

# RX64M グループ

## カメラ活用センサソリューション

### RX Driver Package Application

R01AN2462JJ0100  
Rev.1.00  
2015.10.01

#### 要旨

本資料は、カメラ活用センサソリューションのアプリケーションノートです。本アプリケーションノートには、カメラから取り込んだデータを元に人物検知、動体検知、歪み補正を行うサンプルプログラムが含まれており、検知状態をWebブラウザ上で視認することができます。

本アプリケーションノートは、RX Driver Package Applicationの1つであり、Webサーバ機能を備えた「RX64M グループ TCP/IPプロトコルスタックを用いた産業向けネットワークソリューション RX Driver Package Application」(r01an2153)をベースとして、画像処理システム、Webカメラシステム機能を追加しています。

画像処理システムは、株式会社 日立産業制御ソリューションズの画像処理ミドルウェアを使用しています。また、ミドルウェアの保護の為、RX64M搭載のTrusted Memory(以下、TMと称す)機能を使用しています。

TM機能は、第三者によるソフトウェアのリード防止機能であり、外部だけでなく内蔵フラッシュメモリ等のマイコン内部からもコード読出し(コピー)を不可能とし、命令実行のみ可能となります。

本アプリケーションノートは、RX64M搭載のGR-KAEDE上で動作します。

GR-KAEDE仕様、購入先情報:

<http://gadget.renesas.com/ja/product/kaede.html>

RX Driver Package Applicationは、RX Driver Packageと組み合わせて動作するサンプルアプリケーションの総称です。

RX Driver Packageとそれを構築するモジュール群であるFirmware Integration Technology(以下、FITと称す)については、下記を参照してください。

<http://japan.renesas.com/products/mpumcu/rx/child/fit.jsp>

#### 動作確認デバイス

RX64M グループ (GR-KAEDE)

## 目次

1. 概要 .....	4
1.1 本アプリケーションノートについて .....	4
1.2 動作環境 .....	5
1.3 モジュール構成 .....	6
1.4 プロジェクトについて .....	8
2. 開発環境の入手 .....	9
2.1 e <sup>2</sup> studio の入手とインストール方法 .....	9
2.2 コンパイラパッケージの入手方法 .....	10
2.3 GR-KAEDE ボードの入手方法 .....	10
3. プロジェクトの構築 .....	11
3.1 ワークスペースの作成 .....	11
3.2 プロジェクトのインポート .....	12
4. 動作確認 .....	15
4.1 GR-KAEDE ボードの Dip Switch 確認 .....	15
4.2 プロジェクトのビルド .....	15
4.3 デバッグの準備 .....	17
4.3.1 機器の構成 .....	17
4.3.2 クライアント PC の設定 .....	18
4.4 プロジェクトのデバッグ .....	21
5. 画像処理システム仕様 .....	30
5.1 性能概要 .....	30
5.2 動作概要 .....	31
5.2.1 人物検知機能 .....	31
5.2.2 動体検知機能 .....	32
5.2.3 歪み補正機能 .....	32
5.3 ファイル一覧 .....	33
5.4 データ構造体 .....	34
5.5 画像処理ミドルウェア関数リファレンス .....	35
5.5.1 Mdl_IP_Init .....	35
5.5.2 Mdl_IP_PersonDetection .....	36
5.5.3 Mdl_IP_PersonDetection_ParamChg .....	37
5.5.4 Mdl_IP_MovingDetection .....	38
5.5.5 Mdl_IP_MovingDetection_ParamChg .....	39
5.5.6 Mdl_IP_ImgRevise .....	40
5.5.7 Mdl_IP_ImgRevise_ParamChg .....	41
6. CS+を使用する場合 .....	42
6.1 CS+の入手とインストール方法 .....	42
6.2 プロジェクトのインポート .....	43
6.3 設定変更 .....	46
7. 補足 .....	53

---

7.1	無償評価版の「RX ファミリー用 C/C++コンパイラパッケージ」を利用する場合の注意事項 ...	53
7.2	Web ブラウザ上でのキャプチャデータ更新遅延について .....	53
7.3	コードフラッシュ内の書き込み不可な領域について .....	54
8.	カメラソリューション製品紹介 .....	55
8.1	がじえっとるねさず GR-KAEDE ボード .....	55
8.2	HMI 拡張ボードを使ったカメラ機能/音声再生機能デモンストレーション .....	55

## 1. 概要

### 1.1 本アプリケーションノートについて

本資料は、カメラ活用センサソリューションのアプリケーションノートです。本アプリケーションノートには、カメラから取り込んだデータを元に人物検知、動体検知、歪み補正を行うサンプルプログラムが含まれており、検知状態を Web ブラウザ上で視認することができます。

本アプリケーションノートは、RX Driver Package Application の 1 つであり、Web サーバ機能を備えた「RX64M グループ TCP/IP プロトコルスタックを用いた産業向けネットワークソリューション RX Driver Package Application」(r01an2153)をベースとして、画像処理システム、Web カメラシステム機能を追加しています。

Web サーバは、TCP/IP 上で動作するアプリケーションプログラムであり、一般的には Web ブラウザからアクセスされ、Web サーバ上に保存されているコンテンツを TCP/IP を用いて Web ブラウザに転送する機能を提供します。

画像処理システムは、GR-KAEDE と接続する外付けカメラモジュール(Omnivision 社製 OV7740)からのキャプチャデータを使って、人物検知、動体検知、歪み補正の画像処理を実行します。Web サーバの CGI 機能を使って Web ブラウザ上で画像処理システムのパラメータ変更を可能にしています。

Web カメラシステムは、外付けカメラからのキャプチャデータを JPEG ファイルに変換し、Web サーバを通して Web ブラウザ上で表示させます。本アプリケーションノートでは、画像処理システムを通した結果も視認できます。

画像処理システムは、ミドルウェアの保護の為に、RX64M 搭載の TM 機能を使用しています。

TM 機能は、コードフラッシュメモリのブロック 8、9 に対するプログラムの不正リードおよび追加プログラミングを防止します。詳細は、RX64M グループのユーザーズマニュアルでご確認ください。

<http://japan.renesas.com/products/mpumcu/rx/rx600/rx64m/index.jsp>

本アプリケーションノートは、GR-KAEDE 上で動作します。

本アプリケーションノートでは、カメラ活用センサソリューションを動作させる為の手順について説明します。

## 1.2 動作環境

本アプリケーションノートの動作環境を以下に示します。

表 1.2.1 動作環境

対応MCU	RX64Mグループ
評価ボード	GR-KAEDE 他に、カメラモジュール、ACアダプタ、LANケーブルが必要です。  購入方法などについては、下記URL参照。 <a href="http://gadget.renesas.com/ja/product/kaede.html">http://gadget.renesas.com/ja/product/kaede.html</a>
統合開発環境 (IDE)	e <sup>2</sup> studio V4.0.1.07以降 または、 CS+ V3.01.00以降
クロスツール	RXファミリ用C/C++コンパイラパッケージV2.03.00以降
エミュレータ	E1 (別途購入必要)
RX Driver Package	RX64M用RX Driver Package Ver1.00 (R01AN2144JJ0100) ※
Webブラウザ	Internet Explorer8 Internet Explorer11(互換表示設定で「192.168.0.3」を追加してください)

※ 本アプリケーションノートは、RX Driver Package Ver1.00 をベースに、FIT モジュールを更新しています。適用バージョンは、「1.3 節 モジュール構成」でご確認ください。

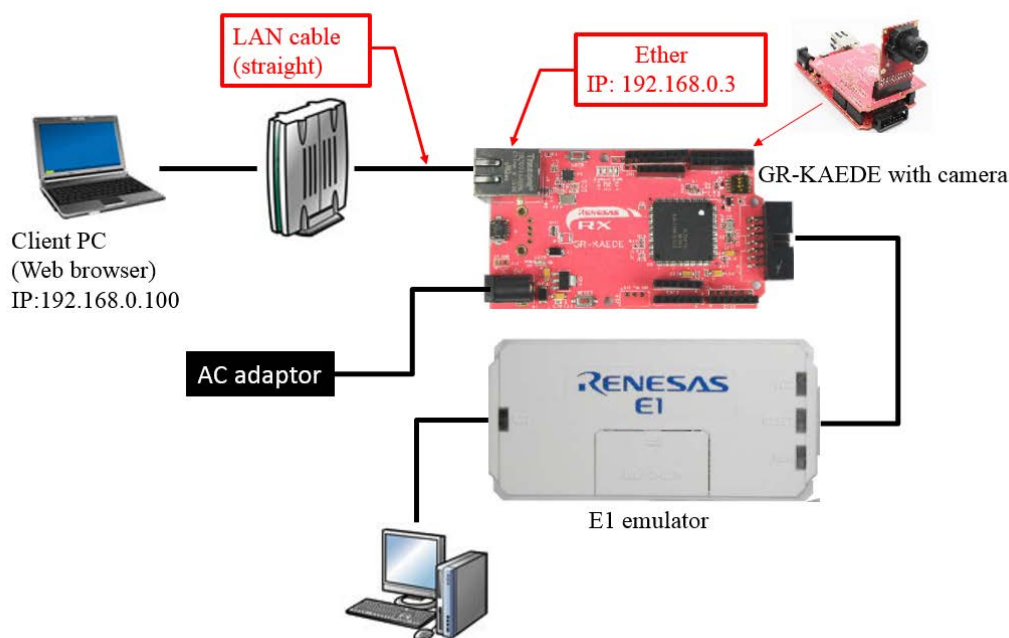


図 1.2.1 動作環境の例

### 1.3 モジュール構成

本アプリケーションノートのモジュール構成を以下に示します。

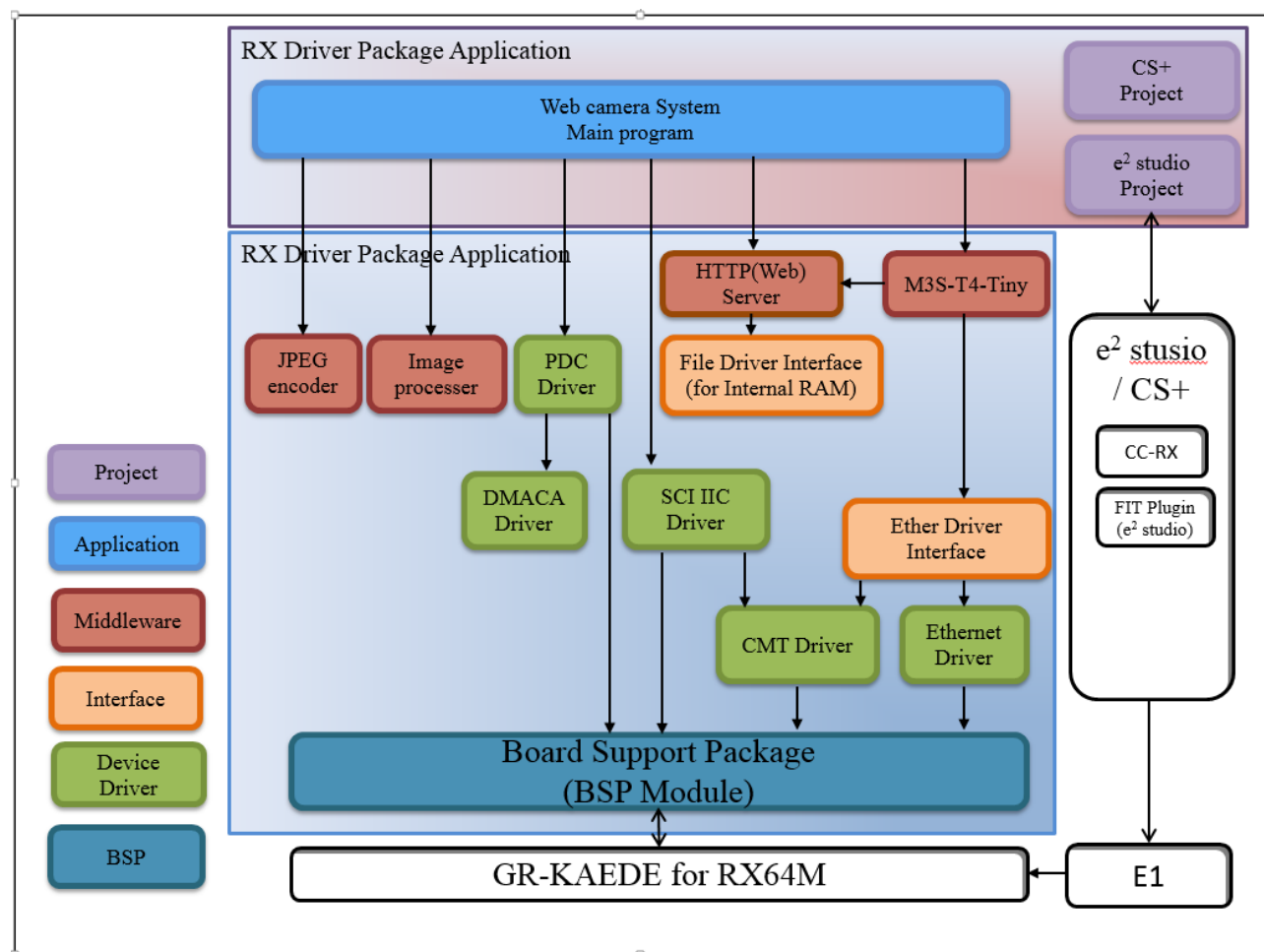


図 1.3.1 モジュール構成

モジュール構成を以下に示します。

表 1.3.1 モジュール一覧

種類	モジュール名	FITモジュール名	バージョン
Board Support Package	ボードサポートパッケージ (BSPモジュール)	r_bsp	2.80
Device Driver	コンペアマッチタイマ (CMT)	r_cmt_rx	2.41
Device Driver	イーサネットコントローラ (ETHERC)	r_ether_rx	1.01
Device Driver	DMA コントローラ (DMAC)	r_dmaca_rx	1.02
Device Driver	SCI 簡易 IIC ドライバ	r_sci_iic_rx	1.50
Device Driver	パラレルデータキャプチャ (PDC)	r_pdc_rx	1.01
Middleware	JPEG エンコーダ	r_jpege_rx	1.00
Middleware (Interface)	M3S-T4-Tiny インタフェース変換モジュール	r_t4_driver_rx	1.02
Middleware (Interface)	FTP/Webサーバ用ファイルドライバモジュール	r_t4_file_driver_rx	1.01
Middleware	TCP/IPプロトコルスタックライブラリ (M3S-T4-Tiny)	r_t4_rx	2.03
Application	Webサーバ	r_t4_http_server_rx	1.04
Application	Webサーバシステムメインプログラム	r_httpd_main_rx64m	1.00 ベースとして使用。

※ 各 FIT モジュールの機能説明は、モジュール内の”doc”フォルダに置かれています。

表 1.3.2 FIT 対象外モジュール一覧(renesas\_sw\_library フォルダ内)

種類	モジュール名	フォルダ名	バージョン
Middleware	画像処理ミドルウェア (株式会社 日立産業制御ソリューションズ作成)	image_proc※	1.00

※主要処理は、TM 領域に格納されています。デバッガなどから参照できません。

## 1.4 プロジェクトについて

本アプリケーションノートには、センサソリューションをビルド及び評価するための、e<sup>2</sup> studio 及び CS+ 用のプロジェクトが付属しています。プロジェクトには、ビルド設定を保存した「ビルド構成 (CS+ではビルド・モード)」と、デバッグ設定を保存した「デバッグ構成 (CS+ではデバッグ・ツール)」を登録しています。

以下に、プロジェクトに登録してあるビルド構成とデバッグ構成の一覧を示します。

表 1.4.1 プロジェクト設定

	構成名	説明
ビルド構成 ※CS+では「ビルド・モード」	HardwareDebug (Debug on hardware)	デバッグ情報付きのロードモジュールを生成するための構成です。 <b>■主な設定</b> <ul style="list-style-type: none"><li>• デバッグ情報あり</li><li>• 最適化あり (-optimize=2)</li></ul>
デバッグ構成 ※CS+では「デバッグ・ツール」	HardwareDebug (E1) ※CS+では「RX E1(JTAG)」	「HardwareDebug (Debug on hardware)」で生成したロードモジュールを使用して、E1 エミュレータ経由でのハードウェアデバッグを行います。

CS+を使用される場合は、「6章 CS+を使用する場合」をご覧ください。



## 2. 開発環境の入手

### 2.1 e<sup>2</sup> studio の入手とインストール方法

e<sup>2</sup> studio は、ルネサスのホームページからダウンロードできます。

- 以下の URL にアクセスし、e<sup>2</sup> studio のダウンロードページを表示します。

[http://japan.renesas.com/e2studio\\_download](http://japan.renesas.com/e2studio_download)

- 表示された項目の中から、「統合開発環境 e<sup>2</sup> studio 4.0.1.007 インストーラ」をクリックします。オフライン用、オンライン用とありますが、どちらでもかまいません。指定より最新のバージョンがある場合は、そちらをご使用ください。

その後表示されたページの指示に従い、e<sup>2</sup> studio のインストーラをダウンロードします。

The screenshot shows the 'e<sup>2</sup> studio' download page. The main content area displays a table of software products. The table has columns for '分類' (Category), 'ソフトウェア名' (Software Name), '登録日' (Registration Date), and '説明' (Description). Two entries for '統合開発環境 e<sup>2</sup> studio 4.0.1.007 インストーラ' are highlighted with a red box. A red arrow points to these entries with the text 'いずれかのリンクをクリックします'.

