CSIDEチュートリアルブック

第 4 版　2007 年 12 月

Computex®
株式会社 コンピューテックス

CS0007(E)1107

CSIDE チュートリアルブック 目次

Chapter 1.はじめに

CSIDE チュートリアルブックは、CSIDE を初めて導入されるなど CSIDE に不慣れな方向けにデバッグの準備からデバッグ操作までの流れについて説明を行う資料です。各設定項目についてご不明な点がありましたら、ユーザーズ・マニュアル、オンライン・マニュアルと合わせてご覧ください。

○本資料を読むに当たっての注意点

・本資料は、各 CPU 共通の資料です。個々の詳しい設定については、各ユーザーズ・マニュアルをご確認ください。
・資料中の各画面は、必ずしもお使いの CSIDE と同じとは限りません。
・お使いの CSIDE のバージョンによっては、資料中の機能が無い場合があります。
・本資料は、PALMiCE2 向けの資料です。

○本資料の流れ

Chapter2. PALMiCE2について
↓
JTAG エミュレータの仕組みや特長について解説します。
JTAG エミュレータを初めて使用される場合などには、一度参照してください。

Chapter3. デバッグを開始する準備
↓
CSIDE にてデバッグを開始するまでに必要となる準備についての解説を行います。
CSIDE を起動される前にご確認ください。

Chapter4. ハードウェアの接続
↓
ターゲット・システムとの接続ついて解説します。

Chapter5. CSIDE の設定
↓
プログラムのデバッグを開始するまでに行う CSIDE の設定とターゲット・システムの起動について説明を行います。

Chapter6. デバッグの開始
↓
CSIDE にてデバッグを行う際の、実行/ブレークなど一通りの基本動作の確認とプロジェクトの保存について解説を行います。

Chapter7. CSIDE の便利な使い方

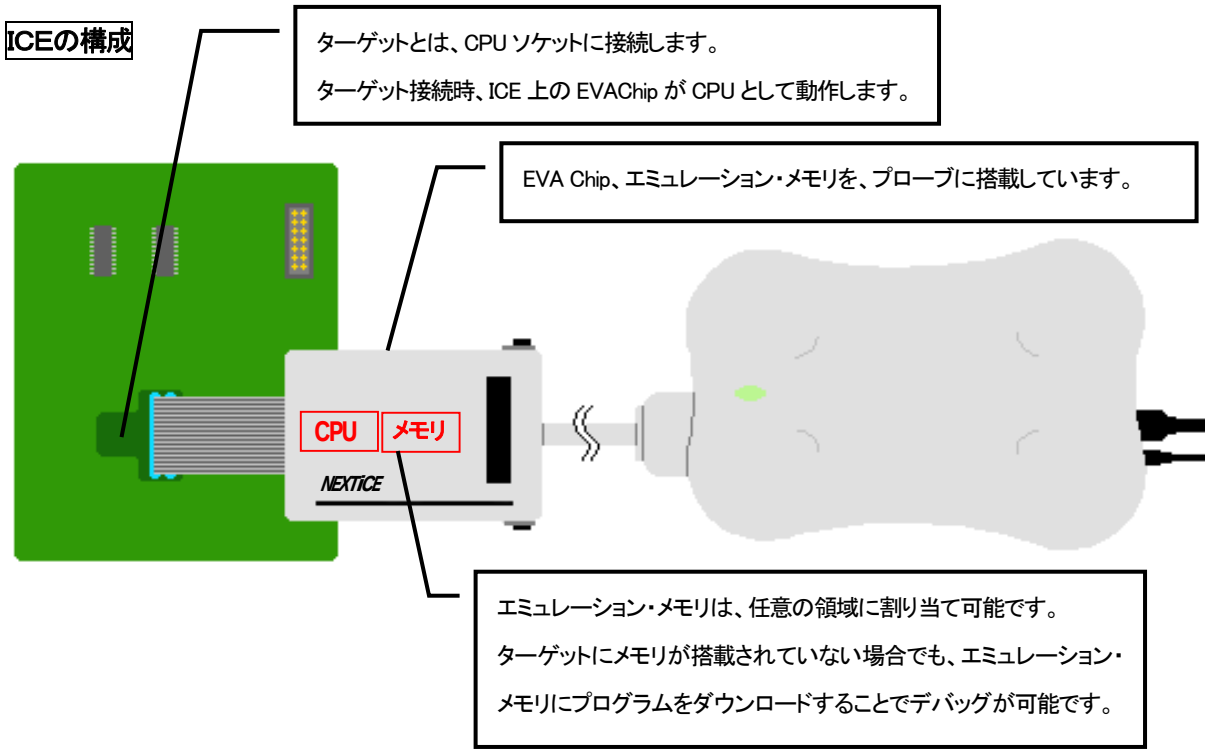
CSIDE を使用する上で便利な使い方について解説します。
Chapter5 までよりも実践的な各機能の使用方法について説明を行います。

Chapter 2.PALMiCE2 について

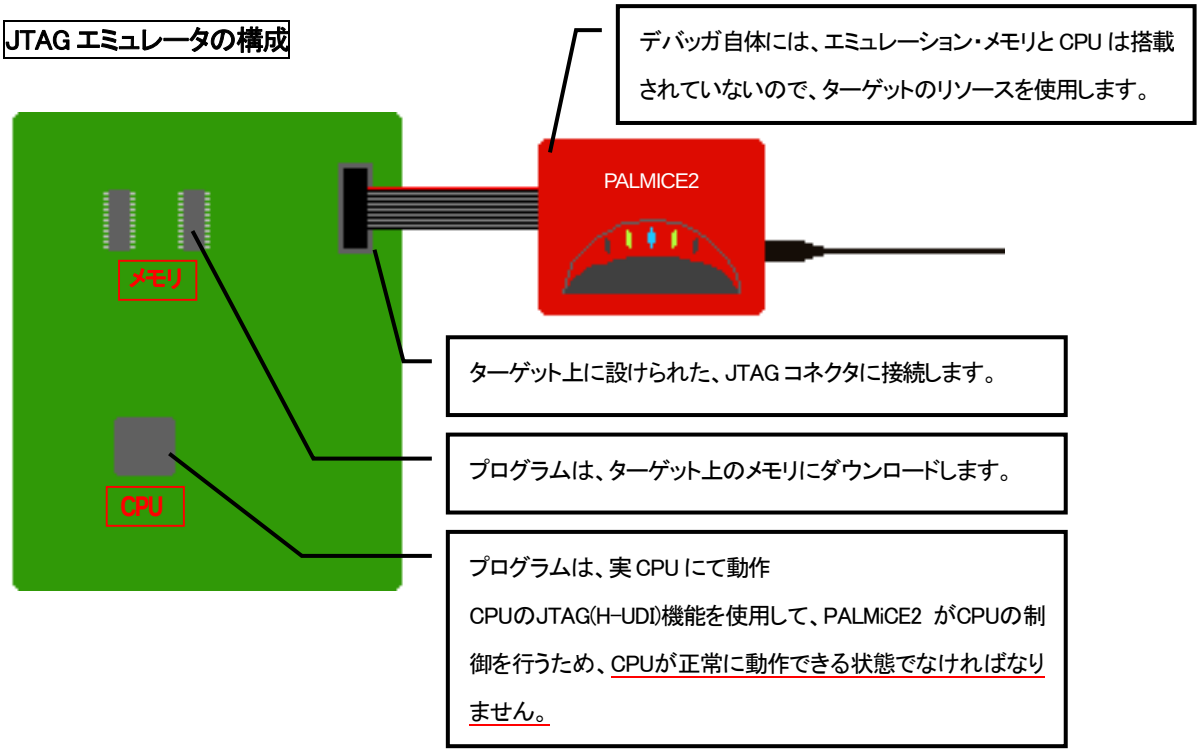
PALMiCE2 は、CPU に内蔵される JTAG 機能を使用してデバッグを行うオンチップ・デバッグ対応の JTAG エミュレータです。最近多くの CPU が JTAG インターフェースを実装しており、CPU 自体がデバッグインターフェースを持つことでデバッグ手法も変化しています。

従来の ICE(In-Circuit Emulator)と比較して JTAG エミュレータとでは、大きく以下の点が異なります。

	ICE	JTAG エミュレータ
デバッグ時の CPU	EVA Chip	実 CPU
エミュレーション・メモリ	有り	無し



JTAG エミュレータの構成



※PALMiCE2 には、オプションでエミュレーション・メモリユニットをご用意しています

デバッグの環境は ICE と JTAG エミュレータでは上図のように異なりますが、基本的なデバッグ機能は同じであり、最終的なデバッグ結果に違いは出ません。

JTAG エミュレータである PALMiCE2 では、ターゲット上の実 CPU を動作させてデバッグを行うため、PALMiCE2 を外せばターゲット単体での動作が可能な点などデバッグの透過性が高い所が特長と言えます。ユーザのメモリ資源は一切使用しませんし(※1)、プログラムの実行中にターゲットにアクセスすることもないため、ターゲット単体での動作に近い形でのデバッグを行うことが出来ます。

※1　CSIDE for PALMiCE2 SH7055 シリーズを除く

Chapter 3.デバッグの開始準備

CSIDE を使用してデバッグを行う前に、予め準備しておいていただく点について説明します。

以下の準備が必要となります。

・ソフトウェアのインストール

・ライセンス・ファイルの取得

・USB ドライバのインストール

・デバッグ対象ターゲットの構成の把握

・コンパイル環境の準備

各所の詳しい操作方法は、ユーザーズ・マニュアルをご覧いただき設定を行ってください。

3-1 ソフトウェアのインストール

使用するパソコンに、デバッガソフト CSIDE とライセンス・ツールをインストールします。

付属の CD-ROM を CD/DVD ドライブへ挿入すると以下の画面が表示されますので、必ずライセンス・ツール、CSIDE の順にインストールを行ってください。

①ライセンス・ツールをインストールします。

ライセンス・ツールは、3 種類のツールで構成されます。

・CSIDE アップデート・ウィザード

・ライセンス申請ウィザード

・ライセンス・ビューア

②CSIDEをインストールします。

Note

・各ソフトウェアの最新バージョンは、それぞれ以下の方法で入手できます。

ライセンス・ツール：当社ホームページのサポートページからダウンロードを行ってください。

CSIDE　　：サポート制度の期間内であれば、ライセンス・ツール“CSIDE アップデート・ウィザード”にてダウンロードができます。バージョンアップ時には、パスワード等はありません。

・CD-ROM を使わずに CSIDE をインストールする

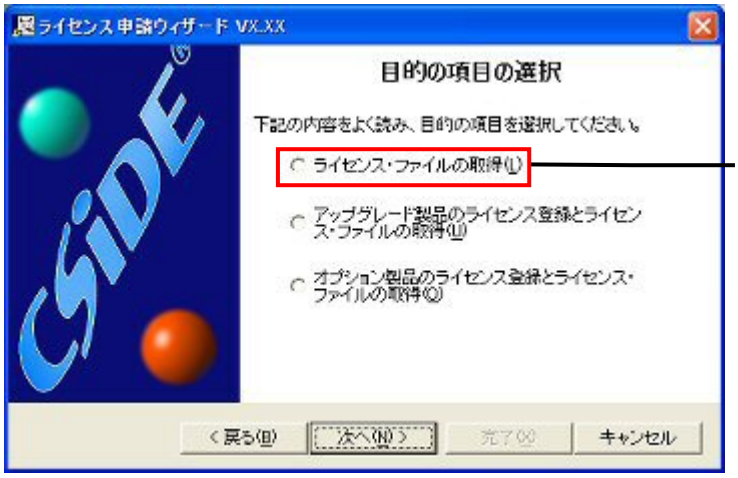
CSIDE アップデート・ウィザードを使用してダウンロードを行う CSIDE のセットアップファイルは、アップグレード専用ではなくこれを使用して新規にインストールすることが可能です。お手元に CD-ROM が無い場合や CD-ROM の CSIDE のバージョンが古い場合に以下の方法でインストールを行うことができます。

1. 弊社ホームページからライセンス・ツールをダウンロードしてください。
2. ライセンス・ツールをインストールしてください。
3. CSIDE ライセンス申請ウィザードより、ライセンス・ファイルを取得してください。
4. CSIDE アップデート・ウィザードより CSIDE のダウンロードを行ってください。
   ※サポート期間内でなければ、CSIDE アップデート・ウィザードより CSIDE のダウンロードを行うことができません。
5. ダウンロードしたセットアップファイルは、ライセンス・ツールのインストールフォルダ ¥LICENSE¥UPDATE に保存されます。
6. セットアップファイルを実行します。
7. 実行時に使用可能なバージョンであるかライセンスの確認が行われます。取得済みのライセンス・ファイルの一覧から使用するライセンスキーを選択してください。
8. 後は画面に従いインストールを行ってください。

3-2 ライセンス・ファイルの取得

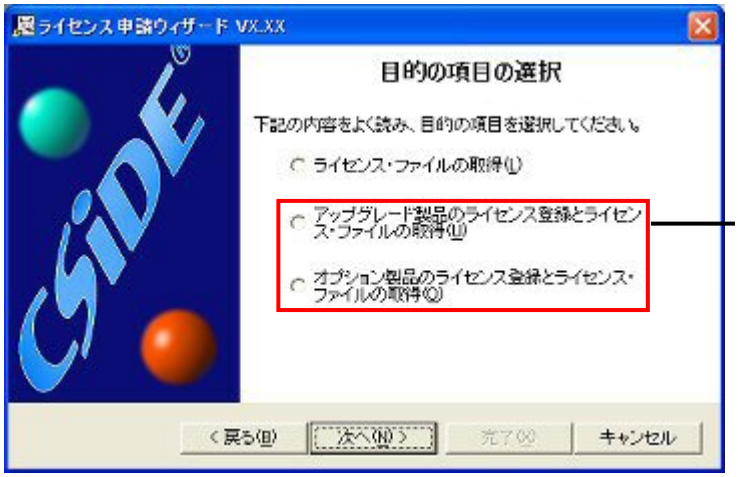
PALMiCE2 を使用するには、使用するパソコンにライセンス・ファイルを取得する必要があります。

スタートメニューから“ライセンス申請ウィザード”を起動させ、ライセンス・ファイルを取得します。



“ライセンス・ファイルの取得”を選択し、[次へ]を押し、ウィザードに従って操作を進めてください。

RTOSデバッグライブラリなどオプション製品やアップグレード製品をご購入された場合には、オプション製品、もしくは、アップグレード製品のライセンスの登録とライセンス・ファイルの取得を行います。



登録されるアップグレード製品、オプション製品の何れかに合わせて選択し、ウィザードに従いライセンスの登録とライセンス・ファイルの取得を行います。

ライセンス・ツールの使用方法に関して詳しくは、ライセンス・ツール・マニュアルを参照してください。

注意

・アップグレード製品、オプション製品のライセンスの登録は、1 ライセンスにつき 1 台の PALMiCE2 にしか登録することはできません。また、一度登録されたライセンスを、他の PALMiCE2 のライセンスに登録しなおすことはできません。

・アップグレード製品、オプション製品のライセンス登録は、1 度行うと PALMiCE2 のライセンスに登録されます。登録以降は、PALMiCE2 のライセンスを再取得すれば、アップグレード製品、オプション製品もお使いいただけます。

・アップグレード製品、オプション製品付き PALMiCE2 をレンタルでご使用の場合には、予めアップグレード製品、オプション製品の登録が PALMiCE2 のライセンスにされていますので、再度オプション登録を行っていただく必要はありません。

Note

PALMiCE2 は、1ユーザ1ライセンスとなっています。同一ユーザ内で複数のパソコンにて 1 台の PALMiCE2 を使用される場合には、使用するそれぞれのパソコンにてライセンス・ファイルを取得することで使用可能です。

3-3 USB ドライバのインストール

PALMiCE2 とパソコンを接続し PALMiCE2 の電源を入れ、USB のドライバをインストールしてください。
お使いの OS によってドライバのインストール手順が異なりますので、詳しくは PALMiCE2 ユーザーズ・マニュアルをご確認ください。

Note

最新の USB ドライバは、当社ホームページにてダウンロードすることができます。

3-4 デバッグ対象ターゲットの構成の把握

CSIDE を使用する際に、デバッグ対象となるターゲットの情報を設定する必要があります。
デバッグを開始する前にお使いのターゲットの構成を確認しておいてください。

- CPU 名(CPU コア名)/動作クロック
- CPU 動作モード
- ROM の種類、接続バス幅、容量(フラッシュ・メモリの場合、メーカ名、型名)
- メモリ・マッピング

3-5 コンパイル環境の準備

デバッグ対象となるオブジェクト・ファイルのコンパイル環境を準備します。
CSIDEでの高級言語レベル・デバッグは、コンパイラが出力するデバッグ情報を読み込むことで実現するため、デバッグ時にはデバッグ情報付きオブジェクト・ファイルをダウンロードする必要があります。デバッグ情報の出力などデバッグの際に必要となるコンパイラ・オプションの解説を、オンライン・マニュアルの［対応言語について］欄に記載していますのでお使いのコンパイラの欄を必ずご確認ください。
なお、[対応言語について]に記載のないコンパイラには CSIDE は対応していません。

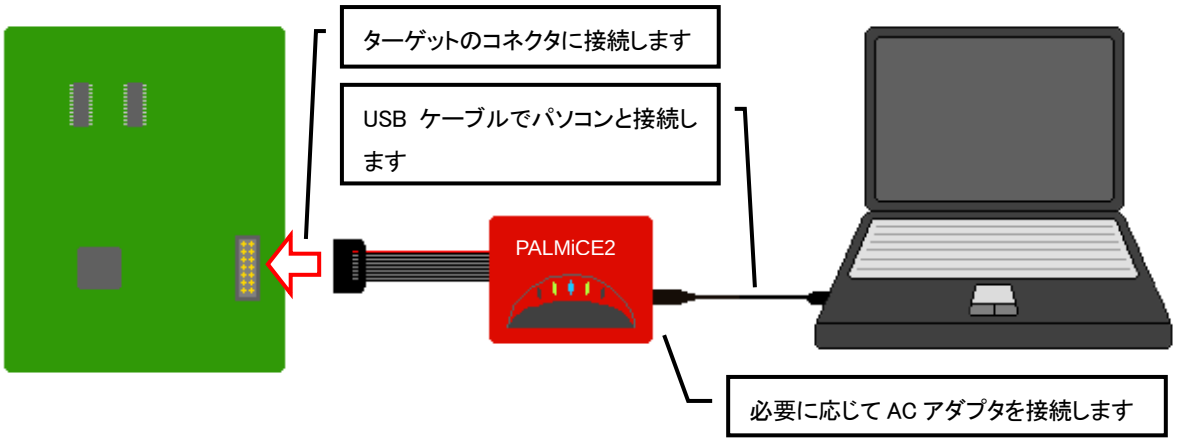
Note

CSIDEには、ソース・ファイルの編集、コンパイルをCSIDE上で行うことができるワークスペース機能が実装されています。この機能を使用することで、ソース編集からコンパイル、デバッグまでをCSIDE上でシームレスに行うことができます。ただし、ワークスペース機能は、必ず使用しなければならない機能ではありませんので、オブジェクト・ファイルの作成は別のソフトで行い、デバッグはCSIDEというように分けてご使用いただくことも可能です。

