

R01AN2608JJ0103 Rev.1.03 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 2016.02.29 **RX Driver Package Application** 

## 要旨

本資料は、TCP/IP プロトコルスタック(M3S-T4-Tiny)を用いた産業向けネットワークソリューションの 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、ネットワークミドルウェア(DHCP クライアント、DNS クライアント、 HTTP サーバ、FTP サーバ) モジュールを使用したメインプログラムのサンプル・コードが含まれており、 RX64M, RX71M グループ用 RX Driver Package と組み合わせることで、各種ネットワークシステムを構築でき ます。

## 動作確認デバイス

RX64M グループ (Renesas Starter Kit+ RX64M)

RX71M グループ (Renesas Starter Kit+RX71M)

## 使用した RX Driver Package 環境

RX64M, RX71M グループ RX Driver Package Ver.1.02 (R01AN2606JJ)

お客様の製品にてご利用される際は、お客様の環境に合わせて十分に評価してください。

また、本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、 十分評価してください。



## 目次

1. 概要		3
1.1 本アプ	<sup>?</sup> リケーションノートについて	3
1.2 動作環	境	4
1.3 モジュ	.ール構成	5
1.4 ファイ	ル構成	7
1.5 プロジ	<sup>;</sup> ェクトについて	8
2. 開発環境	意の入手	9
2.1 e <sup>2</sup> stud	lio の入手方法	
2.2 コンパ	ペイラパッケージの入手方法	9
3. プロジェ	- クトの構築	10
3.1 ワーク	スペースの作成	
3.2 プロジ	<sup>;</sup> ェクトの作成	
3.3 プロジ	<sup>;</sup> ェクトのインポート	
3.4 変更情	報	
3.4.1 ⊐:	ンフィギュレーションの変更	
3.4.2 プロ	ロジェクト設定の変更	
4. 動作確認	ខ្ល	27
4.1 プロジ	<sup>;</sup> ェクトのビルド	
4.2 デバッ	グの準備	
4.2.1 機器	器の構成	
4.2.2 評価	価ボードの設定	
4.2.3 クラ	ライアント PC の設定	
4.2.4 US	3B メモリの準備	
4.3 プロジ	<sup>,</sup> ェクトのデバッグ	
4.3.1 HT	TP サーバ機能の確認	
4.3.2 FT	゙P サーバ機能の確認	
5. ネットワ	フークミドルウェア仕様	
6. メインプ	プログラム仕様	47
6.1 ファイ	ル構成	
6.2 モジュ	ールー覧	
6.3 処理フ	□−	
7. ユーザ定	2義関数について	57
8. 補足		58
8.1 USB F	ミライバの制限事項	
8.2 本アプ	<sup>?</sup> リケーションの制限事項	
8.3 DHCP	クライアント及び FTP サーバの制限事項	
8.4 無償評	<sup>:</sup> 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 58

## 1. 概要

## 1.1 本アプリケーションノートについて

本アプリケーションノートでは、メインプログラムと RX64M, RX71M グループ用 RX Driver Package 内蔵 のネットワークミドルウェア、TCP/IP プロトコルスタック(以下、M3S-T4-Tiny と略す)、Ethernet ドライ バ、M3S-TFAT-Tiny FAT ファイルシステム(以下、TFAT と略す)、USB ドライバ(ホストマスストレージ クラスドライバ USB HMSC と Basic Firmware)、ボードサポートパッケージ(以下、BSP と略す)の FIT モ ジュール等と組み合わせて評価するまでの手順について説明します。

本アプリケーションノートは、Renesas Starter Kit+ for RX64M・Renesas Starter Kit+ for RX71M(以降、RSK と表記)上で動作します。



## 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 1-1 動作環境

対応 MCU	RX64M グループ
評価ボード	Renesas Starter Kit+ RX64M
	http://japan.renesas.com/products/tools/introductory_tools/renesas_starter_kits/rsk_rx64m/index.jsp
統合開発環境(IDE)	e² studio V4.1.0 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(Renesas Starter Kit+ RX64M に同梱), E20
RX Driver Package	RX64M, RX71M グループ RX Driver Package Ver.1.02(R01AN2606JJ)注

表 1-2 動作環境

対応 MCU	RX71M グループ			
評価ボード	Renesas Starter Kit+ RX71M			
	http://japan.renesas.com/products/tools/introductory_tools/renesas_starter_kits/rsk_rx71m/index.jsp			
統合開発環境(IDE)	e² studio V4.1.0 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(Renesas Starter Kit+ RX71M に同梱), E20			
RX Driver Package	RX64M, RX71M グループ RX Driver Package Ver.1.02(R01AN2606JJ)注			
【注】本アプリケーションノートは、上記の RX Driver Package に入っているモジュールと組みわせて動作を				
[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[				

確認しています。本アプリケーションノートで使用するモジュールを、別のモジュールと入れかえた 場合については、各自で動作を確認してください。



#### 図 1-1 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とモジュール一覧を以下に示します。



図 1-2 モジュール構成イメージ



種類	モジュール名	FIT モジュール名	Rev.
Board Support	ボードサポートパッケージ(BSP)	r_bsp	3.00
Package			
Device Driver	コンペアマッチタイマ(CMT)	r_cmt_rx	2.60
Device Driver	USB Basic Firmware	r_usb_basic	1.10
Device Driver	USB Host Mass Storage Class	r_usb_hmsc	1.10
Device Driver	イーサネットコントローラ(ETHERC)	r_ether_rx	1.02
Middleware	組み込み用 M3S-T4-Tiny モジュール【注 1】	r_t4_rx	2.02
Interface	Ethernet ドライバと組み込み用 TCP/IP	r_t4_driver_rx	1.02
Middleware	組み込み用 TCP/IP M3S-T4-Tiny を用いた DHCP	r_t4_dhcp_client_rx	1.03
	クライアントモジュール		
Middleware	組み込み用 TCP/IP M3S-T4-Tiny を用いた DNS	r_t4_dns_client_rx	1.02
	クライアントモジュール		
Middleware	組み込み用 TCP/IP M3S-T4-Tiny を用いた FTP	r_t4_ftp_server_rx	1.03
	サーバモジュール		
Middleware	組み込み用 TCP/IP M3S-T4-Tiny を用いた Web	r_t4_http_server_rx	1.04
	サーバモジュール		
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	オープンソース FAT ファイルシステム	r_tfat_rx	3.02
	M3S-TFAT-Tiny モジュール		
Interface	M3S-TFAT-Tiny メモリドライバインタフェース	r_tfat_driver_rx	1.02
	モジュール		
Application	メインプログラム	src	1.01

表 1-3 モジュール-	一覧
--------------	----

【注 1】本パッケージには、評価版の「M3S-T4-Tiny(TCP/IP プロトコルスタックライブラリ)」が含まれています。製品版については、以下の URL を参照してください。 http://japan.renesas.com/mw/t4

## 1.4 ファイル構成

本アプリケーションノートのファイル構成を以下に示します。



図 1-3 ファイル構成

本アプリケーションノート提供 ZIP ファイルを解凍すると、同名のフォルダが作成され、その中に各フォルダやファイルが入っています。

「workspace」フォルダの下にある「ネットワークソリューションプロジェクト (r\_network\_solution\_rx64m)」 又は「ネットワークソリューションプロジェクト (r\_network\_solution\_rx71m)」は、本アプリケーションを 構築したプロジェクトです。 $e^2$  studio のワークスペースにインポートすることでアプリケーションの動作確 認が可能です。

「reference\_documents」フォルダの中には、FIT モジュールを各開発環境で使用するための説明書が入って います。「e<sup>2</sup> studio に取り込む方法(R01AN1723JU)」は、FIT プラグインを使用して、FIT モジュールを e<sup>2</sup> studio のプロジェクトに組み込む方法について説明したドキュメントです。「CS+に取り込む方法 (R01AN1826JJ)」は、FIT モジュールを CS+のプロジェクトに組み込む方法について説明したドキュメント です。

## 1.5 プロジェクトについて

本アプリケーションノートには、本アプリケーションをビルド及び評価するための、e<sup>2</sup> studio 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」と、デバッグ設定を保存 した「デバッグ構成」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 プロジェクト設定

	構成名	説明
ビルド構成	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。
デバッグ構成	HardwareDebug (E1)	「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。

また、ターゲット固有の設定は以下のとおりです。

#### 表 1-5 ターゲット固有の設定

項目	設定
ツールチェーン・バージョン	v2.03.00
デバッグ・ハードウェア	E1 (RX)
データ・エンディアン	Little-endian data
ターゲットの選択	R5F564MLCxFC(RX64M LQFP 176pin)
	又は
	R5F571MLCxFC(RX71M LQFP 176pin)
Renesas RTOS サポート(注1)	None

注1:使用するためには、別途 OS 環境をダウンロードする必要があります。



## 2. 開発環境の入手

## 2.1 e<sup>2</sup> studio の入手方法

以下の URL にアクセスし、e2 studio をダウンロードしてください。

http://japan.renesas.com/e2studio\_download

なお、本ドキュメントは e2 studio V4.1.0 以降を使用することを前提としています。V4.1.0 よりも古い Ver. を使用した場合、e<sup>2</sup> studio の一部機能を使用できない可能性があります。ダウンロードする場合、ホームペー ジに掲載されている最新 Ver.の e2 studio を入手してください。

