

RX62N グループ、RX63N グループ

組み込み用 TCP/IP M3S-T4-Tiny を用いたサンプルプログラム

要旨

本資料は、RX62N 及び RX63N で動作する TCP/IP と各ドライバについて説明します。本資料を導入ガイドと呼びます。

RX ファミリ組み込み用 TCP/IP M3S-T4-Tiny（以下、T4 と略します）は、ルネサスマイコンで動作する組み込み用 TCP/IP プロトコルスタックです。T4 はライブラリ形式で提供され、ユーザプログラムに組み込むことで簡単に TCP/IP 機能を付加することが出来ます。T4 を使用するためのドライバのプログラムには、CMT ドライバ、システムタイマー、イーサネットドライバ、T4 ドライバと共通ドライバをインクルードしています。

各種 [Renesas Starter Kit](#) 同梱の CPU ボードで簡単に TCP/IP 通信の動作確認可能なサンプルを用意しております。このサンプルはネットワーク接続方法、PC の設定方法、CPU ボードの設定方法について確認することが出来ます。

T4 に関する最新情報は以下 URL をご参照ください。

<https://www.renesas.com/mw/t4>

エコーサーバサンプル : R20AN0051

T4 は、Firmware Integration Technology(FIT)モジュールとして提供されます。FIT の概念については以下 URL を参照してください。

<https://www.renesas.com/ja-jp/solutions/rx-applications/fit/about-fit.html>

動作確認デバイス

RX62N グループ

RX63N グループ

対象コンパイラ

- ・ Renesas Electronics C/C++ Compiler Package for RX Family
- ・ GCC for Renesas RX
- ・ IAR C/C++ Compiler for Renesas RX

各コンパイラの動作確認内容については「2.動作確認条件」を参照してください。

目次

1. 概要	3
1.1 概要	3
2. 動作確認条件	4
3. 関連ドキュメント	6
4. ハードウェア説明	7
4.1 RSK+RX62N/RSK+RX63N を使用する場合の端子設定例	7
5. ソフトウェア説明	9
5.1 サンプルプログラム構成	9
5.2 サンプルソース	13
5.2.1 TCP エコーバックサーバ関数（ブロッキングコールの場合）のフロー	13
5.2.2 TCP エコーバックサーバ関数（ノンブロッキングコールの場合）のフロー	13
5.2.3 UDP エコーバックサーバ関数（ブロッキングコールの場合）のフロー	13
5.2.4 UDP エコーバックサーバ関数（ノンブロッキングコールの場合）のフロー	13
5.3 RX62N/RX63N 用サンプルプログラム環境	20
5.3.1 ソフトウェア構成	20
5.3.2 CS+プロジェクトへの変換方法	21
5.4 サンプルプログラム動作確認	22
5.4.1 環境構築	22
5.4.2 TCP 接続を確認する	26
5.4.3 UDP 接続を確認する	28
5.5 コンパイル時の設定	29
5.6 セクション配置	31
5.6.1 CCRX for Renesas RX のセクション設定例	31
5.6.2 GCC for Renesas RX のセクション設定例	32
5.6.3 IAR C/C++ Compiler for Renesas RX のセクション設定例	34
5.7 オプション設定メモリ	36
6. サンプルコード	37
7. 注意事項	37
7.1 T4 ライブラリ	37
7.2 サンプルプログラム	37
改訂記録	38

1. 概要

1.1 概要

本プログラムは RX62N グループ、RX63N グループ向けのサンプルで、TCP / IP プロトコルとデバイスドライバ（共通ドライバ、CMT ドライバ、イーサネットドライバ、システムタイマー、T4 ドライバを含む）を提供します。

2. 動作確認条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表 2-1 動作確認条件(RX62N)

項目	内容
使用マイコン	R5F562N8BDBG (RX62N グループ)
動作周波数	メインクロック: 12 MHz PLL: 12 MHz (メインクロック1 分周) システムクロック (ICLK): 96 MHz (メインクロック8通倍) 周辺モジュールクロック(PCLK): 48 MHz (メインクロック4通倍) 外部バスクロック(BCLK): 24 MHz (メインクロック2通倍)
動作電圧	3.3V
統合開発環境	ルネサスエレクトロニクス製 e ² studio V7.6.0 IAR Embedded Workbench for Renesas RX 4.12.1
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V3.01.00 コンパイルオプション: 統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99 GCC for Renesas RX 4.8.4.201902 コンパイルオプション: 統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99 IAR C/C++ Compiler for Renesas RX version 4.12.1 コンパイルオプション: 統合開発環境のデフォルト設定
iodefine.h のバージョン	Version 1.4
エンディアン	リトルエンディアン・ビッグエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
サンプルコードのバージョン	Version 1.00
使用ボード	Renesas Starter Kit+ for RX62N (製品型名: R0K5562Nxxxxxxx)

表 2-2 動作確認条件(RX63N)

項目	内容
使用マイコン	R5F563NFDDFC (RX63N グループ)
動作周波数	メインクロック: 12 MHz PLL: 192 MHz (メインクロック1 分周16逓倍) システムクロック (ICLK): 96 MHz (PLL2分周) 周辺モジュールクロックA(PCLKA): 96 MHz (PLL2分周) 周辺モジュールクロックB(PCLKB): 48 MHz (PLL4分周) 外部バスクロック(BCLK): 48 MHz (PLL4分周) FlashIFクロック(FCLK): 48 MHz (PLL4分周) IEBUSクロック(IECLK): 48 MHz (PLL4分周)
動作電圧	3.3V
統合開発環境	ルネサスエレクトロニクス製 e ² studio V7.6.0 IAR Embedded Workbench for Renesas RX 4.12.1
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V3.01.00 コンパイルオプション: 統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99 GCC for Renesas RX 4.8.4.201902 コンパイルオプション: 統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99 IAR C/C++ Compiler for Renesas RX version 4.12.1 コンパイルオプション: 統合開発環境のデフォルト設定
iodefine.h のバージョン	Version 1.8A
エンディアン	リトルエンディアン・ビッグエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
サンプルコードのバージョン	Version 1.00
使用ボード	Renesas Starter Kit+ for RX63N (製品型名: R0K50563Nxxxxxx)

3. 関連ドキュメント

RX ファミリ 組み込み用 TCP/IP M3S-T4-Tiny 導入ガイド Firmware Integration Technology
(R20AN0051)

4. ハードウェア説明

4.1 RSK+RX62N/RSK+RX63N を使用する場合の端子設定例

表 4-1 に RSK+RX62N/RSK+RX63N での端子設定例を示し、図 4-1 と図 4-2 に PHY-LSI との接続例を示します。

表 4-1 RSK+RX62N/RSK+RX63N での端子設定例

MII モードを使用する場合	RMII モードを使用する場合 ^{*3}	I/O ポート
ET_TX_CLK		PC4
ET_RX_CLK	REF50CK	P76
ET_TX_EN	RMII_TXD_EN	P80
ET_ETXD3		PC6
ET_ETXD2		PC5
ET_ETXD1	RMII_TXD1	P82
ET_ETXD0	RMII_TXD0	P81
ET_TX_ER		PC3
ET_RX_DV		PC2
ET_ERXD3		PC0
ET_ERXD2		PC1
ET_ERXD1	RMII_RXD1	P74
ET_ERXD0	RMII_RXD0	P75
ET_RX_ER	RMII_RX_ER	P77
ET_CRS	RMII_CRS_DV	P83
ET_COL		PC7
ET_MDC		P72
ET_MDIO		P71
ET_LINKSTA		P54 ^{*1}
ET_EXOUT		_*2
ET_WOL		_*2

【注】 *1 ETHER_CFG_USE_LINKSTA を値 0 に設定している場合は設定不要です。

【注】 *2 イーサネットドライバでは使用しない端子なので設定不要です。

【注】 *3 RMII モードは RSK+RX63N でのみ使用できます。

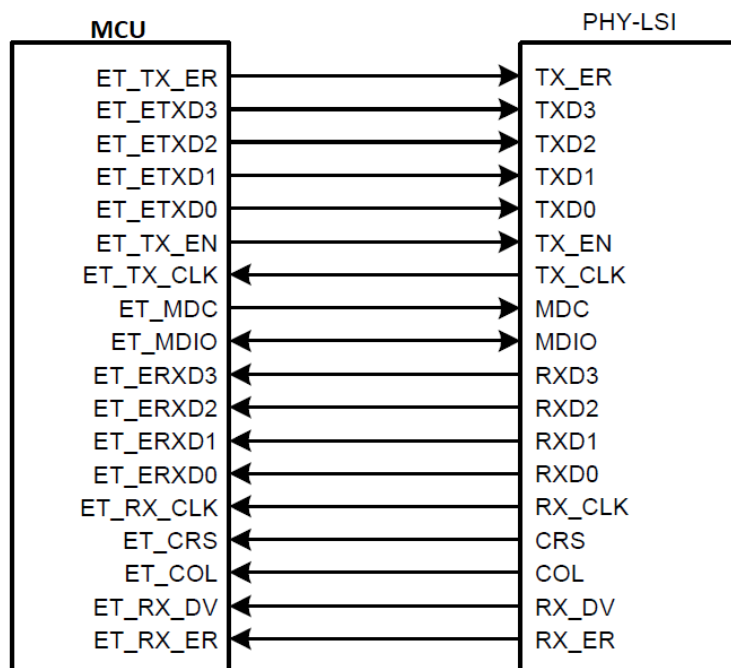


図 4-1 RX62N, RX63N の PHY-LSI との接続例 (MII)

