



μC3/Standard チュートリアルガイド

Cortex-R4 - RZ/T1 e2Studio 編

Rev. 1.0

February 22, 2016

改訂記録

Rev.	発行日	改訂内容	
		ページ	内容
1.0	2016.02.22	—	—

目次

目次.....	3
1 はじめに.....	4
2 動作確認条件	5
3 関連ドキュメント.....	6
4 RTK7910022C00000BR ボードの設定	7
4. 1 スイッチ・ジャンパの設定	7
4. 2 シリアル設定	7
4. 3 Ethernet ポートの設定	7
5 フォルダ構成	8
6 サンプルプログラムのビルド	9
7 サンプルの説明	14
7. 1 μ C3 のサンプル概要	14
7. 2 μ Net3 サンプル概要	16
7. 3 e2Studio でデバッグ	19

1 はじめに

本ドキュメントは、ルネサス エレクトロニクス製 RZ/T1 RTK7910022C00000BR ボード (以下、評価ボード) 上で μ C3/Standard・ μ Net3 のプログラムを動作させるまでの、手順を示したドキュメントです。

2 動作確認条件

本書のプログラムは下記の条件で動作確認しています。

項目	説明
CPU	RZ/T1 (Cortex-R4Core)
コンパイラ	KPIT-GNUARM-NONE-EABI 14.02(GCC4.9.1)
デバッガ	Segger jLink Ultra+
使用ボード	RTK7910022C00000BR ボード

3 関連ドキュメント

本書と併せて次のドキュメントも参照ください。

- μ C3/Standard ユーザーズガイド
プロセッサ非依存のカーネル共通機能について記述しています。
- μ C3/Standard ユーザーズガイドープロセッサ依存部 Cortex-R4
プロセッサに依存したカーネル機能とデバイスドライバ等について記述しています。
- その他，開発環境のマニュアル
適宜，使用されるコンパイラ，デバッガ ICE のマニュアルを参照ください。

4 RTK7910022C00000BR ボードの設定

4. 1 スイッチ・ジャンパの設定

ボードの設定はすべてデフォルトで使⽤します。

スイッチ名称	Pin 番号	設定	意味
SW4	1	ON	出荷時の設定
	2	ON	
	3	ON	
	4	ON	
	5	ON	
	6	OFF	
JP2		2-3 をショート	出荷時の設定
JP7		1-2 をショート	出荷時の設定

4. 2 シリアル設定

SCIFA2 は μ C3/Standard サンプルがシリアル通信として使⽤します。ホスト PC と USB miniAB コネクタ(J8)を接続してください。

ホスト PC のターミナルソフトは次のように設定してください。

SCIFA2 (J8)
<ul style="list-style-type: none">・通信速度 : 115200bps・データ長 : 8 ビット・パリティ : なし・ストップビット長 : 1 ビット・フロー制御 : なし

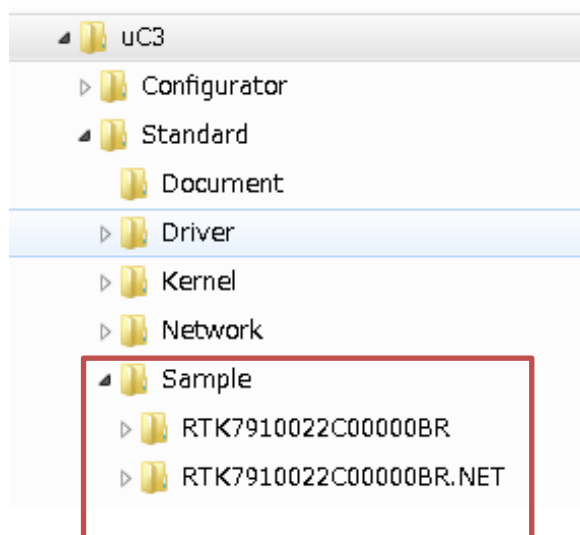
4. 3 Ethernet ポートの設定

J2 (EtherCAT2)に LAN ケーブルを接続して下さい。

コンフィギュレータで次の設定を変更することで、J2(EtherCAT2)の代わりに J7(EtherMAC)で動作します。

5 フォルダ構成

RZ/T1 評価ボード用サンプルは C:\uC3\Standard\Sample\ RTK7910022C00000BR ¹フォルダにあります。



フォルダ	サンプル概要
RTK7910022C00000BR	シングルコアサンプルプログラム
RTK7910022C00000BR.NET	デュアルコアサンプルプログラム

