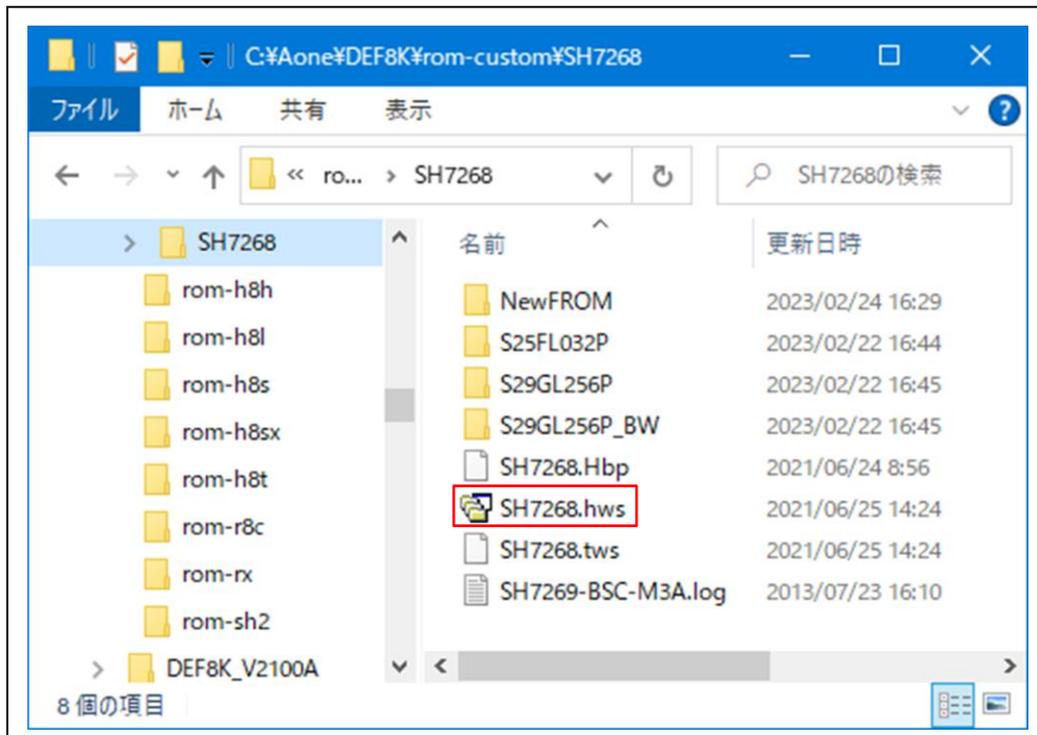


ROM レス品種にて、外付けシリアル FlashROM 対応のローダプログラム を追加する場合の説明

1. 対象 MCU
 - ・ROM レス品種が対象です。(SH7262/7264/7266/7267/7268/7269)
2. 機能
 - ・Hew4にてシリアル FlashROM の品種追加が出来るよう対応する。
 - ・FlashROM 仕様に合わせたバイトリードをプログラミングするだけで追加が可能になるよう対応する。
3. ローダー追加前の準備

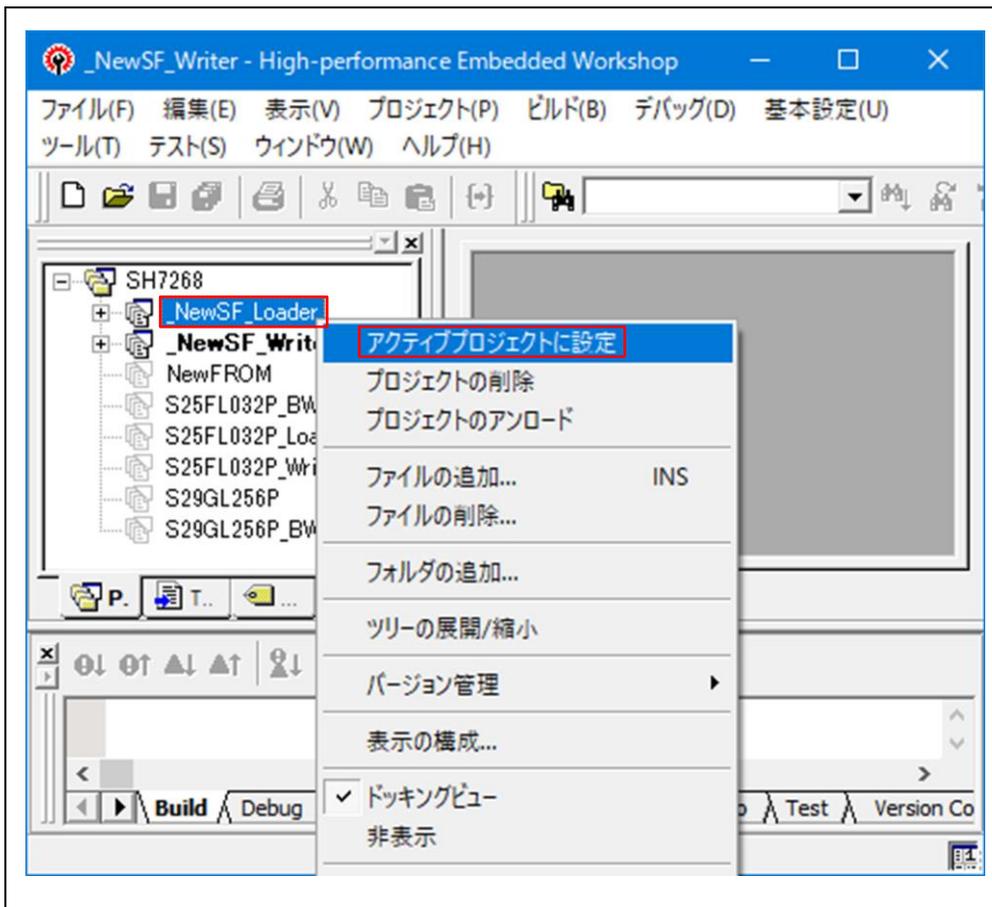
3-1) AH8000 コントロールソフト(DEF8K)のインストール DIR にあるワークスペースを開く。
(SH7268 での追加例)



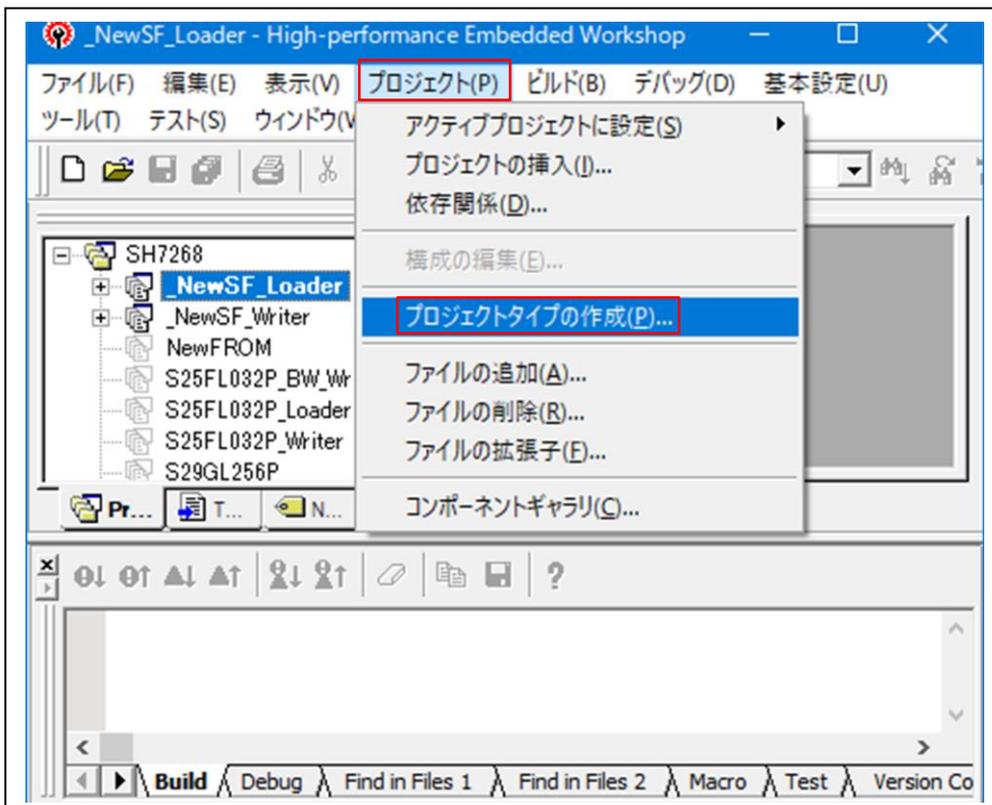
<default directory>

"c:\Aone\DEF8K\rom-custom\SH7268"

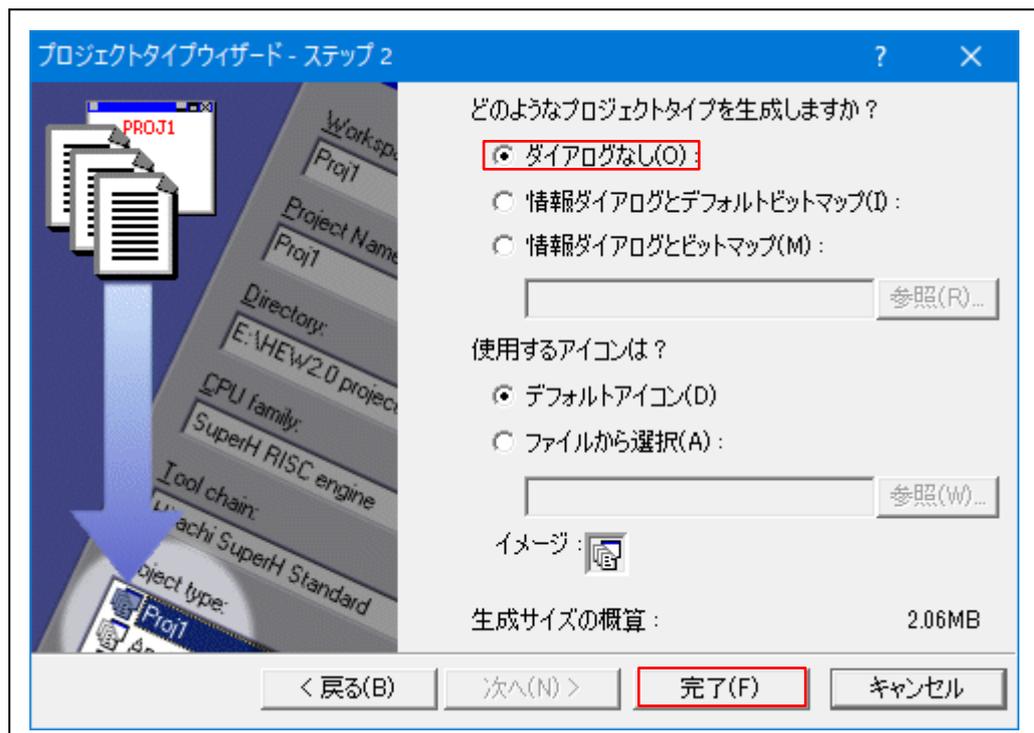
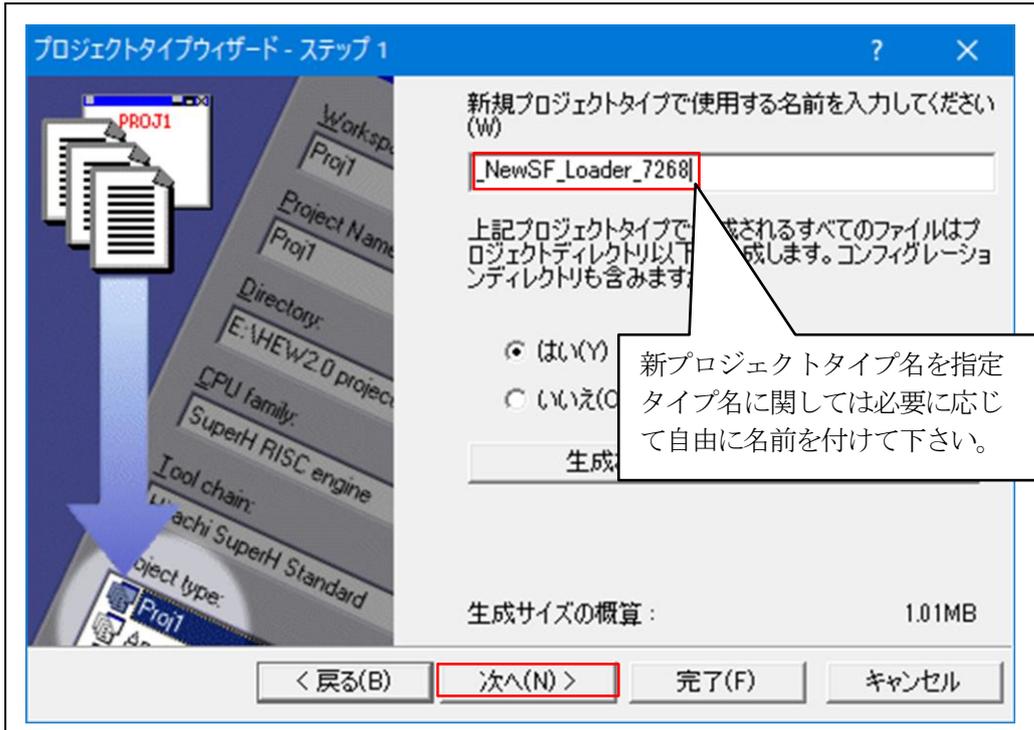
3-2) プロジェクト名「_NewSF_Loader」をアクティブプロジェクトにする。