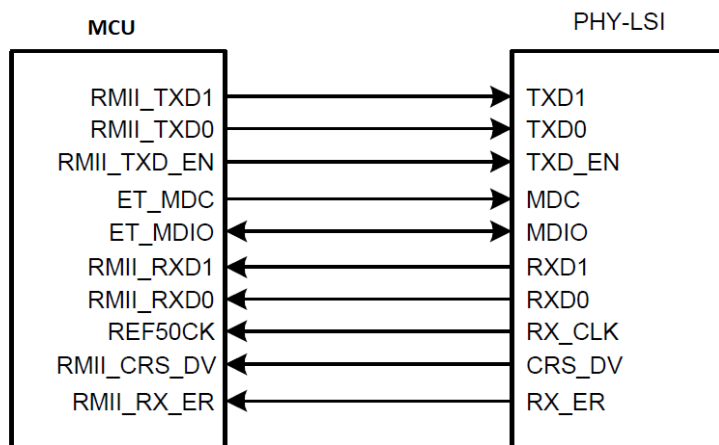


図 4-2 RX62N, RX63N の PHY-LSI との接続例(RMII)

5. ソフトウェア説明

5.1 サンプルプログラム構成

本サンプルプログラムは以下のファイルが含まれます。

表 5-1 サンプルプログラム構成

ファイル/ディレクトリ名		内容
r01an5161xx0100-rx62nrx63n-t4-connectivity		
	workspace	
	r01an5161_rx62n_t4_connectivity_ccrx	RX62N の CCRX サンプルプロジェクト
	r01an5161_rx63n_t4_connectivity_ccrx	RX63N の CCRX サンプルプロジェクト
	r01an5161_rx62n_t4_connectivity_gcc	RX62N の GCC サンプルプロジェクト
	r01an5161_rx63n_t4_connectivity_gcc	RX63N の GCC サンプルプロジェクト
	r01an5161_rx62n_t4_connectivity_iar	RX62N の IAR サンプルプロジェクト
	r01an5161_rx63n_t4_connectivity_iar	RX63N の IAR サンプルプロジェクト
	r01an5161jj0100-rx62nrx63n-t4-connectivity.pdf	導入ガイド(日本語、本書)
	r01an5161ej0100-rx62nrx63n-t4-connectivity.pdf	導入ガイド(英語)
	readme_j.txt	readme 文書(日本語)
	readme_e.txt	readme 文書(英語)

説明例として、workspace 直下の RX63N の CCRX サンプルプロジェクトは表 5-2 のファイルが含まれています。

表 5-2 RX63N の CCRX サンプルプロジェクト構成

ファイル/ディレクトリ名		内容
r01an5161_rx63n_t4_connectivity_ccrx		
	doc	ドキュメントフォルダ
	r01an5161jj0100-rx62nrx63n-t4-connectivity.pdf	導入ガイド(日本語、本書)
	r01an5161ej0100-rx62nrx63n-t4-connectivity.pdf	導入ガイド(英語)
	generate	MCU リセットプログラム
	r01an5161_src	サンプルプログラムソース
	main.c	メイン関数
	echo_srv_sample.h	エコーサーバーサンプルプログラムヘッダファイル
	r_pincfg	端子設定
	mcu_initial	クロック設定、存在しないポートの設定、リセット後に動作している周辺機能の停止
	r_t4_rx	T4 モジュール
	tcp_blocking_sample	TCP ブロッキングコール
	echo_srv.c	TCP ブロッキングサーバーエコー
	tcp_nonblocking_cancel_sample	TCP ノンブロッキングキャンセルコール
	echo_srv.c	TCP ノンブロッキングキャンセルサーバーエコー
	tcp_nonblocking_sample	TCP ノンブロッキングコール
	echo_srv.c	TCP ノンブロッキングサーバーエコー
	udp_blocking_sample	UDP ブロッキングコール
	echo_srv.c	UDP ブロッキングサーバーエコー
	udp_nonblocking_sample	UDP ノンブロッキングコール
	echo_srv.c	UDP ノンブロッキングサーバーエコー
	driver	ドライバフォルダ
	cmt_driver	CMT ドライバ
	common_driver	共通ドライバ
	ether_driver	イーサネットドライバ
	systime_driver	システムタイマー
	t4_driver	T4 ドライバ
	r_driver_rx_config.h	ドライバコンフィグレーション
	HardwareDebug	デバッグフォルダ
	Release	リリースフォルダ
	.settings	プロジェクトセッティングズフォルダ
	.cproject	C プロジェクトファイル
	.project	プロジェクトファイル
	r01an5161_rx63n_t4_connectivity.x.launch	Launch ファイル
	r01an5161_rx63n_t4_connectivity.rcpc	RCPC ファイル

表 5-2 の r_t4_rx フォルダには、表 5-3 のファイルが含まれます。

表 5-3 r_t4_rx フォルダ構成

ファイル/ディレクトリ名		内容
T4 FIT Module 本体 (r_t4_rx)		
T4 ライブラリ(lib)		
CC-RX		
	T4_Library_ether_ccrx_rxv1_big.lib	T4 ライブラリ(RXV1 コア/ビッグエンディアン/Ethernet 用)
	T4_Library_ether_ccrx_rxv1_little.lib	T4 ライブラリ(RXV1 コア/リトルエンディアン/Ethernet 用)
	T4_Library_ether_ccrx_rxv1_big_debug.lib	T4 デバッグ情報付きライブラリ(RXV1 コア/ビッグエンディアン/Ethernet 用/QE for TCP/IP 使用時)
	T4_Library_ether_ccrx_rxv1_little_debug.lib	T4 デバッグ情報付きライブラリ(RXV1 コア/リトルエンディアン/Ethernet 用/QE for TCP/IP 使用時)
GCC		
	libT4_Library_ether_gcc_rxv1_big.a	T4 ライブラリ(RXV1 コア/ビッグエンディアン/Ethernet 用)
	libT4_Library_ether_gcc_rxv1_little.a	T4 ライブラリ(RXV1 コア/リトルエンディアン/Ethernet 用)
IAR		
	T4_Library_ether_iar_rxv1_big.a	T4 ライブラリ(RXV1 コア/ビッグエンディアン/Ethernet 用)
	T4_Library_ether_iar_rxv1_little.a	T4 ライブラリ(RXV1 コア/リトルエンディアン/Ethernet 用)
	r_t4_itcpip.h	T4 ヘッダファイル
	r_stdint.h	型定義ヘッダファイル
	r_mw_version.h	バージョン情報ヘッダファイル
T4 ドキュメント(doc)		
ja		
	r20uw0031jj0111-t4tiny.pdf	ユーザーズマニュアル(日本語)
	r20uw0032jj0108-t4tiny.pdf	ドライバインタフェース仕様書(日本語)
	r20an0051jj0209-rx-t4.pdf	T4 導入ガイド(日本語)
En		
	r20uw0031ej0111-t4tiny.pdf	ユーザーズマニュアル(英語)
	r20uw0032ej0108-t4tiny.pdf	ドライバインタフェース仕様書(英語)
	r20an0051ej0209-rx-t4.pdf	T4 導入ガイド(英語)
T4 ライブラリ生成環境(make_lib)		
	make_lib.zip	T4 ライブラリ生成環境 (ソースコード入り)
T4 コンフィグリファレンス(ref)		
	config_tcpudp_reference.tpl	T4 コンフィグファイル(テンプレート)
	r_t4_rx_config_reference.h	T4 コンフィグヘッダ(リファレンス)
src		
	config_tcpudp.c	T4 コンフィグファイル
	readme.txt	readme

表 5-2 の driver フォルダに、表 5-4 のファイルが含まれます。

表 5-4 driver フォルダ構成

ファイル/ディレクトリ名		内容
driver		
cmt_driver		
	src	
	cmt_driver.c	CMT ドライバソースファイル
	cmt_if.h	CMT ドライバインターフェース
common_driver		
	src	
	drv_locking.c	ロッキングソースファイル
	r_rx_compiler.h	各コンパイラの機能定義を統一するファイル
	r_rx_inrtinsic_funtions.c	GCC/IAR コンパイラで実装されていない組み込み関数を定義したファイル
	r_rx_inrtinsic_funtions.h	各コンパイラの組み込み関数の定義を統一するファイル
	common_driver.h	共通ドライバインターフェース
ether_driver		
	src	
	phy	
	phy.c	イーサネット PHY デバイスドライバ
	phy.h	イーサネット PHY デバイスドライバインターフェース
	targets	
	rx63n	
	ether_setting_rx63n.c	割り込みと PHY マネージメントインターフェースの設定ファイル
	ether_driver.c	イーサネットドライバソースファイル
	ether_private.h	イーサネット内部に使用するマクロ、構造体と関数を定義するヘッダファイル
	ether_if.h	イーサネットドライバインターフェース
systime_driver		
	src	
	sys_time_driver.c	システムタイマーソースファイル
	sys_time_private.h	システムタイマー内部に使用するマクロ、構造体と関数を定義するヘッダファイル
	sys_time_if.h	システムタイマーインターフェース
t4_driver		
	src	
	ether_callback.c	イーサネットのコールバック関数
	t4_driver.c	T4 ドライバソースファイル
	timer.c	タイマソースファイル
	timer.h	タイマインターフェース
	r_driver_rx_config.h	ドライバコンフィグレーション

5.2 サンプルソース

サンプルプロジェクト毎に、以下のソースファイルを用意しています。

- TCP ブロッキングコール (tcp_blocking ディレクトリ)
- TCP ノンブロッキングキャンセルコール(tcp_nonblocking_cancel ディレクトリ)
- TCP ノンブロッキングコール (tcp_nonblocking ディレクトリ)
- UDP ブロッキングコール (udp_blocking ディレクトリ)
- UDP ノンブロッキングコール (udp_nonblocking ディレクトリ)

サンプルプログラムは共通の main 関数を持ちます。main 関数は echo_srv()関数を呼び出します。上記 5 パターンはそれぞれエコーバックサーバの実装例です。いずれか 1 パターンのプロジェクトを選択してください。

