

# How to Compile with a Mixture of Assembly and C Files Using ImageCraft's ICC12

## Application Note AN0003

---

Author : Lin Zhao

Date : August 12, 2003

Abstract: This application note shows you how to compile with a mixture of assembly and C files by using ImageCraft's ICC12. The programs for this application note were tested on the DRAGON12 and MiniDRAGON+ evaluation boards in EVB Mode. They should also work on most HC12 and HCS12 boards.

---

There are mainly two approaches to writing assembly language with ICC12. One of the method is to insert a single assembly instruction directly into a C function using *asm("string");*. Another method is to insert an entire assembly file into a C program using *asm(".include 'filename'");*. In this application note, we will concentrate on how to insert an entire assembly file into a C file.

In the example C file as shown in Listing 1, an assembly file, "ex.s", was embedded using *asm(".include 'ex2.s'");*. In the main program, a message "Calling Assembly Subroutine..." was printed in the terminal window before calling a function defined in the assembly file. Then, the "start()" function that was defined in the assembly file was called to start a counter, which will be explained in "ex.s" assembly file. Notice that the assembly function "start()" does not need a prototype in the C program.

**Listing 1:** The example C program with an assembly file embedded.

```
/*
 * assem_c.c
 * Author: Lin zhao
 * Date: 8/10/2003
 * Description: Sample file for C file embeds an assembly language file.
 */

#include <hcs12dp256.h>

asm(".include 'ex2.s'"); // insert assembly here

main()
{
    setbaud(BAUD9600);
    while(1) {
        puts("Calling Assembly subroutine...\n\r");
        start(); // call to assembly subroutine
    }
}
```

The assembly program, “ex.s”, that was embedded in the above C program is shown in Listing 2. This assembly program decreases a counter loaded with 0x0F and displays the content of the counter on the LEDs on Port B. Once the content of the counter reaches zero, the assembly program returns to the C program.

The *.area text* is the standard place for programs (in ROM). Assembly level function “**start()**” was defined beginning with an underscore, “\_”. Notice that in the assembly file the name has an underscore, but the same name in the C file does not.

**Listing 2:** Assembly file “ex.s” embedded in the example C file

```
.area text
portb = 0x1
REGBLK = 0x0000
SPEED = 0xffff

_start: ldab #255
        stab 0x3
        clr 0x1

        ldx #REGBLK
        ldaa #0x0f
back:   staa portb,x
        ldab #20
wait:   jsr delay
        decb
        bne wait
        deca
        bne back

delay:  ldy #SPEED
dly:   dey
        bne dly
        rts
```

Notice that the assembly program “ex.s” and the C program “assemcall.c” must be put in the same folder before you create your project and have these files compiled. You also must put “putchar\_dp256.c” in your folder. Refer to the Application Note, *Run “Hello World” on DRAGON12 and MiniDRAGON+ Using ImageCraft’s ICC12 and D-Bug12*, for how to use ICC11 C compiler.

Refer to Jonathan W. Valvano’s “*Developing Embedded Software in C Using ICC11/ICC12/Hiware*”[1] for more information about assembly programming in ICC11 and ICC12.

## Reference:

1. Jonathan W. Valvano. “*Developing Embedded Software in C Using ICC11/ICC12/Hiware*”. <<http://www.ece.utexas.edu/~valvano/embed/toc1.htm>>