Chapter 4.ハードウェアの接続

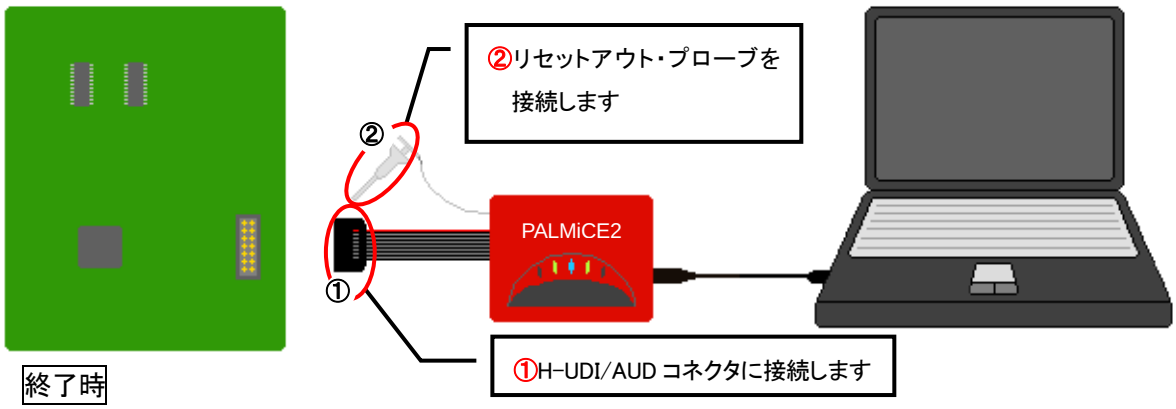
4-1 ハードウェアの接続

PALMiCE2 とターゲットを接続します。
故障の原因となりますので必ずPALMiCE2 とターゲットの電源は切った状態で接続を行ってください。

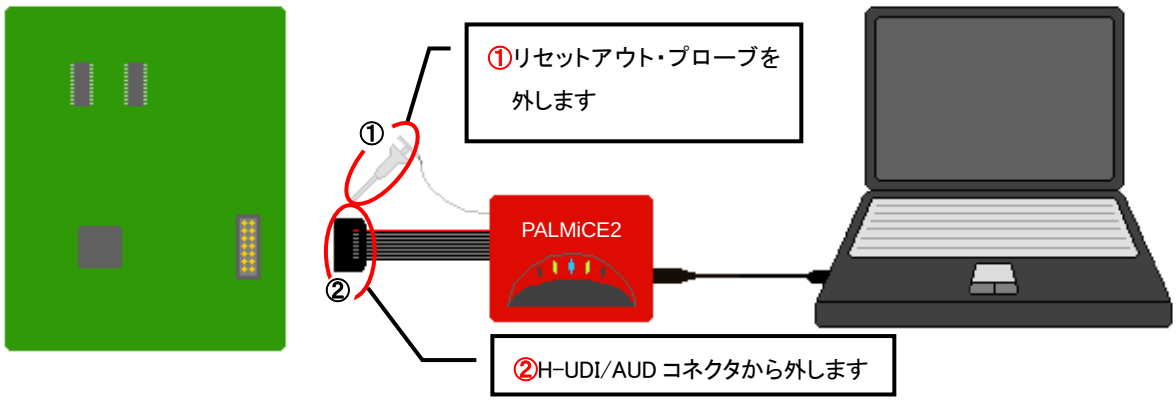


リセットアウト・プローブが付属している機種では、さらに下図のように①→②の順で付け外しを行ってください。

接続時



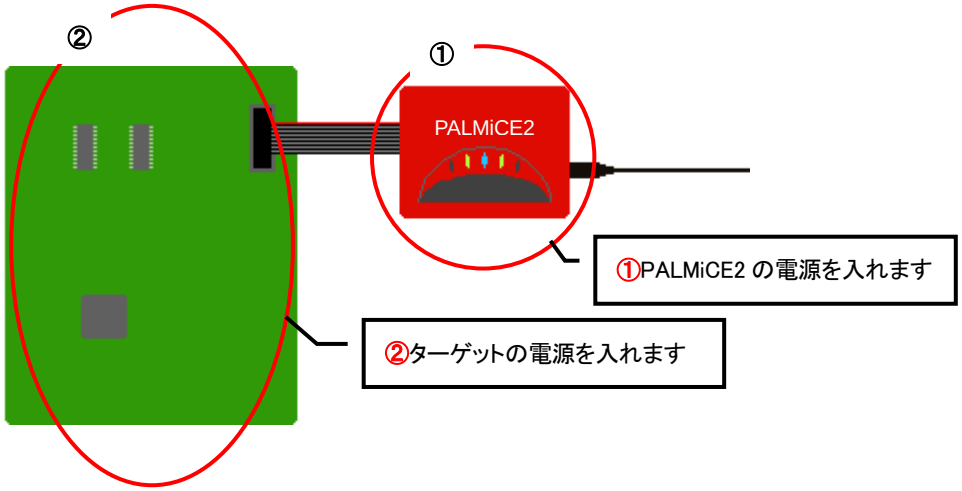
終了時



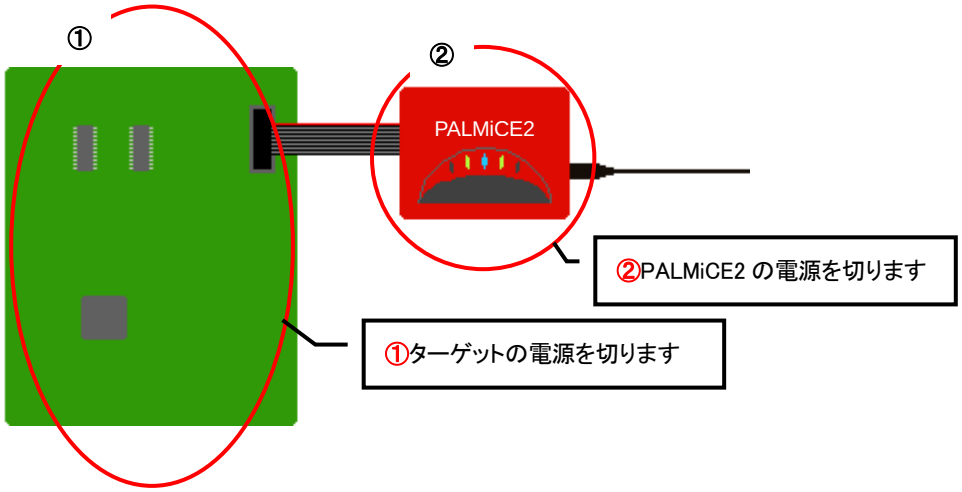
4-2 電源の投入順序

ターゲットと PALMiCE2 の電源は、以下の順で入切してください。

起動時　PALMiCE2 の電源投入後、ターゲットの電源を入れます。



終了時　ターゲットの電源を切った後に、PALMiCE2 の電源を切ります。

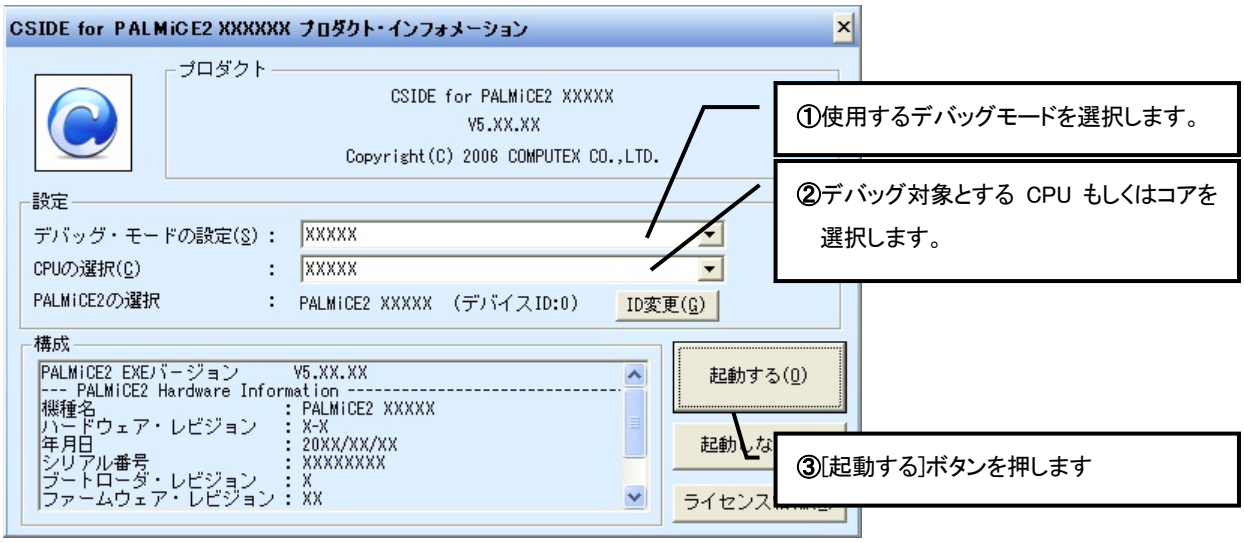


Chapter 5.CSIDE の設定

5-1 CSIDE の起動

スタートメニューから CSIDE を選択し起動させます。
起動後、ツールバー[ターゲット設定]ボタンを押します。

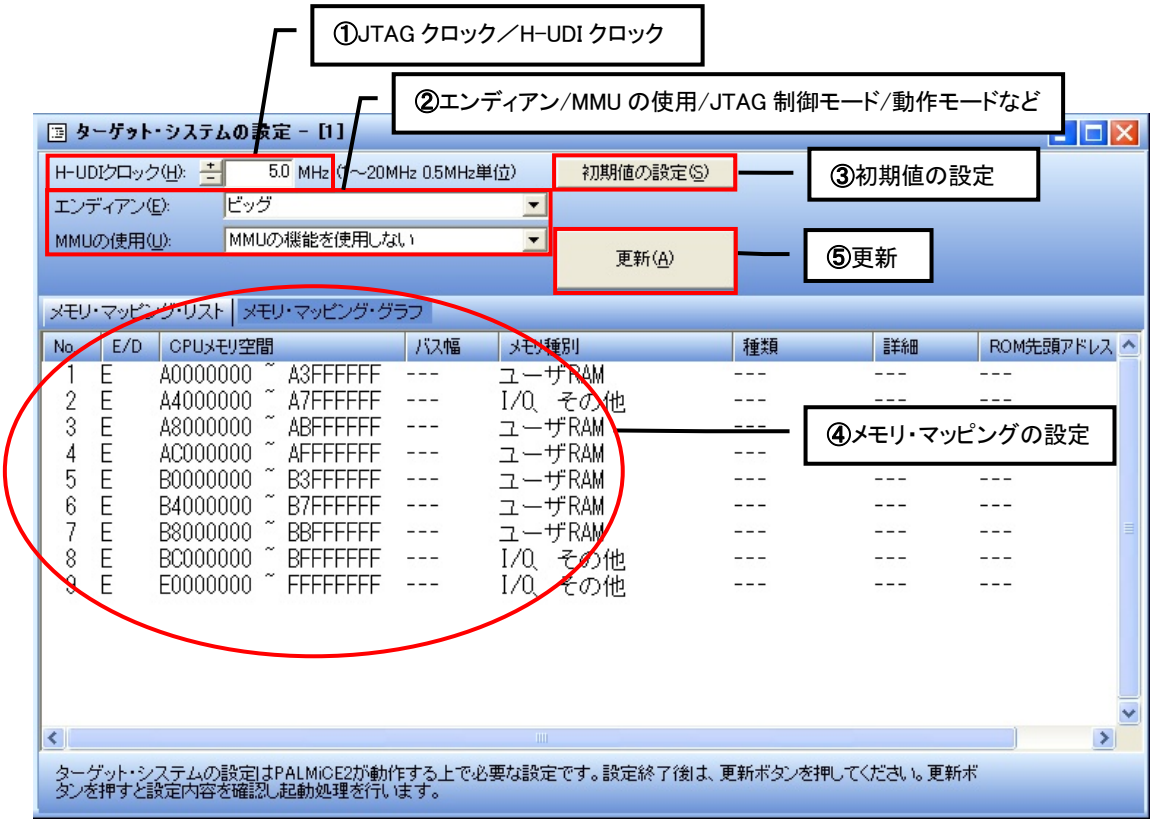
[プロダクト・インフォメーション]ダイアログが表示されます。



注意

CPU の選択を間違えないよう注意してください。
CPU の選択を間違った場合、ターゲット・システムの更新処理が正常に行えない一因となります。
選択した CPU は、起動後にヘルプメニュー[インフォメーション]で確認することができます。

画面上に[ターゲット・システムの設定]ダイアログが表示されますので設定を行います。
[ターゲット・システムの設定]の表示項目や設定項目は、使用するCPUによって異なります。ユーザーズマニュアルを参照の上、設定を行ってください。



・ターゲット・システムの設定とは・・・

ターゲット・システムの仕様やメモリ・マップの構成に関する設定などを行い、CSIDE の動作を使用環境に合わせるための設定です。CSIDE は、ここで設定された内容に従い動作しますが、これが実行中のユーザプログラムの動作に影響が及ぶことは一切ありません。もし、設定がお使いの環境に合っていない場合、更新処理が完了できなかったり、メモリへのアクセスが正常に行えなかったりなどの原因となります。

①JTAG クロック／H-UDI クロック

JTAGクロック／H-UDIクロックとは、CPUとPALMiCE2 間の通信クロックです。

クロックが速ければCSIDEの操作の応答が速くなり、遅ければCSIDEの動作が安定します。各CPUの規定の速度よりクロックが速い場合、起動に失敗する、CSIDEが不安定な状態となるなどの原因となります。この場合、クロックを下げることで症状が改善することがあります。設定可能な値は、使用するCPU、環境によって上限が異なりますので、ユーザーズマニュアルの記載を確認の上、設定を行なってください。

②エンディアン/MMU の使用/JTAG 制御モード/動作モードなど

ターゲット・システムのエンディアンや MMU の有無などが表示されます。使用するターゲット・システムの状態に合わせて設定を行ってください。

③初期値の設定

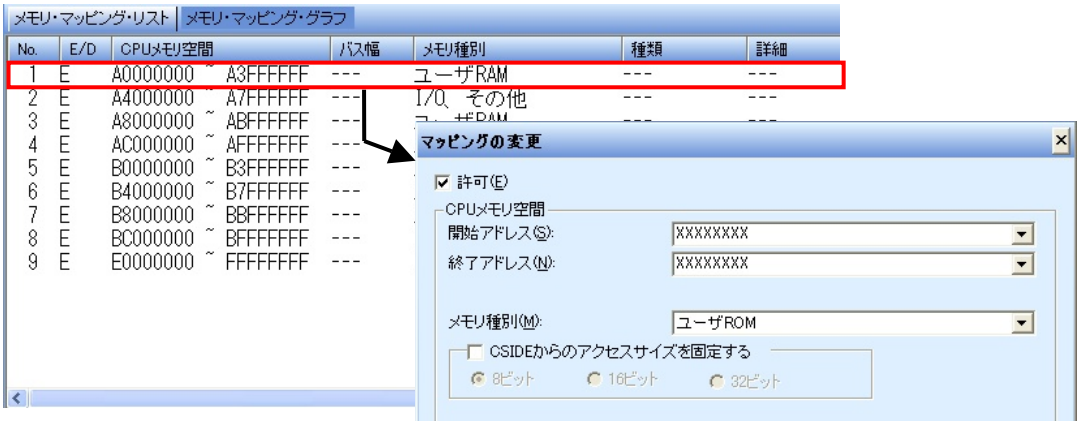
ターゲット・システムの仕様や CSIDE の動作に関する設定を行います。使用するターゲット・システムの状態に合わせて設定を行ってください。

④メモリ・マッピングの設定

ターゲット・システムのメモリ・マッピングを、メモリ空間ごとに種類、割付アドレス等の設定を行います。

CSIDE は、ここで設定された内容に従い動作し、マッピングされているアドレスにのみ CSIDE 上でアクセスが可能となりますので、CSIDE からアクセスしたいアドレスを登録してください。

なお、マッピングされていないアドレスに対しては、メモリが実装されていたとしても CSIDE からアクセスすることはできませんので、アクセスすることでロックするような領域がある場合には、メモリ・マッピングを行わないことで CSIDE からの不当なアクセスを防ぐことができます。(フラッシュメモリの設定は、5-2 フラッシュメモリの設定を参照してください。)



⑤更新

更新ボタンを押すと、[ターゲット・システムの設定]ダイアログに設定された内容で、CPU と PALMiCE2 の通信を開始します。

[ターゲット・システムの設定]ダイアログの設定後、[更新]ボタンを押してください。
CPU と PALMiCE2 の間で通信が開始され、画面上、処理状況が Step 表示されます。
起動処理では、大まかに以下の処理内容が行われます。

・PALMiCE2 の初期化
・リセットの入力
・JTAG 信号の確認(バイパスチェック)
・モニタのダウンロード
・データ送受信チェック

問題なく起動処理が行われると、「起動処理は正常終了しました。」と表示され処理が終了します。
正常に起動処理が終了すると、PALMiCE2 からターゲット・システムの制御が可能となり、デバッグを開始することができます。

なお、更新処理中に何らかの問題が発生した場合、Step が停止しエラーが表示されます。
[ターゲット・システムの設定]ダイアログで行った設定が、ターゲット・システムに合っていない可能性があります。また、PALMiCE2 は、CPU と通信し制御を行うため、CPU が正常に動作できていない可能性があります。
表示されるメッセージの内容に従って原因の調査をしてください。

注意

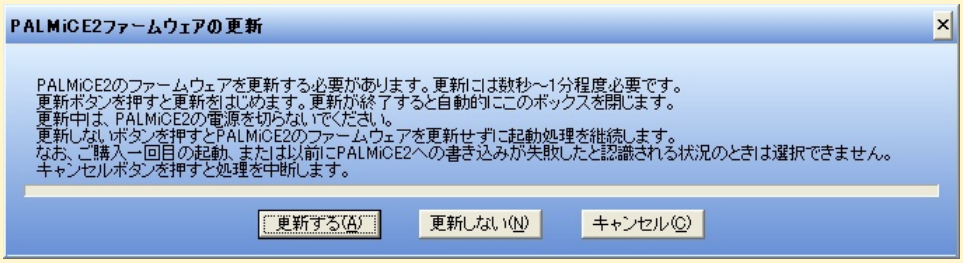
停止する個所とエラーメッセージの内容にもよりますが、起動処理途中に停止する場合に良くある要因として以下の点が考えられます。

- CSIDE 起動時の CPU(コア)選択が間違っている。
- 設定されている JTAG クロックが速い。
- 選択しているエンディアンが間違っている
- 起動時に正常に PALMiCE2 からリセットが入力できない。または、リセットの入力確認ができない。
- CPU の端子がフローティングしているなどして CPU が動作できない状態にある。
- JTAG 信号の接続に問題があり、通信の確認が行えない。
- リセットや NMI などが入りっぱなしの状態になっている。
- リセット解除待ち時間の設定時間が短い。

Note

・ファームウェアの更新について

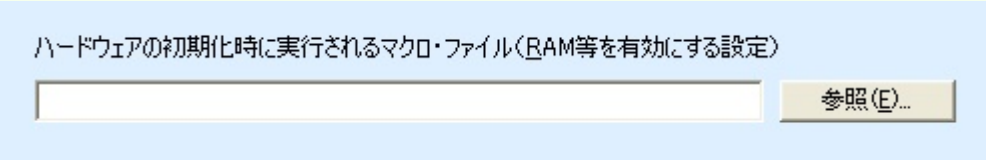
初めてお使いなる場合や使用する CSIDE のバージョンが変わった時に、更新処理の開始時にファームウェアの更新が表示されます。このダイアログが表示されましたら、[更新する]ボタンを押して、ファームウェアの更新を行ってください。
なお、ファームウェアはバージョンごとに用意されており、バージョンダウンする場合にも表示されます。古いバージョンのファームウェアに書き換えることで、使用できなくなるということはありませんので、それぞれのバージョンで更新を行ってください。



Point!

初期化マクロについて

[初期値の設定]ダイアログには、”ハードウェアの初期化時に実行されるマクロ・ファイル”という設定項目が各 CSIDE にあります。登録されたマクロ・ファイルを、ターゲット・システムの更新処理中に実行する機能です。これを利用することでターゲット・システムの初期化を簡単に行うことができます。



ターゲット・システムの更新直後は、CPU はリセット直後の初期状態で起動します。
このため、例えば SDRAM にファイル・ロードを行おうとした場合、CPU の各レジスタの設定を行わなければダウンロードすることができません。この設定を、起動ごとに手動で行っていたのでは効率が悪く大変手間となります。
そこでマクロ・ファイルにターゲットの初期化を行うコマンドを記述し、これに登録することで、ターゲット・システムの更新やハードウェアの初期化時にマクロ・ファイルの内容が自動的に実行され、起動直後から SDRAM へのアクセスを行うことが可能となります。
このようにターゲットの初期化やマクロの定義などに使用すると便利な機能です。

マクロ・ファイルの作成方法・・・
マクロ・ファイルは、テキストエディタにてコマンドを記述し、ファイルの保存時に拡張子.mcr にて保存することで作成できます。単純に内蔵レジスタの設定を行うのみであれば、Outport コマンドを使い記述することで初期化マクロを作成できます。

注意

・実行を行うコマンドを、ここに登録するマクロ・ファイルに含めますと正常に起動処理が行えなくなります。ここでは、ターゲットの初期化を行うコマンドやマクロの定義などを記述してください。

・ターゲットを単体で動作させる場合、マクロ・ファイルで行っているターゲットの初期化内容をユーザプログラムに忘れずに反映してください。

Note

CSDIE には、GUI 上で行う様々な操作を実行することが出来るコマンドが用意されています。CSIDE 上の一連の操作をコマンドで記述して、ファイルにしたものをマクロ・ファイルといいます。詳しくは、オンライン・マニュアルのコマンドリファレンス欄を参照してください。

5-2 フラッシュ・メモリの設定

CSIDE 上から外付けのフラッシュ・メモリの書き換えは、[ターゲット・システムの設定]でフラッシュ・メモリをマッピングすることで可能となります。

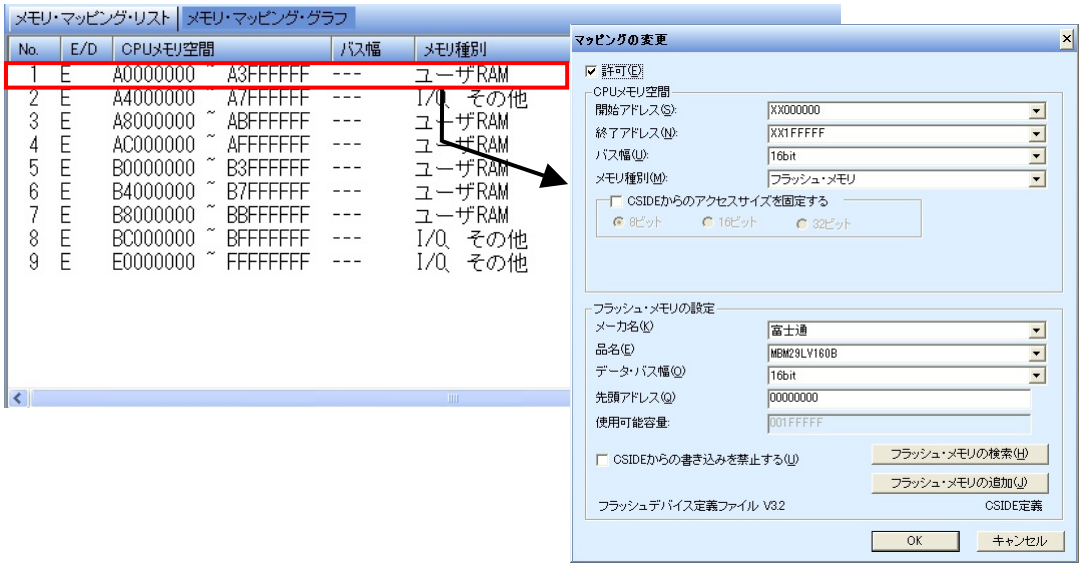
注意

外付けフラッシュ・メモリへの書き込みが出来るか出来ないかは、[ターゲット・システムの設定]でのメモリ・マッピングの設定で決まります。ファイルメニュー[フラッシュメモリ・マップ]にも、フラッシュ・メモリに関する設定がありますが、これは起動後の CSIDE の動作に関する設定であり、書き換えに関して影響する設定ではありません。