3-3) 「プロジェクトタイプの作成」を指定する。

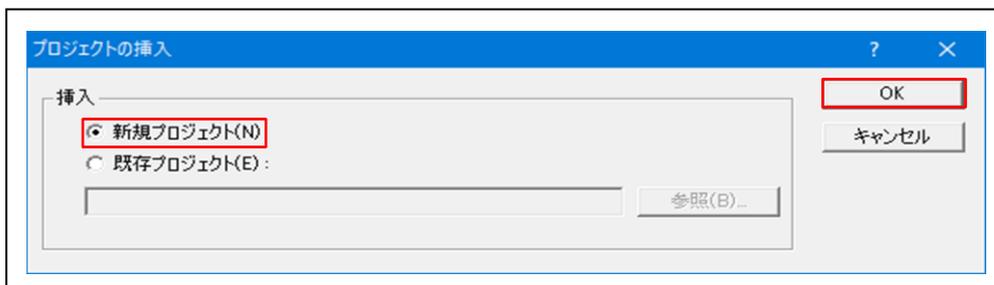
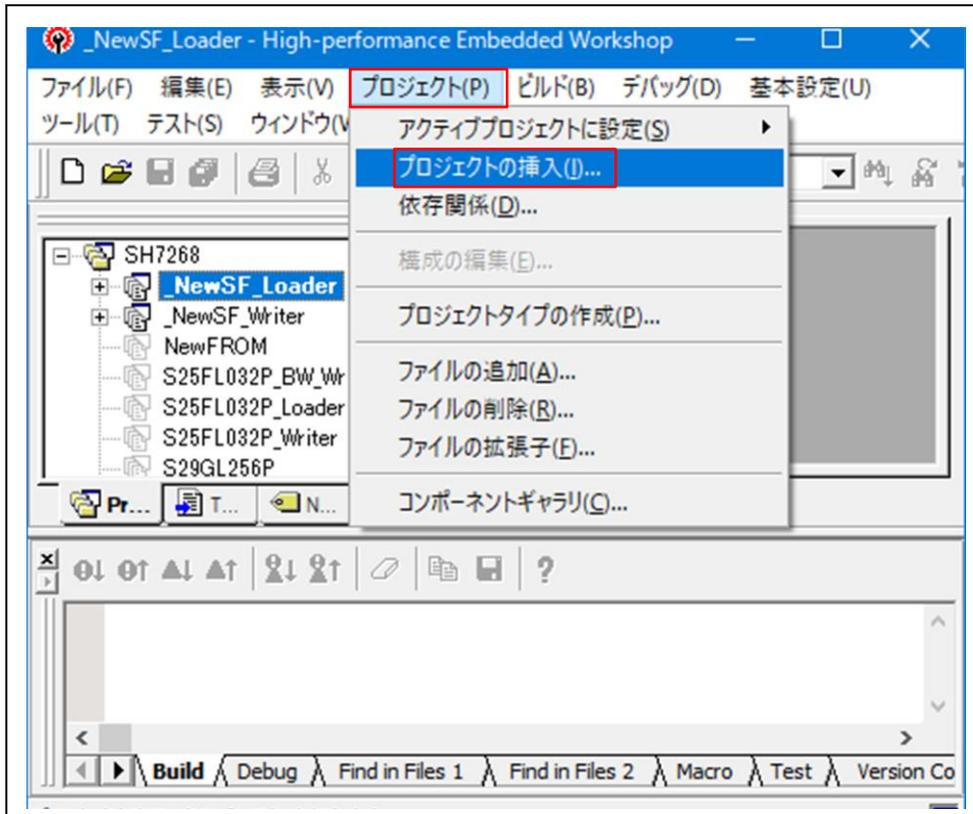


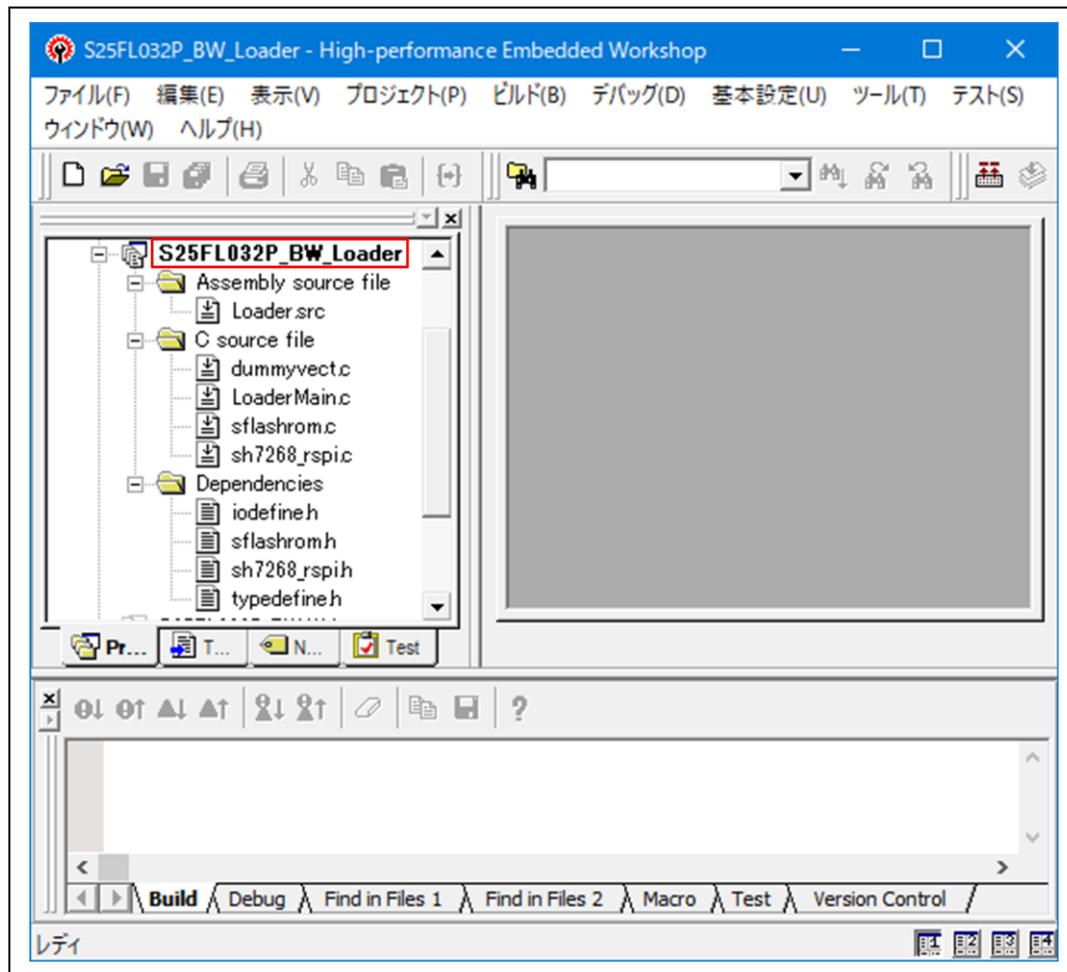
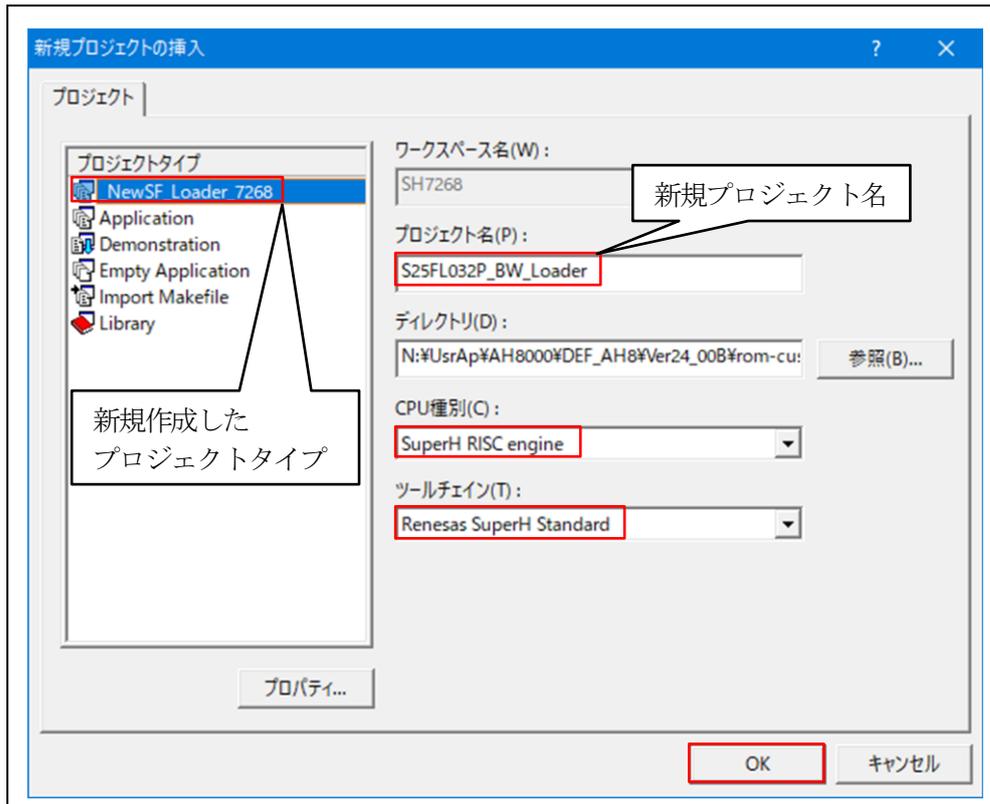
3-4) 追加FlashROMの新プロジェクトタイプを作成する。