分類	ソフトウェア名	登録日	説明
ソリューション・ツールキット	【テクニカルプレビュー版】USB特化ツール QE for USB V1.0.0	Jul.21.15	QE for USB は、統合開発環境 e <sup>2</sup> studio 上で動作するソリューション・ツールキットのひとつで、USBシステムの開発に特化した機能をサポートしている組み込みソフト開発ツールです。
統合開発環境 e <sup>2</sup> studio	統合開発環境 e <sup>2</sup> studio 4.0.1.007 インストーラ (オフライン用)	Jun.22.15	Eclipseベースの統合開発環境です。コンパイラは別製品のため、別途インストールが必要です。
統合開発環境 e <sup>2</sup> studio	統合開発環境 e <sup>2</sup> studio 4.0.1.007 インストーラ (オンライン用)	Jun.22.15	Eclipseベースの統合開発環境です。コンパイラは別製品のため、別途インストールが必要です。

- ダウンロードした e<sup>2</sup> studio のインストーラを実行し、e<sup>2</sup> studio を PC にインストールします。

インストール方法は「e<sup>2</sup> studio 統合開発環境 ユーザーズマニュアル入門ガイド」を参照してください。

[http://documentation.renesas.com/doc/products/tool/doc/r20ut2858jj0300\\_e2\\_start\\_s.pdf](http://documentation.renesas.com/doc/products/tool/doc/r20ut2858jj0300_e2_start_s.pdf)

## 2.2 コンパイラパッケージの入手方法

本ファームウェアをビルドするには、RX ファミリー用 C/C++コンパイラパッケージ V2.03.00 以降が必要です。お持ちでない場合は、無償評価版の「RX ファミリー用 C/C++コンパイラパッケージ (統合開発環境なし)」をインストールします。

1. 以下の URL にアクセスし、e<sup>2</sup> studio のダウンロードページを表示します。

[http://japan.renesas.com/e2studio\\_download](http://japan.renesas.com/e2studio_download)

2. 示された項目の中から、「【無償評価版】RX ファミリー用 C/C++コンパイラパッケージ V2 (統合開発環境なし) V2.03.00」をクリックします。

その後表示されたページの指示に従い、コンパイラのインストーラをダウンロードします。

分類	ソフトウェア名	登録日	説明
統合開発環境 e <sup>2</sup> studio	統合開発環境 e <sup>2</sup> studio 4.0.0.26 インストーラ (オフライン用)	Jun.05.15	Eclipseベースの統合開発環境です。コンパイラは別製品のため、別途インストールが必要です。
統合開発環境 e <sup>2</sup> studio	【無償評価版】RX ファミリー用C/C++コンパイラパッケージ V2 (統合開発環境なし) V2.03.00	Feb.05.15	コンパイラ、アセンブラ、リンクを含むコンパイラパッケージ(統合開発環境、シミュレータはパッケージに含まれません)

3. ダウンロードしたコンパイラのインストーラを実行し、コンパイラを PC にインストールします。

## 2.3 GR-KAEDE ボードの入手方法

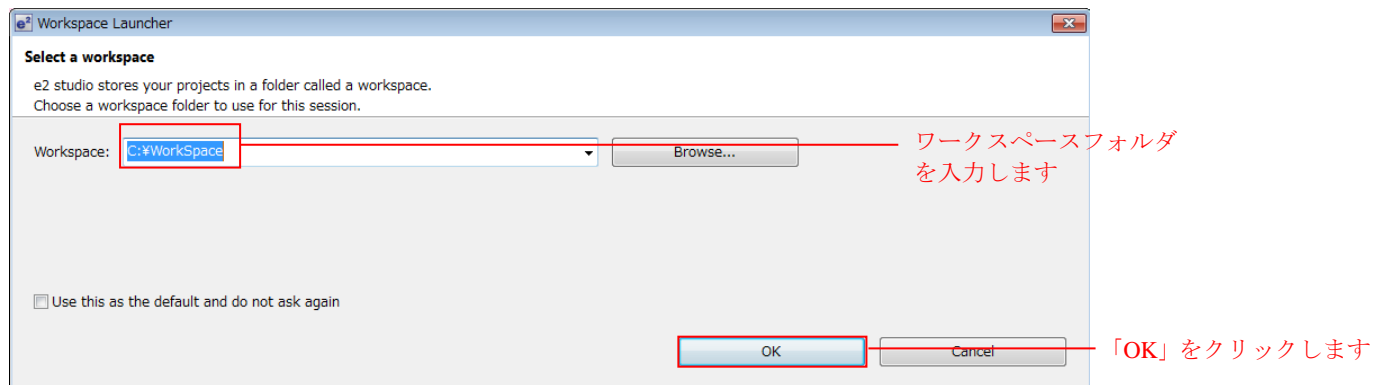
下記 URL に、購入先や GR-KAEDE ボード仕様が記載されています。

<http://gadget.renesas.com/ja/product/kaede.html>

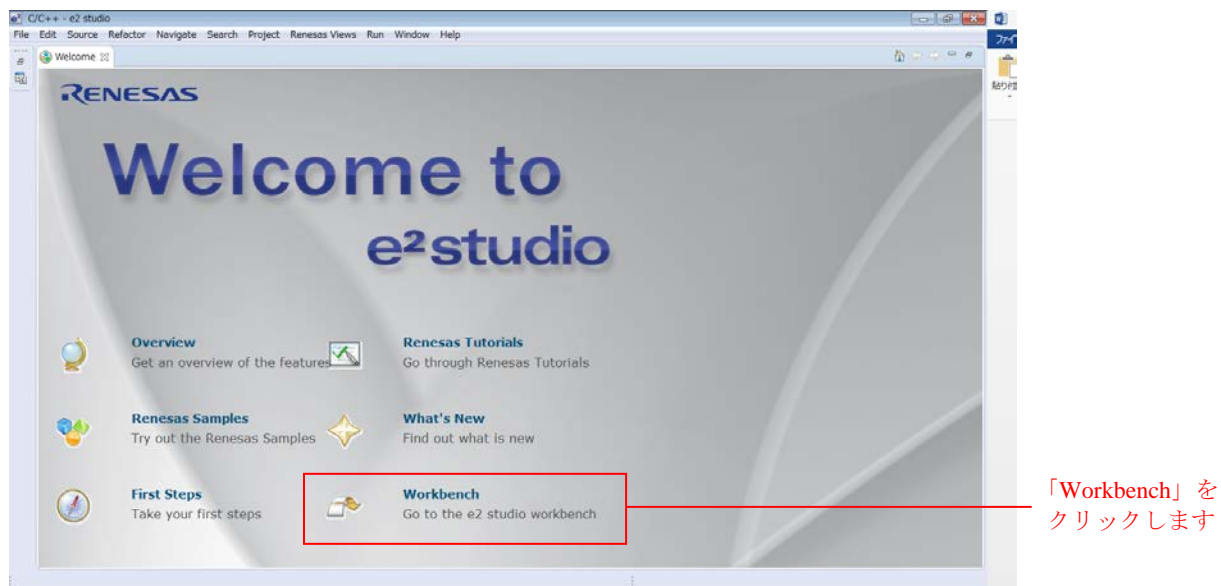
### 3. プロジェクトの構築

#### 3.1 ワークスペースの作成

1. e<sup>2</sup> studio を起動します。
2. 表示されたダイアログに、任意のワークスペースフォルダを入力し、「OK」をクリックします。



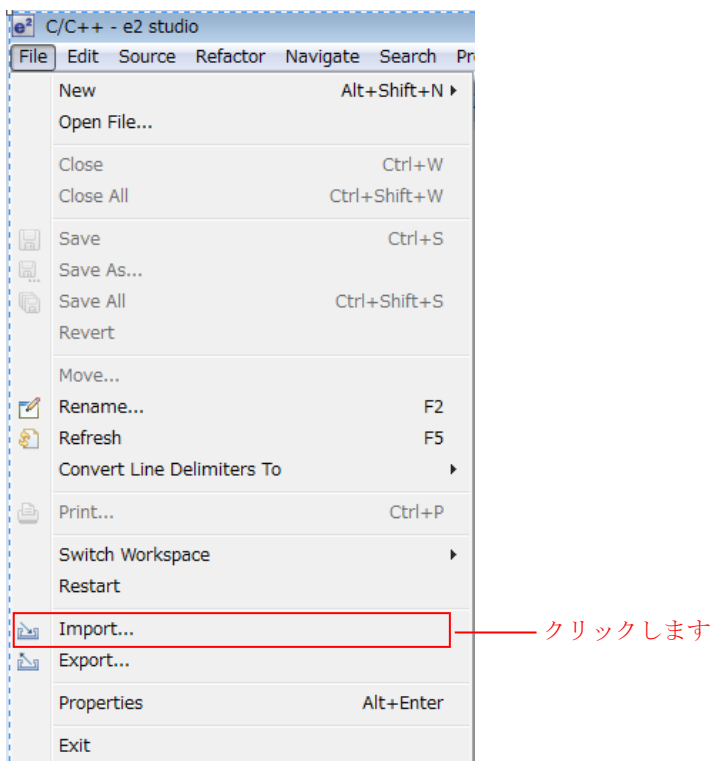
3. 以下の画面が表示されたら、「Workbench」をクリックします。



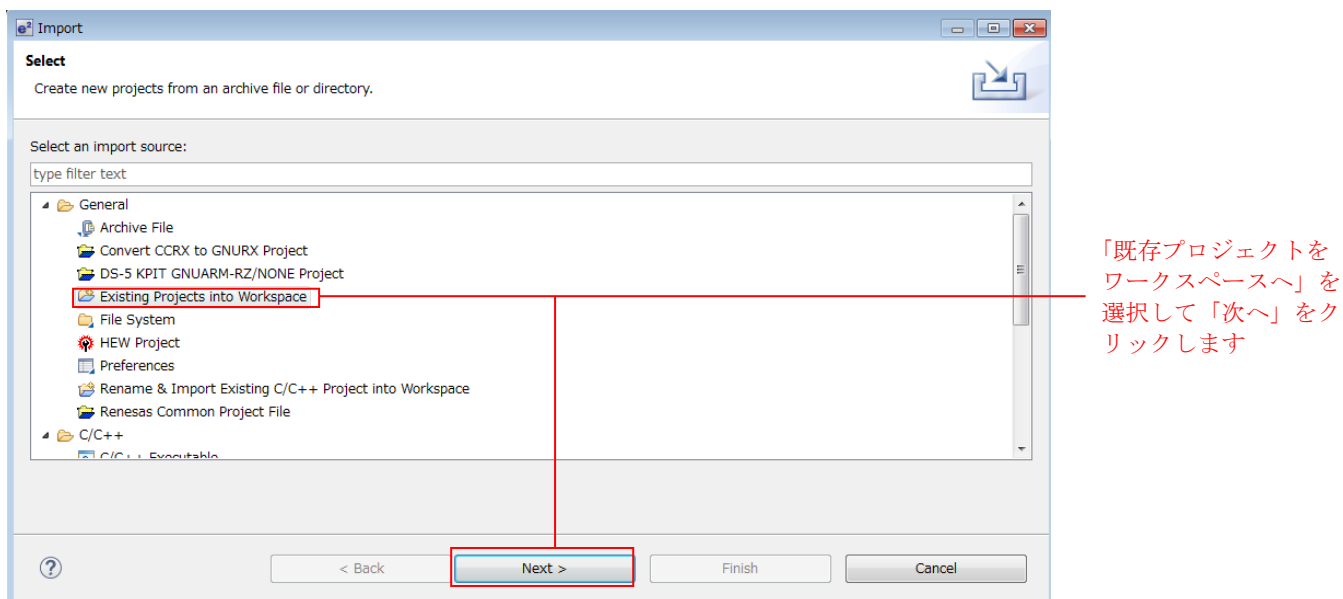
## 3.2 プロジェクトのインポート

本アプリケーションノートに付属しているプロジェクトを、作成したワークスペースにインポートします。

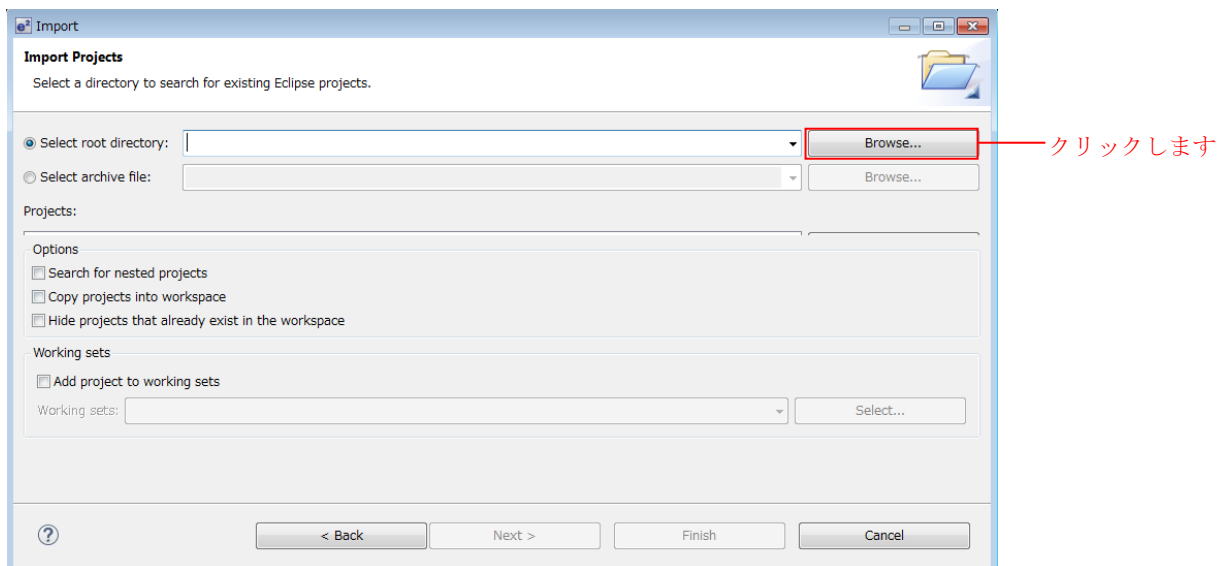
1. e<sup>2</sup> studio の「ファイル」メニューの「インポート」をクリックします。



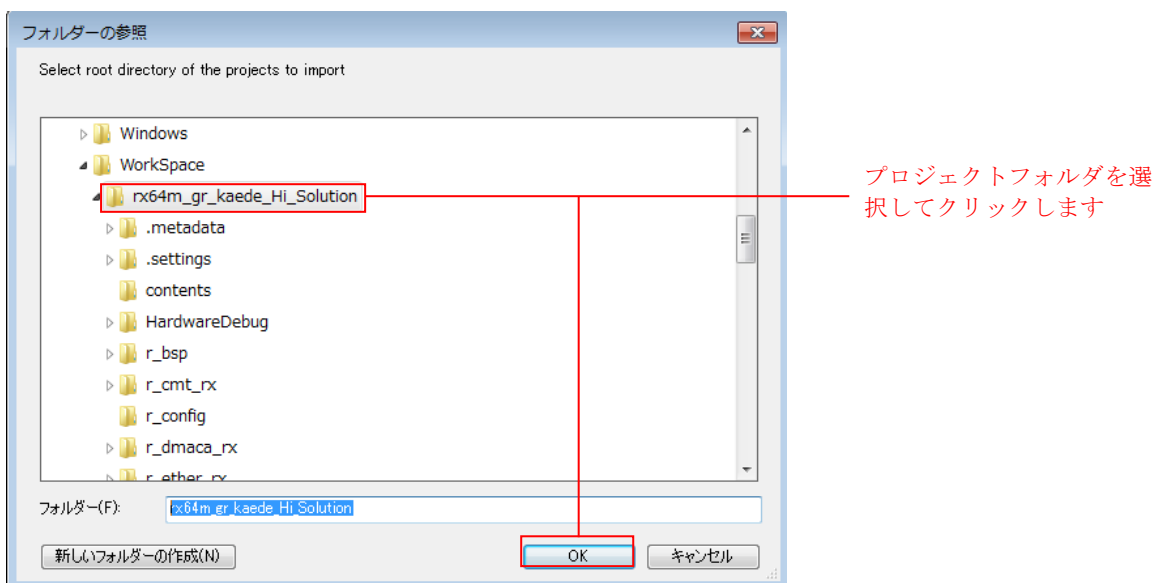
2. 「一般」から「既存プロジェクトをワークスペースへ」を選択し、「次へ」をクリックします。



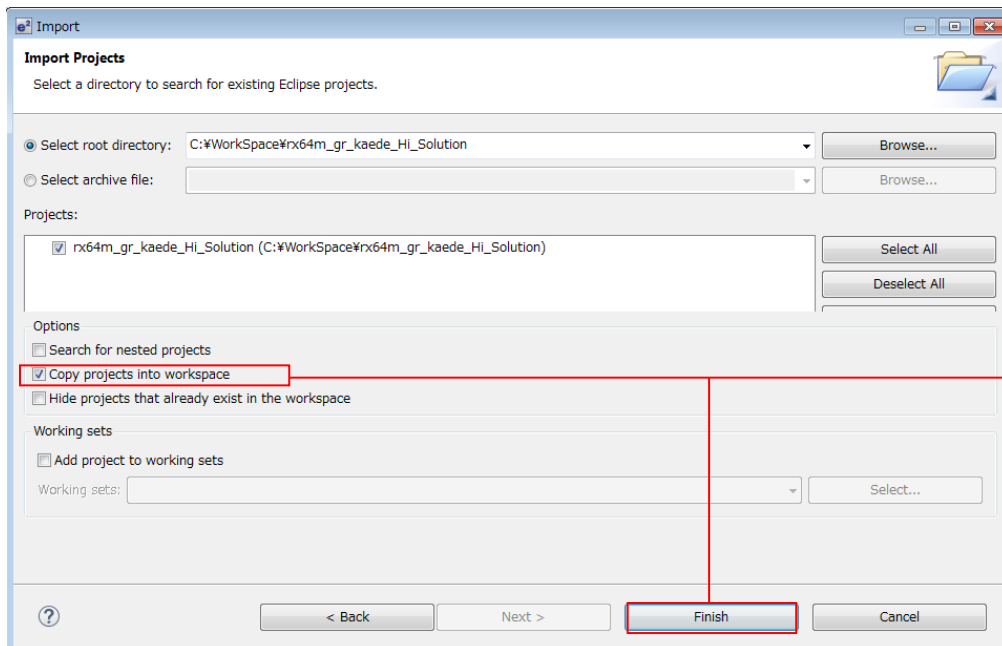
3. 「Browser」 をクリックします。



4. 本アプリケーションノートに付属しているプロジェクトのフォルダ ( ) を選択し、「OK」をクリックします。



5. 「プロジェクトをワークスペースにコピー」にチェックを入れ、「完了」をクリックします。



チェックを入れ、  
クリックします

※ワークスペースへのコピーは必須ではありません。チェックしない場合は、ルートディレクトリ内がビルド対象になります。

## 4. 動作確認

### 4.1 GR-KAEDE ボードの Dip Switch 確認

GR-KAEDE の Dip Switch の状態を確認します。

下記図 4.1.1 の設定にあわせてください。

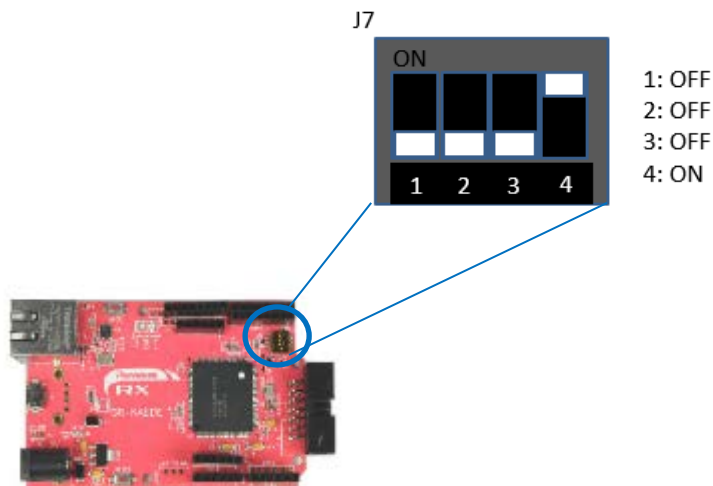
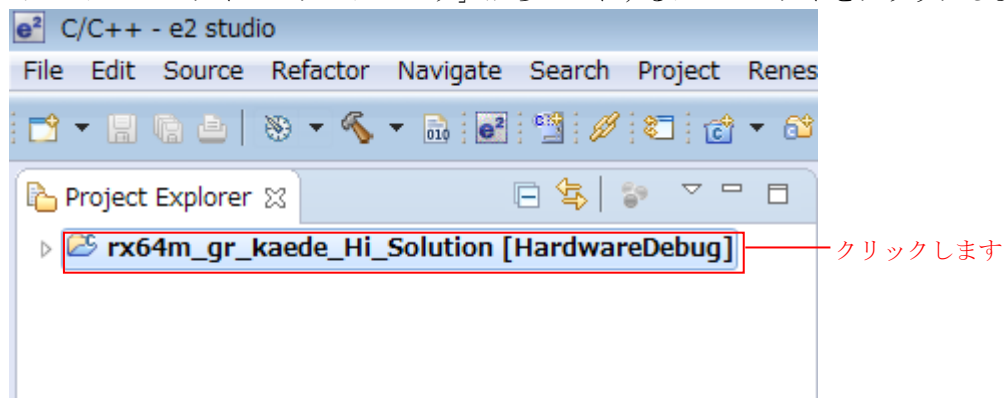