## 2.2 コンパイラパッケージの入手方法

以下の URL にアクセスし、RX ファミリ用 C/C++コンパイラパッケージをダウンロードしてください。

http://japan.renesas.com/e2studio\_download



## 3. プロジェクトの構築

本アプリケーションノートは、環境構築済みのプロジェクトを同梱しています。e2 studio のスマート・ブラウザーを用いてプロジェクトをインポートする手順について、以下に説明します。

## 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。

〒 − クスペース・ランチャー 📃	
ワークスペースの選択	
e2 studio は、ワークスペースと呼ばれるフォルダにプロジェクトを保存します。 このセッションに使用するワークスペース・フォルダを選択してください。	
フークスペース(W): C:¥workspace - 参照(8)	任意のワークスペース
	フォルダを人力します
□ この選択をデフォルトとして使用し、今後この質問を表示しない(U)	
	└── 「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。

● C - e2 studio	
Welcome	
e <sup>2</sup> studio	
Vorkbench Go to the workbench	「Workbench」を クリックします
RENESAS	



## 3.2 プロジェクトの作成

スマート・ブラウザーの機能を使用するときは、対象となるプロジェクトあるいはファイルを選択してお く必要があります。スマート・ブラウザーの機能を使用するために、対象デバイスを指定したプロジェクト を作成します(注1)。

- 注1:ここで作成するプロジェクトは、スマート・ブラウザーを使用するために作成するためのダミープ ロジェクトです。インポートするプロジェクトの設定は「表 1-5 ターゲット固有の設定」を参照し てください。
- 1.  $[ファイル(F)] \rightarrow [新規(N)] \rightarrow [C Project]の順にクリックして新しい C プロジェクトを作成します。 新$ 規プロジェクト作成ウィザードを起動します。

e²	C - e2 studio							
7	マイル(F) 編集(E) ソース(S)	リファクタリング(T) ナビゲート(N)	検索	(A) プロジェクト(P) Re	enesas Views 実行(R	) ウィンドウ	(W) ∕~I	,
	新規(N)	Alt+シフト+N♪	Ċ	Renesas FIT Module				C Project L
	ファイルを開く(.)		<b>B</b>	C Project				
	閉じる(C)	Ctrl+W	E	C++ Project				7777089
	すべて閉じる(L)	Ctrl+シフト+W		Makefile Project with Ex	isting Code			
B	保管(S)	Ctrl+S		ノロシエクド(K)				
	別名保存(A)		<b>C</b> ++	C/C++ プロジェクトに変	陸 (C/C++ ネーチャ-	-を追加)		
C	すべて保管(E)	Ctrl+シフト+S	67	ソース・フォルタ				
	前回保管した状態に戻す(T)			フォルダー				
	at at a		C	ソース・ファイル				
	移動(V)		h	ヘッダー・ファイル				
ď	名前変更(M)	F2	<b>₽</b>	テンプレートからファイル	L			
8	更新(F)	F5	G	クラス				
	行区切り文字の変換(D)	•	Ċ	タスク				
8	印刷(P)	Ctrl+P		その他(0)		Ct	rl+N	
	ワークフペーフの打り株子(い)		1					

2. 任意のプロジェクト名を入力し、 "Renesas RXC Toolchain"を選択します。[次へ(N)>] をクリックし てください。

C プロジェクト		- • •	
<b>c プロジェクト</b> 選択したタイプの C プロジェクトを作成します			
プロジェクト名(E): sample ⑦ デフォルト・ロケーションの使用(D) □ケーション(L): D:₩work¥RDP¥build¥sample ⑦ Create Directory for Project プロジェクトの種類: ● Executable (Renesas) ● Sample Project ● Sample Project ● Debug-Only Project	ツールチェイン: KPIT GNUARM-NONE-EABI Toolchain KPIT GNURL78-ELF Toolchain KPIT GNURX-ELF Toolchain KPIT GNUSH-ELF Toolchain Renesse CCEL Toolchain	参照(近)	
<ul> <li>▶ Construction (Notice Section (Notice Section</li></ul>	Renesas RXC Toolchain Renesas SHC Toolchain Renesas SHC Toolchain	Renesas 選択します	RXC Toolchain] を
(?) < 戻る( <u>B</u> ) 次	へ( <u>N</u> ) > 株了(E)	キャンセル	「次へ( <u>N</u> )」をクリック します



- 「ターゲットの選択:」を設定します。RX64Mの場合「R5F564Mxxxx」としてください。RX71Mの 場合「R5F571Mxxxx」としてください。その他の項目は任意の設定で構いません(注1)。設定が完 了したら[終了(F)]をクリックします。
- 注1:ダミープロジェクト用の設定です。スマート・ブラウザーを使用してインポートするプロジェクト 環境の設定は「表 1-5 ターゲット固有の設定」を参照ください。

e <sup>2</sup> C プロジェクト			
e2 studio - プロジェクト生成 ターゲット固有の設定の選択	č		
<ul> <li>ツールチェーン・パージョン:</li> <li>デパッグ・ハードウェア:</li> <li>データ・エンディアン:</li> <li>ターゲットの選択:</li> <li>Renesas RTOS サポート:</li> <li>構成の選択:</li> <li>マリハードウェア・デパッグ</li> <li>マシミュレーターを使用して</li> </ul>	v2.03.00 E1 (RX) Little-endian data R5F564MLCXFC None : ハードウェアを使 Cデバッグする : シミュレーターを付	R5F564MLCxFC 又は、 R5F571MLCxFC <sup>用してデバッグする</sup> 使用してデバッグする	
<ul> <li>ワリース (デバッグしない</li> <li>ビルド構成は選択したデバック リレーブから選択された最初の材 (R5F564MLCxFC)をもとに、</li> </ul>	<ul> <li>) :デバッグ情報のない</li> <li>グ・モード・オブションのプロジェク 構成)用にビルドされます。選択した ユーザーのためのデバッグ構成が自動</li> </ul>	いプロジェクト トにのみ作成されます。ただしデフォルト .デバイス (RX600) 、デバッグ・ハードウ: 動的に作成されます。	では、そのプロジェクトはアクティブな構成(グ ェア (E1 (RX)) 、およびデバッグ・ターゲット
0	< 戻る(B)	次へ(N) >	「終了(F)」をクリック します <sup>終了(F)</sup> キャンセル

4. [OK]をクリックします。

Project summary for sample	
The following target device settings and files will be generated.	
プロジェクト名: sample プロジェクト・ディレクトリー: D:¥WorkSpace¥e2studio	×
CPU シリーズ: RX600 CPU 型名: RX64M ツールチェーン名: Renesas_RXC ツールチェーン・バージョン: v2.03.00	
生成ファイル:	
Stack File ¥src¥stacksct.h	
Vector Table ¥src¥vector_table.c	-



RX64M, RX71M 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

## 3.3 プロジェクトのインポート

メインプログラムのプロジェクトを、作成したワークスペースにインポートします。

1. プロジェクト・エクスプローラーから「3.2 プロジェクトの作成」で作成したプロジェクトを選択します。

e <sup>2</sup> C - e2 studio	
ファイル(E) 編集(E) ソース(S) リファクタリング(T) ナビゲート(N)	
📑 🕶 🖶 🕼 📥   🛞 🕶 🔦 🕶 🕵 🖻 🗙 💽 💆 🖉 😂 🖄	
<sup>▶</sup> プロジェクト・エクスプローラー ☆ ▶ <sup>▶</sup> sample [HardwareDebug]	プロジェクトを選択します

2. [Renesas Views]→[e2 ソリューション・ツールキット]→[スマート・ブラウザー]の順にクリックし、ス マート・ブラウザーを起動します。

Renesas Views 実行(R) ウィンドウ(W)	^	リレプ(	H)	
C/C++(C)	►	-	5 ( <b>- - - - -</b>	
e2 ソリューション・ツールキット	×	\$	競合状態表示	F
コード生成	►	殟	FIT Configurator	L
デバッグ	×	<b>R</b>	最適化アシスタント	L
パートナーOS	►	۹	スマート・ブラウザー	ł
ルネサス OS	×	ц,	スマート・マニュアル	

3. [スマート・ブラウザー]タブの[アプリケーション・ノート]タブをクリックします。





RX64M, RX71M 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

4. [更新] 🚱 をクリックします。

						クリックします
age 📴 スタック角	産析 👒 スマート・	・ブラウザー 🛛 🦹	📱 周辺機能 🕎 端子配置表 归 端子	配置図 🛃 コード・プレビュー 💯	FIT Configurator	®⊅_ ‡
						更新
アプリケーション・	ノート					
リビジョン	発行日	サンプル・コ	備考			

 本アプリケーションノートを選択し、右クリックします。コンテキストメニューの[サンプル・コード (プロジェクトのインポート)]をクリックします。(注1)

ューザーズ・マニュアル:	(ードウェア テクニカル・アップデート	アプリケーション・	)-r		
13件					
タイトル	ドキュメントNo.	リビジョン	発行日	t	
				C	
No INCOMPANY			a	C	
NUMBER OF TAXABLE PARTY.			2010/07/07	C	カロッカト:
	サンフル・コート (フロシェクトのイン	///-/)	Cor and a maintain	C	
AND REPORTS ADDRESS			20 miles (0)	C	
	Real and March Manager Street Street	100 C 170	ALC: NOT THE OWNER.	C	

