Rev 1.10.00

ルネサス提供ビットフィールド宣言「iodefine.h」を デバッガ「ＡＨ８０００」で可視化する方法

１．対象品種

・CC-RX（ELF/Dwarf2 or 3）でコンパイル／リンクしたＲＸ品種が対象。
・GCC for RenesasRX（ELF/Dwarf2 or 3 or 4）でコンパイル／リンクしたＲＸ品種が対象。

２．概要説明

・ルネサス提供の「iodefine.h」を利用してプログラムした場合に、ビットフィールド情報を可視化する方法を説明する。

３．実装準備

品種	「iodefine.h」をベースにした構造体ポインタ宣言
RX230/231	Iostruct_rx230.c
RX23T	Iostruct_rx23t.c
RX631/63N	Iostruct_rx63n.c
RX64M	Iostruct_rx64m.c
RX651/65N	Iostruct_rx65x.c
RX66N	Iostruct_rx66n.c
RX72M	Iostruct_rx72m.c

<Iostruct_rx65x.c>　ソース内容（一部）

```
#include "iodefine.h"          // Include iodefine.h
volatile struct st_bsc    * const  _BSC    =(struct st_bsc    *)&BSC;
volatile struct st_cac    * const  _CAC    =(struct st_cac    *)&CAC;
volatile struct st_can    * const  _CAN0   =(struct st_can    *)&CAN0;
volatile struct st_can    * const  _CAN1   =(struct st_can    *)&CAN1;
volatile struct st_cmt    * const  _CMT    =(struct st_cmt    *)&CMT;
volatile struct st_cmt0   * const  _CMT0   =(struct st_cmt0   *)&CMT0;
```

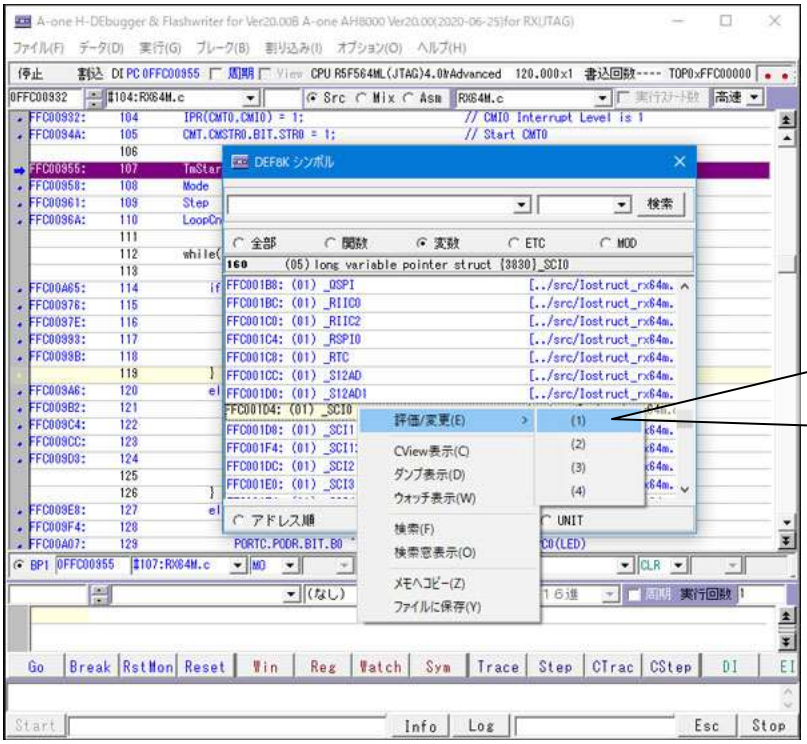
＜準備＞

（1）上記の RX 品種別の「iodefine.h」をベースにした構造体ポインタ宣言の C ソースを下記 URL からダウンロードする。
https://aone.co.jp/tools/AH8000/Ctext/index.html

（2）ダウンロードした C ソースをプロジェクトに実装してビルドする。なお、この C ソースは、各構造体ポインタのアドレス値を ROM テーブル化するだけの宣言のみのソースとなります。

