

## XsSight ユーザーズマニュアル補足（Dual Core 編）

1.1版 2023年10月02日

## 1. 概要

本書では、XsSightにてDual Core CPU（SH7205、SH7265、SH7786）のデバッグを行う上で「XsSight ユーザーズマニュアル」に記載されていないDual Core用の機能の補足と、弊社製SH7205ボードAP-SH2AD-0Aをターゲットとしたデバッグのチュートリアルを紹介いたします。

なお、本書に記載されていないXsSightの操作、機能につきましては、XsSightに付属の「XsSight ユーザーズマニュアル」をご覧ください。

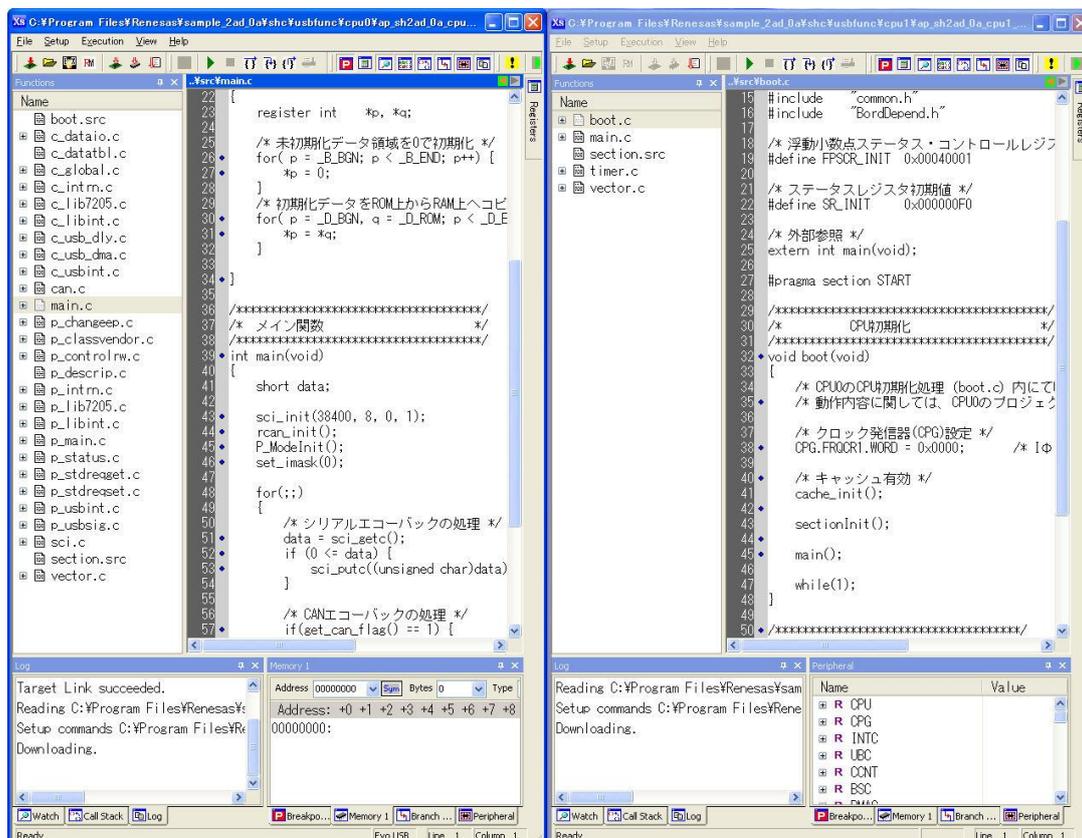
## 2. 画面構成

Dual Core CPUには、CPU0とCPU1の二つのCPUコアがあります。

XsSightでは、それぞれのCPUコア毎にデバッグが行えるよう、XsSightのSystem SetupでDual Core CPUを選択した場合、下記のように、それぞれのCPUコア用に二つのメインウィンドウが開きます。

各メインウィンドウのタイトルバーには、コア名が表示され、どちらのコア用か認識することができます。

また、CPU1用のメインウィンドウは、閉じることができ、Master Control ツールバーを使用することで、再度、開くことができます。



### 3. 機能

本章では、Dual Core CPU をデバッグするための機能について説明いたします。

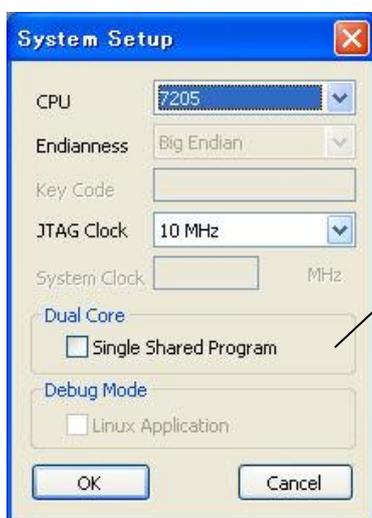
#### (1) System Setup 設定ダイアログ

System Setup ダイアログの CPU に Dual Core CPU が選択されると、Single Shared Program チェックボタンが、有効になります。

このチェックボタンは、CPU0 と CPU1 のプログラムを一つのプロジェクトで作成されているか、CPU コア毎に別々のプロジェクトとして作成されているかを選択します。

一つのプロジェクトの場合には、チェックボタンをオンに、別々のプロジェクトの場合には、オフにしてください。

なお、このチェックボタンが、オンの場合には、CPU1 用ウィンドウのメニューの一部（Download Program や Read Program 等）が無効表示となります。