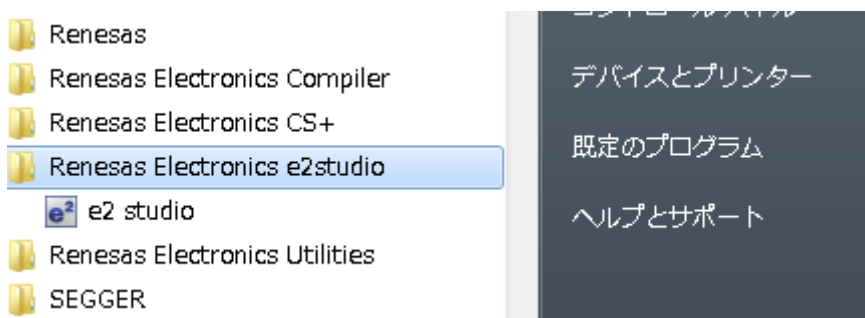
¹ 本書ではパッケージをデフォルトインストール先 C:\uC3 とした場合を例として説明します。別フォルダにインストールされた場合は適宜読み替えてください。

6 サンプルプログラムのビルド

各サンプルプログラムは次の手順でビルドします。

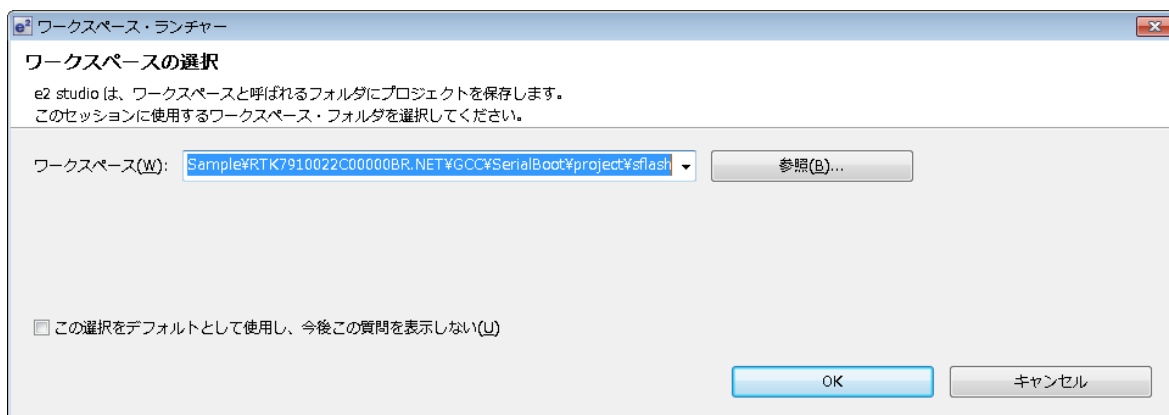
(1) e2 Studio の起動

Windows のスタートメニューより「Renesas Electronics e2studio」を選択して e2 studio を起動します。



(2) ワークスペースの選択

e2 studio のワークスペースを選択するダイアログが起動します。ここではサンプル配下のプロジェクト配下のディレクトリを選択します。



ワークスペースのディレクトリは以下のものを選択できます。

■ μ C3 のサンプル

○RAM 起動版(JTAG より起動)

`$(SAMPLE)¥RTK7910022C00000BR ¥GCC¥RAM¥project¥ram`

○SerialBoot 起動版(QSPI より起動)

`$(SAMPLE)¥RTK7910022C00000BR ¥GCC¥SerialBoot¥project¥sflash`

■ μ Net3 のサンプル

○RAM 起動版(JTAG より起動)