4. シリアル FlashROM 用ローダーの追加

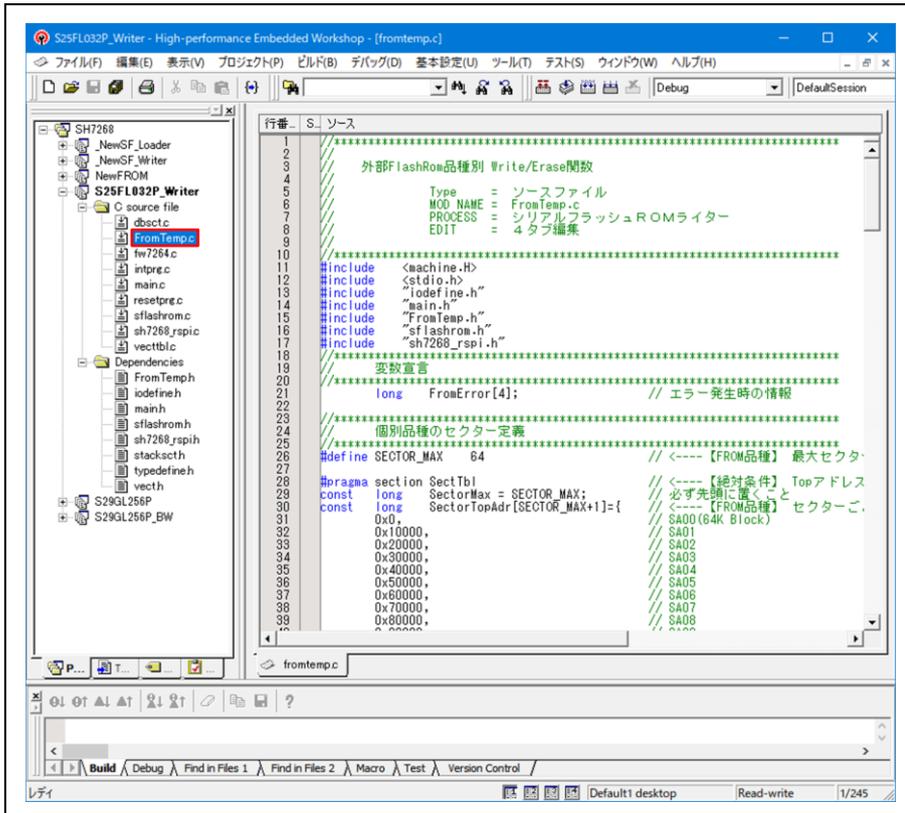
4-1) 新規プロジェクトの挿入



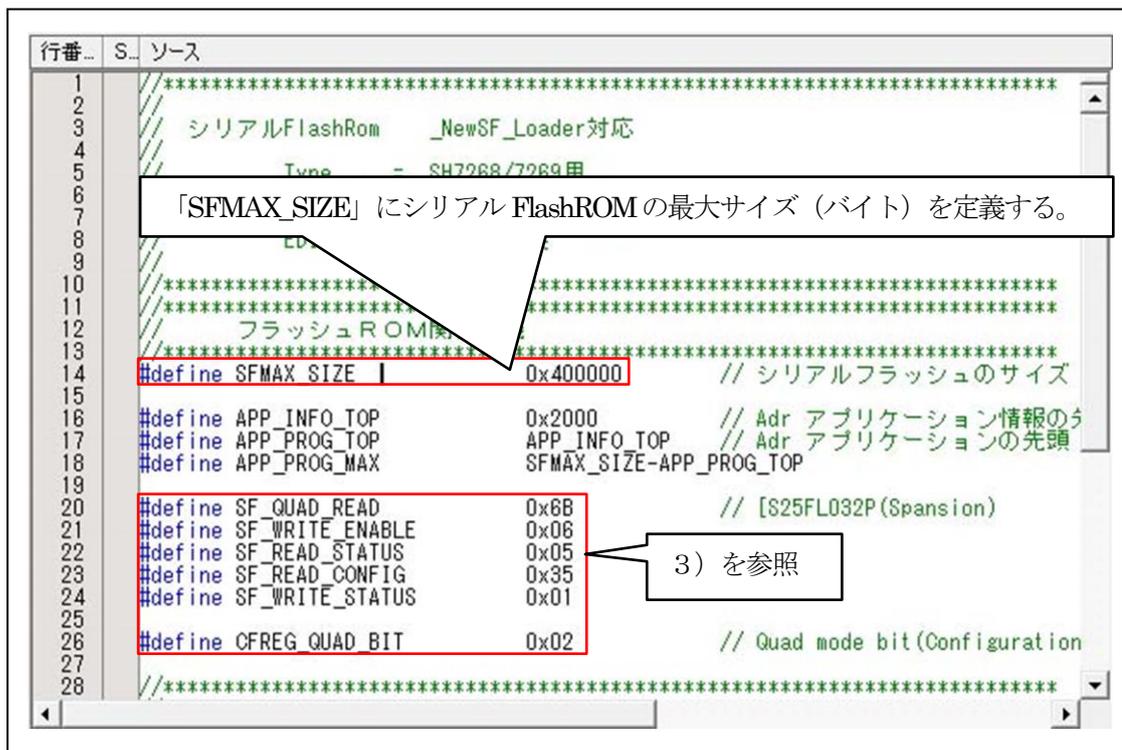


4-2) シリアル FlashROM 用ローダーの定義およびプログラムを作成する。

1) 「sflashrom.c」を開く



2) 「sflashrom.h」の#defineを定義する。



3) 「sflashrom.h」の#defineの定義説明。

シリアル FlashROM の各コマンドを定義します。

定義名	コマンド	機能
SF_QUAD_IO_READ	0xEB	QUAD I/O 高性能読み取り
SF_WRITE_ENABLE	0x06	Write 許可
SF_READ_STATUS	0x05	ステータスリード
SF_READ_CONFIG	0x35	コンフィグレーションリード
SF_WRITE_STATUS	0x01	ステータスライト
CFREG_QUAD_BIT	0x02	QUAD モードビット (Configuration)

この例は、Spansion の「S25FL032P」のコマンドです。コマンド仕様は各デバイスメーカーによって異なります。使用するデバイスのデータシートを参照して下さい。

4) 「sFlashSetMode()」関数を確認または変更する。

【機能】 QUAD モードに設定する。

```

行番... S... ソース
33 //*****
34     sFlashSetModet    sFlash モードセット
35 //*****
36 //*****
37 void    sFlashSetMode(void)
38 {
39     set_mode(SF_REQ_QUADMODE);
40 }
41 //*****
  
```

5) 「AppInfoRead」関数を確認または変更する。

```

行番... S... ソース
40 //*****
41     AppInfoRead    アプリケーション情報 READ
42 //*****
43 //*****
44 void    AppInfoRead(UCHAR *rdat,ULONG rsize)
45 {
46     sFlashByteRead(APP_INFO_TOP,rdat,rsize);
47 }
48 //*****
  
```

関数「AppInfoRead(...)」は、アプリケーションのローダ情報をシリアル FlashROM の「APP_INFO_TOP(0x2000)」番地より読み出します。

<引数の仕様>

UCHAR *rdat:// 読み出したデータを格納する先頭アドレス

ULONG rsize:// 読み出しデータのサイズ (バイト)

6) 「AppProgRead」関数の確認または変更する。

```

行番... S... ソース
48 //*****
49 AppProgRead      アプリケーションプログラム READ
50 //*****
51 return(0) : 正常終了
52 return(-1) : 異常終了
53 //*****
54 int AppProgRead(UCHAR *rdat,ULONG rsize)
55 {
56     if (rsize <= APP_PROG_MAX) {
57         sFlashByteRead(APP_PROG_TOP,rdat,rsize);
58         return(0);
59     }
60     return(-1);
61 }
62 //*****
  
```

関数「AppProgRead(...)」は、アプリケーションのローダ情報をシリアル FlashROM の「APP_PROG_TOP(0x2000)」番地より読み出します。

<引数の仕様>

UCHAR *rdat; // 読み出したデータを格納する先頭アドレス

ULONG rsize; // 読み出しデータのサイズ (バイト)

7) 「sFlashByteRead」関数の確認または変更する。

```

行番... S... ソース
62 //*****
63 sFlashByteRead   指定バイト数の読み込み
64 //*****
65 //*****
66 void sFlashByteRead(ULONG adr,UCHAR *rdat,ULONG rsize)
67 {
68     LONG unit;
69
70     if ((rsize % 4) == 0) {unit = 4;}
71     else if ((rsize % 2) == 0) {unit = 2;}
72     else {unit = 1;}
73
74     while(rsize > 0) {
75         read_data_quad(adr,rdat,unit);
76         adr += unit;
77         rdat += unit;
78         rsize -= unit;
79     }
80 }
81 //*****
  
```

関数「sFlashByteRead(...)」は、シリアル FlashROM の指定アドレスから指定バイト数読み出します。

<引数の仕様>

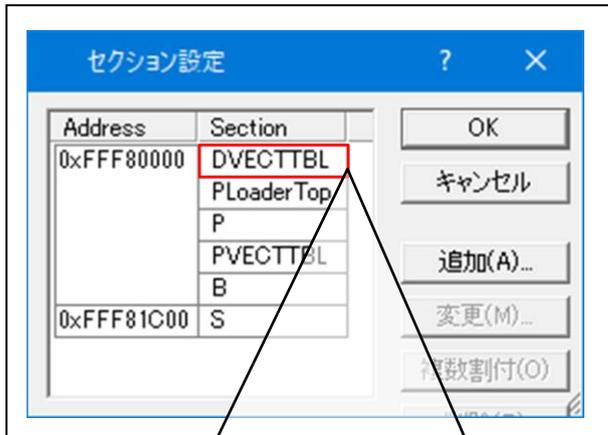
ULONG adr; // 読み出しするデバイス内の ROM の先頭アドレス

UCHAR *rdat; // 読み出したデータを格納する RAM の先頭アドレス

ULONG rsize; // 読み出しデータのサイズ (バイト)

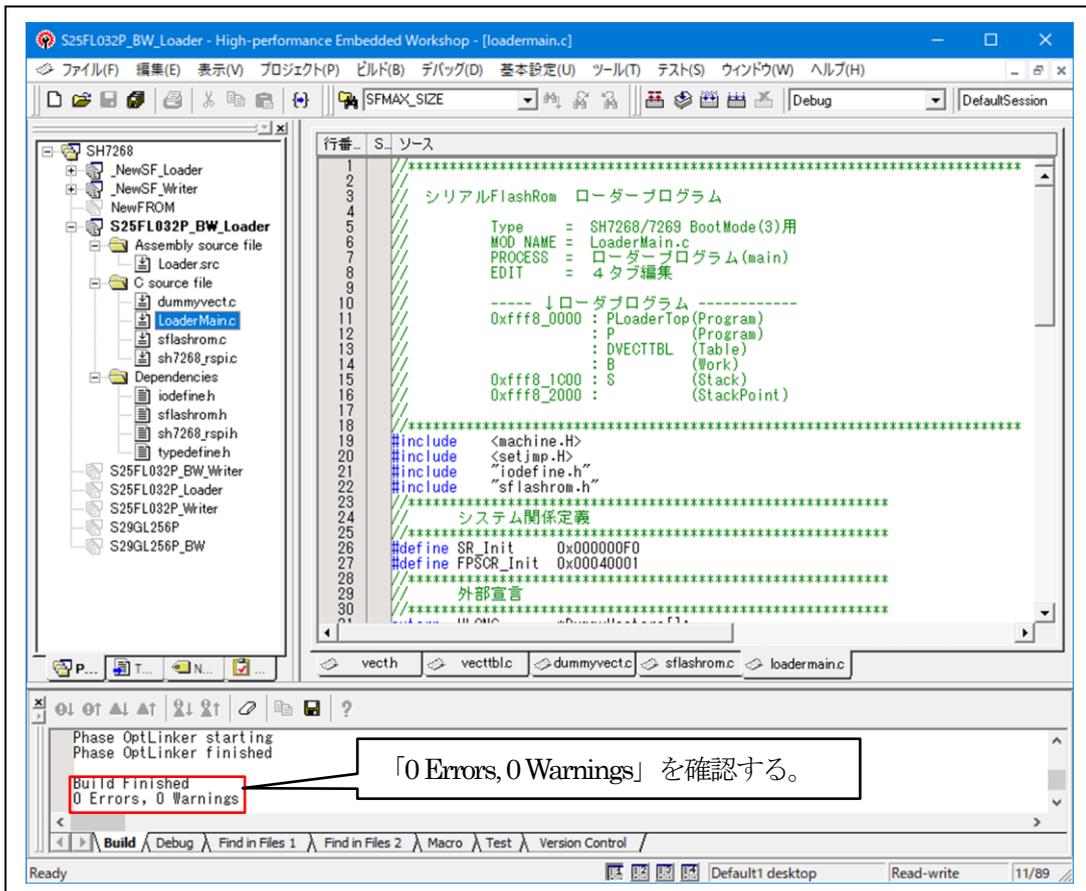
4-3) シリアル FlashROM 用ローダープログラムをデバッグ用にビルドする。

1) セクションの設定



<ルール 1>
 デバッグ中は、「DVECTTBL」セクションを内蔵 RAM の先頭にロケーションする。
 <ルール 2>
 スタックまで含めたオブジェクトサイズは「0x2000」8KB までにする。
 以上のルールは厳守して下さい。

2) ビルドの実施



5. シリアル FlashROM 用ローダープログラムのデバッグ方法

5-1) ターゲット基板側のモード設定

デバッグ中は、「ブートモード0」（CS0 空間に接続されたメモリからブートする）にして下さい。理由は、「ブートモード3」（シリアルフラッシュメモリからブートする）のままですとリセット解除後、ブートプログラムが起動され内蔵 RAM の先頭から「0x2000」エリアにローダプログラムがロードされてしまうからです。

5-2) 「DEF8K」の「CPU 設定」にデバッグ環境を設定する。

作成した「FlashROM」プログラムは、ターゲットの内蔵 RAM に転送して実行させますので「RAM」側を選択して先頭アドレスとサイズを設定する。

SH7268 の場合
開始 0xffff8_0000 サイズ 0x10000 になる。

ターゲット側の基板構成による

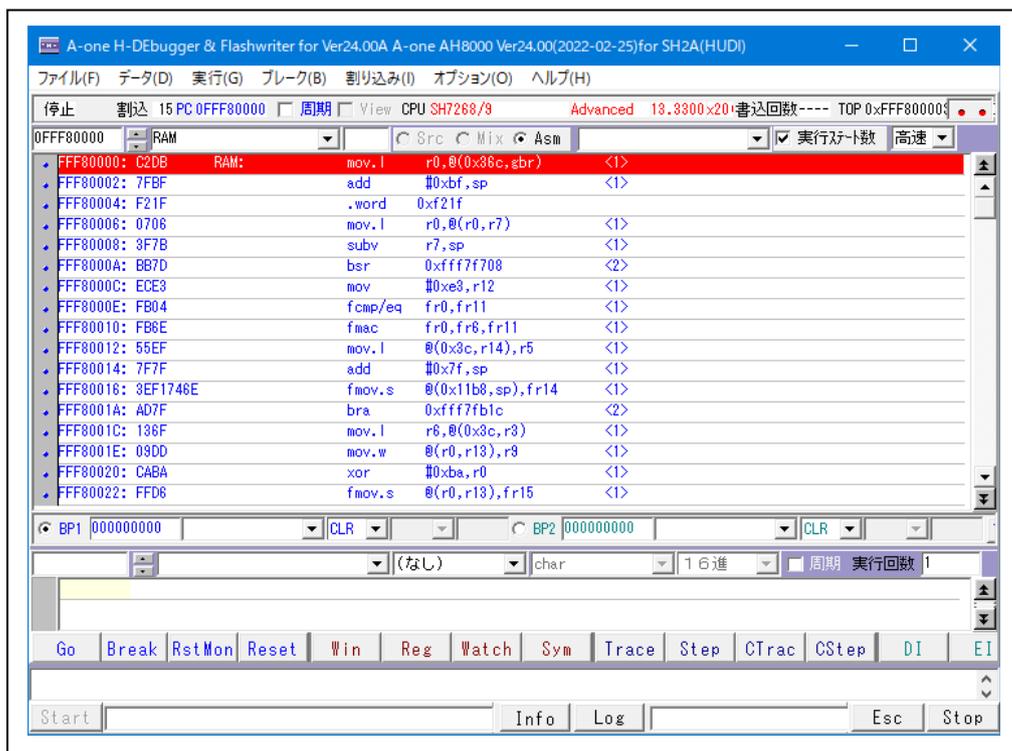
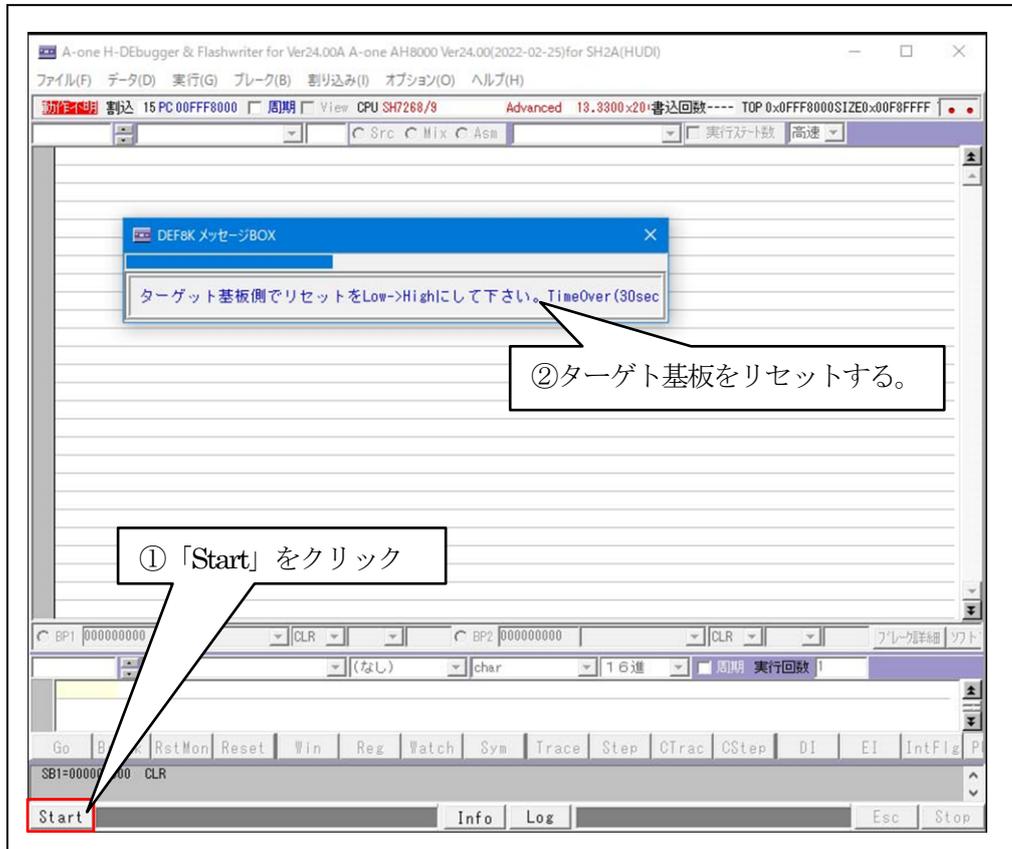
作成した「FlashROM」プログラムのデバッグに BSC 設定が必要な場合は、スクリプトファイルを指定する。

