

構造体変数を評価／変更窓で参照する方法

1. 対象

ルネサス C (ELF/Dwarf2or3) および GCC for RenesasRX/KPIT-GNU/IAR-ICC/ (ELF/Dwarf2or3or4) でコンパイル／リンクした全 CPU 品種が対象になります。

2. 概要

デバッグ時に構造体の個別メンバを参照する方法を例に沿って説明します。

2-1) グローバル変数の宣言例

```
/******  
//      構造体宣言  
/******  
typedef struct {  
    char    ct1;  
    char    ct2[2][6];  
    char    ct3[3][2][2];  
    long    lt1;  
    long    lt2[2][6];  
    long    lt3[3][2][2];  
    Int64    lt1;  
    Int64    lt2[2][6];  
    Int64    lt3[3][2][2];  
} TEST1C;  
  
typedef struct {  
    char    dt1;  
    struct  {  
        char    dt2;  
        short   dt3;  
        TEST1C  stest1c;  
        long    dt4;  
        Int64    dt5;  
    }       stest1b;  
    short    dt5;  
    long     dt6;  
    Int64    dt7;  
} TEST1B;  
/******  
//      グローバル変数宣言例  
/******  
TEST1B    STest1B;  
TEST1B    *PTest1B;
```

2-2) ローカル変数の宣言例

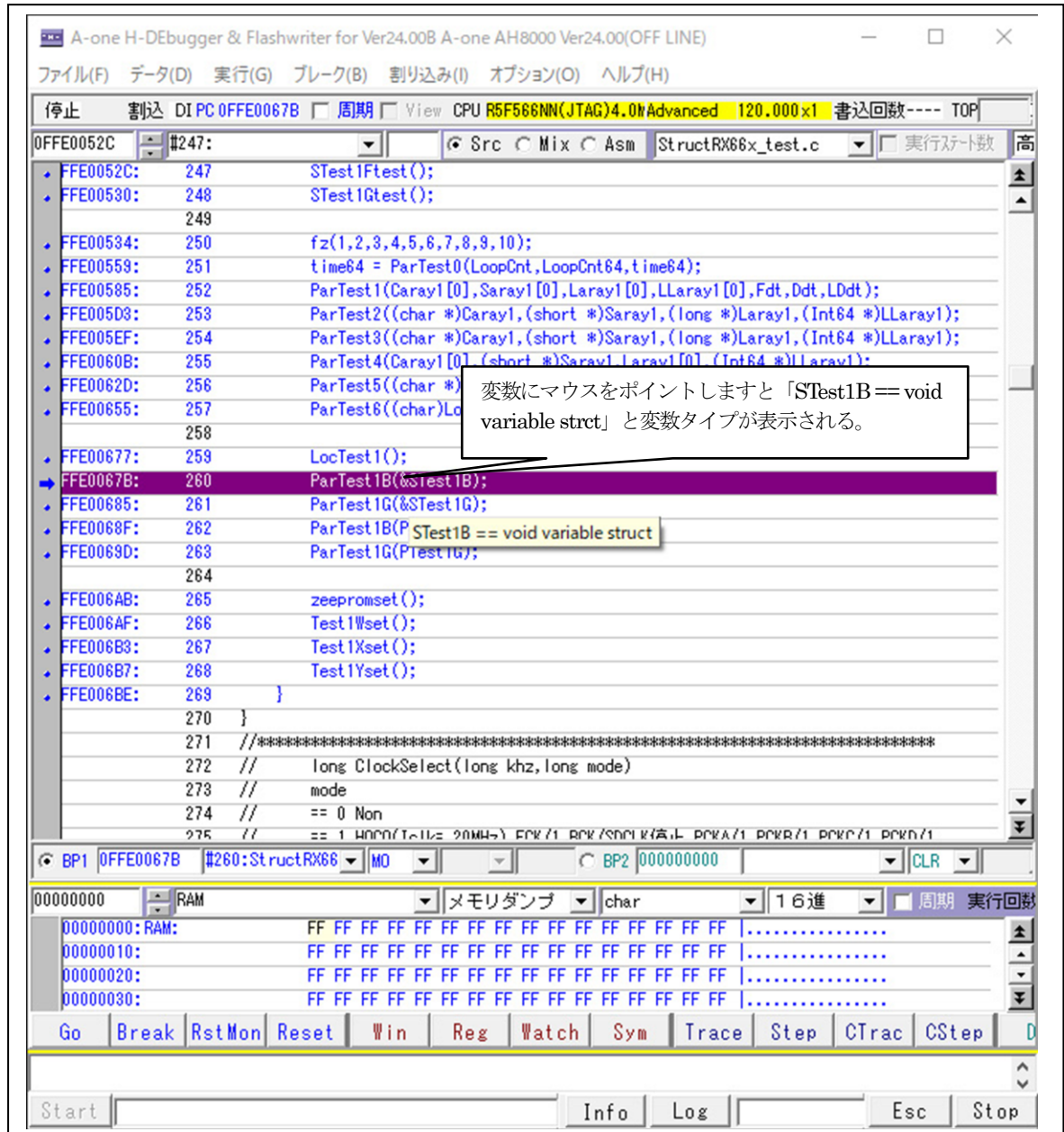
```
/******  
//      ローカル変数宣言例  
/******  
void    ParTest1B(TEST1B *ptest1b)  
{  
    TEST1B  temp;  
    TEST1B  temp22[2][2];  
    TEST1B *ptemp;  
    TEST1B *ptemp22[2][2];  
  
    temp      = *ptest1b;  
    temp22[0][0] = *ptest1b;  
    temp22[0][1] = *ptest1b;  
    temp22[1][0] = *ptest1b;  
    temp22[1][1] = *ptest1b;  
  
    ptemp      = &temp;  
    ptemp22[0][0] = &temp;  
    ptemp22[0][1] = &temp;  
    ptemp22[1][0] = &temp;  
    ptemp22[1][1] = &temp;  
}
```