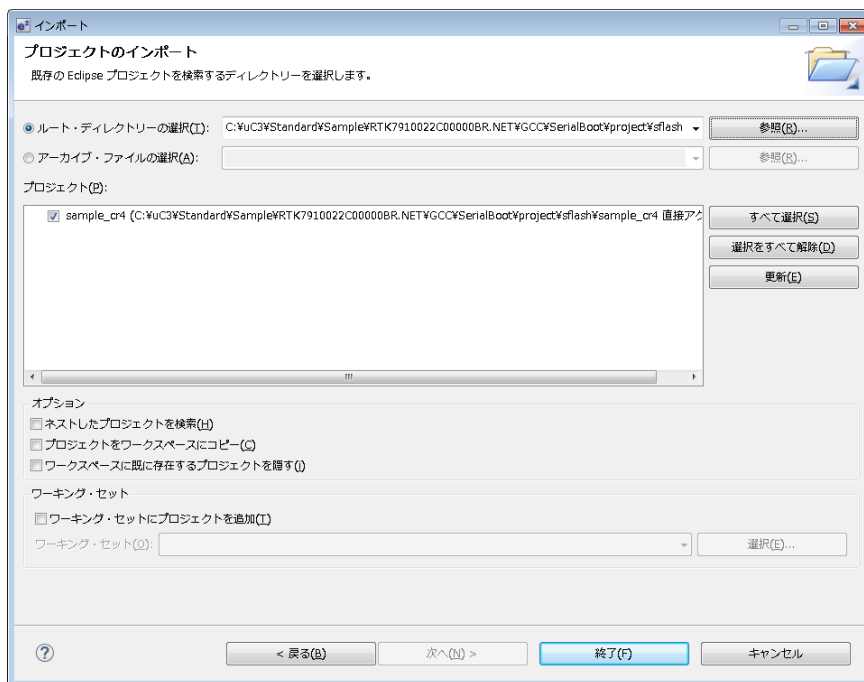
`$(SAMPLE)¥RTK7910022C00000BR.NET¥GCC¥RAM¥project¥ram`

○SerialBoot 起動版(QSPI より起動)

`$(SAMPLE)¥RTK7910022C00000BR.NET¥GCC¥SerialBoot¥project¥sflash`

(3) プロジェクトのインポート

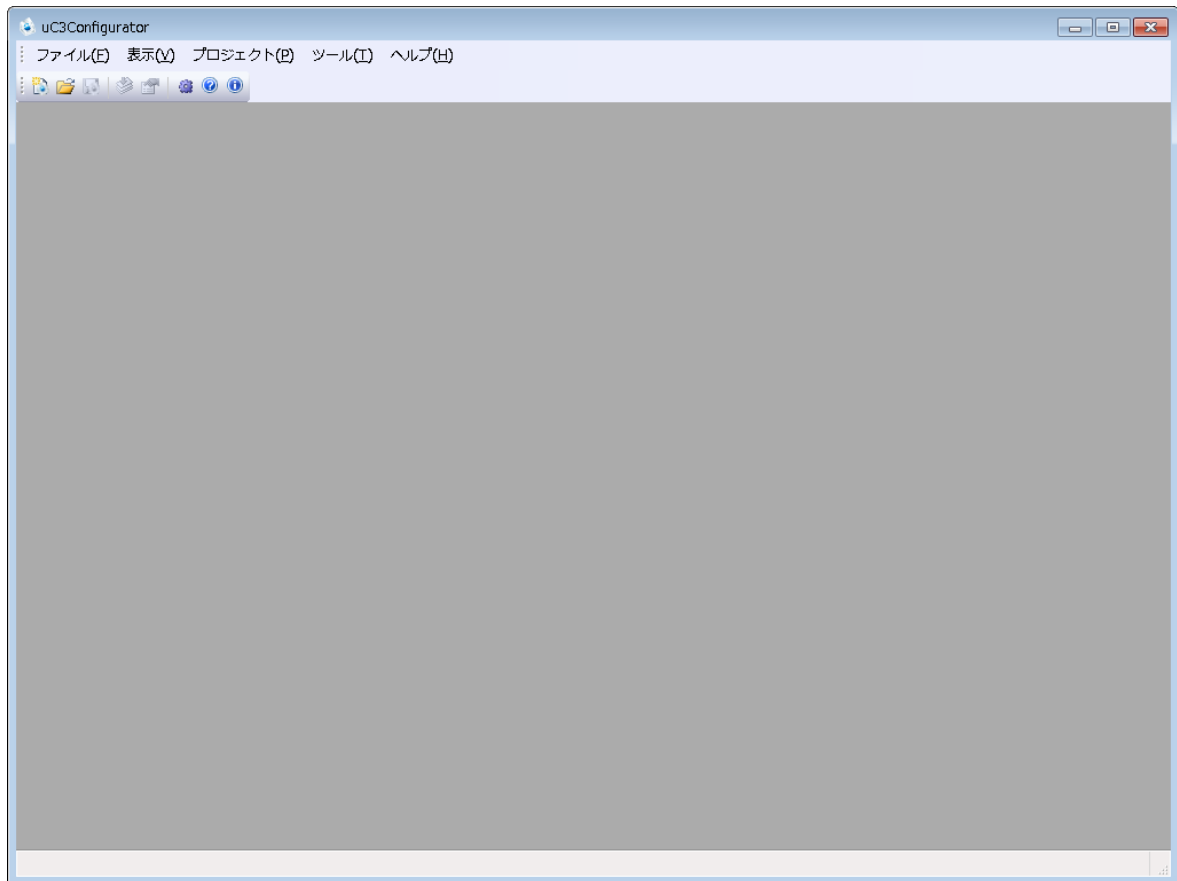
メニューから「ファイル」→「インポート」を選択し、「既存のプロジェクトをインポート」を選択します。