デバッグ時は、「Soft-TAP」を指定する。

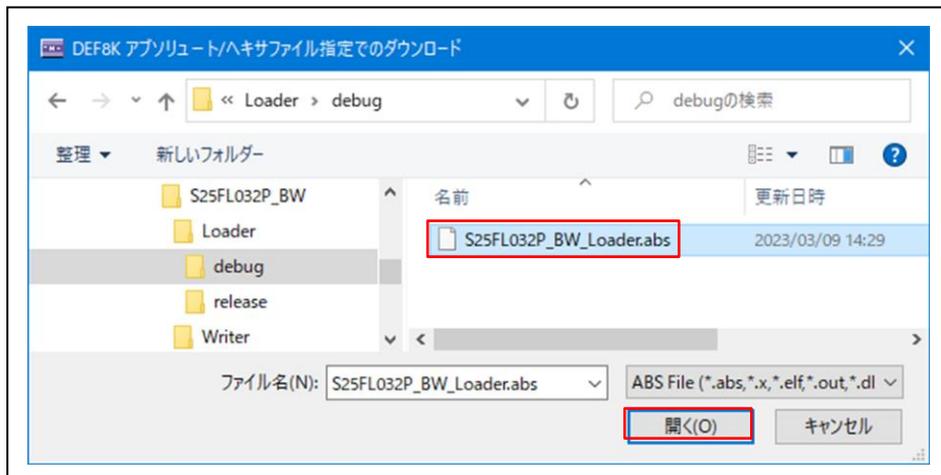
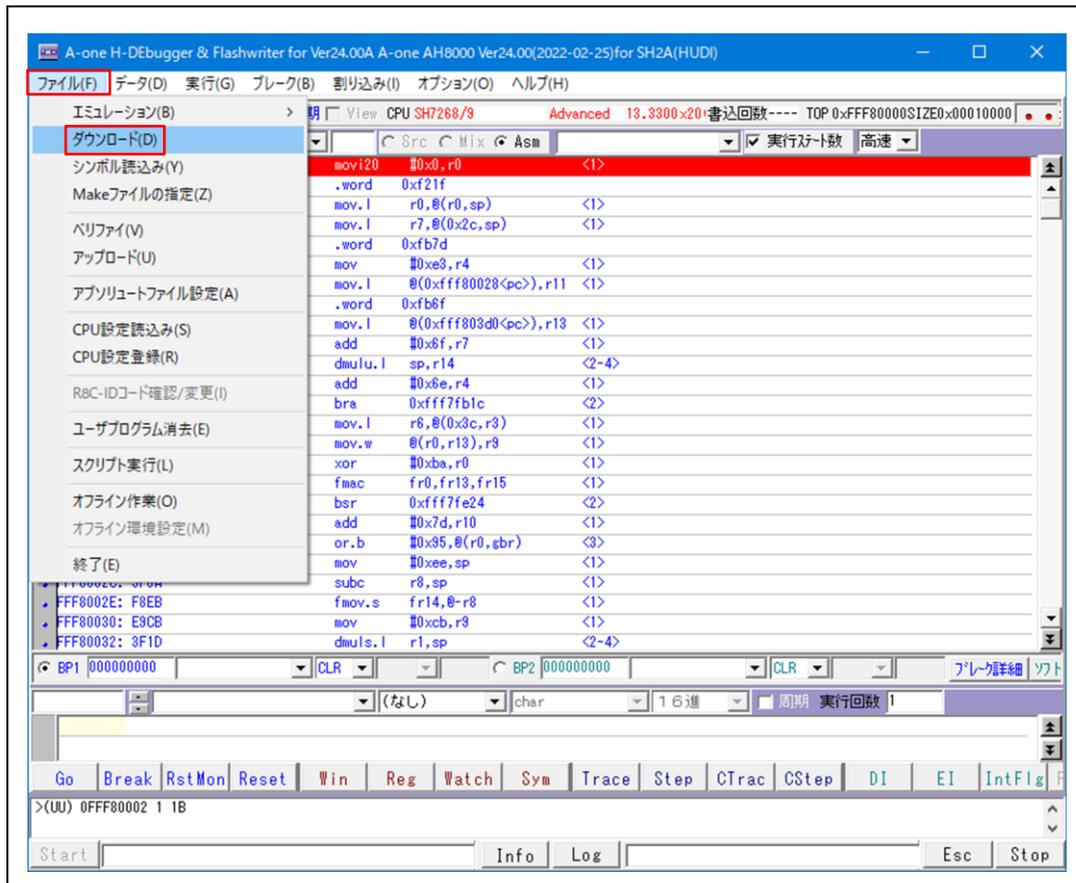
5-3) 「DEF8K」でデバッグを開始する。