Note

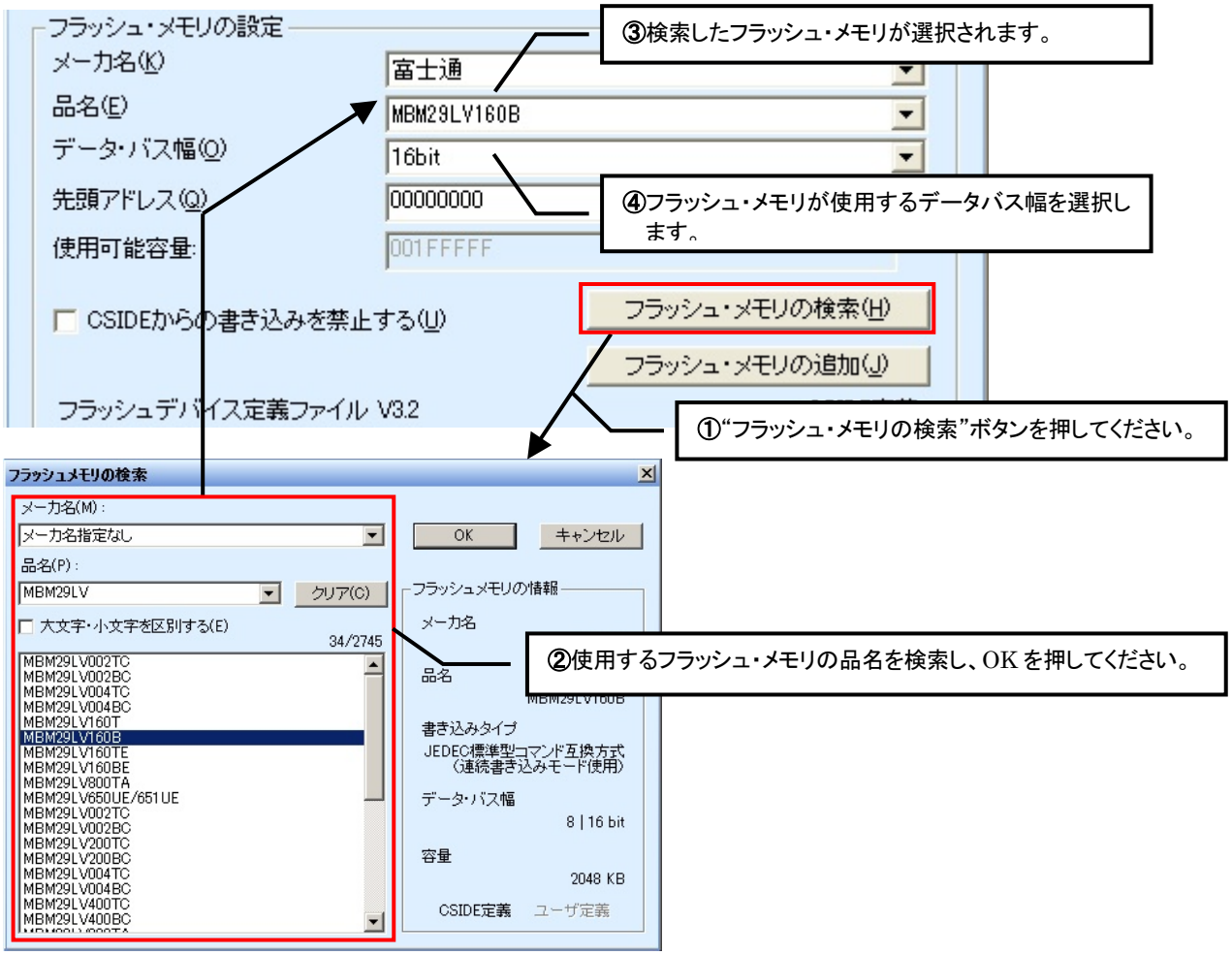
・フラッシュ・メモリの書き換えを CSIDE 上で行わず、メモリの内容を参照するだけであれば、メモリ種別を”ユーザ ROM”と設定しておくことで、メモリ・ウィンドウ等から参照することができます。

[ターゲット・システムの設定]ダイアログから、ポップアップメニュー[追加]、もしくは、[変更]を選択しメモリ・マッピングのダイアログを開きます。



使用するフラッシュ・メモリの設定を行います。

メモリ種別で”フラッシュ・メモリ”を選択すると、フラッシュ・メモリの設定欄が表示されます。



フラッシュ・メモリの設定にて、以下の点に注意してください。

・フラッシュ・メモリの品名

フラッシュ・メモリの品名を間違って選択している場合には、正常にアクセスできません。

・バス幅の設定

CPU メモリ空間のバス幅、フラッシュ・メモリで使用するバス幅、それぞれを設定します。

◎1個のフラッシュ・メモリを 16bit バスで接続する場合

CPU メモリ空間　　:16bit

フラッシュ・メモリの設定　:16bit

◎複数個のフラッシュ・メモリを並列に接続する場合

16bit バスのフラッシュ・メモリを 2 個使用して、CPU に 32bit バスで接続するとすると

CPU メモリ空間　　:32bit

フラッシュ・メモリの設定　:16bit

と設定します。CSIDE は、バス幅の組み合わせによりフラッシュ・メモリの接続個数を自動的に判断します。また、使用可能容量も選択されたバス幅に応じて変化しますので、使用可能容量を見ることで確認することができます。

・先頭アドレス

通常は、初期値の 00000000 のままで問題ありません。

この設定項目は、フラッシュ・メモリの先頭以外からを CPU のメモリ空間に割り付ける場合に、フラッシュ・メモリの何番地からメモリ空間に割り付けるのかを指定します。例えば、00010000 と指定した場合、フラッシュ・メモリの 10000 番地からを CPU のメモリ空間に割り付け、0 番地から FFFF 番地までを使用しないという設定になります。

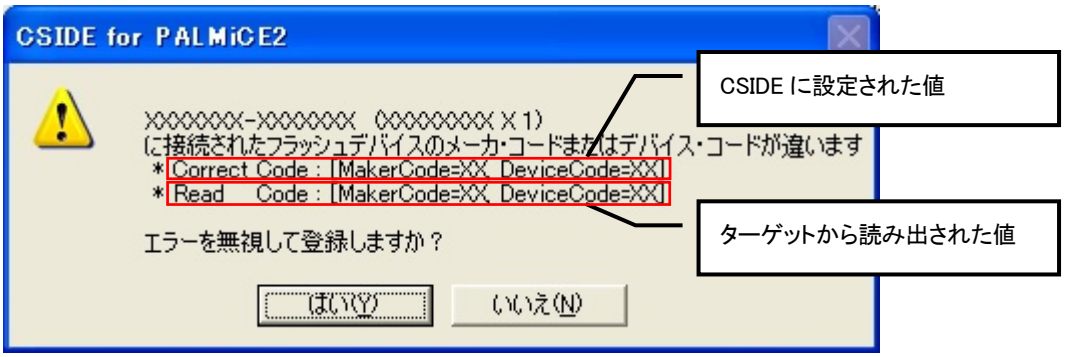
Note

・[フラッシュ・メモリの検索]で表示されないフラッシュ・メモリは、[フラッシュ・メモリの追加]で定義パラメータを追加することで使用可能となります。(一部、未対応なデバイスもあります。)

設定は以上です。設定後、[ターゲット・システムの設定]から更新を行ってください。

フラッシュ・メモリの設定が行われると CSIDE では、更新処理の最後に設定された内容でフラッシュ・メモリにアクセスが行えるか確認のために、メーカ・コード、デバイス・コードを読み出して CSIDE の設定値と比較します。正常に読み出すことができて、コードが一致した場合には、メッセージ等は表示されずに起動処理が終了します。この場合、CSIDE 上からフラッシュ・メモリへの書き換え、消去が可能となります。

この時、コードが一致しなかった場合、以下のエラーが表示されます。



このエラーメッセージは、フラッシュ・メモリからコードを読み出すことができない、もしくは、CSIDE の設定値が間違っていることを示しています。このため[はい]を選択しエラーを無視して登録を行っても、基本的にはフラッシュ・メモリの書き換えは行えません。

フラッシュ・メモリの書き換えを行うためには、このメッセージが表示されないようにしなければなりません。
このメッセージが表示される要因としは、以下の事が考えられます。

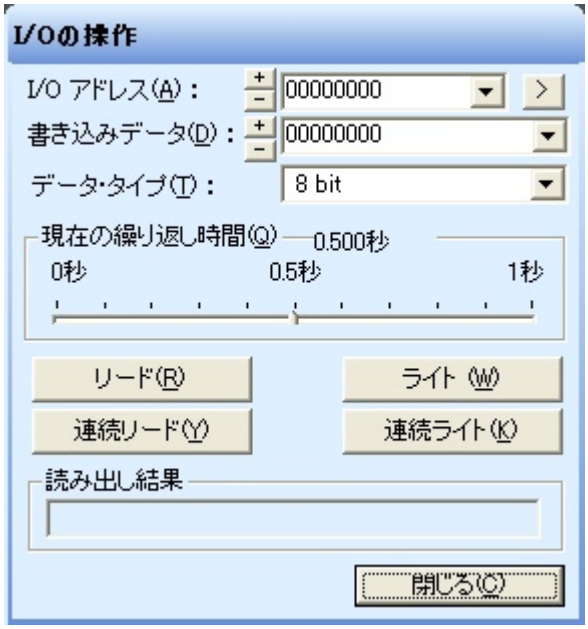
・CSIDE で選択しているフラッシュ・メモリの品名が間違っている。
・CSIDE の設定に誤りがあって読み出せない
・使用しているハードウェアに問題があって読み出せない

◎原因の切り分け方法

エラーメッセージが表示される原因が、CSIDE の設定上の問題か、ハードウェアの問題かを切り分ける方法として、手動でコードの読み出しを行う以下の方法があります。

1. エラーが表示されたら[いいえ]を押して、CSIDE を起動させます。

2. データメニュー[I/O の操作]を開きます。



3. フラッシュ・メモリのデータシートに記載のあるメーカ・コード、デバイス・コードの読み出しコマンドをデータシートの手順に従って入力してください。

4. メーカ・コード、デバイス・コードが読み出せたか、読み出せなかったかで以下の判断ができます。

読み出せた場合　→　CSIDE の設定に誤りがあると考えられます。
読み出せなかった場合　→　ハードウェアに問題がある可能性が考えられます。

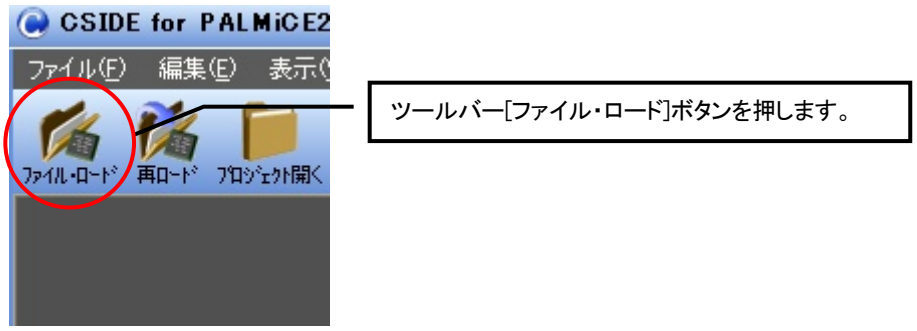
Chapter 6.デバッグの開始

正常に更新処理を終了することができましたら、プログラムをダウンロードしてのデバッグを行います。
ここでは、一通りの基本動作の確認を例に説明を行います。

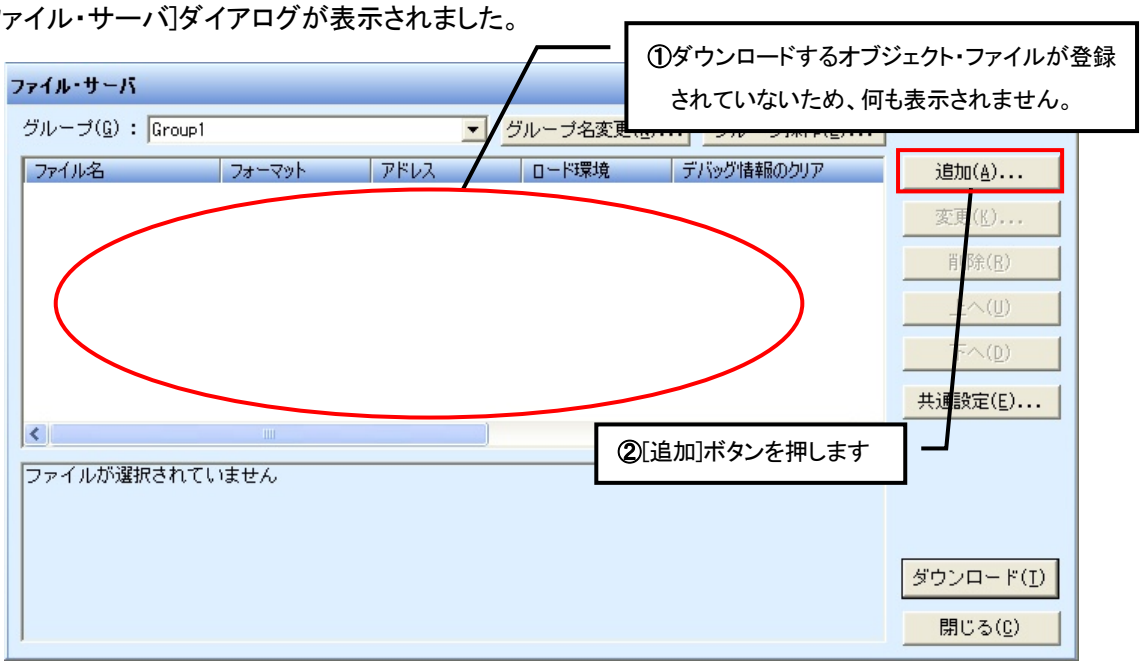
6-1 ファイル・ロード

デバッグを行うために、まずは、プログラムのダウンロードを行います。

ツールバー[ファイル・ロード]ボタンを押してください。



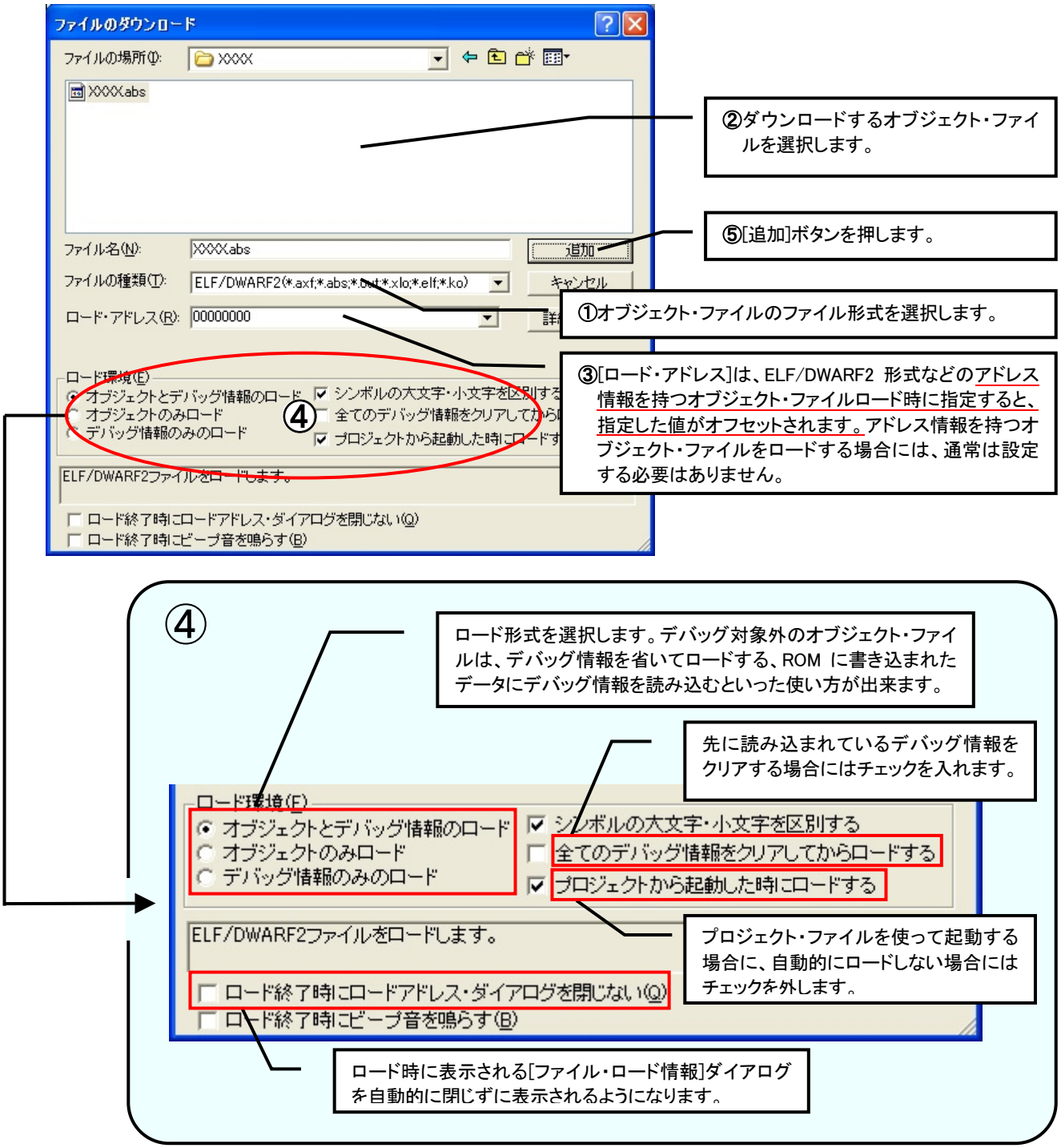
・[ファイル・サーバ]ダイアログが表示されました。



・ファイル・サーバとは・・・

ダウンロードを行うオブジェクト・ファイルを管理するための機能です。複数のオブジェクト・ファイルの一括ダウンロードやグループ単位でのオブジェクト・ファイルの管理などが容易に行うことができます。

ダウンロードを行うオブジェクト・ファイルを選択します。



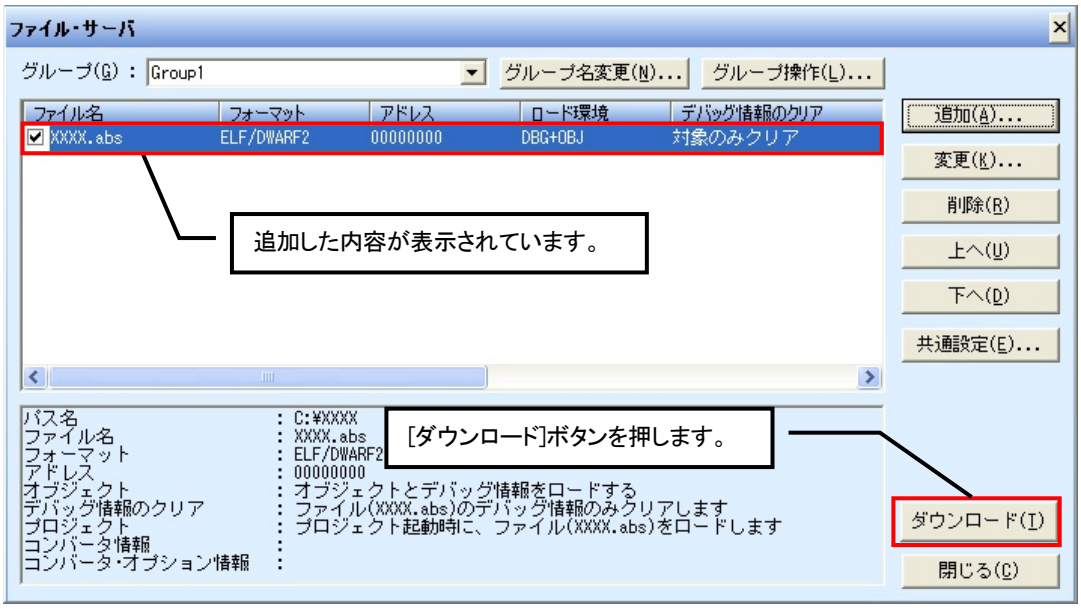
注意

高級言語レベルでのデバッグを行うには、デバッグ情報付きオブジェクト・ファイルのロードが必要となります。デバッグ情報付きオブジェクトの形式は、お使いのコンパイラによってファイル形式が異なりますので、詳しくはオンライン・マニュアルの[対応言語について]をご覧ください。

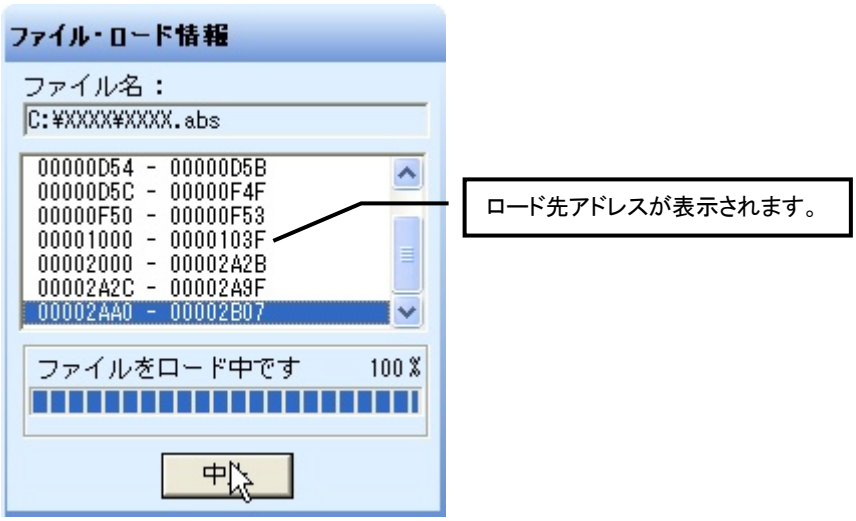
Note

プログラムをどのアドレスに配置するかは、コンパイラの設定で決定します。CSIDEは、その設定されたアドレス情報に従ってダウンロードを行います。プログラムのダウンロード先を変更したい場合には、コンパイラの設定を変更してください。配置アドレスの設定方法については、各コンパイラのマニュアルを参照してください。

ファイル・サーバに選択したオブジェクト・ファイルが追加されます。



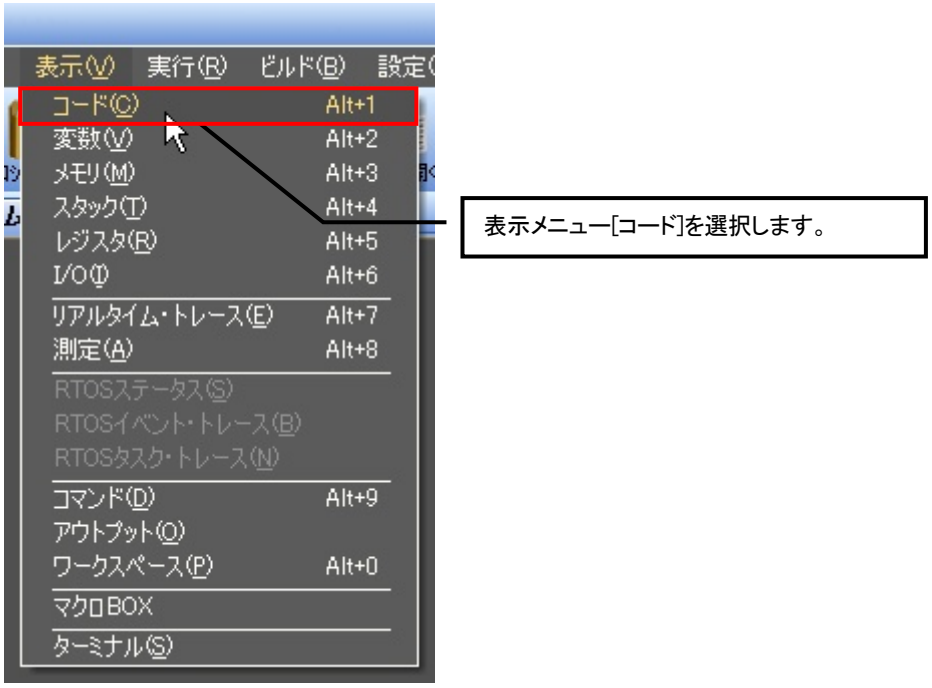
[ダウンロード]ボタンを押すとファイル・ロードが開始され、[ファイル・ロード情報]ダイアログが表示されます。