注1:一度も My Renesas による認証をしていない場合、ファイルをダウンロードする際に「My Renesas」 ダイアログがオープンします。ルネサス Web サイトで登録しているメール・アドレスとパスワードを 入力してください。

e <sup>2</sup> My Renesas					<b>—</b> ×
My Renesas					
My Renesasに登録いただいたメ	ール・アドレスとパスワードを入力ください	۱.			
スマート・ブラウザーから各種ド	キュメント/ソフトウェアをダウンロードで	きます。			
メール・アドレス:					
パスワード:					
My Renesasへご登録いたた	くと、ツール製品などのダウンロードサー	ビスやメールニュースなどの各種サー	・ビスをご利用いただけるようにな	ります。	
My Renesasへの新規登録は	、 [My Renesasについて]から行えます。				
			My Renesasについて	ОК	キャンセル



6. [同意する]をクリックします。

■2 免責事項 (サンプル・コード)	x
ご注意	
1. 本資料は、お客様に用途に応じた適切な弊社製品をご購入いただくための参考資料であり、本資料中に記載の技術情報について弊社または第 三者の知的財産権その他の権利の実施、使用を許諾または保証するものではありません。	
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例など全ての情報の使用に起因する損害、第三者の知的財産 権その他の権利に対する侵害に関し、弊社は責任を負いません。	
3. 本資料に記載の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、あるいはその他軍事用途の目的で使用しないでください。 また、輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。	
4. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例などの全ての情報は本資料発行時点のものであり、弊社は 本資料に記載した製品またけ仕様等を予告なしに変更することがあります。整社の主導体製品のご購入ませるごで使用に当たりましては、重新に整	Ŧ
同意する 同意しない	

クリックします

7. 本アプリケーションノートを保存します。

e <sup>2</sup> 名前を付けて保存				×
	- >	<b>▼ 4</b> 9	コンピューターの検索	٩
整理 ▼				0
	ヽード ディスク ドライブ (3) ――			
▶ ■ ピクチャ       ▶ ■ ビデオ       ▶ ● ミュージック	ローカル ディスク (C:) 空き領域 62.3 GB/186 GB	ローカル ディ ごろう 日本 空き領域 6.9	イスク (D:) 6 GB/46.2 GB	
<ul> <li>▲ コンピューター</li> <li>▶ ▲ ローカルディス</li> <li>▶ □ ーカルディス</li> <li>▶ □ ーカルディス</li> <li>■ ローカルディス</li> <li>■ □ ーカルディス</li> </ul>	ローカルディスク (Q:) リムーバブル記憶域があるデバイス (	(1)		
ファイル名(N): an_r01an2	608jj0102_rx.zip			-
ファイルの種類(T): [*.*				<b></b>
🍝 フォルダーの非表示			保存(S) キャン	ンセル

8. 「プロジェクト(P):」に表示されている「r\_network\_solution\_rx64m」または「r\_network\_solution\_rx71m」 いずれか一方を選択し、「終了(F)」をクリックします。

- 				
プロジェクトのインポート 既存の Eclipse プロジェクトを検索する	ディレクトリーを選択します。			
◎ ルート・ディレクトリーの選択(T):				▼ 参照(R)
◎ アーカイブ・ファイルの選択(A):	▼ 参照(R)			
プロジェクト(P): ア_network_solution_rx64m (r. ア_network_solution_rx71m (r. いずれか一方を選択します	_network_solution_rx64m 直 _network_solution_rx71m 直	妾アクセス) 妾アクセス)		すべて選択(S) 選択をすべて解除(D) 更新(E)
オプション マネストしたプロジェクトを検索(H) マプロジェクトをワークスペースにコ マークスペースに既に存在するプロ ワーキング・セット 同ワーキング・セットにプロジェクト	ビー(C) ジェクトを隠す(i) を追加(T)			
ワーキング・セット(0):	< 戻る(B)	次へ(N) >	終了(F)	▼ 選択(E) 単ヤンセル
			クリックします	

9. 下図は「r\_network\_solution\_rx64m」を選択した場合です。スマート・ブラウザーを使用するために作成したプロジェクト(ここでは sample)は不要ですので削除してください。

e <sup>2</sup> C - e2 studio				
ファイル(F) 編集(E) ソース(S)	リファ	▼クタリング(T) ナビゲート(N) 検索(A)	プロジェクト(P) Rene	
📬 • 🛛 🖕 🛎   🗞 • 🗞 • 🤅		) 💽 🧐 🥖 😂 🔂 🛨 🖒 👻 🤆	• @ •   🎋 • 💽 • (	
Ъ プロジェクト・エクスプローラー	23	□ 🔄 😜 🗸 🗖 🗖		
F_network_solution_rx64m				
Sample [HardwareDebug		新規(N)	•	
		次ヘジャンプ(I)		
クリックします		新規ウィンドウで開く(N)		
	D	⊐ピー(C)	Ctrl+C	
	ß	貼り付け(P)	Ctrl+V	
	×	削除(D)	削除	クリックします
	<u>_</u>	コンテキストから除去 (	trl+Alt+シフト+下へ	
		ソース	+	
		移動(V)		



#### 3.4 変更情報

本プロジェクトでは、本アプリケーションを構成するために各 FIT モジュールのコンフィグレーションファ イル設定とプロジェクト設定を変更しています。以下に詳細を示します。

なお、本変更情報は、新規にプロジェクトを構築する場合に参照してください。インポートしたプロジェ クトを使用する場合は「4 動作確認」に進んでください。

#### 3.4.1 コンフィギュレーションの変更

本アプリケーションを構成する各 FIT モジュールのコンフィギュレーションファイルを変更します。

コンフィギュレーションファイルの項目と設定内容については、各 FIT モジュールの doc フォルダに入っているマニュアル等を参照してください。

以下にコンフィギュレーションファイルの変更箇所を示します。

#### (1) 割り込みスタックサイズの変更

本アプリケーションは、Ethernet コントローラの割り込みハンドラから、TCP/IP の処理を行っており、約 2.5kbyte の割り込みスタックを必要とします。

r\_bspのコンフィギュレーションファイルで定義されている割り込みスタックサイズを、以下の様に変更してください。

[r\_config/r\_bsp\_config.h]

/\* Interrupt Stack size in bytes. The Renesas RX toolchain sets the stack size
using the #pragma stacksize directive.
 \* If the interrupt stack is the only stack being used then the user will likely
want to increase the default size
 \* below.
 \*/
#pragma stacksize si=0x1000

#### (2) コンペアマッチタイマドライバの設定変更

コンペアマッチタイマの割り込みレベルを、USB ドライバの割り込みレベル (IPR=3) より低く設定します。

[r\_config/r\_cmt\_rx\_config.h]

```
/* The interrupt priority level to be used for CMT interrupts. */
#define CMT_RX_CFG_IPR (2)
```



#### (3) USB ドライバの設定変更

チャネル0を未使用(USB\_NOUSE\_PP)、チャネル1をホスト(USB\_HOST\_PP)にしてください。

#### [r\_config/r\_usb\_basic\_config.h]

```
/*
   Select USB mode to USB IP(USBb)
   USB_HOST_PP : USB Host Mode
   USB_PERI_PP : USB Peripheral Mode
   USB_NOUSE_PP : Not Used (USBb)
*/
   #define USB_FUNCSEL_USBIP0_PP USB_NOUSE_PP
/*
   Select USB mode to USB IP(USBAa/USBA)
   USB_HOST_PP : USB Host Mode
   USB_PERI_PP : USB Peripheral Mode
   USB_NOUSE_PP : Not Used (USBAa/USBA)
*/
   #define USB_FUNCSEL_USBIP1_PP USB_HOST_PP
```

[r\_config/r\_usb\_hmsc\_config.h]

#define USB\_MEDIA\_INITIALIZE(data1) R\_tfat\_disk\_initialize((uint8\_t)data1)



#### (4) T4 の設定変更

T4の設定を以下のように変更します。

t4\_callback 関数の外部参照宣言をコメントアウトし、新たに http\_callback、ftp\_callback、ftp\_data\_callback、 dns\_callback 関数の外部参照宣言を追加してください。

[r\_t4\_rx/src/config\_tcpudp.c]

```
#include "r_t4_itcpip.h"
//extern ER t4_callback(ID cepid, FN fncd , VP p_parblk);
extern ER http_callback(ID cepid, FN fncd , VP p_parblk);
extern ER ftp_callback(ID cepid, FN fncd , VP p_parblk);
extern ER ftp_data_callback(ID cepid, FN fncd , VP p_parblk);
extern ER dns_callback(ID cepid, FN fncd , VP p_parblk);
```

ローカルポートの設定を、以下のように変更してください。

[r\_t4\_rx/src/config\_tcpudp.c]

```
/*** Definition of TCP reception point (only port number needs to be set) ***/
T TCP CREP tcp crep[18] =
{
   /* HTTP server use setting below. */
   \{ 0 \times 0000, \{ 0, 80 \} \},\
   \{ 0x0000, \{ 0, 80 \} \},\
   \{ 0 \times 0000, \{ 0, 80 \} \},\
   \{ 0 \times 0000, \{ 0, 80 \} \},\
   \{ 0x0000, \{ 0, 80 \} \},\
   \{ 0x0000, \{ 0, 80 \} \},\
   /* FTP server use setting below. */
   \{ 0x0000, \{ 0, 21 \} \},\
   { 0x0000, { 0, 20 }}, /* The port number in the active mode is 20, and
the port number in the passive mode is 1024. */
   \{ 0 \times 0000, \{ 0, 21 \} \},\
   \{ 0 \times 0000, \{ 0, 20 \} \},\
                                 /* The port number in the active mode is 20, and
the port number in the passive mode is 1025. */
    \{ 0 \times 0000, \{ 0, 21 \} \},\
    \{ 0x0000, \{ 0, 20 \} \},\
                                 /* The port number in the active mode is 20, and
the port number in the passive mode is 1026. */
   \{ 0 \times 0000, \{ 0, 21 \} \},\
    { 0x0000, { 0, 20 }}, /* The port number in the active mode is 20, and
the port number in the passive mode is 1024. */
   \{ 0 \times 0000, \{ 0, 21 \} \},\
    \{ 0 \times 0000, \{ 0, 20 \} \},\
                              /* The port number in the active mode is 20, and
the port number in the passive mode is 1025. */
   \{ 0x0000, \{ 0, 21 \} \},\
                                 /* The port number in the active mode is 20, and
    \{ 0 \times 0000, \{ 0, 20 \} \},\
the port number in the passive mode is 1026. */
};
```