【条件】デバッグできる条件は、シリアル FlashROM の 0x2000 番地から何らかのアプリケーションプログラムが書き込まれている必要があります。

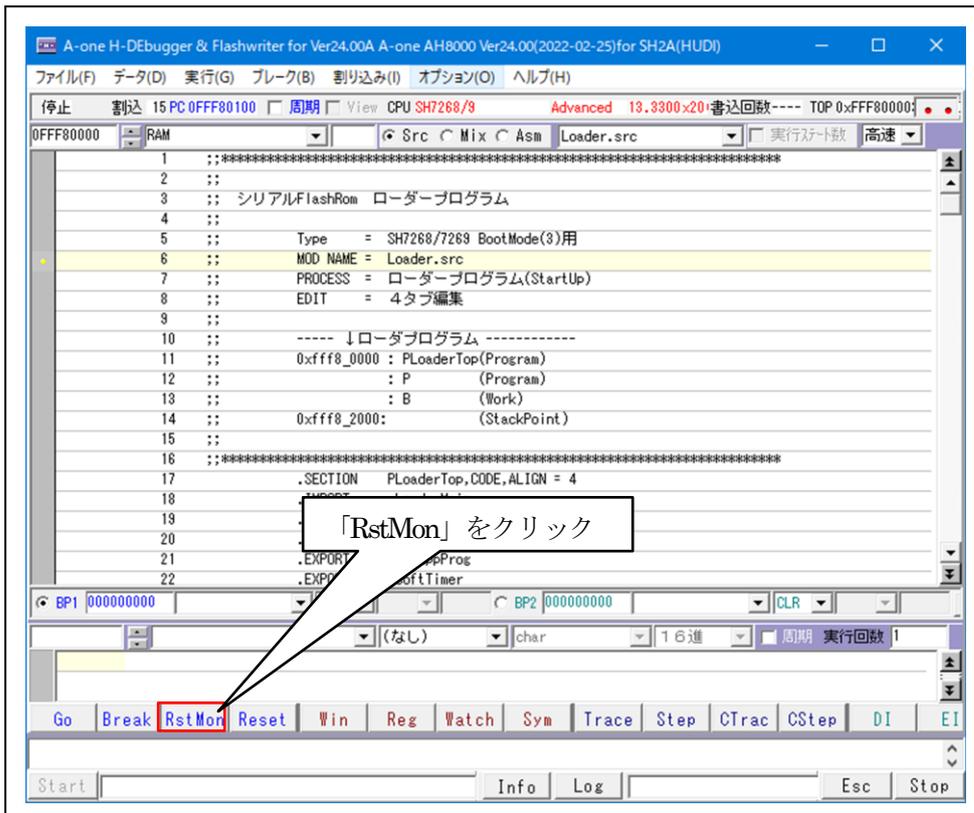
1) ターゲットと接続する。



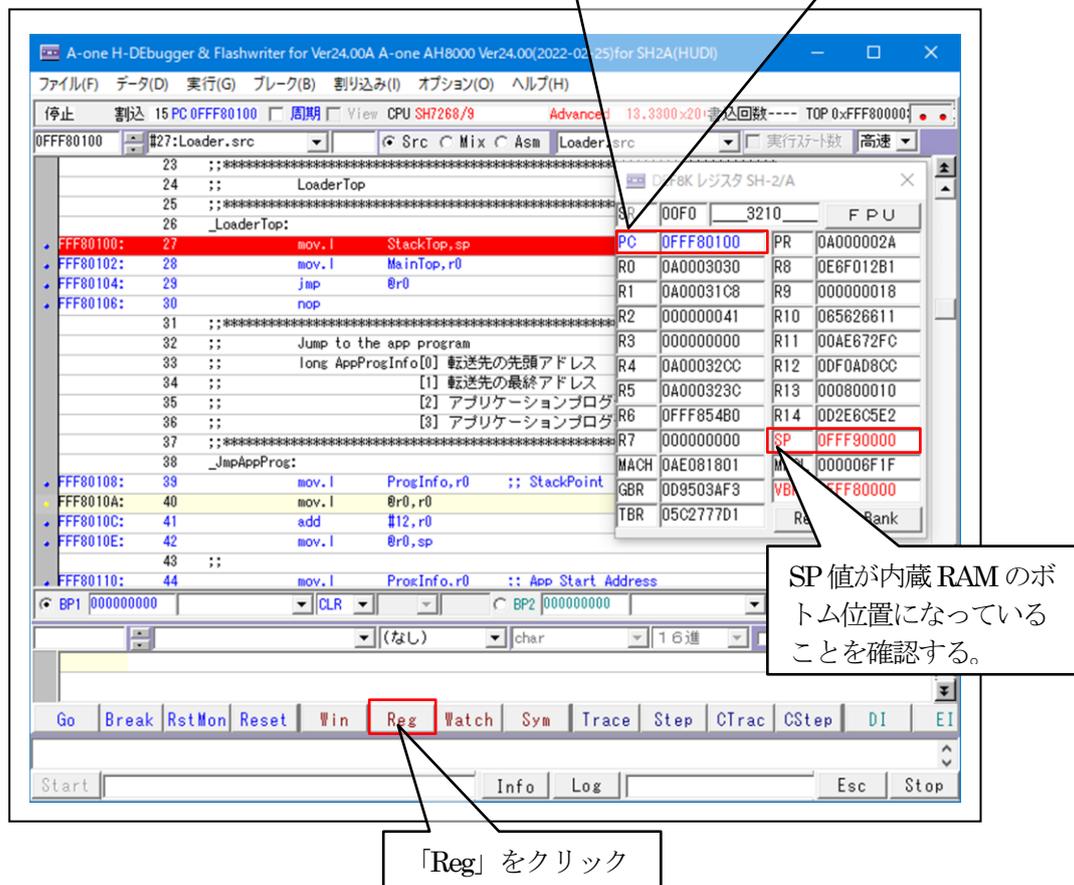
2) シリアルFlashROM用ローダープログラムをダウンロードする。



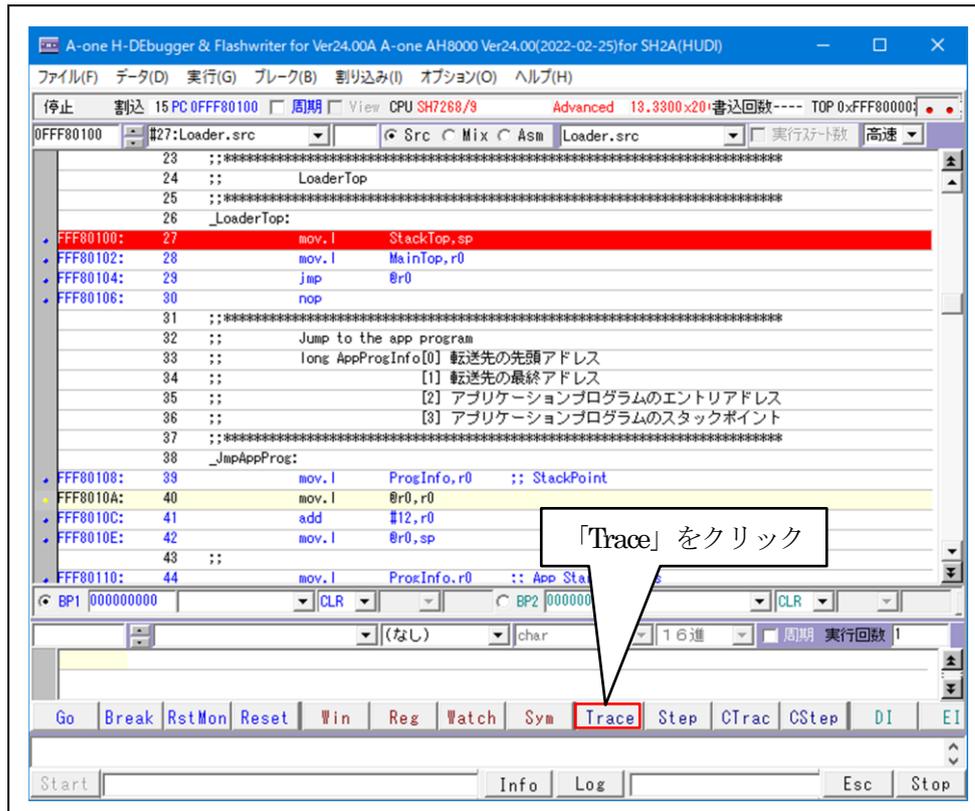
3) 「RstMon」 をクリックする。



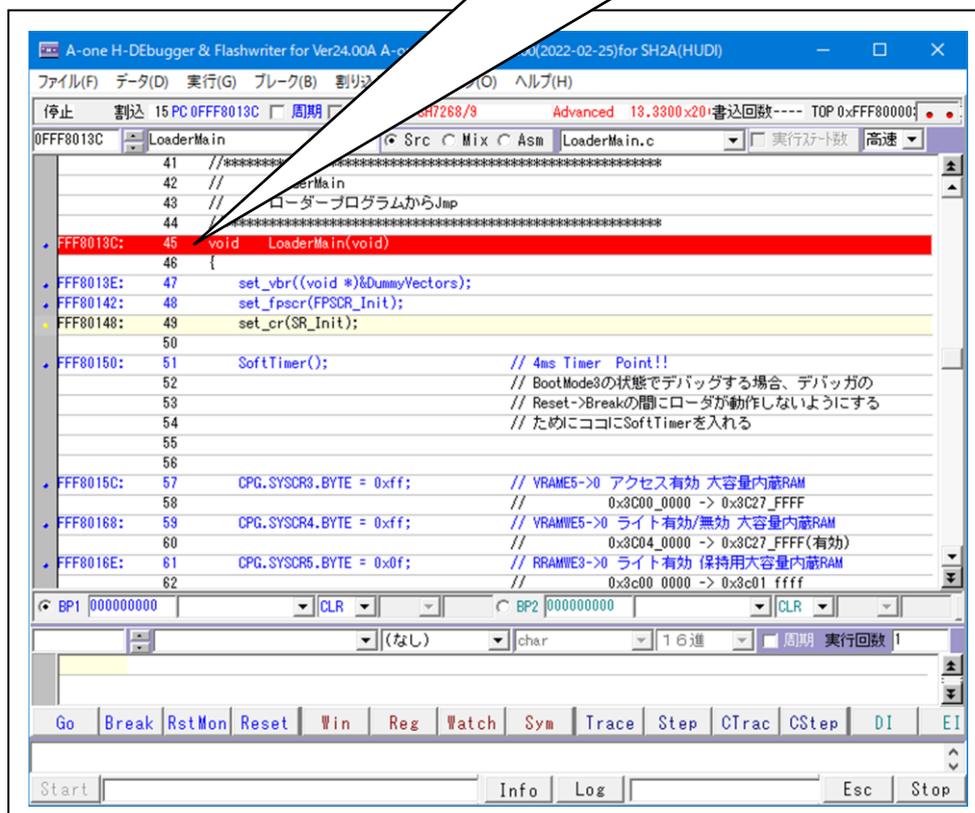
「_Loader_TOP」関数(0xFFFF80100)の先頭の PC 値になっている事を確認する。



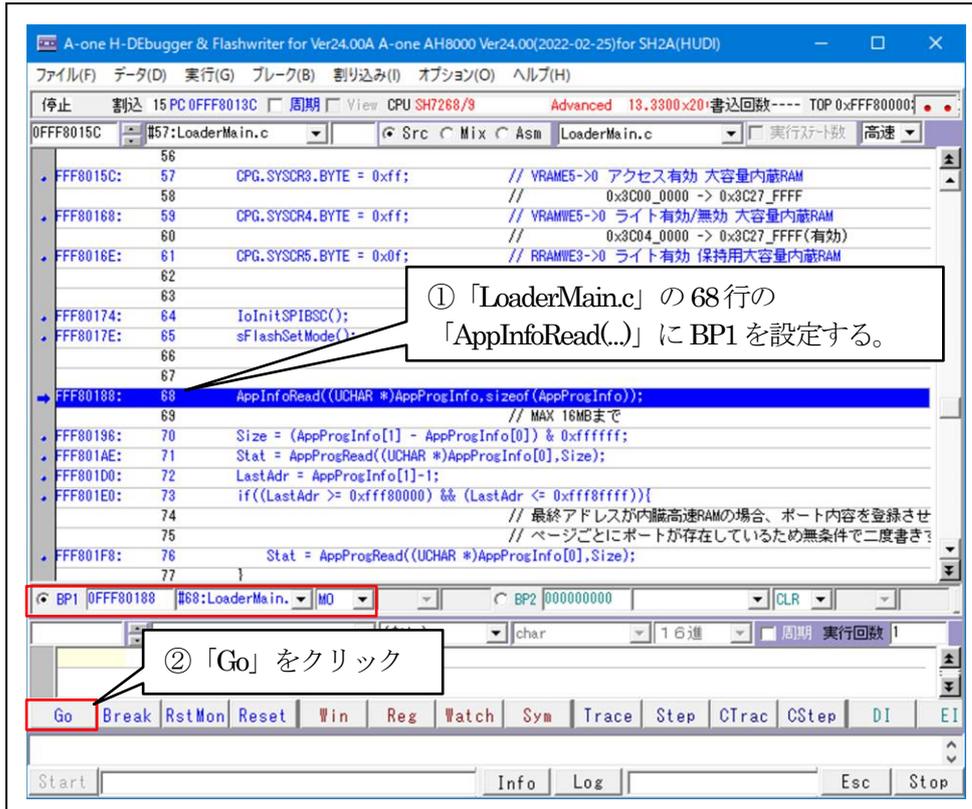
4) 「Trace」 をクリックして 「LoaderMain0」 まで PC を進める。



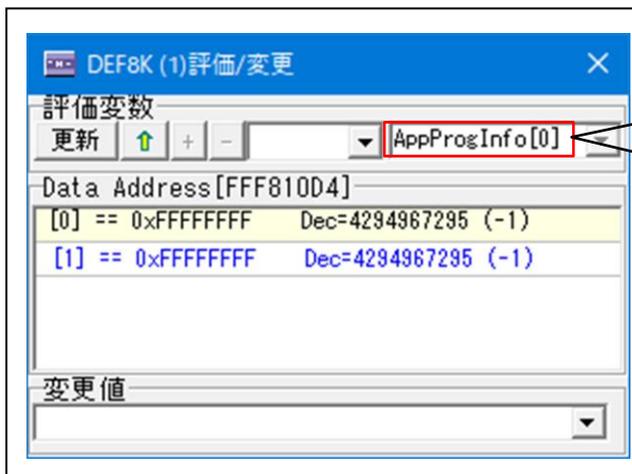
「LoaderMain0」 まで進んだことを確認する。



5) 「AppInfoRead0」関数を確認するため準備する。

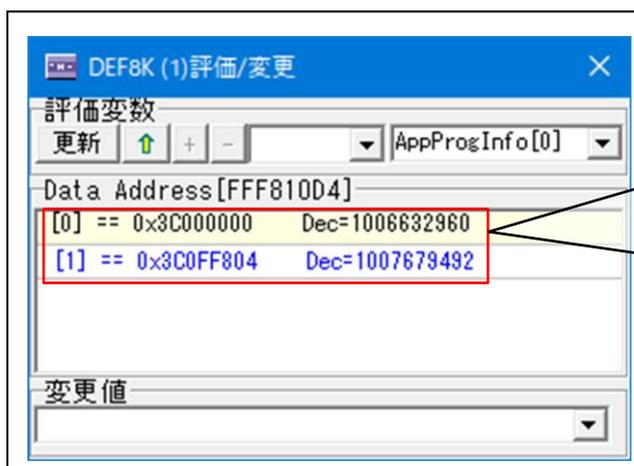
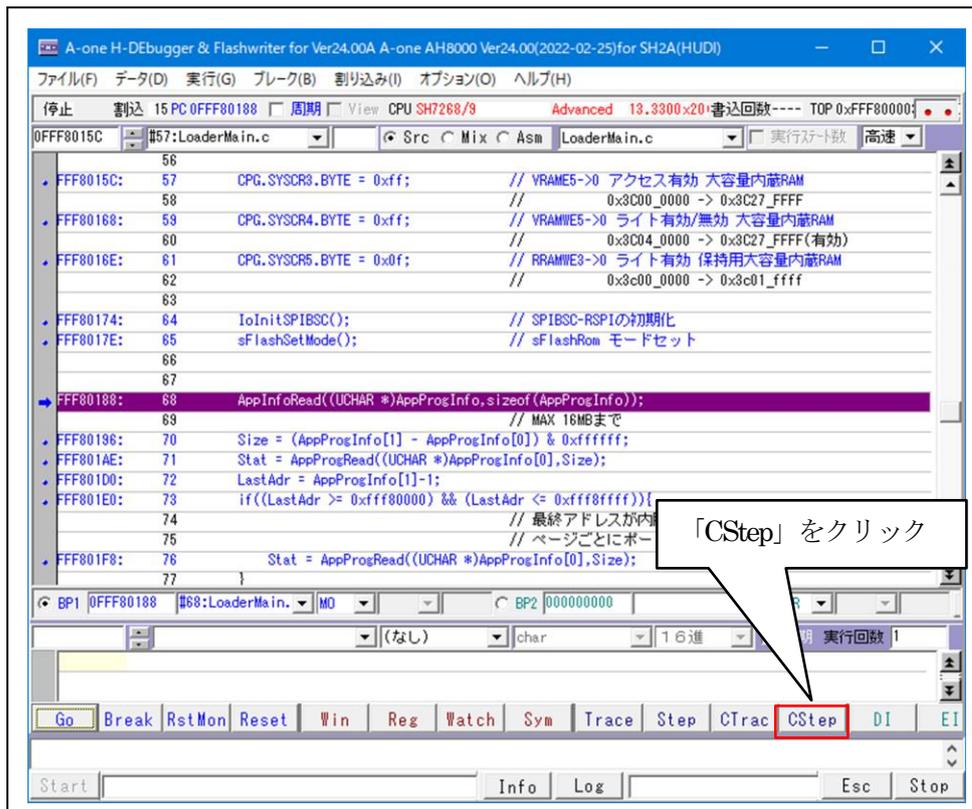