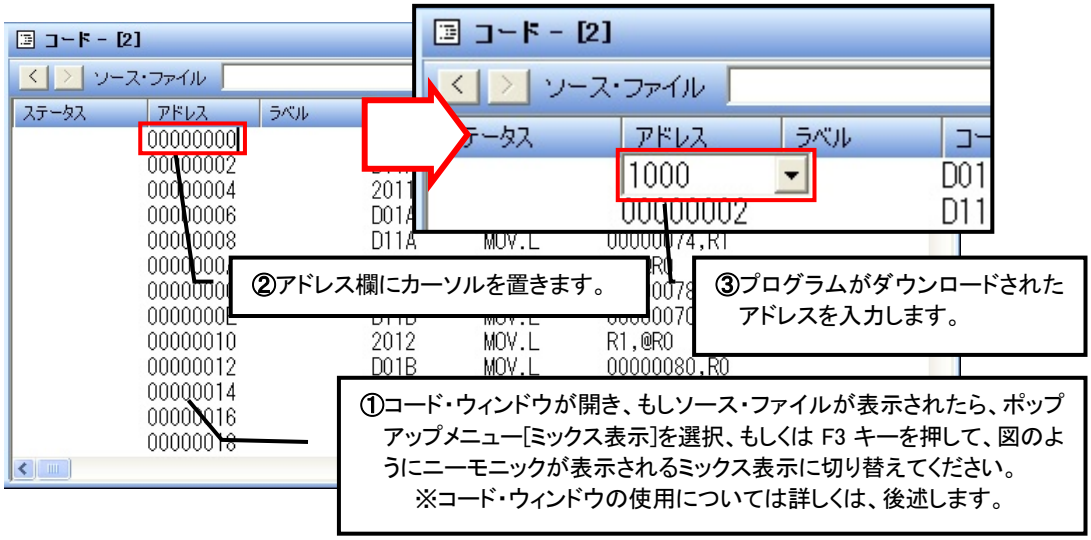
このダイアログはダウンロード終了後自動的に閉じられます。閉じずに表示させる場合には、[ロード終了時にロードアドレス・ダイアログを閉じない]にチェックを入れてください。このダイアログを確認することで、ダウンロードを行ったプログラムが何番地に配置されたかを確認することが出来ます。

ダウンロードが完了しましたら、メモリにプログラムが書き込まれているかを確認します。
なお、初期設定では、メモリへの書き込みに対するベリファイ・チェックは行われないため、ダウンロードがエラーメッセージを表示せず完了したとしても、必ずしもオブジェクト・ファイルがメモリに書き込まれたとは限りません。

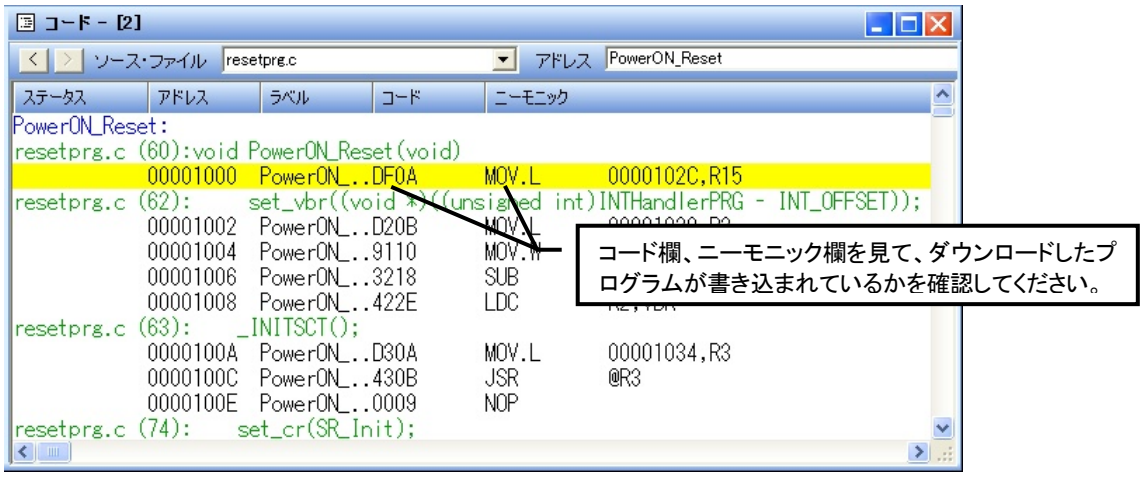
表示メニュー[コード]を選択し、コード・ウィンドウを開きます。



コード・ウィンドウの表示を切り替えます。



表示が入力したアドレスに切り替わります。



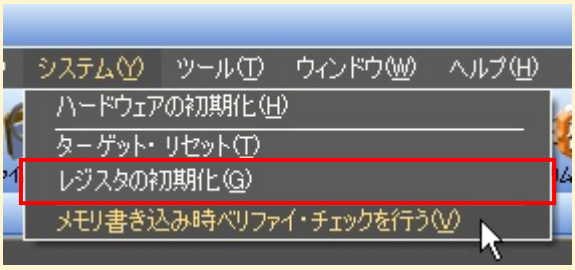
例えば、ニーモニックの内容が正しくなかったり、コード欄のデータが以下のような状態になったりしている場合、メモリが正しく書き換えられていないと考えられます。

・上位のみ、下位のみのデータが全て FF となっている
・規則的に特定の値が並んでいる
・全て FF になっている

このように正しく書き換えられていなかった場合、CPU の設定やターゲットに問題があるかもしれません。

Note

・メモリの書き込みに対してベリファイ・チェックを行うには、システムメニュー[メモリ書き込み時ベリファイ・チェックを行う]を選択してください。フラッシュ・メモリに対しては、ファイルメニュー[フラッシュ・メモリ・マップ]にて設定します。

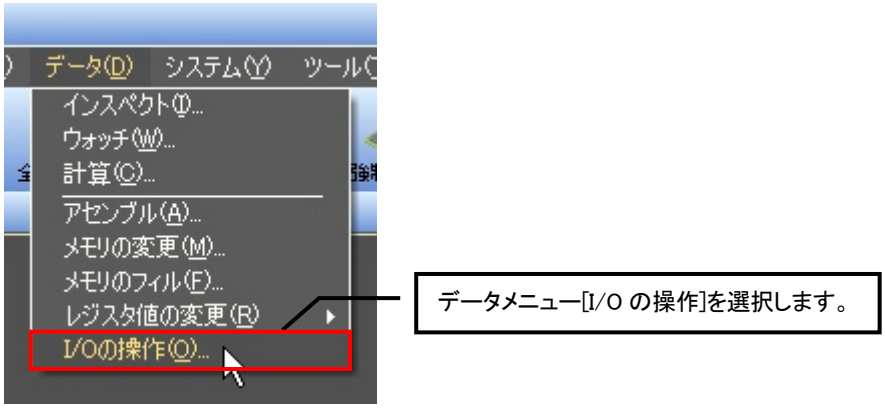


・フラッシュメモリエリアに書き込みが発生した場合、画面左下のステータス・バーに書き換え状況が表示されます。

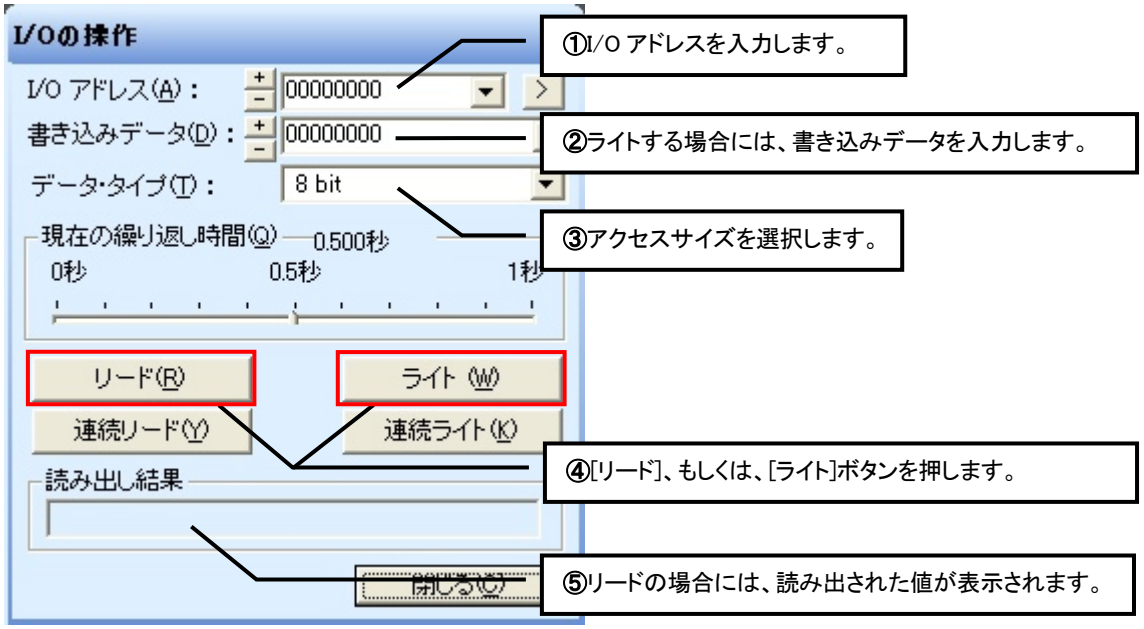
6-2 I/O ポートの設定

プログラムがメモリに書き込まれない場合、メモリに対するアクセスが正常に行えていないことを示しています。アクセスできない原因としては、CPU の各レジスタが未設定であったり設定に誤りがあったりなど、設定上アクセス出来ないことやターゲットにハードウェア上の問題があるなどが考えられます。まずは、CPU の設定がアクセス可能なように正しく設定されているかを確認するために、I/O の操作を使用して各レジスタの値を読み出して確認を行います。

[データ]メニュー[I/O の操作]を選択します。



[I/O の操作]は、指定アドレスの読み出しと書き換えを行うことができます



[I/O の操作]を使用して、ターゲットシステムの初期化の設定に誤りがないかを確認していきます。
CPU の設定に関しては、CPU のハードウェアマニュアルを確認してください。

また、ターゲット自体の問題と思われる場合も、[連続リード]、[連続ライト]を使用すると便利です。
これらは CSIDE から特定番地に対して一定周期で、リード、もしくは、ライトを行う機能です。これを使うと一定周期で各信号が出力されますので、オシロスコープ等で観測しやすくなります。

注意

メモリ・ウィンドウでも I/O レジスタを表示することは可能ですが、ウィンドウの特性上、正しく表示できないなどしますのでご使用は避けてください。I/O レジスタの確認は、I/O の操作、もしくは、I/O ウィンドウを使用してください。

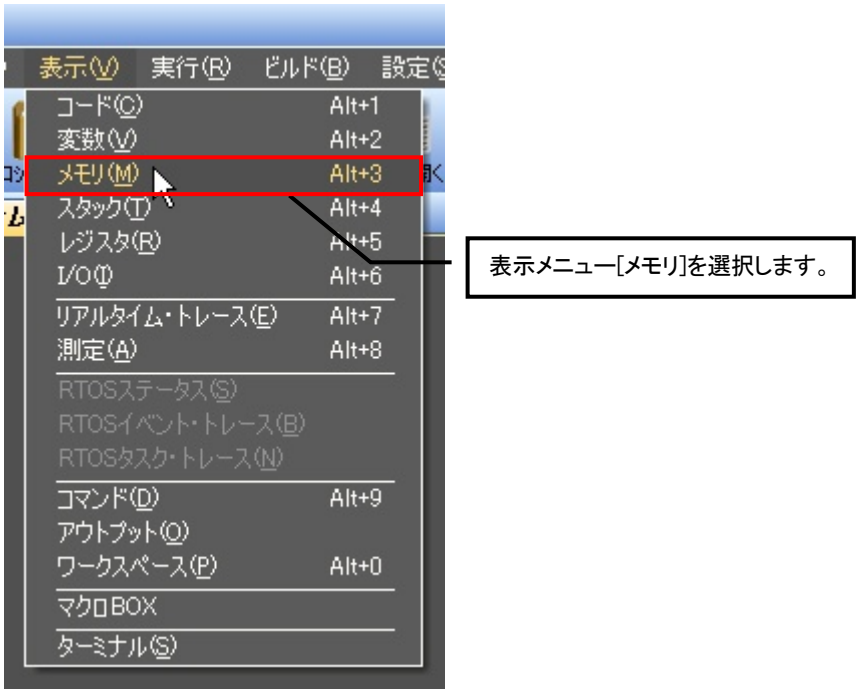
Note

CSIDE には、[I/O の操作]以外に I/O ポートを参照するために I/O ウィンドウがあります。
I/O ウィンドウでは、参照したい I/O ポートを登録することで、ビット単位での状態の確認やブレークごとの I/O ポートの変化の確認などが一覧で表示することが出来ます。

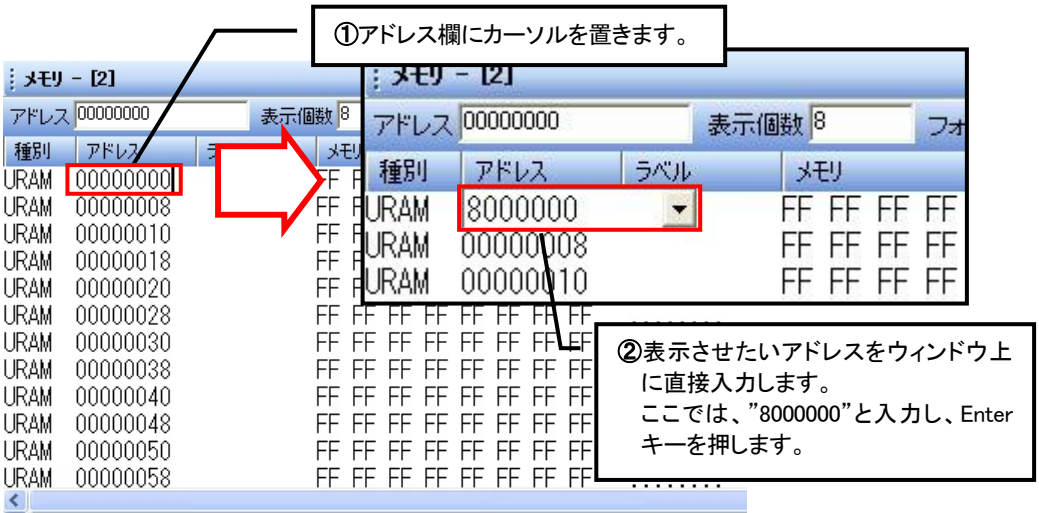
6-3 メモリの書換えチェック

I/O の操作にて各レジスタの設定の確認を行いましたら、再度、メモリアクセスが行えるかを確認してみます。
今度は、メモリ・ウィンドウで確認する方法を説明します。ここでは、8000000 番地に接続された RAM エリアの確認を例に説明を行います。
以下の方法で、正常にメモリの内容を書き換えることが出来れば、正常にプログラムをダウンロードすることができるかと思いますので、再度プログラムをダウンロードしてください。

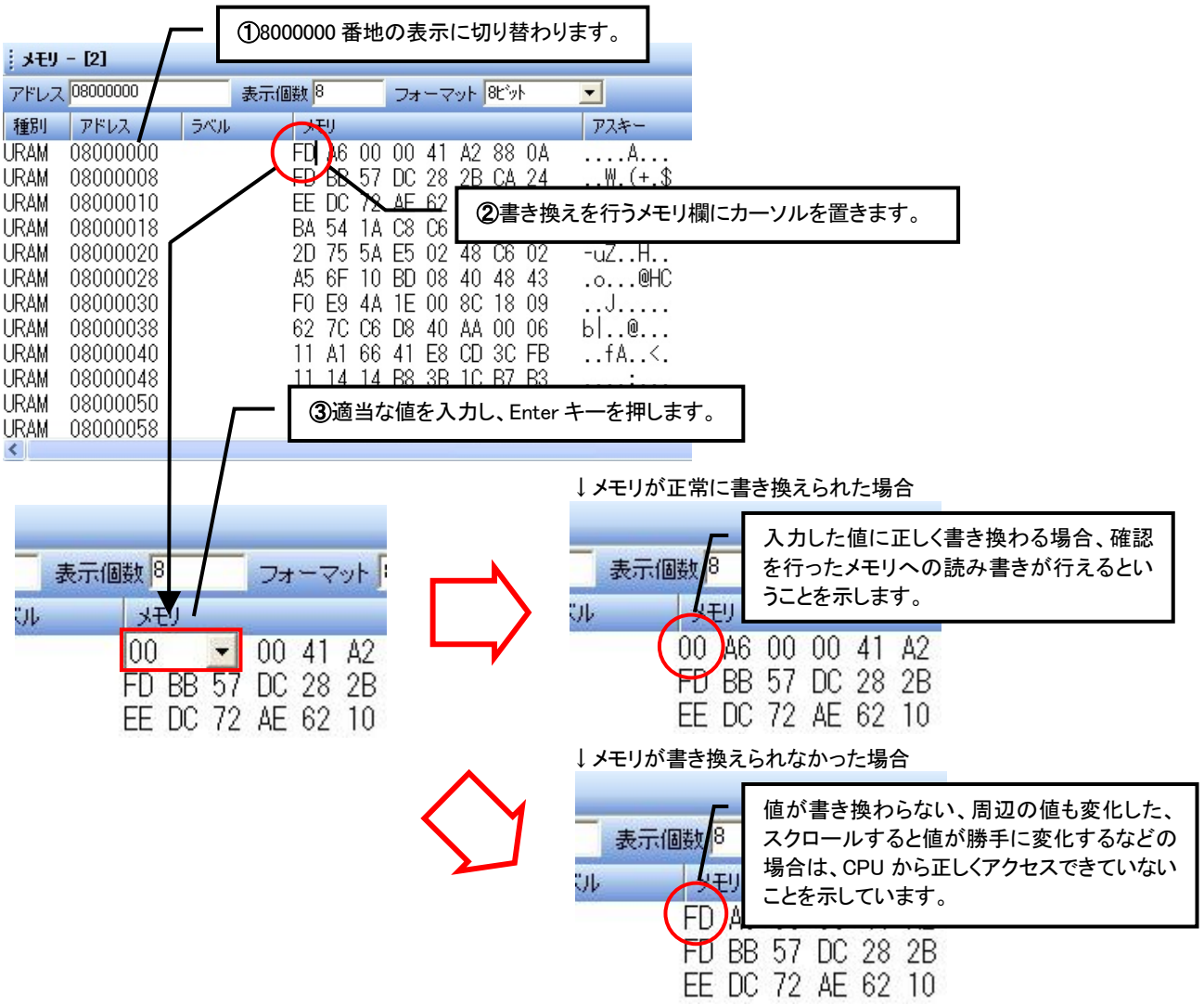
メモリ・ウィンドウを開きます。



メモリ・ウィンドウの表示を、メモリの割り付けられているアドレスに切り替えます。

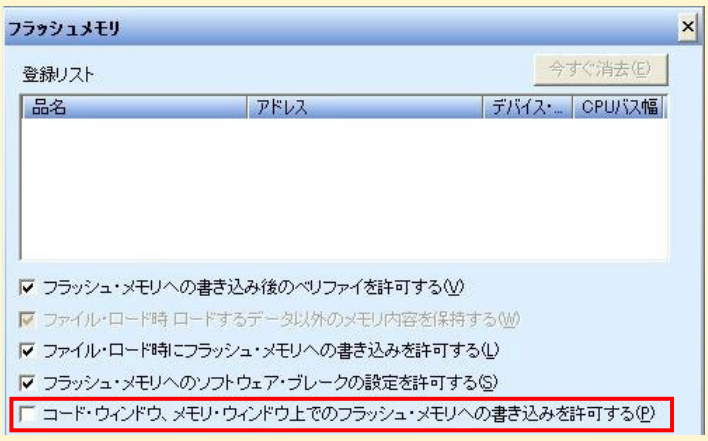


入力したアドレスの表示に切り替わりました。表示したアドレスのメモリが書き換えられるかを確認します。



Note

フラッシュ・メモリの領域をメモリ・ウィンドウ、コード・ウィンドウから書換えを行う場合には、ファイルメニュー[フラッシュメモリ・マップ]から”コード・ウィンドウ、メモリ・ウィンドウ上でのフラッシュ・メモリへの書込みを許可する”にチェックを入れておく必要があります。



Point!