図 4.1.1 Dip Switch 設定

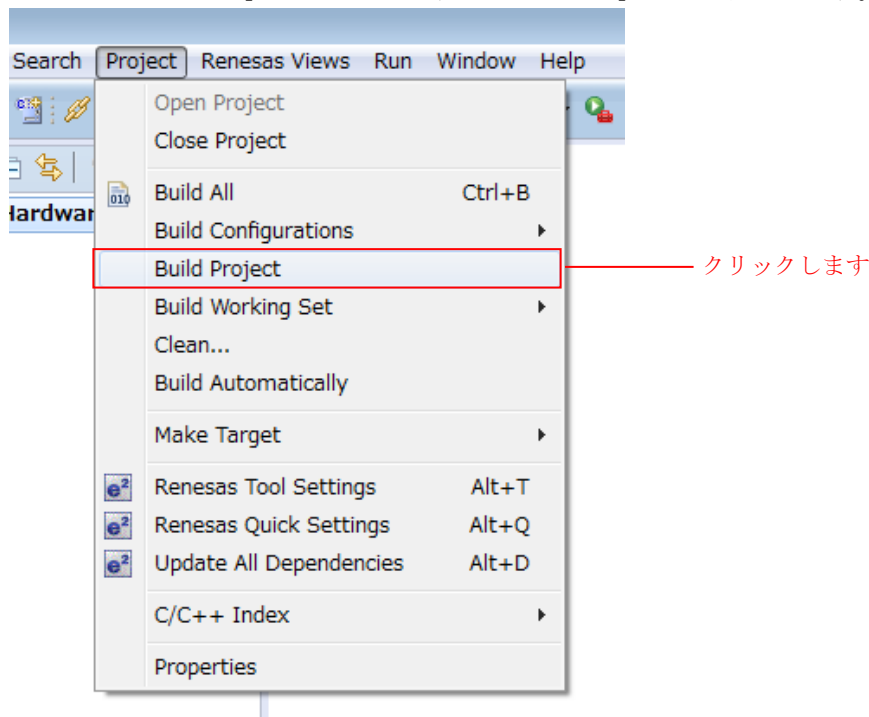
### 4.2 プロジェクトのビルド

以下の手順に従い、プロジェクトをビルドしてロードモジュールを生成します。

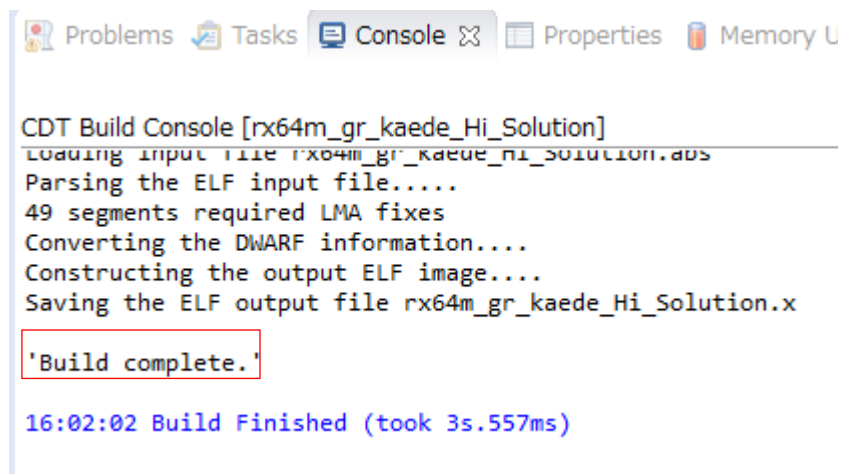
1. 「プロジェクト・エクスプローラ」からビルドするプロジェクトをクリックします。



2. 「プロジェクト」メニューの「すべてをビルド」をクリックします。



3. 「コンソールパネル」に「Build complete.」と表示されたらビルド完了です。





### 4.3 デバッグの準備

#### 4.3.1 機器の構成

デバッグを開始する前に、評価ボードを準備します。

必要な機器の一覧と構成を以下に示します。

表 4.3.1.1 機器構成

No.	機器	補足
1	開発 PC	開発を行う PC です。
2	GR-KAEDE	
3	クライアント PC (Web ブラウザ)	開発 PC で代用可能です。
4	クライアント PC と GR-KAEDE (Web サーバ) を接続するためのネットワーク環境として以下のいずれか 1. スイッチング・ハブを使用する場合 a. スイッチング・ハブ b. LAN ケーブル (ストレート) 2 本 2. クロスケーブルを使用する場合 a. LAN ケーブル (クロス) 1 本	

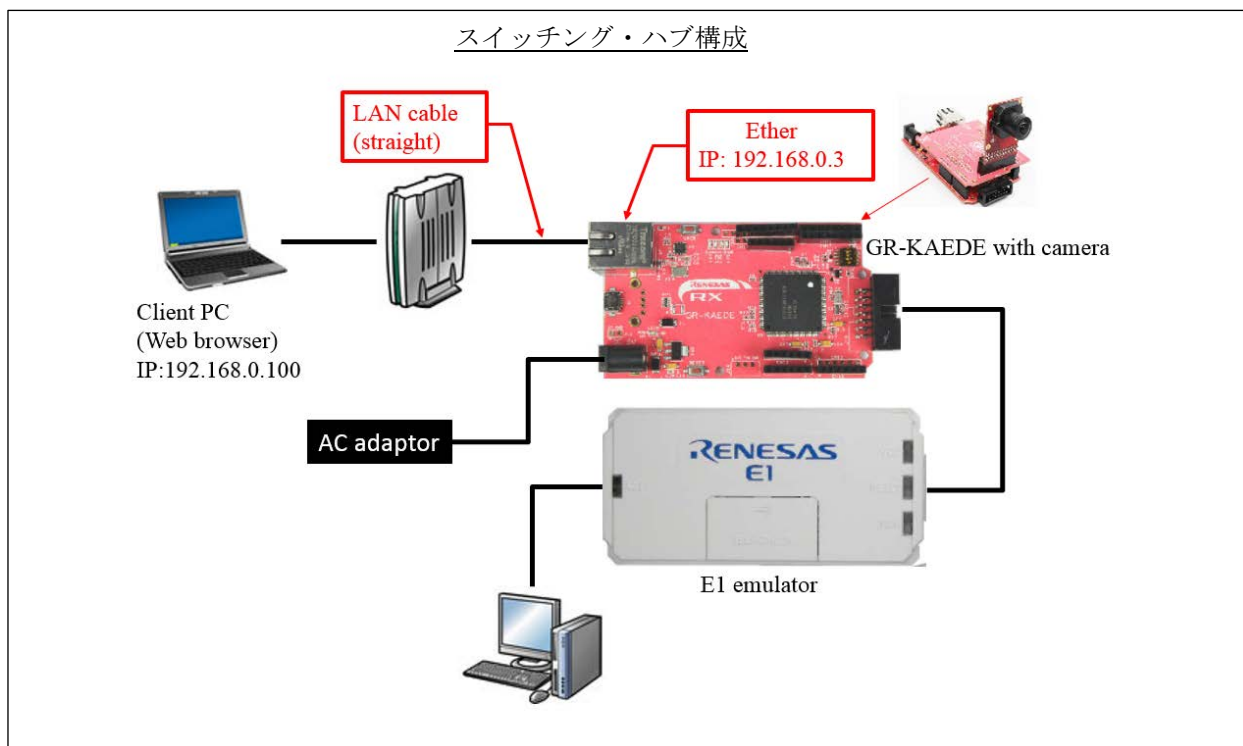


図 4.3.1.1 スイッチング・ハブ構成

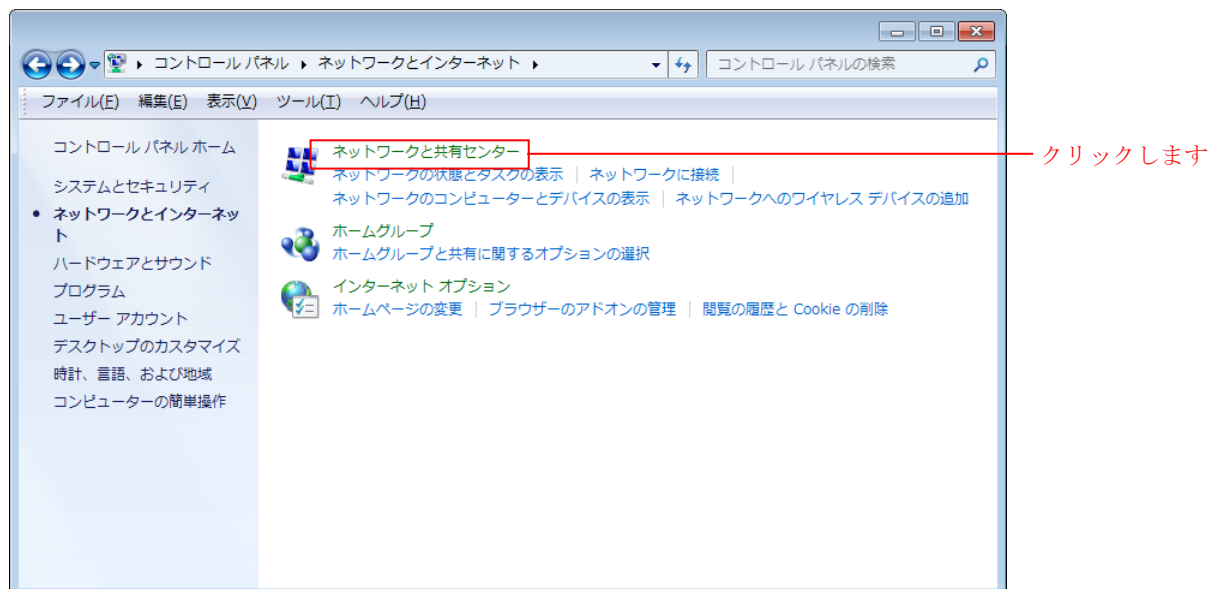
### 4.3.2 クライアント PC の設定

クライアント PC のネットワークを設定します。

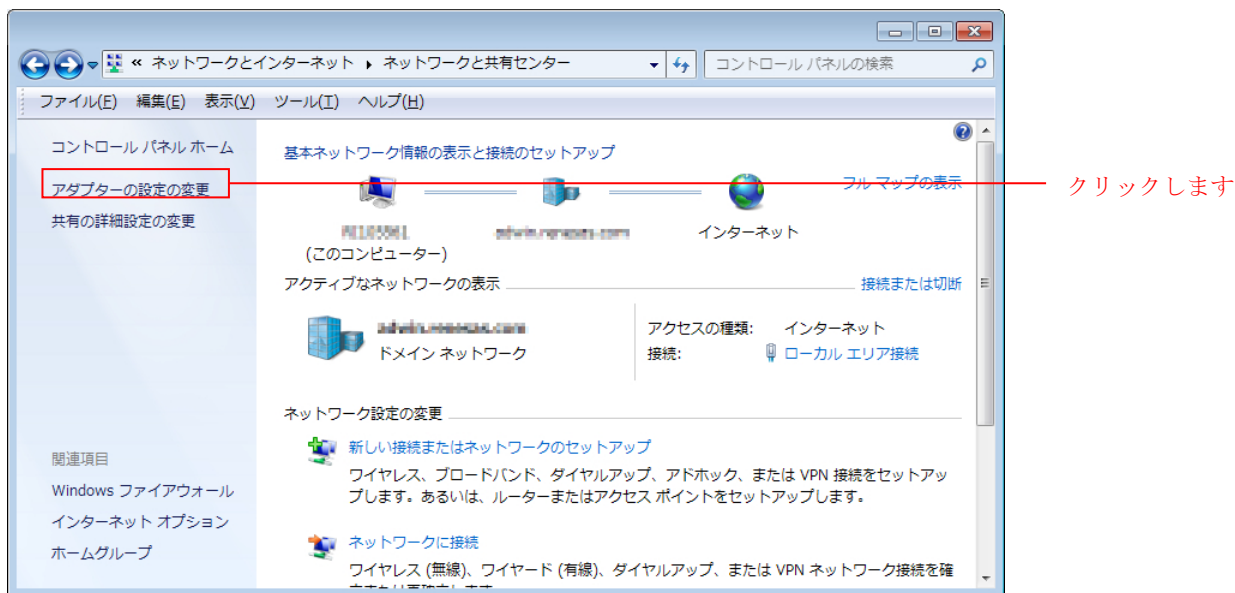
1. クライアント PC の「コントロールパネル」を開き、「ネットワークとインターネット」をクリックします。



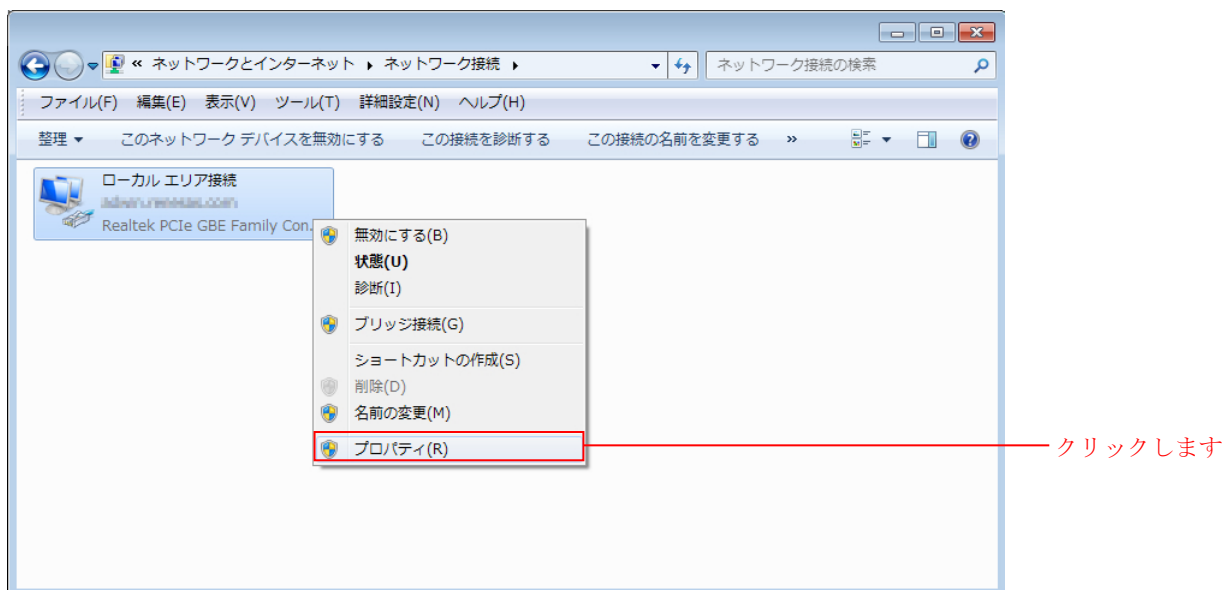
2. 「ネットワークと共有センター」をクリックします。



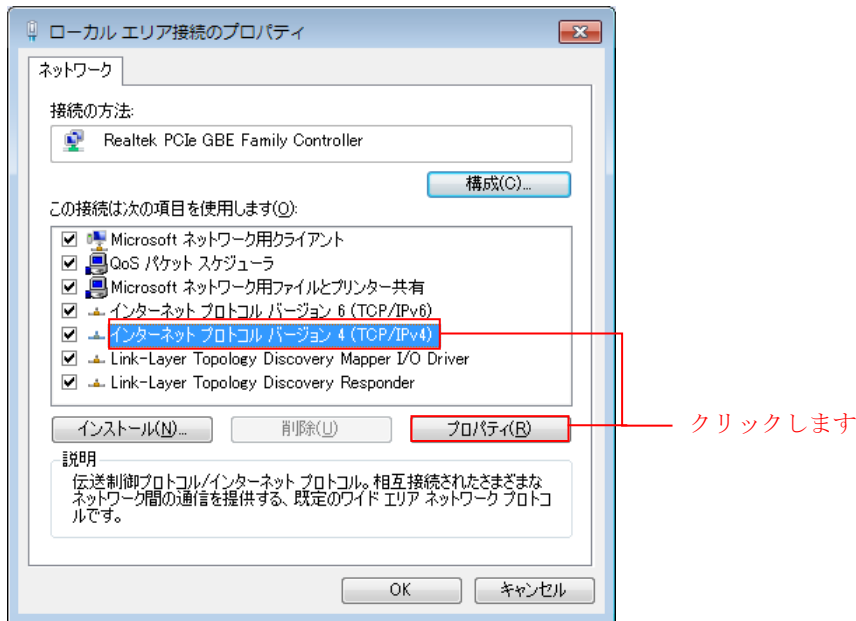
3. 「アダプターの設定の変更」をクリックします。



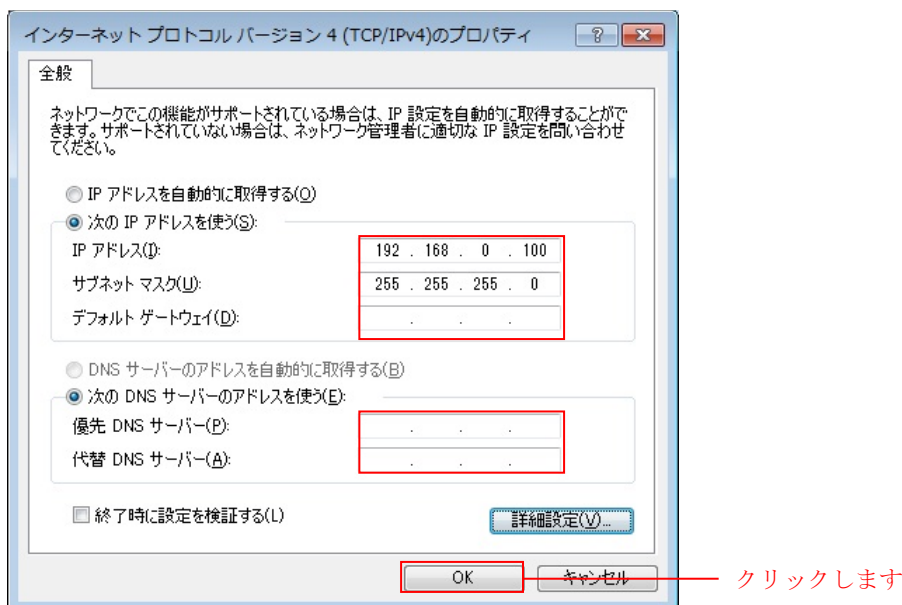
4. 「ローカル エリア接続」を右クリックして、プロパティをクリックします。



5. 「インターネットプロトコルバージョン 4 (TCP/IPv4)」を選択して「プロパティ」をクリックします。



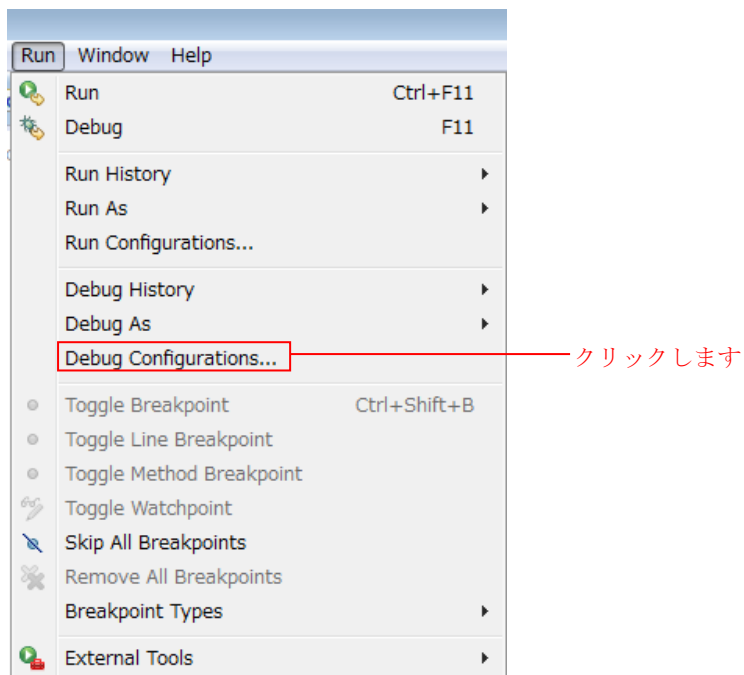
6. IP アドレス等の設定情報が表示されます。以下のように設定し「OK」をクリックします。



