the instructions on the "Communications Setup" screen!
Test Com Port Passes?
No
Yes
The computer or cable is bad! Short Pin 2 to Pin3 of the PC's Com Port connector.
Test Com Port Passes?
No
Yes
The computer is bad! Repair or replace.
The cable is bad! Repair or replace.
Press "OK" to exit the Communications Setup screen.
With power to the programming hardware turned off, insert a chip into the appropriate socket.
Connect the serial cable to the programming hardware and turn power on.
Press the "Initialize Device" button.
Device Initialized OK?
No
Yes
Good! Things are working!
There is something wrong with the programming hardware! The most likely causes, listed from the most likely to the least likely are:
1) Bad HC11 chip - replace.
2) Bad socket - inspect the socket for bent pins. Repair or replace the socket.
3) Bad power supply to the programming hardware.
4) Bad RS-232 level shifter chip - replace.
5) Bad clock oscillator - replace.
6) Bad pc board traces. Repair or replace the programming hardware.

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.

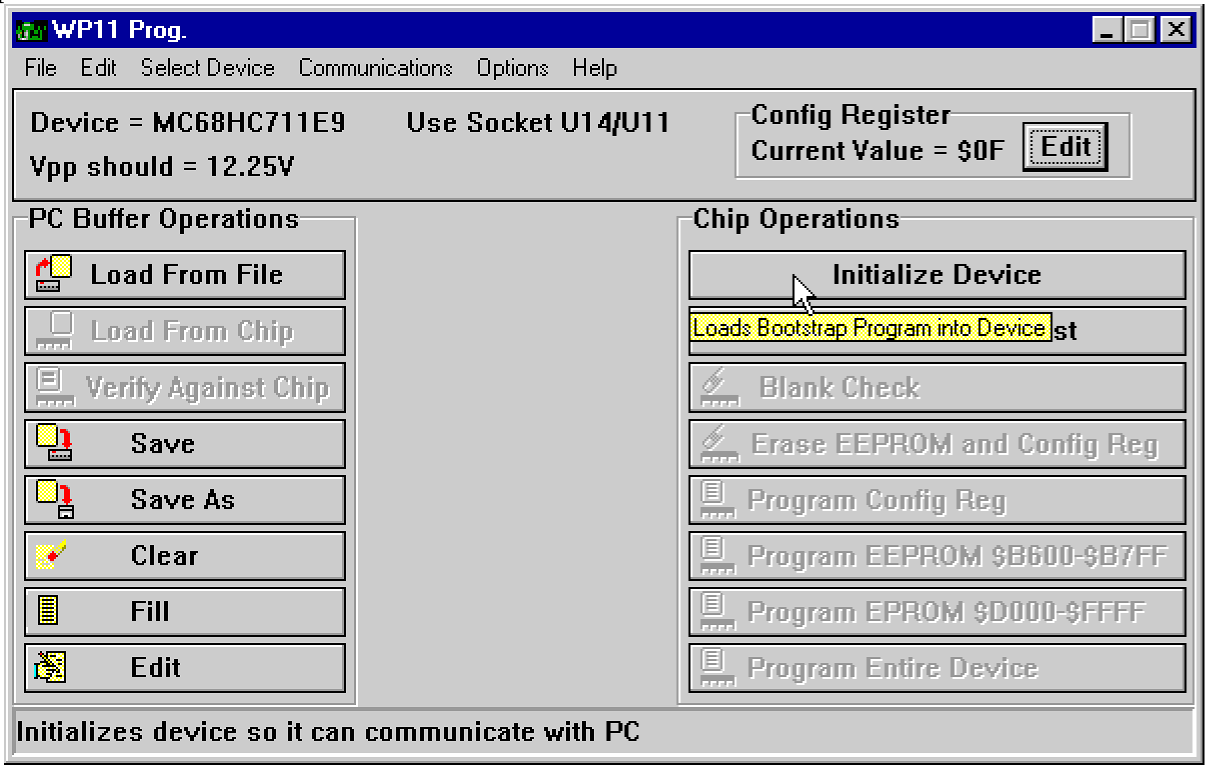
Note: When loading a file into the buffer the file format is automatically detected so these options come into play only when saving the buffer contents to a file.