System Setup 設定ダイアログ

Single Shared Program  
プロジェクトが一つの場合は ON  
CPU0 と CPU1 別々の場合は OFF

#### (2) プログラムのダウンロード

プログラムのロードは、(1) の Single Shared Program チェックボタンがオンの場合には、CPU0 用ウィンドウでダウンロードを行ってください。

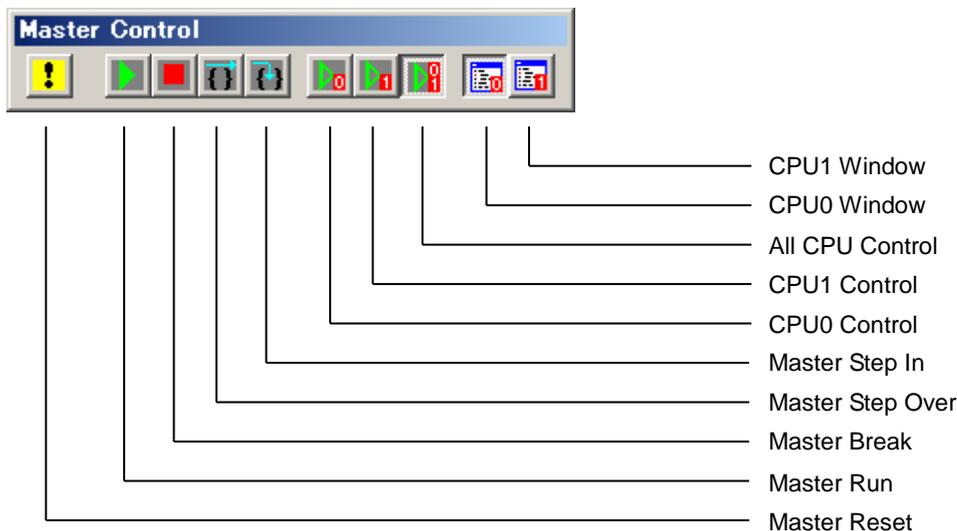
オフの場合には、CPU0 用のプロジェクトを CPU0 用のウィンドウで、CPU1 用のプロジェクトを CPU1 用のウィンドウでダウンロードしてください。

### (3) Master Control ツールバー

Master Control ツールバーは、選択された CPU コアに対して実行、停止、ステップなどを行うためのツールバーです。

CPU コアの選択は、CPU0 のみ、CPU1 のみ、CPU0 と CPU1 両方の3種類から選べます。

また、Master Control ツールバーにより、CPU0 用のウィンドウと CPU1 用のウィンドウを切り替えることができます。



Master Control ツールバー

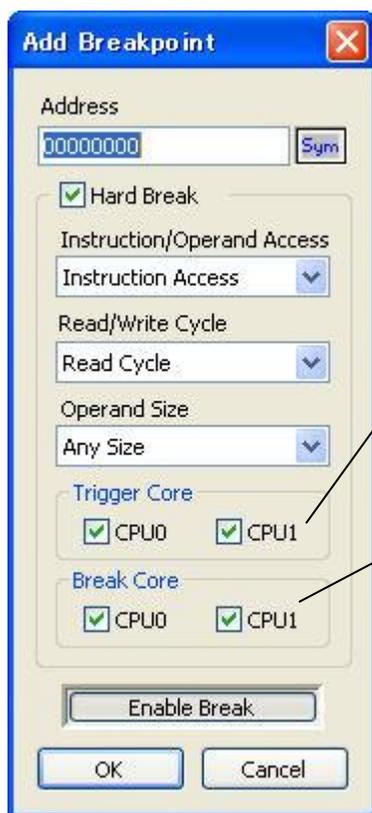
ボタン項目	機能概要
Master Reset	 選択された CPU コアへ内部リセットを発行します。 リセット後は、リセットアドレスで停止します。
Master Run	 選択された CPU コアを実行します。
Master Break	 選択された CPU コアを停止します。
Master Step Over	 選択された CPU コアをステップ実行します。 「Source」ウィンドウに表示されている 1 ステートメントを実行し、停止します。 サブルーチン呼出の場合、サブルーチン実行後停止します。
Master Step In	 選択された CPU コアをステップ実行します。 「Source」ウィンドウに表示されている 1 ステートメントを実行し、停止します。 サブルーチン呼出の場合、サブルーチン内の最初のステートメント手前で停止します。
CPU0 Control	 実行、停止等を行う CPU コアとして CPU0 を選択します。
CPU1 Control	 実行、停止等を行う CPU コアとして CPU1 を選択します。
All CPU Control	 実行、停止等を行う CPU コアとして CPU0 と CPU1 を選択します。
CPU0 Window	 CPU0 用のウィンドウをアクティブにします。
CPU1 Window	 CPU1 用のウィンドウをアクティブにします。 また、CPU 1 用のウィンドウが閉じているときにウィンドウを開くことができます。

## (4) ハードウェア・ブレーク

## ① SH-2A Dual (SH7205、SH7265 の場合)

ハードウェア・ブレークを使用する場合、どちらの CPU コアをトリガ（ブレーク条件）の対象とするか、ブレーク発生時にどちらの CPU コアを停止させるか選択することができます。

この設定によって、ブレーク時に、CPU1 の動作は停止させず、CPU0 を停止させることや、CPU1 がブレークの条件となる処理をしたときにのみ、両方の CPU コアを停止させるといったことが可能となります。