TCP 通信端点の設定を、以下のように変更してください。

[r\_t4\_rx/src/config\_tcpudp.c]

```
/*** Definition of TCP communication end point
     (only receive window size needs to be set) ***/
T TCP CCEP tcp ccep[18] =
{
   /* { 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 }
   */
   /* HTTP server use setting below. */
   { 0, 0, 0, 0, 1460, http callback },
   { 0, 0, 0, 0, 1460, http_callback },
   { 0, 0, 0, 0, 1460, http callback },
   { 1, 0, 0, 0, 1460, http callback },
   { 1, 0, 0, 0, 1460, http callback },
   { 1, 0, 0, 0, 1460, http callback },
   /* FTP server use setting below. */
   { 0, 0, 0, 0, 128, ftp callback },
                                                                       */
                                             /* FTP control connection
   { 0, 0, 0, 0, 1460, ftp_data_callback }, /* FTP data connection
                                                                       */
   { 0, 0, 0, 0, 128, ftp_callback },
                                             /* FTP control connection
                                                                       */
                                          /* FTP data connection
                                                                       */
   { 0, 0, 0, 0, 1460, ftp data callback },
   { 0, 0, 0, 0, 128, ftp callback },
                                             /* FTP control connection
                                                                       */
   { 0, 0, 0, 0, 1460, ftp data callback },
                                                                       */
                                            /* FTP data connection
                                                                       */
   { 1, 0, 0, 0, 128, ftp callback },
                                             /* FTP control connection
   { 1, 0, 0, 0, 1460, ftp_data_callback },
                                                                       */
                                             /* FTP data connection
   { 1, 0, 0, 0, 128, ftp_callback },
                                             /* FTP control connection
                                                                       */
   */
                                            /* FTP control connection
                                                                       */
   { 1, 0, 0, 0, 128, ftp_callback },
   { 1, 0, 0, 0, 1460, ftp data callback },
                                                                       */
                                            /* FTP data connection
};
```

2-MSL 待ち時間を、1 分から 10ms に変更してください。

[r\_t4\_rx/src/config\_tcpudp.c]

```
/*** 2MSL wait time (unit:10ms) ***/
const UH _tcp_2msl[] =
{
    (1),    /* 10 ms */
    (1),    /* 10 ms */
};
```

UDP 通信端点の設定を、以下のように変更してください。

```
[r_t4_rx/src/config_tcpudp.c]
```

```
/*** Definition of UDP communication end point ***/
T_UDP_CCEP udp_ccep[1] =
{
    /* only setting port number */
    { 0x0000, { 0, 1365 }, dns_callback },
};
```

#### (5) HTTP サーバの設定変更

MAX\_CGI\_FILE を2に設定します。

[r\_config/r\_t4\_http\_server\_rx\_config.h]

```
// If user uses CGI function, please specify cgi file name, and number of cgi files.
#define MAX CGI FILE 2
```

CGI\_FILE\_NAME\_TABLE\_LIST を以下の様に設定します。

[r\_config/r\_t4\_http\_server\_rx\_config.h]

```
/*#define CGI_FILE_NAME_TABLE_LIST ¥*/
/* {"cgi_smpl.cgi", NULL}, ¥*/
extern ER cgi_sample_function(ID cepid, void *res_info);
extern ER cgi_dns_demo_pending_release(ID cepid, void *res_info);
#define CGI_FILE_NAME_TABLE_LIST ¥
    {"cgi_smpl.cgi", cgi_sample_function, NULL}, ¥
    {"dns demo.cgi", cgi dns demo function, cgi dns demo pending release}, ¥
```

HTTP サーバで使用する通信端点の数を、r\_t4\_rx/src/config\_tcpudp.c の tcp\_ccep テーブル数に合わせて 6 個に変更します。

[r\_config/r\_t4\_http\_server\_rx\_config.h]

// set same value number of CEPID in config\_tcpudp.c
#define HTTP TCP CEP NUM 6

#### (6) DNS クライアントの設定変更

動作確認で使用する DNS サーバのアドレスを設定します。任意のアドレスを設定してください(下記は一例です)。

[r\_config/r\_t4\_dns\_client\_rx\_config.h]



RX64M, RX71M 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

#### (7) FTP サーバの設定変更

T4のTCP 通信端点の設定に合わせて、FTP サーバで使用するTCP 通信端点の位置を設定します。 【r\_config/r\_t4\_ftp\_server\_rx\_config.h】

#### (8) FTP/Web サーバ用ファイルドライバの設定変更

USB を使用する設定に変更します。

[r\_config/r\_t4\_file\_driver\_rx\_config.h]



(9) M3S-TFAT-Tiny メモリドライバの設定変更

USB を使用する設定に変更します。

[r\_config/r\_tfat\_driver\_rx\_config.h]

```
/* Number of logical drives to be used.
  Setting to 0 : unused memory
          other : number of logical drives
  (USB and SDHI can be used together.)
*/
#define TFAT USB DRIVE NUM
                              (1)
#define TFAT SDHI DRIVE NUM
                               (0)
#define TFAT_USB_MINI_DRIVE_NUM
                               (0)
/* alocate a drive number
 <valid define>
 TFAT_CTRL USB : for USB
 TFAT CTRL SDHI : for SDHI
 NULL
           : unallocated drive
 MAX 10 drives (TFAT module spec)
 ex.)
    #define TFAT_DRIVE_ALLOC_NUM_0 TFAT_CTRL_USB
    #define TFAT DRIVE ALLOC NUM 1 TFAT_CTRL_SDHI
    #define TFAT DRIVE ALLOC NUM 2 TFAT CTRL SDHI
    #define TFAT DRIVE ALLOC NUM 3 TFAT CTRL USB
*/
#define TFAT DRIVE ALLOC NUM 1 NULL
```

#### (10) Ethernet の設定変更

Ethernet コントローラから PHY IC を制御するために使用する PHY IC のチャネル設定を ch0 に変更します。

[r\_config/r\_ether\_rx\_config.h]

/\* The register bus of PHY0/1 for ETHER0/1 select 0 = The access of the register of PHY uses ETHER0. 1 = The access of the register of PHY uses ETHER1. \*/ #define ETHER\_CFG\_CH0\_PHY\_ACCESS (0) #define ETHER\_CFG\_CH1\_PHY\_ACCESS (0)



## 3.4.2 プロジェクト設定の変更

ビルド時の設定はデフォルト設定から「表 3-1 変更したビルド設定」に示す内容に変更しています。 プロジェクト設定を確認する場合は、以下の手順で行ってください。

1. e<sup>2</sup> studio の対象プロジェクト (r\_network\_solution\_rx64m or r\_network\_solution\_rx71m) を選択し右クリッ クします。その後「プロパティ(R)」をクリックします。

- T_Network_solution_ イル(F) 編集(E) ソー	-ス(S) リファクタリング(T) ナビグ	ート(N) 検索(A) プロジェク	クト
• 8 6 8 • 8 •	<b>≪ ▼ ≪ ⊡</b> ≥ <b>∞</b>		•
プロジェクト・エクスプ	u-ə- x 🗆 🗣	💱 🍸 🗖 🗖 🔂 reset	tpr <u>c</u>
r_network_soluti	on rx64m [HardwareDebud]	260	
	新規(N)	•	
	水ヘシヤンノ(1)		
	新規ウィンドウで開く(N)		
	⊐ピー(C)	Ctrl+C	
Ē	貼り付け(P)	Ctrl+V	
×	削除(D)	削除	
S.	コンテキストから除去	Ctrl+Alt+シフト+下へ	
	ソース	•	
	移動(V)		Z
	名前を変更(M)	F2	۴
<u>r~</u> 3	インポート(I)		
2	エクスポート(0)		i i
	プロミェクトのビルド(B)		
	プロジェクトをクローンにする		1 E
		F5	
<u></u>	えっ() プロジェクトを閉じる(S)	10	
	毎期係なプロジェクトを閉じる(U)		
			-
		*	
		*	
	1,25,992	,	
	ビルドから除外		
	プロファイル(P)	•	
	デバッグ(D)	+	
	実行(R)	+	
	比較対象(A)	+	
	ローカル履歴から復元(Y)		
e²	Renesas Quick Settings	Alt+Q	
e²	Renesas Tool Settings	Alt+T	
*	C/C++ コード解析を実行		
	Team	•	
1	Windows Explorer		
	Command Prompt		
	プロパティ(R)	Alt+Enter	┣━━━━クリックします







