

ビットフィールド宣言「iodef.h」等の評価可能にする (ルネサスC)

Rev1.00 2009/10/05
DEFバージョン 7.70B 以上

【対象】

ルネサスC (ELF/Dwarf2) でコンパイル/リンクした全CPU品種が対象になります。

【概要】

ルネサス提供の「iodef.h」で宣言しているビットフィールドを利用して評価・変更を可能にする一例を記述します。

【宣言および記述例】 [H8/S/2612](#) における例を示します。

```
<iodef.h>

#define MSTP      (*(volatile struct st_mstp *)0xFFFFDE8) /* MSTP Address*/
#define P1        (*(volatile struct st_p1   *)0xFFFFE30) /* P1   Address*/
#define TPU       (*(volatile struct st_tpu  *)0xFFFFE80) /* TPU  Address*/
#define TPU0      (*(volatile struct st_tpu0 *)0xFFFFF10) /* TPU0 Address*/
#define INTC      (*(volatile struct st_intc *)0xFFFFE12) /* INTC Address*/
```

ルネサスC提供「iodef.h」の一部抜粋

「#define」で宣言されている為、シンボル情報を抽出することが出来ません。シンボル情報を作成するため独自の宣言ソースを作成します。

```
<iestruct.c>
#include "iodef.h"
volatile struct st_mstp *const _MSTP    = {&MSTP}      /* MSTP Address*/
volatile struct st_p1   *const _P1     = {&P1}         /* P1   Address*/
volatile struct st_tpu  *const _TPU    = {&TPU};        /* TPU  Address*/
volatile struct st_tpu0 *const _TPU0   = {&TPU0};       /* TPU0 Address*/
volatile struct st_intc *const _INTC   = {&INTC};       /* INTC Address*/
```

「iestruct.c」<—追加作成

構造体ポインタとして宣言をして変数エリアを確保します。ポインタ変数の配置場所が「ROM」エリアになるように「const」を付けます。変数名は取り合えずdefine名にアンダーバー“_”を追加します。

```
<iestruct.h>

extern volatile struct st_mstp *const _MSTP; /* MSTP Address*/
extern volatile struct st_p1   *const _P1;  /* P1   Address*/
extern volatile struct st_tpu  *const _TPU; /* TPU  Address*/
extern volatile struct st_tpu0 *const _TPU0; /* TPU0 Address*/
extern volatile struct st_intc *const _INTC; /* INTC Address*/
```

「iestruct.h」<—追加作成

「iestruct.c」で宣言した変数を各モジュールで使用する為、ヘッダファイルを作成します。

【使用例】

```
#include "iestruct.h"

void main(void)
{
    _MSTP->CRA.BIT_TPU    = 0;
    _P1->DR.BYTE          = 0xff;
    _TPU->TSTR.BIT_CST0   = 1;
    _TPU0->TGRA           = 50000;
    _INTC->IPRF.BIT_TPU0  = 5;
}

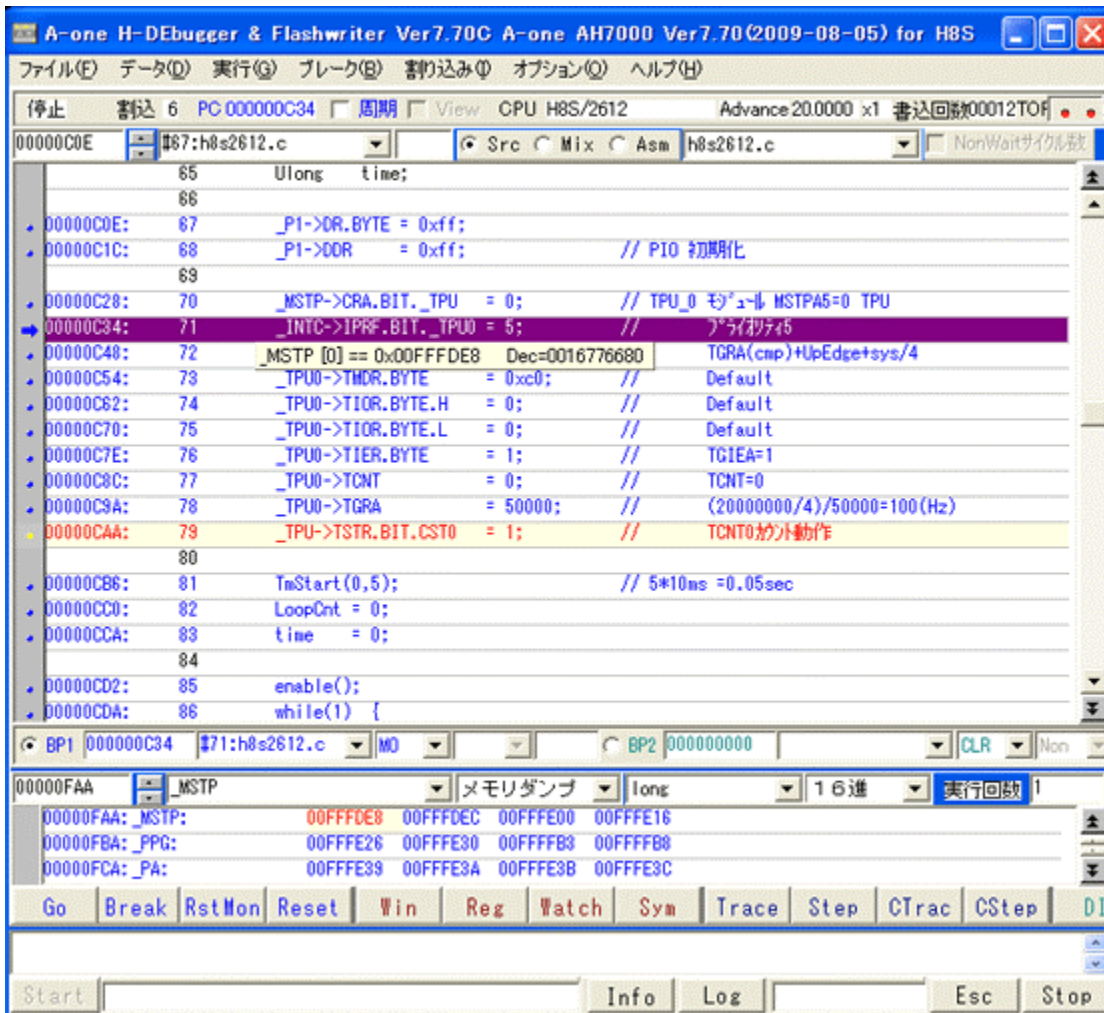
```