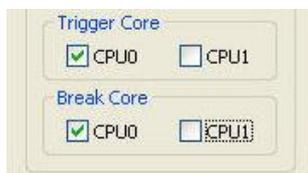
## Trigger Core

トリガの対象となる CPU を選択します。  
ここで選択した CPU がブレークの条件を満たしたときにブレークが発生します。

## Break Core

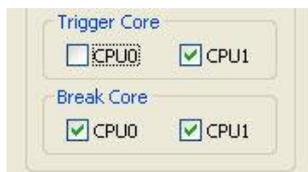
ブレーク時に停止させる CPU を選択します。  
ブレークが発生したときに、ここで選択した CPU を停止させます。

## 設定例 1



トリガの判定は、CPU0 のみで行い、ブレーク発生時には、CPU0 のみ停止します。

## 設定例 2

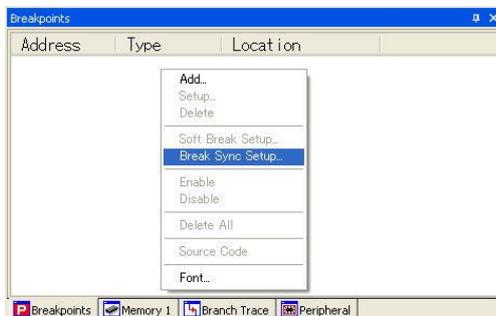


トリガの判定は、CPU1 のみで行い、ブレーク発生時には、両方の CPU を停止します。

② SH-4A Multi (SH7786 の場合)

ハードウェア・ブレイクを使用する場合、ブレイクを同期させてもう一方の CPU を停止させるかの選択することができます。

この設定によって、ブレイク時に、CPU1 の動作は停止させず、CPU0 を停止させることや、CPU1 がブレイクの条件となる処理をしたときのみ、両方の CPU コアを停止させるといったことが可能となります。

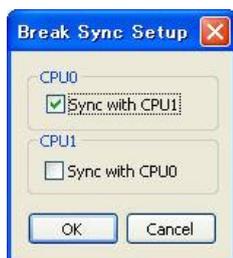


Breakpoints ダイアログのポップアップメニュー

**CPU0**  
CPU1 がブレイク時に、CPU1 と同期して CPU0 をブレイクさせるか選択します。  
チェックボタンをオンにした場合、CPU0 は CPU1 のブレイクに同期してブレイクします。

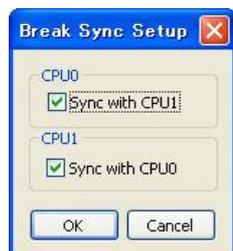
**CPU1**  
CPU0 がブレイク時に、CPU0 と同期して CPU1 をブレイクさせるか選択します。  
チェックボタンをオンにした場合、CPU1 は CPU0 のブレイクに同期してブレイクします。

設定例 1



CPU0 は、CPU1 のブレイクに同期してブレイクするが、CPU1 は CPU0 のブレイクに同期してブレイクしない。

設定例 2



CPU0、CPU1 どちらのブレイクが発生しても、両方の CPU が停止する。

(5) ソフトウェア・ブレーク(SH-2A Dual のみ)

① SH-2A Dual (SH7205、SH7265 の場合)

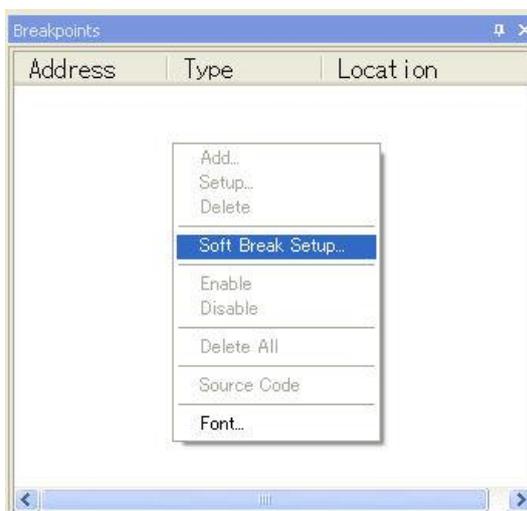
ソフトウェア・ブレークを使用する場合にも、停止させる CPU コアを選択することができます。この設定により、トリガとなった CPU コアのみを停止させるか、両方の CPU コアを停止させるかを選択できます。

ただし、ハードウェア・ブレークのように、トリガの対象となる CPU コアを選択することはできません。

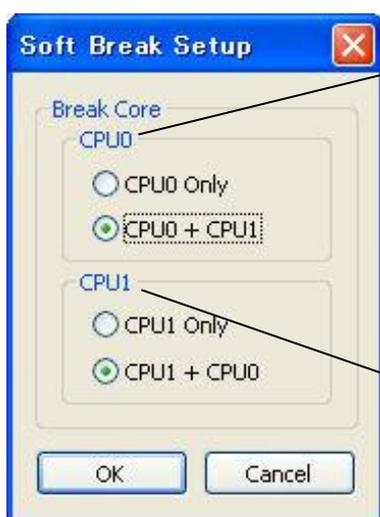
また、CPU0 のトリガで、CPU1 のみを停止させること（もしくは、その逆）は、できません。

ソフトウェア・ブレークの設定は、Breakpoints ウィンドウ内で右クリックしたときに表示されるポップアップメニューの Soft Break Setup をクリックすることで開きます。

