

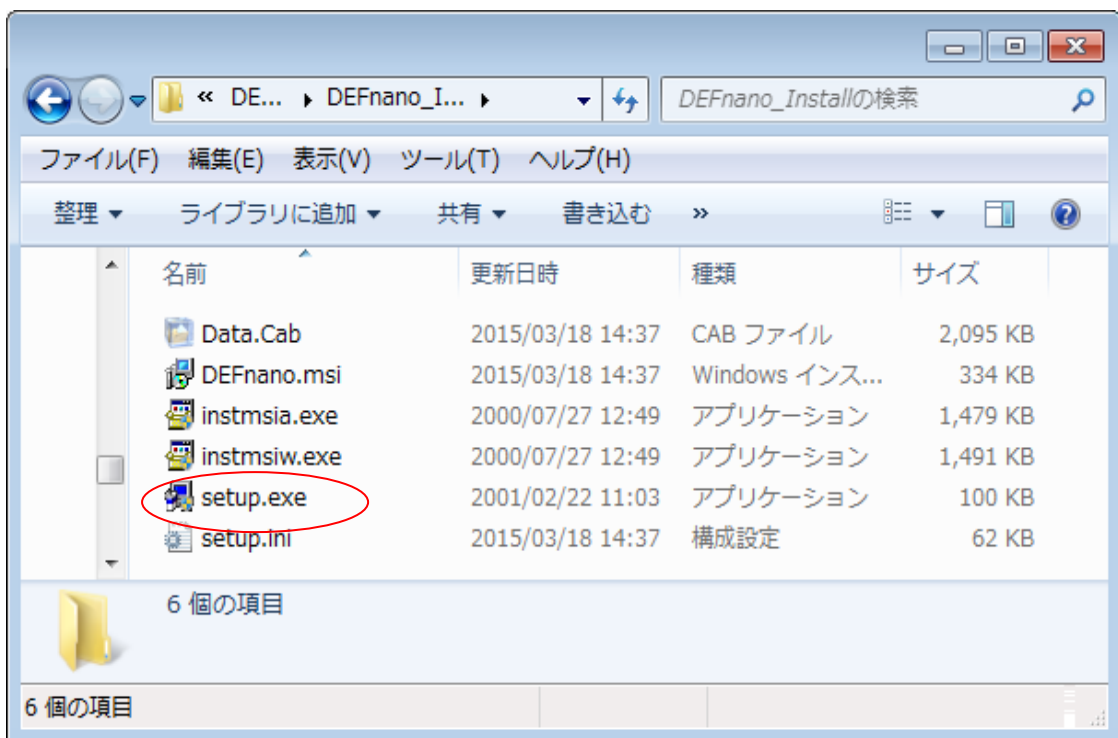
Renesas RZ/A1H 用リモートデバッガ・ナノ(DEFnano)の初期導入と

操作手順の概略説明

(ルネサス RZ/A1H)

1. 「DEFnano」をインストールします。

ご購入CDに保存してある「DEFnano_Install」の「setup.exe」をクリックします。



後の操作は、操作案内に従い進めて下さい。

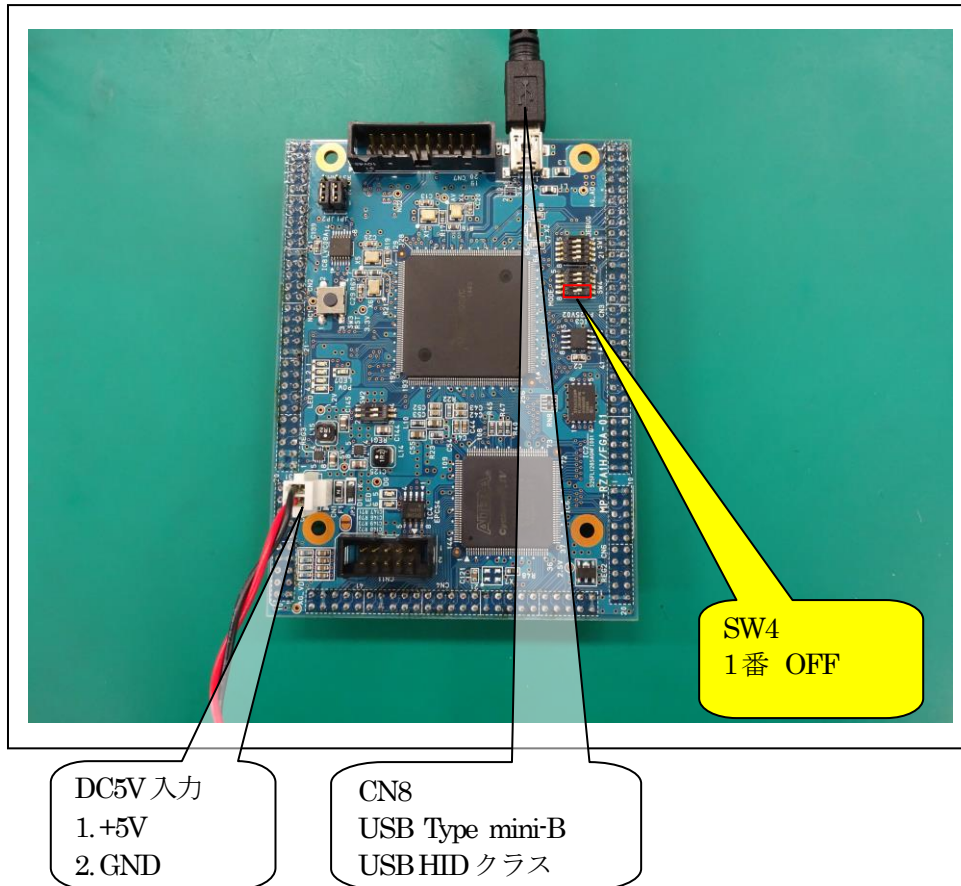


デスクトップ上にこのアイコンが登録される。

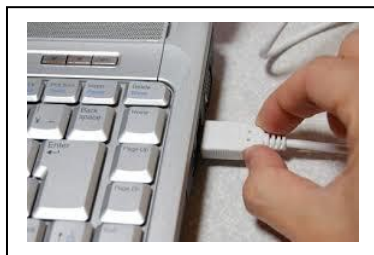
インストール終了画面

2. ハード環境を用意する。

1) MCU基板の用意（電源 OFF の状態）



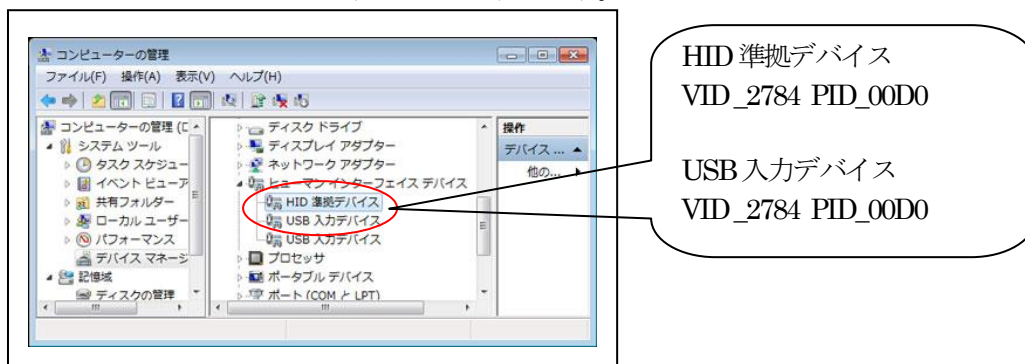
2) USBケーブルのType A側をPCのUSB口に接続する。(USB2.0/USB3.0)