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



Device = MC68HC711E9

Vpp should = 12.25V

Use Socket U14/U11

Current Value = $0F

Loads Bootstrap Program into Device

Initializes device so it can communicate with PC

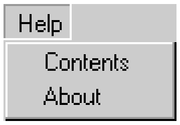
Turn Off Status Line Hints

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 button 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

Program EPROM $D000-$FFFF

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

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 WP11 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.

8.0 THE P11.EXE PROGRAM

All interactions with the programming hardware are accomplished by selecting the various options provided by the P11.EXE program.

8.1 Invoking P11.EXE

P11.EXE is invoked by typing at the DOS prompt:

```
>P11<Enter>
```

Assuming the use of a hard disk and the use of the PATH command in the AUTOEXEC.BAT you may invoke P11.EXE from any directory.

If there are multiple projects or multiple versions of the same project that require different settings, then you should create a different directory for each project or version. If your project or version requires the use of a different microcontroller, you should start P11.EXE from within the directory of that project or version because, when P11.EXE is exited, a file called P11.INI is created to save the current settings and this file is saved in the current directory. Whenever you invoke the program from that directory again, the P11.INI file determines the settings the program starts with. Any time P11.EXE is invoked from a directory that does not contain a P11.INI file, the default settings for P11.EXE (shown below in Section 6.2) will be in place.

Assuming the use of P11.EXE from a bootable working floppy disk with one project per disk, you may invoke P11.EXE from the A:\ or B:\ prompt, whichever contains the floppy disk..

8.1.1 Command Line Options

Command line options take precedence over settings in the P11.INI file.

From the Command Line the user may change the Communications Port that P11.EXE starts up with to either **COM1** or **COM2**. The default is COM1. The communication port setting is saved in the P11.INI file found in the directory that P11.EXE was last started up in.

For example:

```
C:\HC11PROJ\>p11 com1<Enter>
```

will start P11.EXE on COM Port 1. The COM port may also be set from the P11.EXE main menu.

8.1.2 Command Line Only Options

There are two program options of P11.EXE that can *only* be modified from the Command Line. These parameters are the **Video Mode** and the **Program Beep**.
For most circumstances the P11.EXE video is auto-sensing. However for some configurations, notably for computers with older color video cards that have pseudo color monitors attached, the video mode may be unacceptable.

BW80 ;sets the video mode to Black & White, 80 columns

Example:

```
C:\HC11PROJ\>p11 BW80<Enter>
```

will start P11.EXE in Black & White video mode. Video modes are not saved to the P11.INI file.

There is an optional **Program Beep** for those who wish it. It sounds whenever a task is completed. This option is saved in the P11.INI file.

It may be set by typing:

```
C:\HC11PROJ\>p11 beep<Enter>
```

when beginning a P11.EXE session. To return to silent operation, type:

```
C:\HC11PROJ\>p11 nobeep<Enter>
```

at the command line when beginning the next P11.EXE session.
Command line options may be in any order:

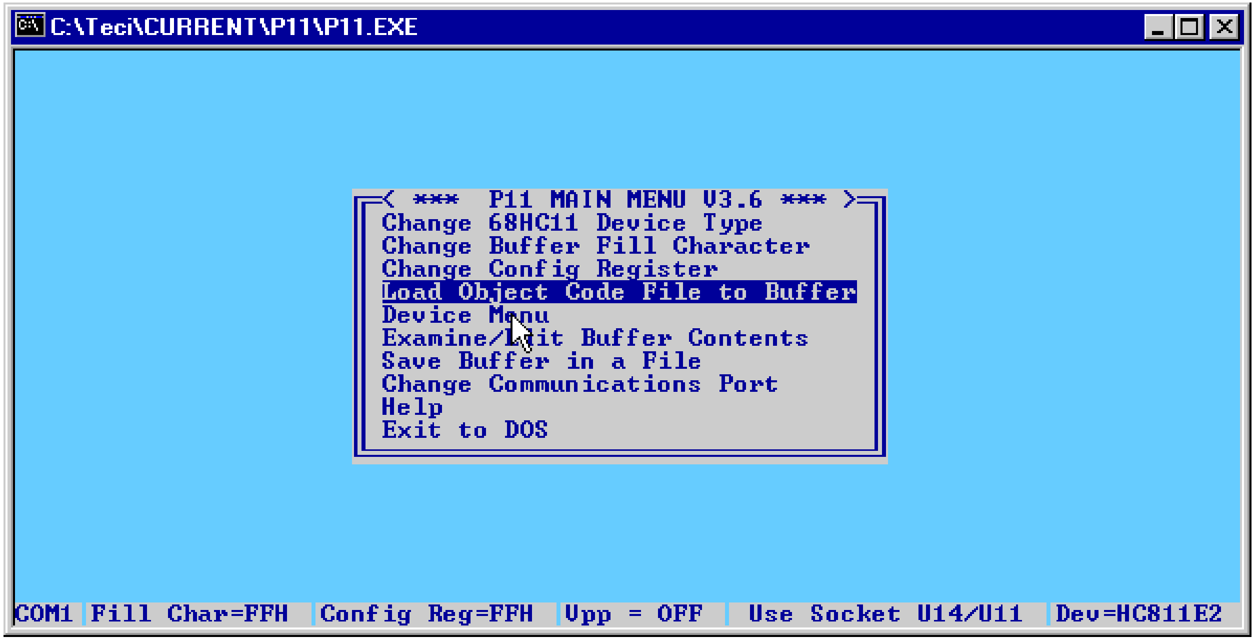
Example:

```
C:\HC11PROJ\>p11 nobeep BW80<Enter>
```

will turn the beep off and set the video to Black & White 80 columns.

8.2 P11.EXE Screen Display

When `P11.EXE` is invoked, the program menu screen is displayed. It looks something like this:



```
C:\Teci\CURRENT\P11\P11.EXE

< *** P11 MAIN MENU V3.6 *** >
Change 68HC11 Device Type
Change Buffer Fill Character
Change Config Register
Load Object Code File to Buffer
Device Menu
Examine/Edit Buffer Contents
Save Buffer in a File
Change Communications Port
Help
Exit to DOS

COM1 | Fill Char=FFH | Config Reg=FFH | Vpp = OFF | Use Socket U14/U11 | Dev=HC811E2
```

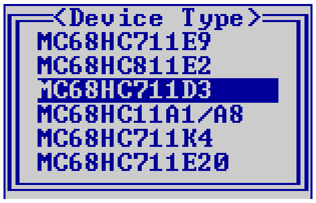
This is the default setting for the `P11.EXE`. Before describing each of the menu options some words on the general operation of the program are in order.

`P11.EXE` maintains an exact image of the microcontroller's memory in the PC. This memory image is called the *buffer*. During programming operations, data from the *buffer* is copied (programmed) into the memory of the microcontroller device. The *buffer* must be loaded with the desired data before a programming operation is performed. The *buffer* can be loaded from an object code file or from another microcontroller that already contains the desired data. The *buffer* can be displayed and edited before a programming operation is performed. With this general understanding, each of `P11.EXE`'s options will now be described.

8.3 P11.EXE Menu Options

Change Device Type

Selecting the **Change Device Type** option brings up the device type selection menu that looks like this:



```
<Device Type>
MC68HC711E9
MC68HC811E2
MC68HC711D3
MC68HC11A1/A8
MC68HC711K4
MC68HC711E20
```

With the up and down arrow keys, select the processor desired and press `<Enter>`.

* * * WARNING * * *

Changing the device type or even selecting the same device type as the current selection purges the contents of the buffer. The buffer is filled with the erased state of the selected device.

Change Buffer Fill Character

When you change the 68HC11 device type the buffer fill character automatically changes to the erased state of the selected device. You may, however, wish to change the buffer fill character to some other value. Pressing `<Enter>` at this **Main Menu** selection will enable you to specify a HEX value between 0H & 0FFH inclusive. Changing the buffer fill character will completely overwrite the buffer.

Change Config Register

When you change the 68HC11 device type the Config Register value (in the PC) automatically changes to the erased state of the selected device. You may, however, wish to change the Config Register to some other value. Pressing `<Enter>` at this **Main Menu** selection will enable you to specify a HEX value between 0H & 0FFH inclusive. You must be careful to specify values for the Config Register that are valid for the selected device. Changing the Config Register in the Main Menu will not effect the microcontroller until the Config Register is programmed from the Device Menu.