なお、この設定は、指定した CPU コアのソフトウェア・ブレーク全てが対象となり、個々のブレークに対して指定することはできません。



Breakpoints ダイアログのポップアップメニュー

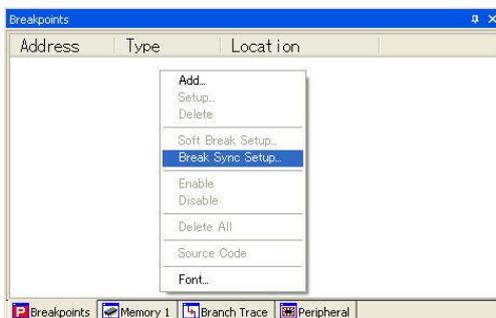


Soft Break Setup ダイアログ

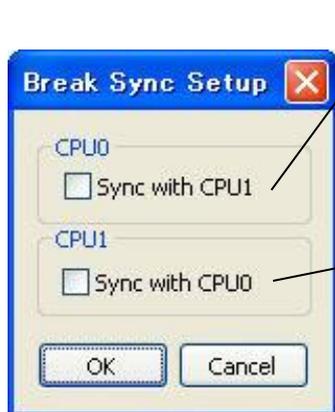
- Break Core (CPU0)  
CPU0 がブレーク時に停止させる CPU コアを選択します。
- CPU0 Only      CPU0 がブレーク時には、CPU0 のみ停止します。
- CPU0 + CPU1    CPU0 がブレーク時に両方の CPU が停止します。
- Break Core (CPU1)  
CPU1 がブレーク時に停止させる CPU コアを選択します。
- CPU1 Only      CPU1 がブレーク時には、CPU1 のみ停止します。
- CPU1 + CPU0    CPU1 がブレーク時に両方の CPU が停止します。

## ② SH-4A Multi (SH7786 の場合)

ハードウェア・ブレークと同様に、Break Sync Setup で選択したブレークの同期条件によって、ソフトウェア・ブレークも動作します。ソフトウェア・ブレークとハードウェア・ブレークで別の同期条件を設定することはできませんので、ご注意ください。



Breakpoints ダイアログのポップアップメニュー



## CPU0

CPU1 がブレーク時に、CPU1 と同期して CPU0 をブレークさせるか選択します。

チェックボタンをオンにした場合、CPU0 は CPU1 のブレークに同期してブレークします。

## CPU1

CPU0 がブレーク時に、CPU0 と同期して CPU1 をブレークさせるか選択します。

チェックボタンをオンにした場合、CPU1 は CPU0 のブレークに同期してブレークします。

## 4. チュートリアル

本章では、弊社製 SH7205 ボード AP-SH2AD-0A を使用して、SH2A-Dual を対象とした XsSight の使用方法を説明します。

また、プログラムは、AP-SH2AD-0A に付属の USB ファンクション用のサンプルプログラムを使用します。

このプログラムでは、CPU0 で USB 仮想シリアル通信を、CPU1 でボード上の LED の点滅を行っています。

なお、AP-SH2AD-0A、および、ボード付属のサンプルプログラムにつきましては、ボード付属の「AP-SH2AD-0A ハードウェアマニュアル」、「AN152 サンプルプログラム解説」を参照してください。

### (1) 必要なもの

本章の内容を実際に確認される場合には、下記のものが必要となります。

- ・ XrossFinder、または XrossFinder Evo (以下、H-UDI アダプタと記載します)
- ・ AP-SH2AD-0A
- ・ AP-SH2AD-0A 付属の CD-ROM
- ・ 5V 電源
- ・ パソコン (XsSight がインストールされ、正常に動作していること)
- ・ ターミナルソフト
- ・ USB ケーブル 2 本

### (2) 準備

実際に動作をさせる前に下記の準備を行ってください。

#### ① AP-SH2AD-0A の設定

AP-SH2AD-0A の各ディップスイッチの設定をボード付属の CD-ROM 内にあります「AN152 サンプルプログラム解説」の「1. 2 動作モード」の内容に合わせてください。

なお、動作モードは DEBUG、USB チャネル設定はファンクションとしてください。

#### ② サンプルプログラムのコピー

ボード付属の CD-ROM の sample フォルダ以下をパソコンのハードディスク上へコピーしてください。

本書では、コピー先を C:\ap\_sh2ad\_0a フォルダとして、説明を行います。

#### ③ パソコンとボードの接続

H-UDI アダプタとパソコンを付属の USB ケーブルで接続します。

H-UDI アダプタと AP-SH2AD-0A を H-UDI ケーブルで接続します。

(H-UDI アダプタとパソコン、ボードの接続の詳細につきましては、「XrossFinder ユーザーズマニュアル」または「XrossFinder Evo 取扱説明書」を参照してください。)

AP-SH2AD-0A の CN3 の USB ポートとパソコンの USB ポートを USB ケーブルで接続してください。

#### ④ 電源の接続

電源をボードに接続してください。

### (3) 動作確認

下記の手順で XsSight の操作を説明します。

- ・プログラムのダウンロード (① ~ ③)
- ・各 CPU コアの動作を確認 (④ ~ ⑧)
- ・ブレークの確認 (⑨ ~ ⑭)