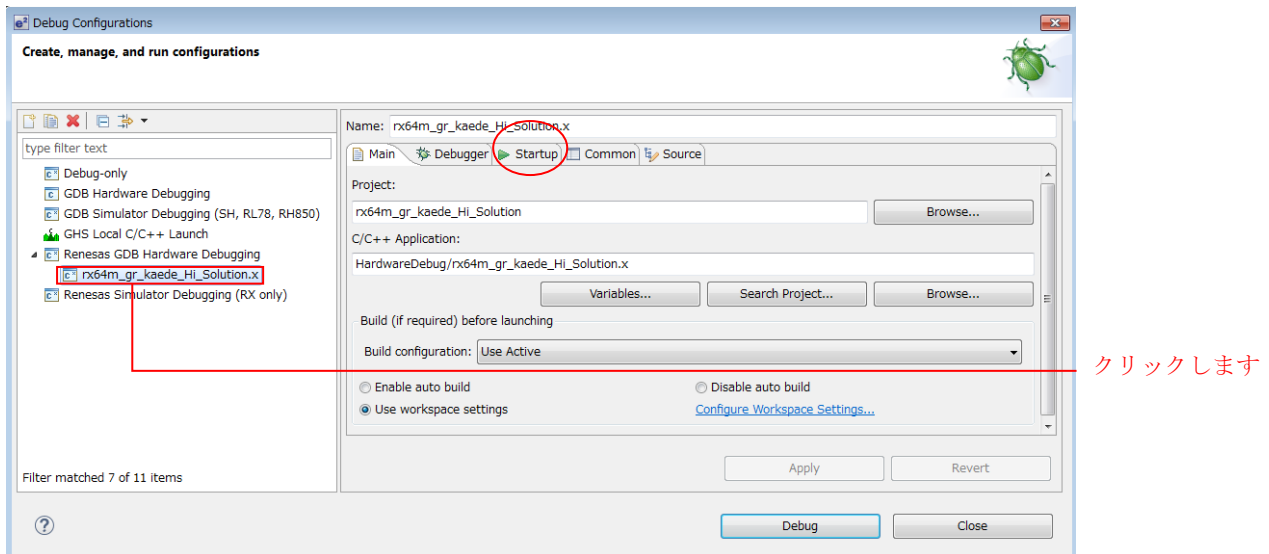
#### 4.4 プロジェクトのデバッグ

以下の手順に従い、プロジェクトのデバッグを開始します。

1. 開発 PC と E1 エミュレータを USB ケーブルで接続します。
2. GR-KAEDE にアダプタを接続し、電源を入れます。
3. e<sup>2</sup> studio の「実行」メニューの「デバッグ構成」をクリックします。

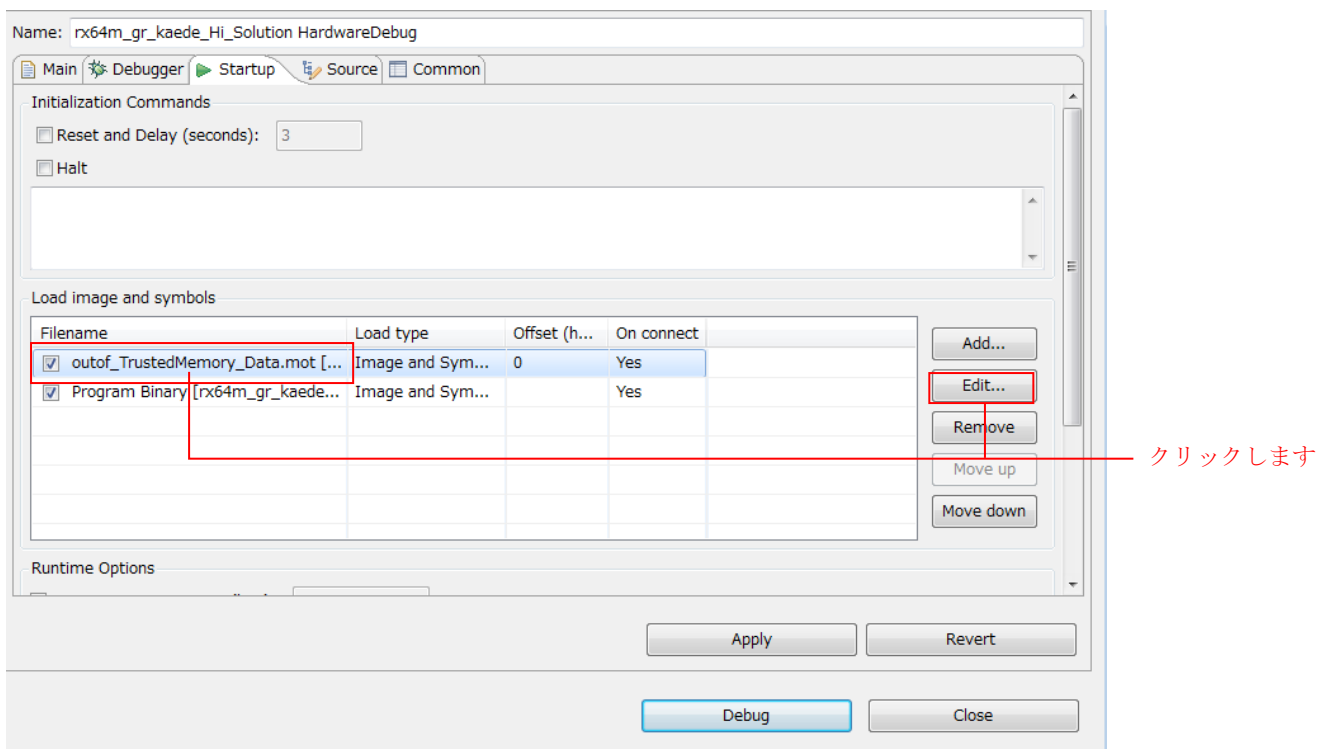


4. 「Renesas GDB Hardware Debugging」の「rx64m\_gr\_kaede\_Hi\_Solution HardwareDebug」をクリックします。



5. 「Startup」タブをクリックします。

6. 「outof\_TrustedMemory\_Data.mot」をクリックし、「Edit」をクリックします。

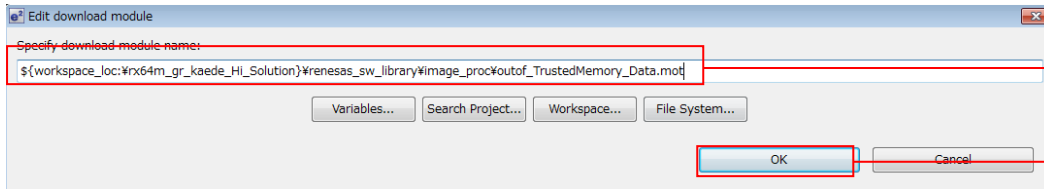


7. ファイルの格納先を変更します。

「`{workspace_loc:¥rx64m_gr_kaede_HI_Solution}¥renesas_sw_library¥image_proc¥outof_TrustedMemory_Data.a.mot`」

と入力してください。

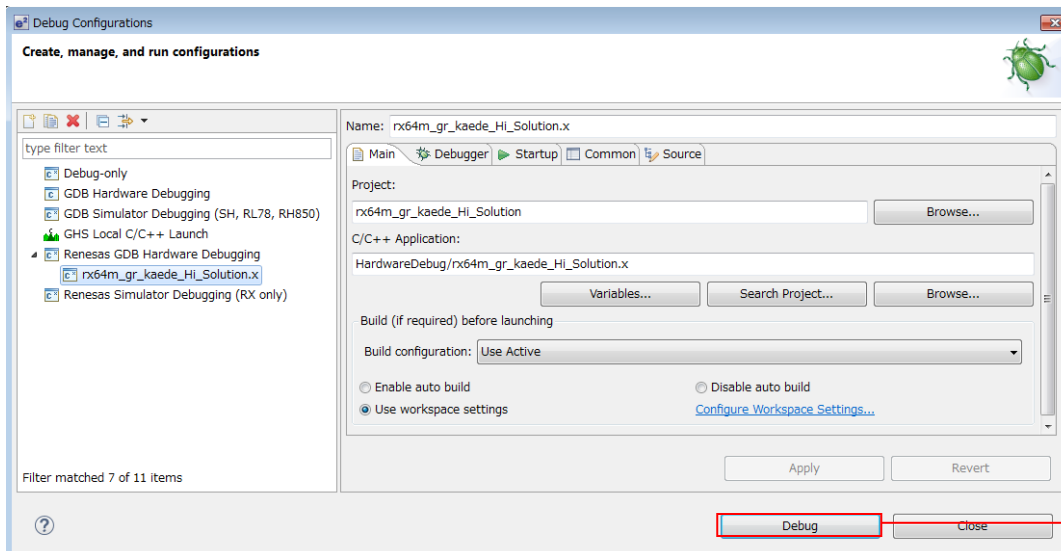
「OK」をクリックします。



入力します

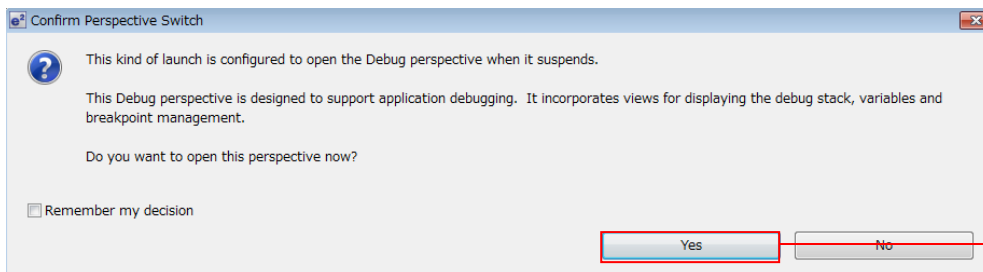
クリックします

8. 「デバッグ」をクリックします。



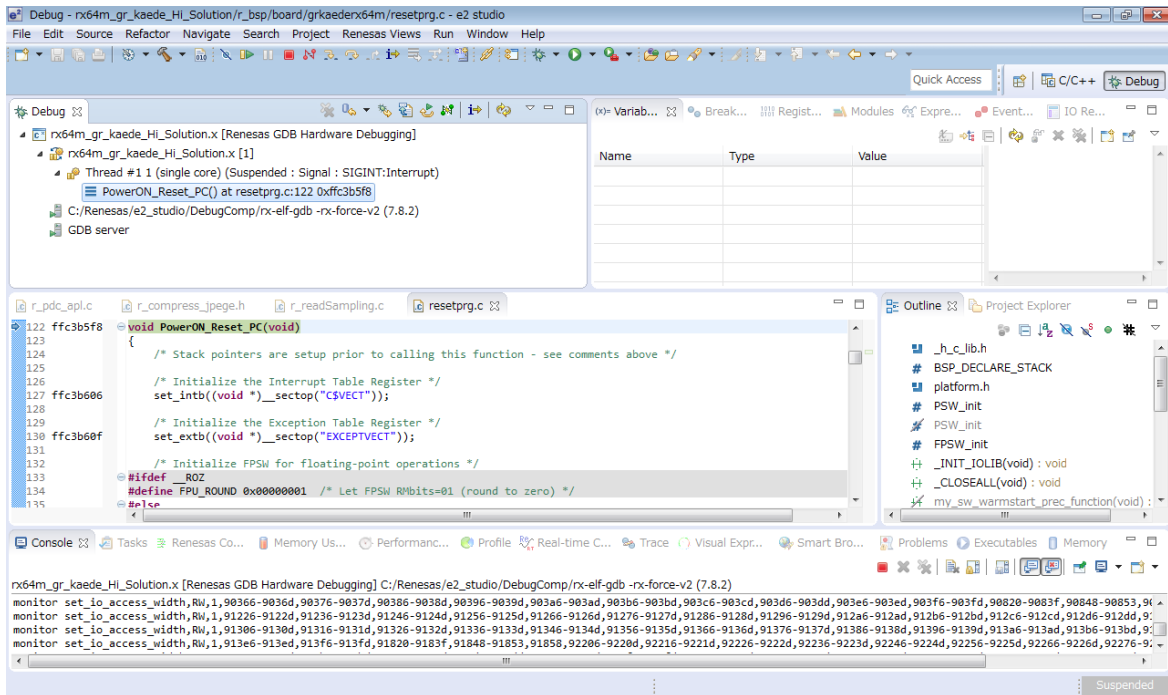
クリックします

以下のメッセージが表示されたら、「はい」をクリックします。

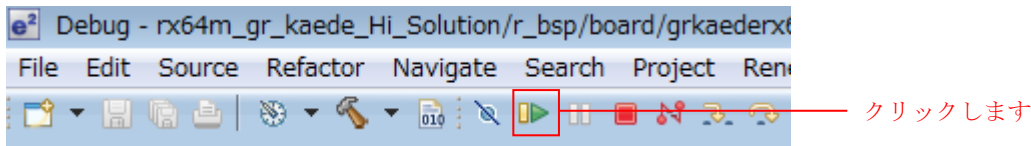


クリックします

ロードモジュールのダウンロードが完了すると、「デバッグ」パースペクティブが開きます。



9. ツールバーの「再開」をクリックします。プログラムが実行され、main 関数の先頭でブレークします。



main 関数の先頭でブレークした後に、もう一度ツールバーの「再開」をクリックします。

※ main 関数先頭でのブレークは、e<sup>2</sup> studio のデバック構成で解除可能です。

10. クライアント PC で Web ブラウザを起動し、以下のアドレスを入力します。

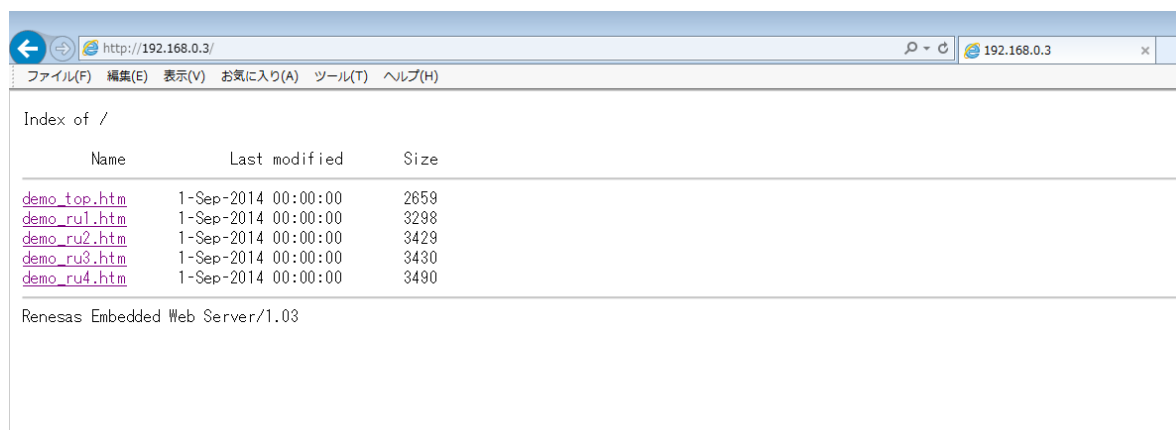
Web サーバアドレス : <http://192.168.0.3>

【注】 Web サーバアドレスは、コンフィギュレーションで変更可能です。

(rx64m\_gr\_kaede\_Hi\_Solution /r\_t4\_rx/src/config\_tcpudp.c )

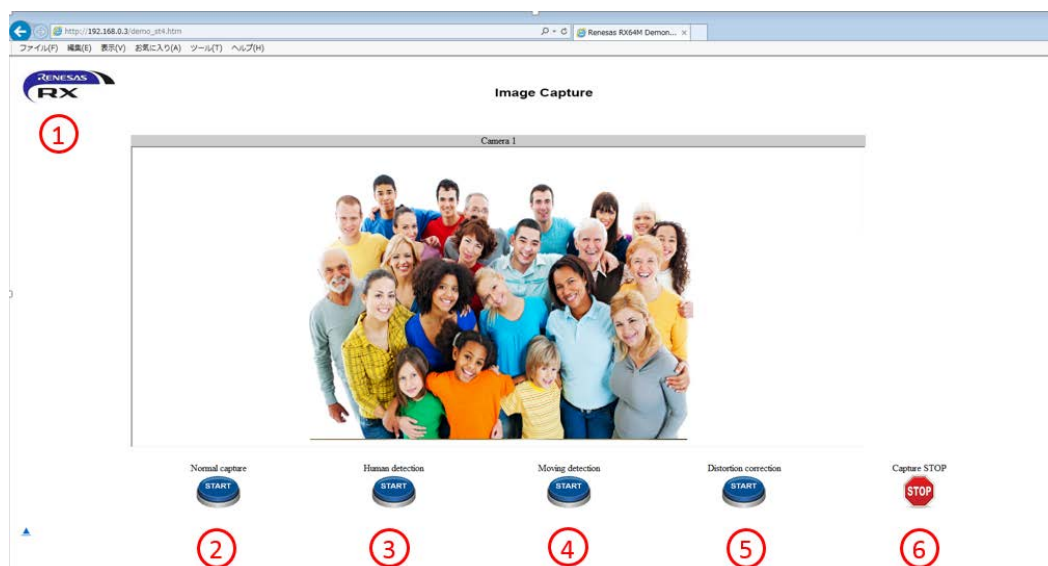


11. Web ブラウザ上に、ファイルの一覧が表示されます。



12-1. 11.の画面で「demo\_top.htm」をクリックすると以下のような画面が表示されます。

[デモトップ画面]