3) MCU基板のDC 5 V電源を ONにする。

エニユメレーション処理を開始する。

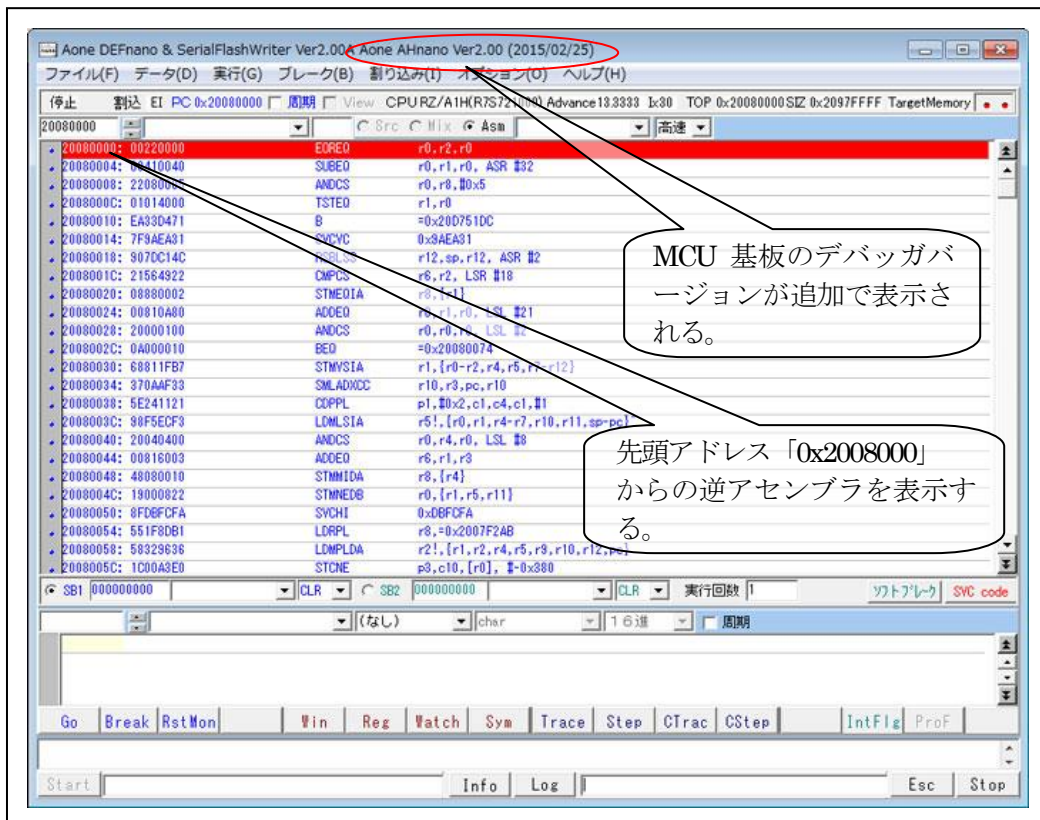
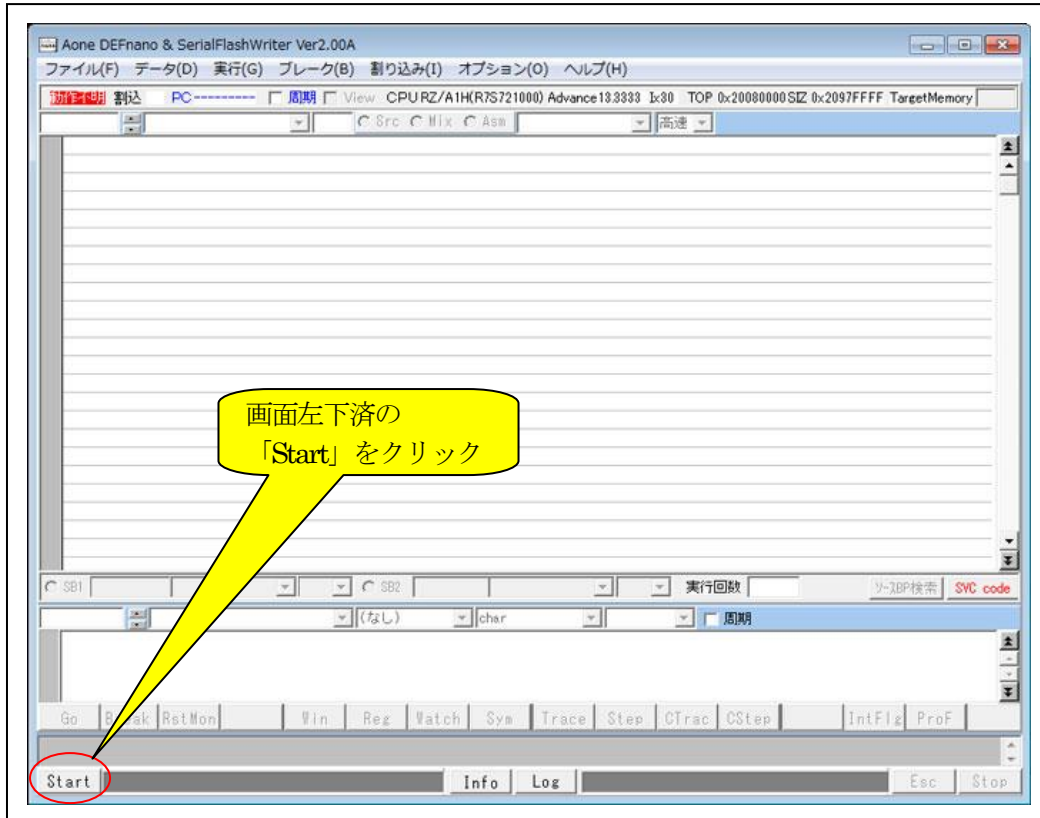
4) PCにHIDドライバーが登録されるのを待ちます。



3. DEFnano を起動し、デバッグ操作をする。

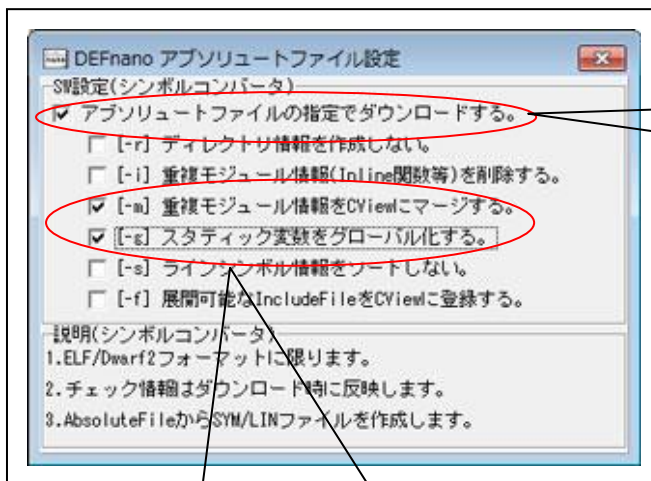
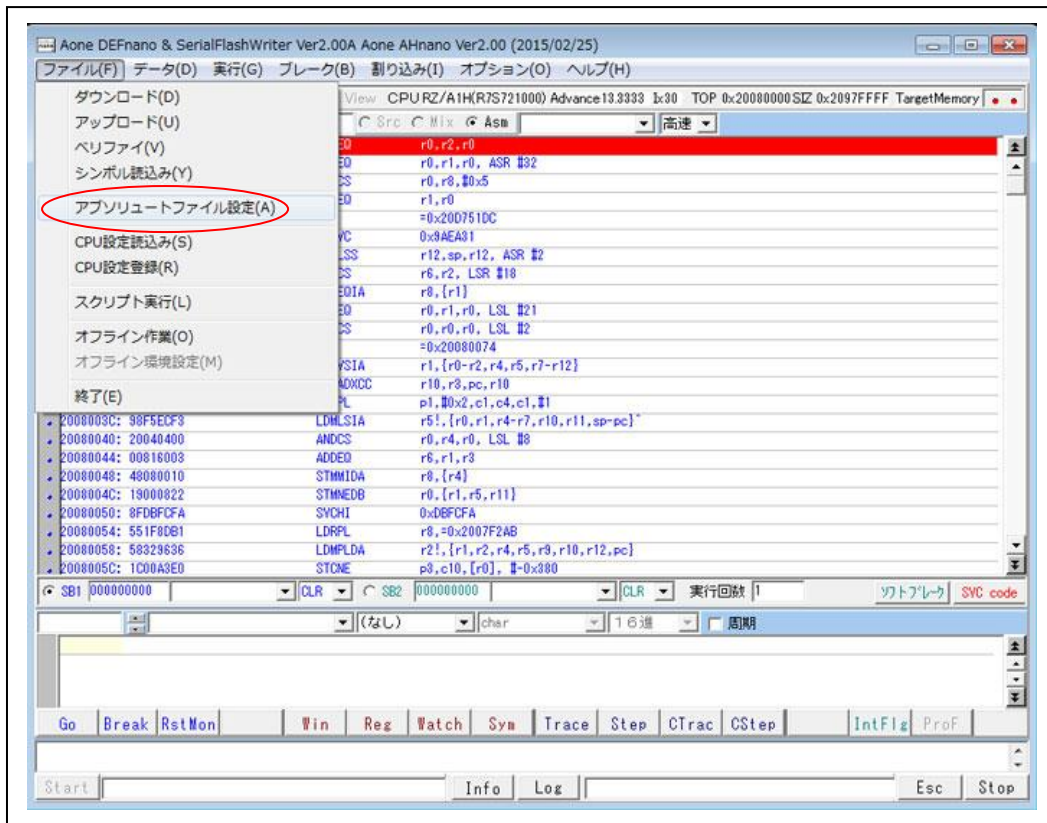
1) デスクトップ上のアイコンをクリックする。