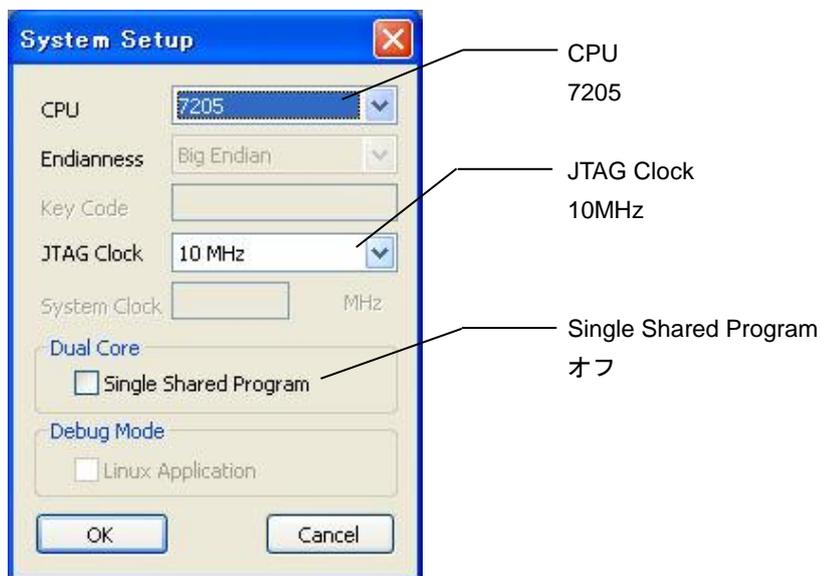
まず、起動からプログラムのダウンロードまでの手順を説明します。

#### ① XsSight の起動

XsSight を起動し、System Setup 設定ダイアログを下記のように設定し、ボードの電源を投入後、OK をクリックしてください。

このとき、CPU1 用のウィンドウが開かない場合、Master Control ツールバーを使用してウィンドウを開いてください。

なお、本章で使用するサンプルプログラムは、CPU0 と CPU1 でプロジェクトが分かれていますので、Single Shared Program の設定はオフとなります。



## ② コマンドファイルの指定

CPU0 用ウィンドウのコマンドファイルに (2) の②でコピーをしたサンプルプログラムの **sample** フォルダ内にある「XrossFinder\_sh2ad\_0a.xfc」を設定します。

メニューの [Setup] - [Command File...] をクリックし、Command File ダイアログを開きます。

ダイアログで「XrossFinder\_sh2ad\_0a.xfc」を選択、Execute before download チェックボタンをオンにし OK をクリックします。

CPU1 用のウィンドウでも同様の設定を行ってください、



メニューの [Setup] - [Command File...] をクリックします。



XrossFinder\_sh2ad\_0a.xfc を選択します。

Execute before download チェックボタンをオンにします。

## ③ ダウンロード

サンプルプログラムをダウンロードします。

CPU0、CPU1 別のプロジェクトとなりますので、それぞれのウィンドウから下記のプログラムをダウンロードします。

## CPU0 用ウィンドウ

フォルダ `sample\shc\usbfunc\cpu0\ap_sh2ad_0a_cpu0_usbfunc\Debug`  
 ファイル `ap_sh2ad_0a_cpu0_usbfunc.abs`

## CPU1 用ウィンドウ

フォルダ `sample\shc\usbfunc\cpu1\ap_sh2ad_0a_cpu1_usbfunc\Debug`  
 ファイル `ap_sh2ad_0a_cpu1_usbfunc.abs`

CPU0 用ウィンドウの [File] - [Download Program...] をクリックし Download ダイアログを開きます。

ダイアログで「ap\_sh2ad\_0a\_cpu0\_usbfunc.abs」を選択し、OK をクリックします。

プログレスバーが表示されダウンロードが実行されます。

CPU1 用ウィンドウでも同様に、「ap\_sh2ad\_0a\_cpu1\_usbfunc.abs」をダウンロードします。

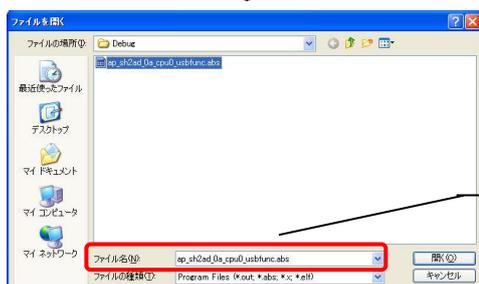
ダウンロード後、CPU0 では、C ソースが表示され、CPU1 では、アセンブリが表示されます。

なお、CPU0 のソースが表示されない場合には、一度、サンプルプログラムをビルドしてください。

ビルドの方法につきましては、「AN152 サンプルプログラム解説」を参照してください。

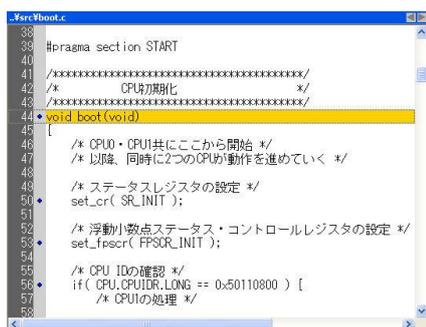


メニューの [File] - [Download Program...] をクリックします。

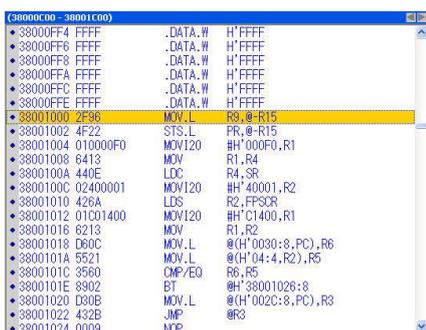


ap\_sh2ad\_0a\_cpu0\_usbfunc.abs を選択します。  
(CPU1 の場合には ap\_sh2ad\_0a\_cpu1\_usbfunc.abs)