上記各番号の上に置かれたビットマップをクリックすると下記動作をします。

1. 当画面をリロード。
2. キャプチャ画面(ノーマル)へ遷移。
3. キャプチャ画面(人物検知)へ遷移。
4. キャプチャ画面(動体検知)へ遷移。
5. キャプチャ画面(歪み補正調整画面)へ遷移。
6. 当画面へ遷移(リロード相当)。

12-2. 11.の画面で「demo\_ru1.htm」をクリックすると以下のような画面が表示されます。

[キャプチャ画面(ノーマル)]



上記各番号の上に置かれたビットマップをクリックすると下記動作をします。

1. デモトップ画面へ遷移。
2. 当画面を再実行
3. キャプチャ画面(人物検知)へ遷移。
4. キャプチャ画面(動体検知)へ遷移。
5. キャプチャ画面(歪み補正調整画面)へ遷移。
6. ノーマルモードを維持したままキャプチャ更新停止。

この画面では2つ機能のパラメータ変更が可能です。  
上段は、現在値(設定後の結果)を表します。

- ・ JPEG QUALITY : JPEG エンコード時の品質 1(最低画質)~128(最高画質)
- ・ IMAG SENSOR Read DT: カメラモジュール内のイメージセンサのレジスタ値を表示します。下段で書き込んだレジスタに対するリード結果を表示します。(一部 Write only のレジスタもあります)

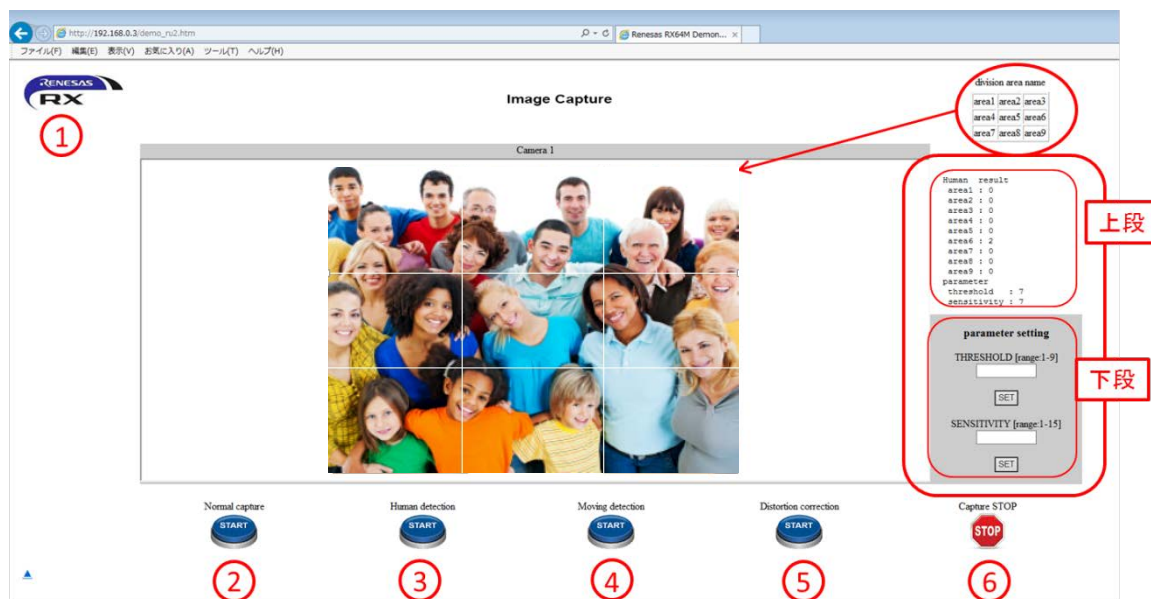
下段では、各パラメータを更新できます。

空白欄に値を記載後、”SET”ボタンをクリックしてください。(一旦ページが切り替わるのは、Web サーバ FIT モジュールの仕様です。)

IMAGE SENSOR のレジスタ情報については、本アプリケーションノートでは公開していません。使用される場合は、使用者の責任でご使用ください。

12-3. 11.の画面で「demo\_ru2.htm」をクリックすると以下のような画面が表示されます。

[キャプチャ画面(人物検知)]



上記各番号の上に置かれたビットマップをクリックすると下記動作をします。

1. デモトップ画面へ遷移。
2. キャプチャ画面(ノーマル)へ遷移。
3. 当画面を再実行。
4. キャプチャ画面(動体検知)へ遷移。
5. キャプチャ画面(歪み補正調整画面)へ遷移。
6. 人物検知モードを維持したままキャプチャ更新停止。

この画面では、人物検知結果の表示と2つ機能のパラメータ変更が可能です。

上段は、検知結果とパラメータの現在値(設定後の結果)を表します。

・Human result : 人物検知結果です。1画面を9エリアに分割し、各エリアでの検出結果を表示します。エリアをまたぐ場合はいずれか1エリアのみにカウントされます。

キャプチャデータに対する各エリアの割り当てはWEBブラウザ右上部”division area name”の通りです。

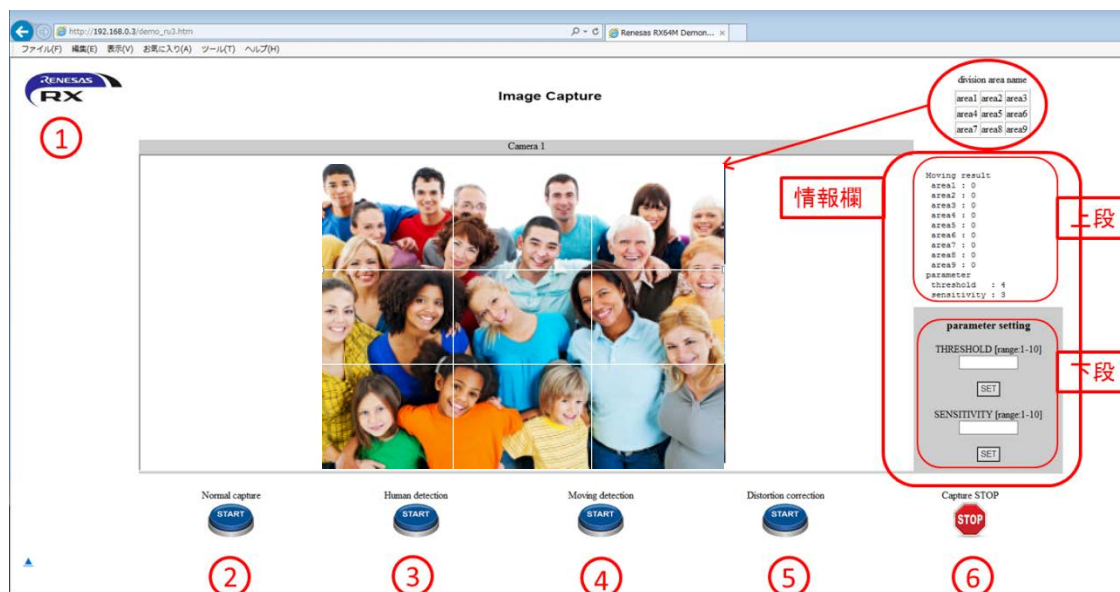
・THRESHOLD : 人物を検出する際の人物らしさを判別するしきい値です。  
 ・sensitivity : 人物を検出する際、検出候補箇所のコントラスト(明暗)及び鮮やかさ(色数)を判別するしきい値です。

下段では、THRESHOLDとSENSITIVITY設定値を変更できます。空白欄に値を記載後、”SET”ボタンをクリックしてください。(一旦ページが切り替わるのは、HTTPサーバFITモジュールの仕様です。)

・THRESHOLD : 人物の検出が行われにくい場合は設定値を小さくし、過検出する場合は設定値を大きくしてください。  
 ・sensitivity : コントラストが小さく、鮮やかさが乏しい箇所を有効する場合は設定値を小さくし、無効とする場合は設定値を大きくしてください。

12-4. 11.の画面で「demo\_ru3.htm」をクリックすると以下のような画面が表示されます。

[キャプチャ画面(動体検知)]



上記各番号の上に置かれたビットマップをクリックすると下記動作をします。

1. デモトップ画面へ遷移。
2. キャプチャ画面(ノーマル)へ遷移。
3. キャプチャ画面(人物検知)へ遷移。
4. 当画面を再実行。
5. キャプチャ画面(歪み補正調整画面)へ遷移。
6. 動体検知モードを維持したままキャプチャ更新停止。

この画面では、動体検知結果の表示と2つ機能のパラメータ変更が可能です。  
上段は、検知結果とパラメータの現在値(設定後の結果)を表します。

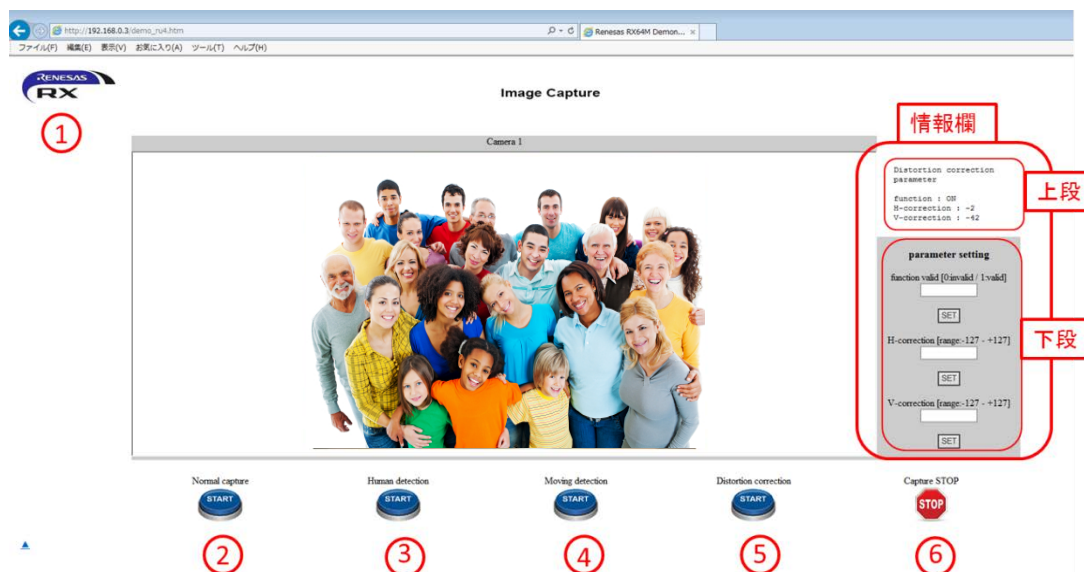
- **Moving result** : 動体検知結果です。1画面を9エリアに分割し、各エリアでの検出結果を表示します。エリアをまたぐ場合はいずれか1エリアのみにカウントされます。  
キャプチャデータに対する各エリアの割り当てはWEBブラウザ右上部”division area name”の通りです。
- **THRESHOLD** : 動体を検出する際の判別しきい値です。
- **sensitivity** : 動体を検出する際、画像差分(1枚目と2枚目、2枚目と3枚目)で明るさの変化があったかを判別するしきい値です。

下段では、**THRESHOLD** と **SENSITIVITY** 設定値を変更できます。空白欄に値を記載後、”SET”ボタンをクリックしてください。(一旦ページが切り替わるのは、WebサーバFITモジュールの仕様です。)

- **THRESHOLD** : 動体の検出が行われにくい場合は設定値を小さくし、過検出する場合は設定値を大きくしてください。
- **sensitivity** : 明るさの変化が小さい箇所を検出したい場合は設定値を小さくし、検出しにくくする場合は設定値を大きくしてください。

12-5. 11.の画面で「demo\_ru4.htm」をクリックすると以下のような画面が表示されます。

[キャプチャ画面(歪み補正調整画面)]



上記各番号の上に置かれたビットマップをクリックすると下記動作をします。

1. デモトップ画面へ遷移。
2. キャプチャ画面(ノーマル)へ遷移。
3. キャプチャ画面(人物検知)へ遷移。
4. キャプチャ画面(動体検知)へ遷移。
5. 当画面を再実行。
6. 歪み補正調整モードを維持したままキャプチャ更新停止。

この画面では、動体検知結果の表示と2つ機能のパラメータ変更が可能です。  
上段は、検知結果とパラメータの現在値(設定後の結果)を表します。

- **function** : 歪み補正機能状態を表示します。
- **H-correction** : 歪み補正を行う際の基準点X座標を画像中心位置からのオフセットで設定します。(水平方向への補正)
- **V-correction** : 歪み補正を行う際の基準点Y座標を画像中心位置からのオフセットで設定します。(垂直方向への補正)

下段では、設定値を変更できます。空白欄に値を記載後、「SET」ボタンをクリックしてください。(一旦ページが切り替わるのは、WebサーバFITモジュールの仕様です。)

- **function** : 歪み補正を有効にする場合は、1を設定してください。
- **H-correction** : 画像中心からのオフセット値を設定します。左方向は負値、右方向は正値
- **V-correction** : 画像中心からのオフセット値を設定します。上方向は負値、下方向は正値

## 5. 画像処理システム仕様

### 5.1 性能概要

本アプリケーションノートの画像処理システムは、株式会社 日立産業制御ソリューションズの画像処理ミドルウェアを使用しています。外付けカメラモジュールからのキャプチャデータを使って、人物検知、動体検知、歪み補正を実行します。本アプリケーションノートでは、Web カメラシステムと組み合わせることによって、検知結果を Web ブラウザ上で視認することができます。さらに、Web サーバの CGI 機能を使うことによって、像処理ミドルウェアのパラメータ調整が Web ブラウザ上から行えます。

表 5.1.1 に示すように、本システムは各メモリの固定領域を参照しています。他の処理で記載領域を使用しないように注意してください。

表 5.1.1 使用領域

項目	性能
内蔵 ROM 使用領域	<アドレス> FFC0 0200h~FFC0 C66Bh FFFE 0000h~FFFE FFFFh(TM 領域)  TM 使用時は、全 TM 領域が書き込み禁止になります。TM 領域として記載した範囲は実際の使用サイズでなく全 TM 領域になります。
内蔵 RAM 使用領域	約 110Kbyte <アドレス> 0000 4100h~0001 F942h
オンボード SDRAM	約 4Mbyte <アドレス(概ね)> 0800 0000h~0840 0000h  <詳細> <ul style="list-style-type: none"> <li>・キャプチャ領域(カメラモジュールからのデータ) 640(縦)x480(横) YCbCr422 : 3 面分</li> <li>・画像処理入出力用 入力 320(縦)x240(横) グレースケール/RGB565 兼用 : 3 面分 出力 320(縦)x240(横) RGB565: 3 面分</li> <li>・画像型式変換(YCbCr422→RGB565 変換)用ワーク領域 640(縦)x480(横) : 1 面分</li> <li>・画像処理ミドルウェア用ワーク領域 320(縦)x240(横) : 4 面分、160(縦)x120(横) : 2 面分</li> </ul>

## 5.2 動作概要

表 5.2.1 機能一覧

機能	機能名称	概要
人物検知機能	人物切出し機能	画像の中から立っている人、及び座っている人が居る位置を切出す。
	人数計測機能	画像を9エリアに分割し、各エリアに存在する人数を計測する。(各エリア最大5人)
動体検知機能	画像比較機能	連続して取り込んだ3枚の画像をそれぞれ比較し、変化があった箇所を抽出する。
歪み補正機能	画像補正機能	使用するカメラとレンズの組合せで求めた歪み特性データを用いて、画像の座標変換を行う。

## 5.2.1 人物検知機能

カメラで撮影した画像から人物（立っている人、座っている人）を検出します。入力画像を9分割し、エリア毎に何人いるかを最大5人まで計測します。人物が重なって映っている場合は、1人毎の検出、及び人数計測は対象外となり、検出結果としては0名からN名の検出誤差が発生します。

表 5.2.1.1 人物検知仕様一覧

項目	データ
カメラ設置高さ	2.5[m]
カメラ設置角度	22[°]
カメラ視野角	130[°]
検知奥行き距離	最大感度 4[m]
画像サイズ	320(横) x 240(縦) [画素]
画像形式	グレースケール,RAW

本アプリケーションノートでは、「4.4 プロジェクトのデバッグ」の12-3. [キャプチャ画面(人物検知)]で確認することができます。

### 5.2.2 動体検知機能

連続して撮影した3枚の画像に対して画像差分を行い、しきい値以上の変化があった箇所を動体として検出します、画像を9分割し、検出した箇所の面積が大きい順に最大5箇所の位置情報を出力します。

表 5.2.2.1 動体検知入力画像仕様一覧

項目	データ
画像サイズ	320(横) x 240(縦) [画素]
画像形式	グレースケール,RAW
入力画像枚数	3(連続して撮影した画像)

本アプリケーションノートでは、「4.4 プロジェクトのデバッグ」の12-4 [キャプチャ画面(動体検知)]で確認することができます。

### 5.2.3 歪み補正機能

入力した画像をカメラとレンズの組合せで求めた歪み特性データを用いて、補正します。

表 5.2.3.1 歪み補正入力画像仕様一覧

項目	データ
カメラ型式	OV7740(OmniVision 社製)
レンズ型式	--
画像サイズ	320(横) x 240(縦) [画素]
画像型式	RGB565, RAW

本アプリケーションノートでは、「4.4 プロジェクトのデバッグ」の12-5 [キャプチャ画面(歪み補正調整画面)]で確認することができます。



### 5.3 ファイル一覧

画像処理のファイル一覧を以下に示します。画像処理ミドルウェアは非公開の為、表 5.4.1 に示すファイル名は、ヘッダファイルが中心になります。

表 5.4.1 画像処理ファイル一覧

フォルダ名	ファイル名	内容
renesas_sw_library /image_proc	Mdl_IP_main.h	画像処理ミドルウェア関数ヘッダファイル
	Mdl_IP_ex.h	マクロ定義
	Mdl_IP_prot.h	各モード構造体定義用ヘッダファイル
	outof_TrustedMemory_Data.mot	画像処理ミドルウェア(公開部)
src	imag_proc.c	画像処理ミドルウェア制御用アプリケーション
	imag_proc.h	パラメータ定義用ヘッダファイル