【補足】

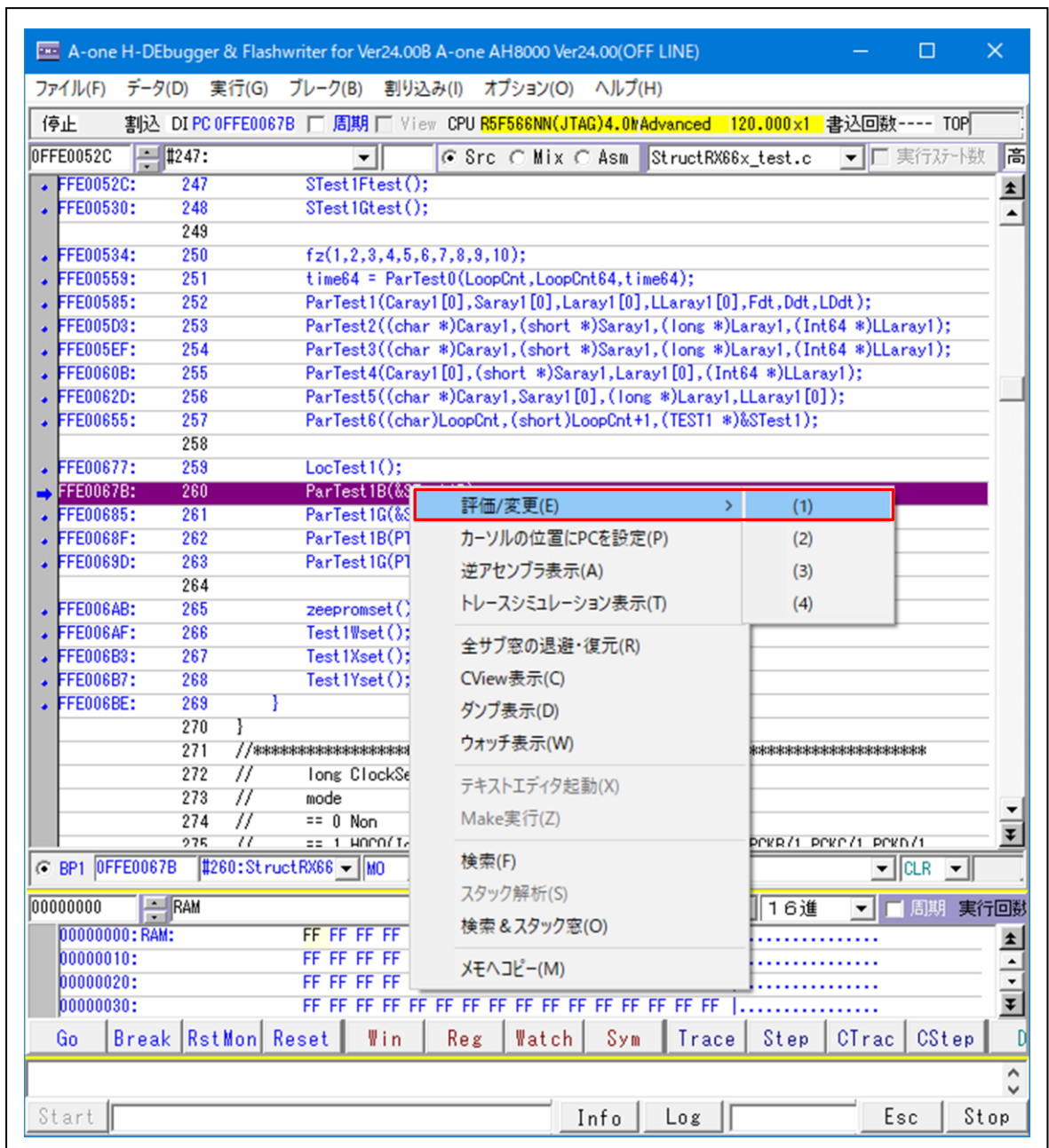
宣言例のように、構造体を2重にネストさせ、かつ最大3次元配列の宣言をした場合のグローバルおよびローカル宣言した場合の参照方法を説明します。

3. 操作説明 (グローバル変数)

3-1) 構造体変数「TEST1B STest1B」を参照する。

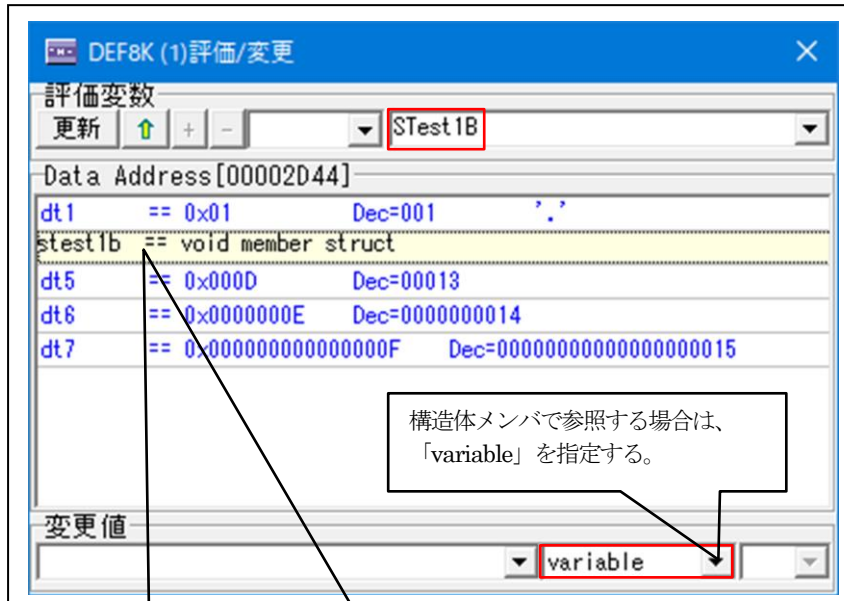


3-2) マウスをポイントした状態で右クリック後、評価/変更 (1) ~ (4) を選択します



3-3) 評価/変更窓の操作説明

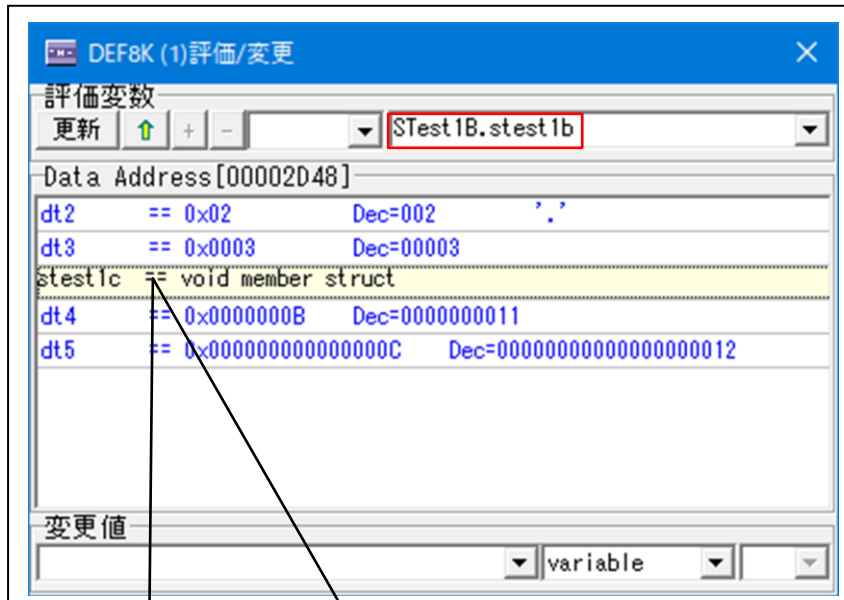
[3-1]