ファイル名(N): ap\_sh2ad\_0a\_cpu0\_usbfunc.abs



ダウンロード後の CPU0 側の Source ウィンドウは、左のような表示になります。



上記の手順で、CPU1 側のダウンロードを行うと CPU1 側の Source ウィンドウは、左のような表示になります。

これで、ボード上の RAM へプログラムをダウンロードすることができました。  
次に、各 CPU コアの動作を確認します。

## ④ 実行

両方の CPU コアで同時にプログラムを実行します。

Master Control ツールバーの All CPU Control ボタンをクリックし両方の CPU コアを選択した後、Master Run ボタンをクリックしプログラムを実行します。



All CPU Control ボタンをクリックします。



Master Run ボタンをクリックします。

## USB 仮想シリアルドライバ

プログラムの実行時に、USB 仮想シリアルドライバを要求された場合には、ボード付属の「AN152 サンプルプログラム解説」の「4. 2. 2 USB ファンクション動作」をご覧ください。ドライバのインストールを行ってください。

## ⑤ CPU0（仮想シリアル通信）の動作確認

CPU0 では、USB 仮想シリアルを使用してエコーバック処理を行っていますので、ターミナルソフトを開き、USB 仮想シリアルポートから通信が行えることを確認します。

ターミナルの設定を、USB 仮想シリアルポート、38400bps、ビット長 8、パリティなし、ストップビット 1、フロー制御なしと設定してください。

USB 仮想シリアルポートのポート番号の確認方法につきましては、「4. 2. 2 USB ファンクション動作」をご確認ください。

ターミナルソフトを起動後、文字を送信するとエコーバックが返ってくることを確認してください。

## ⑥ CPU1（LED の点滅）の動作確認

CPU1 では、CPU 内蔵タイマを使用し、定期的に LED のオン・オフを繰り返しています。

ボード上の LED（LD1、LD2）が点滅していることを確認してください。

## ⑦ CPU0 のみの実行

Master Control ツールバーからの CPU0 のみを実行を確認します。

Master Control ツールバーの Master Break ボタンをクリックし、CPU コアを停止させます。

CPU0 Control ボタンをクリック (CPU0 のみを選択) 後、Master Run ボタンをクリックし、CPU0 のみを実行させます。

CPU0 が処理しているエコーバック処理は動作していますが、LED の点滅は止まっている状態となります。



Master Break ボタンをクリックし、CPU を停止させます。



CPU0 Control ボタンをクリックした後、Master Run ボタンをクリックします。その後、CPU0、CPU1 の動作を確認してください。

## ⑧ CPU1 のみの実行

Master Control ツールバーからの CPU1 のみを実行を確認します。

Master Control ツールバーの Master Break をクリックし、CPU コアを停止させます。

CPU1 Control ボタンをクリック (CPU1 のみを選択) 後、Master Run ボタンをクリックし、CPU1 のみを実行させます。

CPU1 が処理している LED は点滅していますが、エコーバック処理は、止まっている状態となります。



Master Break ボタンをクリックし、CPU を停止させます。



CPU1 Control ボタンをクリックした後、Master Run ボタンをクリックします。その後、CPU0、CPU1 の動作を確認してください。

以上により、Master Control ツールバーによる CPU の選択方法とサンプルプログラムの CPU0、CPU1 の動作がご理解いただけたと思います。

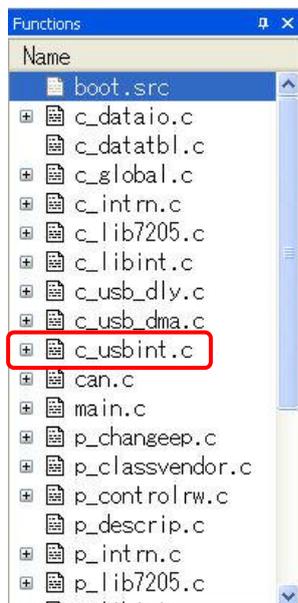
最後に、ブレークの設定について説明します。

## ⑨ ハードウェア・ブレイクの確認 1

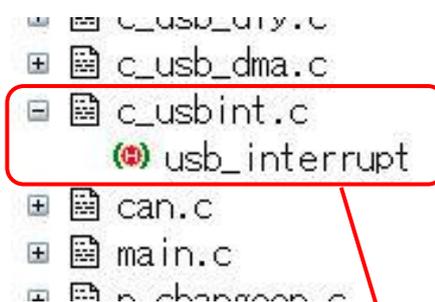
ブレイクの確認を行うために CPU0 が実行している USB ファンクションの割り込み関数にブレイクポイントを仕掛けます。

CPU0 用ウィンドウ内の Function ウィンドウ内のツリーにあるソース c\_usbint.c の usb\_interrupt 関数の左にある緑の括弧をクリックするとブレイクポイントが設定され、括弧の中に赤い丸が表示されます。