【「評価/変更」を開く】



「AppInfoRead()」関数によつて、変数「AppProgInfo[2]」にアプリケーションのローダ情報がセットされる。

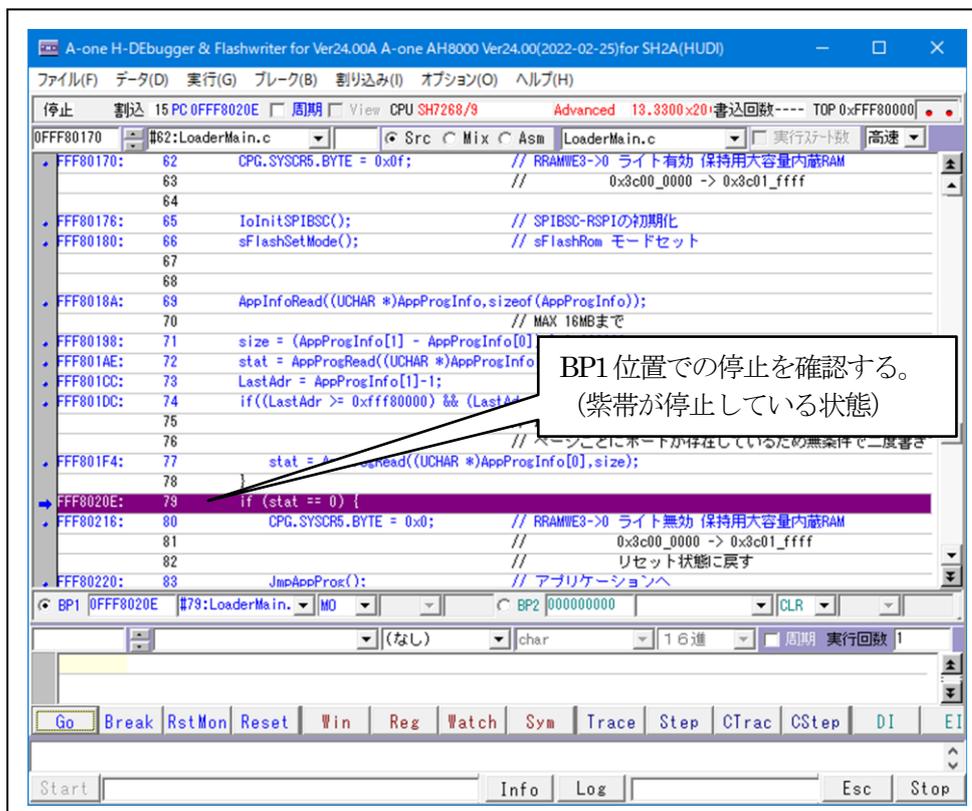
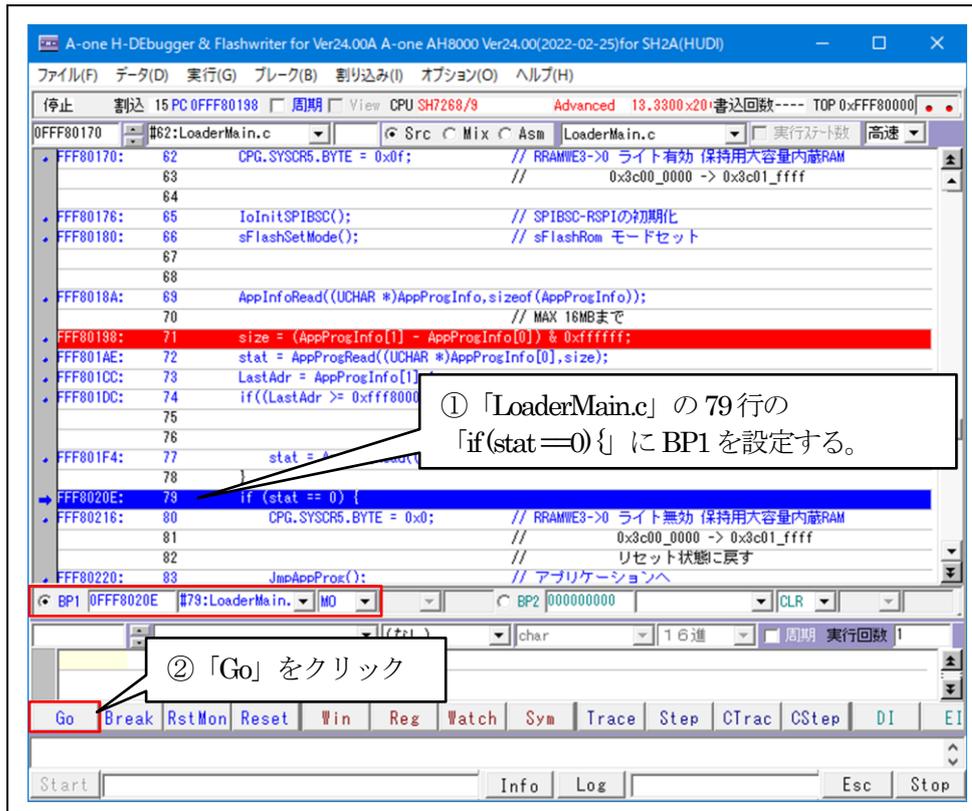
6) 「AppInfoRead0」関数を実行する。



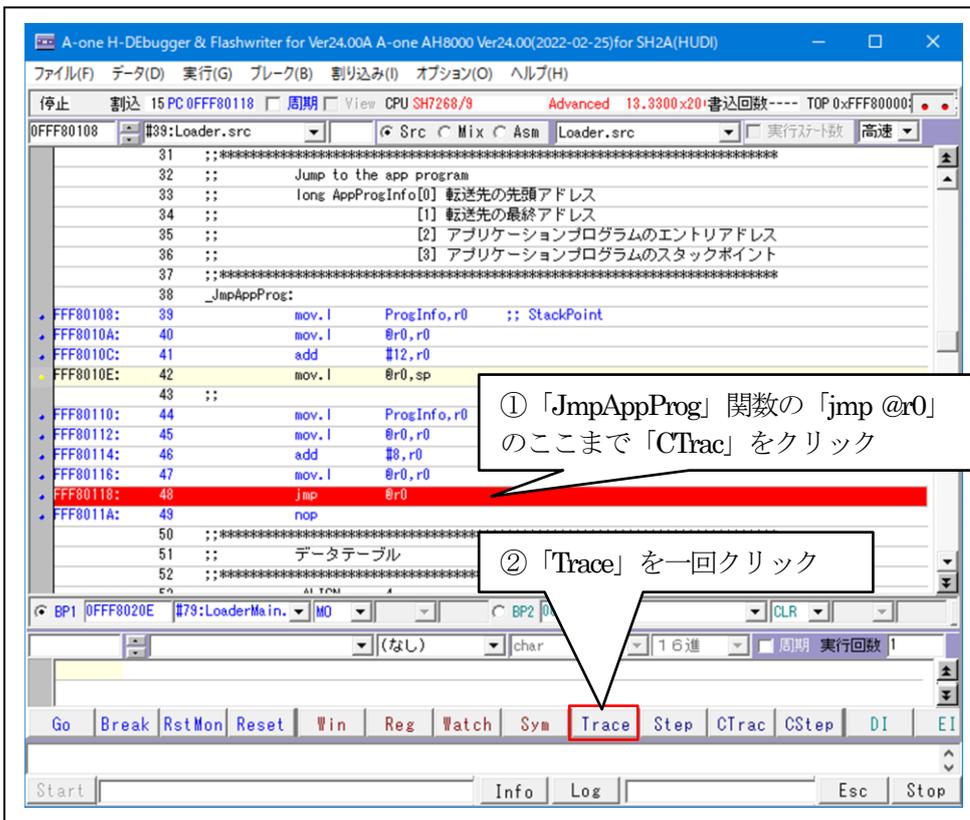
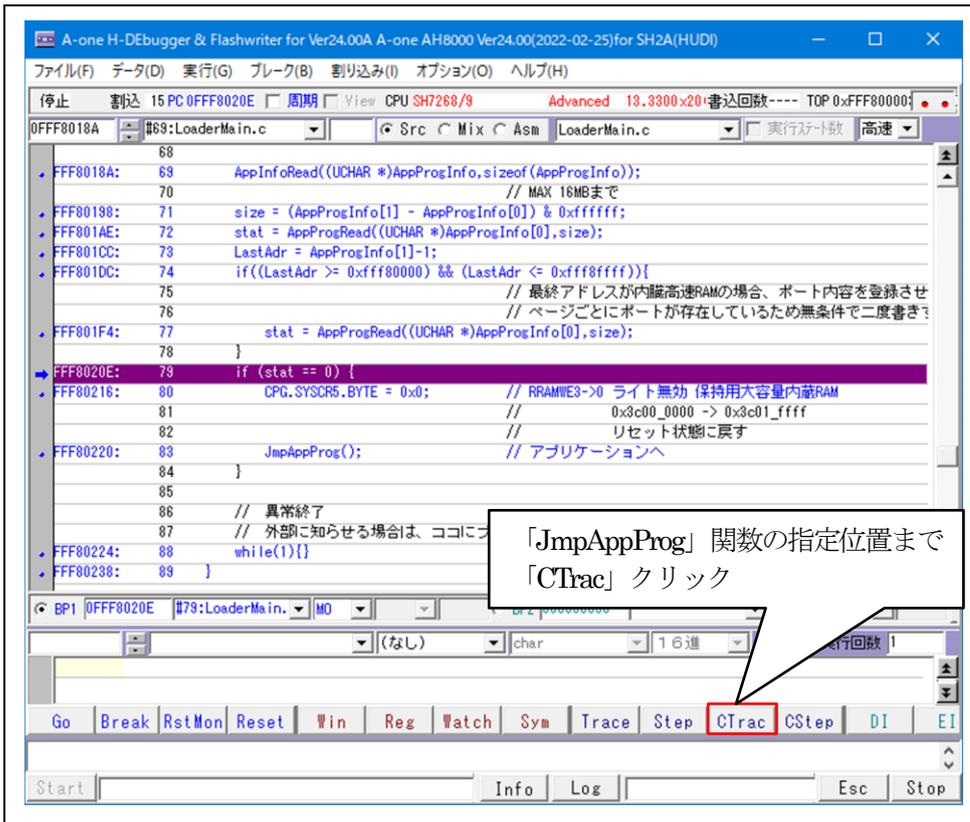
[0]は、アプリケーションプログラムを内蔵メモリへロードする先頭アドレスになる。

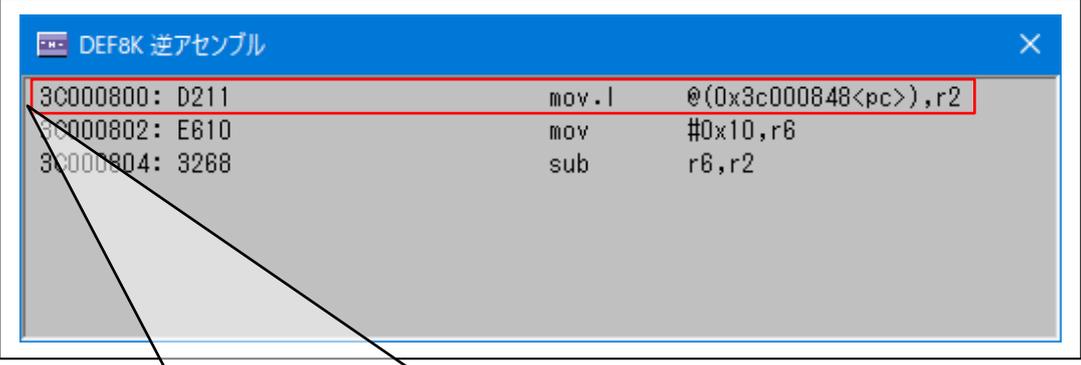
[1]は、アプリケーションプログラムを内蔵メモリへロードする終了アドレスになる。

7) アプリケーションプログラムが「AppProgInfo」を元に内蔵 RAM にロードされるか確認する。



9) アプリケーション側にPCが切り替わることを確認する。





```

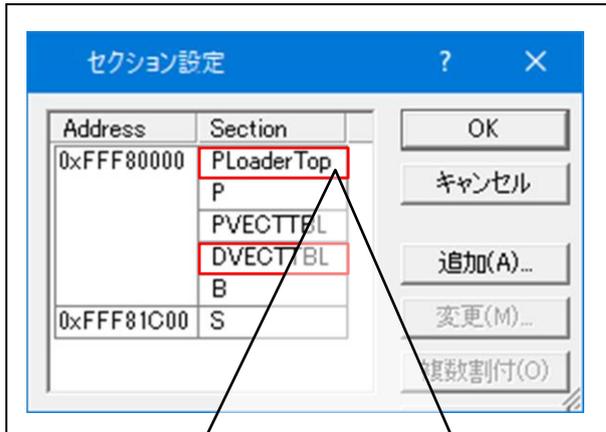
DEF8K 逆アセンブル
3C000800: D211      mov.l    @(0x3c000848<pc>),r2
3C000802: E610      mov     #0x10,r6
3C000804: 3268      sub     r6,r2
  
```

逆アセンブル表示でアドレス「0x3C00_0800」のニモニックは表示されたらアプリケーション側に切り変わったことの確認になる。これ以上の実行は中止して下さい。

以上でシリアル FlashROM 用ローダープログラムの動作確認は終了です。

6. シリアル FlashROM 用ローダープログラムを正規リリース用にビルドする。

6-1) セクションの設定



<ルール 1>

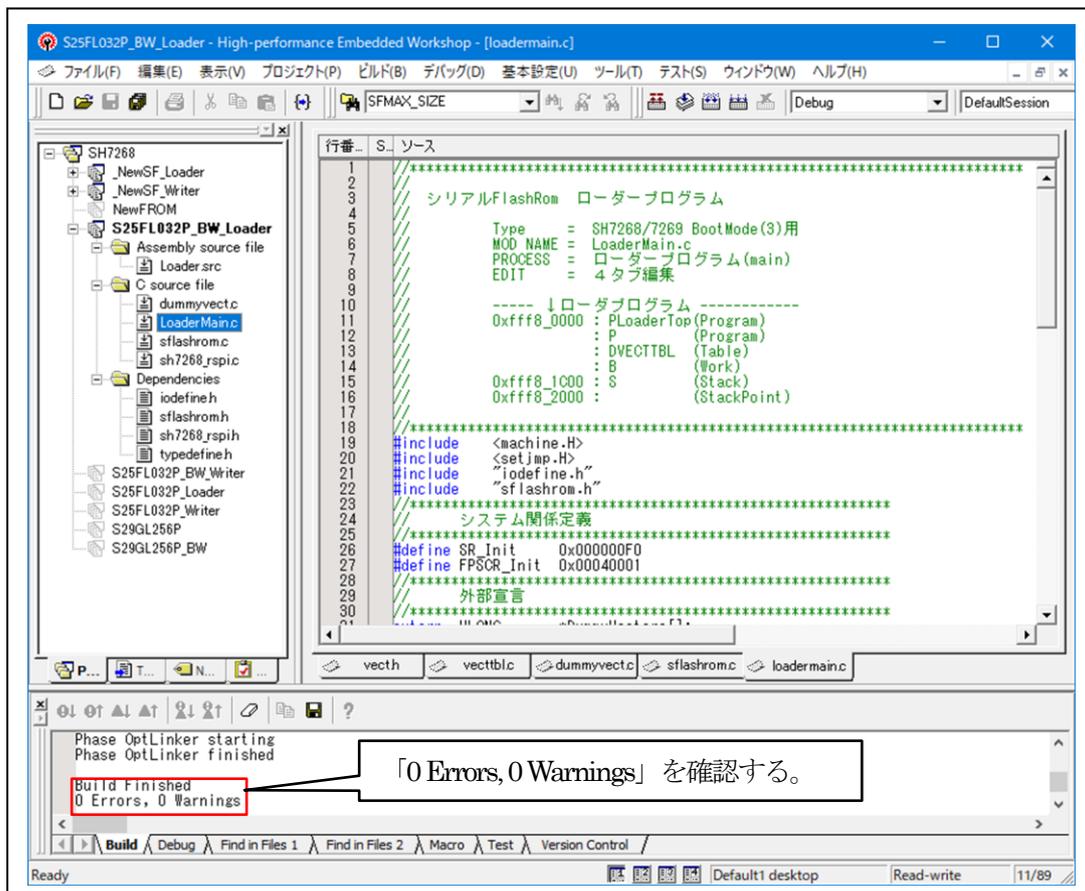
正規リリースは、「PLoaderTip」セクションを内蔵 RAM の先頭にロケーションする。