各個別メンバーの
数値を表示します。

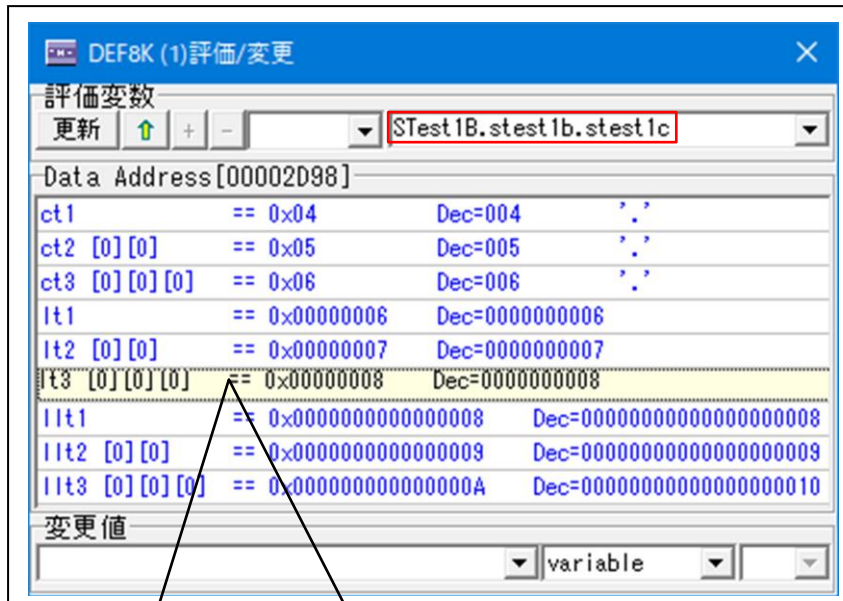
ネストしているメンバを参照する場合は、「ダブルクリック」する。

[3-2]



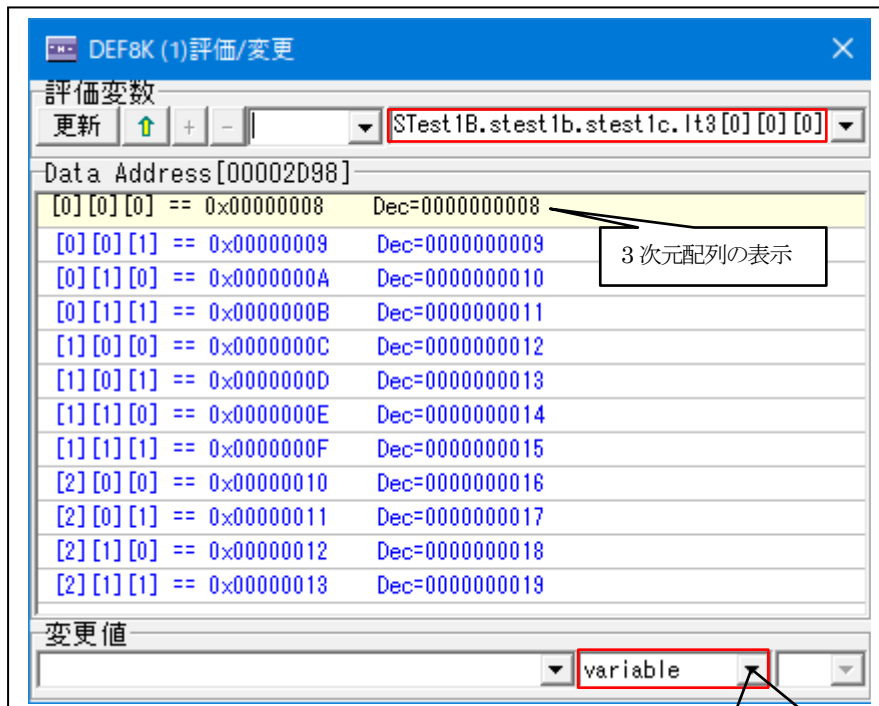
ネストしているメンバを参照する場合は、「ダブルクリック」する。

[3-3]



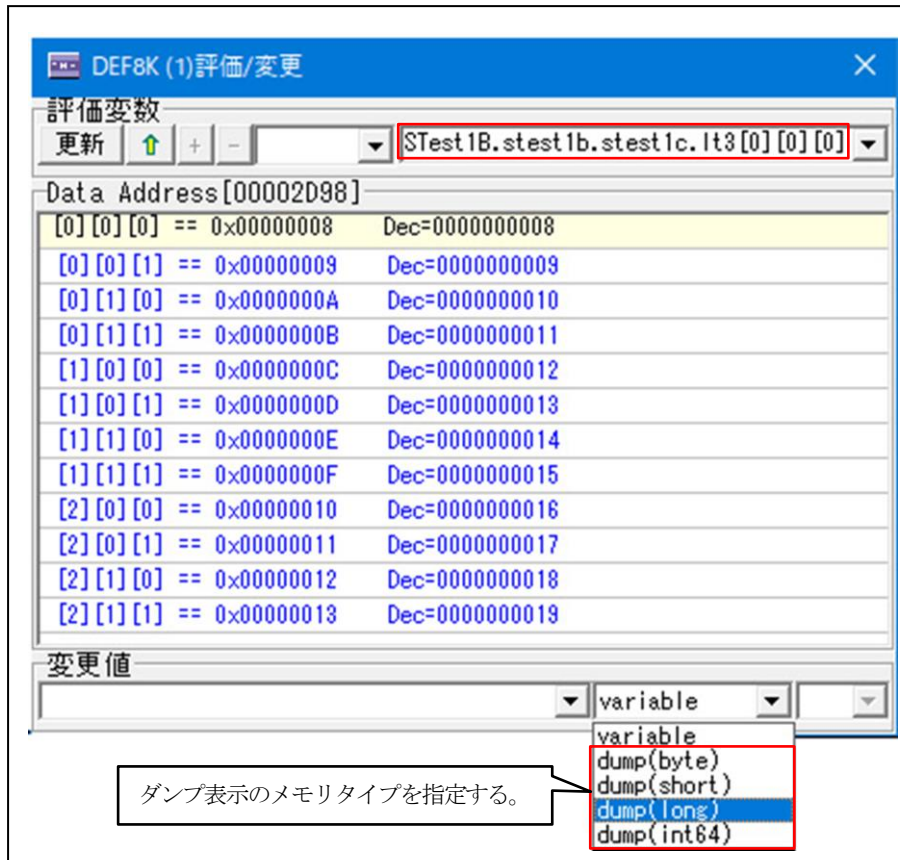
ネストしているメンバを参照する場合は、「ダブルクリック」する。

[3-4]