ダイレクト・データ・チェンジ機能

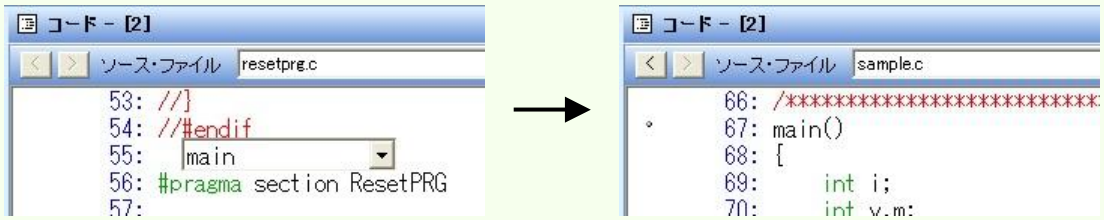
ダイレクト・データ・チェンジ機能とは、入力ダイアログを介することなくウィンドウ上からダイレクトに値や設定の変更、表示画面の切り替えを行う機能です。CSIDE は、各所にこの機能を使用すること出来ますので、これを活用することでスムーズな操作を行うことが出来ます。

◎ダイレクト・データ・チェンジの使用例

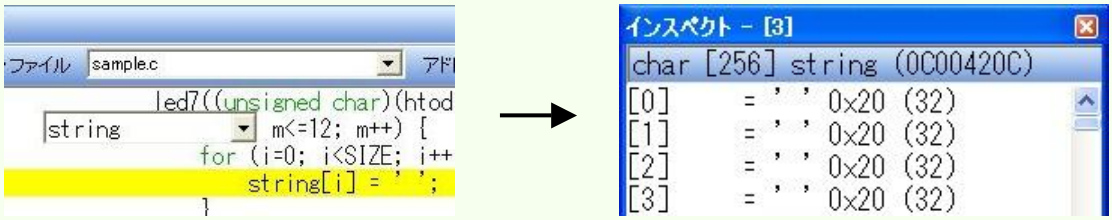
・各ウィンドウでの値を書き換え

・表示アドレスの切り替え

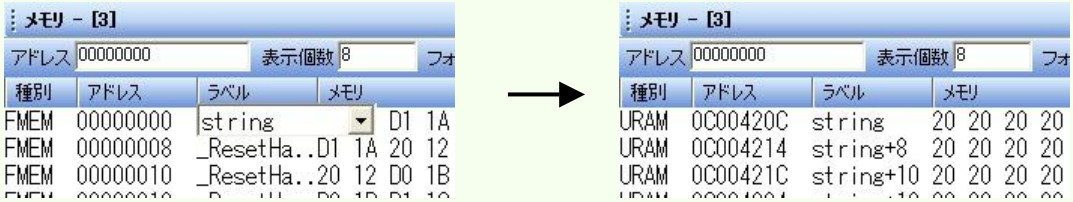
・コード・ウィンドウ上で関数名を入力すると、入力した関数に表示が切り替わります。



・コード・ウィンドウ上で変数名を入力すると、インスペクトウィンドウが開きます。



・メモリ・ウィンドウ上でシンボル名を入力すると、シンボルの割付先が表示されます。



6-4 プログラムの実行

プログラムをダウンロードできましたら、基本動作の確認を行っていきます。
プログラムの動きを見るためのウィンドウが、コード・ウィンドウです。まずはウィンドウの見方から説明します。

表示メニュー[コード]を選択し、コード・ウィンドウを開きます。

ソース表示

注意

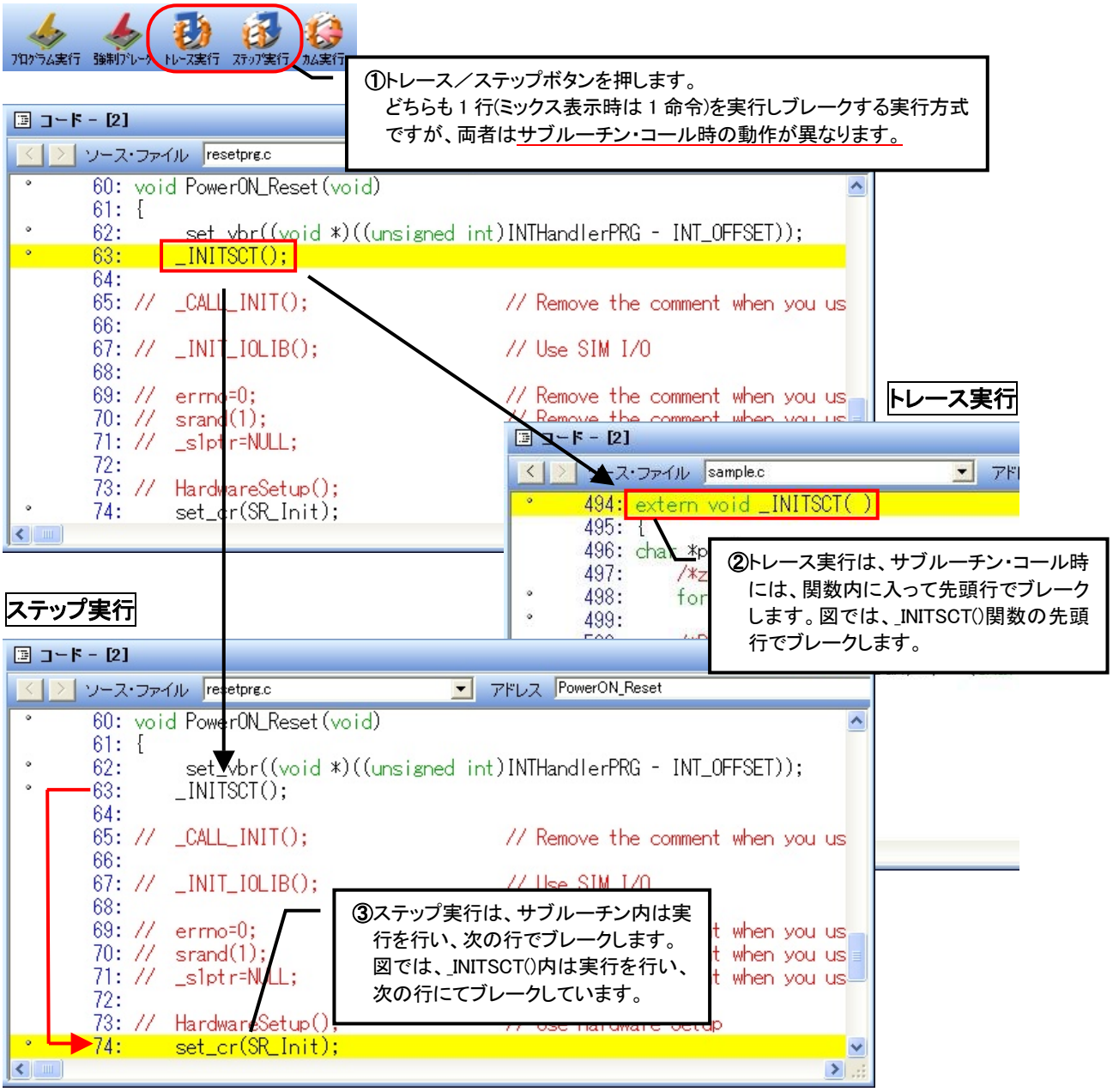
・ソース表示は、デバッグ情報を元に表示を行うため、デバッグ情報が読み込まれていなければ表示することはできません。

・正しくソース表示が行われていても、ターゲットの初期化状態やロード・アドレスの間違い等により、プログラムが正しくダウンロード出来ているとは限りません。ミックス表示にてご確認ください。

Note

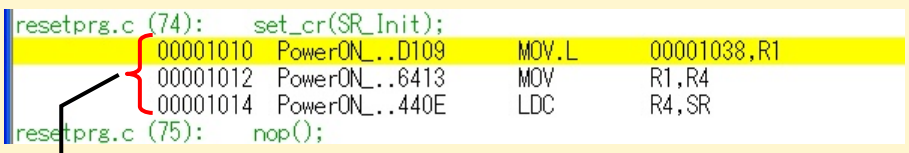
ロード直後、プログラム・カウンタは、リセット番地に設定されています。(一部の CPU では、[初期値の設定]にてプログラム・カウンタの初期値を指定するものもあります。)

CSIDE の実行方法には、大きく分けて、実行、トレース実行、ステップ実行、カム実行の 4 種類があります。
まずは、トレース実行/ステップ実行で動作を確認します。



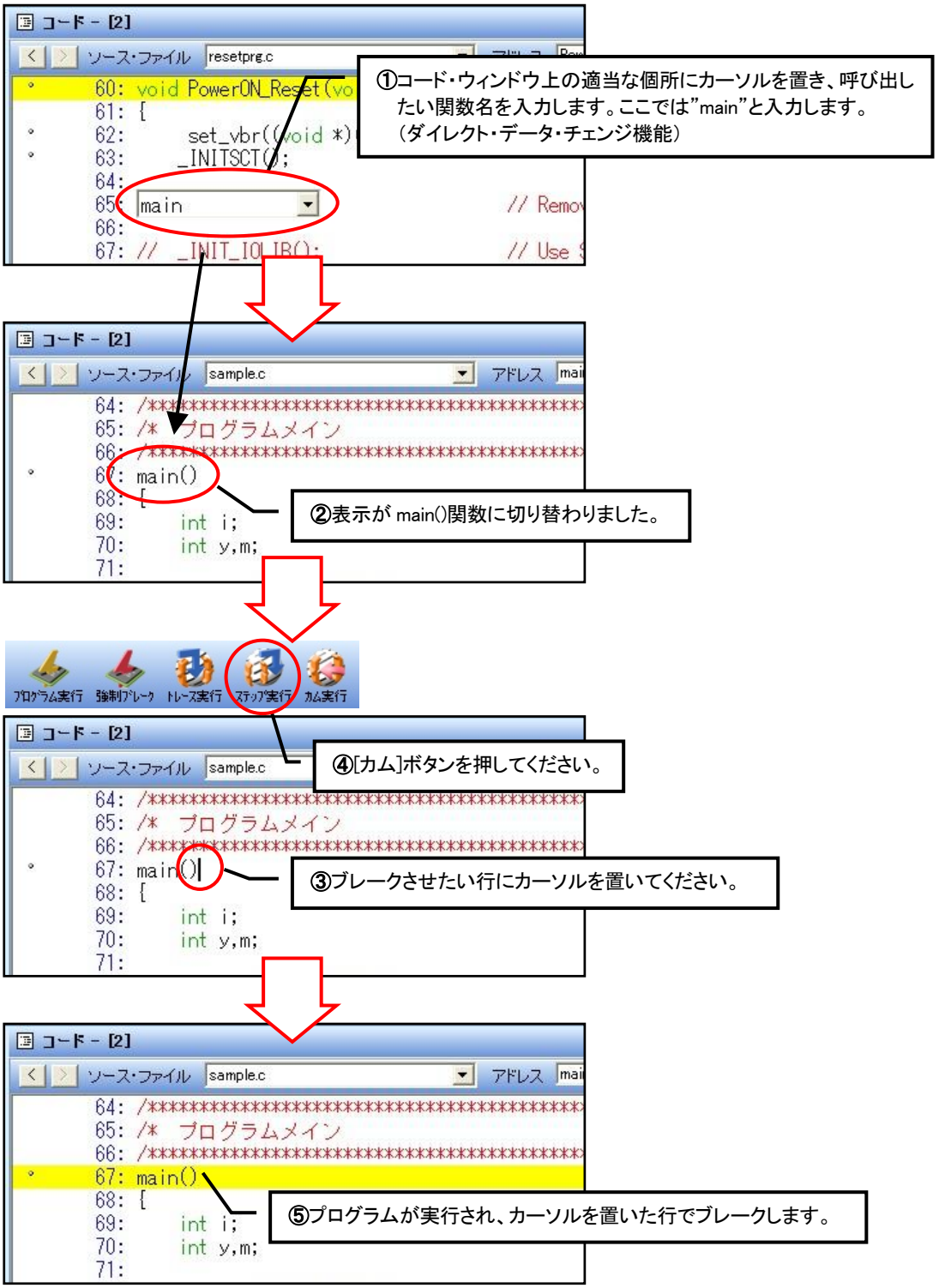
Note

ソース表示でトレース/ステップ実行を行った場合、C ソース 1 行に対応するアセンブラ・コード数命令を実行するということになります。ミックス表示では、アセンブラ・コードを 1 命令単位で実行します。
細かな動作を確認するには、ミックス表示にて確認してください。



図では、C ソース 1 行に対して、アセンブラ・コード 3 命令が対応しています。
ソース表示でこの行を実行すると対応する 3 命令が実行されることになります。

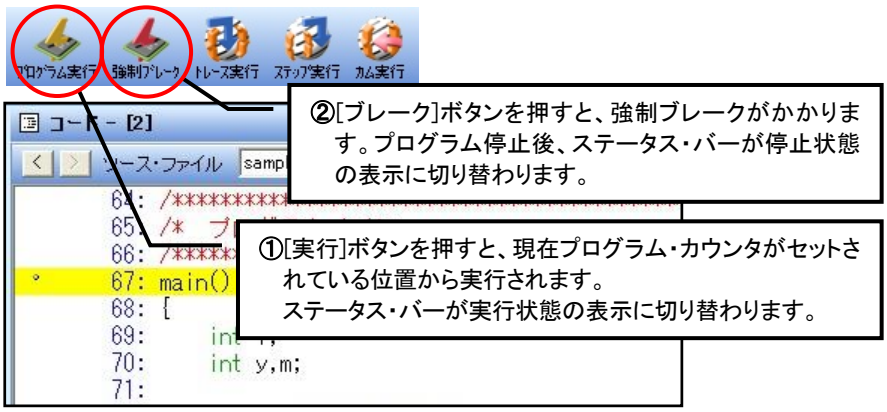
次にカーソル位置まで実行できる”カム実行”を見てみます。
ここでは main 関数の先頭まで実行しブレークさせる例を示します。



Note

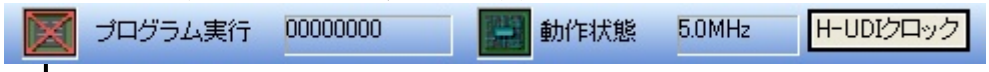
トレース/ステップ/カム実行を行った際に、実行状態となりブレークしない場合には、[ブレーク]ボタンを押してブレークさせてください。ブレークしない場合には、ハードウェアの初期化を行ってください。これは割り込み先やサブルーチン内等でプログラムが暴走してしまうなど、ブレーク位置までプログラム・カウンタが来ないために実行状態に陥っていることが考えられます。

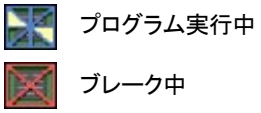
実行を確認してみます。



ステータス・バ

ステータス・バーは、CSIDE 画面左下に表示されます。





Note

空白行にも関わらずデバッグ情報を示すマークが表示される、逆にマークが表示されるはずの行に無いといった場合、何らかの原因でロードしているオブジェクト・ファイルとソース・ファイルが一致していないと考えられます。この様な場合、下記の方法を試してください。

1.マイコンピュータ等でダウンロードしているオブジェクト・ファイルを削除または、他のフォルダに移動する。

2.CSIDE 上でロードを行います。削除または移動しているので、ファイルが見つからないというエラーが表示されるかを確認します。

3.コンパイラで再度ビルドを行い、オブジェクト・ファイルが生成されることを確認する。

4.生成されたオブジェクト・ファイルを再選択し、ロードオブジェクト選択ダイアログのロード環境[全てのデバッグ情報をクリアしてからロードする]にチェックを入れてロードを行ってください。

6-5 ブレーク・ポイントの設定

次にブレーク・ポイントの設定を行います。
ブレーク・ポイントには、ソフトウェア・ブレークと CPU ブレークの 2 種類があります。

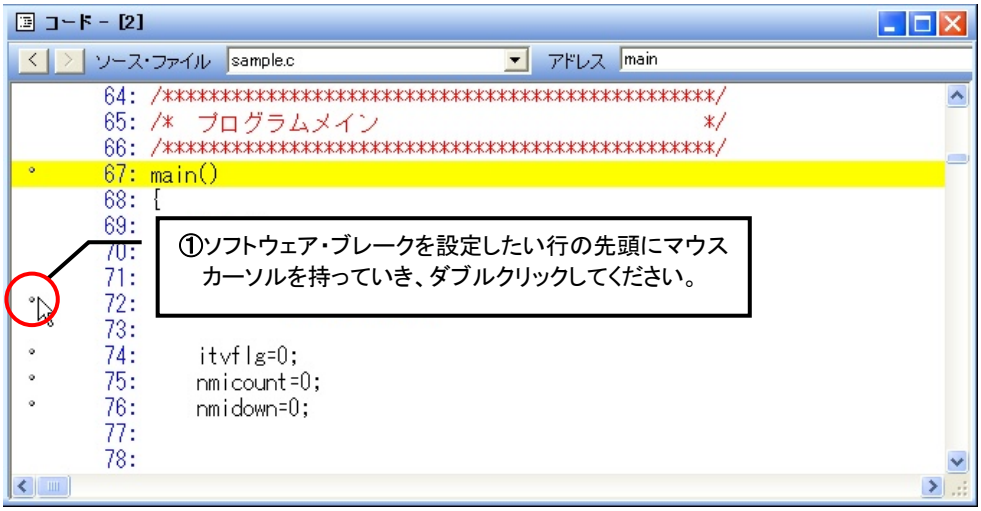
ソフトウェア・ブレーク　　指定したアドレスの命令を、デバッガ専用のコードに書き換えることで、ユーザプログラムを停止させます。このため CSIDE から書換え可能なメモリ領域でなければ使用できません。最大で 256 点まで設定できます。

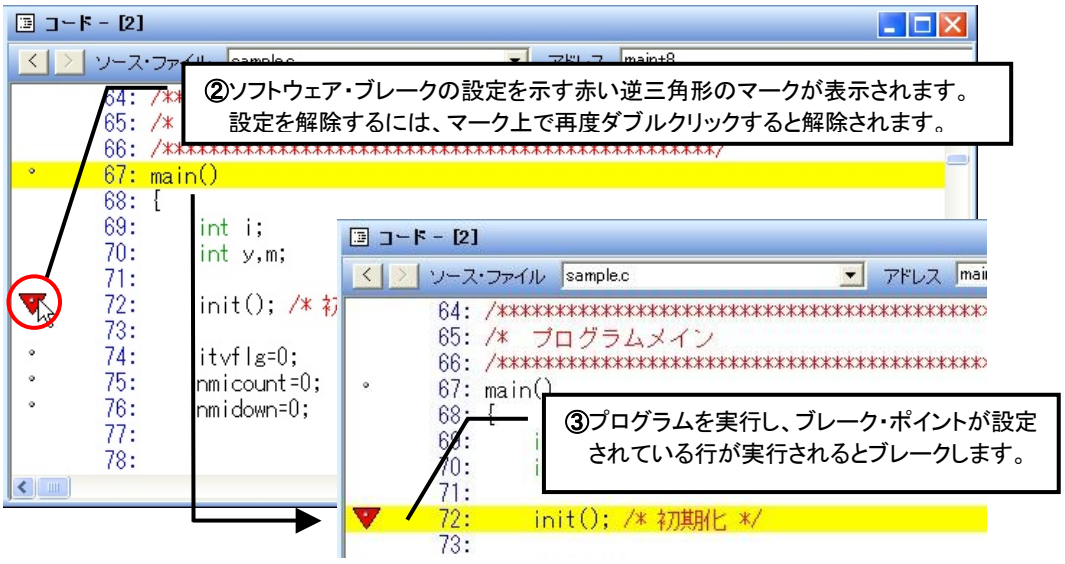
CPU ブレーク　　CPU の持つ機能を使用してブレークさせます。書き換えの出来ない領域にも設定ができ、ブレークの条件を設定できるため、設定次第で様々な条件にてブレークさせることが出来ます。設定できる点数は、CPU により異なります。
本資料では、CPU ブレークと表記しますが、各 CPU により CPU ブレークの呼称や設定項目が異なりますので、詳細は CSIDE のオンライン・マニュアルをご覧ください。

Note

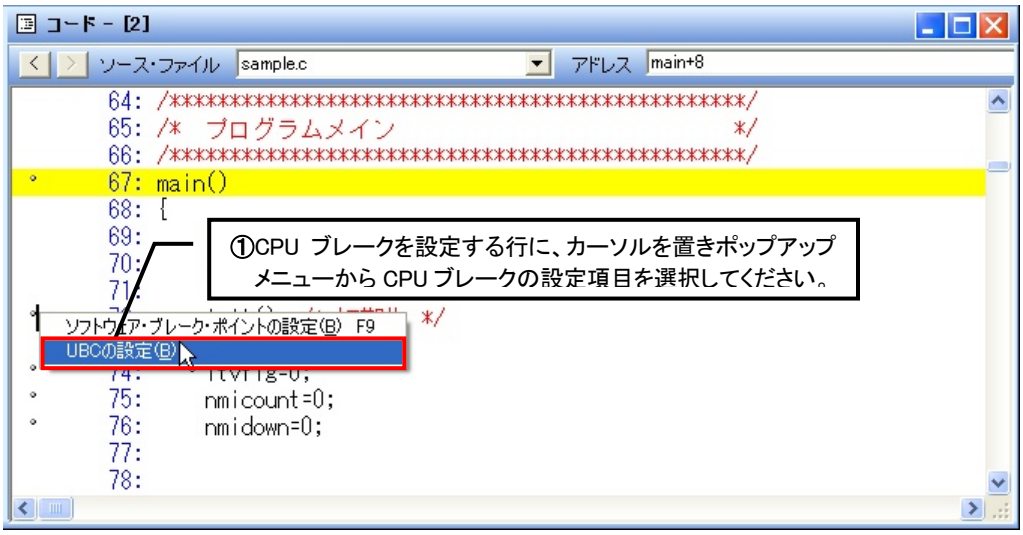
ソース表示では、デバッグ情報の出力されている行にしかブレーク・ポイントを設定することはできません。
ミックス表示であれば、任意のアドレスに設定することが可能です。