Load Object Code File to Buffer

This command is used to load an object code file into the host PC's RAM buffer.
When this option is selected, the operator is prompted to enter the file name of the object code file. The complete file name must be entered including the file extension. File formats are automatically detected by `P11.EXE`, and may be either 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 `P11.EXE`, a memory image of the target chip is created in the buffer. If the file can not 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 `<Esc>`. The operator is then given the option of exiting the screen.

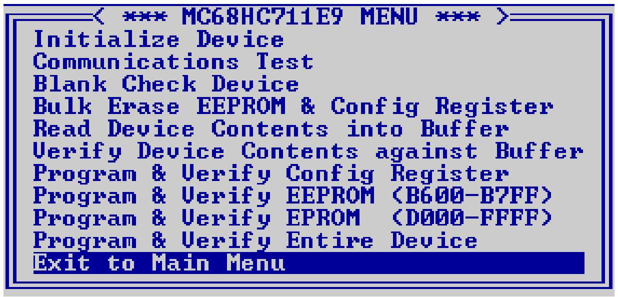
You may then **Examine/Edit Buffer Contents**. For a complete description of this option see below.

* * * WARNING * * *

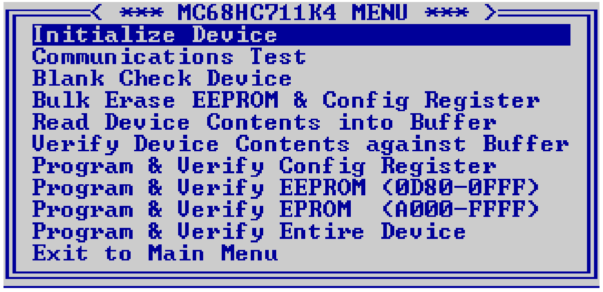
The order in which you use the commands from the main menu is very important. You should select the desired HC11 device before you fill the buffer with programming data. When you change the device type (or even select the same device type from the **Change 68HC11 Device Type** menu) the buffer is filled with the erased state of the selected device. Next if you wish to put a different fill character in the buffer you must do so before you **Load Object Code File to Buffer**. If you load the object code file first, then change the fill character, the fill character will completely overwrite the contents of the buffer erasing any information currently in it. After selecting the device type and changing the fill character, the object code file may be loaded.

Device Menu

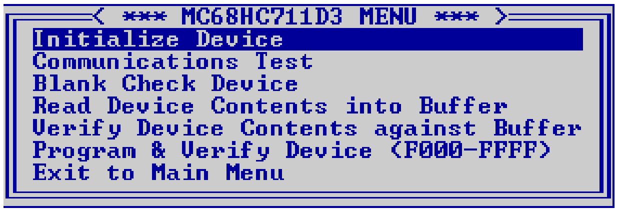
These sub menus let you select various options for each respective selected device.



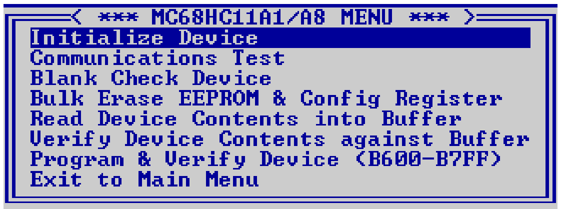
```
< *** MC68HC711E9 MENU *** >
Initialize Device
Communications Test
Blank Check Device
Bulk Erase EEPROM & Config Register
Read Device Contents into Buffer
Verify Device Contents against Buffer
Program & Verify Config Register
Program & Verify EEPROM (B600-B7FF)
Program & Verify EPROM  (D000-FFFF)
Program & Verify Entire Device
Exit to Main Menu
```



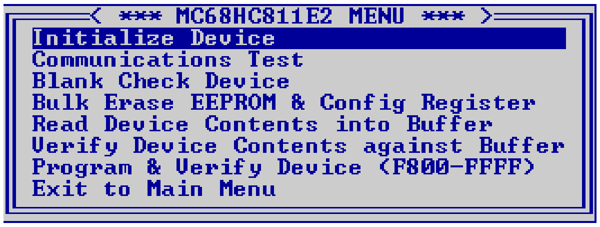
```
< *** MC68HC711K4 MENU *** >
Initialize Device
Communications Test
Blank Check Device
Bulk Erase EEPROM & Config Register
Read Device Contents into Buffer
Verify Device Contents against Buffer
Program & Verify Config Register
Program & Verify EEPROM (0D80-0FFF)
Program & Verify EPROM  (A000-FFFF)
Program & Verify Entire Device
Exit to Main Menu
```



