

# DESIGNER'S NOTEBOOK



## Bootload of C Code for the TMS320C5x

Contributed by Jason Chyan

**Design Problem**      How can I generate my boot code with C?

**Solution**      Use the `-c` (not `-cr`) option in the linker and build a single section that includes the `.text`, `.cinit`, `.bss`, etc., sections that you want to be in the boot code. Then use DSPHEX to convert this single section into boot code. Following is an example linker command file to link several sections and `.cinit` into one output section.

```
-c
-o filename.out
-m filename.map
filename.obj
-stack 64
-l rts50.lib
-l flib50.lib

MEMORY
{
  PAGE 0: PROG: origin = 0x0800,  length = 0x1a00
  PAGE 1: DATA: origin = 0x0060,  length = 0x0020
}
SECTIONS
{
  bootsect: {
    rts50.lib(.text) = 0800h
    *(.text)
    .=e00h;
    .cinit=.;
    *(.cinit)
    .+= 1;
    .=00f00h;
    *(.const)
    .=01000h;
    *(.stack)
    .=01040h;
    *(.bss) } load=0800h  PAGE 0
  }
}
```

---

The command file for the DSPHEX will be:

```
filename.out
-t
-bootorg 08000h
SECTIONS { bootsect = boot }
```

The program entry point of C code is `_c_int0`. Therefore, in the linker command file the `_c_int0` has to be assigned the starting address. This was done by first line in the SECTIONS:

```
rts50.lib(.text) = 0800h
```

Since the `.cinit` section was hidden in another section, we need to make it visible to the linker by

```
cinit=.;
*(.cinit)
.+=1;
```

The commands in the SECTIONS assign a starting address to each input section and it is relative to the starting address of first section. This means that `.cinit` starts from 0x1600, `.const` starts from 0x1700, `.stack` starts from 0x1800, and `.bss` starts from 0x1840. If you don't want to generate unused space in between each section, you can remove the `".=0xxxxxh;"` command and all the sections will be placed consecutively. When you link the file with the example linker command file above, you will get the following warning messages "output file has no `.text` section" and "output file has no `.bss` section." You can ignore these messages.