## 5.4 データ構造体

### 【人物検知情報構造体】

```
typedef struct _PersonDetection_Info {  
    int flg;           // 検知 0:無し / 1:有り  
    float score;      // 一致度  
    int sx, sy, ex, ey; // 矩形 始点 X, 始点 Y, 終点 X, 終点 Y  
} PersonDetection_Info;
```

### 【人物検知結果構造体】

```
typedef struct _PersonDetection_Rslt {  
    int p_dct_cnt[MAX_AREA];           // エリア内人数 ( [0]:エリア 1 ~ [8]:エリア 9 )  
    PersonDetection_Info p_dct_inf[MAX_PERSON]; // 人物検知情報  
} PersonDetection_Rslt;
```

### 【動体検知情報構造体】

```
typedef struct _MovingDetection_Info {  
    int flg;           // 検知 0:無し / 1:有り  
    int area;         // エリア No ( 1 ~ 9 )  
    int sx, sy, ex, ey; // 矩形 始点 X, 始点 Y, 終点 X, 終点 Y  
} MovingDetection_Info;
```

### 【動体検知結果構造体】

```
typedef struct _MovingDetection_Rslt {  
    int p_dct_cnt; // 動体検知数  
    MovingDetection_Info p_dct_inf[MAX_MOVING]; // 動体検知情報  
} MovingDetection_Rslt;
```

### 【マクロ定義】

```
#define MAX_AREA 9  
#define MAX_PERSON 5  
#define MAX_MOVING 5
```

## 5.5 画像処理ミドルウェア関数リファレンス

画像処理ミドルウェア関数一覧を表 5.6.1 に示します。

表 5.6.1 関数一覧

関数名	機能概要
Mdl_IP_Init()	標準画像処理ミドルウェアの初期化処理
Mdl_IP_PersonDetection()	人物検知処理
Mdl_IP_PersonDetection_ParamChg()	人物検知処理のパラメータ変更
Mdl_IP_MovingDetection()	動体検知処理
Mdl_IP_MovingDetection_ParamChg()	動体検知処理のパラメータ変更
Mdl_IP_ImgRevise()	歪み補正処理
Mdl_IP_ImgRevise_ParamChg()	歪み補正処理のパラメータ変更

### 5.5.1 Mdl\_IP\_Init

#### **Description**

アプリケーションは本関数を画像処理機能使用開始前に 1 度だけ呼び出します。本関数は、画像処理ミドルウェア内の内部変数確保、初期化を実行します。

#### **Usage**

```
#include "Mdl_IP_ex.h"
void Mdl_IP_Init( unsigned char *wk_adr );
```

#### **Parameters**

wk\_adr    入力    ワーク用先頭アドレス

#### **Return Value**

無し

#### **Remark**

引数は、画像処理ミドルウェア内で使用するワーク用メモリ(320x240 : 4 面、160x120 : 2 面)の先頭アドレスとなります。

## 5.5.2 Mdl\_IP\_PersonDetection

**Description**

本関数は、第1引数で指定されたアドレスに格納されている画像に対し、人物が存在する位置検出及び人数計測を実行します。

実行結果詳細を第2引数で指定された人物検知結果構造体のアドレスに書き出し、実行結果画像を第3引数で指定されたアドレスに書き出します。

**Usage**

```
#include "Mdl_IP_ex.h"
int Mdl_IP_PersonDetection(
    unsigned char *in_img, PersonDetection_Rslt *rslt, unsigned char *extension );
```

**Parameters**

in_img	入力	入力画像格納先頭アドレス
rslt	出力	人物検知処理結果詳細
extension	出力	拡張機能 (NULL を指定のこと)

**Return Value**

0	正常終了
-1	初期化処理 Mdl_IP_Init() 未実行

**Remark**

本アプリケーションノートでは、デモンストレーションとして拡張機能(extension)の出力結果を使用しています。

## 5.5.3 Mdl\_IP\_PersonDetection\_ParamChg

**Description**

本関数は、「5.5.3 Mdl\_IP\_PersonDetection」の内部で使用する画像処理パラメータに対し、第1引数でパラメータの種別を指定して、第2引数で指定された設定値に変更します。

**Usage**

```
#include "Mdl_IP_ex.h"  
int Mdl_IP_PersonDetection_ParamChg( int kind, int val );
```

**Parameters**

Kind	入力	パラメータ種別
val	入力	設定値

**Return Value**

0	正常終了
-1	初期化処理 Mdl_IP_Init() 未実行
-2	パラメータ種別範囲外
-3	設定値範囲外

**Remark**

パラメータ種別

0 : 人物検出しきい値 (設定値範囲 : 1~9)

人物を検出する際の人物らしさを判別するしきい値を指定します。人物の検出が行われにくい場合は設定値を小さくし、過検出する場合は設定値を大きくします。

(デフォルト値 : 7)

1 : 人物検出感度 (設定値範囲 : 1~15)

人物を検出する際、検出候補箇所のコントラスト (明暗) 及び鮮やかさ (色数) を判別するしきい値を指定します。コントラストが小さく、鮮やかさが乏しい箇所を有効する場合は設定値を小さくし、無効とする場合は設定値を大きくします。

(デフォルト値 : 7)

## 5.5.4 Mdl\_IP\_MovingDetection

**Description**

本関数は、第 1 引数から第 3 引数で指定されたアドレスに格納されている連続取込した画像に対し、移動体の検出処理を実行します。

実行結果詳細を第 4 引数で指定された動体検知結果構造体のアドレスに書き出し、実行結果画像を第 5 引数で指定されたアドレスに書き出します。

**Usage**

```
#include "Mdl_IP_ex.h"
int Mdl_IP_MovingDetection( unsigned char *in_img1, unsigned char *in_img2
    unsigned char *in_img3, MovingDetection_Rslt *rslt, unsigned char *extension );
```

**Parameters**

in_img1	入力	入力画像 1 格納先頭アドレス
in_img2	入力	入力画像 2 格納先頭アドレス
in_img3	入力	入力画像 3 格納先頭アドレス
rslt	出力	動体検知処理結果詳細
extension	出力	拡張機能 (NULL を指定のこと)

**Return Value**

0	正常終了
-1	初期化処理 Mdl_IP_Init() 未実行

**Remark**

本アプリケーションノートでは、デモンストレーションとして拡張機能(extension)の出力結果を使用しています。

## 5.5.5 Mdl\_IP\_MovingDetection\_ParamChg

**Description**

本関数は、「5.5.4 Mdl\_IP\_MovingDetection」の内部で使用する画像処理パラメータに対し、第1引数でパラメータの種別を指定して、第2引数で指定された設定値に変更します。

**Usage**

```
#include "Mdl_IP_ex.h"
int Mdl_IP_MovingDetection_ParamChg( int kind, int val );
```

**Parameters**

無し

**Return Value**

0	正常終了
-1	初期化処理 Mdl_IP_Init() 未実行
-2	パラメータ種別範囲外
-3	設定値範囲外

**Remark**

パラメータ種別

0 : 動体検出しきい値 (設定値範囲 : 1~10)

動体を検出する際の判別しきい値を指定します。動体の検出が行われにくい場合は設定値を小さくし、過検出する場合は設定値を大きくします。

(デフォルト値 : 4)

1 : 動体検出感度 (設定値範囲 : 1~10)

動体を検出する際、画像差分 (1枚目と2枚目、2枚目と3枚目) で明るさの変化があったかを判別するしきい値を指定します。明るさの変化が小さい箇所を検出したい場合は設定値を小さくし、検出しない場合は設定値を大きくします。

(デフォルト値 : 3)

2 : 処理領域 上辺 (設定値範囲 : 0~9)

3 : 処理領域 下辺 (設定値範囲 : 0~9)

4 : 処理領域 左辺 (設定値範囲 : 0~9)

5 : 処理領域 右辺 (設定値範囲 : 0~9)

2~5は、動体検出を行う処理領域の画像端からの補正値を10%刻みで指定します。

上辺・下辺は画像高さに対する割合、左辺・右辺は画像幅に対する割合とします。

本アプリケーションノートでは、デフォルト値で固定となっています。

(デフォルト値 : 0)

## 5.5.6 Mdl\_IP\_ImgRevise

**Description**

本関数は、第1引数で指定されたアドレスに格納されている画像に対し、歪み補正処理を実行します。実行結果画像を第2引数で指定されたアドレスに書き出します。

**Usage**

```
#include "Mdl_IP_ex.h"
int Mdl_IP_ImgRevise ( unsigned char *in_img, unsigned char *out_img );
```

**Parameters**

in_img	入力	入力画像格納先頭アドレス
out_img	出力	歪み補正処理結果画像格納先頭アドレス

**Return Value**

0	正常終了
-1	初期化処理 Mdl_IP_Init() 未実行

**Remark**

歪み補正処理結果画像の格納先は、入力画像と同サイズのエリアが確保されていなければなりません。



## 5.5.7 Mdl\_IP\_ImgRevise\_ParamChg

**Description**

本関数は、「5.5.6 Mdl\_IP\_ImgRevise」の内部で使用する歪み補正パラメータに対し、第1引数でパラメータの種別を指定して、第2引数で指定された設定値に変更します。

**Usage**

```
#include "Mdl_IP_ex.h"
int Mdl_IP_ImgRevise_ParamChg( int kind, int val );
```

**Parameters**

kind	入力	パラメータ種別
val	入力	設定値

**Return Value**

0	正常終了
-1	初期化処理 Mdl_IP_Init() 未実行
-2	パラメータ種別範囲外
-3	設定値範囲外

**Remark**

パラメータ種別

- 0 : 歪み補正有効／無効（設定値範囲：0～1）  
歪み補正処理を行うかどうかを指定します。有効時は1、無効時は0となります。  
(デフォルト値：1)
- 1 : 水平方向補正值（設定値範囲：-127～127）  
歪み補正を行う際の基準点X座標を画像中心位置からのオフセットで指定します。画像中心から左方向は負値、右方向は正值となります。  
(デフォルト値：-2)
- 2 : 垂直方向補正值（設定値範囲：-127～127）  
歪み補正を行う際の基準点Y座標を画像中心位置からのオフセットで指定します。画像中心から上方向は負値、下方向は正值とします。  
(デフォルト値：80)

## 6. CS+を使用する場合

本アプリケーションノートは、CS+でも評価することができます。なお、ビルドを行うには「RX ファミリー用 C/C++コンパイラパッケージ V2.02.00」以降が必要です。ここでは、まだ製品版をお持ちでない場合を想定して、無償評価版を利用する例をご紹介します。

### 6.1 CS+の入手とインストール方法

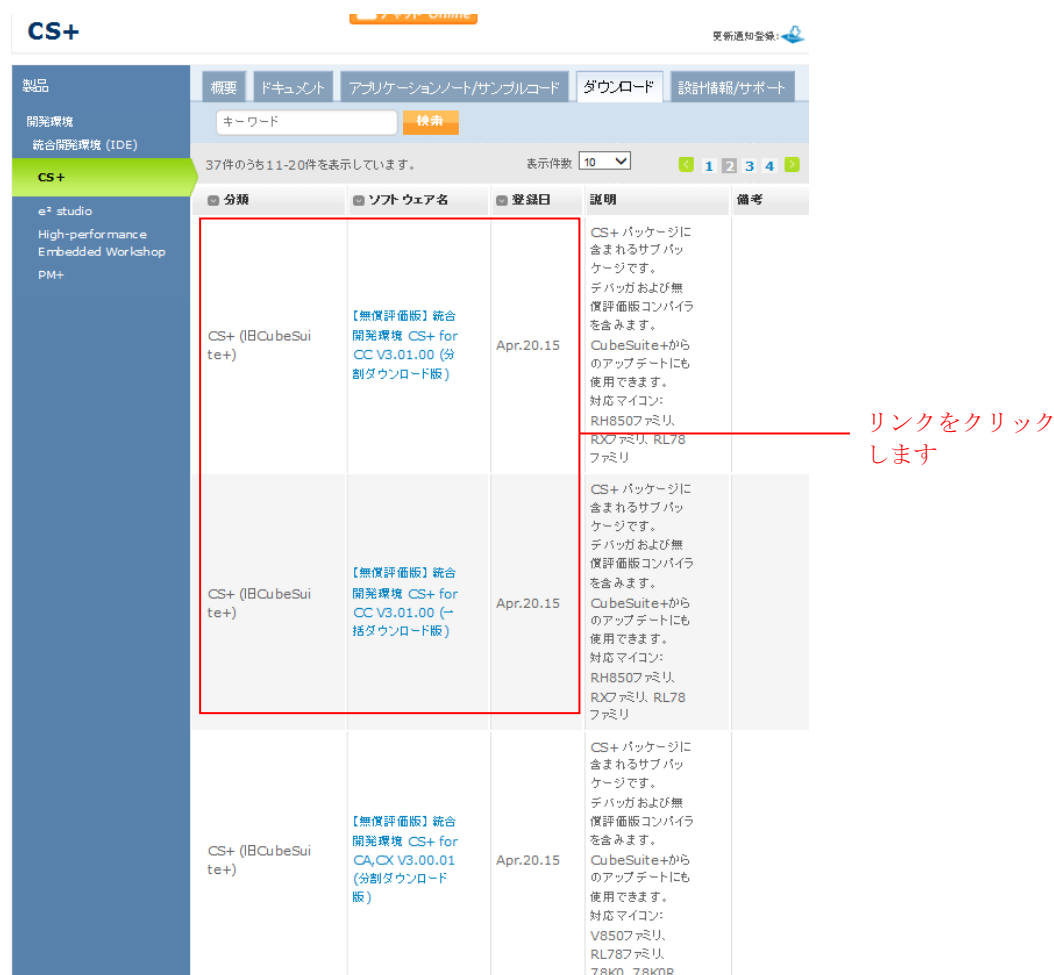
CS+は、ルネサスのホームページからダウンロードできます。

- 以下の URL にアクセスし、CS+のダウンロードページを表示します。

[http://japan.renesas.com/cs+\\_download](http://japan.renesas.com/cs+_download)

- 表示された項目の中から、「【無償評価版】統合開発環境 CS+ V3.01.00」をクリックします。（分割ダウンロード版と一括ダウンロード版がありますが、内容の違いはありません。）

その後表示されたページの指示に従い、CS+のインストーラをダウンロードします。



The screenshot shows the CS+ download page with a table of software packages. The table has columns for '分類' (Category), 'ソフトウェア名' (Software Name), '登録日' (Registration Date), '説明' (Description), and '備考' (Remarks). The entry for 'CS+ (IBCubeSuite+) V3.01.00 (分割ダウンロード版)' is highlighted with a red box. A red arrow points to the link in the '説明' column, with the text 'リンクをクリックします' (Click the link).

分類	ソフトウェア名	登録日	説明	備考
CS+ (IBCubeSuite+)	【無償評価版】統合開発環境 CS+ for CC V3.01.00 (分割ダウンロード版)	Apr.20.15	CS+ パッケージに含まれるサブパッケージです。デバッグおよび無償評価版コンパイラを含みます。CubeSuite+からのアップデートにも使用できます。対応マイコン: RH850ファミリ, RXファミリ, RL78ファミリ	
CS+ (IBCubeSuite+)	【無償評価版】統合開発環境 CS+ for CC V3.01.00 (一括ダウンロード版)	Apr.20.15	CS+ パッケージに含まれるサブパッケージです。デバッグおよび無償評価版コンパイラを含みます。CubeSuite+からのアップデートにも使用できます。対応マイコン: RH850ファミリ, RXファミリ, RL78ファミリ	
CS+ (IBCubeSuite+)	【無償評価版】統合開発環境 CS+ for CA, CX, V3.00.01 (分割ダウンロード版)	Apr.20.15	CS+ パッケージに含まれるサブパッケージです。デバッグおよび無償評価版コンパイラを含みます。CubeSuite+からのアップデートにも使用できます。対応マイコン: V850ファミリ, RL78ファミリ, 78K0, 78K0R	

3. ダウンロードした CS+ のインストーラを実行し、CS+ を PC にインストールします。

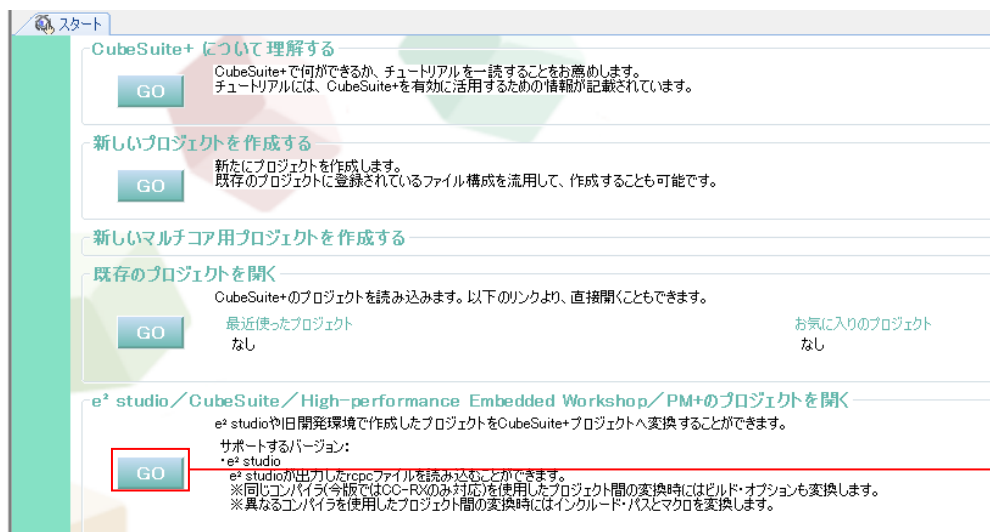
インストール方法は「CS+ V3.00.00 統合開発環境 ユーザーズマニュアルインストーラ編」を参照してください。

[http://documentation.renesas.com/doc/products/tool/doc/r20ut3094jj0100\\_csin.pdf](http://documentation.renesas.com/doc/products/tool/doc/r20ut3094jj0100_csin.pdf)