3. 「ツール設定」タブから「表 3-1 変更したビルド設定」の内容を参照してください。

項目	変更内容				
Compiler	「インクルード・ファイル・ディレク	各 FIT モジュールで設定;	が必要なインクルードパスを		
-ソース	トリ」にインクルードパスを追加する	設定する。			
		「FIT Configurator」を使 み込む場合、自動で設定	用し、各 FIT モジュールを組 されます。		
Linker	■Little エンディアンの場合	TFAT を使用する際には語			
-入力	"\${workspace_loc:/\${ProjName}/r_tfa t_rx/lib/tfat_rx600_little.lib}"を追加す	(FAT ファイルシステムド	ジライバを使用する際に必須)		
	る(注 1)	「FIT Configurator」を使	用し場合でも、自動で設定さ		
		れません。 <mark>新規にプロジ</mark>	ェクトを作成する場合、ユー		
	■Big エンディアンの場合	ザが設定してください(注	2)		
	"\${workspace_loc:/\${ProjName}/r_tfa t_rx/lib/tfat_rx600_big.lib}"を追加する (注 1)				
	■Little エンディアンの場合	Ethernet を使用する際に	は設定する。		
	"\${workspace_loc:/\${ProjName}/r_t4				
	_rx/lib/T4_Library_rxv2_ether_little.li	「FIT Configurator」を使	用し場合でも、自動で設定さ		
	0} を追加する(注1)	れません。新規にフロジェクトを作成する場合、ユー			
	■Big エンディアンの場合	サか設定してくたさい(注	. 2)		
	■big ± 5 / 1 / 5 0 / 2 bit "\${workspace_loc:/\${ProiName}/r_t4				
	_rx/lib/T4_Library_rxv2_ether_big.lib}				
	"を追加する(注 1)				
Linker	RAM 領域に	Ethernet を使用する際に	は設定する。		
-セクション	B_ETHERNET_BUFFERS_1 セク				
	ション、B_RX_DESC_1 セクショ	「FIT Configurator」を使	用し場合でも、自動で設定さ		
	ン、B_TX_DESC_1 セクションを追 mます	れません。新規にプロジ	ェクトを作成する場合、ユー		
	2月 9 る	サか設定してくたさい。			
		く設定例>			
		- ヤクション・ビューアー・			
			1-1-2 × 2		
		0x0000000	B_ETHERNET_BUFFERS_1		
			B_KX_DESC_1		
			B_IX_DESC_I		
			SI		
			D 1		
			D_1		
			N_1		

#### 表 3-1 変更したビルド設定

(注 1) 各 FIT モジュールを組み込むプロジェクトを作成する際に必要な設定変更です。この設定については 各 FIT モジュールの doc フォルダに入っているマニュアル等を参照してください。

(注 2) e2 studio V4.1 以降では自動設定が可能です。但し、今回使用した FIT モジュールが仕様に対応していいないため、自動設定機能を使用できません。最新の FIT モジュールにて対応予定です。

## 4. 動作確認

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

e <sup>2</sup> C - e2 studio	
ファイル(E) 編集(E) ソース( <u>S</u> ) リファクタリング(T) ナビゲート( <u>N</u>	
<sup>1</sup> プロジェクト・エクスプローラー ※ <sup>□</sup> <sup>□</sup> <sup>□</sup> <sup>□</sup> <sup>□</sup> <sup>□</sup> <sup>□</sup> <sup>1</sup>	
<pre>     r_network_solution_rx64m [HardwareDebug]     mincudes     G r hsn </pre>	ーーーーーー クリックします

2. 「プロジェクト(P)」タブの「プロジェクトのビルド(B)」をクリックします。

ナビゲート(N) 検索(A)	プロ	]ジェクト(P) Renesas Views	実行(R) 🖞	ルン	ドウ(W) ヘルプ(H)
∅ 🖾 🔂 🕶 🔂 🔻		プロジェクトを開く(E) <b>プロジェクトを閉じる(S)</b>			] [[] <b>2 → 주 → ★ ← →</b>
г	<u>61</u> 0	すべてビルド(A) ビルド構成	Ctrl-	-В •	
L		フロジェクトのビルド(B)			
		ワーキング・セットのビルト(W クリーン(N) 自動的にビルド(M)	)	•	
		Make ターゲット		•	
	e <sup>2</sup>	Renesas Tool Settings	Alt	ът	
	e <sup>2</sup>	Renesas Quick Settings	Alt+	Q	
	e²	すべての依存関係を更新	Alt+	·D	
		C/C++ Index		•	
		プロパティ(P)			

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。





## 4.2 デバッグの準備

## 4.2.1 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 4-1 機器構成

-		
No.	機器	補足
1	開発 PC	開発を行う PC です。
2	評価ボード(Renesas Starter Kit+ for RX64M 又は	
	Renesas Starter Kit+ for RX71M)	
3	USB メモリ	FAT、または FAT32 でフォーマットしたもの。
4	クライアント PC	開発 PC で代用可能です。
	● Web ブラウザ	
	<ul> <li>FTP クライアント(FFFTP 等)</li> </ul>	
5	DHCP サーバ、DNS サーバ	DHCP クライアントと DNS クライアントの動
		作を確認する場合に必要です。
6	ネットワーク環境によって以下のいずれかが必要に	ネットワーク環境の例は以下の図を参照して
	なります。	ください。
	● ルータ 1台	
	<ul> <li>スイッチング・ハブ 1台</li> </ul>	
	● LAN ケーブル(ストレート又はクロス)1~3 本	



図 4-1 DHCP サーバ、DNS サーバあり ールータ (DHCP サーバ機能付き)構成



図 4-2 DHCP サーバ、DNS サーバあり -スイッチング・ハブ構成



図 4-3 DHCP サーバ、DNS サーバなし -スイッチング・ハブ構成



図 4-4 DHCP サーバ、DNS サーバなし -LAN クロスケーブル構成



## 4.2.2 評価ボードの設定

本アプリケーションを動作させるための、評価ボードの設定を以下に示します。

- (1) USB 設定
  - USB ch0 のモード(ホスト/ペリフェラル)を設定します。r\_usb\_config.hの 「USB\_FUNCSEL\_USBIP0\_PP」の設定に合わせて、ジャンパ J2 及び J6 を設定します。
  - 2. USB ch1 のモード (ホスト/ペリフェラル) を設定します。r\_usb\_config.h の 「USB\_FUNCSEL\_USBIP1\_PP」の設定に合わせて、ジャンパ J7 及び J9 を設定します。

No.	設定内容	ジャンパ	設定内容	動作確認で
				使用した設定
1	USB0 をホストモードで使用する場合	J2	1-2 をショート	未使用
	(USB_FUNCSEL_USBIP0_PP = USB_HOST_PP)	J6	2-3 をショート	
	USB0 をペリフェラルモードで使用する場合	J2	2-3 をショート	未使用
	(USB_FUNCSEL_USBIP0_PP = USB_PERI_PP)	J6	1-2 をショート	
2	USB1 をホストモードで使用する場合	J7	1-2 をショート	使用
	(USB_FUNCSEL_USBIP1_PP = USB_HOST_PP)	J9	2-3 をショート	
	USB1 をペリフェラルモードで使用する場合	J7	2-3 をショート	未使用
	(USB_FUNCSEL_USBIP1_PP = USB_PERI_PP)	J9	1-2 をショート	

表 4-2 USB 設定一覧



図 4-5 Renesas Starter Kit+ for RX64M/RX71M のジャンパ位置【USB】



## (2) Ethernet 設定

- 1. Ethernet コントローラから PHY IC を制御するために使用する PHY IC のチャネルを指定します。 r\_ether\_rx\_config.h の「ETHER\_CFG\_CH0\_PHY\_ACCESS」と「ETHER\_CFG\_CH1\_PHY\_ACCESS」の 設定に合わせて、ジャンパ J3 及び J4 を設定します。
- 2. Ethernet コントローラ (ET0) を使用する場合、スイッチ、ジャンパを設定します。
- 3. Ethernet コントローラ (ET1) を使用する場合、ジャンパを設定します。

No.	設定内容	ジャン	パ & スイッチ	動作確認で
		No.	設定内容	使用した設定
1	PHY IC を ch0 で制御する	J3	1-2 をショート	使用
		J4	1-2 をショート	
	PHY IC を ch1 で制御する	J3	2-3 をショート	未使用
		J4	2-3 をショート	
2	Ethernet コントローラ(ET0)を使用する場合	J10	1-2 をショート	使用
		J11	2-3 をショート	
		J12	2-3 をショート	
		J13	2-3 をショート	
		J14	2-3 をショート	
		SW5	1 On	
			2 On	
			3 Off	
			4 On	
			5 Off	
			6 On	
			7 Off	
			8 Off	
			9 On	
		014/0	10 Off	
		SW6	1 On	
			2 On 2 On	
			4 011 5 0n	
			6 Off	
			7 On	
			8 Off	
			9 On	
			10 Off	
		SW7	1 On	
			2 Off	
			3 Off	
			4 On	
			5 Off	
			6 Off	
			7 On	
			8 Off	
			9 On	
			10 Off	

#### 表 4-3 Ethernet 設定一覧



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

3	}	Ethernet コントローラ(ET1)を使用する場合	J16	1-2 をショート	未使用
			J18	1-2 をショート	
			J20	1-2 をショート	