または、「C:\Program Files\Aone\DEFnano\DEFnano.exe」を実行させます。



2) ダウンロード方式を設定する。

【ファイル】 - 【アブソリュートファイル設定】

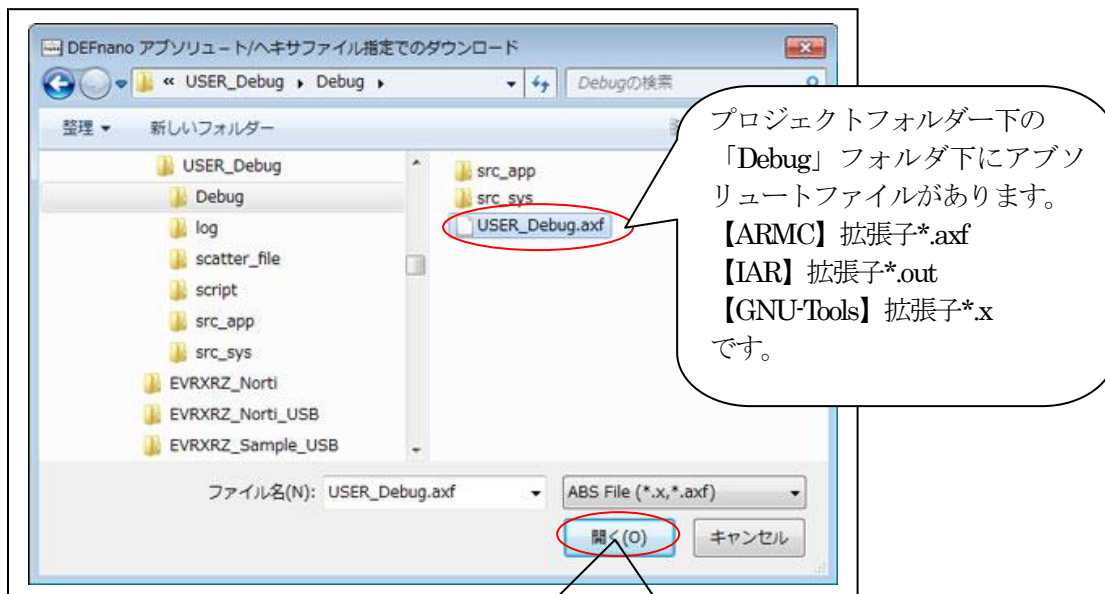
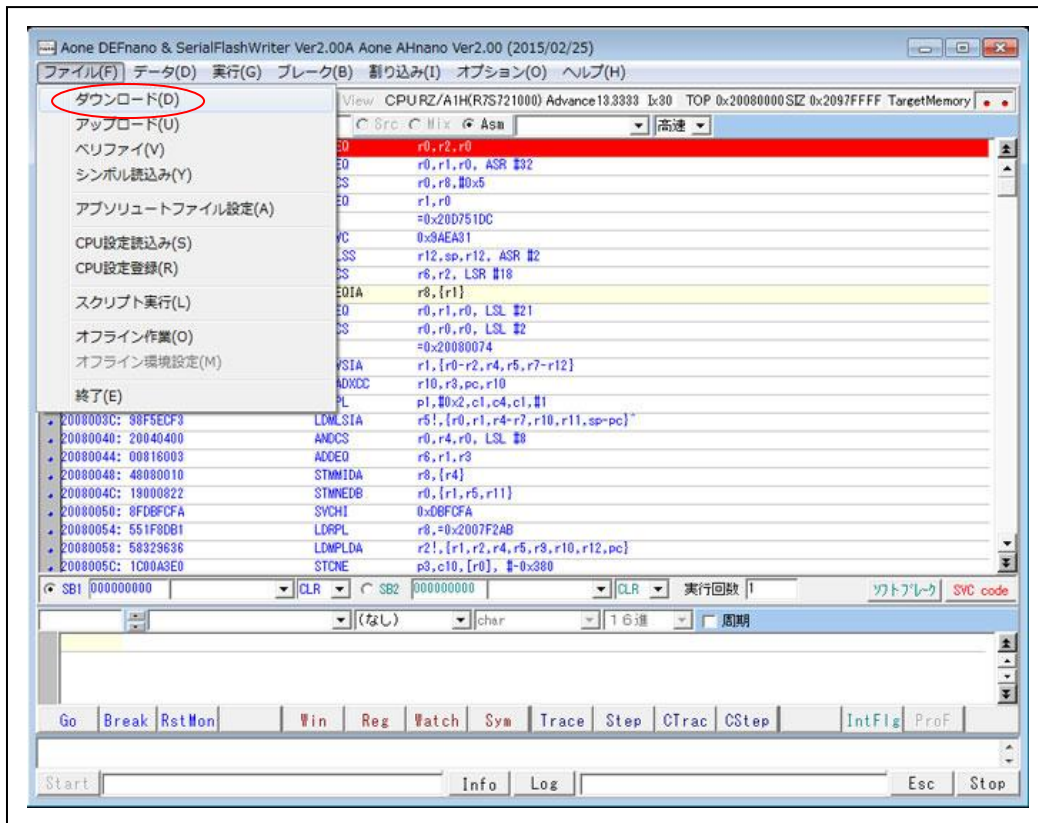


アブソリュートファイルの指定でダウンロードする。

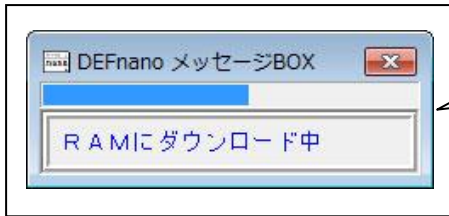
[-m] 重複モジュール情報を CView にマージする。
 [-g] スタティック変数をグローバル化する。
 この2アイテムを する。

- ① 「[-e]ディレクトリ情報を作成しない」 するとCソースデバッグが不可になる。
- ② 「[-i]重複モジュール情報をを削除する」 時として不整合な情報が生成された場合に
- ③ 「[-s]ラインシンボル情報をソートしない」 旧 ELFバージョン時に追加、現状不要
- ④ 「[-f]展開可能なインクルードファイルを CView に登録」 * .h 等の取得可能なファイルを登録

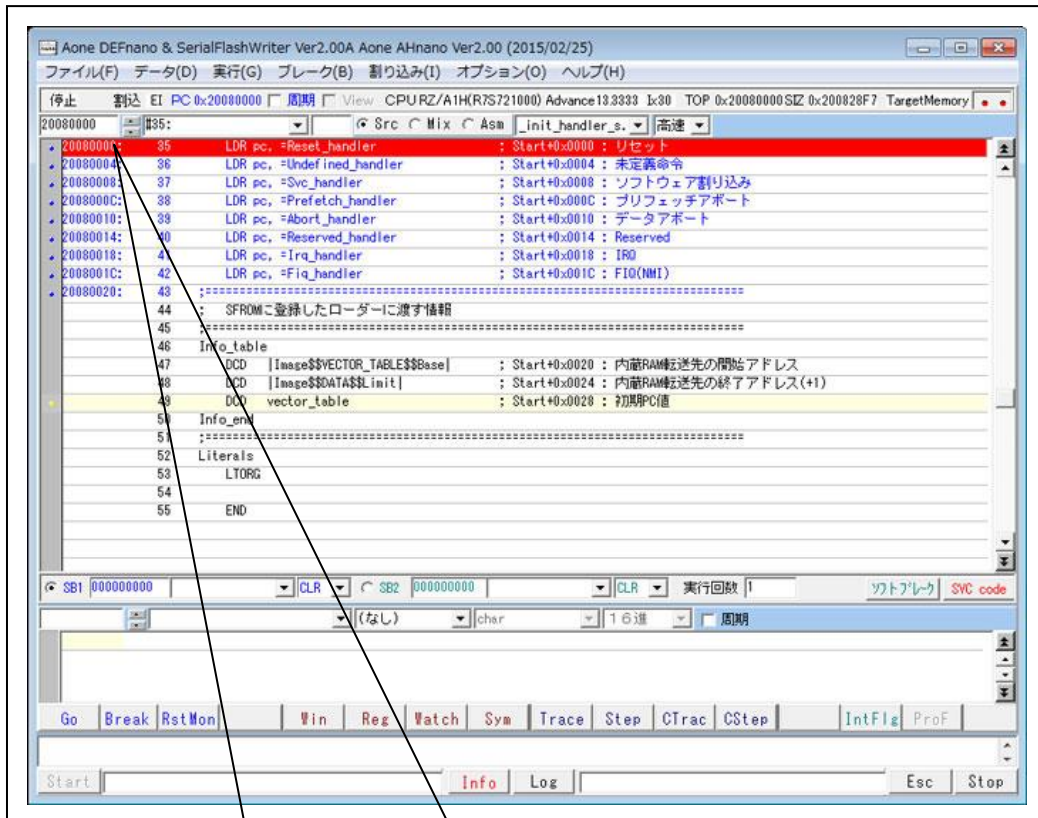
- 3) サンプルプロジェクトをダウンロードする。
 【ファイル】 - 【ダウンロード】