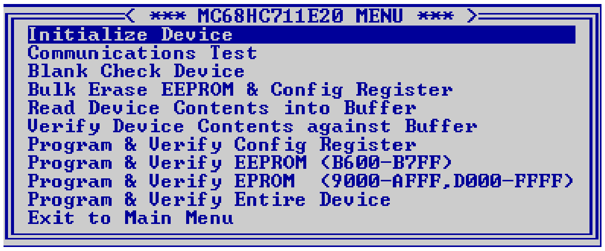
```
< *** MC68HC711D3 MENU *** >
Initialize Device
Communications Test
Blank Check Device
Read Device Contents into Buffer
Verify Device Contents against Buffer
Program & Verify Device (F000-FFFF)
Exit to Main Menu
```



```
< *** MC68HC11A1/A8 MENU *** >
Initialize Device
Communications Test
Blank Check Device
Bulk Erase EEPROM & Config Register
Read Device Contents into Buffer
Verify Device Contents against Buffer
Program & Verify Device (B600-B7FF)
Exit to Main Menu
```



```
< *** MC68HC811E2 MENU *** >
Initialize Device
Communications Test
Blank Check Device
Bulk Erase EEPROM & Config Register
Read Device Contents into Buffer
Verify Device Contents against Buffer
Program & Verify Device (F800-FFFF)
Exit to Main Menu
```



```
< *** MC68HC711E20 MENU *** >
Initialize Device
Communications Test
Blank Check Device
Bulk Erase EEPROM & Config Register
Read Device Contents into Buffer
Verify Device Contents against Buffer
Program & Verify Config Register
Program & Verify EEPROM (B600-B7FF)
Program & Verify EPROM  (9000-AFFF,D000-FFFF)
Program & Verify Entire Device
Exit to Main Menu
```

For each command option there is a text screen displayed on the host computer. The user must follow these instructions and wait for the completion of the command.

Initialize Device

P11 uses the HC11 special bootstrap mode during programming. This mode requires that special programming code be downloaded to the target device's RAM to control the programming process. The Initialize Device menu option performs this task. No chip may be programmed or communicated with successfully without first performing this menu function. Normally, this initialization process has to be performed only once after the target device is brought out of reset. It does *not* have to be repeated unless power is turned off to the target device.

Communications Test

This command is used to determine if the target device is properly initialized.

Blank Check Device

Tests the target device's Config Register, EPROM and EEPROM memory to determine if every location is in the erased state.

Bulk Erase EEPROM & Config Reg.
This menu command returns the EEPROM and Config Register in the target device to its erased state.

Read Device Contents into Buffer
This command reads the contents of the target device's programmable memory into the RAM buffer in the host computer.

Verify Device Contents against Buffer
This command compares the contents of the target device's programmable memory with the contents of the P11 buffer.

Program & Verify Config Register
This command programs the value of the Config Register shown on the P11 **status line** into the Config Register of the target device then tests to determine if the operation was successful.

Program & Verify EEPROM
This command programs the contents of the EEPROM portion of the P11 buffer into the EEPROM memory of the target device then tests to determine if the operation was successful.

Program & Verify EPROM
This command programs the contents of the EPROM portion of the P11 buffer into the EPROM memory of the target device then tests to determine if the operation was successful.

Program & Verify Entire Device
This command programs the contents of the EEPROM & EPROM portion of the P11 buffer and the value of the Config Register shown on the P11 **Status Line** into the target device then tests to determine if the operation was successful.

Exit to Main Menu
This option returns the user to the `P11.EXE` Main Menu.

This is the end of the description of the **Device Menu** sub menu. We will now return to a description of the remaining selections in the **Main Menu**.

Examine/Edit Buffer Contents
Selecting this option allows the user to examine and/or edit the contents of the P11 buffer.