表示されたプロジェクト「sample_cr4」にチェックし「終了」ボタンを押下するとインポートが完了します。

(4) コンフィギュレータの起動

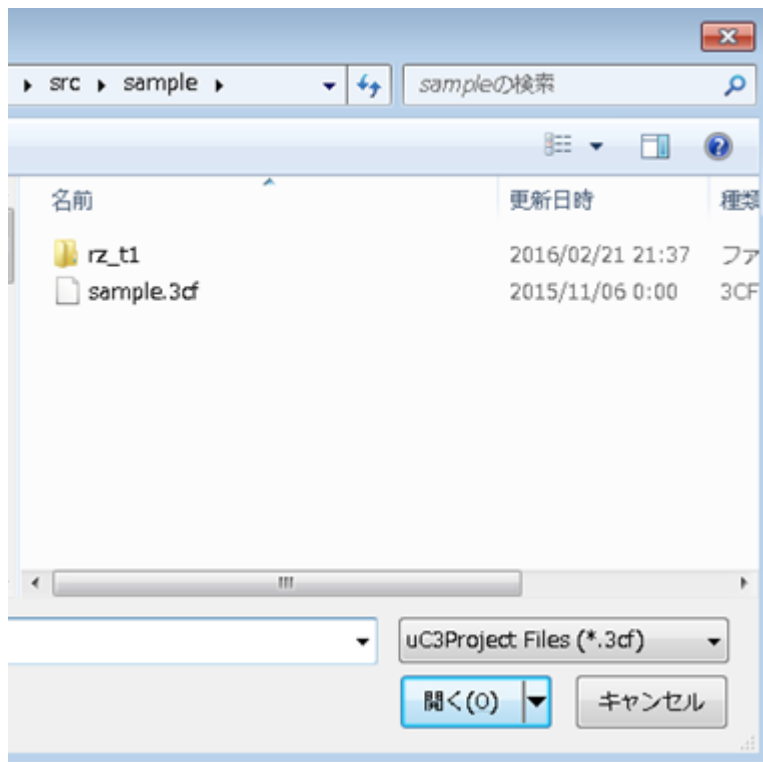
C:\¥uC3¥Configurator¥Configurator.exe を起動します。



「ファイル(F)」 → 「開く」 から対象となる sample.3cf ファイルを開きます。

3cf ファイルは以下の場所に存在します。

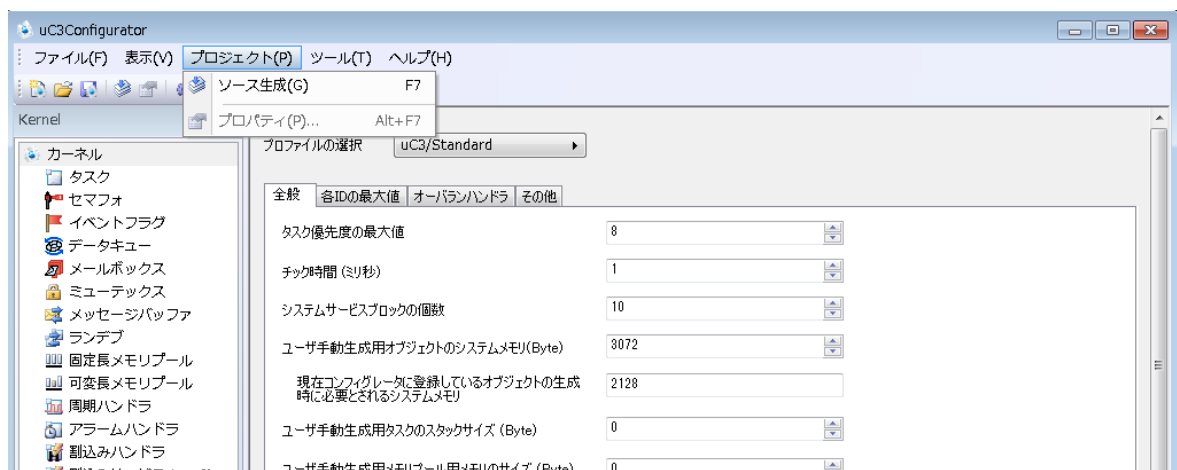
- RTK7910022C00000BR ¥GCC¥SerialBoot¥src¥sample¥配下
- RTK7910022C00000BR ¥GCC¥RAM¥src¥sample¥配下
- RTK7910022C00000BR.NET ¥GCC¥SerialBoot¥src¥sample¥配下
- RTK7910022C00000BR.NET ¥GCC¥RAM ¥src¥sample¥配下