4．操作例

<4-1>


評価／変更（１）を選択します。
操作例「SCI0」で示す。

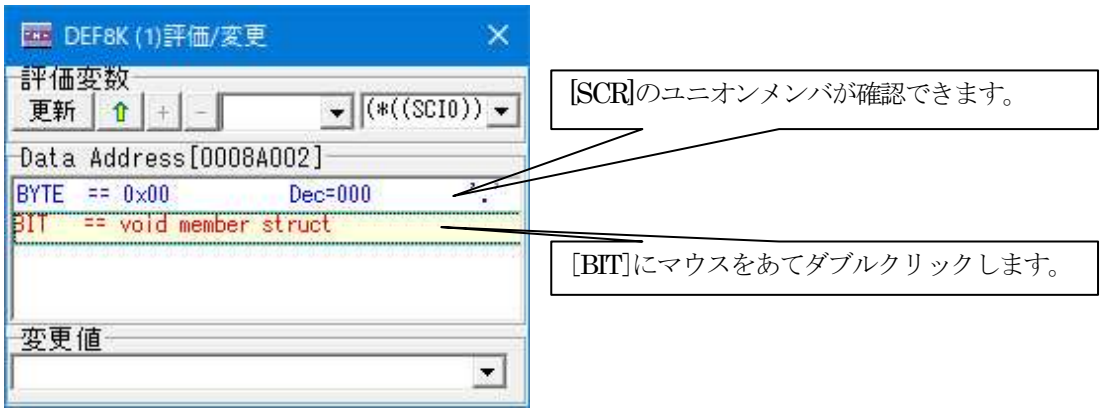
<4-2>


DEF8K (1)評価/変更
評価変数
更新
SCI0
Data Address[FFF801C4]
[0] == 0x0008A000 Dec=0000565248
変更値
[SIO0]は、ROM エリア[0xFFF801C4]に配置され、I/O アドレス値[0x0008A000]が確認できます。
マウスをあてダブルクリックします。

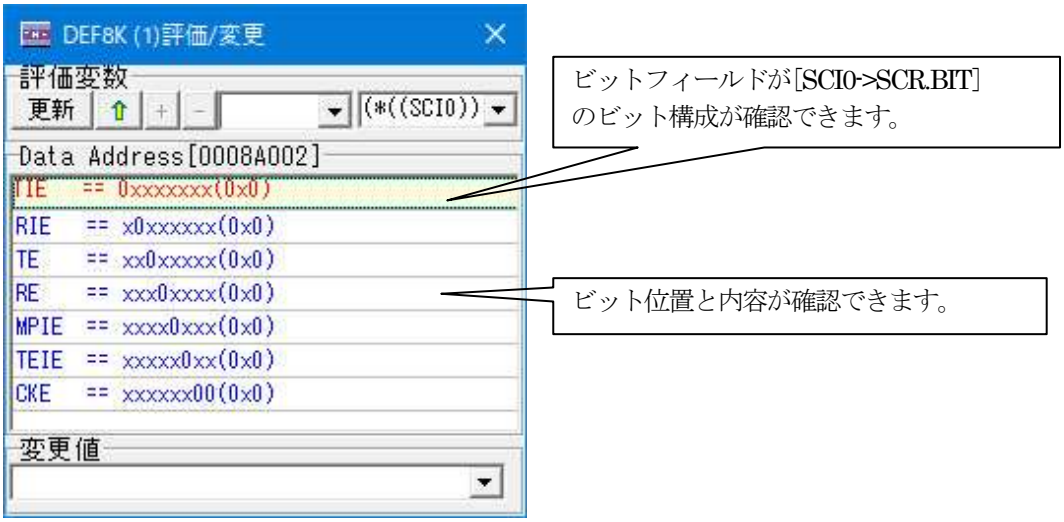
<4-3>


DEF8K (1)評価/変更
評価変数
更新
*((SCI0))
Data Address[0008A002]
SMR == void member struct
BRR == 0x00 Dec=000
SCR == void member struct
TDR == 0x00 Dec=000
変更値
[SCI0]のメンバ宣言「struct st_sci0」が確認できます。
[SCR]にマウスをあてダブルクリックします。

<4-4>



<4-5>



以上でビットフィールドの参照例の説明を終了します。

５．注意事項

・本文書の著作権は、エーワン（株）が保有します。
・本文書を無断での転載は一切禁止します。
・本文書に記載されている内容についての質問やサポートはお受けすることが出来ません。
・本文章に関して、ルネサス　エレクトロニクス社への問い合わせは御遠慮願います。
・本文書の内容に従い、使用した結果、損害が発生しても、弊社では一切の責任は負わないものとします。
・本文書の内容に関して、万全を期して作成しましたが、ご不審な点、誤りなどの点がありましたら弊社までご連絡くだされば幸いです。
・本文書の内容は、予告なしに変更されることがあります。