「開く」をクリックしますと、DEFnano は、指定されたアプソリュートファイルからシンボル情報を抽出する。[*.*sym*.lin]
 【ARMC】 ACsymconvexe 「シンボルコンバータ」
 【IAR】 IARsymconvexe 「シンボルコンバータ」
 【GNU-Tools】 GCsymconvexe 「シンボルコンバータ」



シンボルの抽出終了後、
ターゲット MCU の内蔵 RAM
に実行コードをダウンロードす
る。

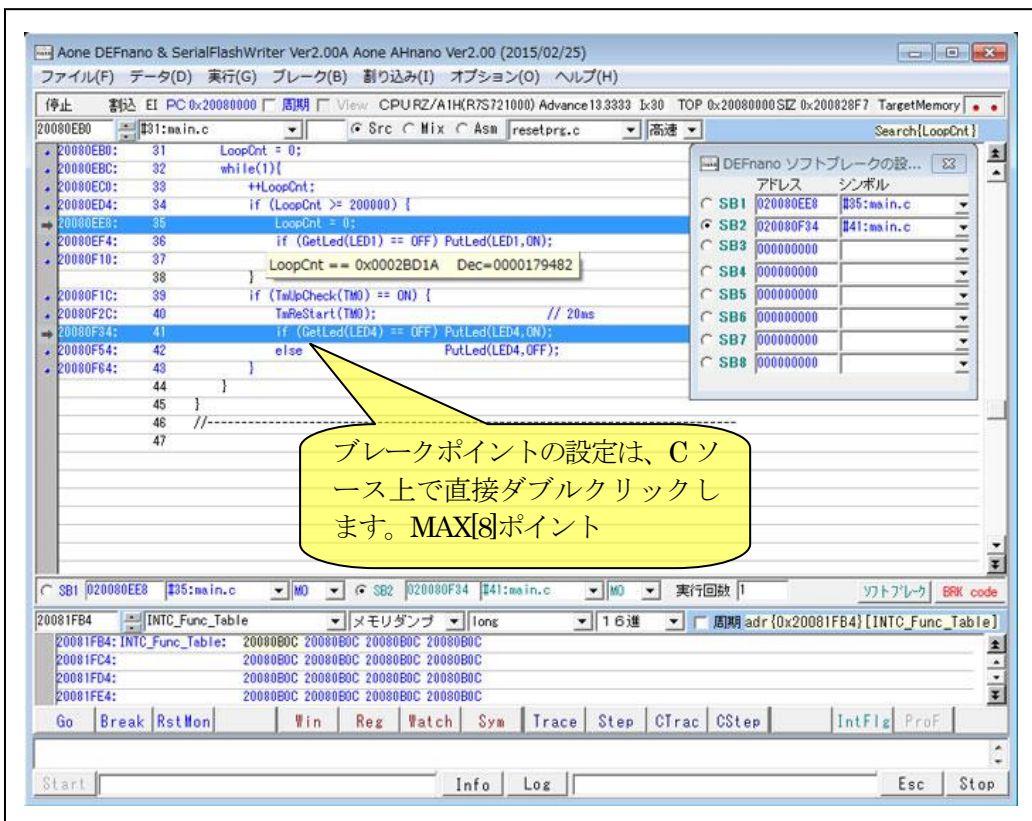
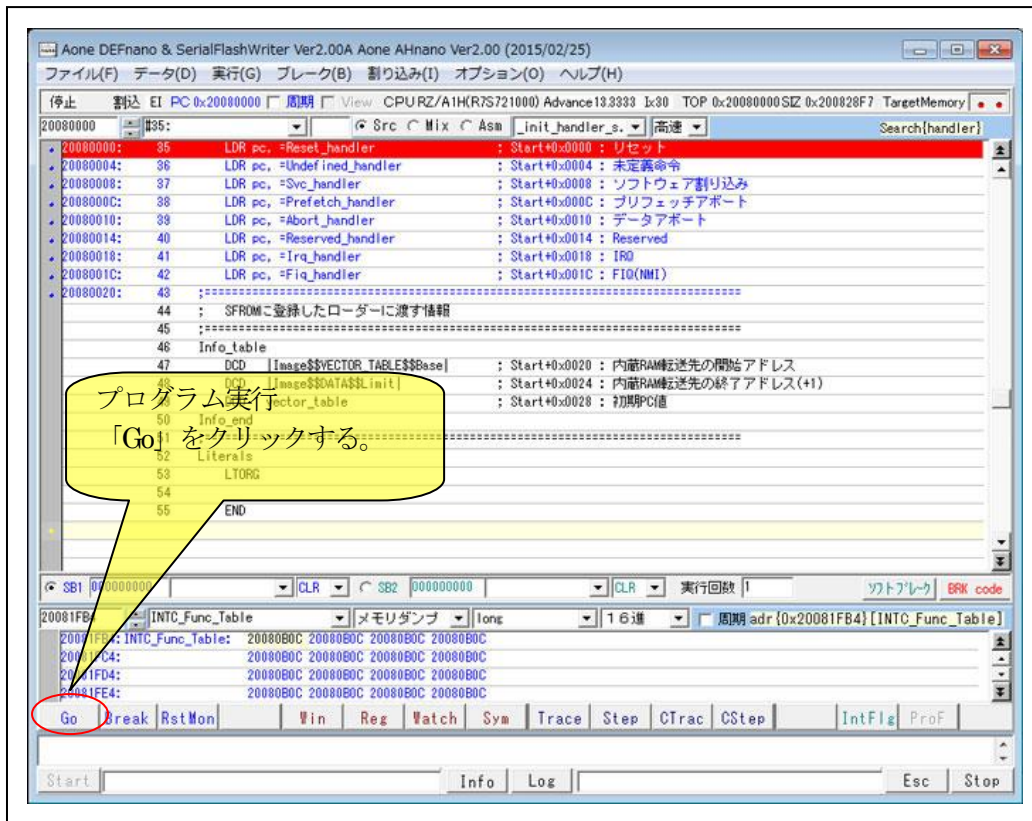