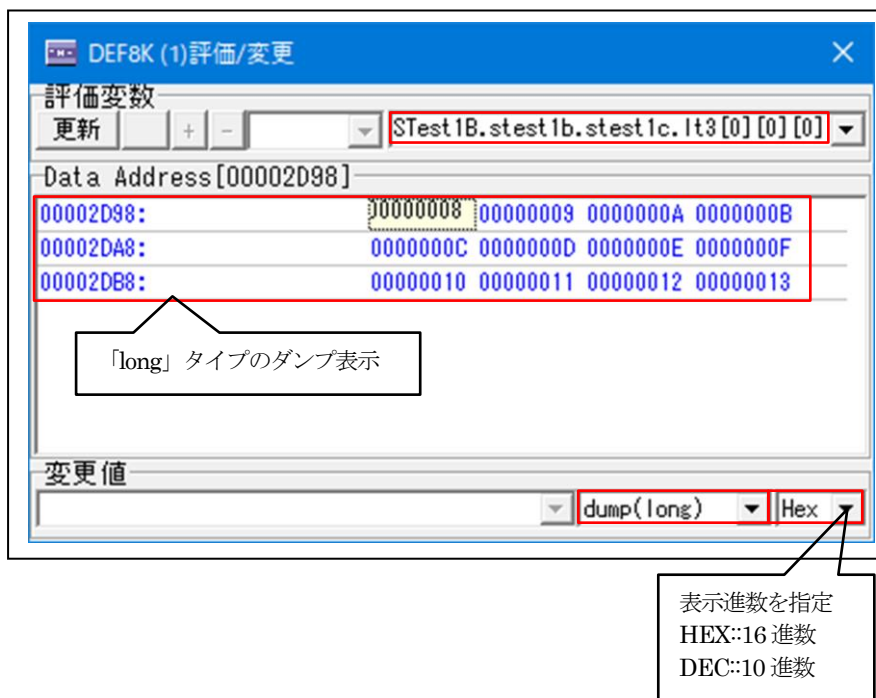


ダンプ表示に切り替える場合は、
ここの「PullDown」で指定する。

[3-5]



[3-6]



4. 操作説明 (ローカル変数)

4-1) ローカル構造体ポインタ引数「TEST1B *ptest1b」を参照する。

A-one H-Debugger & Flashwriter for Ver24.00B A-one AH8000 Ver24.00(OFF LINE)

ファイル(F) データ(D) 実行(G) ブレーク(B) 割り込み(I) オプション(O) ヘルプ(H)

停止 割込 DI PC 0FFE02FCD 周期 View CPU R5F566NN(JTAG)4.0n Advanced 120.000x1 書込回数---- TOP

0FFE02FC1 ParTest1B Src Mix Asm paratest.c 実行ステップ数 高

```

357 //*****
358 // ローカル変数宣言例
359 //*****
360 void ParTest1B(TEST1B *ptest1b)
361 {
362     TEST1B temp;
363     TEST1B temp2[2];
364     TEST1B *ptemp;
365     TEST1B *ptemp2[2];
366
367     temp = *ptest1b;
368     temp2[0][0] = *(0x2D40) == void register local symbol pointer struct
369     temp2[0][1] = *ptest1b;
370     temp2[1][0] = *ptest1b;
371     temp2[1][1] = *ptest1b;
372
373     ptemp = &temp;
374     ptemp2[0][0] = &temp;
375     ptemp2[0][1] = &temp;
376     ptemp2[1][0] = &temp;
377     ptemp2[1][1] = &temp;
378 }
379 //*****
380 void ParTest1G(TEST1G *ptest1g)
381 {
382     TEST1G temp;
383     TEST1G temp2[2];
384     TEST1G temp2[2][3];
385     TEST1G *ptemp;

```

変数にマウスをポイントしますと「*(0x2D40) == void register local symbol pointer struct」と変数タイプが表示される。

BP1 0FFE0067B #260:StructRX66 MO BP2 00000000 CLR

00000000 RAM メモリダンプ char 16進 周期 実行回数

00000000: RAM: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |.....

00000010: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |.....

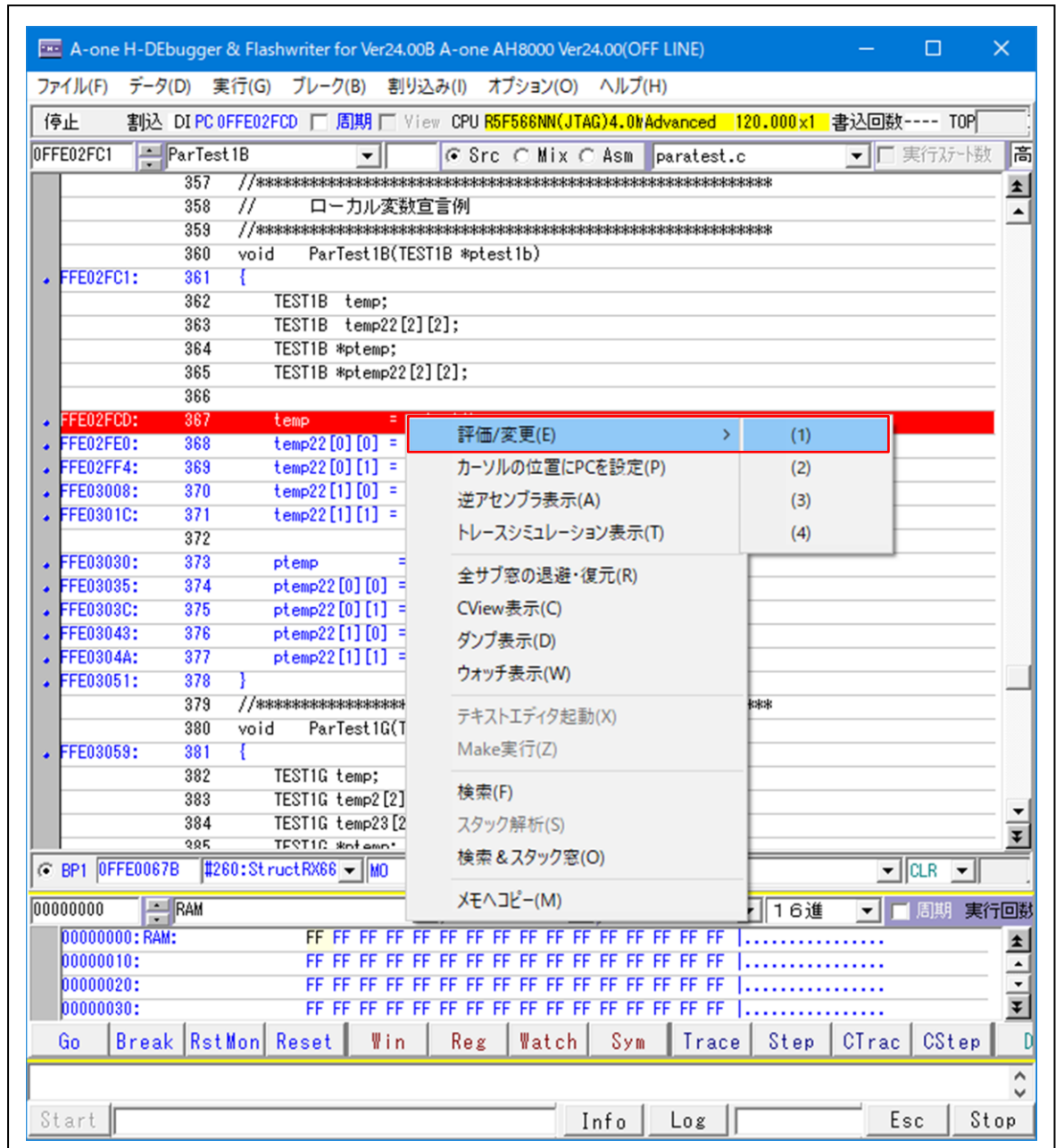
00000020: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |.....

