Hitachi America, Ltd.

TN-0216

Application Engineering
TechNote

Balu Donthi

HAL/GNU: Reserving Memory for a Peripheral.

This document describes one of the ways of reserving a range of memory addresses using the GNU linker (ld). If certain locations in the memory are being used by a peripheral device (Memory Mapped I/O) in an Embedded Controller and you want make sure that neither code nor data gets located at these addresses than you can use the following example:

In this example, a section called .resadd is created using an assembly program (resadd.s) to reserve an area of 1kb memory.

file   "resadd.s"

```
        .global _resdata
        .section        .resadd
_resdata:
        .byte 0
        .space 1023
        .end
```

A corresponding linker script file is created to locate the section at the address 0xA000, so as to reserve the addresses from 0xA000 to 0xA400.

file "resadd.ld"

```
OUTPUT_ARCH(h8300)
 OUTPUT_FORMAT(srec)

SECTIONS
{
        .text 0x8000 : { *(.text) _etext = . ; }
        .resadd :{      } > RESMEM
        .data .   : { *(.data) _edata = . ;}
        .bss  .     : { *(.bss)  *(COMMON) _end = . ; }
        .stack 0xf800 : { *(.stack) }
}

MEMORY {
        RESMEM : org = 0xA000, l = 1K
        }
```

To see if the assembly file and linker script file reserve the addresses from 0xA000 to 0xA400, we can compile and link these files with the hello.c given below.

file "hello.c"

```
#include <stdio.h>
int i=0;
int arr[100];
fact(x)
{
  if (x > 1) return (x*(fact(x-1)));
  return 1;
}
main()
{

        for (i = 1; i < 5; i++)
                iprintf("Hello World %d\n",fact(i));
}
```

The following commands can be used for assembling, compiling and linking the resadd.s and the hello.c programs.

```
AS -oresadd.o resadd.s    /* Command line for assembling the resadd.s assembly program */
GCC -c hello.c      /* Command for compiling the hello.c program to produce a hello.o re-locatable object file */
LS -Tresadd.ld  resadd.o hello.o c:\gnuh83\lib\libc.a  -M > resadd.map   /* Command to link the objects generated by
                                                                            the previous commands to the Run time
                                                                            library */
```

<u>file "resadd.map" (snap shot)</u>

```
name              origin   length   r_size   c_size    is    attributes
*default*         00000000 ffffffff 00000000 00000000 ()
RESMEM            0000a000 00000400 00000000 00000400 ()

**LINK EDITOR MEMORY MAP**

output   input     virtual
section  section   address    tsize
.resadd   0000a000   400     0 2**1 load alloc contents
 .resadd  0000a000   400   400 2**1  coff-h8300 resadd.o
```

By examining the file resadd.map we can see that the address range between 0xA000 and 0xA400 has been allocated to section resadd, which was created to reserve these addresses for memory mapped I/O or any other peripheral usage.

The information in this document has been carefully checked; however, the contents of this document may be changed and modified without notice. Hitachi America, Ltd. shall assume no responsibility for inaccuracies, or any problem involving a patent infringement caused when applying the descriptions in this document. This material is protected by copyright laws.