5.2.1 TCP エコーバックサーバ関数（ブロッキングコールの場合）のフロー

エコーバックサーバ関数

図 5-2 TCP のエコーバックサーバ（ブロッキングコール）の処理フロー参照

5.2.2 TCP エコーバックサーバ関数（ノンブロッキングコールの場合）のフロー

エコーバックサーバ関数

図 5-3 TCP のエコーバックサーバ（ノンブロッキングコール）の処理フロー参照

コールバック関数のフロー

図 5-4 TCP のコールバック（ノンブロッキングコール）の処理フロー参照

5.2.3 UDP エコーバックサーバ関数（ブロッキングコールの場合）のフロー

エコーバックサーバ関数

図 5-5 UDP のエコーバックサーバ（ブロッキングコール）の処理フロー参照

5.2.4 UDP エコーバックサーバ関数（ノンブロッキングコールの場合）のフロー

エコーバックサーバ関数

図 5-6 UDP のエコーバックサーバ（ノンブロッキングコール）の処理フロー参照

コールバック関数のフロー

図 5-7 UDP のコールバック（ノンブロッキングコール）の処理フロー参照

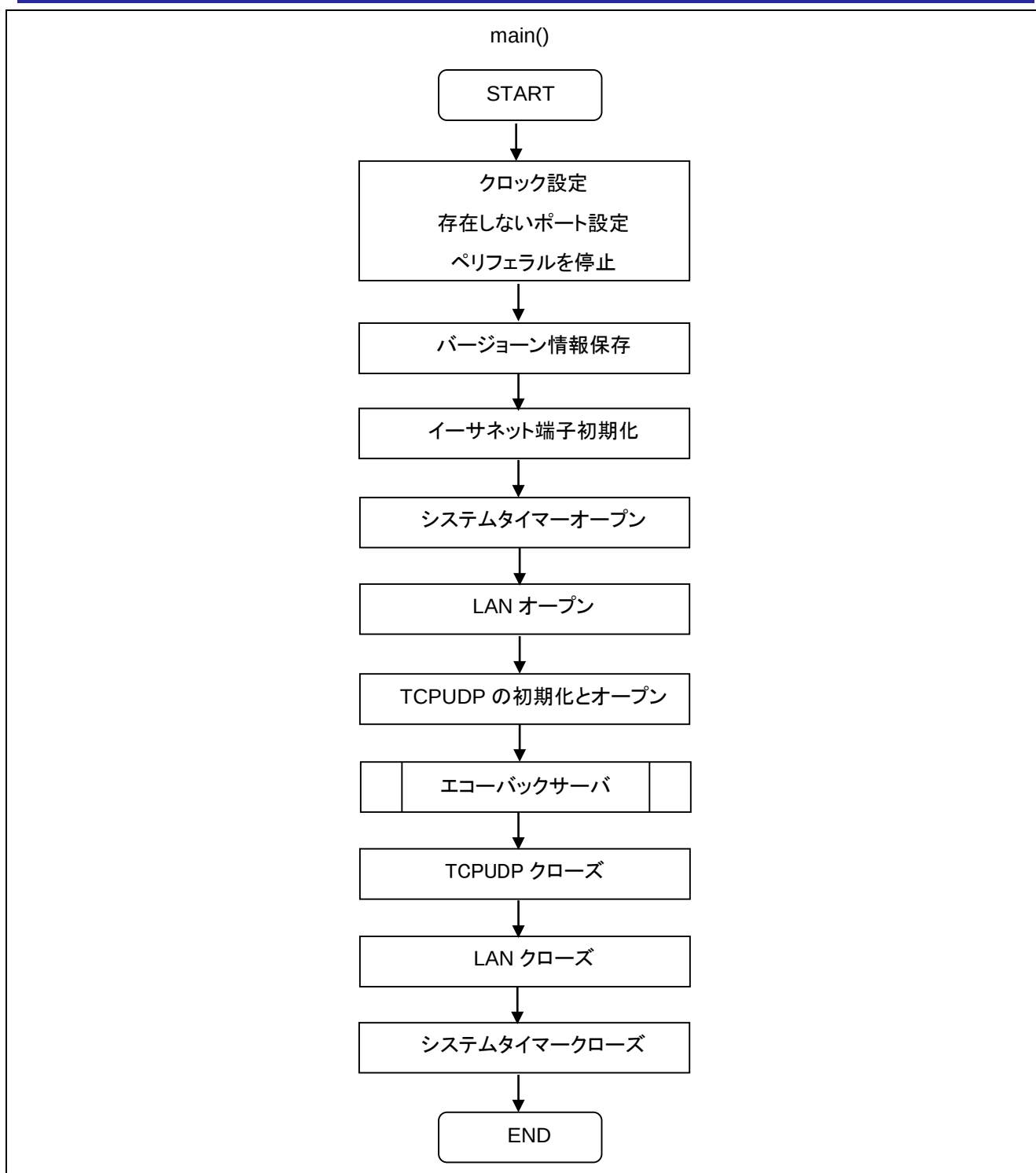


図 5-1 main 関数の処理フロー

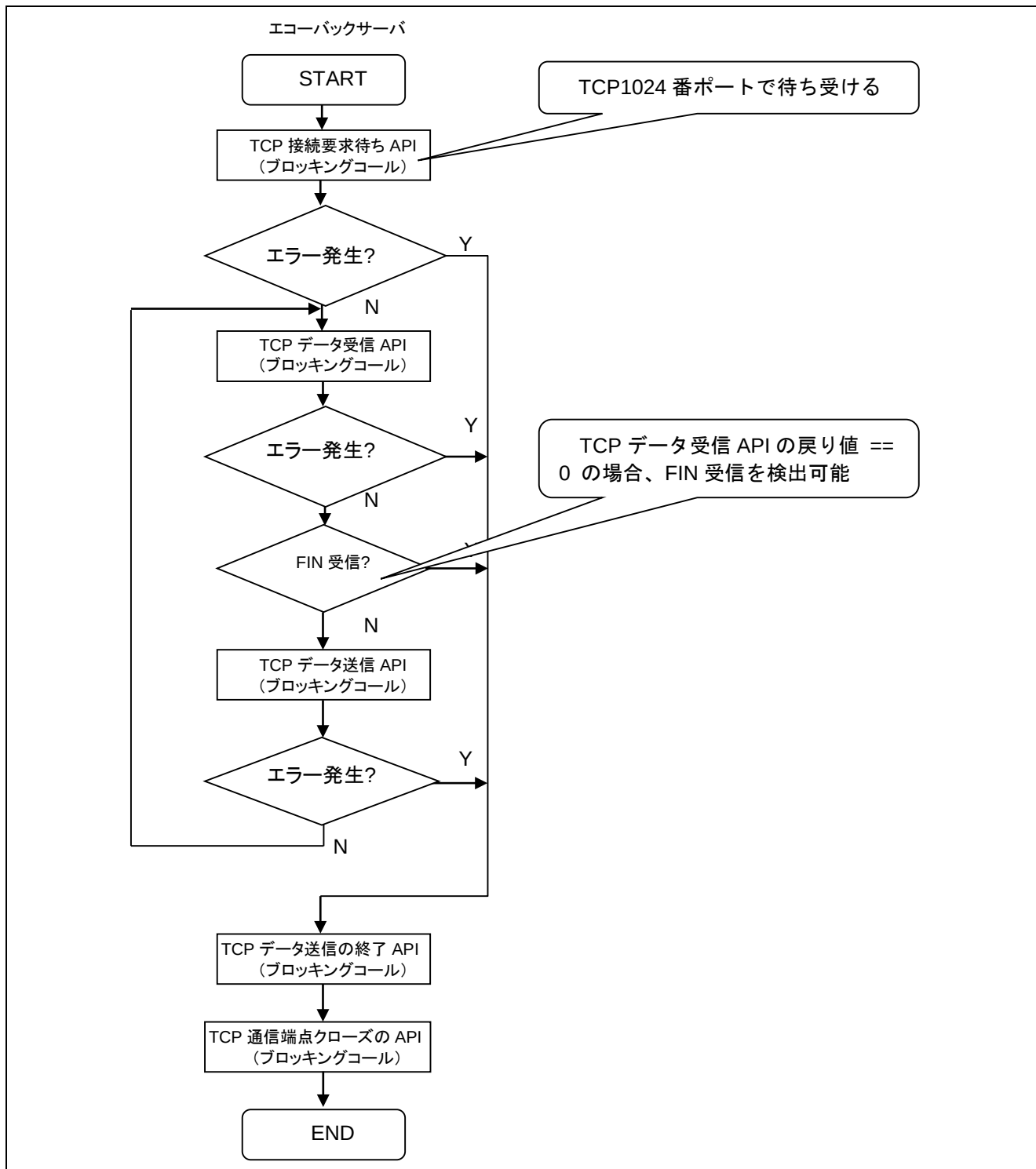


図 5-2 TCP のエコバックサーバ (ブロッキングコール) の処理フロー

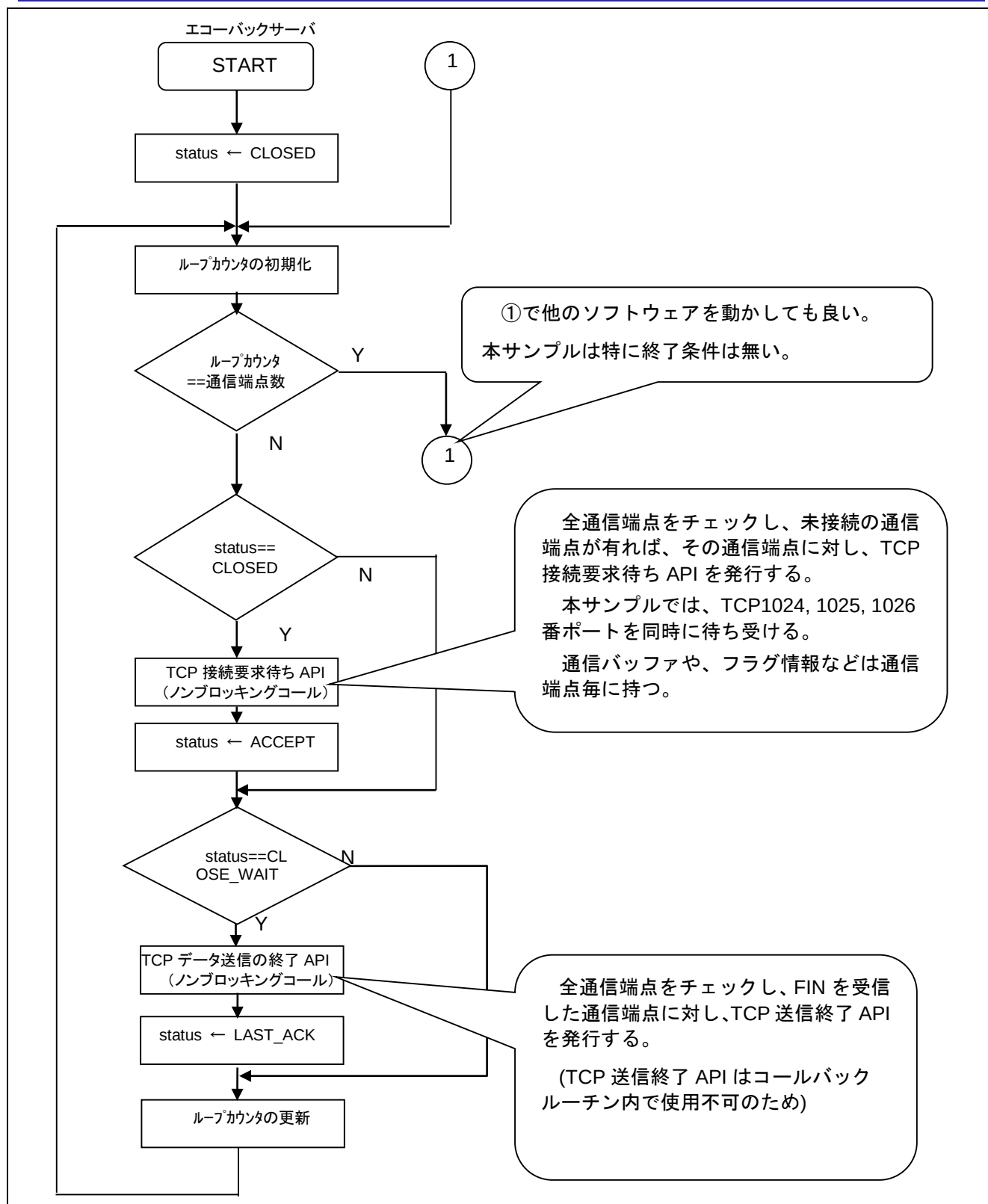


図 5-3 TCP のエコーバックサーバ（ノンブロッキングコール）の処理フロー

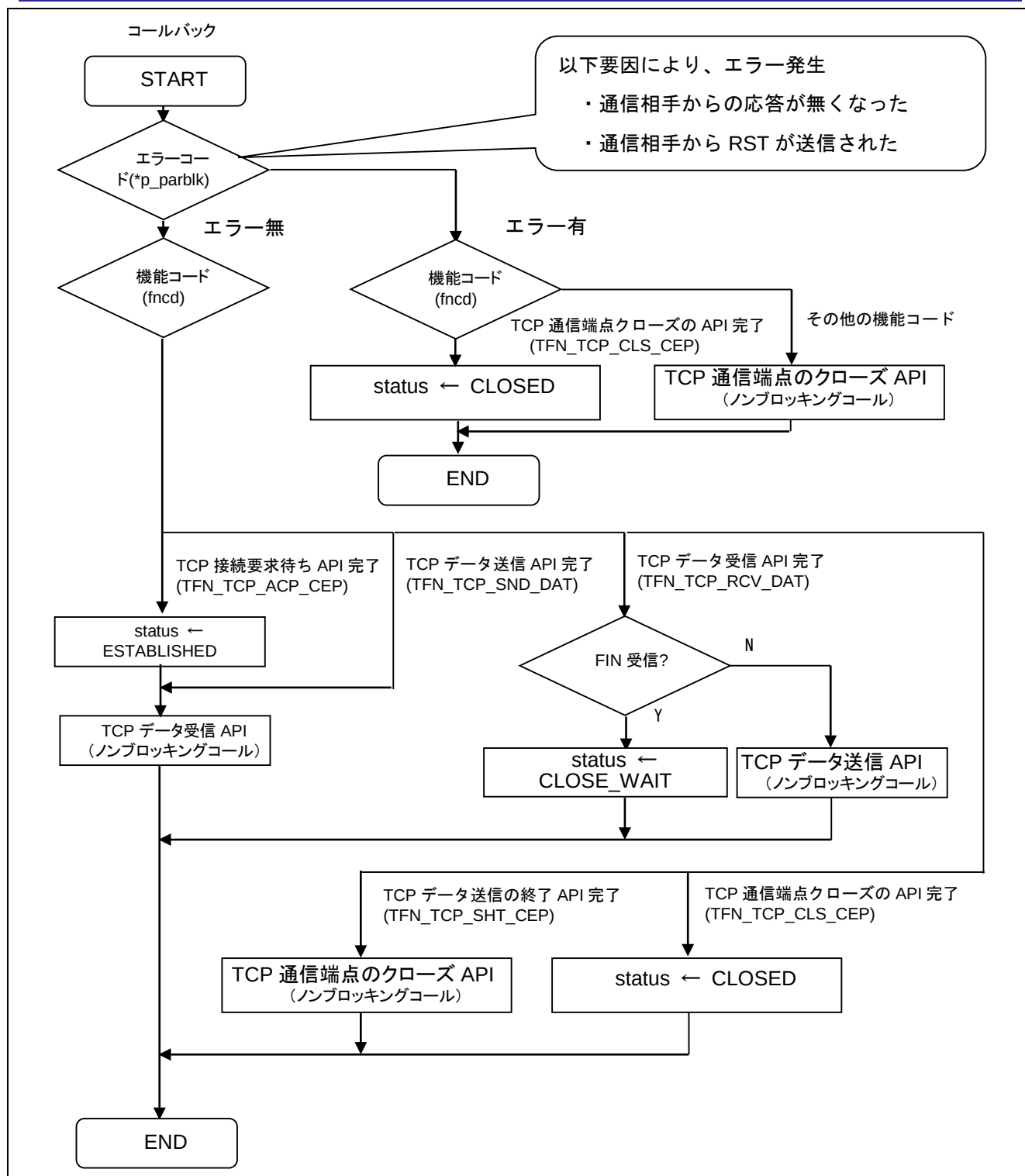


図 5-4 TCP のコールバック（ノンブロッキングコール）の処理フロー

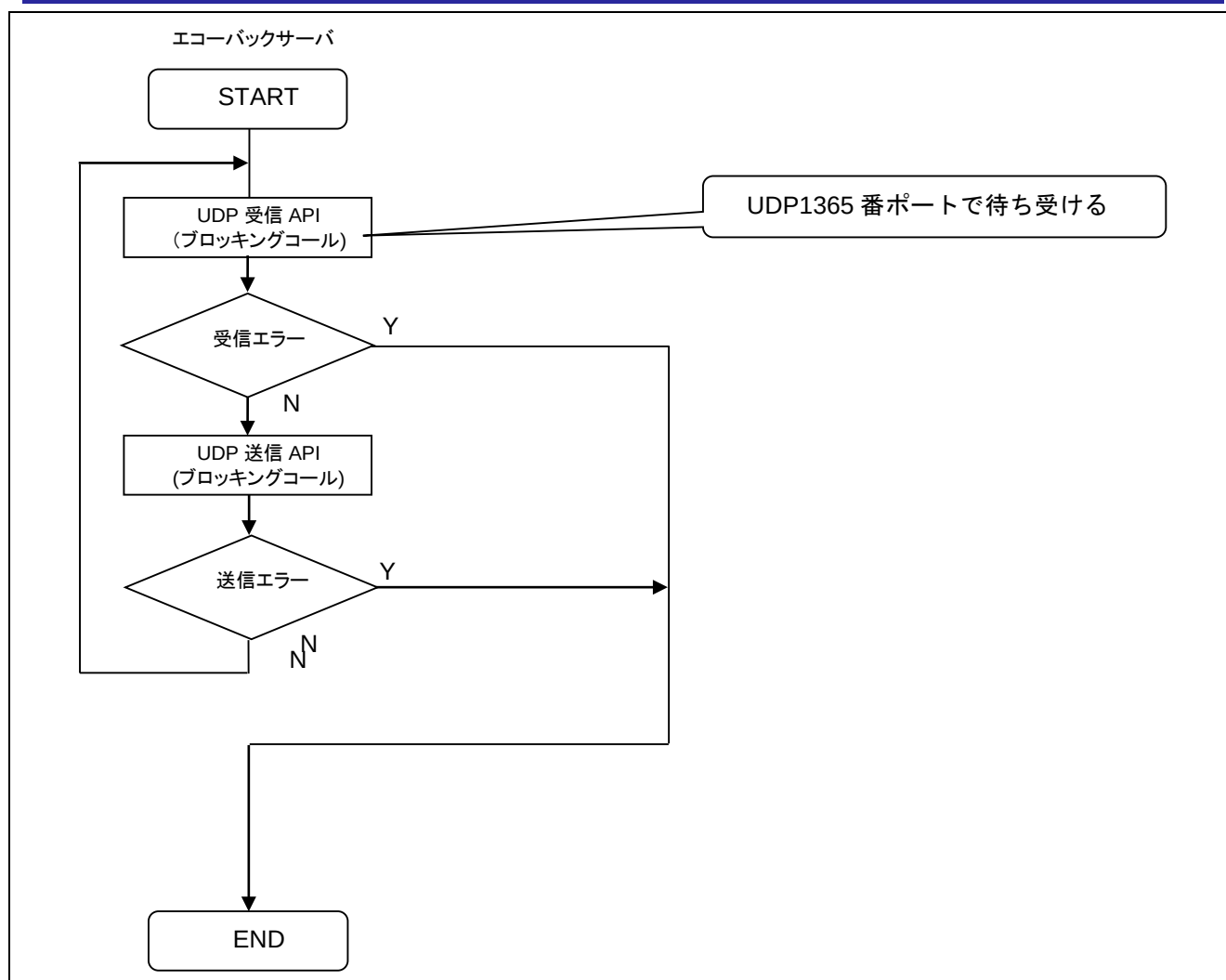


図 5-5 UDP のエコーバックサーバ（ブロッキングコール）の処理フロー

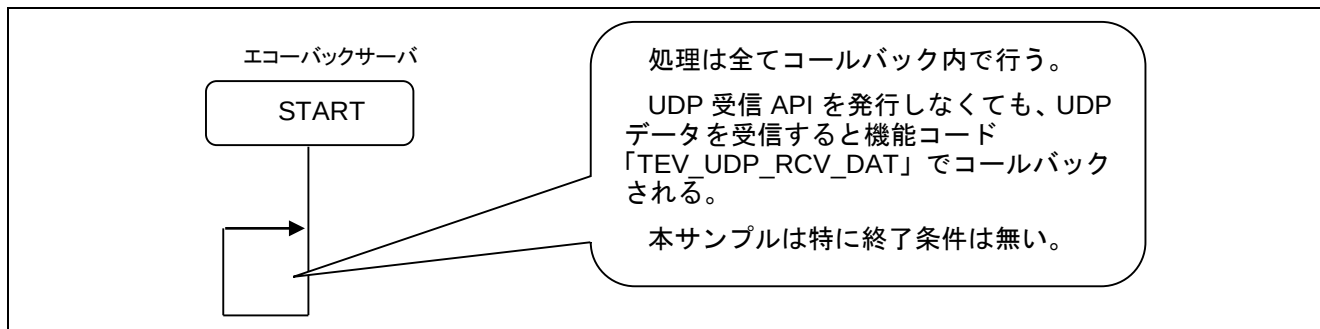


図 5-6 UDP のエコーバックサーバ（ノンブロッキングコール）の処理フロー