(5) ソースの生成

プロジェクトを読み込んだらソース生成を行います。

生成先は **uC3** コンフィギュレータのプロジェクトファイル(3cf)と同じフォルダに行います。



(6) ビルド構成の選択

OSの動作の構成を TCM, SRAM, SDRAM から選択できます。

プロジェクトマネージャのポップアップメニューよりビルド構成を変更してください。

◆RAM プロジェクトの場合

RAM_TCM

RAM_SRAM

RAM_SDRAM

◆SerialBoot プロジェクトの場合

SerialBoot_TCM

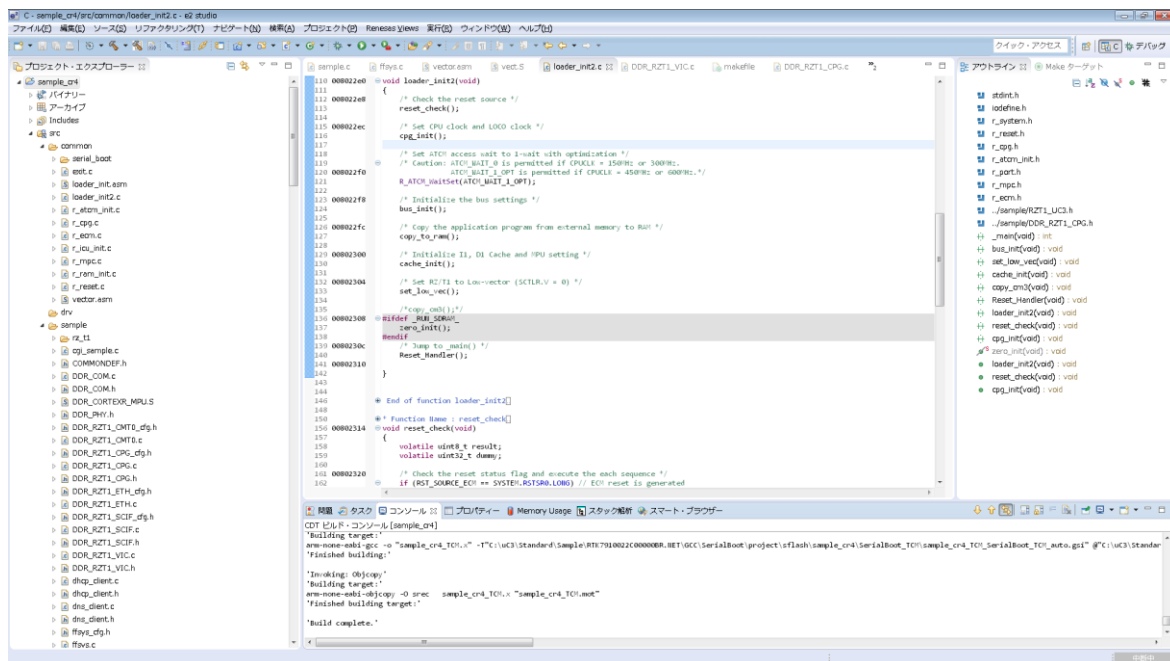
SerialBoot_SRAM

SerialBoot_SDRAM

(7) ビルド

プロジェクト→すべてビルドを選択してビルドします。

うまくいくと実行バイナリが作成されます。



7 サンプルの説明

7. 1 μ C3 のサンプル概要

Cortex-R4 コアのみ使用したサンプルです。このサンプルプログラムはデバッガ による RAM 起動や SerialBoot 用のブートローダーなどによって TCM, RAM, SDRAM の初期化が済んでいる状態でロードされて実行する前提で作成しています。よって、本サンプルの起動前にブートローダーやデバッガ経由でクロックやメモリなどの初期化を行ってください。