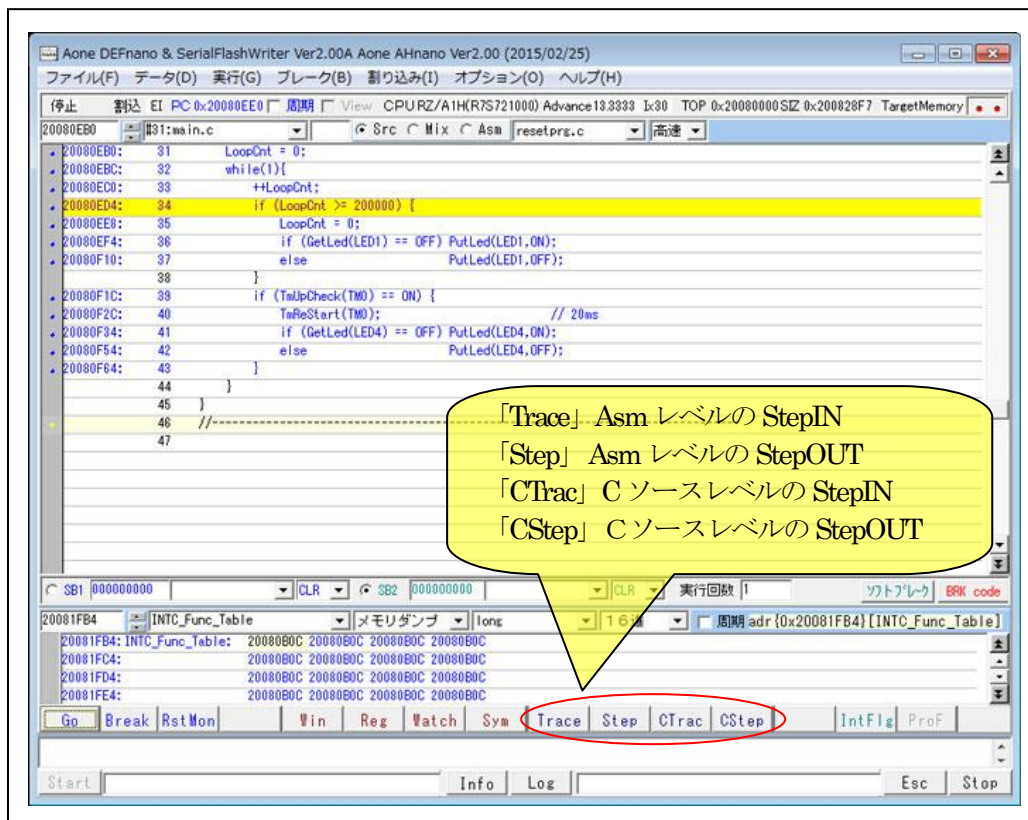
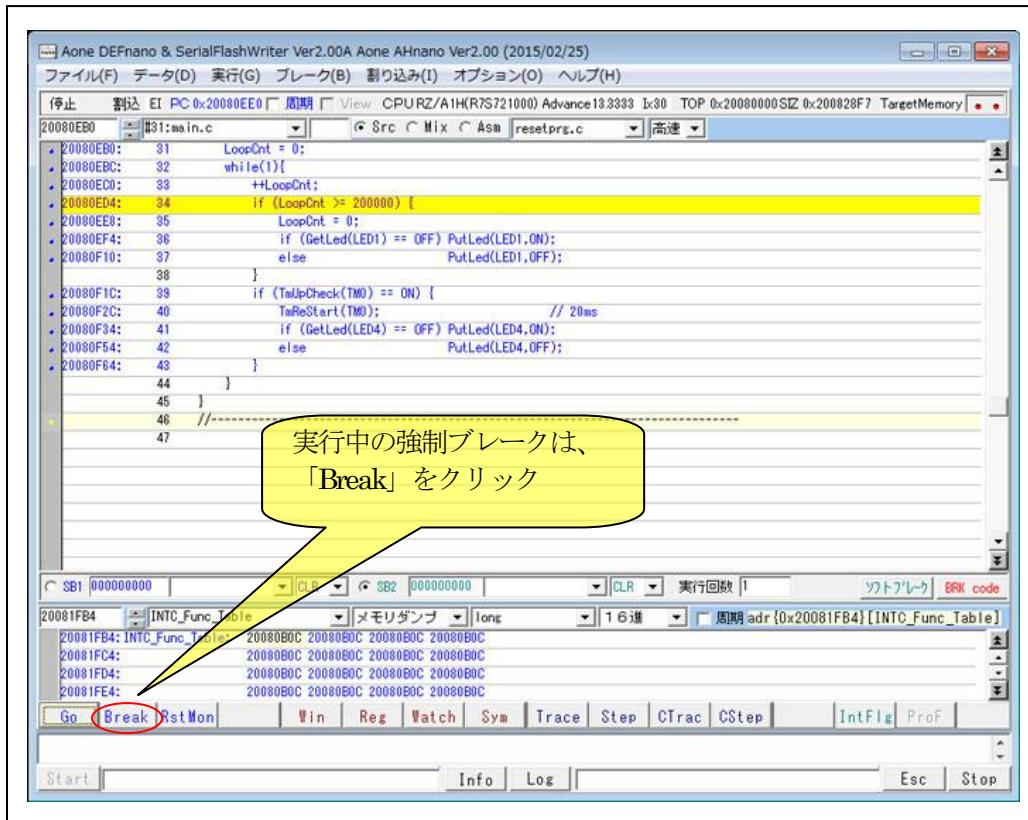
ダウンロードが正常に終了しますと、
このようにベクタテーブル定義ソース「先頭アドレス 0x20080000」が
CView 画面に表示される。
前回 DWL してその時のアドレスが記憶されている場合は、表示され
ませんのでショート PB 「RstMon」をクリックしますとベクタテ
ブル定義ソース画面になる。

⚠再ダウンロード時の注意事項

USB-CH1(ITS_LIB)を使用したアプリ開発時に再ダウンロードする場合、出来るだけ USB-CH1 側のケーブルを抜いた状態で実施して下さい。理由は、ダウンロード完了時にターゲットをリセットする仕組みになっていますので、ケーブルを刺した状態だと Windows が USB-CH1 のドライバー切り離し処理とデバッグ側の USB(HID)のエnumレーション処理を開始しますので完了まで通常より時間が必要になります。

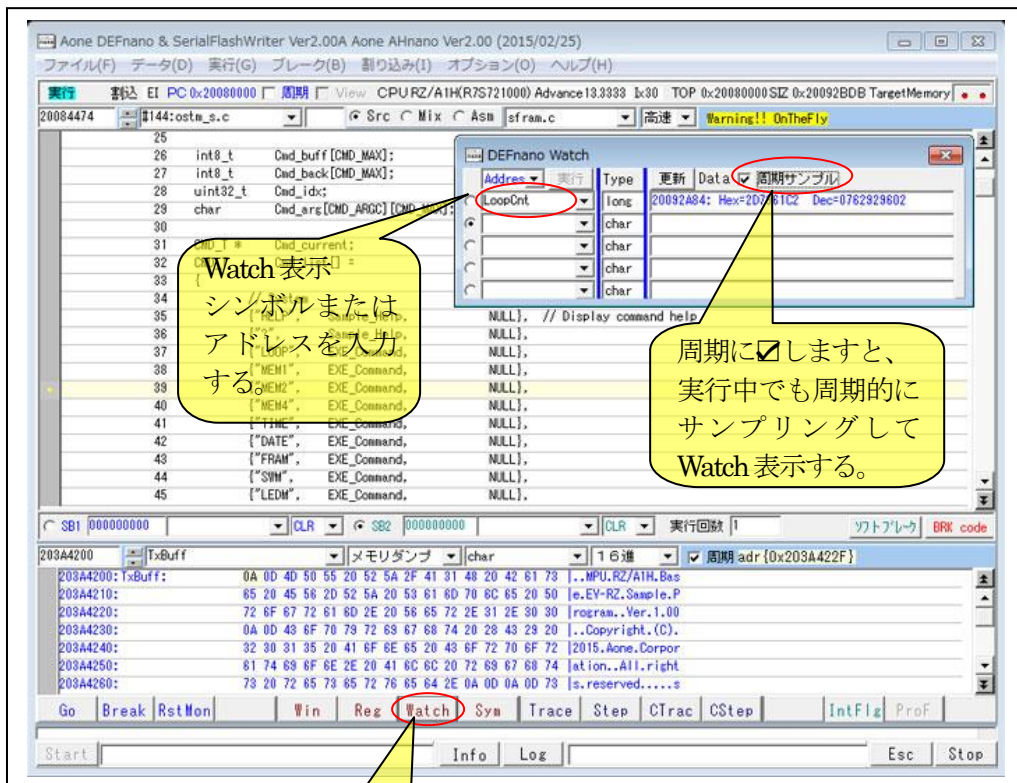
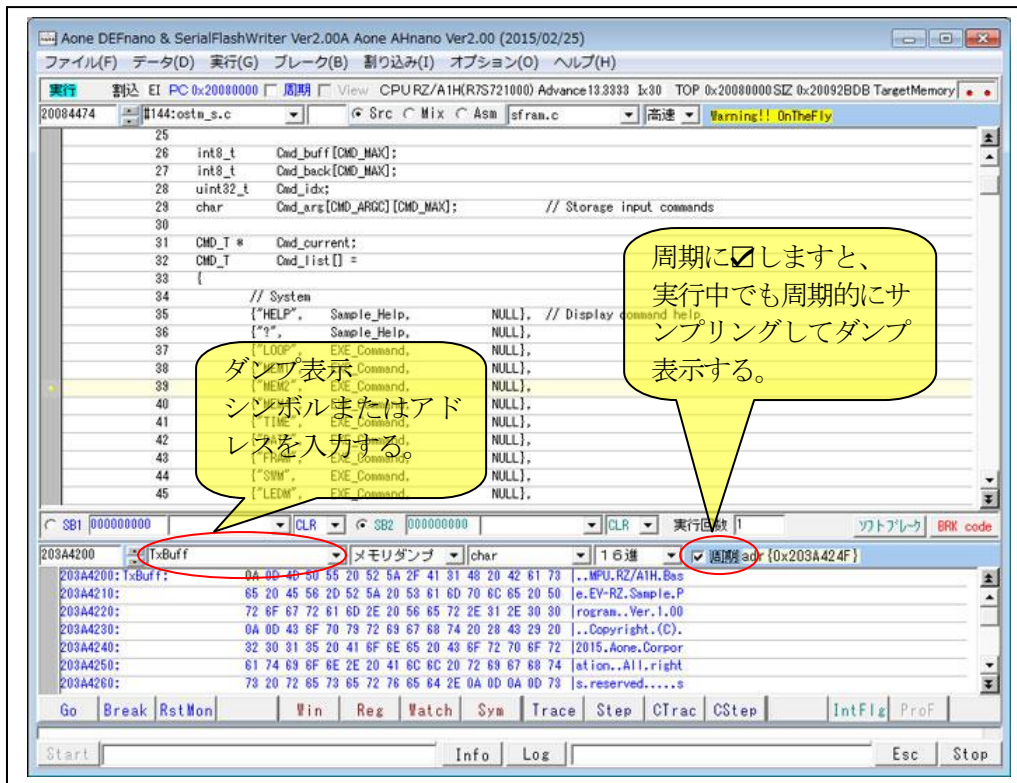
4) プログラム実行とブレーク設定と強制停止方法



