

# WASM11

# USER'S MANUAL

A 68HC11 EDITOR/MACRO CROSS ASSEMBLER  
for  
Windows

The logo for TECI (The Engineers Collaborative, Inc.) features the letters "TECI" in a bold, sans-serif font. To the right of the letters, there are several horizontal lines that curve upwards and to the right, resembling a stylized wing or a signal waveform.  
*The Engineers Collaborative, Inc.*  
web [www.tec-i.com](http://www.tec-i.com)  
e-mail [info@tec-i.com](mailto:info@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](http://www.tec-i.com)  
Email support@tec-i.com  
All Rights Reserved

**TABLE OF CONTENTS**

**1.0 INTRODUCTION..... 4**

**2.0 DEVICES SUPPORTED..... 4**

**3.0 SYSTEM REQUIREMENTS ..... 4**

**4.0 GETTING STARTED..... 5**

**4.1 SOFTWARE INSTALLATION..... 5**

**4.2 UNINSTALLING WASM11 ..... 5**

**5.0 OVERVIEW..... 6**

**5.1 WASM11 PROGRAM OVERVIEW ..... 6**

**6.0 TUTORIAL ..... 8**

**6.1 USING WASM11..... 8**

**7.0 CONVERTING TASM11 PROJECTS TO WASM11 PROJECTS..... 9**

**APPENDIX A – TUTORIAL LISTING FILE ..... 10**

# WASM11 USER'S MANUAL

---

## 1.0 INTRODUCTION

WASM11 is a Windows hosted Text Editor / 68HC11 Macro Cross Assembler. This program is one of a set of tools provided by TECI to facilitate the development of 68HC11 single chip microcomputer applications.

*TECI* provides several other development tools for 68HC11 family microcontrollers such as our P11 programmer and TECICE-HC11 real time in-circuit emulator.

## 2.0 DEVICES SUPPORTED

WASM11 supports all 68HC11 family members.

## 3.0 SYSTEM REQUIREMENTS

This product requires an IBM PC or compatible running a Windows 3.1/95/98/ME/NT/2000/XP operating system. Minimum recommended system is at least a 486 class PC with 64MB ram, 4 MB of free hard disk space, one 3 1/2 inch diskette drive. A color monitor is recommended.

## 4.0 GETTING STARTED

### 4.1 SOFTWARE INSTALLATION

The software is contained in a single self-extracting .EXE file called wasm11setup.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 wasm11setup.exe file from our web site at [www.tec-i.com](http://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 wasm11setup.exe are:

- WASM11.EXE - Editor/Macro Assembler executable file.
- WASM11.HLP - Editor/Macro Assembler online help file.
- WASM11.INI - Editor/Macro Assembler initialization file.
- WASM11MANUAL.PDF - This manual in .PDF format.
- 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.

**\*\*\* Important Note \*\*\***

Refer to this file for examples of proper WASM11 Source Code Syntax.

- WTTDEMO.11P - Project file used in the tutorial.
- WTTDEMO.ASM – Assembly source code file used in the tutorial.
- UNINSTALL.EXE - Uninstall executable.
- INSTALL.LOG - Installation log file - required by the uninstall program.

**\*\*\* 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

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 wasm11setup.exe program.

## 5.0 OVERVIEW

### 5.1 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 over 200 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 file can be up to 16MB in size and several files can be grouped together and associated with one another in a project. 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 must be saved with the .11P file extension; and (2) the main source code file of the project must 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  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

The screenshot shows the WASM11 main screen with the following callouts:

- 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

The main window displays the following code:

```

temp_accum    rmb    1                ;
;
;Note: The last ram location, not counting the stack is at $BF.
;
power_on_reset org    rom                ;start of program memory
               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                       ;
    
```

The status bar at the bottom shows:

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

File tabs at the bottom are labeled:

- MCPM3B.ASM
- MCPM3B.LST
- MCPM3B.S19
- MCPM3B.MAP

Additional callouts for the status bar and file tabs:

- File Modified
- Column Number
- Line Number
- Insert/Overwrite
- File Lines
- File Size
- Current Time
- File Tabs - Click to change file.

## 6.0 TUTORIAL


The quickest way to get started using the WASM11 program is to use it to perform some common program development tasks. So please take a few minutes to work through this tutorial.

### 6.1 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  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 29 of file WTTDEMO.ASM . Also note that the cursor in the WTTDEMO.ASM file was placed at the beginning of line 29, 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 29 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  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  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.

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

## 7.0 CONVERTING TASM11 PROJECTS TO WASM11 PROJECTS

Before going any further in this section, it is very important that you complete the Tutorial and become familiar with the basics about WASM11. Having accomplished this, if you have developed project(s) in TECI’s TASM11 Cross Assembler Program, you may follow these steps to convert each project to WASM11.