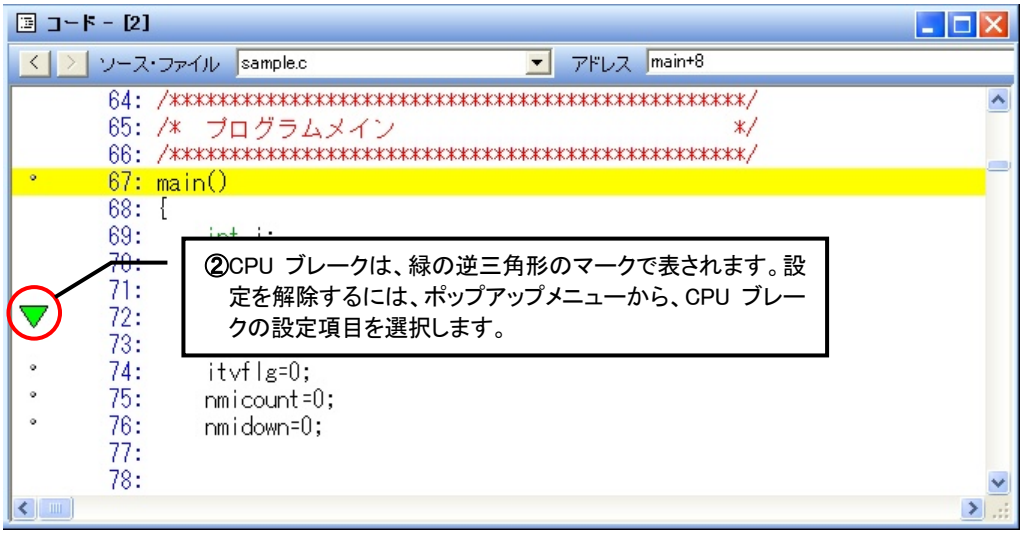
・ソフトウェア・ブレークの設定





・CPU ブレークの設定



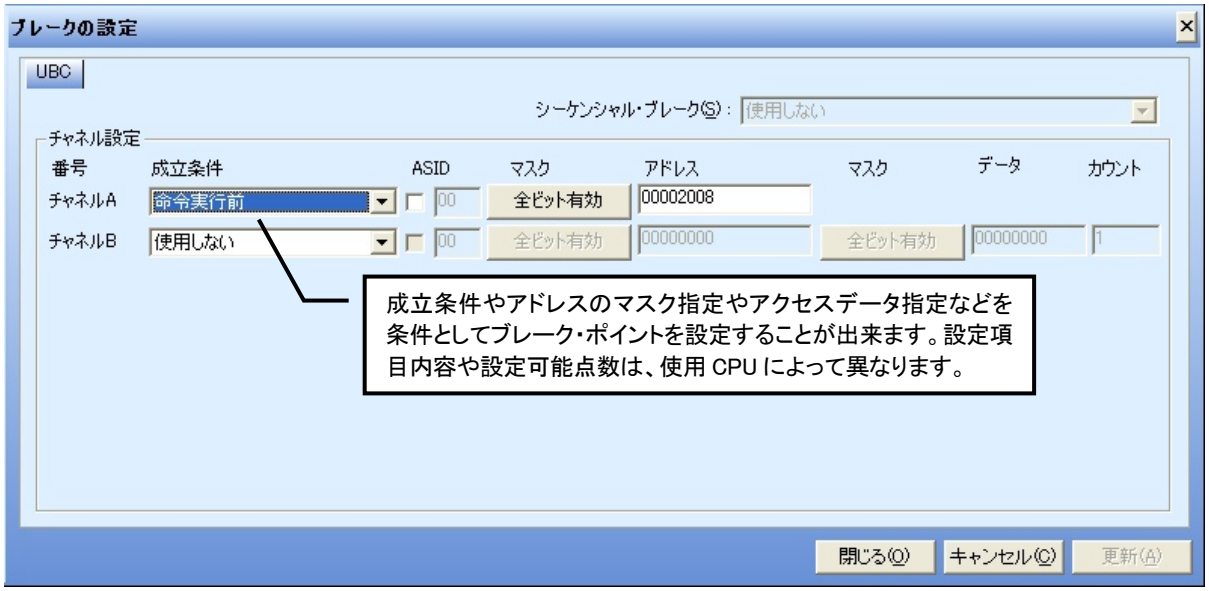


Note

フラッシュ・メモリに対してブレーク・ポイントを設定する場合には、CPU ブレークの使用をお勧めします。ソフトウェア・ブレークでは、実行/ブレークの度にブレークコードへの書き換えが発生するため、処理に時間がかかります。 また、内蔵フラッシュ・メモリの場合は、書き換え回数に制限があるものが多く、書き換えの発生しない CPU ブレークが向いています。

コード・ウィンドウ上からの CPU ブレークの設定は、アドレス一致条件のみの設定となりますが、以下の方法でアドレスやデータ等を条件にして、ブレーク・ポイントを設定することが出来ます。

[ブレークの設定]ダイアログが開きます。



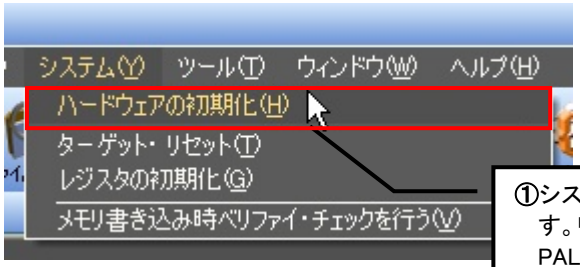
※この画面は、CSIDE for PALMiCE2 SH です。

注意

CPU ブレークは、ご使用の CSIDE により、他のメニューに割り当てられている場合があります。
詳細についてはオンライン・マニュアルを参照してください。

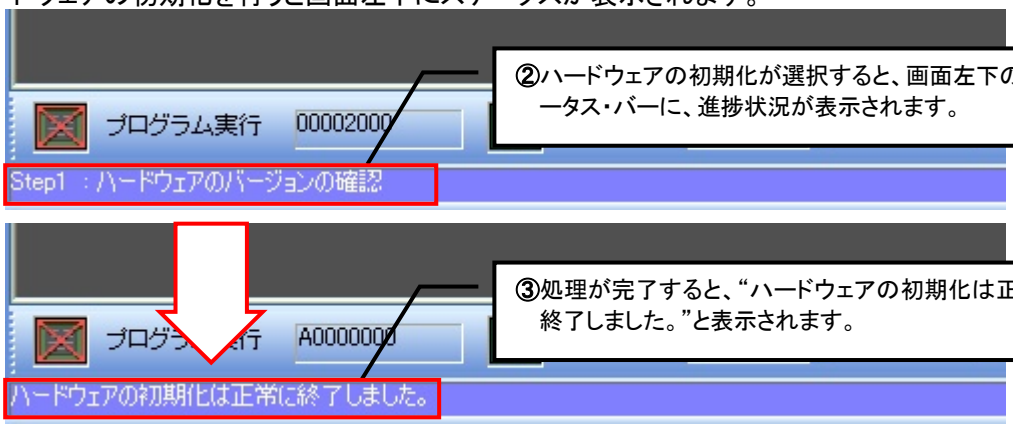
6-6 デバッグをやり直す

操作中に CSIDE が何らかの原因で PALMiCE2、CPU の制御が出来なくなった場合や、プログラムの開始状態に戻してデバッグを行う場合には、[ハードウェアの初期化]を行います。



①システムメニュー[ハードウェアの初期化]を選択します。リセット出力によるターゲット・システムの初期化と PALMiCE2 の初期化を行います。

ハードウェアの初期化を行うと画面左下にステータスが表示されます。



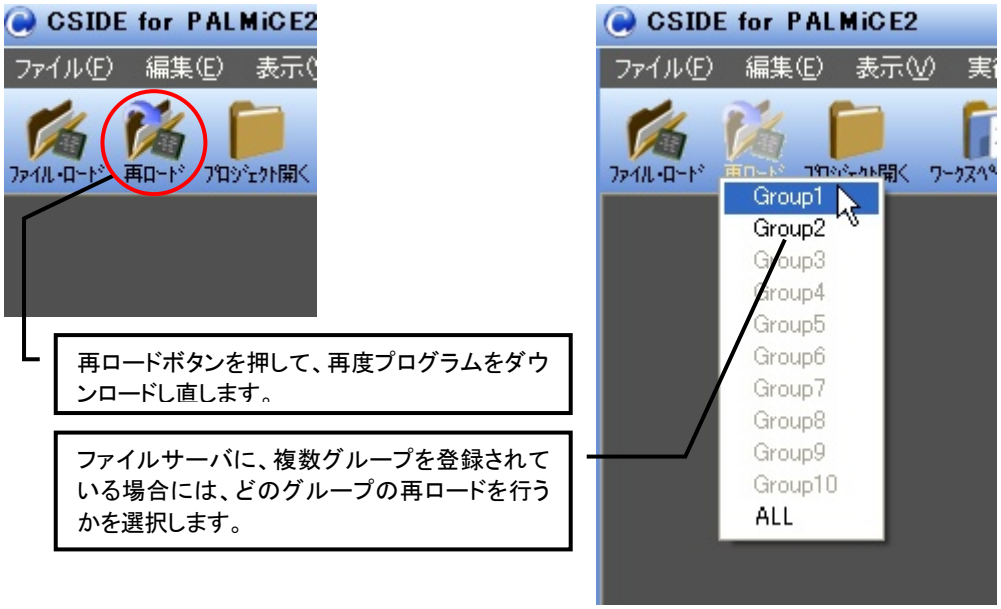
②ハードウェアの初期化が選択すると、画面左下のステータス・バーに、進捗状況が表示されます。

③処理が完了すると、“ハードウェアの初期化は正常に終了しました。”と表示されます。

Note

ターゲット・システムの設定の[更新]ボタンと[ハードウェアの初期化]は、ほぼ同じ処理を行いますが、JTAG(H-UDI)信号の通信チェックの有無が異なります。JTAG 信号の通信確認は、初回起動時のターゲット・システムの更新で正常であることが１度確認できれば以降は必要ありませんので、[ハードウェアの初期化]では省略しています。このため、処理時間が短くなりますので、起動後の初期化には[ハードウェアの初期化]をお使いください。

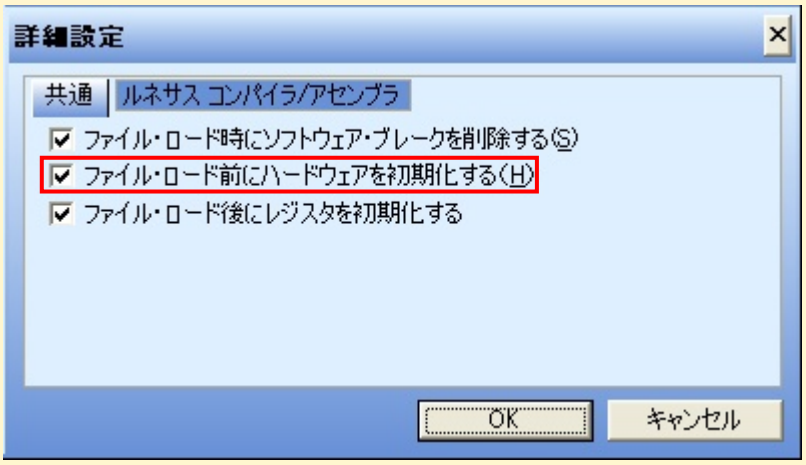
[ハードウェアの初期化]を行うことで、ターゲット・システムの初期化が行われ、レジスタ等が初期状態に戻ります。さらに、ターゲット上のプログラムが書き換わっている場合には、再度プログラムをダウンロードし直します。



Note

・再ロードは、ファイルサーバに登録されている内容通りにロードを行うことができます。
登録内容を変更せずにロードを繰り返し行う場合には、再ロードを使用してください。

・ロード時に自動的にハードウェアの初期化を行うことも可能です。
ファイルサーバ[共通設定]ボタンを押して表示されるダイアログにある[ファイル・ロード前にハードウェアの初期化する]にチェックを入れてください。これにチェックを入れておくことで、ファイル・ロード前にハードウェアの初期化が行われるので、ロードを行うだけでプログラムの開始状態に戻すことが可能です。

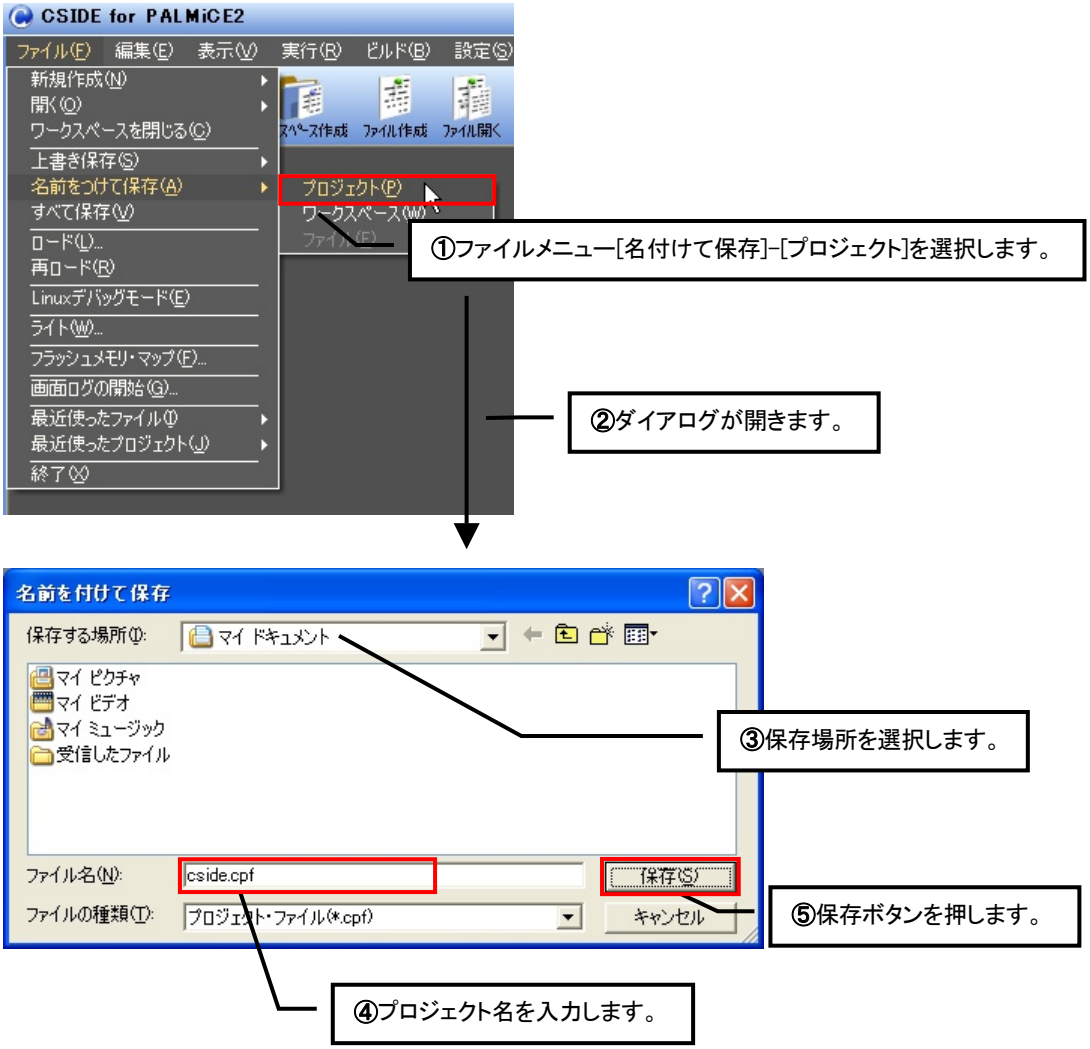


6-7 プロジェクト・ファイルの保存と読み込み

CSIDE の設定を行い、基本動作の確認ができましたら、最後に CSIDE の設定内容をプロジェクトに保存します。
次回以降、プロジェクト・ファイルを読み込むことで、保存された状態に復元することが可能です。

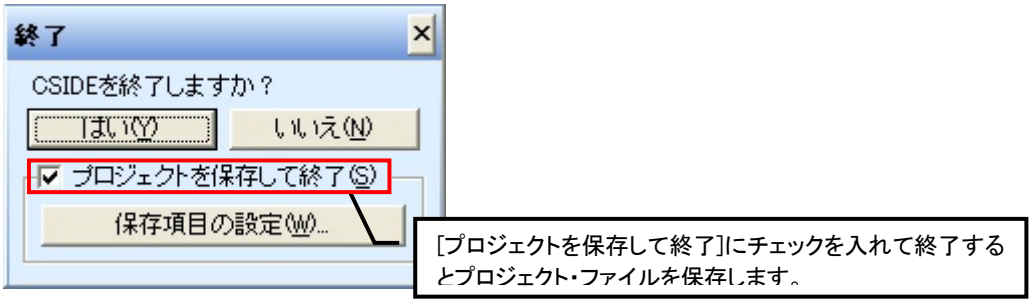
プロジェクト・ファイルの保存

ファイルメニューからプロジェクト・ファイルを保存します。



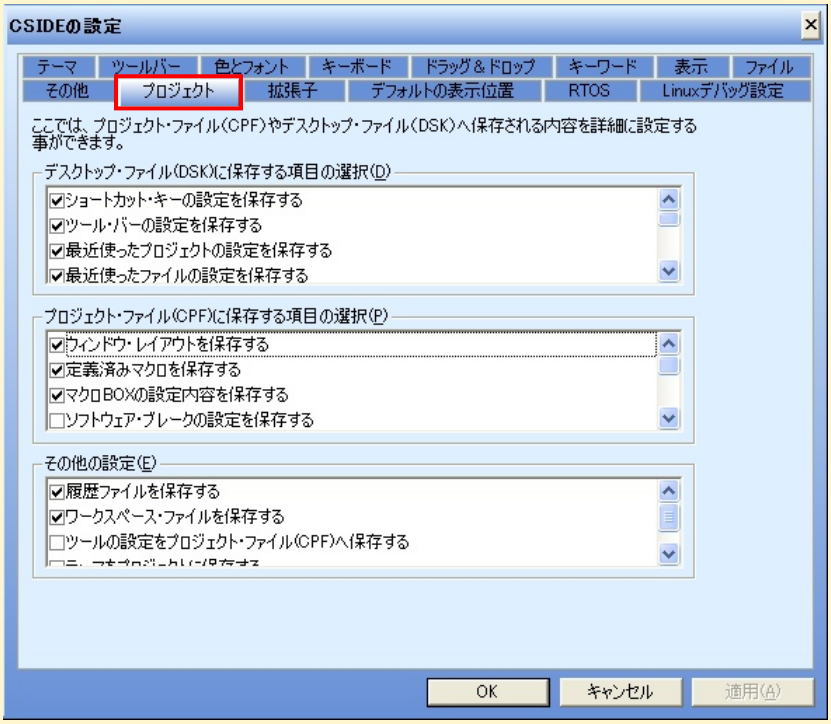
プロジェクト・ファイルは、拡張子.cpfで保存されるファイルです。

その他、CSIDE 終了時の確認ダイアログからプロジェクト・ファイルを保存することが可能です。



Note

プロジェクト・ファイルに保存する項目は、設定メニュー[CSIDE の設定]にて指定することが出来ます。

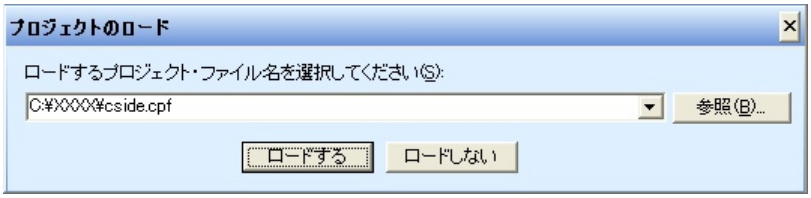


プロジェクト・ファイルを開く

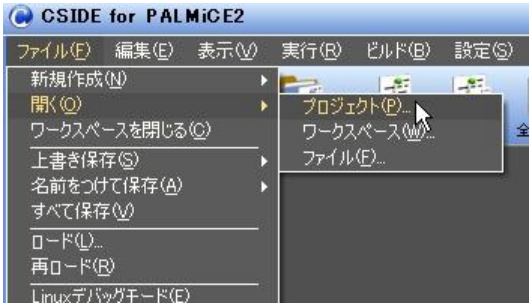
保存されたプロジェクト・ファイルは、以下の方法で読み込むことが出来ます。

プロジェクト・ファイルをダブルクリックして開く

CSIDE 起動時にメニューから開く



CSIDE 起動後、ファイルメニューから開く

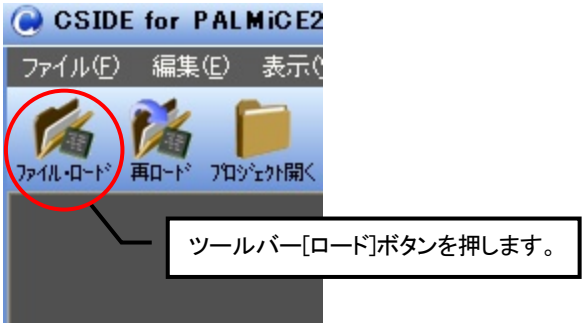


Chapter 7.CSIDE の便利な使い方

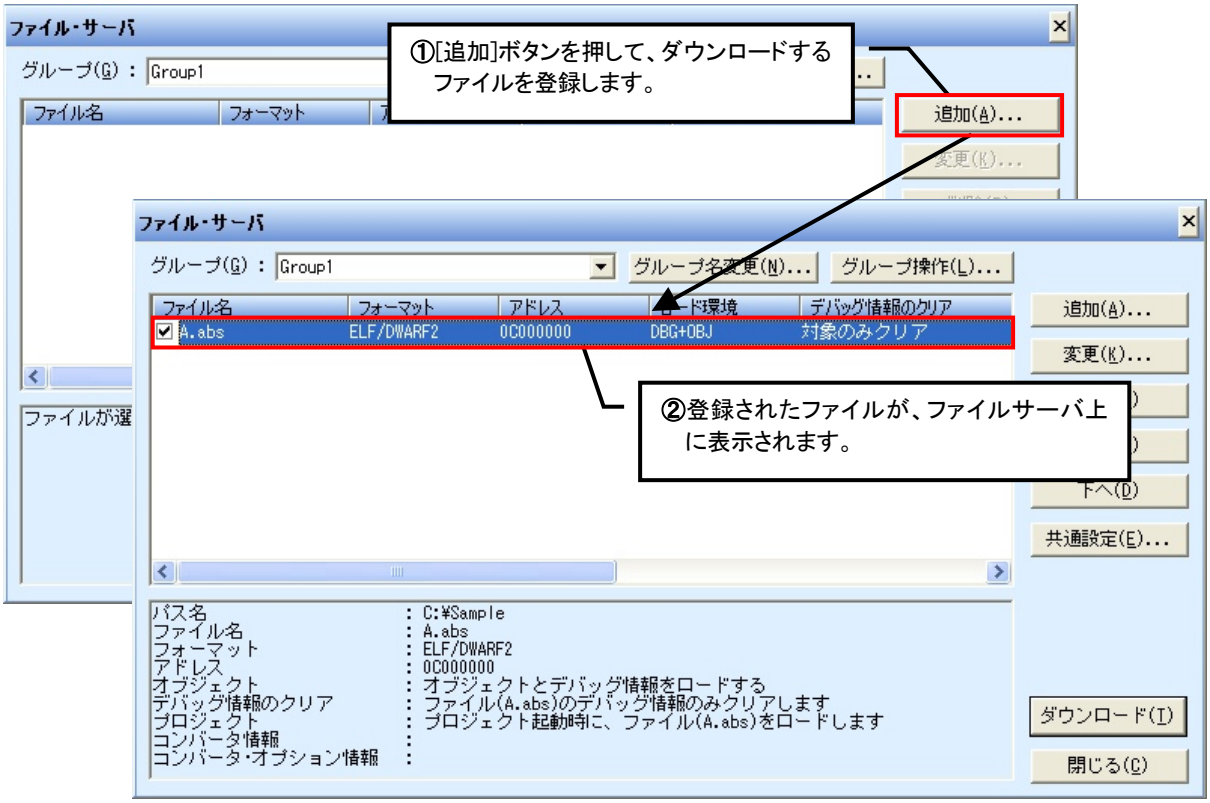
7-1 複数のオブジェクト・ファイルの管理

CSIDE は、ダウンロードするオブジェクト・ファイルを管理するための機能として、ファイル・サーバ機能を備えています。ファイル・サーバは、ロードするオブジェクト・ファイルを複数のグループ単位に別けて、管理することができ、これを活用することで余計なファイル・ロードを省くことができ、効率良くファイル・ロードを行うことができます。