## 6.2 プロジェクトのインポート

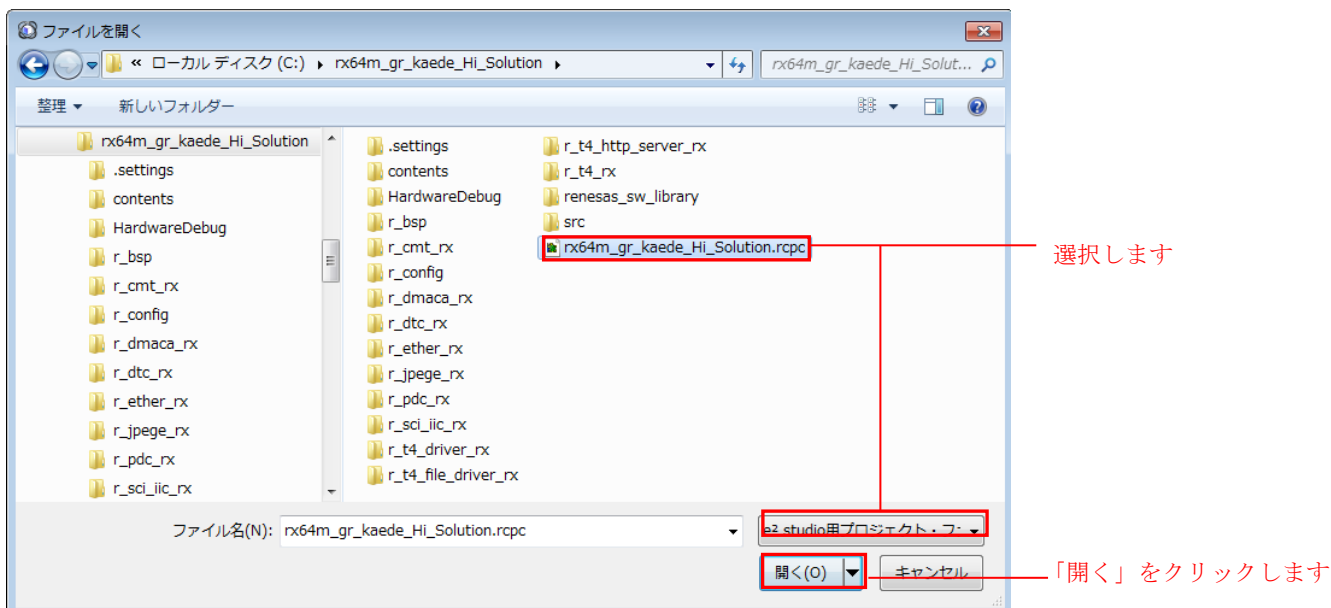
CS+ に、本アプリケーションノートに付属している「Renesas 共通プロジェクト・ファイル」をインポートします。

1. 「本アプリケーションノート提供 ZIP」ファイルを、任意のフォルダに解凍します。
2. CS+ を起動し、スタート画面から「e<sup>2</sup> studio / CubeSuite / High-performance Embedded Workshop / PM+ のプロジェクトを開く」の「GO」をクリックします。

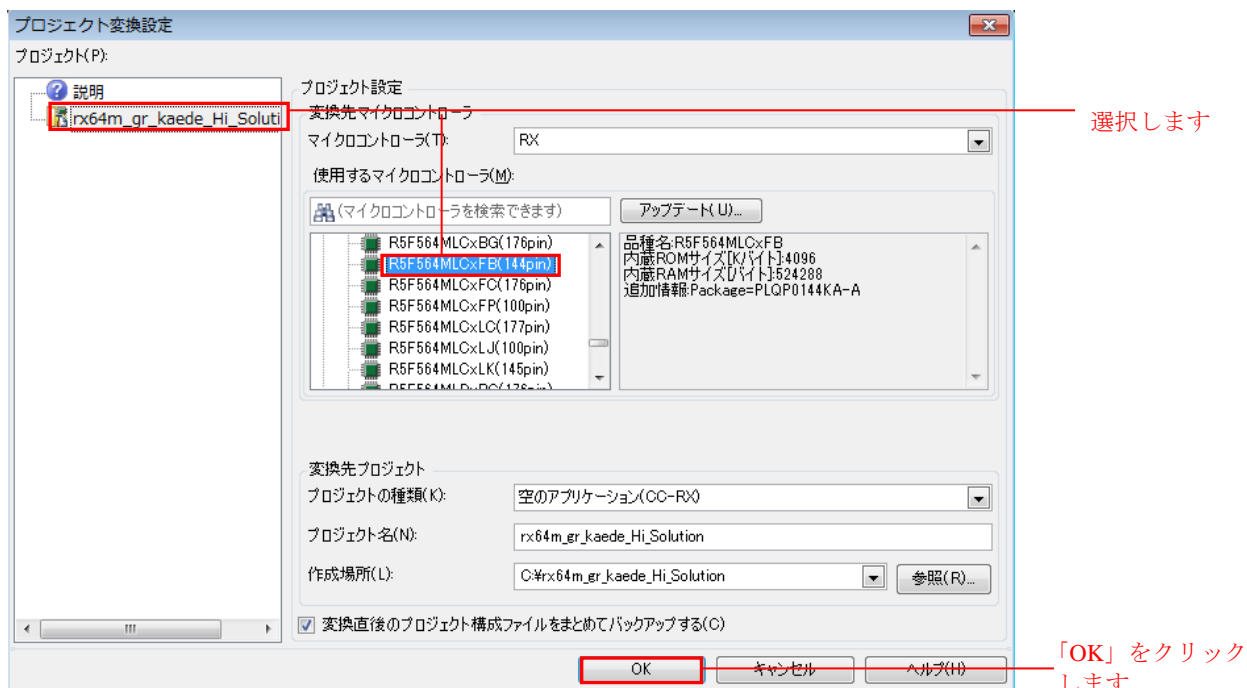


「GO」をクリックします

3. 1.で解凍したフォルダを開き、その中にある「Web サーバシステムプロジェクト (rx64m\_gr\_kaede\_Hi\_Solution フォルダ)」を開き、その中にある「Renesas 共通プロジェクト・ファイル (rx64m\_gr\_kaede\_Hi\_Solution.rcpc)」を選択し、「開く」をクリックします。



4. 「プロジェクトツリー」からプロジェクトを選択した後、以下に示す様に各項目を選択し、「OK」をクリックします。尚、「使用するマイクロコントローラ」は、RX64M の R5F564MLCxFB(144pin)を選択してください。



5. プロジェクトの変換が行われ、変換されたプロジェクトが開かれます。また、e<sup>2</sup> studio のプロジェクトは、バックアップが作成されます。

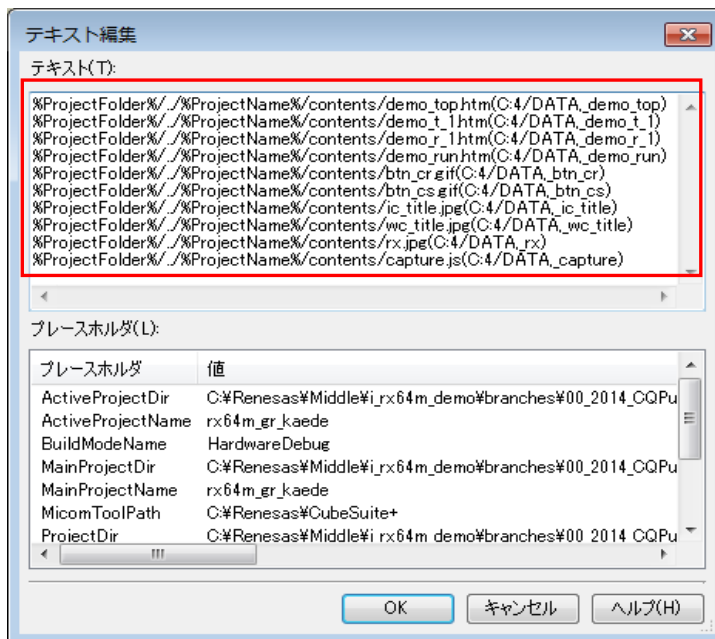
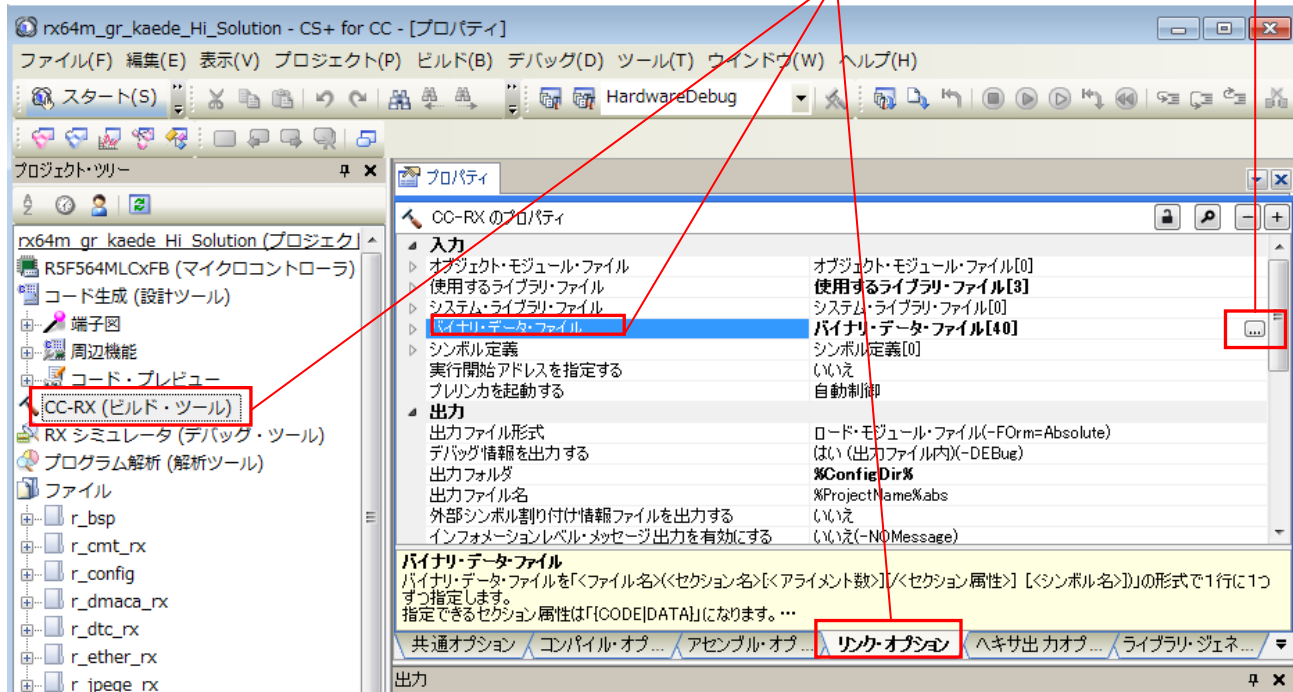
### 6.3 設定変更

「Renesas 共通プロジェクト・ファイル (rx64m\_gr\_kaede\_Hi\_Solution.rcpc)」では継承できない設定を変更します。

1. 継承できない部分を編集します。

バイナリ・データ・ファイルを選択します

クリックします



編集します

[編集前]

```
%ProjectFolder%/../%ProjectName%/contents/capture.js}(C:4/DATA,_capture)
%ProjectFolder%/../%ProjectName%/contents/com.css}(C:4/DATA,_com)
```

```
%ProjectFolder%/../%ProjectName%/contents/demo_r_1.htm}(C:4/DATA,_demo_r_1)
%ProjectFolder%/../%ProjectName%/contents/demo_r_2.htm}(C:4/DATA,_demo_r_2)
%ProjectFolder%/../%ProjectName%/contents/demo_r_3.htm}(C:4/DATA,_demo_r_3)
%ProjectFolder%/../%ProjectName%/contents/demo_ru1.htm}(C:4/DATA,_demo_ru1)
%ProjectFolder%/../%ProjectName%/contents/demo_ru2.htm}(C:4/DATA,_demo_ru2)
%ProjectFolder%/../%ProjectName%/contents/demo_ru3.htm}(C:4/DATA,_demo_ru3)
%ProjectFolder%/../%ProjectName%/contents/demo_ru4.htm}(C:4/DATA,_demo_ru4)
%ProjectFolder%/../%ProjectName%/contents/demo_st1.htm}(C:4/DATA,_demo_st1)
%ProjectFolder%/../%ProjectName%/contents/demo_st2.htm}(C:4/DATA,_demo_st2)
%ProjectFolder%/../%ProjectName%/contents/demo_st3.htm}(C:4/DATA,_demo_st3)
%ProjectFolder%/../%ProjectName%/contents/demo_st4.htm}(C:4/DATA,_demo_st4)
%ProjectFolder%/../%ProjectName%/contents/demo_t_1.htm}(C:4/DATA,_demo_t_1)
%ProjectFolder%/../%ProjectName%/contents/demo_t_2.htm}(C:4/DATA,_demo_t_2)
%ProjectFolder%/../%ProjectName%/contents/demo_top.htm}(C:4/DATA,_demo_top)
%ProjectFolder%/../%ProjectName%/contents/ic_title.gif}(C:4/DATA,_ic_title)
%ProjectFolder%/../%ProjectName%/contents/rx.jpg}(C:4/DATA,_rx)
%ProjectFolder%/../%ProjectName%/contents/start.gif}(C:4/DATA,_start)
%ProjectFolder%/../%ProjectName%/contents/stop.gif}(C:4/DATA,_stop)
%MainProjectDir%¥%ProjectName%/contents/capture.js(C:4/DATA,_capture)
%MainProjectDir%¥%ProjectName%/contents/com.css(C:4/DATA,_com)
%MainProjectDir%¥%ProjectName%/contents/demo_r_1.htm(C:4/DATA,_demo_r_1)
%MainProjectDir%¥%ProjectName%/contents/demo_r_2.htm(C:4/DATA,_demo_r_2)
%MainProjectDir%¥%ProjectName%/contents/demo_r_3.htm(C:4/DATA,_demo_r_3)
%MainProjectDir%¥%ProjectName%/contents/demo_ru1.htm(C:4/DATA,_demo_ru1)
%MainProjectDir%¥%ProjectName%/contents/demo_ru2.htm(C:4/DATA,_demo_ru2)
%MainProjectDir%¥%ProjectName%/contents/demo_ru3.htm(C:4/DATA,_demo_ru3)
%MainProjectDir%¥%ProjectName%/contents/demo_ru4.htm(C:4/DATA,_demo_ru4)
%MainProjectDir%¥%ProjectName%/contents/demo_st1.htm(C:4/DATA,_demo_st1)
%MainProjectDir%¥%ProjectName%/contents/demo_st2.htm(C:4/DATA,_demo_st2)
%MainProjectDir%¥%ProjectName%/contents/demo_st3.htm(C:4/DATA,_demo_st3)
%MainProjectDir%¥%ProjectName%/contents/demo_st4.htm(C:4/DATA,_demo_st4)
%MainProjectDir%¥%ProjectName%/contents/demo_t_1.htm(C:4/DATA,_demo_t_1)
%MainProjectDir%¥%ProjectName%/contents/demo_t_2.htm(C:4/DATA,_demo_t_2)
%MainProjectDir%¥%ProjectName%/contents/demo_top.htm(C:4/DATA,_demo_top)
%MainProjectDir%¥%ProjectName%/contents/ic_title.gif(C:4/DATA,_ic_title)
%MainProjectDir%¥%ProjectName%/contents/rx.jpg(C:4/DATA,_rx)
%MainProjectDir%¥%ProjectName%/contents/start.gif(C:4/DATA,_start)
%MainProjectDir%¥%ProjectName%/contents/stop.gif(C:4/DATA,_stop)
```

## [編集後]

不要な”}”と不要なパスを削除します。(下記は削除後の状態)

```
%ProjectFolder%/../%ProjectName%/contents/capture.js(C:4/DATA,_capture)
%ProjectFolder%/../%ProjectName%/contents/com.css(C:4/DATA,_com)
%ProjectFolder%/../%ProjectName%/contents/demo_r_1.htm(C:4/DATA,_demo_r_1)
%ProjectFolder%/../%ProjectName%/contents/demo_r_2.htm(C:4/DATA,_demo_r_2)
%ProjectFolder%/../%ProjectName%/contents/demo_r_3.htm(C:4/DATA,_demo_r_3)
%ProjectFolder%/../%ProjectName%/contents/demo_ru1.htm(C:4/DATA,_demo_ru1)
%ProjectFolder%/../%ProjectName%/contents/demo_ru2.htm(C:4/DATA,_demo_ru2)
%ProjectFolder%/../%ProjectName%/contents/demo_ru3.htm(C:4/DATA,_demo_ru3)
%ProjectFolder%/../%ProjectName%/contents/demo_ru4.htm(C:4/DATA,_demo_ru4)
```

---

%ProjectFolder%/../%ProjectName%/contents/demo\_st1.htm(C:4/DATA,\_demo\_st1)  
%ProjectFolder%/../%ProjectName%/contents/demo\_st2.htm(C:4/DATA,\_demo\_st2)  
%ProjectFolder%/../%ProjectName%/contents/demo\_st3.htm(C:4/DATA,\_demo\_st3)  
%ProjectFolder%/../%ProjectName%/contents/demo\_st4.htm(C:4/DATA,\_demo\_st4)  
%ProjectFolder%/../%ProjectName%/contents/demo\_t\_1.htm(C:4/DATA,\_demo\_t\_1)  
%ProjectFolder%/../%ProjectName%/contents/demo\_t\_2.htm(C:4/DATA,\_demo\_t\_2)  
%ProjectFolder%/../%ProjectName%/contents/demo\_top.htm(C:4/DATA,\_demo\_top)  
%ProjectFolder%/../%ProjectName%/contents/ic\_title.gif(C:4/DATA,\_ic\_title)  
%ProjectFolder%/../%ProjectName%/contents/rx.jpg(C:4/DATA,\_rx)  
%ProjectFolder%/../%ProjectName%/contents/start.gif(C:4/DATA,\_start)  
%ProjectFolder%/../%ProjectName%/contents/stop.gif(C:4/DATA,\_stop)



2. 複数のフォルダをプロジェクトから外します。

プロジェクト・ツリー

CC-RX (ビルド・ツール)  
RX シミュレータ (デバッグ・ツール)  
プログラム解析 (解析ツール)  
ファイル

- r\_bsp
- r\_cmt\_rx
- r\_config
- r\_dmaca\_rx
  - r\_dmaca\_rx\_if.h
  - readme.txt
  - doc
  - ref
  - src
    - r\_dmaca\_rx.c
    - r\_dmaca\_rx\_private.h
  - targets
    - rx64m
    - rx71m

r\_dtc\_rx

- doc
- readme.txt
- ref
- r\_dtc\_rx\_if.h
- src
  - r\_dtc\_rx.c
  - r\_dtc\_rx\_private.h
- targets
  - rx110
  - rx111
  - rx113
  - rx64m
  - rx71m