If the user wishes, the buffer may be edited by typing over the current contents of memory locations. The cursor may be moved around the screen using the arrow keys. Pressing the TAB key toggles the cursor between the HEX portion of the screen and the ASCII portion of the screen. Non-HEX keys are disabled in the Hex portion of the screen, non-printable characters are disabled in the ASCII portion of the screen.

To move through the memory map in the buffer, use the Page Up and Page Down keys. Each page displays 256 bytes of information. The buffer varies in size according to the currently selected device.

Save Buffer in a File
This option allows you to save the current contents of the P11 buffer in one of three object code file formats, Motorola S-Record, Intel HEX or binary image format.

When you select this option, you are asked to name the new file. If you wish to save the file in a directory other than the current directory you must type the full path and file name.

Change Communications Port
This option toggles the active communications port between COM 1 and COM 2. The currently active port is displayed on the left side of the P11 status line.

Help
This option opens the help screen. The help screen may be scrolled through using the `Page Up` and `Page Down` keys or using the arrow keys. To exit the Help screen, press <`Esc`>.
The help screen information is contained in a standard DOS text file. The user may edit this file to add any additional information that may be required.

Exit to DOS
This menu selection will end the `P11.EXE` session. Pressing <End> will take you to this menu selection from anywhere else in the **Main Menu**.

9.0 P11 TUTORIAL EXAMPLE SESSION

It is assumed that you have read the rest of this manual, installed the software in accordance with the *Getting Started* section and performed the *Initial Test of the P11 Programmer* in accordance with that section of this manual. It is further assumed that you are in P11.EXE at the Main Menu level.
It is also assumed that you have a MC68HC711E9 or a MC68HC811E2 chip that can be used with this tutorial. If you are using an E9 chip, we are not going to program the EPROM section, we will only program the EEPROM section so the chip can be erased and re-used for another purpose.

The steps required to program a chip are as follows:
1) Select the desired device type from the *Change Device Type* option of the P11 Main Menu.
2) Configure the P11 Programming Board for the selected device. This means setting the jumpers correctly and inserting the target device into the proper socket.

* * * NOTE * * *

You must insert and remove devices from the ZIF sockets with power removed from the programming board. This means unplugging the wall transformer.....or.....better yet, plugging the transformer into a power strip that has a switch for turning the power on and off.

3) Load the *Buffer* with the data that you want to program into the target device. This is done with the *Load Object Code File into Buffer* option from the P11 Main Menu. Sometimes you may want to manually enter data into the buffer, this can be done with the *Examine/Edit Buffer Contents* option from the P11 Main Menu.
4) Initialize the target device for programming by using the *Initialize Device* option from the Device Menu. This places a control program in the target device's ram memory that allows *P11.EXE* to control the programming process.
5) Program the target device using one of the available options in the *Device Menu*.

Tutorial:

1. Load the Buffer with the desired data to be programmed into the target device by selecting the *Load Object Code File to Buffer* option from the P11 Main Menu.
2. Enter an object code file name, "TEST_E9.MIK" if your using an E9 or "TEST_E2.MIK" if your using an E2 target device. Press "Y" at the "Do you want to exit this from [Y/N]?" prompt.
3. Choose the *Examine/Edit Buffer Contents* P11 Main Menu option, and view the object code by using the Page Up and Page Down keys. (To edit the data, see the section *Examine/Edit Buffer Contents*) Exit the screen by pressing `<Esc>`.
4. Select the *Device Menu*. Press `<Enter>`.
5. Select the *Initialize Device* menu item. Press `<Enter>`. You will be confronted with a screen that shows the initialization progress. When that is complete you are free to select any of the remaining commands in the *Device Menu* except for the *Program & Verify EPROM* and *Program and Verify Entire Device* options**.** These two menu commands will program the EPROM of the 68HC711E9 with the buffer contents and prevent the EPROM from being programmed again.

6. Select the *Program & Verify EEPROM* if using an E9 device or *Program & Verify Device* if using an E2 device. Press <Enter>. This programs the contents of the *Buffer* into the target device.

7. To prove that we can read the contents of a programmed device, go back to the Main Menu and select *Change Device Type*. Select the device that you are working with. This clears the Buffer to the erased state of the selected device. Then go back to the *Device Menu* and select *Read Device Contents into Buffer*. Then go back to the Main Menu and select *Edit/Examine Buffer Contents*. 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.