Hitchhiker's Guide to CodeWarrior
EE371, EE475
Fall 2005

Building an HC12 executable relocatable assembly or C program:

1. Launch CodeWarrior IDE.
   a) From Windows desktop, click **Start > Programs > Metrowerks CodeWarrior > CW12 V3.1 > CodeWarrior IDE.**
2. **Create New Project.**
   a) From IDE main menu bar, select **File > New** New window appears.
   b) Highlight **HC(S)12 New Project Wizard.**
   c) Enter a **Project Name** that makes sense to you (like lab_x)**.**
   d) Enter a **Location** for the project. This should be on your Z drive or wherever you want to keep your HC12 projects. (Don't keep it on the C:/ drive.)
   e) Click **OK**.
3. In the **New Project Wizard**

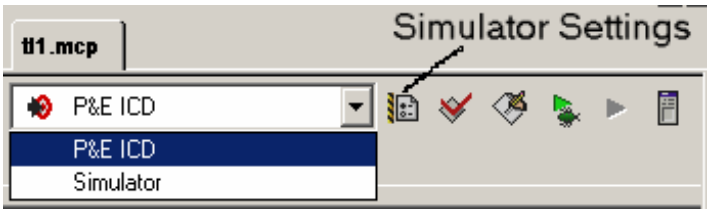
Page 1: Select **MC9S12C32** as the derivative you want to use and click **Next**.
Page 2: Check the language support you want (**Assembly** or **C**) and click **Next**.

If **Assembly**:
Page 3: Check **Relocatable Assembly** and click **Next**.
Page 4: Check **Metrowerks Full Chip Simulator** and **P&E Hardware Debugging** and click **Finish**.

If **C**:
Page 3: Check **No** for **Processor Expert**.
Page 4: Check **No** for **PC-lint.**
Page 5: Select **None** for **Floating Point Format**.
Page 6: Select **Small Memory Model**.
Page 7: Check **Metrowerks Full Chip Simulator** and **P&E Hardware Debugging** and click **Finish**.

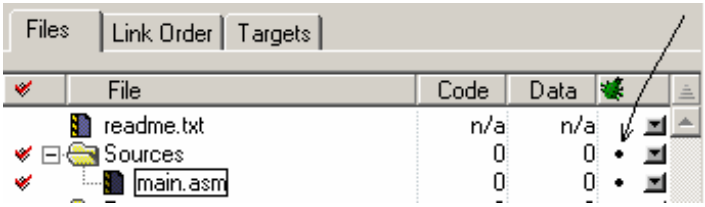
In the **Project Manager** Window

4. Switch to the **P&E ICD** for debugging by clicking the pull down arrow and selecting **P&E ICD**. (This uses the hardware for debugging. If you want to use the simulator, choose **Simulator**.)



5. If you want to create listing files Open **Simulator Settings** panel clicking the icon shown above.
   a) Click on + next to **Target** in the **Target Settings.**
   b) Highlight **Assembler for HC12** or **Compiler for HC12** and in that panel click on **Options**
      i) Check **Generate a listing file**.
      ii) Click **OK**.

6. Open the **Sources** folder (click on the + if it is not open).
7. Double click on **main.asm** or **main.c** and edit it to be the source file you want.
8. Save it to whatever name you want: **File > Save As**. (Make sure you give it an .asm filename, or you can just use the main.asm file if you want.)
9. Check to make sure debugging information is going to be generated.
   a) In the **Project Manager Files** window, check to see that there is a • in the rightmost column under the bug icon.



10. Assemble the file to check for errors:
    a) Click on **Project > Compile**
    b) Correct any errors
11. Have a look at your list file.
    a) **File > Open** and position to the **bin** folder in your project and open the **.lst** file.
12. Add other source files to the relocatable files in the project:
    a) Click on the **Add New Text File** icon.

    b) Edit the file
    c) **File > Save As** (use .asm as the extension for assembler file)
    d) Add it to the project: **Project > Add** *.asm to the project. In **Add Files** check both **Simulator** and **Monitor**.
    e) Compile it: **Project > Compile**
13. View the linker parameter file
    a) Click the + next to **Prm** and double click the P&E_ICD_linker.prm file to open it in the editor.
    b) For assembly applications make sure the default parameter file looks like this: (Note: The linker is CASE SENSITIVE. Things that are shown capitalized must be capitalized.)

```
NAMES
END
SEGMENTS
    RAM = READ_WRITE 0x0800 TO 0x0FFF;
    /* unbanked FLASH ROM */
    ROM_4000 = READ_ONLY  0x4000 TO 0x7FFF;
    ROM_C000 = READ_ONLY  0xC000 TO 0xFEFF;
    /* banked FLASH ROM */
/*  PAGE_3E = READ_ONLY  0x3E8000 TO 0x3EBFFF; not used: equivalent to ROM_4000 */
/*  PAGE_3F = READ_ONLY  0x3F8000 TO 0x3FBFFF; not used: equivalent to ROM_C000 */
END

PLACEMENT
    _PRESTART,            /* Used in HIWARE format: jump to _Startup at the code
start */
```

```
    STARTUP,               /* startup data structures */
    ROM_VAR,               /* constant variables */
    STRINGS,               /* string literals */
    VIRTUAL_TABLE_SEGMENT, /* C++ virtual table segment */
    NON_BANKED,            /* runtime routines which must not be banked */
    DEFAULT_ROM,
    COPY                   /* copy down information: how to initialize variables */
                           /* in case you want to use ROM_4000 here as well, make sure
                              that all files (incl. library files) are compiled with
the
                              option: -OnB=b */
                           INTO  ROM_C000/*, ROM_4000*/;
    DEFAULT_RAM            INTO  RAM;
END

STACKSIZE 0x100

//VECTOR 0 _Startup /* reset vector: default entry point for a C/C++ application. */
VECTOR 0 Entry  /* reset vector: default entry point for a Assembly application. */
INIT Entry      /* initialisation entry point for assembly applications */
```