00000030: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |.....

Go Break RstMon Reset Win Reg Watch Sym Trace Step CTrac CStep

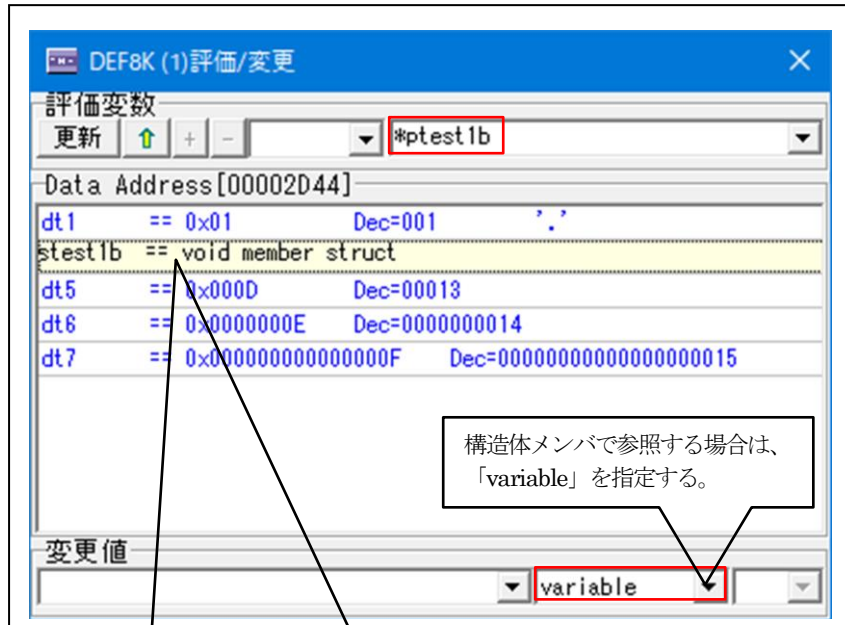
Start Info Log Esc Stop

4-2) マウスをポイントした状態で右クリック後、評価/変更 (1) ~ (4) を選択します



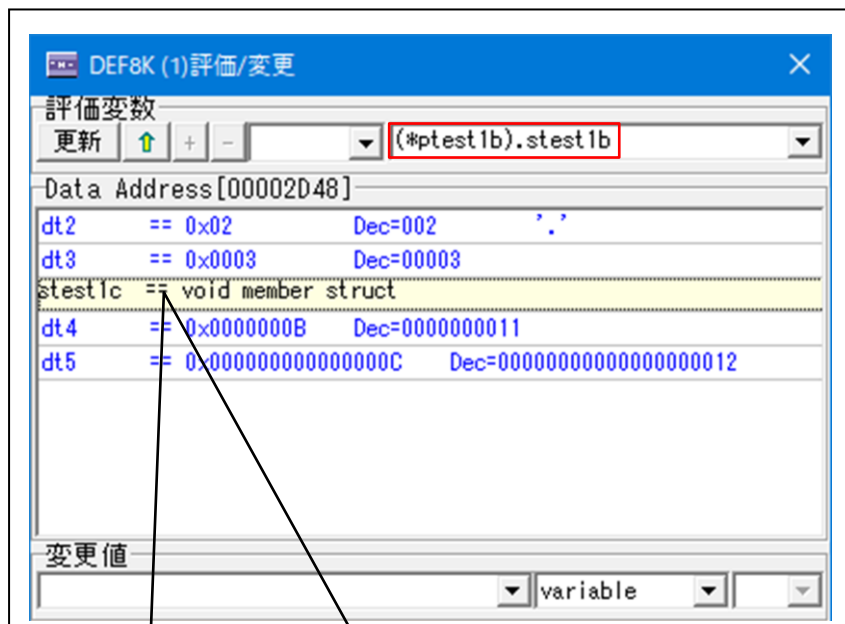
4-3) 評価/変更窓の操作説明

[3-1]

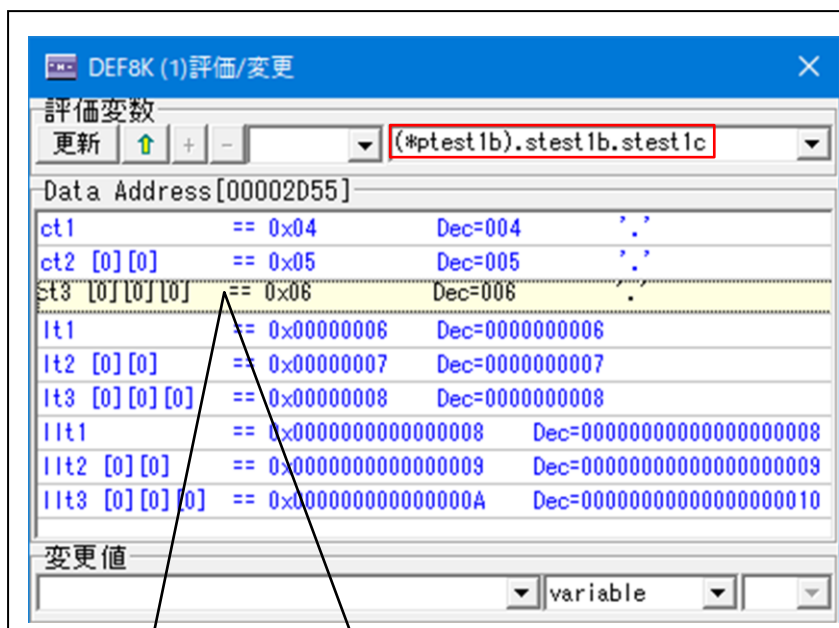


ネストしているメンバを参照する場合は、「ダブルクリック」する。

[3-2]