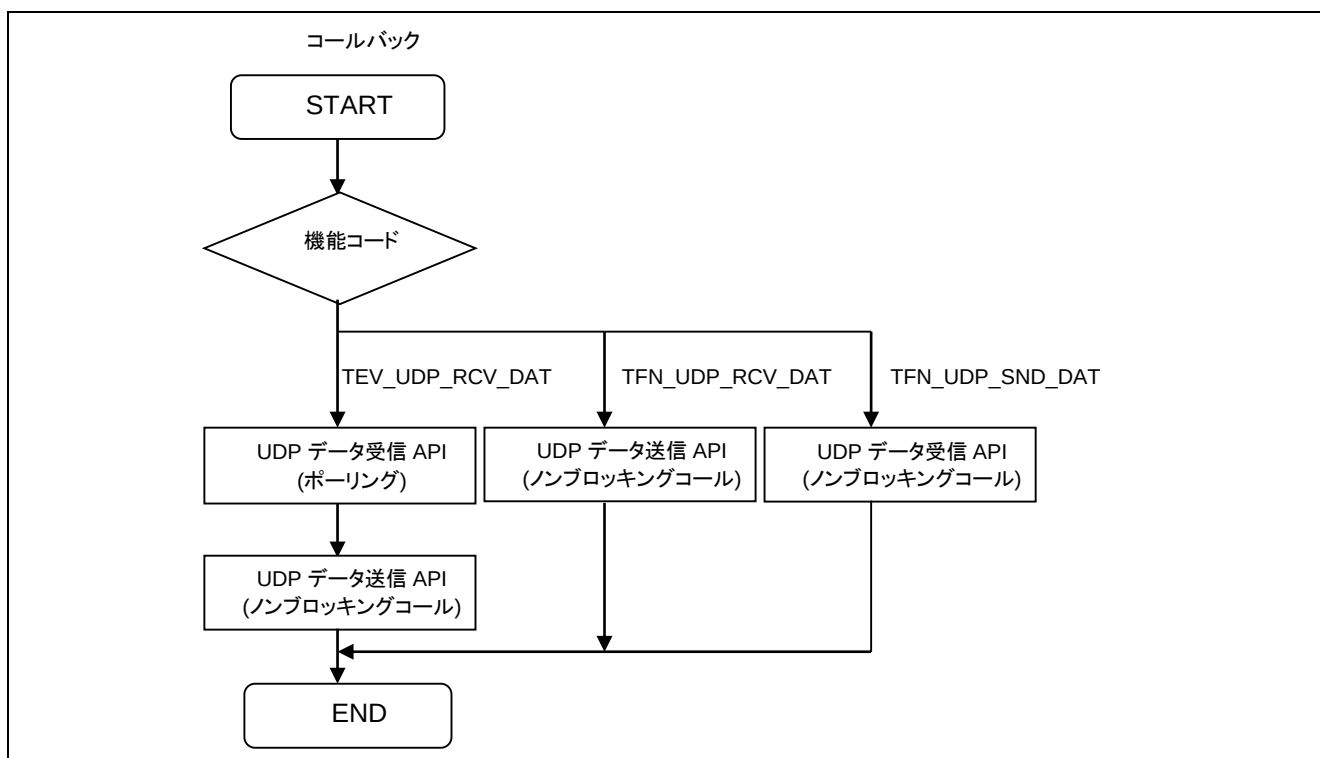


図 5-7 UDP のコールバック（ノンブロッキングコール）の処理フロー

5.3 RX62N/RX63N 用サンプルプログラム環境

RX62N/RX63N のサンプルプログラムのプロジェクトはパッケージで提供され、プロジェクトは e² studio 若しくは IAR Embedded Workbench からインポートできます。

5.3.1 ソフトウェア構成

図 5-8 に、サンプルプログラムのソフトウェアおよび各ドライバの構成図を示します。

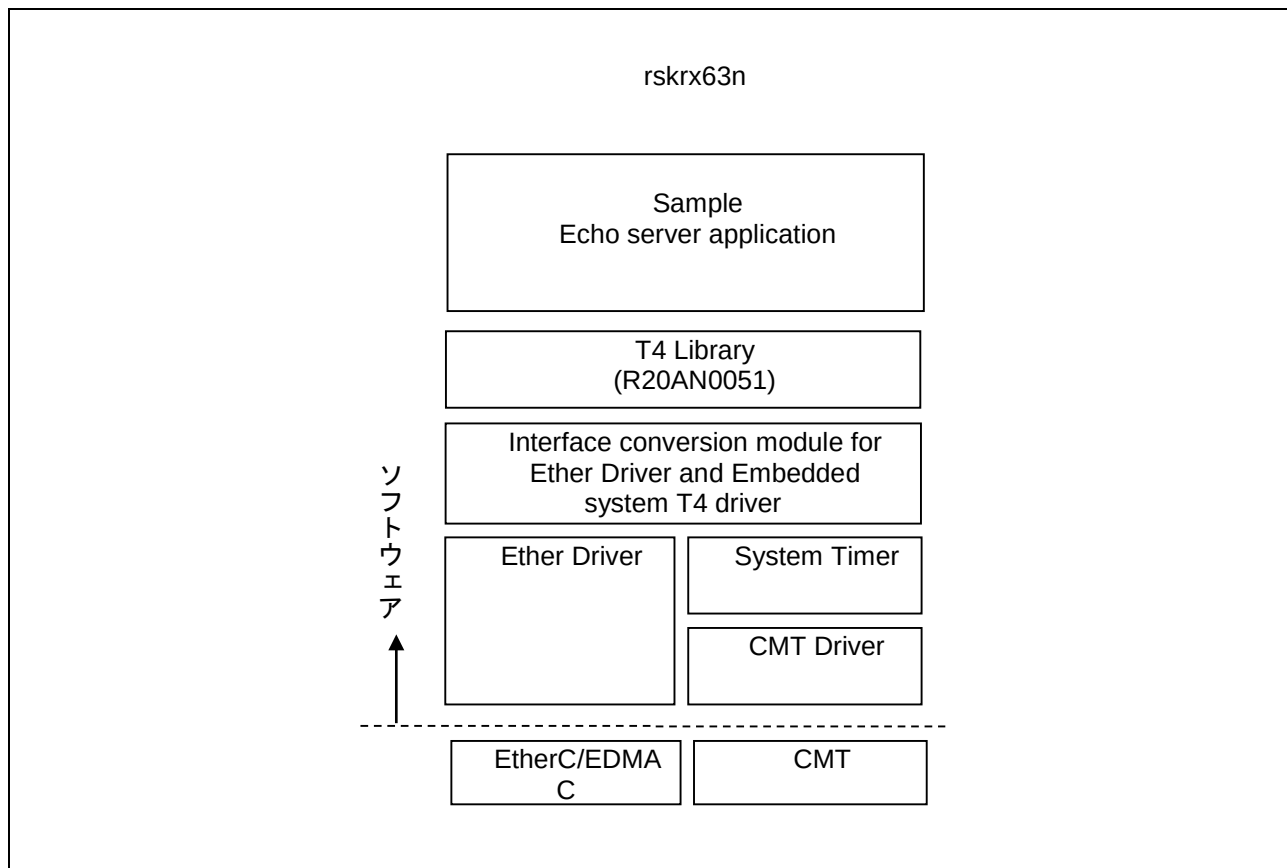


図 5-8 各ドライバの構成図

5.3.2 CS+プロジェクトへの変換方法

e² studio のプロジェクトは、プロジェクト内に含まれている rcpc ファイルを用いて、CS+プロジェクトへコンバートすることができます。変換方法を以下に示します。(RX63N 用のサンプルプログラムでの例)