◆ シリアル通信

起動後にタイトルを表示し、入力文字をエコーバックします。

◆ LED

以下 LED を 500ms 周期で点滅を繰り返します。

GPIO	LED
PORTF_B7	LED5
PORT5_B6	LED0

◆ プロセッサ依存部の初期化

リセットハンドラ (prst.S) にて次の初期化を行います。

CPU を SYSTEM モードに設定しスタックポインタを設定

MPU とキャッシュを初期化

FIQ、Undefined、Abort、Supervisor、IRQ モードのスタックポインタを設定

CPU を SYSTEM モードに設定し、main へ移行

◆ SerialBoot からの起動の場合

シリアルブートの場合 loader_param に基づいてローダーのプログラムがメモリに展開され実行されます。その後、以下のようなステップで起動します。

- ・クロック設定等ボードの処理を初期化します。
(クロックは CPU が 450MHz, 外部バスクロックは 50MHz に設定されます。)
- ・メモリ領域・各種設定を初期化します。
- ・ローダーが例外ベクタ・プログラム・データをメモリに展開します。
- ・リセット処理にジャンプします。

シリアルブートの場合、CPU クロック初期化やバスステートコントローラの初期化はローダーで行っています。そのため、コンフィグレータで外部バスコントローラに関する IO ポート設定は行わないでください。sample.c では hw_init() を呼び出さずに IO ポートの設定コードのみ呼び出しています。

◆メモリマップ

サンプルでは TCM 動作, SRAM 動作, SDRAM 動作で動作する領域を選択できるように設定してあります。

物理アドレス	領域
0x00000000	ATCM
0x00800000	BTCM
0x04000000	命令用 RAM
0x08000000	バッファ用 RAM
0x10000000	SPI FLASH
0x20000000	データ RAM
0x22000000	データ RAM(ミラー)
0x24000000	命令用 RAM(ミラー)
0x30000000	SPI FLASH(ミラー)
0x40000000	CS0,CS1 外部アドレス空間 NOR Flash1,2 (ミラー)
0x48000000	CS2,CS3 外部アドレス空間 SDRAM 1. 2(ミラー)
0x50000000	CS4,CS5 外部アドレス空間(ミラー)
0x60000000	CS0,CS1 外部アドレス空間(NOR Flash)
0x68000000	CS2,CS3 外部アドレス空間(SDRAM)
0x70000000	CS4,CS5 外部アドレス空間
0x80000000	ペリフェラル

プログラムのメモリ配置を変更する場合は、リンカスクリプトの配置（セクション）を変更すると共に、リセットハンドラ（prst.S）に定義している、MPU テーブルも変更してください。MPU テーブルではメモリ配置に依存したアドレスとメモリ属性を設定しています。

7. 2 μ Net3 サンプル概要

Cortex-R4 コアのみ使用した DHCP クライアント,HTTP サーバー,FTP サーバを組み合わせたサンプルプログラムです。このサンプルプログラムはデバッガ による RAM 起動や SerialBoot 用のブートローダーなどによって TCM, RAM, SDRAM の初期化が済んでいる状態でロードされて実行する前提で作成しています。よって、本サンプルの起動前にブートローダーやデバッガ経由でクロックやメモリなどの初期化を行ってください。