<ルール 2>

スタックまで含めたオブジェクトサイズは「0x2000」8KB までにする。

以上のルールは厳守して下さい。

6-2) ビルドの実施



6-3) 「DEF8K」の「CPU設定」にリリース環境を設定する。

1) 「RAM」を選択する。

「ローダとアプリケーションの登録」に登録した「シリアルFlashROMプログラム」がパケット通信対応の場合は選択が可能になる。(128/256/512byte)



開始：内蔵RAMの先頭アドレス+8番地を設定する。
 サイズ：内蔵RAMの最大サイズを設定する。

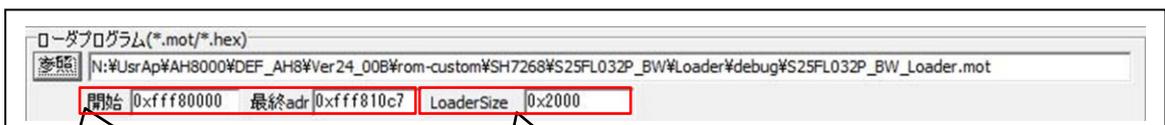
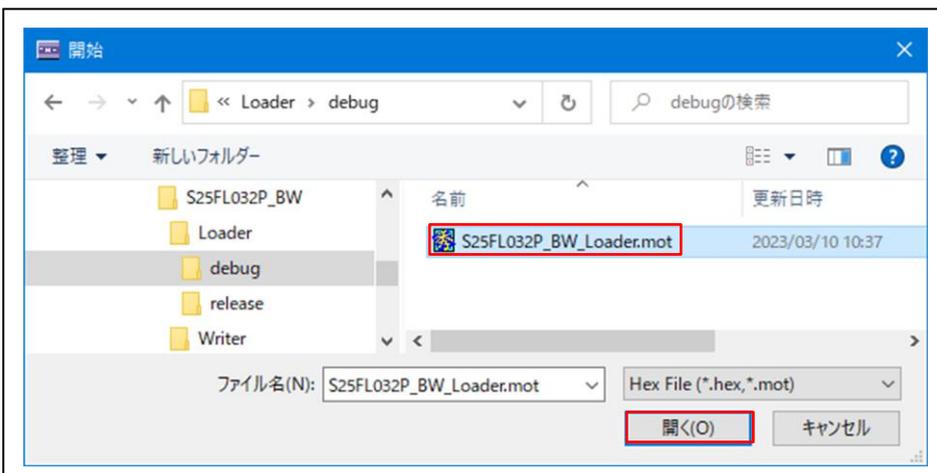
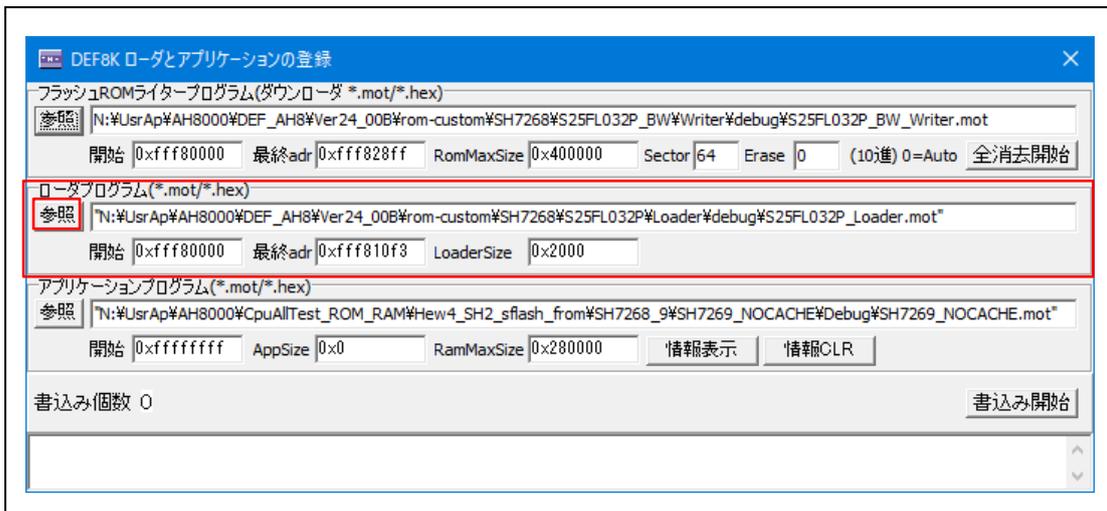
<備考>

アプリケーションの開始アドレスが内蔵RAMの先頭アドレスと終了アドレスがセットされているため。



6-4) 「DEF8K」の「ローダとアプリケーションの登録」に登録する。

1) ローダプログラムの「参照」をクリックする。



シリアルFlashROM用ローダプログラムの開始アドレスと最終アドレスを表示する。

最大ローダプログラムサイズを表示する。

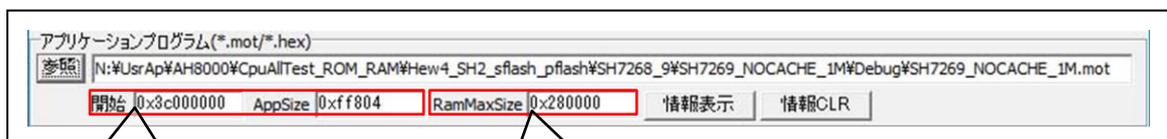
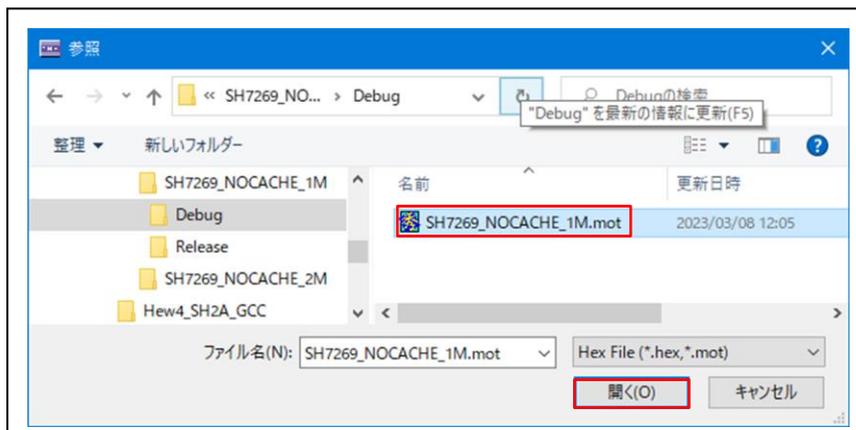
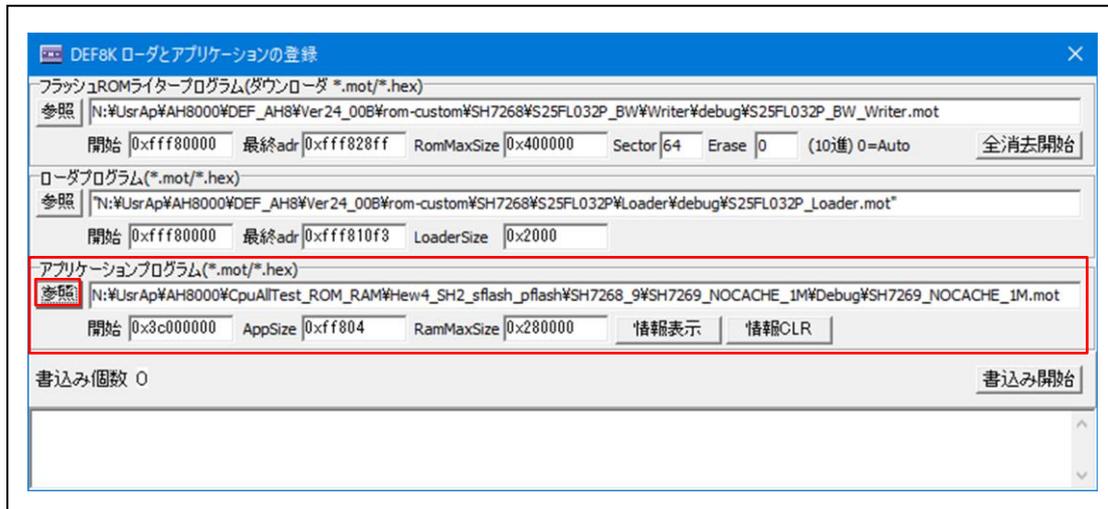
7. シリアル FlashROM 用ローダープログラムの最終確認をする。

7-1) ターゲット基板側のモード設定

「ブートモード3」(シリアルフラッシュメモリからブートする)に設定する。

7-2) 「ローダとアプリケーションの登録」で確認する。

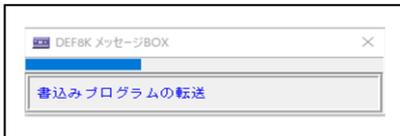
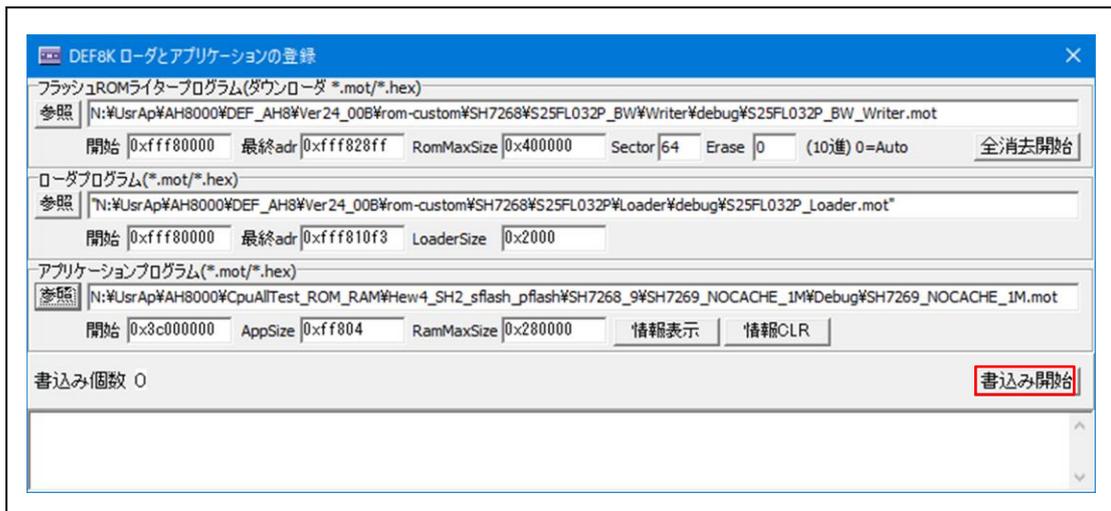
1) アプリケーションプログラムの「参照」をクリックする。



