

Hterm+Renesasモニタでdebug目的
のHEWでのコンパイル
(割り込みなしの場合)

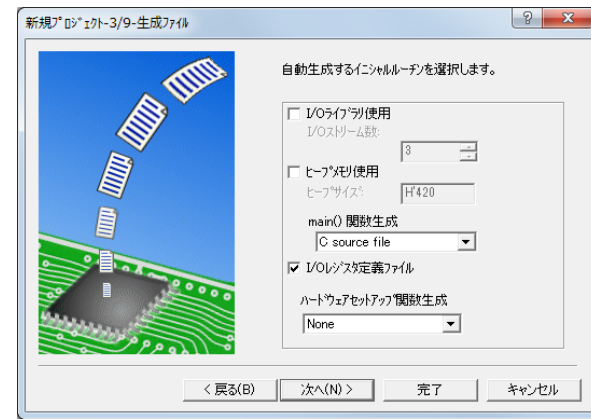
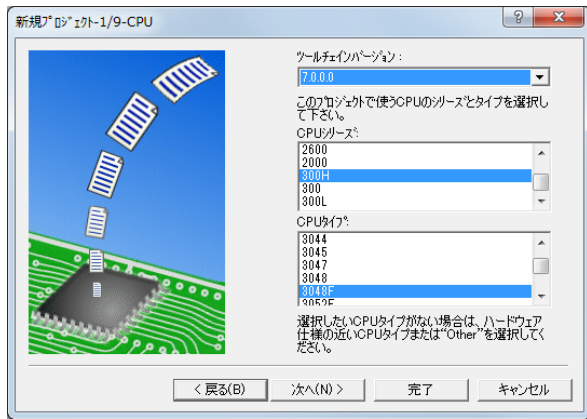
小坂

要点

- H8/3048はRAMが小さく, モニタでdebugすることのできるユーザ領域は約3kbyteなので大きなプログラムは実行できない
- 幸い, モニタにはprintfとscanf相当のルーチンがあるため, ユーザプログラムを小さくできる.
- 割り込みは使わない

プロジェクト生成時

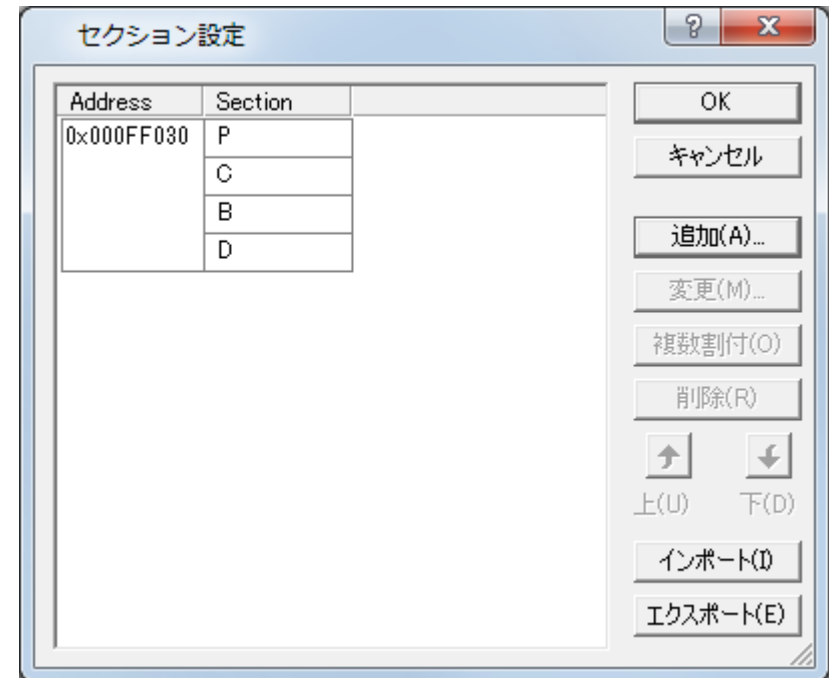
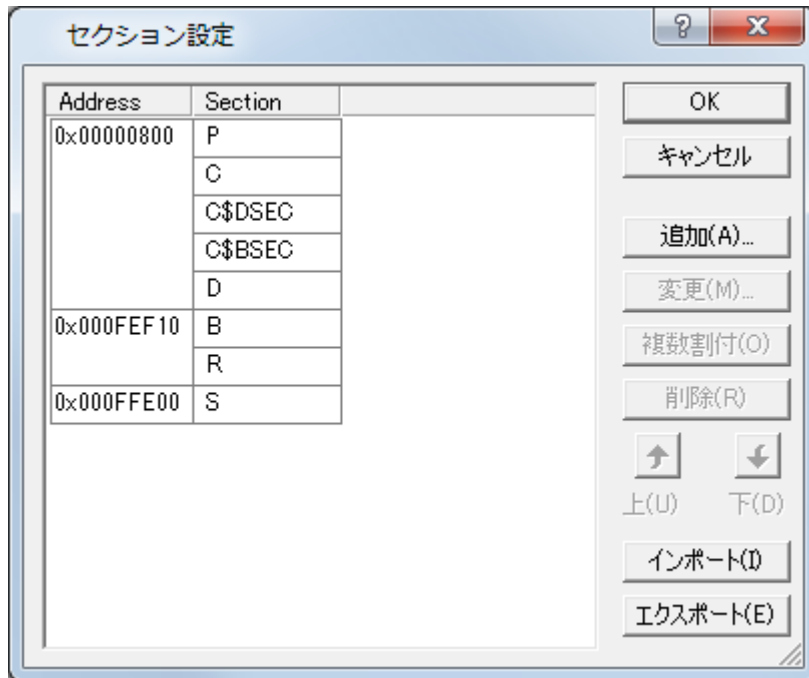
- アドレス空間は1Mbyte
- HEAPは使わないので削除
- ライブラリは何も追加しない
- スタックサイズは0x100に一応設定
- スタックポインタの設定はせず, モニタに任せる
- 割り込みベクタは削除