図 4-6 Renesas Starter Kit+ for RX64M/RX71M のジャンパ位置【Ethernet コントローラ PHY IC】



図 4-7 Renesas Starter Kit+ for RX64M/RX71M のジャンパ位置【Ethernet コントローラ ET0/ET1】

## 4.2.3 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。ここではWindows7の場合の例を示します。

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。

								x	
	ネッ	ト 🔹 ネットワーク接続 🕨	+ 4	<ul> <li>ネットワ</li> </ul>	ワーク接続	の検索		P	
ファイル(F) 編集(E) 表示(V) ツール	/(T)	詳細設定(N) ヘルプ(H)							
整理 ▼ このネットワーク デバイスを	無効	こする この接続を診断する	この接続の名前	を変更する	»		=	0	
ローカル エリア接続									
Realtek PCIe GBE Family Con.	•	無効にする(B) <b>状態(U)</b> 診断(I)							
	•	ブリッジ接続(G)							
	0	ショートカットの作成(S) 削除(D) 名前の変更(M)							
		プロパティ(R)	3					-	― 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。DHCP サーバの有無に合わせて、以下のように設定し「OK」 をクリックします。

A. DHCP サーバがある場合(IP 自動取得)

インターネット プロトコル バージョン 4 (Tr	CP/IPv4)のプロパティ	? <b>×</b>	
全般(代替の構成			
ネットワークでこの機能がサポートされている場合 きます。サポートされていない場合は、ネットワーク てください。	は、IP 設定を自動的に取得 )管理者に適切な IP 設定を	することがで 問い合わせ	
<ul> <li>IP アドレスを自動的に取得する(Q)</li> </ul>			
次の IP アドレスを使う(S):			
IP アドレス(I):			
サブネット マスク(山):			
デフォルト ゲートウェイ( <u>D</u> ):			
◎ DNS サーバーのアドレスを自動的に取得す	# <u>3(B)</u>		
○ 次の DNS サーバーのアドレスを使う(E):			
優先 DNS サーバー( <u>P</u> ):			
代替 DNS サーバー( <u>A</u> ):			
□ 終了時に設定を検証する(L)		没定(⊻)	
	ОК	**>セル	── クリックします

B. DHCP サーバがない場合 (IP 固定)

全般 ネットワークでこの機能がサポートされている場合は、IP 設定を自動的に取得することがで	
ネットワークでこの機能がサポートされている場合は、IP 設定を自動的に取得することがで	
きます。サポートされていない場合は、ネットワーク管理者に通切な IP 設定を問い合わせ てください。	
<ul> <li>IP アドレスを自動的に取得する(Q)</li> </ul>	
<ul> <li>⑥ 次の IP アドレスを使う(S):</li> </ul>	
IP アドレス(D: 192 . 168 . 0 . 100	
サブネット マスク(山): 255 . 255 . 0	
デフォルト ゲートウェイ(型):	
ONS サーバーのアドレスを自動的に取得する(E)	
③ 次の DNS サーバーのアドレスを使う(E):	
優先 DNS サーバー(P):	
代替 DNS サーバー( <u>A</u> ):	
○ 終了時に設定を検証する(L)	
<u>ОК</u>	ー クリックします

#### 4.2.4 USB メモリの準備

事前に USB メモリヘ HTML コンテンツを格納します。

1. プロジェクト内の「src」フォルダを開き、その中にある「contents」フォルダを開きます。「contents」 フォルダに入っている「contents.zip」を開き、「contents」フォルダを USB メモリにコピーします。





RX64M, RX71M 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

## 4.3 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. 評価ボード(Renesas Starter Kit+ for RX64M 又は Renesas Starter Kit+ for RX71M)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。

) (	実行	(R) ウィンドウ(W) ヘルプ(H)		
<b>*</b>	<b>Q</b>	実行(R)	Ctrl+F11	
ŕ	*∿	デバッグ(D)	F11	
		実行履歴(T)	۱.	
		実行(S)	+	
		実行構成(N)		
		デバッグ履歴(H)	•	
		デバッグ(G)	+	
		デバッグの構成(B)		―― クリックします
	Θ	ブレークポイントの切り替え(K)	Ctrl+Shift+B	
	Θ	行ブレークポイントの切り替え(L)		
	Θ	メソッド・ブレークポイントの切り替え(M)		
	65j	監視ポイントの切り替え(W)		
	R	すべてのブレークポイントをスキップ(I)		
	*	すべてのブレークポイントを削除(V)		
		ブレークポイント型(R)	۲.	
	<b>Q</b>	外部ツール(E)	+	



4. 「Renesas GDB Hardware Debugging」の「r\_network\_solution\_rx64m HardwareDebug 又は r\_network\_solution\_rx71m HardwareDebug」をクリックし、「デバッグ」をクリックします。 「Debugger」タブをクリックし、「Connection Setting」タブをクリックします。 「EXTAL 周波数」を「24.0000」に修正し、「エミュレータから電源を供給する」を「いいえ(注1)」 に変更します。 完了したら「デバッグ(D)」をクリックします。

注1:外部電源を使用する場合の設定です。エミュレータから電源を供給する場合は「はい」を選択 してください。

● <sup>2</sup> デバッグ構成			
構成の作成、管理、および実行			
○         ●         ⇒         ○           ○         ////////////////////////////////////	名前(N): r_network_solution_rx64m HardwareDebu	ug □ 共通(C) t Device: R5F564ML	
<ul> <li>C Debug-only</li> <li>C Debug-only</li> <li>C GDB Simulator Debugging (F</li> <li>C GDB /(-ドウェア・デ/(ッギ)</li> <li>M GHS Local ⊄/C++ Launch</li> </ul>	GDB Settings Connection Settings デバック・ 4 クロック メイン・クロック・ソース	ソール設定 EXTAL -	クリックしょう
<ul> <li>■ E: Refless GUB hardware Deb</li> <li>■ C: Renesas Simulator Debuggir</li> <li>■ 起動グループ</li> </ul>	EXTAL 周波数(MH2) 内部フラッシュメモリー書き換え時にクロック <b>4 ターケット・ボードとの接続</b> エミュレーター 接続タイプ	24.0000 ケ・ソ:(はい ・ (Auto) ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	「24.0000」に降 正します
	JTag クロック周波数[MHz] Fine ボーレート[Mbps] ホット・ブラグ ▲ 電源	16.5 ー 2.00 ー いいえ ー	「いいえ」に変更
	エミュレーターから電源を供給する (MAX 20 供給電圧 <b>4 CPU 動作モード</b> レジスター設定	00mA) いしえ	します
< Ⅲ		適用(Y) 前回保管した状態に戻す(V)	
?		デバッグ(D) 見じる	―― 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。

e <sup>2</sup> パース/	クティブ切り替えの確認
?	この種類の起動は、中断時に デバッグ パースペクティブが開くように構成されています。
	このデバッグ・パースペクティブは、アプリケーション・デバッグをサポートするために設計されています。これには、デバッグ・スタック、変数、お よびブレークポイント管理を表示するビューが組み込まれています。
	このパースペクティブを開きますか?
■ 常に	の設定を使用する( <u>R</u> )
	(まし)( <u>Y</u> ) しいいえ( <u>N</u> )
	et パースへ ? 常にこ



## 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

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。

e <sup>1</sup> デバッグ - r_network_solution_rx64m/r_bsp/board/rskrx64m/resetprg.c - e2 studio						
1 + 日 氏 白   彩 + ⑤ * ⑥ 1   2   2   2   1   1   1   1   1   1						
* テバッグ ② * ①、 netvork_solution_rx64m HardwareDebug [Renesas GDB Hardware Debugging] * ② r_netvork_solution_rx64m.x [1] * ② Thread [1] 1 (single core) (Suspended : シグナル : SIGTRAP:Trace/breakpoint trap)	M* 変数 33 % ブレークボイント IIII レジスター 副 F 名前 タイプ	Eジュール 영式 ● イベントボイント 〒10 Register				
E PowerON_Reset_PC() at resetprg.c:122 0xfff9255e     J gdb     GDB server	к (Ш					
C restprg. 3 main. (http:sever.)  223 224 224 225 225 225 225 225 225 225 225		<ul> <li>         ■ # P2&gt;542 * D2&gt;29&gt;54&gt;.10220 % ■ □     </li> <li>         &gt; # r_network_solution_rxs4m     </li> </ul>				
🖳 ニンソール 🛛 🖉 Tasks 🕃 Renesas Cov 🔋 Memory Usa 🕐 Performanc 🕐 Profile 🖏 Real-time C 👒 Trace 🔿 Visua	il Expre 🖹 Problems 🕡 Executables 🔋 メモリー 🛷 移	£索 💯 周辺機能 🧾 端子配置表 🧾 端子配置図 🥈 コード・プ 🙄 🗖				
r_network_solution_nx64m HardwareDebug [Renesas GDB Hardware Debugging] gdb						
<pre>monitor set_io_access_indth, Wi, j0386-91306, 91316, 91326-91314, 91326-91324, 91336-91334, 91346-91345, 91356-91315, 91356-91315, 91356-9136, 91366-91364, 92366-92364, 92266-92264,</pre>						
The target endianness is set automatically (currently little endian)						
		•				

5. 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。

e <sup>2</sup> デバッグ - r_network_solution_rx64m/r_bsp/board/rskrx64m/resetprg.c - e2 studio	
ファイル(E) 編集(E) ソース(S) リファクタリング(T) ナビゲート(N) 検索(A) プロジェクト(P) 実行(R) ウィンドウ(W	
■ → 🖩 🦷 ≜   ⊗ → <b>%</b> → 🗟   🕲   🖉   🕬   ■   ≈   ≈   ∞   ∞   ∞   ∎   <del>      = 21   ☆ → Q →</del>	ーーーー クリックします
🎄 🖏 🔻 🍇 🍓 🛃 🚺 🙀 💆 🖉	
🔺 💽 r_network_solution_rx64m HardwareDebug [Renesas GDB Hardware Debugging]	
Image: A state of the state	
⊿ 🔊 Thread [1] 1 (single core) (Suspended : シグナル : SIGTRAP:Trace/breakpoint trap)	
PowerON_Reset_PC() at resetprg.c:122 0xfff9255e	
pdb gdb	
GDB server	
Resetprg.c ⋈	
⇒ 122 fff9255e ⊖ void PowerON_Reset_PC(void)	
123 1 124 /* Stack pointers are setup prior to calling this function - see comments above	