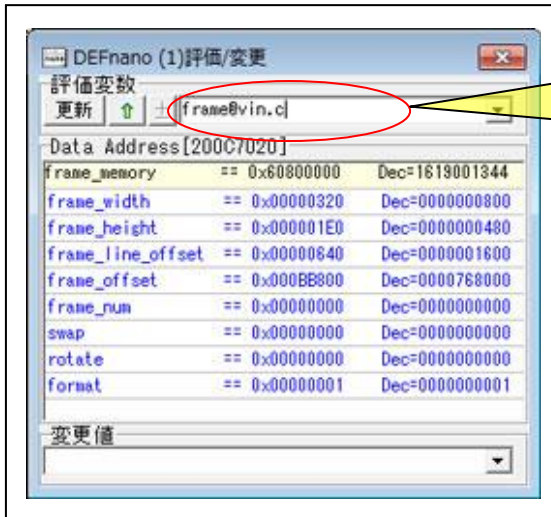


- ステップ/トレース機能を使用する場合は、＜オプション＞＜環境設定＞の「Page1」
 「トレース/ステップ実行の動作」を「許可」にチェックして下さい。

5) 変数の内容をダンプとウォッチと評価について



ショートPB「Watch」をクリック



構造体や多次元配列の場合は、「評価/変更」を使用する。

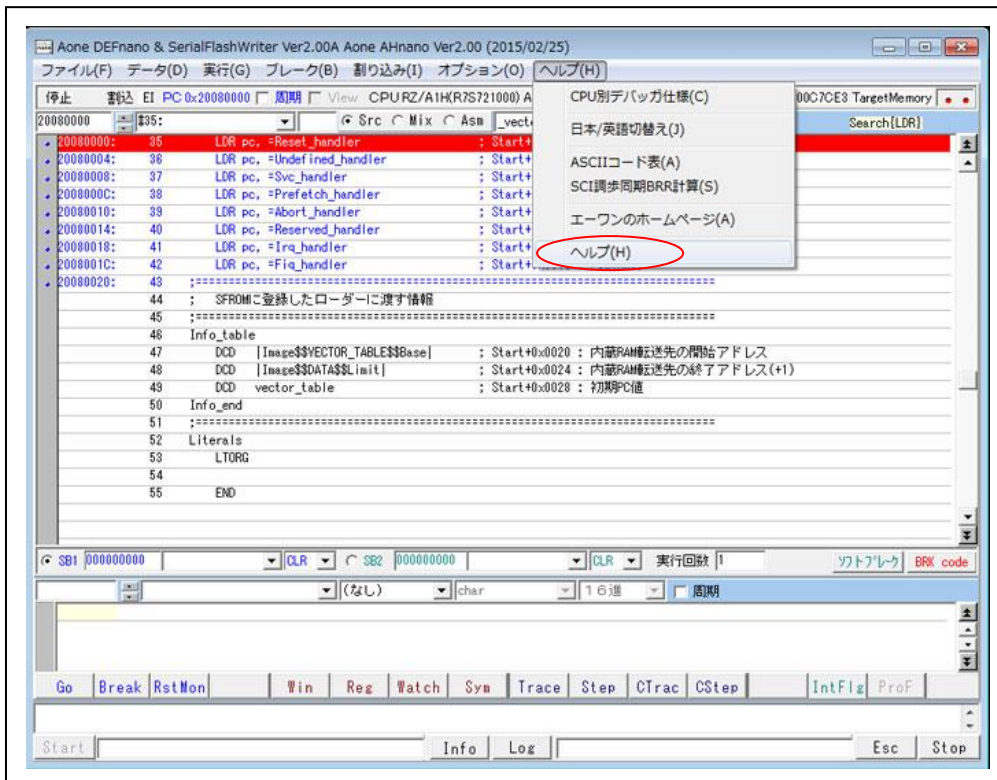
⚠️ 周期サンプリング使用時の注意事項

周期サンプリングの仕組みは、ユーザープログラムの実行中に USB0 の IRQ 割り込みを利用してデータを取得しています。USB 割り込みの特性上、1 バイト~64 バイトまでのデータ情報の取得に「250usec~500usec」の割り込み処理が実行されます。

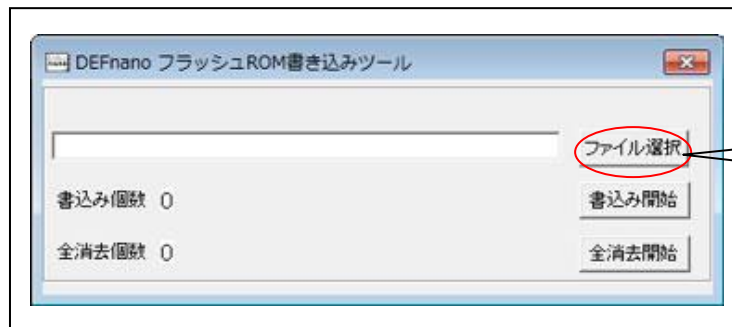
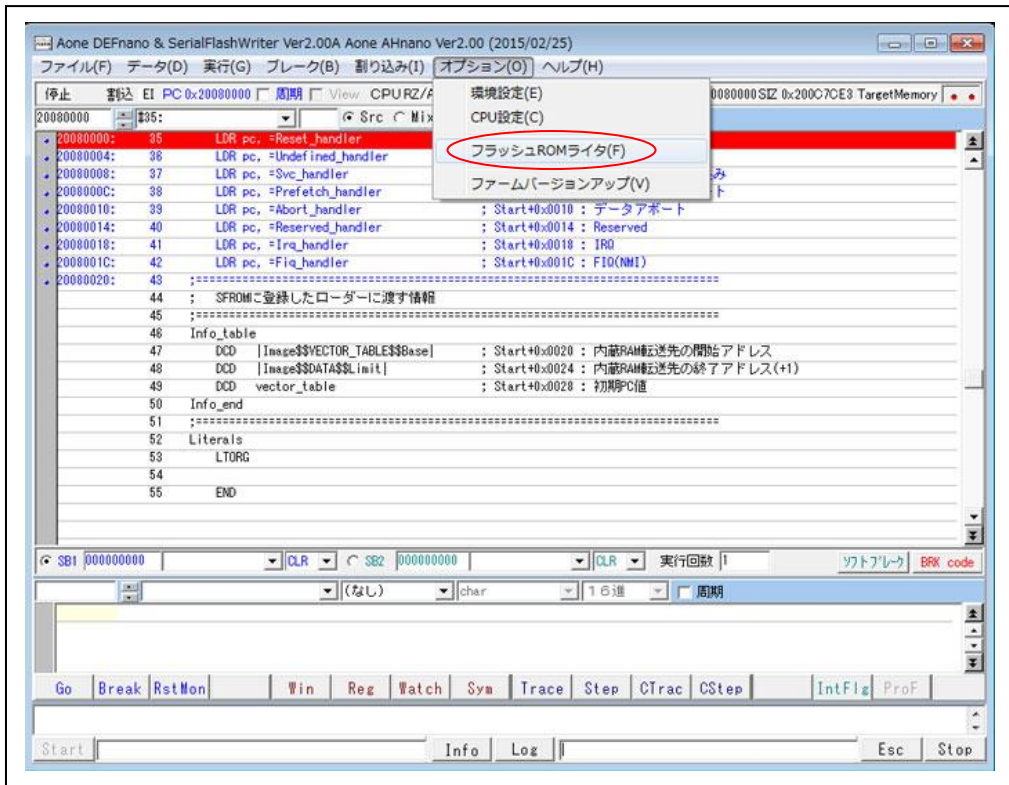
この周期サンプリングがアプリ側処理の妨げになる場合は使用しないで下さい。