コンパイル時のオプション

- Release, Debug切り替えでDebugに
- dbsct.cはプロジェクトから削除
- ビルドのtoolchaineで次の設定を行う
 - コンパイラ:最適化なし
 - 最適化リンカ:入力:エントリポイント:「_main」に設定
 - 最適化リンカ:出力:出力形式:バイナリ(ELF/DWARFアブソリュート付)
 - 最適化リンカ:出力:オプション項目:ROMからRAMへマップを削除
 - 最適化リンカ:セクション:次のページのように設定

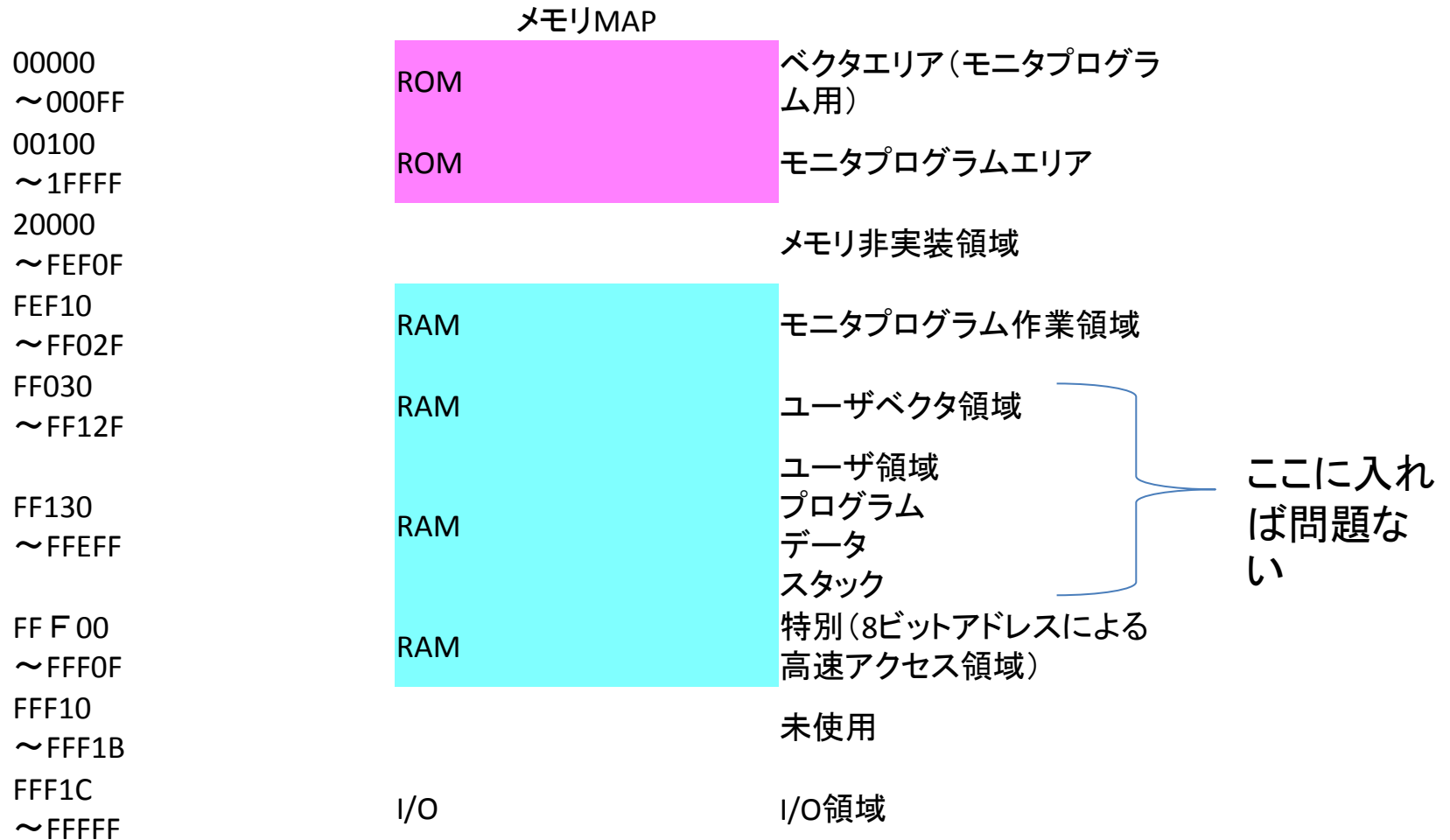
セクションをすべてRAMのユーザ領域 に移動



ビルドしてHtermへ

- HEWでビルドすると
- プロジェクトのディレクトリ内の¥Debugにxxx.absができる
- 同じ場所にxxx.mapができるのでサイズを確認

モニタのメモリマップ



Htermでのソースレベルデバッグ

- Htermを起動し, F9キーで h83048SDebugMiniSample.absをロードする.
- ソースファイルは複数ある
- h83048SDebugMiniSample.cを表示して, ブレークポイントを設定して, F5キーでゴー(起動は, main()の先頭)

Htermでのデバッグの様子

The screenshot shows the Hterm interface with three main panes:

- Console:** Displays the execution flow of a program. It shows several breakpoints (Break at PC=FF082, PC=FF0A4, PC=FF0C0, PC=FF0A4) and the state of registers (ER0-ER7) and memory (SP). The program is in a loop, and the user has entered commands like 'PushSW = f0' and 'DipSW = 5f'.
- Variable - h8304...:** Shows the state of global and local variables. The local variable 'sum' is located at memory address 0004 and has a value of 15.
- Memory - FFEF0...:** Shows the memory dump for the current address range. The data is as follows:

<ADDR>	<DATA>
FFEF0	8080
FFEF2	29C0
FFEF4	000F
FFEF6	0A06
FFEF8	000F
FFEFA	0006
FFEFC	0000
FFEFE	0000
- Source - h83048SDebugMiniSample.c:** Shows the source code of the program. The code includes a main function that initializes variables, sets up hardware (DDR, PCR), and enters a loop. The current execution point is at line FF0A4, which prints the PushSW status.