- CS+ for CC を起動し、「e² studio / CubeSuite / ...」の「GO」ボタンを押します。
- 「e² studio プロジェクト・ファイル(*.rcpc)」を選択して、*.rcpc ファイルを開きます。
- 「プロジェクト変換設定」ウインドウが開き、ツリー上でプロジェクトを選択します。
- ツリーの右側のプロジェクト設定で、使用するマイクロコントローラを「RX63N」->「R5F563NFDxFC」を選択して「OK」を押します。CS+は変換されたプロジェクトを出力します。
- 「プロジェクトツリー」から「CC-RX」を選択します。
- 「共通オプション」タブ->の「CPU」->「命令セット・アーキテクチャ」を「RXv2 アーキテクチャ」に設定します。
- 「プロジェクトツリー」の「ファイル」->「src」の中の各フォルダには、echo_srv.c が登録されています。動作確認する echo_srv.c を除き、他の echo_srv.c は「右クリック」→「プロパティ」を選択後、「ビルドの対象とする」を「いいえ」に設定します。
- プロジェクトをビルドして、ビルド完了します。
- ユーザは環境に合わせてデバッグツールの設定を行ってください。その後、ユーザはサンプルプログラムの動作を確認することができます。

5.4 サンプルプログラム動作確認

Ethernet サンプルプログラムの動作確認方法

5.4.1 環境構築

(1) ハードウェアの接続

以下のようにハードウェアを接続します。

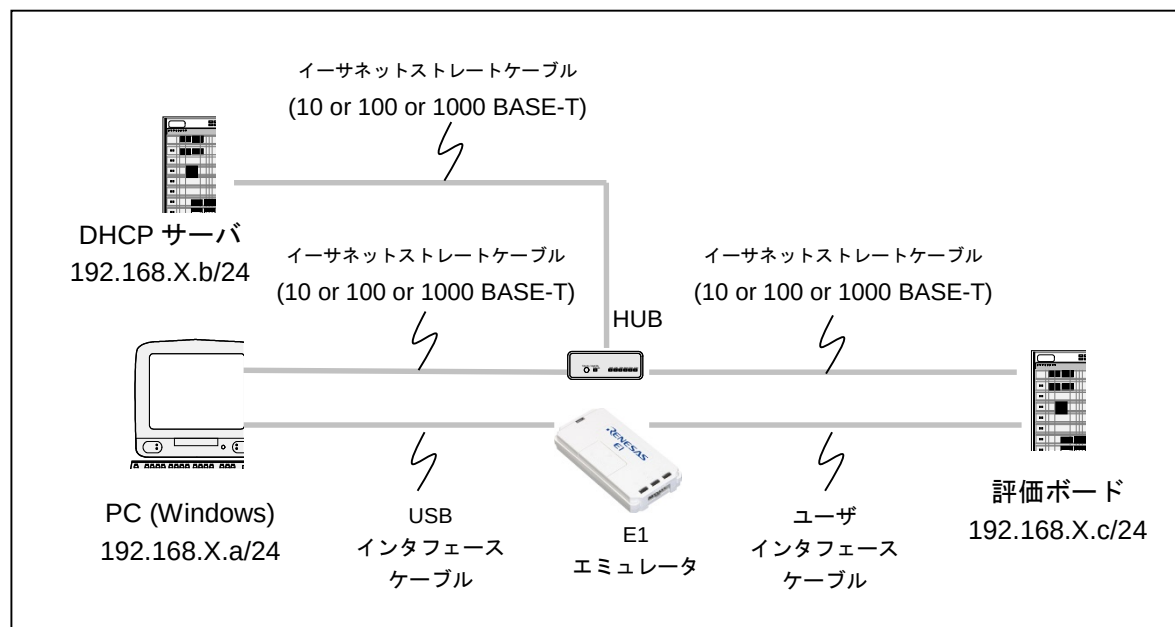


図 5-9 Ethernet サンプルプログラムの実行環境

図 5-9 の HUB について、弊社では以下の製品を使用して動作確認をしています。

- ・ NETGEAR 製 型名: GS108E

この HUB は「ポートミラーリング機能」が有り、Ethernet 上に流れるデータのモニタリング機能を提供します。ポートミラーリング機能は通常の HUB では実現できないパケットモニタリングの環境を実現出来ます。たとえば以下のような環境でボード A からボード B に送信した場合、通常の HUB だとボード B が繋がっているポートにしかデータを出しませんが、ポートミラーリング機能があると、HUB に入力されたデータを無条件で特定ポートにミラーして出力することができます。これにより、1 対 1 通信を別 PC でパケットモニタすることが可能です。パケットモニタのソフトは Wireshark を推奨します。Wireshark を promiscuous モードにすることで、ボード間の 1 対 1 通信をモニタすることができます。

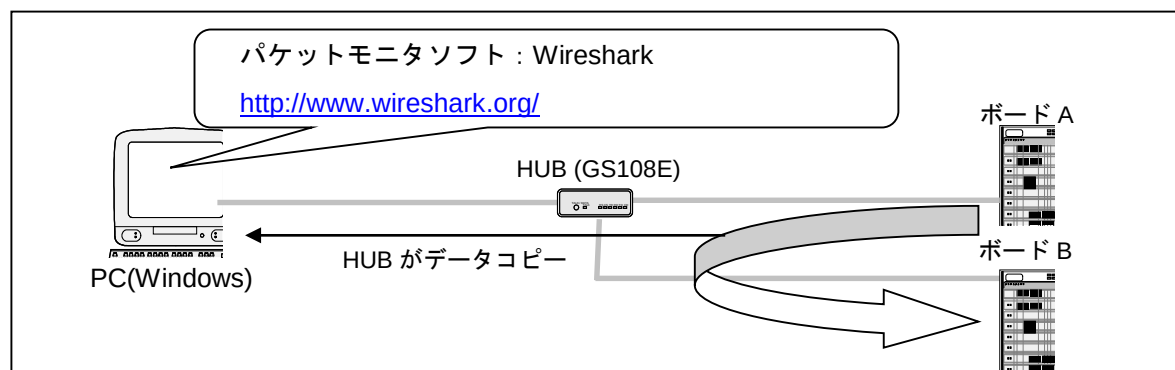


図 5-10 Ethernet サンプルプログラムの動作確認環境

(2) PC の設定変更

Windows 7 の場合：

「コントロールパネル」→「ネットワークと共有センター」→「アダプターの設定の変更」をクリックします。

「ローカルネットワークの接続」を右クリックして、プロパティをクリックして「ローカルエリア接続プロパティ画面」を開きます。

「ネットワーク」タブを選択し、「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」ボタンをクリックします。

以下のダイアログボックスが開くので、「IP アドレスを自動的に取得する」を選択してください。

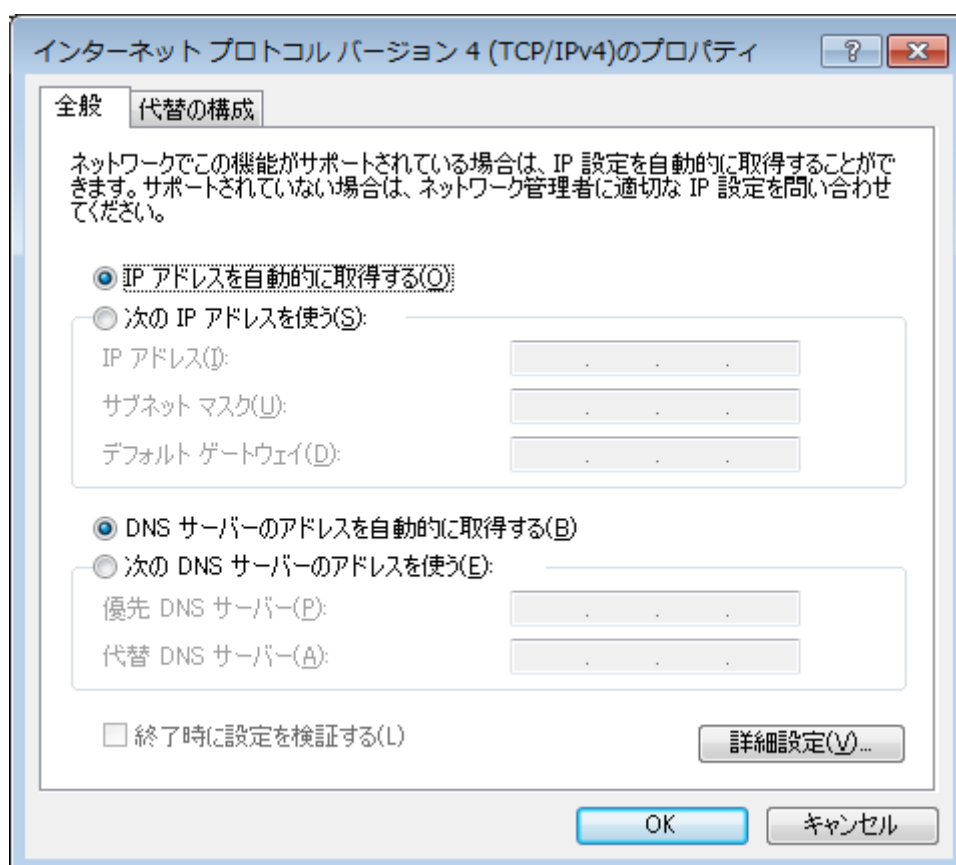


図 5-11 インターネットプロトコルバージョン 4 (TCP/IPv4)のプロパティ

設定後、OK ボタンをクリックしてダイアログボックスを閉じます。

(3) サンプルプログラム(sample フォルダ)の起動

- ・ e²studio を起動し、sample プログラムを開きます。
- ・ [プロジェクト] → [プロジェクトのビルド] の順でクリックします。
- ・ E1 エミュレータを接続し、[実行] → [デバッグ] の順でクリックします。
- ・ [デバッグ]ビューにある  ボタンをクリックするか、[F8] キーを入力して、プログラムを実行します。

(4) マイコンの IP アドレスの確認

サンプルプログラムを実行すると、DHCP サーバから IP アドレスが割り当てられます。

e² studio の Renesas デバッグ仮想コンソール上で、割り当てられた IP アドレスを確認することができます。GCC サンプルプロジェクトでは Renesas デバッグ仮想コンソール上に IP アドレスは表示されません。デバッグウィンドウで直接確認してください。