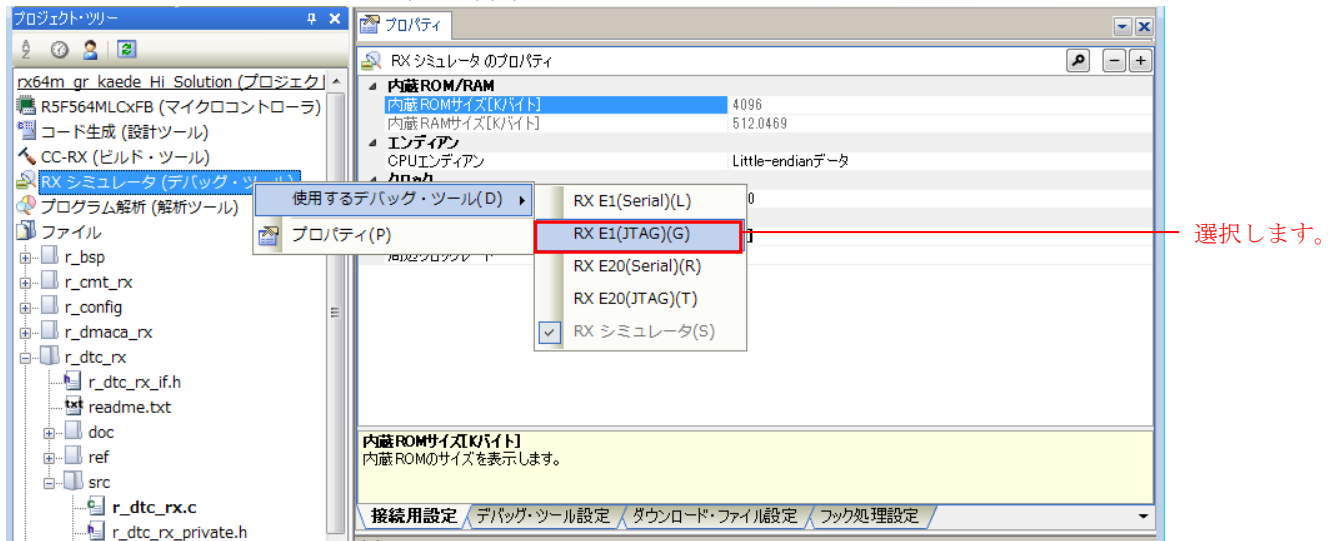
r\_sci\_iic\_rx

- r\_sci\_iic\_rx\_if.h
- readme.txt
- doc
- ref
- src
  - targets
    - sci
      - r\_sci\_iic\_rx.c
      - r\_sci\_iic\_rx\_platform.h
      - r\_sci\_iic\_rx\_private.h
    - rx110
    - rx111
    - rx113
    - rx64m
    - rx71m

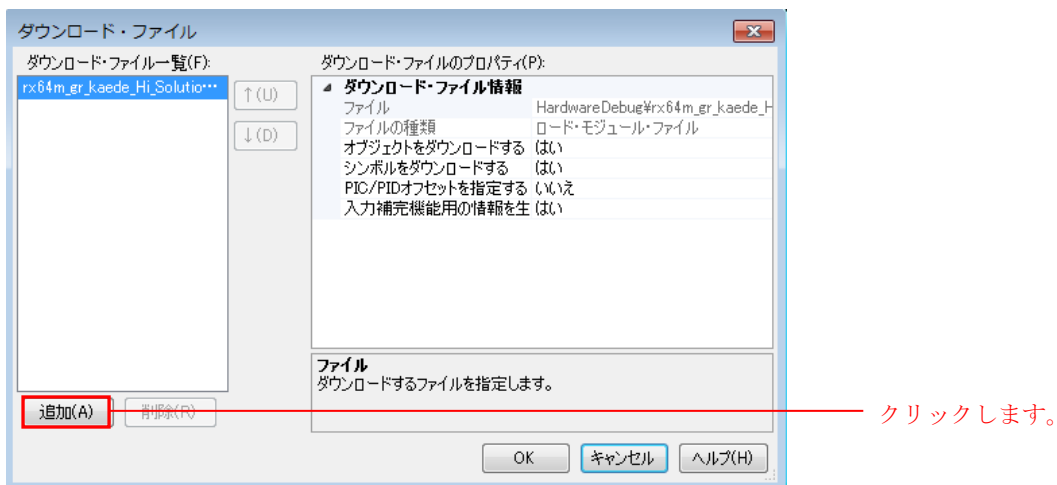
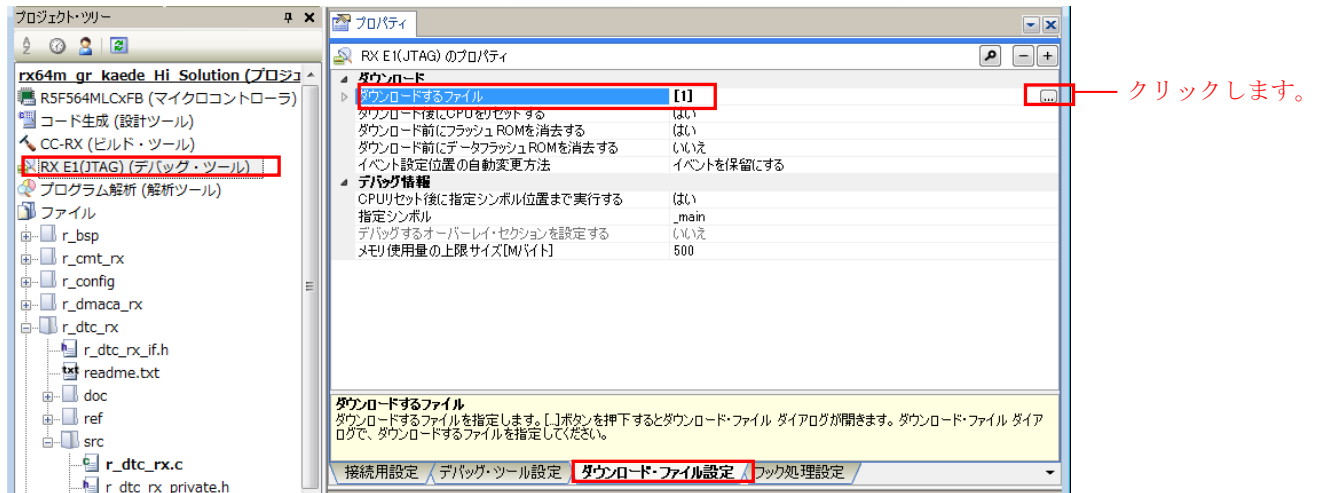
それぞれ選択後に右クリックし、下記「プロジェクトから外す」を選択します

- 追加(D)
- エクスプローラでフォルダを開く(F)
- Windowsエクスプローラのメニュー(W)
- プロジェクトから外す(R) Shift+Delete**
- コピー(C) Ctrl+C
- 貼り付け(P) Ctrl+V
- 名前の変更(N) F2
- プロパティ(P)

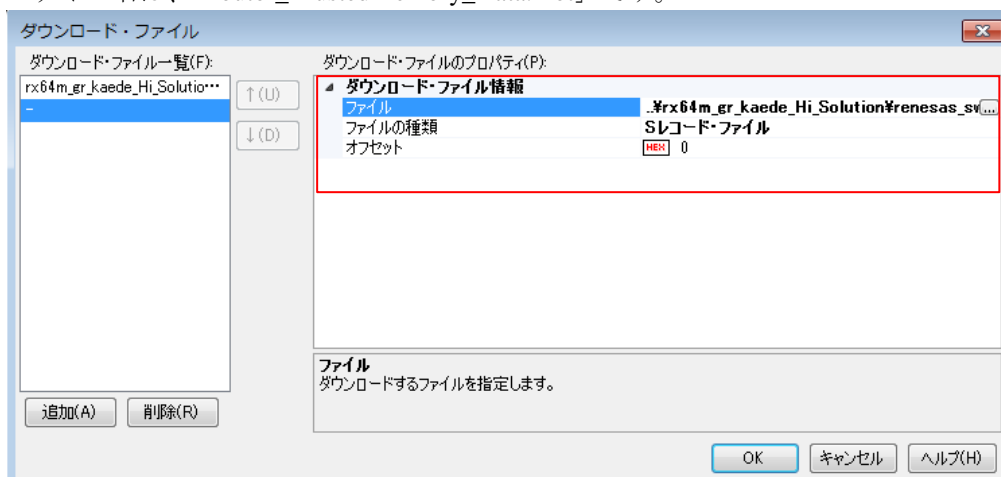
3. デバッグ・ツールを「RX E1(JTAG)(G)」に変更します。



4. ダウンロードファイルを追加します。



5. ファイルの種類を「Sレコード・ファイル」として、ダウンロードするファイルを追加します。ファイルは、「rx64m\_gr\_kaede\_Hi\_Solution¥renesas\_sw\_library¥image\_proc」に置かれています。ファイル名は、「outof\_TrustedMemory\_Data.mot」です。



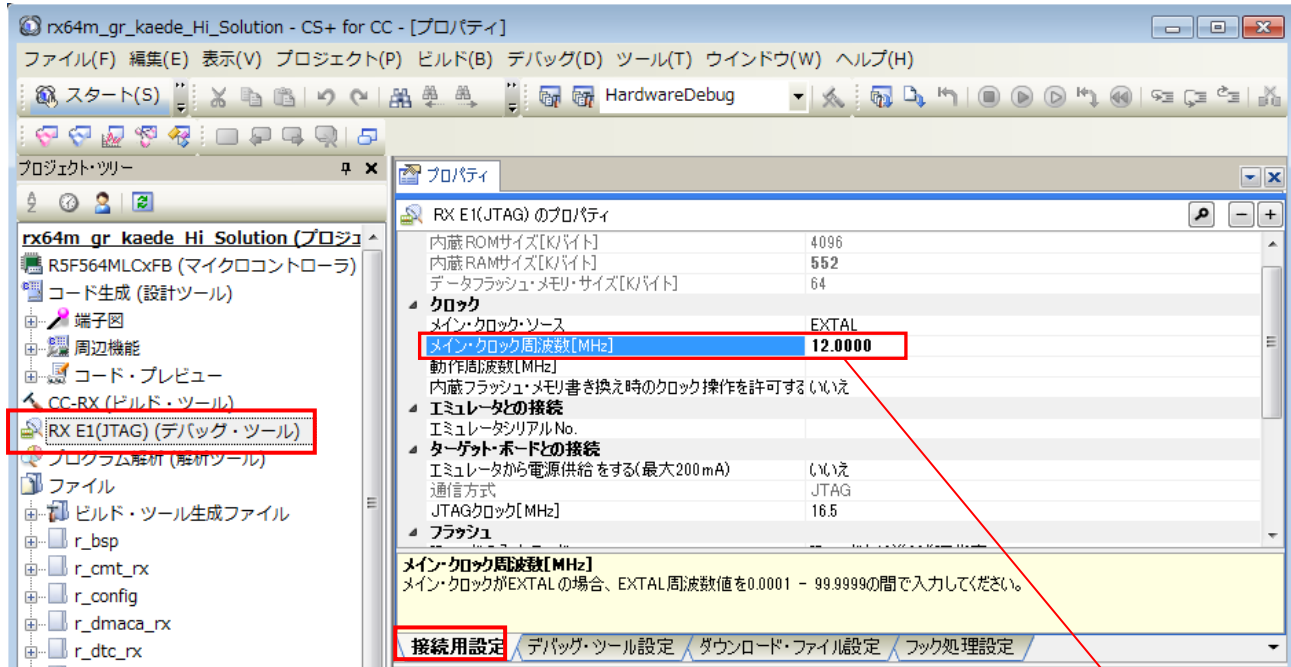
「OK」をクリックします。

6. プロジェクトをビルドします。

[JTAG 使用でダウンロード時の注意点]

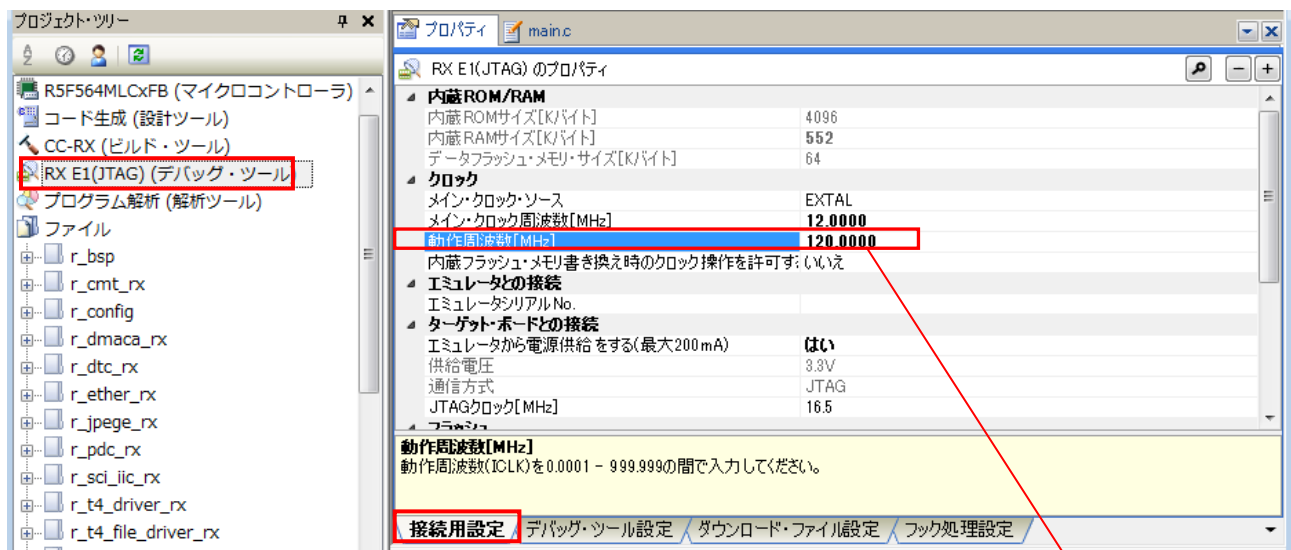
クロック設定を手動で行う必要があります。

1. 「メイン・クロック周波数[MHz]」に「12」と設定してください。



設定します

2. 「動作周波数[MHz]」に「120」と設定してください。



設定します

## 7. 補足

### 7.1 無償評価版の「RX ファミリ用 C/C++コンパイラパッケージ」を利用する場合の注意事項

無償評価版の「RX ファミリ用 C/C++コンパイラパッケージ」には、使用期限と使用制限があります。使用期限が過ぎた場合、使用制限によりロードモジュールが正しく生成されなくなる場合があります。

詳しくは、ルネサスのホームページにある、評価版ソフトウェアツールのページを参照してください。

URL : [http://japan.renesas.com/products/tools/evaluation\\_software/index.jsp](http://japan.renesas.com/products/tools/evaluation_software/index.jsp)

### 7.2 Web ブラウザ上でのキャプチャデータ更新遅延について

お使いのクライアント PC の状態等によっては、スムーズなキャプチャ更新ができない場合があります。そのようなケースでは、コンフィグファイルの通信端点数を増やすことによって、改善できます。

本ファームウェアでは、内蔵 RAM のみでの実装となっている為、通信端点数の上限は約 16 程度となっています。

変更箇所は、下記 3 ファイルの計 4 か所です。増減数はそろえるようにしてください。

【rx64m\_gr\_kaede\_Hi\_Solution /r\_t4\_rx/src/config\_tcpudp.c】

#### 1. TCP 受付口の設定

```

/** Definition of TCP reception point (only port number needs to be set) */
T_TCP_CREP tcp_crep[] =
{
    /* { attribute of reception point, {local IP address, local port number} */
    { 0x0000, { 0, 80 }},
    { 0x0000, { 0, 80 }},
    { 0x0000, { 0, 80 }},
    { 0x0000, { 0, 80 }},
    { 0x0000, { 0, 80 }}, -受け口 5    ←追加時イメージ
    { 0x0000, { 0, 80 }}, -受け口 6
    .
    .
};

```

} 変更前の受け口を 4 とする

## 2. TCP 通信端点の設定

```

/** Definition of TCP communication end point
(only receive window size needs to be set) */
T_TCP_CCEP tcp_ccep[] =
{
    /* { attribute of TCP communication end point,
        top address of transmit window buffer, size of transmit window buffer,
        top address of receive window buffer, size of receive window buffer,
        address of callback routine }
    */
    { 0, 0, 0, 0, 1460, http_callback },
    { 0, 0, 0, 0, 1460, http_callback },
    { 0, 0, 0, 0, 1460, http_callback },
    { 0, 0, 0, 0, 1460, http_callback },
    { 0, 0, 0, 0, 1460, http_callback }, -通信端点 5 ←追加時イメージ
    { 0, 0, 0, 0, 1460, http_callback }, -通信端点 6
    .
    .
}

```

3. HTTP サーバで使用する通信端点  
#define HTTP\_TCP\_CEP\_NUM

【rx64m\_gr\_kaede\_Hi\_Solution /src/main.c】

## 4. T4(TCP/IP プロトコルスタック)で使用するワークエリア

この値は、一度プログラムを動かして tcpudp\_get\_ramsize()の戻り値を元に設定しています。

対象処理は、同ファイル内にあります。

## 7.3 コードフラッシュ内の書き込み不可な領域について

GR-KAEDE に搭載されている RX64M は、TM 機能が ON にされた状態で出荷されています。

TM 機能が ON の場合、コードフラッシュ 0xFFFFE0000~0xFFFFEFFFF への書き込みはできません。

この範囲を避けるようにコーディングしてください。

誤ってコードが配置されたターゲットプログラムをダウンロードした場合に、ツールによっては警報などがでない場合がありますが、該当領域は更新されていませんのでご注意ください。

TM 機能については、「RX64M グループのユーザーズマニュアル ハードウェア編」でご確認ください。

## 8. カメラソリューション製品紹介

本アプリケーションノートは、コンパイラ CC-RX、動作ボード GR-KAEDE ボードで動作するカメラソリューションですが、その他組み合わせでも同様な操作ができます。

### 8.1 がじえっとるねさす GR-KAEDE ボード

本アプリケーションノートと同様の動作をします。

クラウド Web コンパイラとして、GNU GCC を使用できます。

Arduino UNO と互換性のある端子配置、ライブラリが準備されており、マイコンの専門知識がなくても導入が容易になっています。

<http://gadget.renesas.com/ja/product/kaede.html>

### 8.2 HMI 拡張ボードを使ったカメラ機能/音声再生機能デモンストレーション

Renesas Starter Kit+ for RX64M(以下、RSK と称す)と HMI 拡張ボードを使ったカメラソリューションです。

RSK では、Ethernet2 ポート対応など GR-KAEDE に比べて、より多くの RX64M 機能を使用できます。

ただし、画像処理システムは動作しません。

デモンストレーション入手先:

近日中に公開予定。

RSK、HMI 拡張ボード情報:

[http://japan.renesas.com/products/tools/introductory\\_tools/renesas\\_starter\\_kits/rsk\\_rx64m/index.jsp](http://japan.renesas.com/products/tools/introductory_tools/renesas_starter_kits/rsk_rx64m/index.jsp)

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

<http://japan.renesas.com/>

お問い合わせ先

<http://japan.renesas.com/contact/>

すべての商標および登録商標は、それぞれの所有者に帰属します。



## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2015.10.01	-	初版発行

## 製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

### 1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

### 2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

### 3. リザーブアドレス（予約領域）のアクセス禁止

【注意】リザーブアドレス（予約領域）のアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

### 4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後、切り替えてください。

リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

### 5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違っていると、内部ROM、レイアウトパターンの相違などにより、電氣的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

## ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。  
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、  
家電、工作機械、パーソナル機器、産業用ロボット等  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、  
防災・防犯装置、各種安全装置等  
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事情報に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。  
総合お問合せ窓口：<http://japan.renesas.com/contact/>