Htermでのデバッグの様子

変数iはER6の値-2をアドレスとする位置にあるので、FFFEFAが変数i、変数sumはER6の値-4をアドレスとする位置にあるので、FFFEF8が変数sum。ループを抜け出している

```
FF00 00000000 ER0=00000000
FF04 00000000 ER4=00000000
FF08 00000000 ER8=00000000
FF0C 00000000 ER12=00000000
FF10 00000000 ER16=00000000
FF14 00000000 ER20=00000000
FF18 00000000 ER24=00000000
FF1C 00000000 ER28=00000000
FF20 00000000 ER32=00000000
FF24 00000000 ER36=00000000
FF28 00000000 ER40=00000000
FF2C 00000000 ER44=00000000
FF30 00000000 ER48=00000000
FF34 00000000 ER52=00000000
FF38 00000000 ER56=00000000
FF3C 00000000 ER60=00000000
FF40 00000000 ER64=00000000
FF44 00000000 ER68=00000000
FF48 00000000 ER72=00000000
FF4C 00000000 ER76=00000000
FF50 00000000 ER80=00000000
FF54 00000000 ER84=00000000
FF58 00000000 ER88=00000000
FF5C 00000000 ER92=00000000
FF60 00000000 ER96=00000000
FF64 00000000 ER100=00000000
FF68 00000000 ER104=00000000
FF6C 00000000 ER108=00000000
FF70 00000000 ER112=00000000
FF74 00000000 ER116=00000000
FF78 00000000 ER120=00000000
FF7C 00000000 ER124=00000000
FF80 00000000 ER128=00000000
FF84 00000000 ER132=00000000
FF88 00000000 ER136=00000000
FF8C 00000000 ER140=00000000
FF90 00000000 ER144=00000000
FF94 00000000 ER148=00000000
FF98 00000000 ER152=00000000
FF9C 00000000 ER156=00000000
FFA0 00000000 ER160=00000000
FFA4 00000000 ER164=00000000
FFA8 00000000 ER168=00000000
FFAC 00000000 ER172=00000000
FFB0 00000000 ER176=00000000
FFB4 00000000 ER180=00000000
FFB8 00000000 ER184=00000000
FFBC 00000000 ER188=00000000
FFC0 00000000 ER192=00000000
FFC4 00000000 ER196=00000000
FFC8 00000000 ER200=00000000
FFCC 00000000 ER204=00000000
FFD0 00000000 ER208=00000000
FFD4 00000000 ER212=00000000
FFD8 00000000 ER216=00000000
FFDC 00000000 ER220=00000000
FFE0 00000000 ER224=00000000
FFE4 00000000 ER228=00000000
FFE8 00000000 ER232=00000000
FFEC 00000000 ER236=00000000
FFF0 00000000 ER240=00000000
FFF4 00000000 ER244=00000000
FFF8 00000000 ER248=00000000
FFFC 00000000 ER252=00000000
```

```
Variable - h8304...
<GLOBAL VARIABLE LOCATION>
<LOCAL VARIABLE LOCATION>
@(-2,ER6) i
@(-4,ER6) sum

Memory - FFEF0...
<ADDR> <DATA>
FFEF0 8080
FFEF2 29C0
FFEF4 000F
FFEF6 F0A6
FFEF8 000F
FFEFA 0006
FFEFC 0000
FFEFE 0000
```

```
Source - h83048SDebugMiniSample.c
/*****
/* FILE :h83048SDebugMiniSample.c
/* DATE :Wed, Sep 12, 2012
/* DESCRIPTION :Main Program
/* CPU TYPE :H8/3048F
/* This file is generated by Renesas Project Generator (Ver.4.16).
/*
/*****
/*test program KOSAKA 20120912*/

#include "iodefine.h"

#define printf ((int (*)(const char *,...))0x00002b3c)
#define scanf ((int (*)(const char *,...))0x00002b8e)

FF030 void main(void)
{
    FF038     int i;
    FF060     int sum = 0;
    FF048     for (i=1; i<=5; i++) {
                sum += i;
    }
    FF06A     printf("sum=%d\n", sum);
    FF082     P5.DDR = 3; /*set LED port outflow*/
    FF086     P5.DR.BYTE = 1; /*turning on LED0*/
    FF08A     P5.DR.BYTE = 3; /*turning on LED1*/
    FF08E     P5.DR.BYTE = 0; /*turning off LED0 & 1*/
    FF092     P4.DDR = 0; /*set PushSW port inflow*/
    FF096     P4.PCR.BYTE = 0xf0; /*set PushSW port pull up*/
    FF09A     P2.DDR = 0; /*set DSW port inflow*/
    FF09E     P2.PCR.BYTE = 0xff; /*set DSW port pull up*/
    FF0D8     while(1) {
    FF0A4     printf("PushSW = %02x\n",P4.DR.BYTE&0xf0); /*print PushSW
    FF0C0     printf(" DipSW = %02x\n",P2.DR.BYTE); /*print Dip SW bit p
    FF0DA     }
}
```