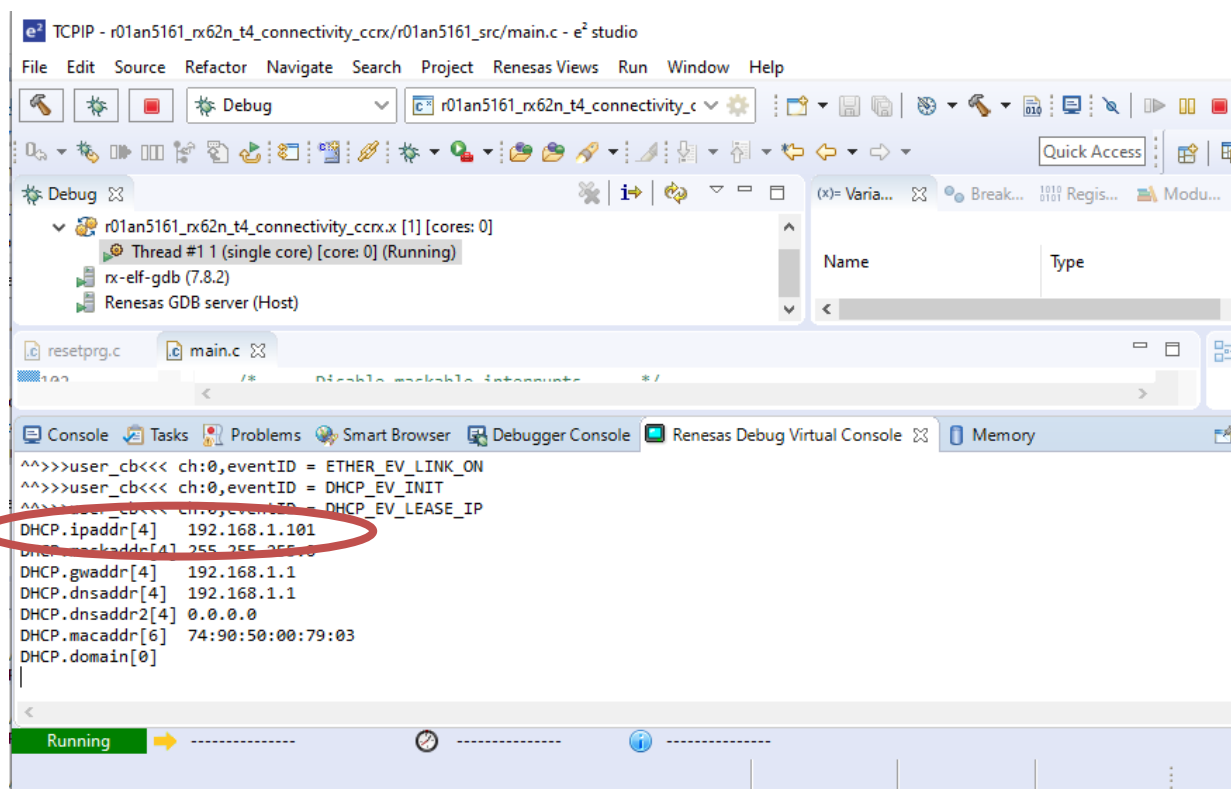


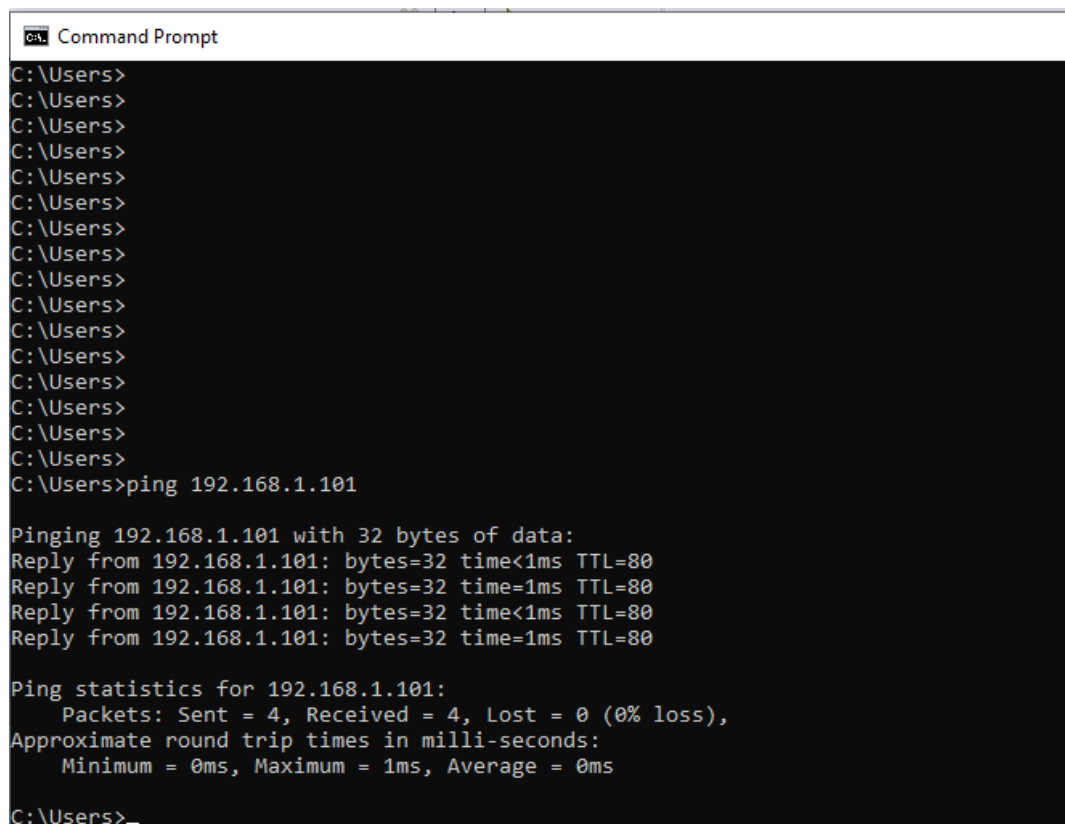
図 5-12 IP アドレスの表示例

図の例は、マイコンに割り当てられた IP アドレスが 192.168.1.101 であることを示します。

PC (Windows) 側の IP アドレスを確認したい場合は、コマンドプロンプトから ipconfig を実行してください。

(5) 通信検査

コマンドプロンプトからマイコンの IP アドレスに対し ping を実行



```

C:\Users>
C:\Users>
C:\Users>
C:\Users>
C:\Users>
C:\Users>
C:\Users>
C:\Users>
C:\Users>
C:\Users>
C:\Users>
C:\Users>
C:\Users>
C:\Users>
C:\Users>
C:\Users>
C:\Users>ping 192.168.1.101

Pinging 192.168.1.101 with 32 bytes of data:
Reply from 192.168.1.101: bytes=32 time<1ms TTL=80
Reply from 192.168.1.101: bytes=32 time=1ms TTL=80
Reply from 192.168.1.101: bytes=32 time<1ms TTL=80
Reply from 192.168.1.101: bytes=32 time=1ms TTL=80

Ping statistics for 192.168.1.101:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users>
```

図 5-13 ping の実行例

5.4.2 TCP 接続を確認する

PC 上の MS-DOS プロンプトで telnet コマンドを実行し、コネクションを確立します。

(1) telnet の有効化(Windows7 のみ)

- ・ スタート→コントロールパネル→プログラムと機能

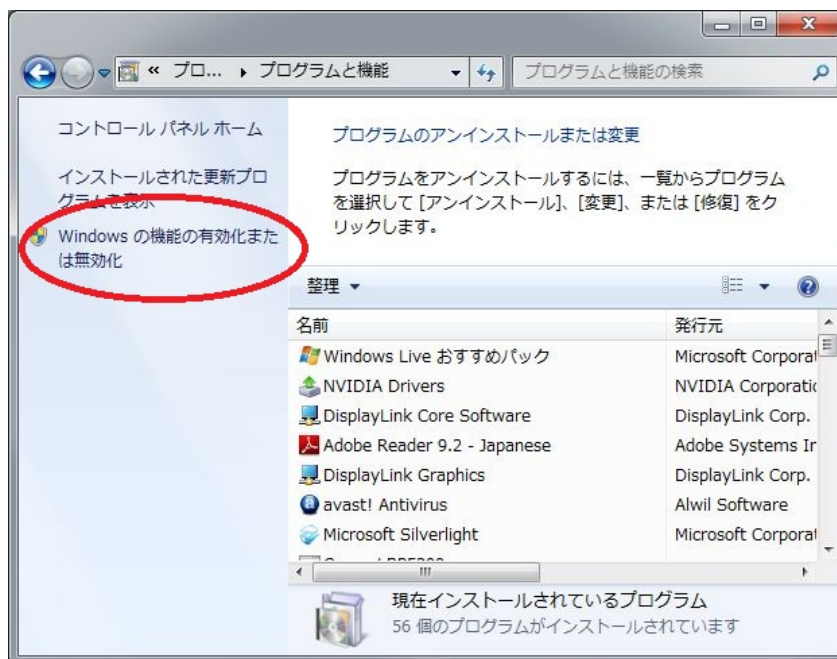


図 5-14 プログラムと機能

- ・ Telnet クライアントにチェックを入れてください。

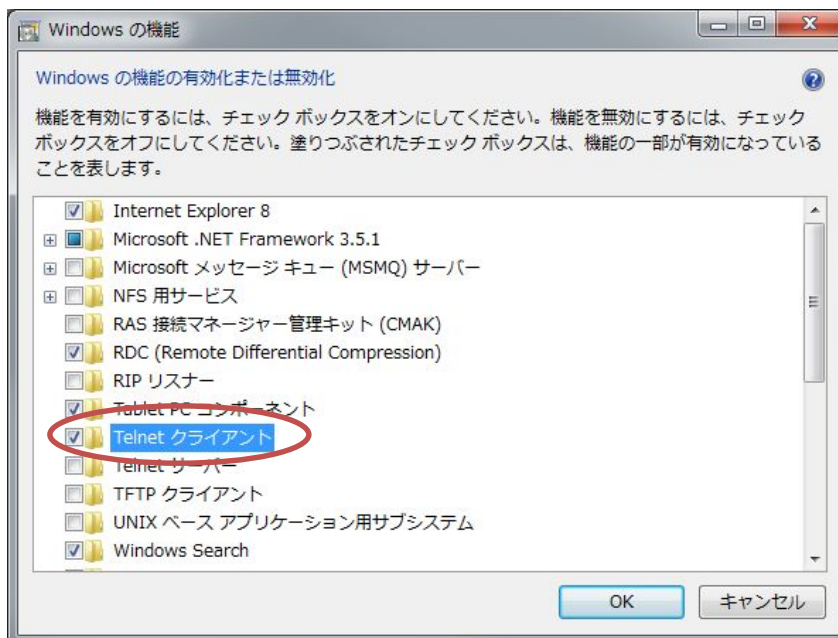


図 5-15 Telnet クライアント

(2) 単一 LAN ボードの場合

サンプルプログラムを実行する環境に合わせて、以下のいずれかを動作させてください。
PC 上の MS-DOS プロンプトで下記コマンドを実行し、コネクションを確立します。