6) その他のデバッグ操作

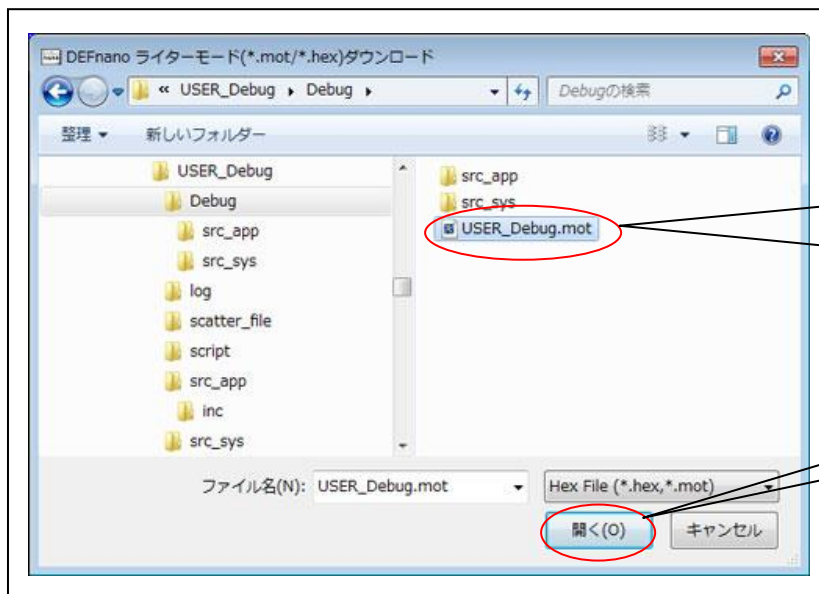
その他、詳細なデバッグ操作は【ヘルプ】 - 【ヘルプ】をご覧ください。



4. デバッグが完了しシリアルフラッシュ ROM にアプリケーションプログラムを書き込む方法
【オプション】 - 【フラッシュ ROM ライタ】

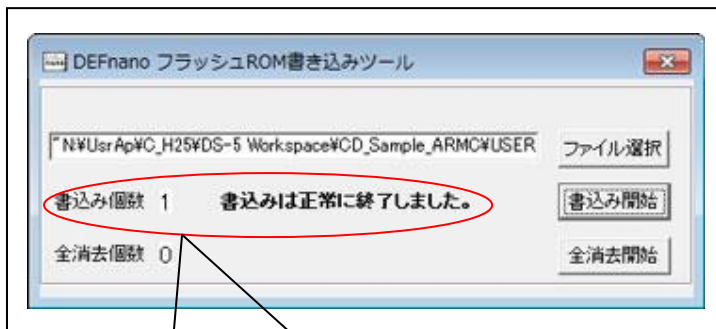
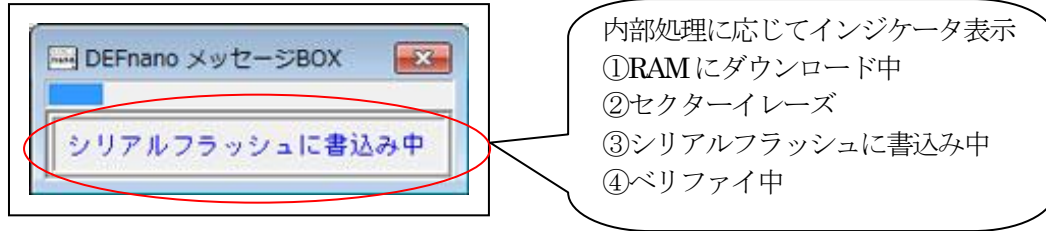
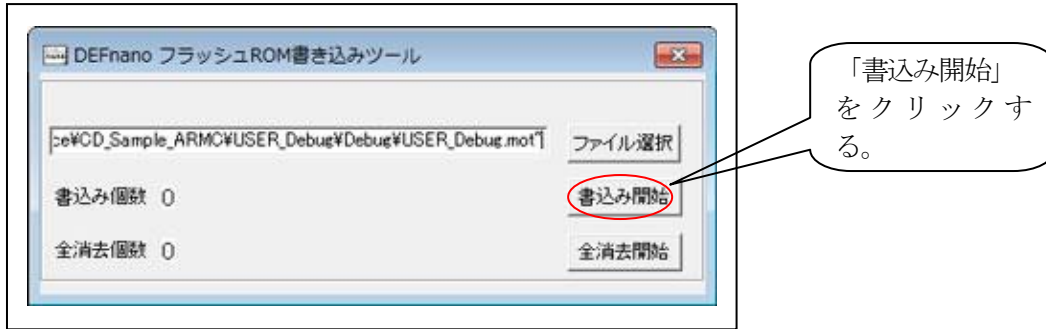


「ファイル選択」をクリック



ライターモードでは、「*.mot」のHex ファイルを指定する。

「開く」をクリックする。

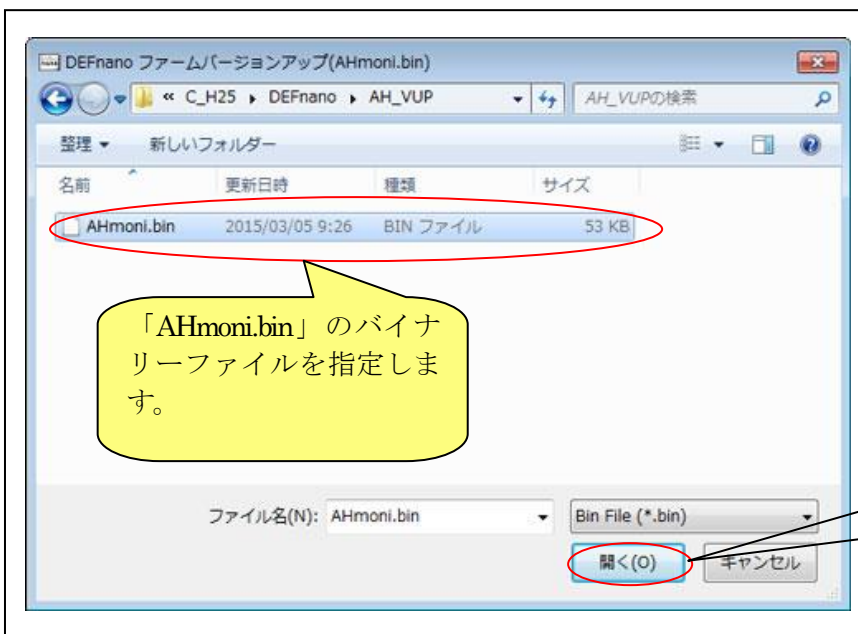
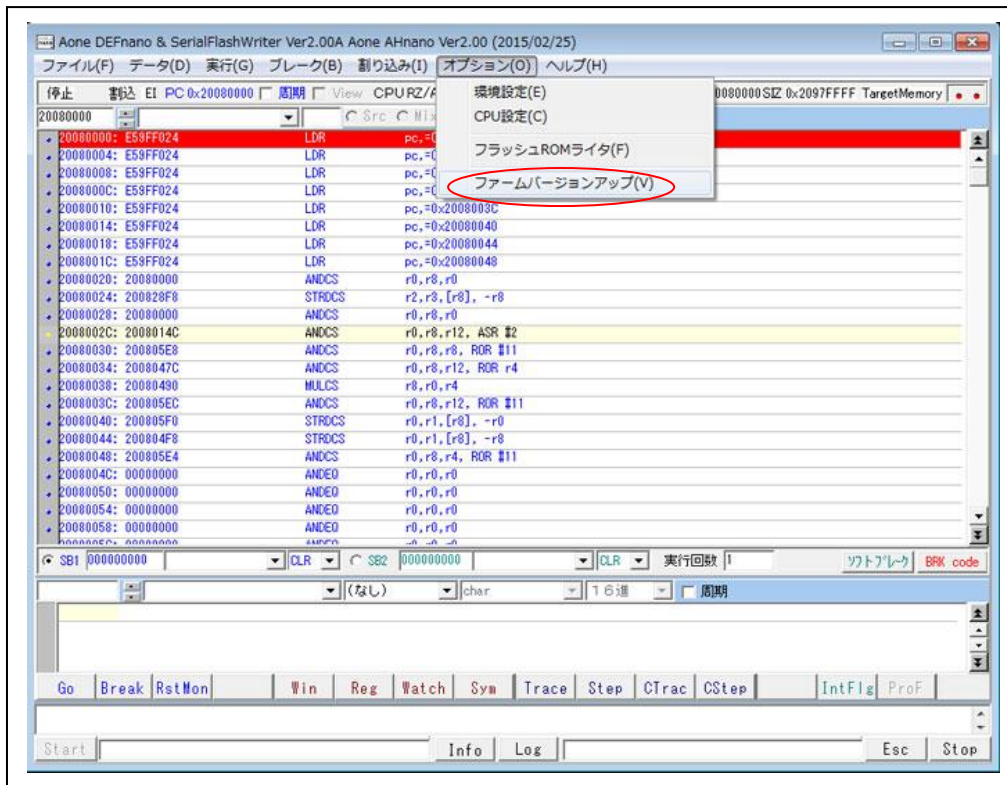