6. 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。

## 4.3.1 HTTP サーバ機能の確認

1. クライアント PC で Web ブラウザを起動し、LAN ケーブルを接続したポートに合わせて、以下のアドレスを入力します。

Ethernet ポート番号	DHCP サーバ	HTTP サーバアドレス
0	なし	http://192.168.0.3
		(コンフィキュレーションで変更可能)
	あり	DHCP サーバから取得
1	未対応	http://192.168.0.10

【注】HTTP 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。

DHCP サーバから取得した IP アドレスを確認する手順を以下に示します。

デバッガ接続後、src/main.c を開きます。

関数 main()内の 159 行目にある関数 nop()にブレークポイントを設定します。

c resetprg.c	le main.c ∞	
150 151 <b>ffc024a8</b> 152	<pre>if (!r_dhcp_open(&amp;dhcp, (unsigned char*)tcpudp_work, &amp;_myethaddr[0][0])) {</pre>	
153 ffc024c5 154	<pre>set_tcpudp_env(&amp;dhcp); }</pre>	
155 <b>ffc024c8</b> 156	CloseTimer();	
157 ffc024cc 158	<pre></pre>	
159 ffc024d0 160	nop(); ブレークポイントを設定します	
161 162	/* Get the size of the work area used by the T4 (RAM size). */	
163 ffc024d1 164 ffc024d5 165	<pre>ramsize = tcpuop_get_ramsize();</pre>	
166 167	/* Then reserve as much memory array for the work area as the size indicated by the returned value. */	
168 <b>ffc024db</b> 169 170	<pre>for( ;; ); }</pre>	
171	/* Initialize the TCP/IP */	
	1	



プログラムを実行し、設定したブレークポイントで停止したのを確認後、変数 dhcp をウォッチウィンドウで確認します。

・ 変数 dhcp の各メンバーに設定される情報 ipaddr : IP アドレス ipaddr : IP アドレス maskaddr : サブネットマスク gwaddr : ゲートウェイアドレス dnsaddr : Primary DNS サーバの IP アドレス dnsaddr2 : Secondary DNS サーバの IP アドレス

🗱 😵 😋 ブレークポイント 🔠 レジスター 🛋 モジュール 😭 式 🥐 イベントポイント 📗 IO Registers

名前	タイプ	値
🔺 🥭 dhcp	DHCP	{}
🔺 🚐 ipaddr	uint8_t [4]	0x2802
(≫= ipaddr[0]	uint8_t	192 'タ'
(≫= ipaddr[1]	uint8_t	168 '4'
(×)= ipaddr[2]	uint8_t	0 '¥0'
(x)= ipaddr[3]	uint8_t	3 '¥003'
> 🍃 maskaddr	uint8_t [4]	0x2806
gwaddr	uint8_t [4]	0x280a
> 🍋 dnsaddr	uint8_t [4]	0x280e
dnsaddr2	uint8_t [4]	0x2812
> 🍃 domain	char [20]	0x2816
macaddr	uint8_t [6]	0x282a

Web ブラウザ上に、USB メモリ内のルートディレクトリにあるファイルの一覧が表示されます。 Name はファイル名を表し、Last modified は最終更新日を表し、Size はディレクトリである場合には(dir) と表し、Byte でファイルサイズを表します。Parent Directory をクリックすれば一つ上のディレクトリ に移動します。

🗲 🕘 👌 🖉 http:/	//192.168.11.3/		۰ م	<i>(</i> 192.168.11.3	×	6 🚖 🗐
ファイル(E) 編集(E)	表示(V) お気に入り(4	) ツール(I)	ヘルプ(圧)			
Index of /						
Name	Last modified	Size				
CONTENTS		(dir)				
Renesas Exhedded Web	Server					
Nellesits Category 100						



2. 「CONTENTS」をクリックし、その中にある「DEMO.HTM」をクリックすると以下のような画面が 表示されます。

	Renesas TCP/IP(T4) Web Server Demonstration Program
RSK	
RSK CPU BOARD	LED Control DEMO
LED1 switch LED2 switch LED	3 switch LED4 switch
www.renesas.com Corrversion	

3. CGI 機能を使った LED 制御を確認します。 「LEDx switch」ボタンを押すと、ボード上の LED を制御(点灯/消灯)することが出来ます。

RSK CPU BOARD LED Control DE	EMO
LED1 switch LED2 switch LED3 switch LED4 switch	クリックします

4. CGI機能を使った DNS クライアント機能を確認します。

「■DNS DEMO」のテキストボックスに任意のドメイン名(例:www.renesas.com)を入力し、Conversion ボタンを押下します。

DNS DEMO	
www.renesas.com	ビッノンタセス
Conversion	ーー・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
	します

以下のような IP アドレス確認画面が表示されます。



## 4.3.2 FTP サーバ機能の確認

- 1. クライアントPCでFTPクライアントソフト(FFFTP等)を起動します。
- 2. 「r\_t4\_ftp\_server\_rx\_config.h」に設定されているユーザ名とパスワードを使用し、LAN ケーブルを接続したポートに合わせて以下の IP アドレスに接続します。

Ethernet ポート番号	DHCP サーバ	FTP サーバアドレス
0	なし	192.168.0.3 (コンフィギュレーションで変更可能)
	あり	DHCP サーバから取得
1	未対応	未対応

#### 以下の様に USB メモリ内のルートディレクトリにあるファイルの一覧が表示されます。

B RX64M_Hub (*)	- FFFTP								• ×
接続(E) コマンド(C) ブックマーク(B) 表示(V) ツール(I) オプション(Q) ヘルプ(H)									
🧚 🗶 🐥 🏠 🗙 🖸 🏠 🗿 📴 Eo Jis 🇮 册 🥑 💷 🅅 😫 🔗									
E									
名前	日付	サイズ	種類	名前	日付	サイズ	種類	属性	所有者
🗅 epuninst.exe	2012/03/27	136,050	exe	CONTENTS	2014/04/10 12:14	<dir></dir>		rwxr	root
D FFFTP.chm	2010/02/08	262,974	chm						
🗅 FFFTP.exe	2010/10/24	425,984	exe						
FFFTP.txt	2010/10/24	6,135	txt						
history.txt	2010/10/24	47,714	txt						
D JRE32.DLL	2010/02/02	27,648	DLL						
			1.						
•	III				III				•
接続しました.									*
150 Opening AS	CII data conn	oction							
226 Transfer complete									
ファイル一覧の取得は正常終了しました.(61 Bytes)									
ローカル、選択の低	() Bytes)	<b>D</b> - <b>b</b>	山空 10	17837 20M Bytee	送待ちファイルの個	1			- Te
	a (o bytes)	0-1	NT IC	7037.2311 Dytes #					



## 5. ネットワークミドルウェア仕様

本アプリケーションで使用している DHCP クライアント、DNS クライアント、HTTP サーバ、FTP サーバ の仕様に関しては、各 FIT モジュールに付属しているドキュメントを参照してください。



## 6. メインプログラム仕様

## 6.1 ファイル構成

メインプログラムはプロジェクトに同梱しています。メインプログラムのソースファイルは src フォルダに入っています。

メインプログラムのファイル一覧を以下に示します。

フォルダ名	ファイル名	内容
src	main.c	メインソースファイル
	led.c	LED 初期化処理ソースファイル
	led.h	LED 初期化処理ヘッダファイル
	r_http_server_cgi_sample.c	CGI サンプルソースファイル
	r_sys_time.c	HTTP サーバ用システムタイマソースファイル
	r_sys_time.h	HTTP サーバ用システムタイマヘッダファイル
	r_usb_hmsc_api.c	USB ドライバ呼び出し処理ソースファイル
	r_usb_hmsc_api.h	USB ドライバ呼び出し処理ヘッダファイル
src¥contents	contents.zip	USB メモリに格納する HTML コンテンツの ZIP ファイル