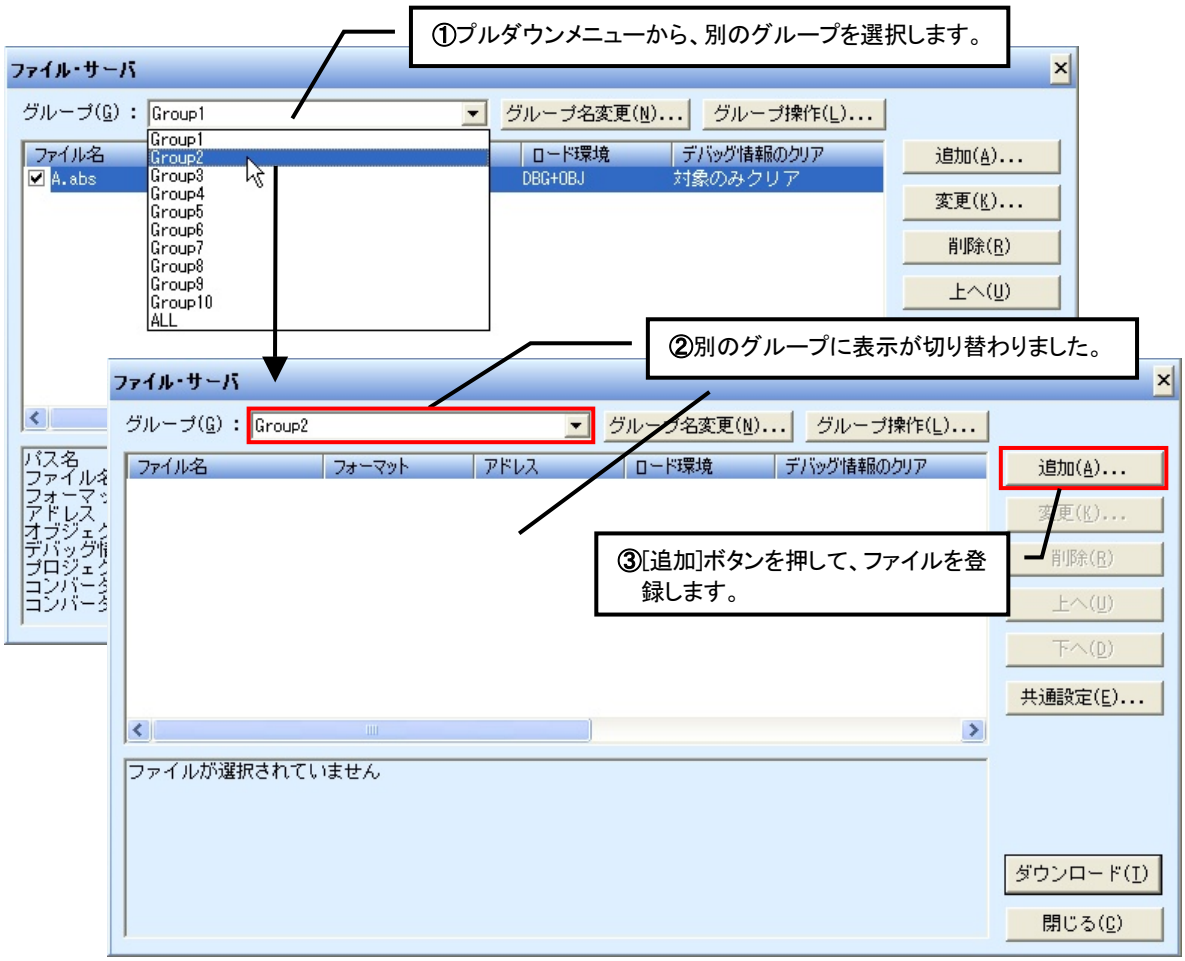
[ロード]ボタンを押してファイル・サーバを開きます。



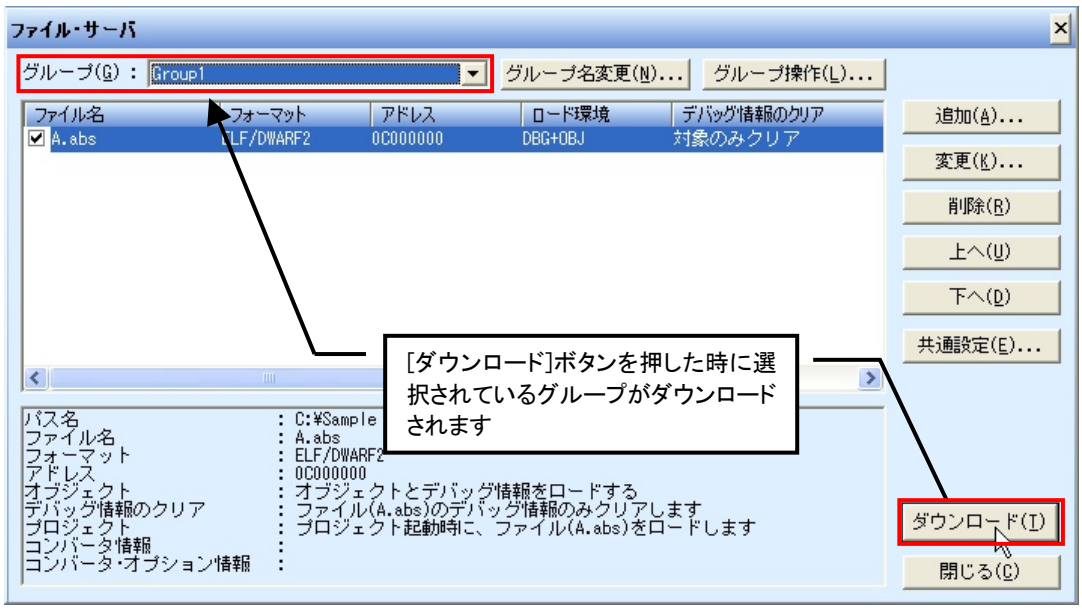
1 つ目のグループにロードするオブジェクト・ファイルを登録します。



次に別のグループに表示を切り替えて、ファイルを登録します。



以上の手順で、複数のグループにダウンロードを行うオブジェクト・ファイルの設定ができます。
複数のグループを登録した状態でダウンロードを行う場合、選択されているアクティブなグループのオブジェクト・ファイルがファイル・ロードされます。



[再ロード]ボタンでは、ファイル・ロードを行うグループを選択するメニューが表示されるので、目的のグループを選択します。

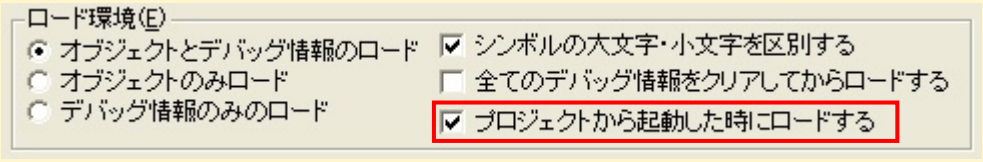
この機能は、複数のオブジェクト・ファイルの中から特定のオブジェクト・ファイルのみを繰り返しファイル・ロードするのに便利な機能です。例えば、OS を使ったプログラムのアプリケーションデバッグでは、カーネル部分は変更しないため、始めに 1 回だけファイル・ロードを行えばよいのですが、デバッグ対象となるアプリケーションはファイル・ロードを繰り返すことになります。この様な場合に、カーネルとアプリケーションを別けて管理することで、アプリケーションのみを簡単にロードすることができ、効率の良くデバッグを行うことができます。

注意

FileLoad コマンドを使用してファイル・ロードを行った場合、全て先頭グループに登録されます。ただし、同一ファイル名が既に登録されている場合には再登録はされません。

Note

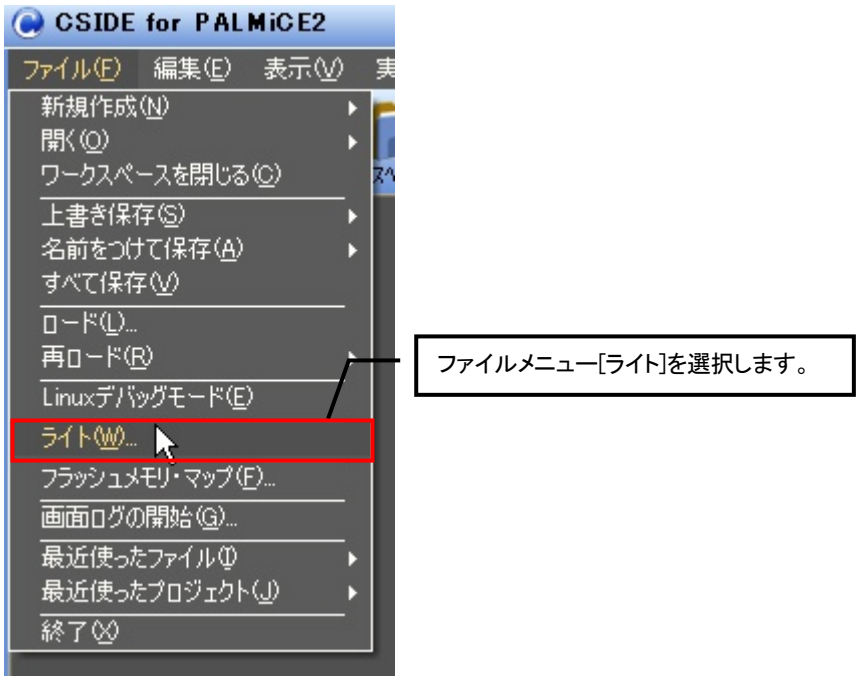
[ファイルのダウンロード]ダイアログにあるロード環境欄にある[プロジェクトから起動した時にロードする]にチェックを入れておくことで、プロジェクト・ファイルからの起動時に、自動的にロードが行われるよう設定できます。1 度だけファイル・ロードを行うオブジェクト・ファイルにこの設定を適用しておくことで、ファイル・ロードの手間を省くことができる便利な設定です。



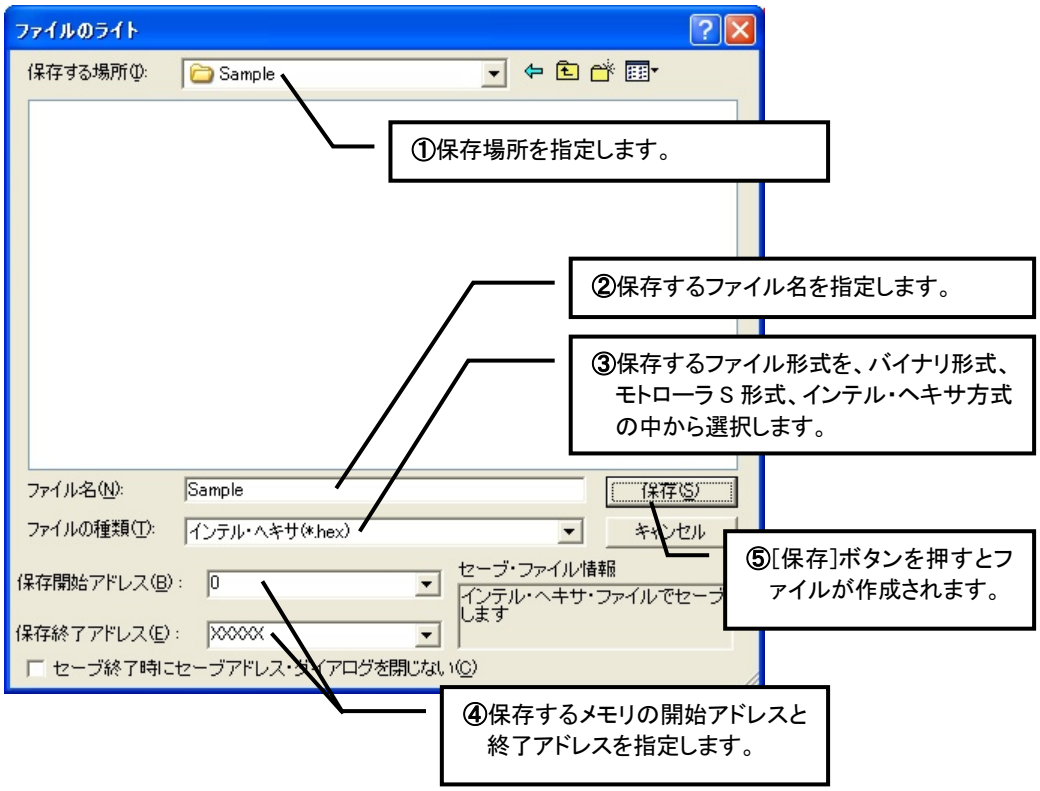
7-2 メモリの内容をファイルへの書き出す

指定した範囲のメモリの内容を、バイナリ形式、モトローラＳ形式、インテル・ヘキサ形式でファイルへ出力することができます。

ファイルへの書き出しは、ファイルメニュー[ライト]から行います。



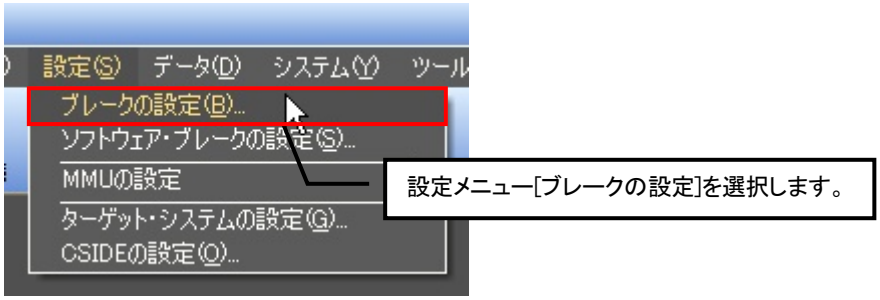
[ファイルのライト]ダイアログが開かれるので、各個所を設定し保存します。



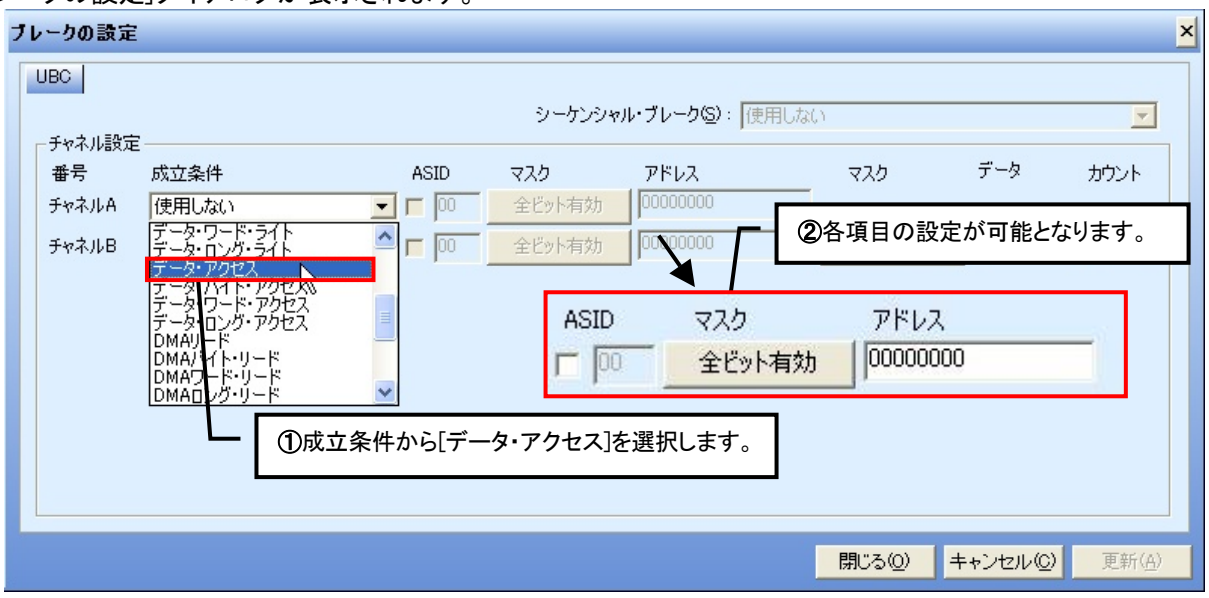
7-3 一定範囲のアドレスへのアクセスでブレークさせるには

CPU ブレークには、ブレーク・ポイントを設定したアドレスでブレークさせる以外に、様々な条件を指定して条件に一致した場合にブレークさせるということができます。ここでは、一定範囲内のアドレスに対してデータアクセスがあった場合にブレークさせる方法として、1000 番地から1FFF 番地の範囲に対してアクセスがあった場合にブレークする設定例を紹介します。

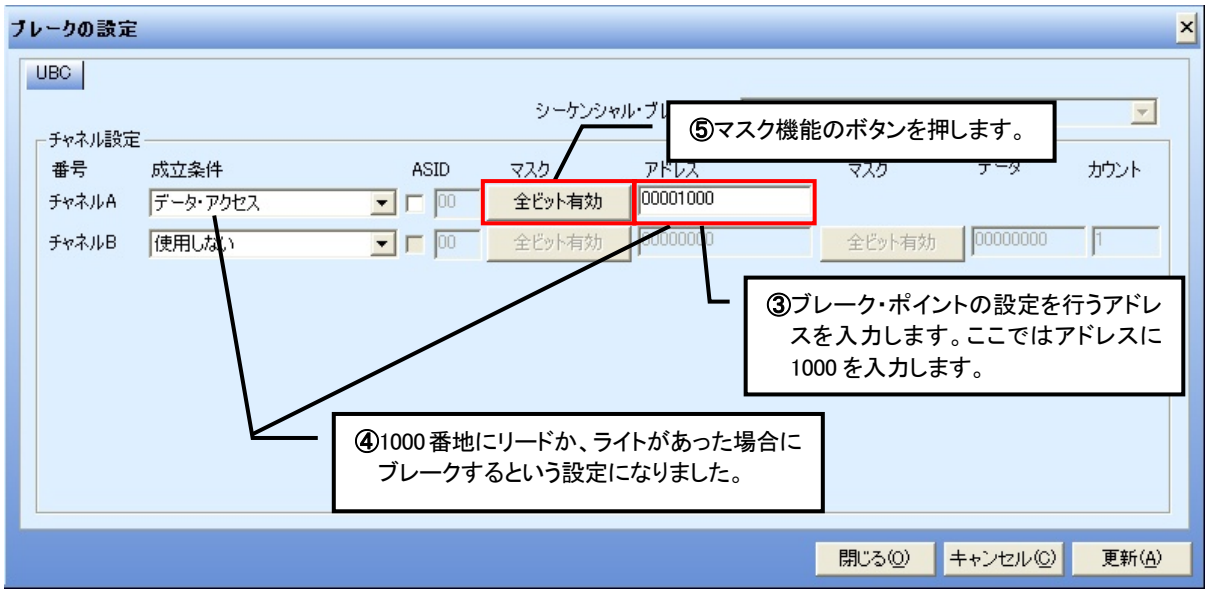
設定メニュー[ブレークの設定]を選択します。



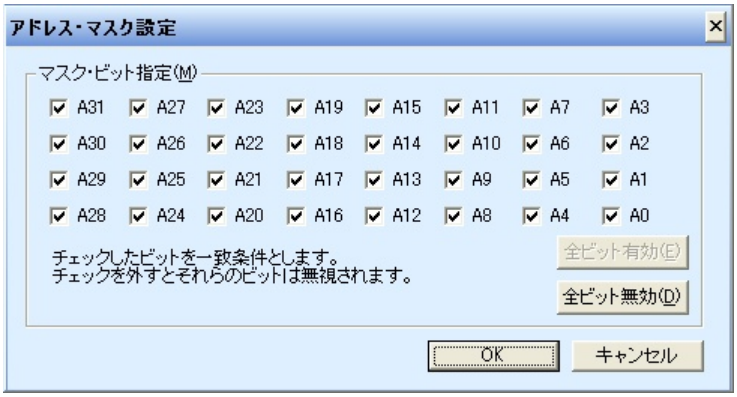
[ブレークの設定]ダイアログが表示されます。



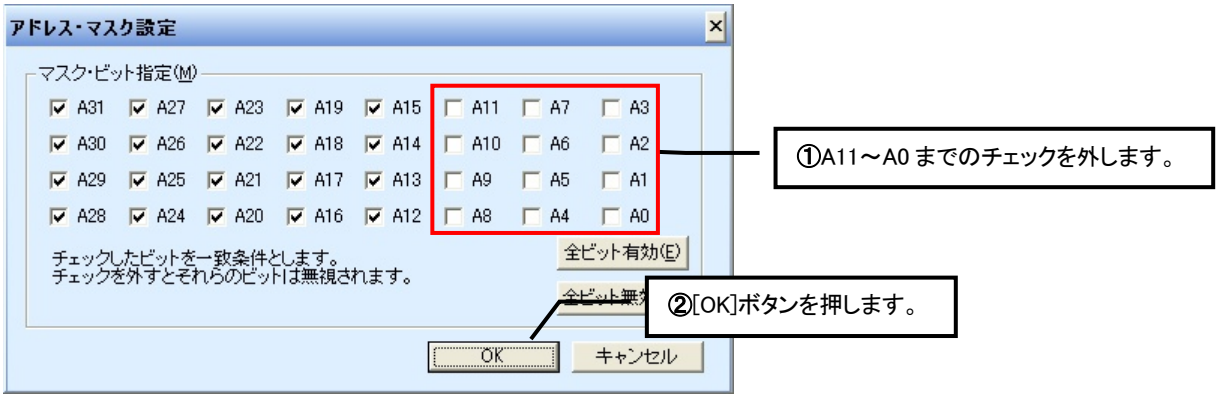
※チャネル数や設定内容、設定画面は、お使いの CPU によって異なります。



[アドレス・マスク設定]が表示されます。

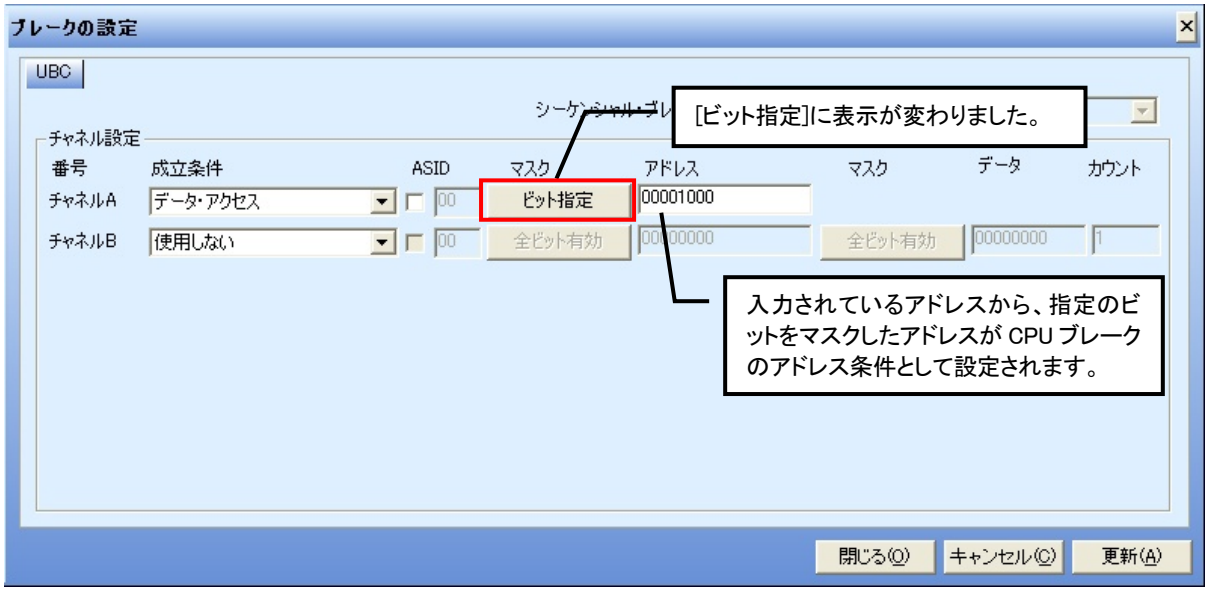


このダイアログは、ブレーク・ポイントを設定するアドレスのマスクを設定します。
ここでは、チェックが入っているビットをアドレスの条件に含み、チェックの入っていないビットがマスクされ条件に含まないというように設定されます。