- 1) Invoke WASM11 and open a New Project (File|New Project).
- 2) Click on the “Add File to Project” speed button. Find the directory your TASM11 files are in and add each file, one at a time.
- 3) Once you have added all of your files, use File|Save Project As to save the project.

Remember, (a) projects must be saved with the .11P file extension; and (b) the main source code file of the project must 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.

# WASM11 USER'S MANUAL

## APPENDIX A—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
**
*****
;
FFC0      irq_vectors      equ      $ffc0
;
;
0000      ram              equ      $0              ;start of ram area
F800      program         equ      $F800          ;start of program area
;
1000      registers      equ      $1000
1003      port_c          equ      registers + 3    ;
1007      port_c_dds      equ      registers + 7    ;
;
0000      count_up        org      ram              ;ram variables
0000      count_down      rmb      1              ;
0001      count_down      rmb      1              ;
;
F800      org             program                 ;start of program memory
F800 86FF      power_on_reset      ldaa     #11111111 ;program starts here after reset
F802 B71003      staa              port_c        ;initialize port A with 1's
F805 B71007      staa              port_c_dds     ;set port A to an output port
F808 7F0000      clr              count_up       ;clear
F80B 7F0001      clr              count_down     ;      the ram variables
F80E 731003      main_loop        com      port_c ;compliment port A
F811 7C0000      inc              count_up       ;increment count up
F814 8D02        bsr              decrement_byte ;increment a word
F816 20F6        bra              main_loop     ;loop forever
;
*****
**          ITEST.INC    4/30/97   RBJ          **
**
** An include file to the WTTDEMO program.
**
*****
;
F818 7A0001      decrement_byte    dec      count_down ;decrement the variable count_down
F81B 39          rts                ;and return
;
;
F81C 39          dummy_isr         rts                ;should never be called
;
FFC0      org             irq_vectors             ;interrupt service routine vectors
FFC0 F81C      dw              dummy_isr          ;reserved
FFC2 F81C      dw              dummy_isr          ;reserved 1
FFC4 F81C      dw              dummy_isr          ;reserved 2
FFC6 F81C      dw              dummy_isr          ;reserved 3
FFC8 F81C      dw              dummy_isr          ;reserved 4
FFCA F81C      dw              dummy_isr          ;reserved 5
FFCC F81C      dw              dummy_isr          ;reserved 6
FFCE F81C      dw              dummy_isr          ;reserved 7
FFD0 F81C      dw              dummy_isr          ;reserved 8
FFD2 F81C      dw              dummy_isr          ;reserved 9
FFD4 F81C      dw              dummy_isr          ;reserved 10
FFD6 F81C      dw              dummy_isr          ;sci serial system
FFD8 F81C      dw              dummy_isr          ;spi serial transfer complete
FFDA F81C      dw              dummy_isr          ;pulse accumulator input edge
FFDC F81C      dw              dummy_isr          ;pulse accumulator overflow
FFDE F81C      dw              dummy_isr          ;timer overflow
FFE0 F81C      dw              dummy_isr          ;timer output compare 5
FFE2 F81C      dw              dummy_isr          ;timer output compare 4
FFE4 F81C      dw              dummy_isr          ;timer output compare 3
FFE6 F81C      dw              dummy_isr          ;timer output compare 2
```

# WASM11 USER'S MANUAL

---

```
FFE8 F81C      dw      dummy_isr      ;timer output compare 1
FFEA F81C      dw      dummy_isr      ;timer input capture 3
FFEC F81C      dw      dummy_isr      ;timer input capture 2
FFEE F81C      dw      dummy_isr      ;timer input capture 1
FFF0 F81C      dw      dummy_isr      ;real time interrupt
FFF2 F81C      dw      dummy_isr      ;!IRQ (external pin or parallel I/O)
FFF4 F81C      dw      dummy_isr      ;!xirq pin
FFF6 F81C      dw      dummy_isr      ;SWI
FFF8 F81C      dw      dummy_isr      ;illegal opcode trap
FFFA F81C      dw      dummy_isr      ;COP failure (reset)
FFFC F81C      dw      dummy_isr      ;COP clock monitor fail
FFFE F800      dw      power_on_reset ;!RESET
```

\*\*\* Symbol Table \*\*\*

```
count_down      0001
count_up        0000
decrement_byte  F818
dummy_isr       F81C
irq_vectors     FFC0
main_loop       F80E
port_c          1003
port_c_ddr     1007
power_on_reset  F800
program         F800
ram             0000
registers       1000
```