c) For C applications, the vector links at the bottom of the file should be:

```
VECTOR 0 _Startup /* reset vector: Default entry point for a C/C++ application. */
//VECTOR 0 Entry  /* reset vector: Default entry point for a Assembly application. */
//INIT Entry      /* This is used for assembly applications also */
```

14. Make the project.
    a) **Project > Make** or click on the icon

    b) Correct any linking errors.

15. Inspect the Linker Map file.
    a) Click the + on **Linker Map** and open **P&E_ICD.map**
    b) Check the **OBJECT-ALLOCATION SECTION** section to see where the **Entry** label is located. This will be the starting address for your program.

16. **Running the program**.
    a) Make sure the SLK board is powered up, the CSM-12C32 is in the proper place, and that the USB cable is connected.
    b) Launch the True-Time Simulator and Debugger by **Project > Debug**, **F5**, or clicking on the icon

When the loader warning appears click **OK**.

True-Time Simulator and P&E ICD target:

17. Running the program using CodeWarrior and the P&E ICD target.
    a) Make sure the SLK board is powered up, the CSM-12C32 is in the proper place, and that the USB cable is connected.
    b) If your program is not loaded or it needs to be reloaded, click **ICD12 > Load** and load **bin\P&E_ICD.abs**.
    c) You can use all the features of the debugger on the actual code on the microcontroller.

True-Time Simulator and Simulator Target:

18. Run the program using CodeWarrior True-Time Simulator target.
    a) Choose **Simulator** instead of P&E_ICD as shown in step 4.
    b) If Simulator is not the debug target, click on **Component > Set Target** and chose **Simulator Target Interface** and click OK.
    c) If your program has not been loaded, or you want to reload it, click **Simulator > Load** and load **bin\simulator.abs**.
    d) You can use all the features of the simulator on the code.

CodeWarrior File Handling

Path Display:

The full path of the file currently in the editor window is shown at the top of the window. If it is not displayed, click the little box in the upper right-hand corner of the window, just above the right margin scroll bars.

Adding Files To Your Project:

Project -> Add Files . . .

This does NOT copy a file to your project. Instead it creates a link to the file you add. When you do this you get the following message:

The following access path has been added to target "Simulator" of project "test4.mcp": {Project}..\blinky_1\Sources

Any editing changes you make will be made in the original file. Do not use Project -> Add Files to copy a file from some other project to your project (unless you want to update a file in use for all projects using it.)

File -> Open

This allows you to position to a directory and open a file in that directory. The editor path display should show the original path. If you File -> Save, Editing changes take place in that file.

File -> Save As

This allows you to save the open file in whatever directory you want, including your current project. The default is the directory that was opened for the file originally. You can position to the sources directory of the current project and save it there.

Copying Files from One Project to Another using Explorer

You can use Explorer to copy files from one directory to another. This is probably the easiest way to add a file from some other project to your current project.

Project -> Add … to Project

After saving or copying a file to your project sources directory, you must activate **Project -> Add <filename> to Project**. If it is a source file, the colors will change appropriately.

File -> Find and Open File

This command allows you to highlight a filename in the editor, e.g. switches.inc, and the system will find the file and open it in the editor.

File -> Save a Copy As . . .

Use this to save a copy of your file as a backup. The filename will be "Copy of …".

File -> Revert
This will remove any editing changes made since the last time you saved the file.

Debugging Information:
Warning, no debug information
If you get this error when launching the debugger, it means one of the files you have copied into your project does not have the debug information being generated when it is being compiled. To fix this, double click in the column under the little green bug in the files window. See Step 9.

Removing Files from a Project:
If you choose to remove a file from the project, either by highlighting it and pressing **Del** or **Edit > Remove** you get an error message saying removing the file from the project cannot be undone. Rubbish! This does not actually "remove", as in delete, the source file. It simply removes it from the list of files in the project. You can always go back and add it back in later.

F:\1wpdocs\Book\Revision\CodeWarrior\CW_hitchhikers_f05.doc