表 6-1 メインプログラムファイル一覧



## 6.2 モジュール一覧

メインプログラムのモジュール一覧を以下に示します。

表 6-2 メインプログラムモジュール一覧

ファイル名	モジュール名	内容
main.c	main	メインプログラムのメイン処理。
		各 FIT モジュールの初期化処理を呼び出し、DNS クラ
		イント、HTTP サーバ、FTP サーバ、USB ドライバ及
		び Ethernet ドライバのメイン処理を駆動する(無限
		ループによる周期起動)。
	set_tcpudp_env	DHCP サーバから取得した IP アドレスなどのネット
		ワーク情報をT4の環境変数に設定する。
	usb_cpu_FunctionUSB0IP	USB ch0 のピン設定及びポート初期化処理
	usb_cpu_FunctionUSB1IP	USB ch1 のピン設定及びポート初期化処理
r_usb_hmsc_api.c	usb_cstd_IdleTaskStart	Low Power Mode で使用するアイドルタスクを起動す る。
	usb_cstd_IdleTask	Low Power Mode で使用するアイドルタスク
		ホスト動作では処理なし。
	usb_hmsc_task_start	HMSC ドライバ起動処理
		USB IP の初期化やクラスドライバの登録を行う。
	usb_apl_task_switch	非 OS 環境用の各 USB ドライバタスクのスケジューリ ング処理
	usb hapl task start	HMSC ドライバアプリケーションタスクを起動する。
	usb hmsc DummyFunction	HMSC ドライバ用ダミー関数
	usb hmsc DriveOpen	HMSC ドライバオープン処理。
	usb hapl registration	HMSC ドライバを登録する。
	usb hmsc apl init	HMSCドライバアプリケーションタスクの内部変数の
		初期化を行う。
	usb_hmsc_StrgCommandRes ult	R_usb_hmsc_StrgDriveSearch()のコールバック処理
	usb_hmsc_SampleAplTask	HMSC ドライバアプリケーションタスク処理
		USB メモリを検出し、ファイルシステムをマウントす
		る。
led.c	led_init	LED 初期化
r_http_server_cgi_	—	HTTP サーバモジュールに付属のドキュメント「RX
sample.c		ファミリ 組み込み用 TCP/IP M3S-T4-Tiny を用いた
		Web サーバモジュール Firmware Integration
		Technology/R20AN0075JJ」を参照してください。
r_sys_time.c	get_sys_time	HTTP サーバモジュールに付属のドキュメント「RX
		ファミリ 組み込み用 TCP/IP M3S-T4-Tiny を用いた
		Web サーバモジュール Firmware Integration
		Iecnnology/R2UANUU/5JJ」を参照してくたさい。
	start_system_time	HIIPサーハで使用するタイマの起動
	update_sys_time	HTTP サーバで使用するタイマの割り込みハンドラ
	stop_system_time	HTTP サーバで使用するタイマの停止



## 6.3 処理フロー

メインプログラムの各モジュールの処理フローを以下に示します。

#### (1) main()

ボードサポートパッケージ(BSP モジュール)のスタートアップルーチンから最初に呼ばれるメイン関数 です。

各ドライバと T4 の初期化を行い、無限ループ処理内で Ethernet ドライバのリンクアップと、DNS クライアント、HTTP サーバ、FTP サーバのメイン処理と USB ドライバのスケジュール処理を定期的に呼び出します。





## (2) usb\_cstd\_ldleTaskStart

USB ドライバ処理のアイドルタスクを起動します。

usb_cstd_IdleTaskStart	
R_usb_cstd_SetTaskPri	// USBサンプルアプリケーション アイドルタスクの優先度設定
USB_SND_MSG	// USBサンプルアプリケーション アイドルタスクヘ起動メッセージ送信
return	

#### (3) usb\_cstd\_ldleTask

USB ドライバ処理のアイドルタスクです。

(usb_cstd_IdleTask)	
return	



#### (4) usb\_hmsc\_task\_start

USB ドライバ内の各タスクの起動と、クラスドライバの登録を行い、USB メモリのマウント処理用タスク を起動します。





#### (5) usb\_apl\_task\_switch

USB ドライバのスケジュール処理です。



#### (6) usb\_hapl\_task\_start

USB メモリのマウント処理用タスクの初期化を行います。





#### (7) usb\_hapl\_registration

クラスドライバの登録処理です。



#### (8) usb\_hmsc\_apl\_init

USB メモリのマウント処理用タスクのシーケンス処理変数を初期化します。



#### (9) usb\_hmsc\_DummyFunction

クラスドライバ登録時に指定する suspend と resume 用のダミー関数です。





#### (10) usb\_hmsc\_DriveOpen

**USB**メモリが挿入されたときに、USB ドライバから呼び出されるコールバック関数です。サンプルアプリケーションタスクに対し、USB\_HMSC\_DRIVE\_OPEN メッセージを送信します。



(11) usb\_hmsc\_SampleAplTask

サンプルアプリケーションタスク処理です。usb\_hmsc\_DriveOpen 関数から送信された USB\_HMSC\_DRIVE\_OPEN メッセージを受信し、マウント可能なドライブの検索を行います。

また、usb\_hmsc\_StrgCommandResult 関数から送信された USB\_HMSC\_DRIVEMOUNT メッセージを受信し、 ファイルシステムへのマウントを行います。





#### (12) led\_init

Renesas Starter Kit+ for RX64M or Renesas Starter Kit+ for RX71M 上の LED を使用するための初期化処理を 行います。

led_init	
LEDポート初期化	
全LED消灯	
return	

#### (13) set\_tcpudp\_env

DHCPサーバから取得したIPアドレスなどのネットワーク情報をT4の環境変数tcpudp\_envに設定します。



#### (14) usb\_cpu\_FunctionUSB0IP、usb\_cpu\_FunctionUSB1IP

USB の ch0 と ch1 を使用するためのピンファンクションと IO ポートの初期化処理を行います。





#### (15) start\_system\_time

HTTP サーバのシステム時刻用のコンペアマッチタイマ(10ms 周期)を起動します。



#### (16) update\_sys\_time

HTTP サーバのシステム時刻をカウントアップする割り込みハンドラです。

update_sys_time	)
10ms単位でシステム時刻を カウントアップする。	
return	

#### (17) stop\_system\_time

HTTP サーバのシステム時刻用のコンペアマッチタイマを停止します。





## 7. ユーザ定義関数について

ユーザ定義関数は、ユーザのシステム環境に合わせて、ユーザが記述する必要があるコードです。FIT モジュールの中には、ユーザ定義関数を必要とするものがあります。

本パッケージでは、以下のユーザ定義関数のサンプルを収録しています。ユーザ定義関数の仕様について は、対応する FIT モジュールのマニュアル等を参照してください。

表 7-1 ユーザ定義関数一覧

ユーザ定義関数	ファイル名	FIT モジュール名	ドキュメント名/型番
システムタイマインタフェース	r_sys_time.c	r_t4_http_server_rx	RX ファミリ 組み込み用 TCP/IP
			M3S-T4-Tiny を用いた Web サーバ
			モジュール Firmware Integration
			Technology/R20AN0075JJ
			「4.2 システムタイマ関連」

## 8. 補足

## 8.1 USB ドライバの制限事項

USB の ch0 と ch1 の両方をホストモードに設定した場合、ch0 側でしか USB メモリを認識しません。ch1 側をホストモードで使用したい場合は、ch0 側を未使用かペリフェラルモードに設定してください。設定例は「3.4.1(3) USB ドライバの設定変更」を参照してください。

### 8.2 本アプリケーションの制限事項

プログラム動作後に USB メモリを抜いて、再度挿入した場合、USB メモリが認識されません。プログラム を再起動してください。

## 8.3 DHCP クライアント及び FTP サーバの制限事項

DHCP クライアント及び FTP サーバは Ethernet の ch0 側でのみ動作します。

# 8.4 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 : <u>http://japan.renesas.com/products/tools/evaluation\_software/index.jsp</u>



## 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/

お問合せ先

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

		改訂内容	
Rev.	発行日	ページ	ポイント
1.00	2014.09.01	_	初版発行
			アプリケーションノート番号:R01AN2153JJ0100
1.01	未発行	_	R01AN2153JJ0100 をベースに、DHCP クライアント、DNS ク
			ライアント、FTP サーバに対応
1.02	2015.10.30	—	R01AN2153JJ0100 をベースに RX71M を追加
			アプリケーションノート番号を R01AN2608JJ0102 に変更
1.03	2016.02.29	1	使用した RX Driver Package 環境 にて、
			ドキュメント番号の rev 番号を削除
		4	1.2 動作環境 にて、
			ドキュメント番号の rev 番号を削除
		6	1.3 モジュール構成 表 1-3 モジュールー覧 にて、
			「Rev.」 元は、「バージョン」であった。
			「メインプログラム」 元は、「メインプログラム FIT モジュー
			ル」であった。
			メインプログラムの Rev. 元は、1.02 であった。
		_	アプリケーションノート英語版の同梱と FIT モジュールの更新

## 製品ご使用上の注意事項

E.

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】雷源投入時は、製品の状態は不定です。
	電源投入時には、ISI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
	→。 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
	の状能は保証できません。
	の K ふ は K 品 と こ こ こ こ こ こ こ 。 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
0.	「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に切り替えてくださ
	◇ ∘ 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
0.	【注音】 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。
	▼。 同じグループのマイコンでも型名が違うと 内部ROM レイアウトパターンの相違たどに上り 雷
	高のないの、「コンモの主心を見て、所聞いい、レーノノーバックの相違なとにより、电気的な性の筋囲で、 特性値、動作マージン、 ノイズ耐害、ノイズ転射害たどが思たスピーがなりましま
	スルリフウリエンンキヒロ゙ヒ、フゥ」ニロに、刧トド //、/ヿハ咖里、/ヿハ釉カリ里なこが夫なの物口がのりよ

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ものですが、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テータ、図、表、フロクラム、アルコリスム、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、者作権その他の知的財産権
	に対する使者に関し、当社は、何らの貢仕を負うものではめりません。当社は、本質料に基づさ当社または第二者の特許権、者作権その他の知的財産権を何ら許 業まるまのでけなりません。
4	出するものではのうまされ。 当社創品を改造 改変 複創等しないでください かかろ改造 改変 複創等により生じた損害に関し 当社は 一切その青任を負いません
5	当社会協定に成長に成長である。それにているがからのほどに成長になり上した時日に戻して当社は、「新たの費任を受いたという。
0.	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用途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が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。

## RENESAS

ルネサスエレクトロニクス株式会社

http://www.renesas.com

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/

■営業お問合せ窓口

ſ

※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。