正常終了しますと、「書き込みは正常に終了しました。」と表示され、個数がカウントアップします。個数は、この窓を閉じますとゼロになります。

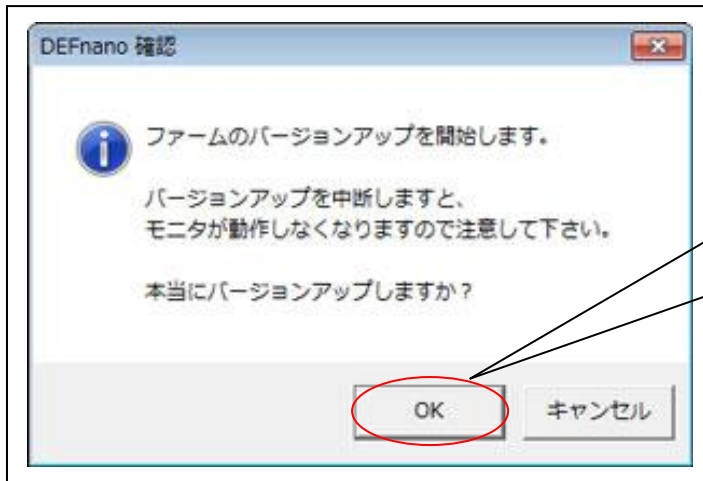


「全消去開始」をクリックしますと、アプリケーションエリア(セクター2以降)のみを消去します。ローダーとデバッガファームエリアは消去しません。

5. ローダーとデバッガファームのバージョンアップ方法
 【オプション】 - 【ファームバージョンアップ】

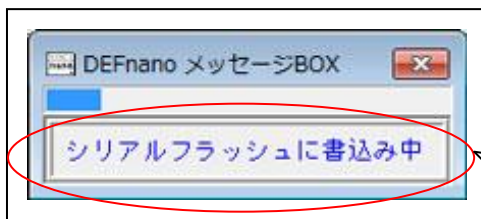


<AH_VUP>のファイル名
 ①AHmoni.bin
 ②CODE_SPIBSC_INIT.bin
 ③VECTOR_TABLE.bin
 の3ファイルになります。



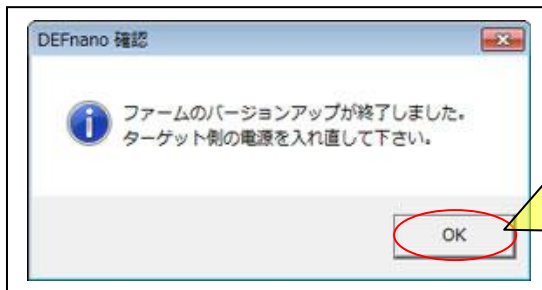
フラッシュ ROM への書き込みに対する警告表示です。

「OK」をクリックします。



内部処理に応じてインジケータ表示

- ①RAMにダウンロード中
 - ②セクターイレーズ
 - ③シリアルフラッシュに書き込み中
 - ④ベリファイ中
- sFROMの2セクター使用する。



ファームのバージョンアップが正常に終了したメッセージです。

この状況ですと、RAM上のファームは古いままで動作していますので、必ず、ターゲットの電源を入れ直して下さい。

「OK」をクリックする。



6. その他の注意事項

- 1) ファームのバージョンアップ中に USB ケーブル等が抜けて失敗した場合は、内蔵 RAM にファームが常駐していますので、リセット及び電源の再立上げ等を実施しないでバージョンアップ操作を再実施して下さい。
- 2) DEFnano デバッガは、USB エミュレーションを実施するタイミングはターゲットの電源 ON 時のみに処理する仕組みになっていますのでデバッガを有効にしたい場合は電源 ON 時にターゲットの USB 口にケーブルを刺した状態でいて下さい。
- 3) デバッガを切り離してシリアルフラッシュ ROM ブートで動作させたい場合は、ターゲット側の USB 口からケーブルを抜いた状態で電源 ON しますとローダーが判断して ROM 側のプログラムを内蔵 RAM に展開して RUN させる仕組みになっています。
- 4) プログラムのデバッグ中は、完成されていないプログラムのため、プログラムミスにより閉ループに入り抜けてこなくなる場合もあります。その時に割り込み禁止状態に陥り、リモートデバッガの性質上、強制 Break さえ効かなくなる場合もあります。このような状態に陥った場合は、MCU 基板上にある「ResetSW」を押して、初期状態からデバッグを開始して下さい。

6. リモートデバッガ・ナノを使用する場合のアプリケーション側ソフトの手続きについて

- 1) デバックモニタは、割り込みは[Prefetch]と[IRQ]を利用しています。

⚠️ 注意事項

- ・ `IRQ_SP` と `ABT_SP` は、別空間になるようにスタックポインタを設定して下さい。

- 2) Prefetch・IRQ ハンドラーには、デバックモニタ用マクロ定義が必要になります。

```

;*****
;* _init_handler.s
;* デバックモニタを使用する場合の割り込みハンドラーの記述
;*****

    INCLUDE _init_handler.inc      ::: デバックモニタ用
    IMPORT SwiMonitor              ::: モニタ用の宣言

;=====
; Prefetch_handler
; BRK 割り込みハンドラー          Start+0x000C : プリフェッチアポート
;=====

Prefetch_Handler
    SUB    lr, lr, #4              ::: [-4]する必要あり BRK コードの PC 位置を知るため
;-----
    GET_REGISTER #ABT_MODE, Abt_end ::: モニタ用にレジスタ数値をメモリにストアする。
;-----
    SRSDB  sp!, #ABT_MODE         ::: 現在のモードの LR_abt と SPSR_abt を Stack に退避
    PUSH  {r0-r12}               ::: レジスタを退避
;-----
    BL    SwiMonitor              ::: デバックモニタ処理
;-----
    POP   {r0-r12}               ::: レジスタの復帰
    RFEIA sp!                    ::: 割り込みの復帰処理
;=====

;* IRQ_Handler                    Start+0x0018 : IRQ
;=====

IRQ_Handler
    sub   lr, lr, #4
;-----
    GET_REGISTER #IRQ_MODE, Irq_end ::: モニタ用にレジスタ数値をメモリにストアする。
;-----
; ↑先頭に定義
; ↓は、ユーザー側で記述
;=====

```

黄色の記述は必ず必要です。詳細は、サンプルプロジェクトの「src_app¥_init_handlers_s.s」を参照

3) デバックモニタプログラムの初期化する関数をメインの先頭で呼ぶ手続きが必要です

```

//*****
//   ①RTOS(NORTi)用の記述例  [Nmonitor.c]のリンクが必要!!
//*****
#include "Nmonitor.h"          // RTOS(NORTi)用のヘッダーファイル

//-----
//   main関数
//-----
int   main(void)
{
    sysini();                 // システム初期化

    MonitorInit();           // モニタの初期化 (必ず sysini()の後で!!)

}
//-----

//*****
//   ②ベアメタル用の記述例  [Umonitor.c]のリンクが必要!!
//*****
#include "Umonitor.h"         // ベアメタル用のヘッダーファイル

//-----
//   main関数
//-----
int   main(void)
{
    MonitorInit();           // モニタの初期化 (必ず main()の先頭で!!)

}
//-----

```

- 1) Nmonitor.h/Umonitor.hは、サンプルで用意してあります。そのまま利用する。
- 2) コメントに記述してある注意事項に従ってポーティングする。

以上です。

7. 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、ARM社およびルネサス 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。

8. 商標

- ・ARM DS-5は、ARM社の登録商標、または商品名称です。
- ・EWARMは、IAR社の登録商標、または商品名称です。
- ・RZ および RZ/A1H は、ルネサス エレクトロニクス株式会社の登録商標、または商品名です。
- ・その他の会社名、製品名は、各社の登録商標または商標です。

9. 参考文献

- ・「RZ/A1H グループ ユーザーズマニュアル ハードウェア編」
ルネサス エレクトロニクス株式会社
- ・ルネサス エレクトロニクス株式会社提供のサンプル集
- ・「armcc ユーザガイド DUI0472JJ」 ARM社
- ・「アセンブラの使用 DUI0473GJ」 ARM社
- ・「リンカの使用 DUI0474GJ」 ARM社
- ・「コンパイラリファレンスガイド DUI0328BJ」 ARM社
- ・「アセンブラリファレンス DUI0489GJ」 ARM社
- ・「armkink リファレンスガイド DUI0804AJ」 ARM社
- ・「IDE プロジェクト管理およびビルドガイド UIDEARM-9j」 IAR社
- ・「IAR C/C++開発ガイド コンパイラおよびリンク DARM-14j」 IAR社
- ・「IAR アセンブラリファレンスガイド AARM-9j」 IAR社
- ・「IAR デバッグプローブガイド IARprobes-2j」 IAR社
- ・その他

〒486-0852
愛知県春日井市下市場町 6-9-20
エーワン株式会社
<http://www.robin-w.com>