ここでは 1000 番地から 1FFF 番地の範囲にアクセスがあった場合にブレークするという設定を行いますので、この場合、下位 12 ビットの状態を問わずに上位が 00001XXX 番地にアクセスがあった場合にブレークするということになります。そこで、A11 以下のアドレスのチェックを外してアドレスをマスクします。

アドレス・マスクの設定を行うと、マスク欄のボタンの表示が変わります。

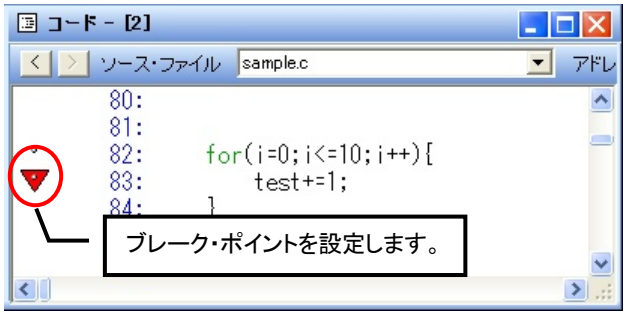


以上の設定で、1000 番地から 1FFF 番地の範囲にアクセスがあった場合にブレークさせるというブレーク・ポイントの設定ができました。これ以外にも、データ条件の指定や成立条件を色々と変更することで様々な条件のブレーク・ポイントを設定することができます。

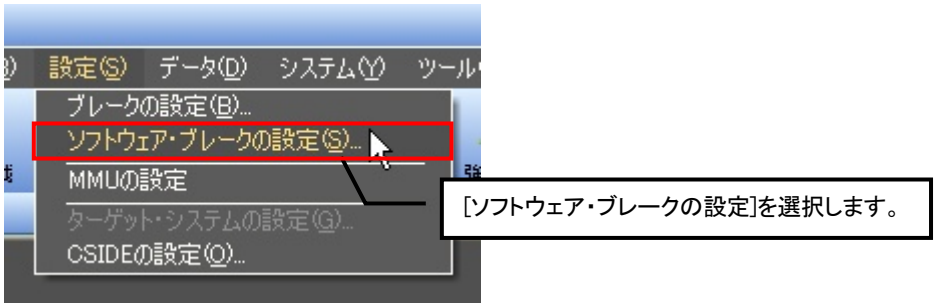
7-4 変数が特定値になった時にブレークさせるには

ソフトウェア・ブレークには、ブレーク時にコマンドを実行する機能があり、この機能を利用することで、変数が特定の値になった時にブレークさせることができます。
ここでは変数 test の値が 5 となった時にブレークする設定を行ってみます。

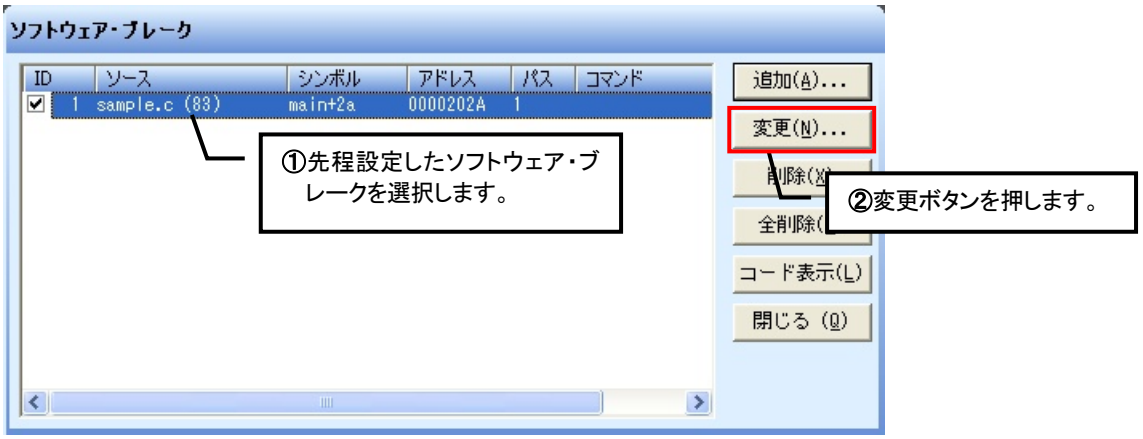
ブレークさせたい行にソフトウェア・ブレークを設定します。



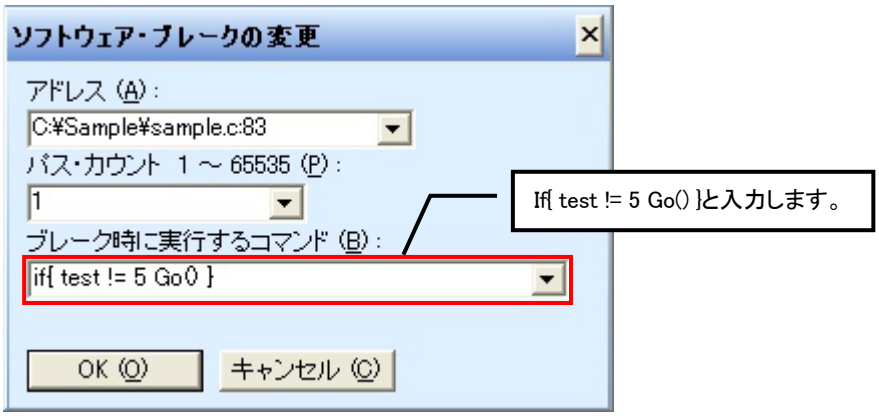
設定メニュー[ソフトウェア・ブレークの設定]を選択します。



[ソフトウェア・ブレーク]ダイアログが開きます。

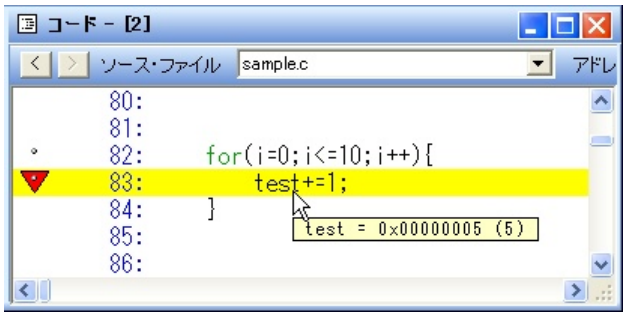


ソフトウェア・ブレークの変更が開きます。



[ブレーク時に実行するコマンド]には、設定されたソフトウェア・ブレークでブレークした際に自動的に実行するコマンドを指定します。ここでは、「変数 test の値が 5 以外ならば再実行する」という意味のコマンドを記述しています。つまり、ブレークした際に、変数 test の値を判定し、5 以外の時には再実行し、5 であればブレークするということになります。

この設定で、プログラムを実行すると変数が特定の値になるまで実行/ブレークを繰り返して特定値になった時にブレークします。



注意

この方法は、ソフトウェア・ブレークでのブレークと実行を繰り返すために、プログラムのリアルタイム性はありません。

Note

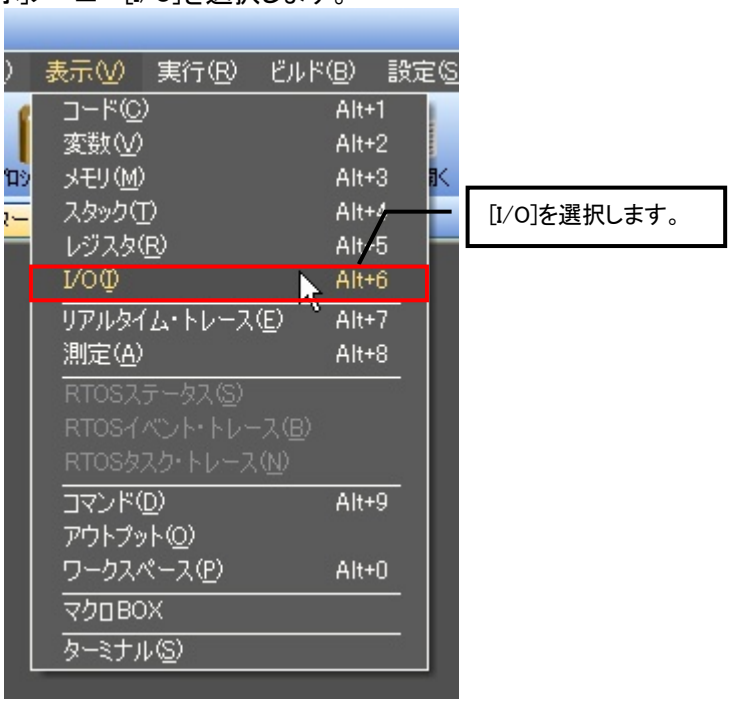
この機能を利用すれば、変数だけでなくレジスタやメモリの値を条件として、様々な条件を設定することができます。

if { !(test == 5 || test == 3) Go() } ····· test が 5 または 3 のときブレーク
if { !(test == 5 && y == 1) Go() } ····· test が 5 でかつ y が 1 のときブレーク
if { *((char *)0x1000) != 0x1} Go() } ····· 1000 番地の内容が 1 のときブレーク
if { _r1 != 0x1234} Go() } ····· レジスタ R1 が 0x1234 のときブレーク

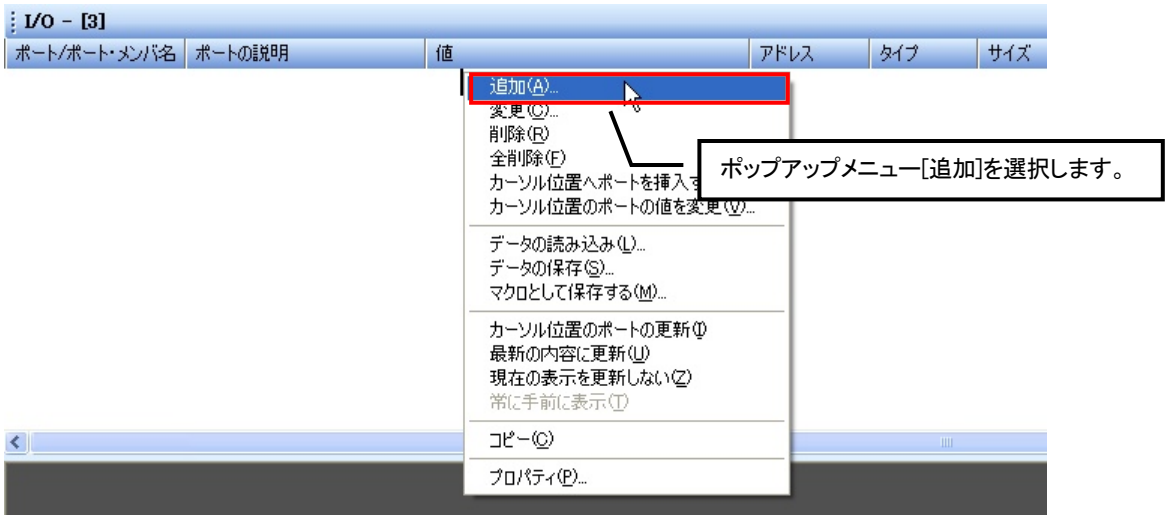
7-5 ポートをビット単位で表示させるには

I/O ウィンドウは、I/O ポートを参照、変更するためのウィンドウです。
I/O ウィンドウに I/O ポートを登録することで、I/O の状態を一覧表示にて確認することが出来ます。
登録された内容は、ブレークごとに内容が更新されますので、一括して I/O ポートの変化を見たい時に便利なウィンドウです。

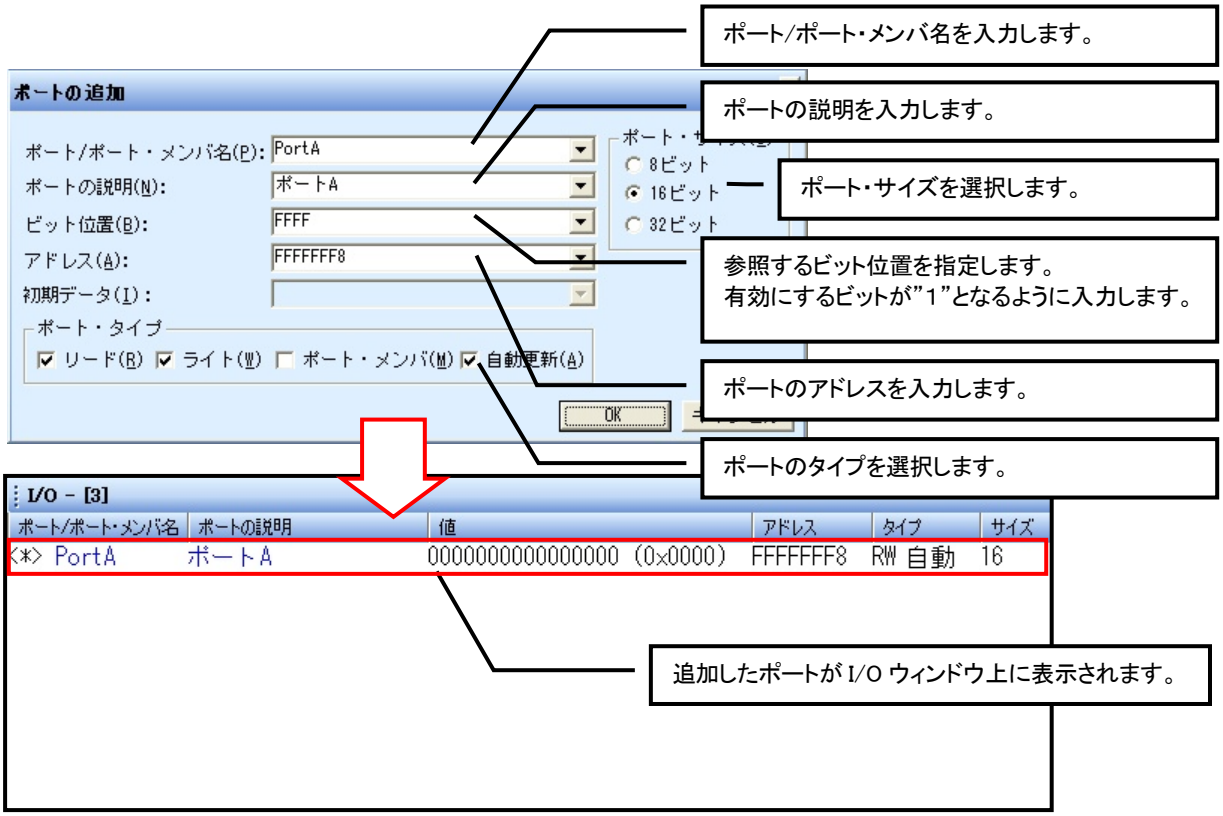
[表示]メニュー[I/O]を選択します。



何も登録されていない[I/O]ウィンドウが表示されます。

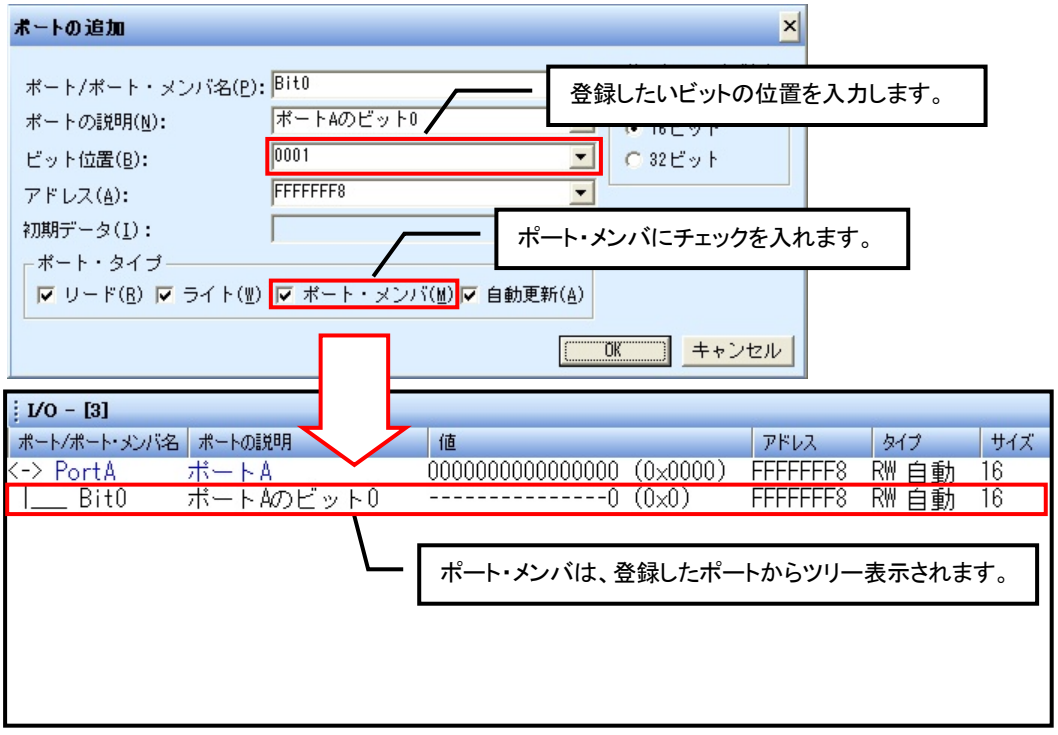


ポートの追加ダイアログが表示されます。ここに追加するポートのデータを登録します。

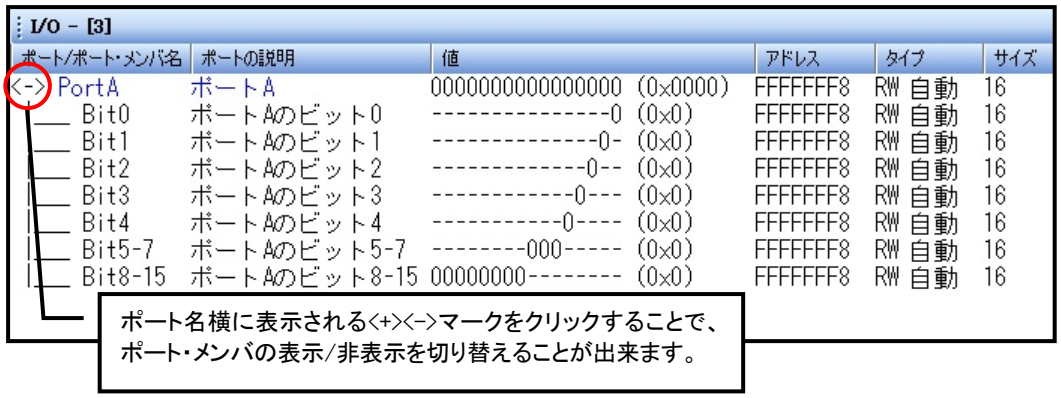


※ビット位置の基数は、値の先頭に2進数="0n",8進数="0o"、10進数="0i"、16進数="0x"を指定して下さい。
なお、基数を指定しない場合は、16進数として判断されます。

さらに、I/O ウィンドウでは、I/O ポートの特定ビットだけに注目して、ポート・メンバとして登録することも出来ます。
先程と同様に、I/O ウィンドウから[追加]を選択します。



ポート・メンバの登録を繰り返し、以下のように登録するとビット単位で見やすく登録することができます。



Note

・I/O ウィンドウからの値の書換えをダイレクトに行うことも出来ます。(ライト可能ポートのみ)
値フィールドに、値を入力すると書き換えることが可能です。

・I/O ウィンドウに登録を行った内容を、I/O 定義ファイルとして保存/読込みを行うこともできます。

・ルネサステクノロジ社のホームページに公開されている I/O レジスタ定義ファイルを、当社の I/O 定義ファイル形式に変換を行うツール「I/O 定義ファイル・コンバータ」を当社ホームページにご用意しています。ルネサステクノロジ製 CPU をお使いの場合には、ご利用ください。

株式会社コンピューテックス
テクニカルセンタ

〒605-0846
京都市東山区五条橋東四丁目 432-13 對嵐坊ビル 4F
TEL.075(551)0373 FAX.075(551)2585
WebSite : http://www.computex.co.jp/
E-Mail : support@computex.co.jp

CSIDE チュートリアルブック

2007 年 12 月 第 4 版