◆ シリアル通信

uNet3 の動作を調整したりするために shell によるコンソールを提供します。
help コマンドで内容を表示することができます。

◆ LED

以下 LED を 500ms 周期で点滅を繰り返します。

GPIO	LED
PORTF_B7	LED0

◆ プロセッサ依存部の初期化

リセットハンドラ (prst.S) にて次の初期化を行います。

CPU を SYSTEM モードに設定しスタックポインタを設定

MPU とキャッシュを初期化

FIQ、Undefined、Abort、Supervisor、IRQ モードのスタックポインタを設定

CPU を SYSTEM モードに設定し、main へ移行

◆ DHCP クライアント

DHCP サーバから動的に IP アドレスを取得し自ホストに割り当てます。

取得した IP アドレスはシリアルコンソール(SCIF2)から「ip」と入力することでシリアルに IP アドレスが出力されます。

◆ HTTP サーバー

ウェブブラウザから LED の点滅間隔を 100msec 単位で変更することが可能です。

ウェブブラウザから下記 URL を開き、ターゲットボードにアクセスしてください。

http://IP アドレス/

◆ FTP サーバー

Windows のコマンドプロンプトから次のコマンドを入力してターゲットボード上の

FTP サーバに接続してください。

ftp ターゲットの IP アドレス

ユーザ名とパスワードは必要ありません。

◆ SerialBoot からの起動の場合

シリアルブートの場合、**loader_param** に基づいてローダーのプログラムがメモリに展開され実行されます。その後、ブートローダーのコードで以下のようなステップで OS が RAM に展開され起動します。

- ・クロック設定等ボードの処理を初期化します。
(クロックは CPU が 450MHz, 外部バスクロックは 50MHz に設定されます。)
- ・メモリ領域・各種設定を初期化します。
- ・ローダーが例外ベクタ・プログラム・データをメモリに展開します。
- ・リセット処理にジャンプします。

シリアルブートの場合、CPU クロック初期化やバスステートコントローラの初期化はローダーで行っています。そのため、コンフィグレーターで外部バスコントローラに係する IO ポート設定は行わないでください。**sample.c** では **hw_init()** を呼び出さずに IO ポートの設定コードのみ呼び出しています。

◆メモリマップ

サンプルでは TCM 動作,SRAM 動作, SDRAM 動作で動作する領域を選択できるように設定してあります。

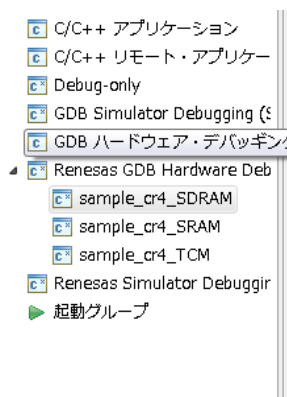
物理アドレス(PA)	領域
0x00000000	ATCM
0x00800000	BTCM
0x04000000	命令用 RAM
0x08000000	バッファ用 RAM
0x10000000	SPI FLASH
0x20000000	データ RAM
0x22000000	データ RAM(ミラー)
0x24000000	命令用 RAM(ミラー)
0x30000000	SPI FLASH(ミラー)
0x40000000	CS0,CS1 外部アドレス空間 NOR Flash1,2 (ミラー)
0x48000000	CS2,CS3 外部アドレス空間 SDRAM 1. 2(ミラー)
0x50000000	CS4,CS5 外部アドレス空間(ミラー)
0x60000000	CS0,CS1 外部アドレス空間(NOR Flash)
0x68000000	CS2,CS3 外部アドレス空間(SDRAM)
0x70000000	CS4,CS5 外部アドレス空間
0x80000000	ペリフェラル

プログラムのメモリ配置を変更する場合は、リンカスクリプトの配置（セクション）を変更すると共に、リセットハンドラ（prst.S）に定義している、MPU テーブルも変更してください。MPU テーブルではメモリ配置に依存したアドレスとメモリ属性を設定しています。