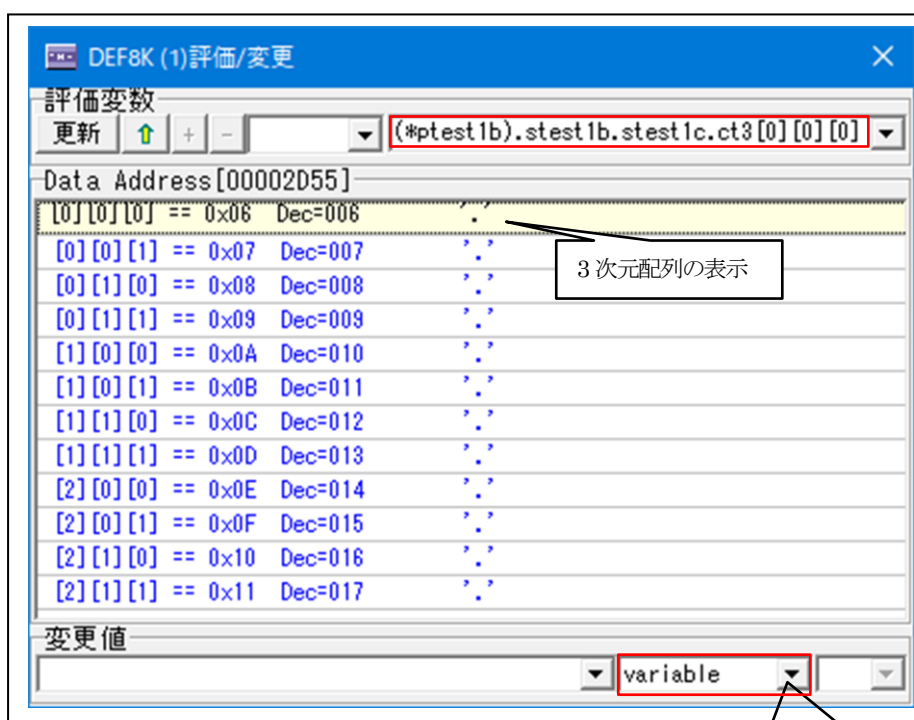


[3-3]



ネストしているメンバを参照する場合は、「ダブルクリック」する。

[3-4]



3次元配列の表示

ダンプ表示に切り替えする場合は、
ここの「PullDown」で指定する。

4-4) ローカル構造体ポインタ変数「TEST1B *ptemp」を参照する

The screenshot shows the A-one H-Debugger & Flashwriter interface. The main window displays the source code of 'paratest.c' with the following code snippet:

```

357 //*****
358 // ローカル変数宣言例
359 //*****
360 void ParTest1B(TEST1B *ptest1b)
361 {
362     TEST1B temp;
363     TEST1B temp2[2][2];
364     TEST1B *ptemp;
365     TEST1B *ptemp2[2][2];
366
367     temp = *ptemp;
368     temp2[0][0] = *ptemp;
369     temp2[0][1] = *ptemp;
370     temp2[1][0] = *ptemp;
371     temp2[1][1] = *ptemp;
372
373     ptemp = &temp;
374     ptemp2[0][0] = &temp;
375     ptemp2[0][1] = &temp;
376     ptemp[0] == 0x00001878 Dec=0000006274
377     ptemp2[1][1] = &temp;
378 }
379 //*****
380 void ParTest1G(TEST1G *ptest1g)
381 {
382     TEST1G temp;
383     TEST1G temp2[2];
384     TEST1G temp23[2][3];
385     TEST1G *ptemp;

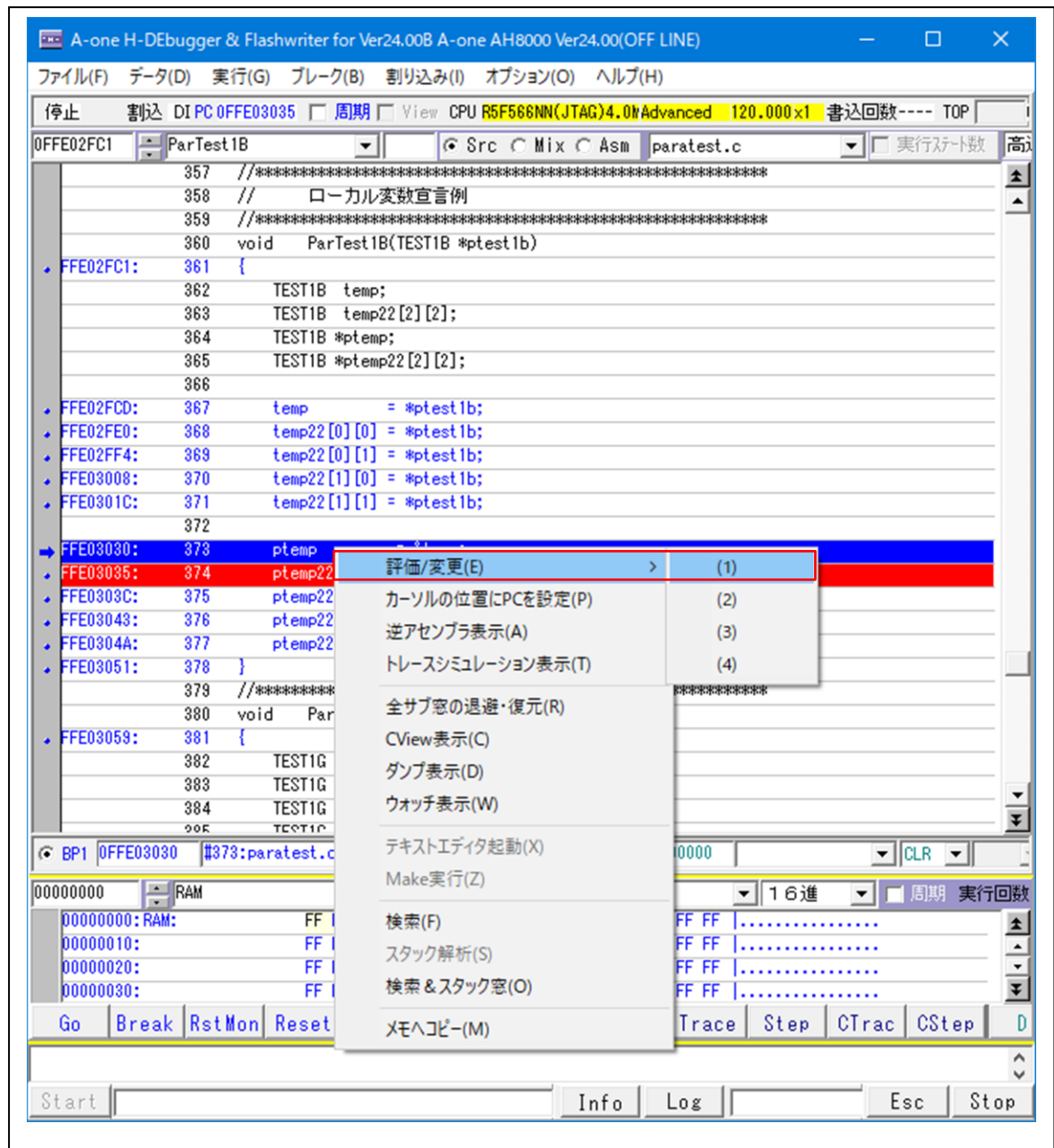
```

A callout box points to the line `ptemp[0] == 0x00001878 Dec=0000006274` with the text: "変数にマウスをポイントしますと「ptemp == 0x00001878 Dec=0000006274」と格納されたアドレス値を表示される。"

The bottom window shows the RAM dump starting at address 00000000:

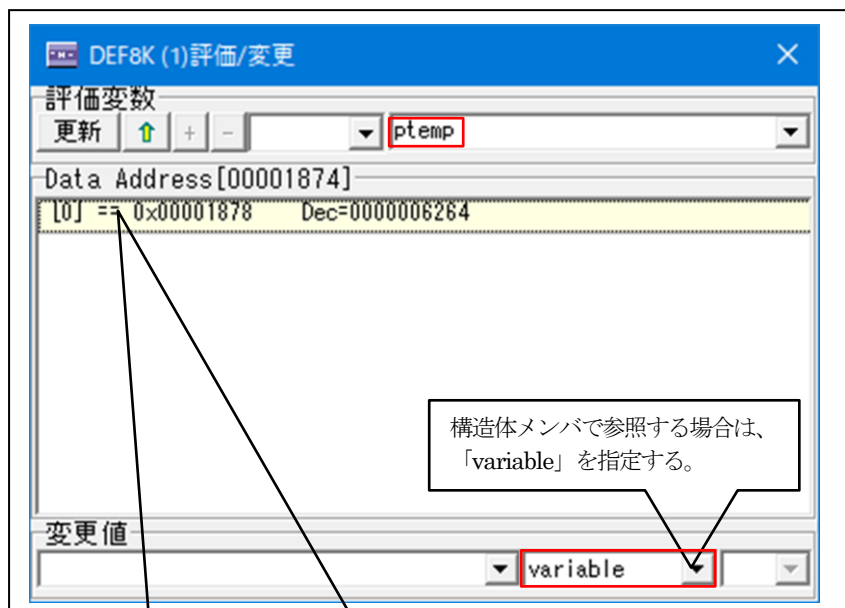
Address	Value
00000000: RAM:	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000010:	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000020:	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000030:	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

4-5) マウスをポイントした状態で右クリック後、評価/変更 (1) ~ (4) を選択します



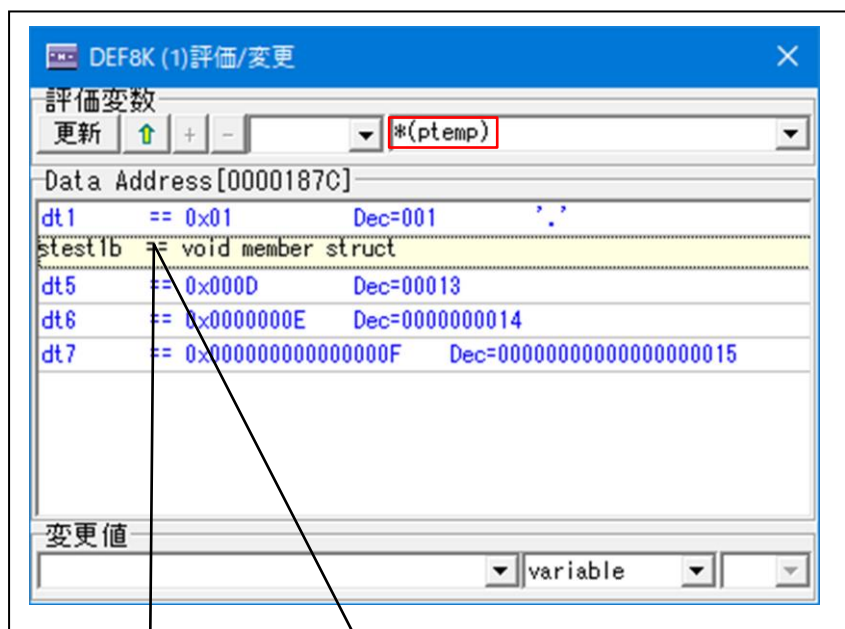
4-6) 評価/変更窓の操作説明

[6-1]



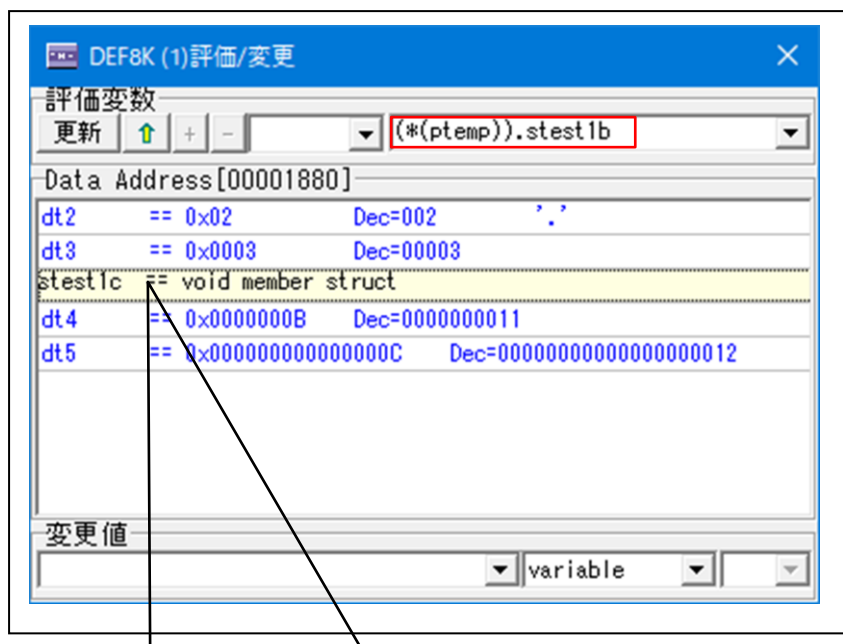
構造体メンバを参照する場合は、「ダブルクリック」する。

[6-2]



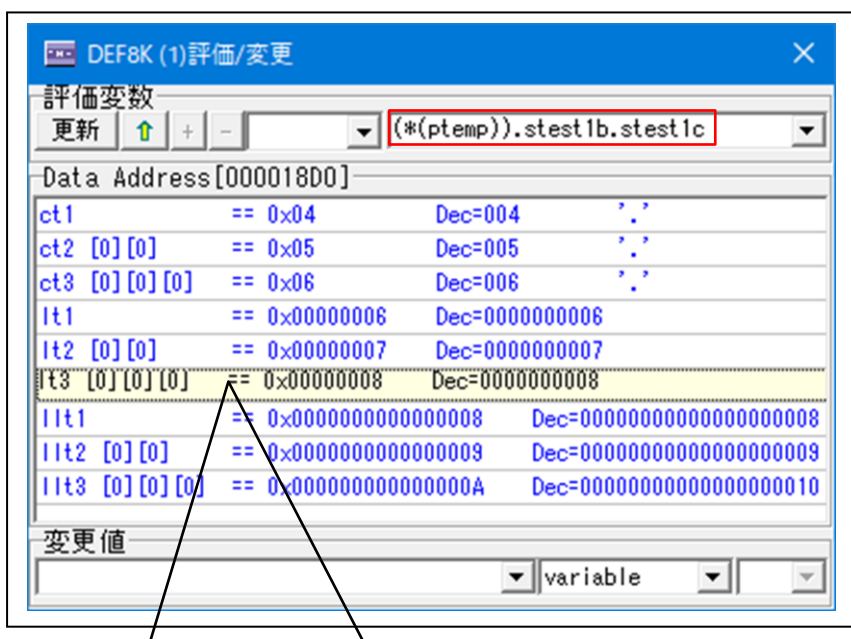
ネストしているメンバを参照する場合は、「ダブルクリック」する。

[6-3]