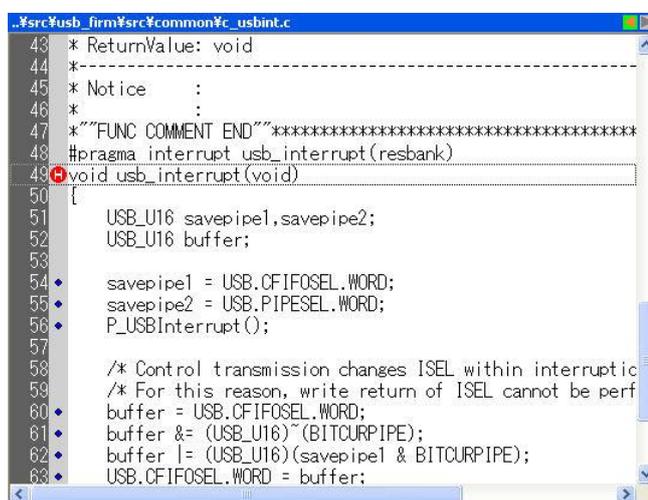
(CPU0 用ウィンドウ内に Function ウィンドウがない場合には、メニューから [View] - [Functions View] をクリックし、Functions ウィンドウを開きます。)



Functions ウィンドウのツリーからソース c\_usbint.c の中にある usb\_interrupt 関数の左にある括弧をクリックしブレイクポイントを設定します。



ブレイクポイントの指定



左の図は、ブレイクポイントを指定した場所を Source ウィンドウで開いたところです。

左端に赤い丸が表示されているところが、ブレイクポイントの位置となります。

⑩ ハードウェア・ブレイクの確認 2

片方の CPU コアを停止させるブレイクの確認を行います。

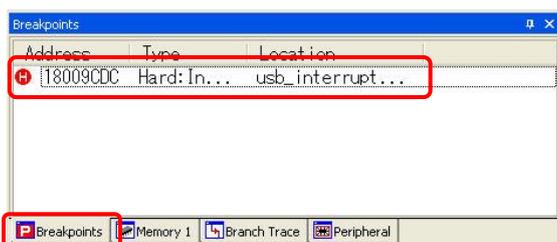
Breakpoints タブをクリックして Breakpoints ウィンドウをクリックし、Breakpoints ウィンドウを表示させます。Breakpoints タブが存在しない場合には、メニューから [View] - [Breakpoints View] をクリックし、Breakpoints ウィンドウを開きます。

Breakpoints ウィンドウには、⑨で登録されたブレイクポイントが表示されています。

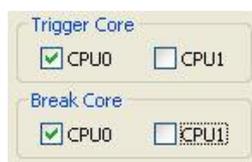
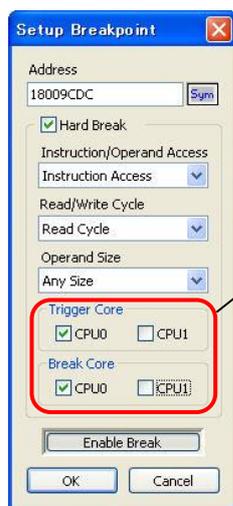
このブレイクポイントをダブルクリック、Setup Breakpoint ダイアログを開きます。

Setup Breakpoint ダイアログの Trigger Core と Break Core の CPU0 をオン、CPU1 をオフにし、OK をクリックします。

以上の設定を行った後、All CPU Control ボタン、Master Run ボタンの順番にクリックし、両方の CPU を動作させます。



Breakpoints タブを選択し、ブレイクポイントをダブルクリックします。



Trigger Core、Break Core の設定を下記のようにし、OK をクリックします。

Trigger Core  
CPU0 オン、CPU1 オフ

Break Core  
CPU0 オン、CPU1 オフ



All CPU Control ボタン、Master Run ボタンの順番にクリックし、両方の CPU を動作させます。

ターミナルソフトから一文字、送信してください。

CPU0 は、受信時の割り込みが発生し、usb\_interrupt 関数の先頭でブレイクしますが、ボード上の LED は、点滅を続け、CPU1 が停止していないことがわかります。

なお、この後、CPU0 を実行するともう一度、同じ箇所でもブレイクが発生しますが、これは、エコーバック処理の送信時の割り込みが発生するためです。

## ⑪ ハードウェア・ブレークの確認 3

両方の CPU コアを停止させるブレークの確認を行います。

⑨で登録したブレークポイントの Setup Breakpoint ダイアログを開き、Break Core の CPU1 の設定もオンに変更し、OK をクリックします。



Setup Breakpoint ダイアログを開き、Break Core の CPU1 をオンにし、OK をクリックします。

(Setup Breakpoint ダイアログを開き方は⑩を参照してください。)

Master Control ツールバーの Master Run ボタンをクリックし、二つの CPU コアが動作している状態にしてください。

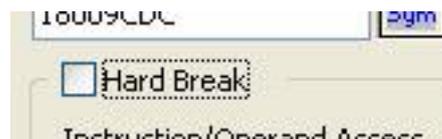
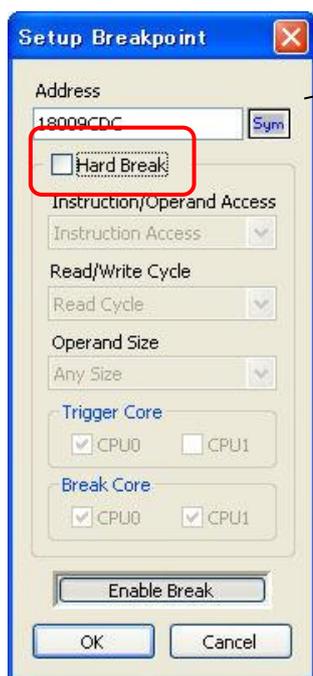
ターミナルソフトから一文字、送信を行ってください。