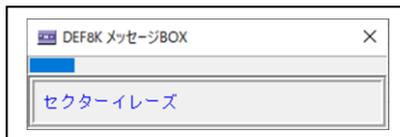
アプリケーションプログラムの開始アドレスとサイズを表示する。

「CPU 設定」に設定した内蔵 RAM サイズを表示する。

2) 「書き込み開始」をクリックする。



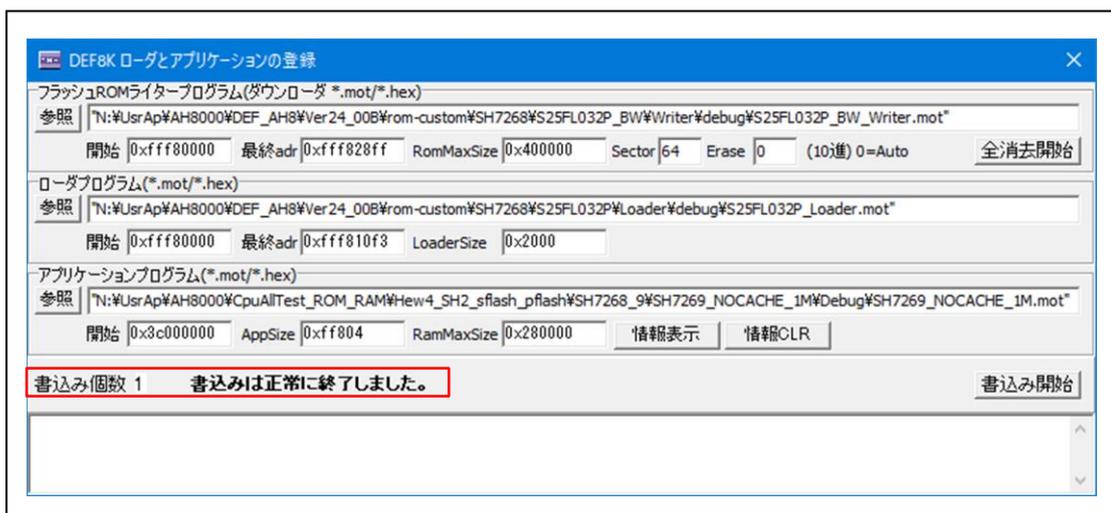
① FROM 書き込みプログラムの転送



② FROM のセクターイレース

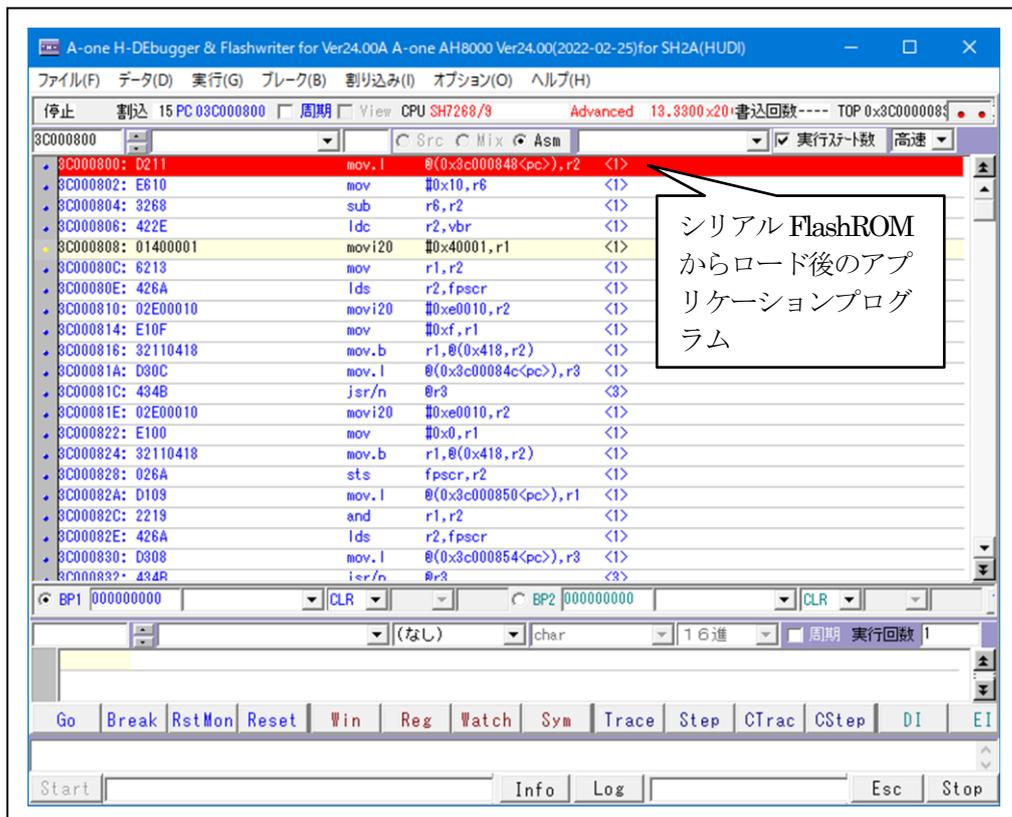
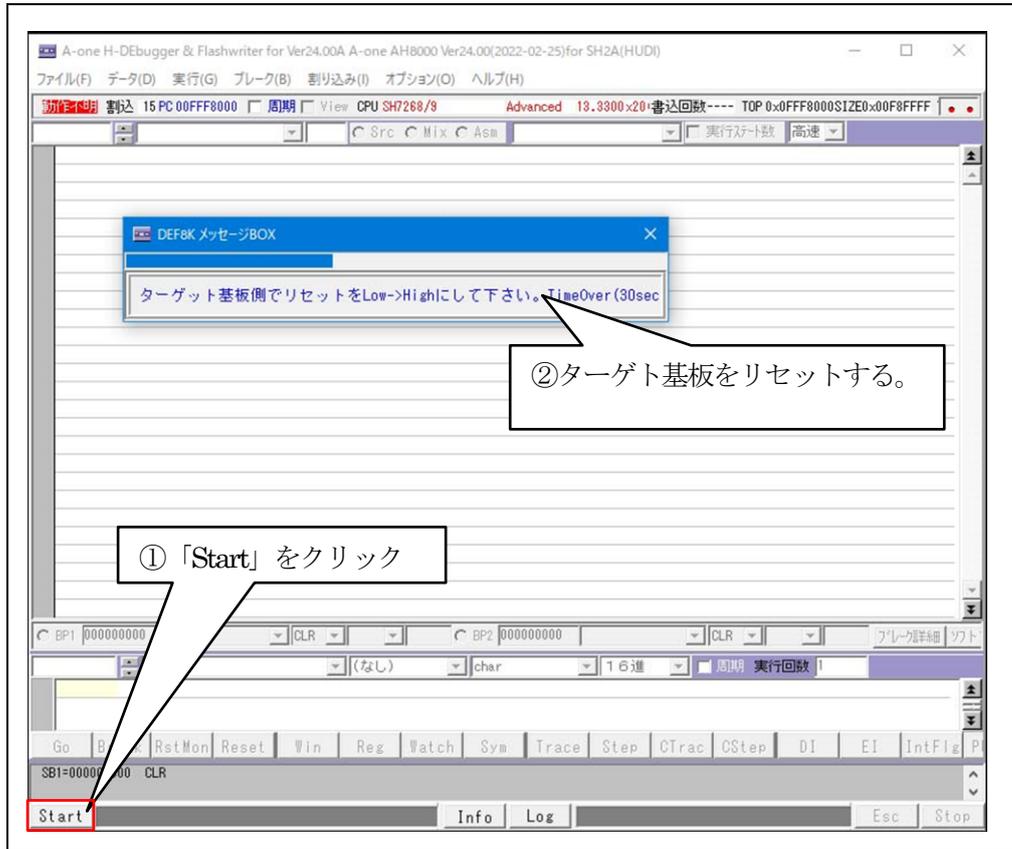


③ 拡張パケットでの FROM への書き込み

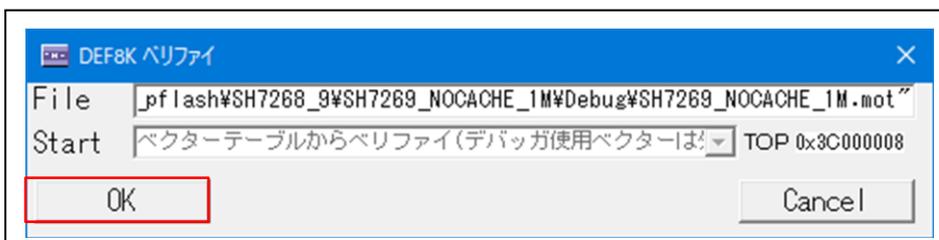
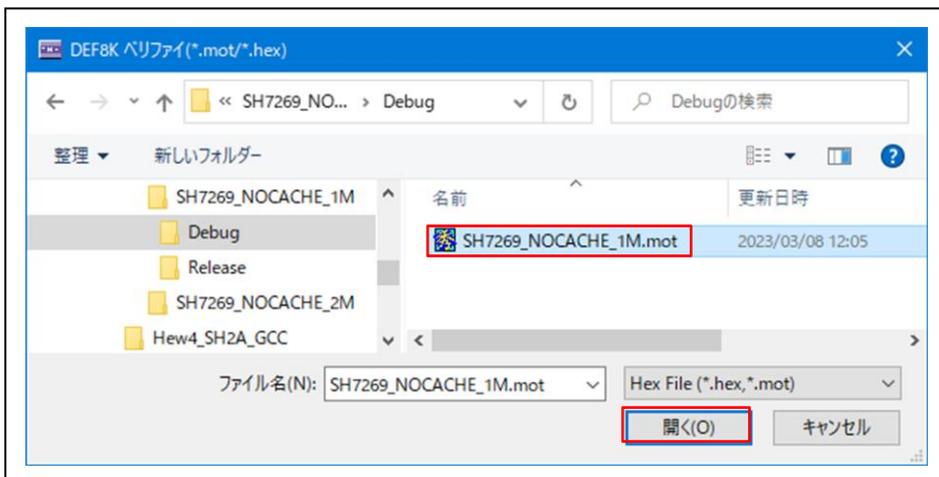
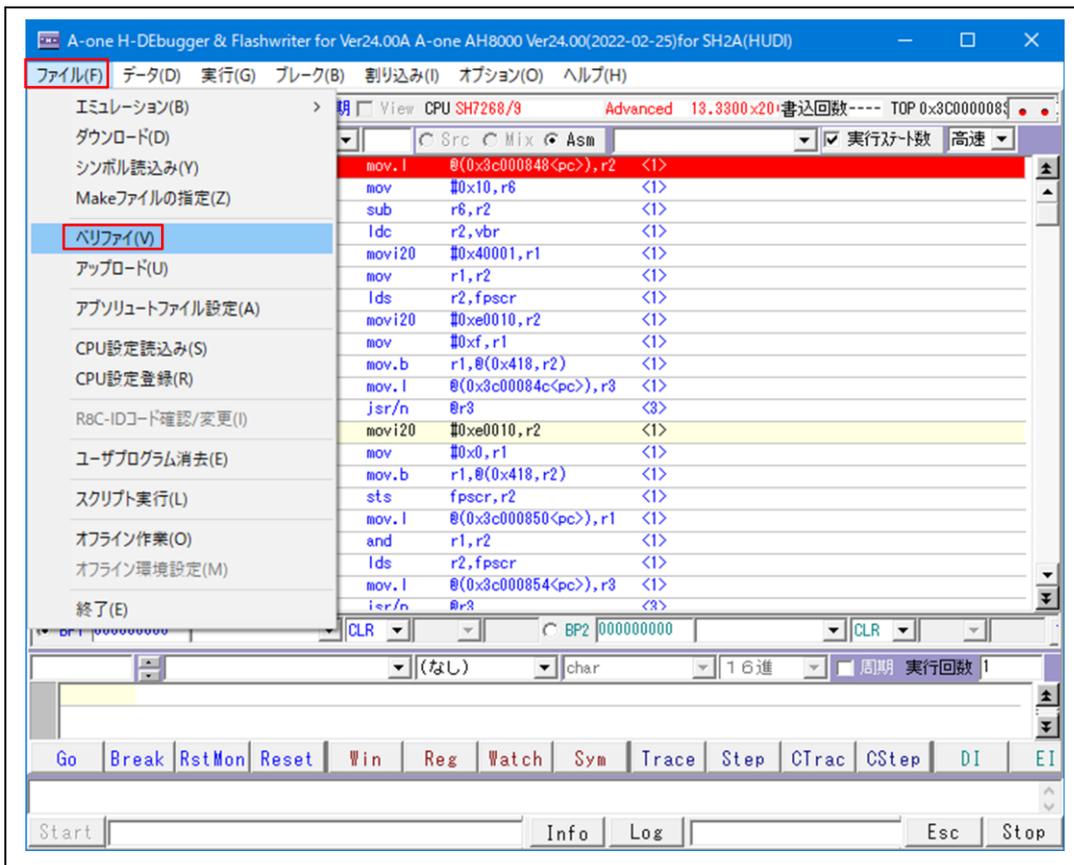


7-3) 「アプリケーションプログラム」が正常に書き込みされたかベリファイで確認する。

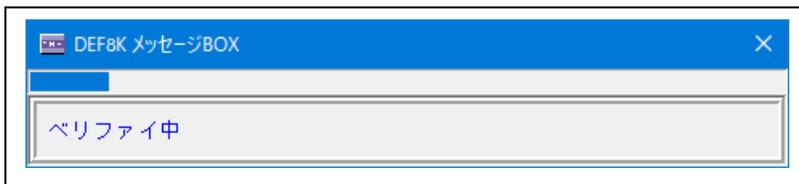
- 1) ターゲットの電源を再投入後、ロード時間を待ってから接続する。
(参考) 1.0M バイトコードのロード時間は約 5 秒



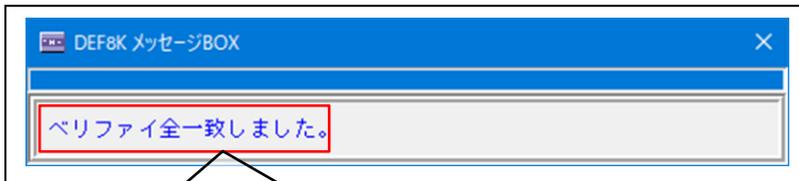
2) 内蔵RAM とベリファイする。



【ベリファイ中】



【ベリファイ結果】



この結果表示によりシリアル FlashROM にアプリケーションプログラムが正常に書かれローダも正常に動作したことになる。

8. 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を負わないもの
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>