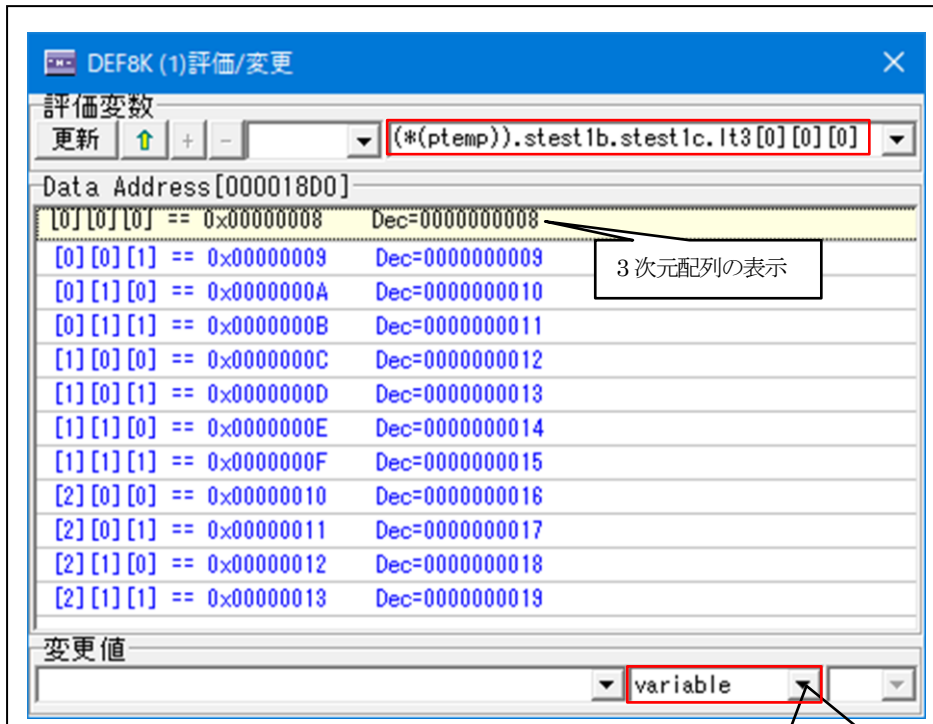
ネストしているメンバを参照する場合は、「ダブルクリック」する。

[6-4]



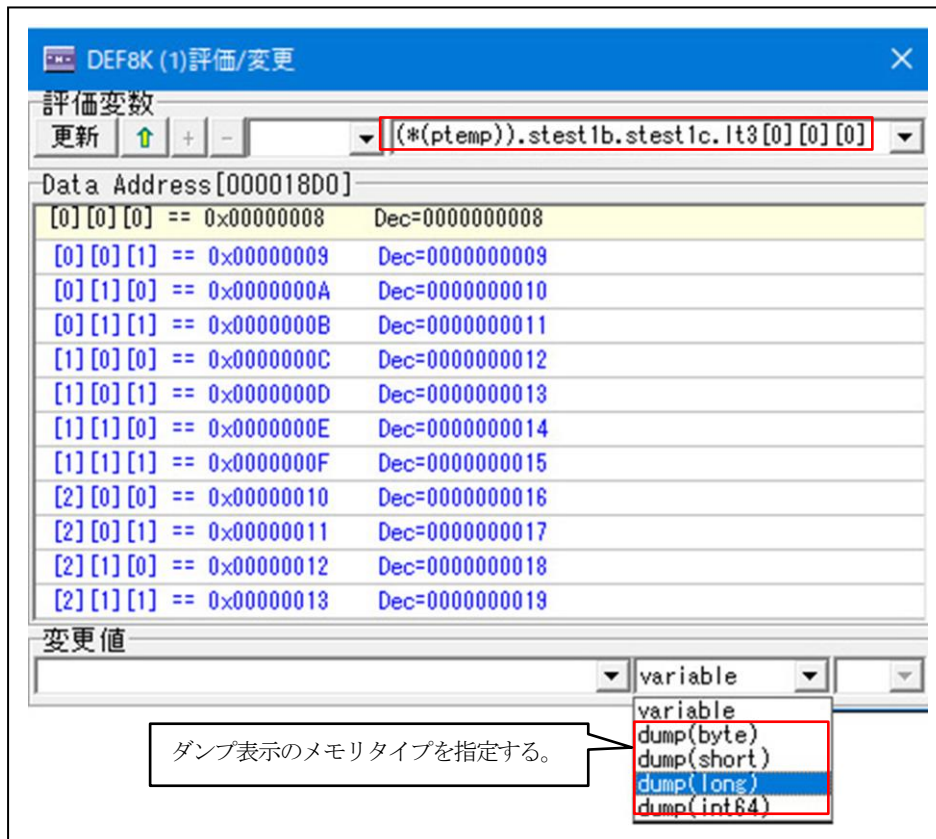
ネストしているメンバを参照する場合は、「ダブルクリック」する。

[6-5]

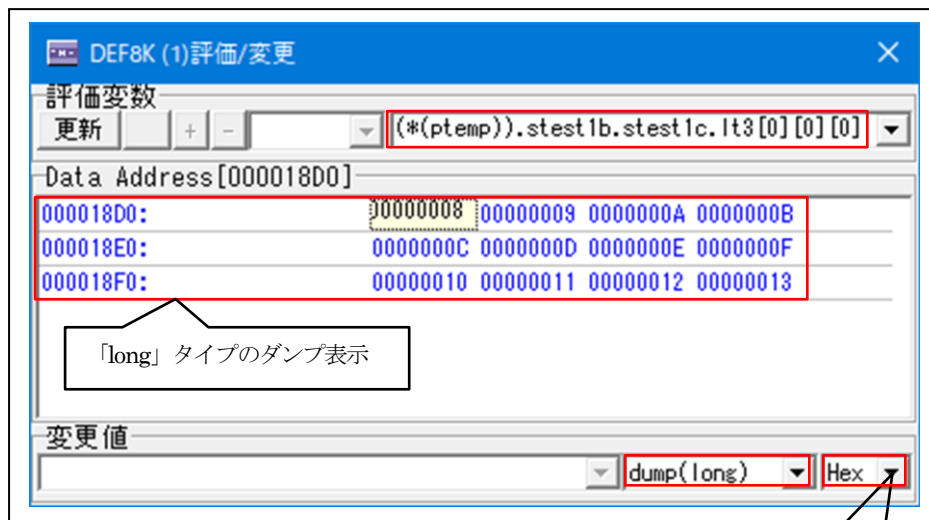


ダンプ表示に切り替えする場合は、
ここの「PullDown」で指定する。

[6-6]



[6-7]



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。

〒486-0852
愛知県春日井市下市場町 6-9-20
エーワン株式会社
<https://www.aone.co.jp>