CPU0 の停止と合わせて、CPU1 も停止し、LED の点滅が止まることを確認してください。

## ⑫ ソフトウェア・ブレークの確認 1

ソフトウェア・ブレークの確認を行うため⑨で登録したブレークポイントの設定をハードウェア・ブレークからソフトウェア・ブレークに変更します。

⑨で登録したブレークポイントの Setup Breakpoint ダイアログを開き、Hard Break チェックボタンをオフにし、OK をクリックします。

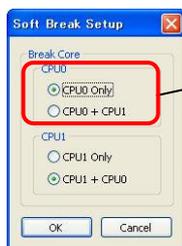
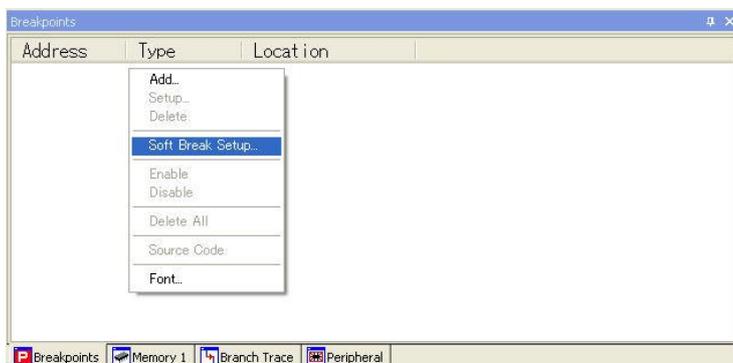


Setup Breakpoint ダイアログを開き、Hard Break チェックボタンをオフにし、OK をクリックします。

## ⑬ ソフトウェア・ブレークの確認 2

ソフトウェア・ブレークを使用して、⑩と同じ動作を確認します。

Breakpoints ウィンドウで右クリックし、表示されるポップアップメニューの **Soft Break Setup** を選択、**Soft Break Setup** ダイアログを開き、**Break Core** の CPU0 の **CPU0 Only** を選択し、**OK** をクリックします。



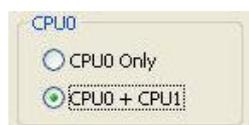
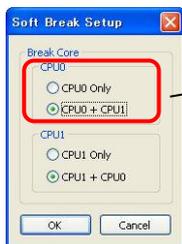
Soft Break Setup ダイアログの、Break Core の CPU0 を CPU0 Only に設定し、OK をクリックします。

Master Control ツールバーの **Master Run** ボタンをクリックし、二つの CPU コアが動作している状態にし、ターミナルソフトから一文字送信後、⑩と同じように CPU0 のみ停止することを確認してください。

## ⑭ ソフトウェア・ブレークの確認 3

ソフトウェア・ブレークを使用して、⑪と同じ動作を確認します。

Soft Break Setup ダイアログを開き、Trigger Core の CPU0 の **CPU0 + CPU1** を選択します。



Soft Break Setup ダイアログを開き、Break Core の CPU0 を CPU0 + CPU1 に設定し、OK をクリックします。

Master Control ツールバーの **Master Run** ボタンをクリックし、二つの CPU コアが動作している状態にし、ターミナルソフトから一文字送信後、⑪と同じように両方の CPU コアが停止することを確認してください。

以上のようにブレークポイントの設定をすることで、どの状況でどの CPU コアを停止させるかを選択することができますので、デバッグ時の必要に応じて使い分けることができます。

## 5. 制限事項

H-UDI アダプタ	項目	内容
XrossFinder	CPU	SH7786 には対応していません。
	Run LED	RUN LED の表示は、両方の CPU コアが停止している状態で消灯し、それ以外の場合は、点灯します。
	外部トリガ	Dual Core CPU のデバッグ時には、外部トリガ機能は使用できません。
	実行時間	Log ウィンドウに出力される実行時間の 1 秒以下の表示が 3 桁となります。
XrossFinder Evo XrossFinder 共通	リセット	SH2A-Dual では、片方の CPU コアだけをリセットすることができませんので、各コア用のメインウィンドウの Reset は、無効表示となります。 リセットを行うときには、Master Control ツールバーの Master Reset を使用してください。

## ご注意

- ・本文書の著作権は（株）アルファプロジェクトが保有します。
- ・本文書の内容を無断で転載することは一切禁止します。
- ・本文書に記載された回路図およびサンプルプログラム等の著作権は（株）アルファプロジェクトが保有しますが、お客様のアプリケーションで使用される場合には、ご自由にご利用いただけます。
- ・本文書の内容は、将来予告なしに変更されることがあります。
- ・本文書に記載されている内容およびサンプルプログラムについての質問等のサポートは一切受け付けておりませんのでご了承ください。
- ・本文書の内容については、万全を期して作成いたしました。万が一不審な点、誤りなどお気づきの点がありましたら弊社までご連絡下さい。
- ・本文書の内容およびサンプルプログラムに基づき、アプリケーションを運用した結果、万一損害が発生しても、弊社では一切責任を負いませんのでご了承下さい。

- ・ SuperH は、ルネサスエレクトロニクスの登録商標、商標または商品名称です。
- ・ 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。



株式会社アルファプロジェクト  
〒431-3114  
静岡県浜松市中央区積志町 834  
<https://www.apnet.co.jp>  
E-Mail: [query@apnet.co.jp](mailto:query@apnet.co.jp)