「iestruct.c」で作成したポインタ構造体のビットフィールド記述例

構造体ポインタになりますので、「_MSTP->CRA.BIT._TPU」の記述になります。

【操作例】

<1-1>



DEFの全体画面

←[_MSTP]を評価します。

マウス右クリックのポップアップメニューで

「評価・変更(1)」を選択します。

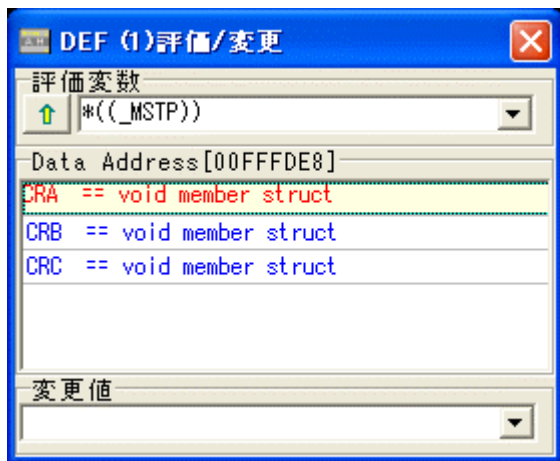
<1-2>



[_MSTP]は、ROM エリア[0xFAA]に配置され、内容は I/O アドレス[0xFFFDE8]になっていることが確認できます。

←マウスをあてダブルクリックします。

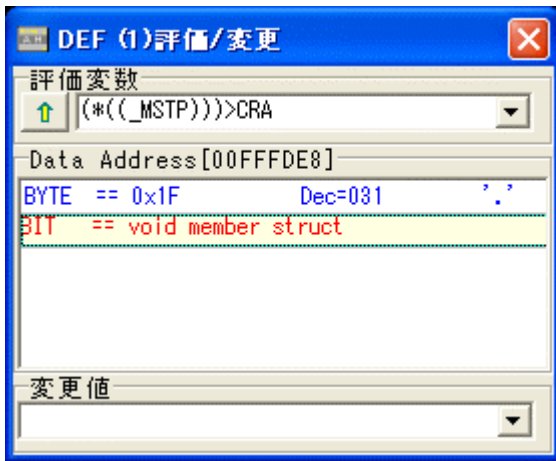
<1-3>



[_MSTP]のメンバ宣言「struct st_mstp」が確認できます。

←[CRA]にマウスをあてダブルクリックします。

<1-4>



[_MSTP->CRA]のユニオンメンバが確認できます。

<-[BIT]にマウスをあてダブルクリックします。

<1-5>



ビットフィールド[_MSTP->CRA.BIT._TPU]がゼロ[0]になっているのが確認できます。

<-ビット位置と内容が確認できます。

以上でビットフィールドの参照例の説明を終了します。