7. 3 e2Studio でデバッグ

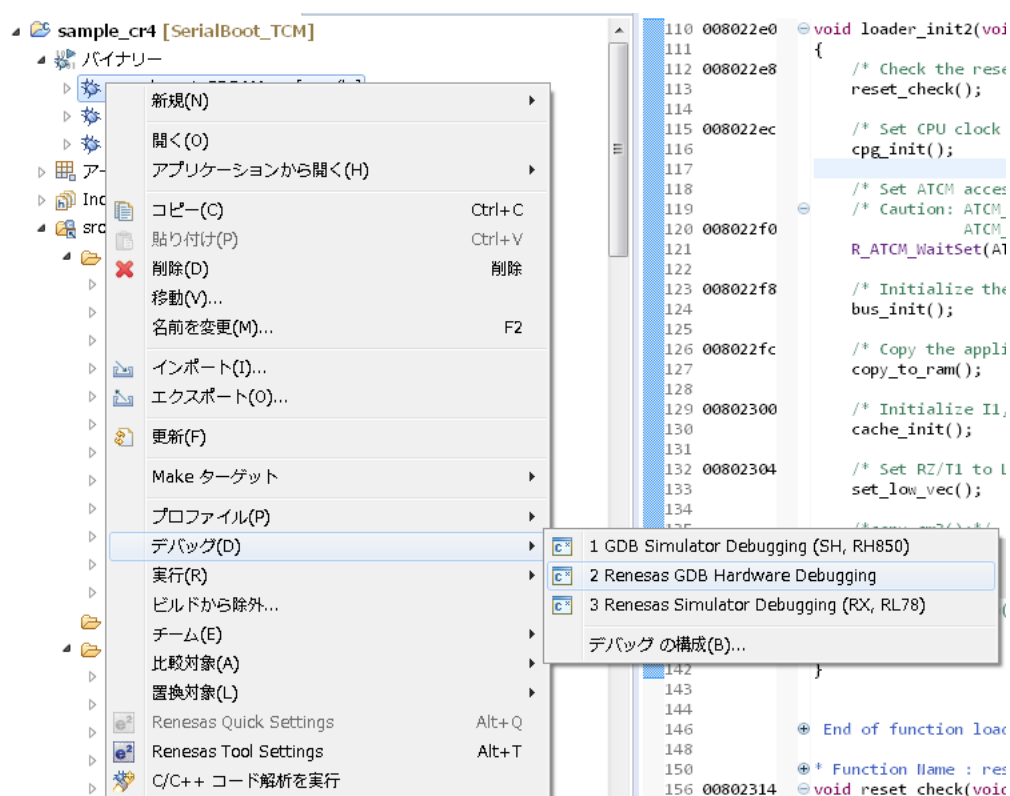
μC3/Standard では e2Studio 用のサンプルプログラムのデバッグ設定を予め用意していますので、それを利用します。

デバッガ設定の設定は sample_cr4_SDRAM, sample_cr4_SRAM, sample_cr4_TCM といった名前になっています。



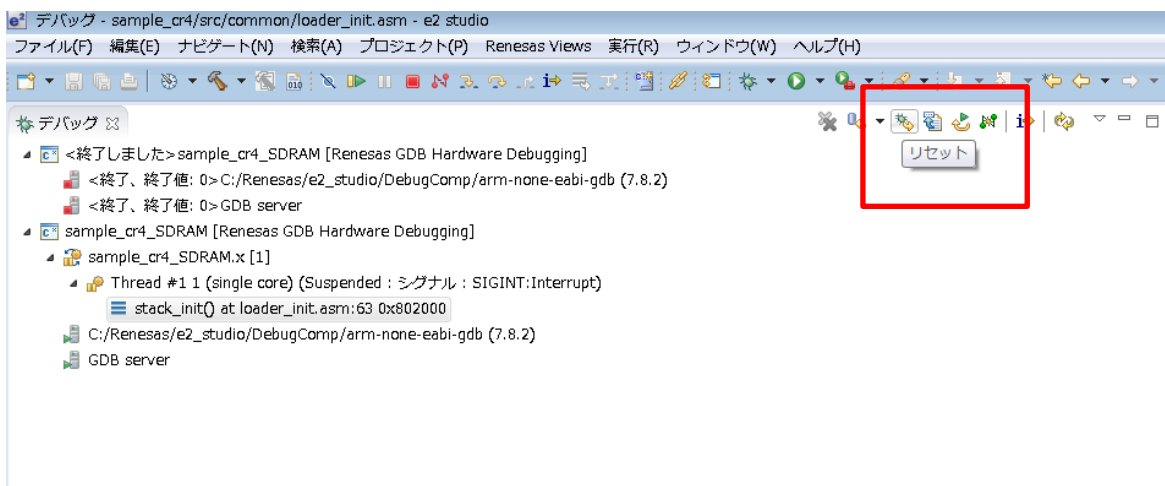
デバッグで「Renesas GDB Hardware Debugging」を選択肢します。

J-Link が初期化を行い（SerialBoot の場合は SPI Flash に書き込みを行い）起動を行いデバッグが可能です。



◆ SerialBoot からの起動の場合

SerialBoot からの起動の場合、ダウンロード（書き込み）後にリセットをしてから動作させてください。リセットは以下の場所から行うことができます。



μ C3/Standard チュートリアルガイド Cortex-R4 e2studio 編

2016 年 02 月 22 日

Rev.1.0

イー・フォース株式会社 <http://www.eforce.co.jp/>

お問い合わせ support@eforce.co.jp

Copyright (C) 2016 eForce Co.,Ltd. All Rights Reserved.