【TCP ブロッキングコールの場合】

```
telnet 192.168.X.c 1024
```

【TCP ノンブロッキングコールの場合（複数同時接続可能）】

```
telnet 192.168.X.c 1024
```

```
telnet 192.168.X.c 1025
```

(3) 複数 LAN ボードの場合

サンプルプログラムを実行する環境に合わせて、以下のいずれかを動作させてください。
PC 上の MS-DOS プロンプトで下記コマンドを実行し、コネクションを確立します。

【TCP ブロッキングコールの場合】

```
telnet 192.168.X.c 1024
```

```
telnet 192.168.X.d 1025
```

【TCP ノンブロッキングコールの場合(同時に複数の通信端点が使用可能です)】

```
telnet 192.168.X.c 1024
```

```
telnet 192.168.X.c 1025
```

```
telnet 192.168.X.d 1026
```

```
telnet 192.168.X.d 1027
```

(4) 通信の終了

画面が暗転した状態で、キーボードから入力を行ってください。

入力したデータが画面上に表示されれば動作確認 OK です。

Ctrl + "]" と入力し、続いて、"quit[enter キー入力]" と入力すると接続を切断できます。

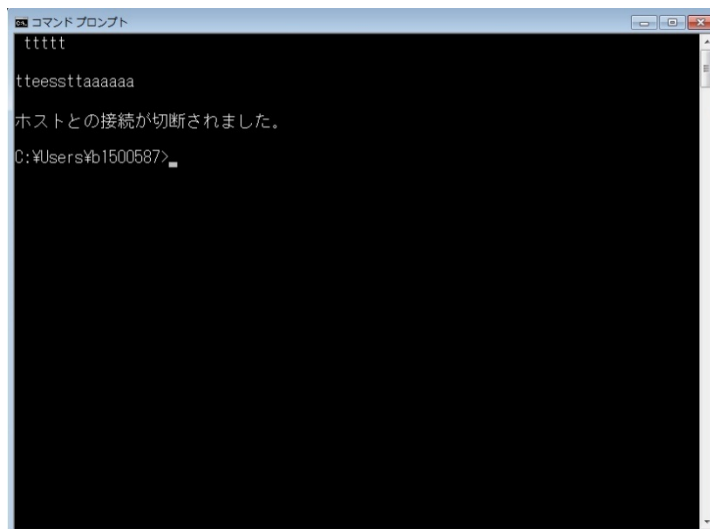


図 5-16 接続の切断

5.4.3 UDP 接続を確認する

PC 上から UDP 送受信フリーソフトを使用し、マイコンにコマンドを送信します。

(1) UDP ソフトの準備

- ・次のサイトで UDP データが送受信可能なフリーソフト : Socket Debugger Free を入手します。

<https://www.udom.co.jp/sdg>

(2) 単一 LAN ボードの場合

PC 上から UDP 送受信フリーソフトを使用します。設定は以下の通り。

【UDP ブロッキングコールの場合】

相手先 IP アドレス 192.168.X.c 、使用ポート番号 1365

【UDP ノンブロッキングコールの場合 (複数同時通信可能)】

相手先 IP アドレス 192.168.X.c 、使用ポート番号 1365

相手先 IP アドレス 192.168.X.c 、使用ポート番号 1366

(3) 複数 LAN ボードの場合

PC 上から UDP 送受信フリーソフトを使用します。設定は以下の通り。

【UDP ブロッキングコールの場合】

相手先 IP アドレス 192.168.X.c 、使用ポート番号 1365

相手先 IP アドレス 192.168.X.d 、使用ポート番号 1366

【UDP ノンブロッキングコールの場合 (複数同時通信可能)】

相手先 IP アドレス 192.168.X.c 、使用ポート番号 1367

相手先 IP アドレス 192.168.X.c 、使用ポート番号 1368

相手先 IP アドレス 192.168.X.d 、使用ポート番号 1369

相手先 IP アドレス 192.168.X.d 、使用ポート番号 1370

(4) 通信の終了

UDP はコネクションを確立しないため、通信の終了にコマンドは不要です。

5.5 コンパイル時の設定

本モジュールのコンフィギュレーションオプションの設定は、`r_t4_rx_config.h` と `r_driver_rx_config.h` で行います。

`r_t4_rx_config.h` のオプション名および設定値に関する説明は、R20AN0051JJ0209 の「2.7 コンパイル時の設定」を参照してください。`r_driver_rx_config.h` のオプション名および設定値に関する説明を、下表に示します。合わせてユーザーズマニュアルを参照してください。

Configuration options in <code>r_driver_rx_config.h</code>	
DRV_MCU_RX63_ALL ※デフォルト値は 1	RX63N グループを使用するか否かを設定します。 RX63N グループを使用しない場合、このオプションをコメント化しておいてください。
DRV_MCU_RX62_ALL ※デフォルト値は 1	RX62N グループを使用するか否かを設定します。 RX62N グループを使用しない場合、このオプションをコメント化しておいてください。
DRV_PCLKB_HZ ※デフォルト値は 48000000	RX63N の PCLKB 周波数を設定します。 設定範囲は RX63N グループの UMH を参照してください。
DRV_PCLK_HZ ※デフォルト値は 48000000	RX62N の PCLKB 周波数を設定します。 設定範囲は RX62N グループの UMH を参照してください。
CMT_RX_CFG_IPR ※デフォルト値は 5	CMT 割り込みで使用される割り込み優先レベルを設定します。
ETHER_CFG_MODE_SEL ※デフォルト値は 0	ETHERC とイーサネット PHY-LSI 間のインタフェースを設定してください。 “0” の場合、MII(Media Independent Interface)を選択します。 “1” の場合、RMII (Reduced Media Independent Interface) を選択します。
ETHER_CFG_CH0_PHY_ADDRESS ※デフォルト値は 31	ETHERC チャンネル 0 が使用する PHY-LSI に割り当てられた PHY アドレスを設定してください。 “0” ~ “31” の範囲で設定してください。
ETHER_CFG_EMAC_RX_DESCRIPTOR ※デフォルト値は 1	受信ディスクリプタの数を設定してください。 “1” 以上の値を設定してください。
ETHER_CFG_EMAC_TX_DESCRIPTOR ※デフォルト値は 1	送信ディスクリプタの数を設定してください。 “1” 以上の値を設定してください。
ETHER_CFG_BUFSIZE ※デフォルト値は 1536	送信バッファ、受信バッファのサイズを設定してください。 バッファは 32 バイト境界で配置しますので、32 バイト単位の値を設定してください。
ETHER_CFG_EINT_INT_PRIORITY ※デフォルト値は 2	EINT 割り込みの優先レベルを設定してください。 “1” ~ “15” の範囲で設定してください。
ETHER_CFG_CH0_PHY_ACCESS ※デフォルト値は 0	ETHERC チャンネル 0 が使用する PHY のアクセスチャネルを設定してください。 “0” の場合、PHY のレジスタアクセスは ETHERC0 を使用します。 “1” の場合、PHY のレジスタアクセスは ETHERC1 を使用します。

ETHER_CFG_PHY_MII_WAIT ※デフォルト値は 8	PHY-LSI のリード/ライトに使用しているソフトウェア ループのループ回数を設定します。ループ回数をご使用 する PHY-LSI に合わせて設定してください。 “1” 以上の値を設定してください。
ETHER_CFG_PHY_DELAY_RESET ※デフォルト値は 0x00020000	PHY-LSI のリセット完了待ちのタイムアウト処理に使用 しているループ回数を設定します。ループ回数をご使用 する PHY-LSI に合わせて設定してください。
ETHER_CFG_LINK_PRESENT ※デフォルト値は 0	PHY-LSI から出力されるリンク信号の極性を設定してく ださい。 “0” の場合、LINKSTA 信号の立ち下がり／立ち上がりで、 リンクアップ／リンクダウンとなります。 “1” の場合、LINKSTA 信号の立ち上がり／立ち下がりで、 リンクアップ／リンクダウンとなります。
ETHER_CFG_USE_LINKSTA ※デフォルト値は 0	リンク状態変化の検出において、LINKSTA 信号の代わり に PHY-LSI のステータスレジスタを使用するかを設定し てください。 “0” の場合、PHY-LSI のステータスレジスタを使用しま す。 “1” の場合、LINKSTA 信号を使用します。
ETHER_CFG_USE_PHY_KSZ8041NL ※デフォルト値は 0	Micrel 社の PHY-LSI KSZ8041NL を使用するかしないか を設定してください。 “0” の場合、KSZ8041 を使用しません。 “1” の場合、KSZ8041 を使用します。

5.6 セクション配置

5.6.1 CCRX for Renesas RX のセクション設定例

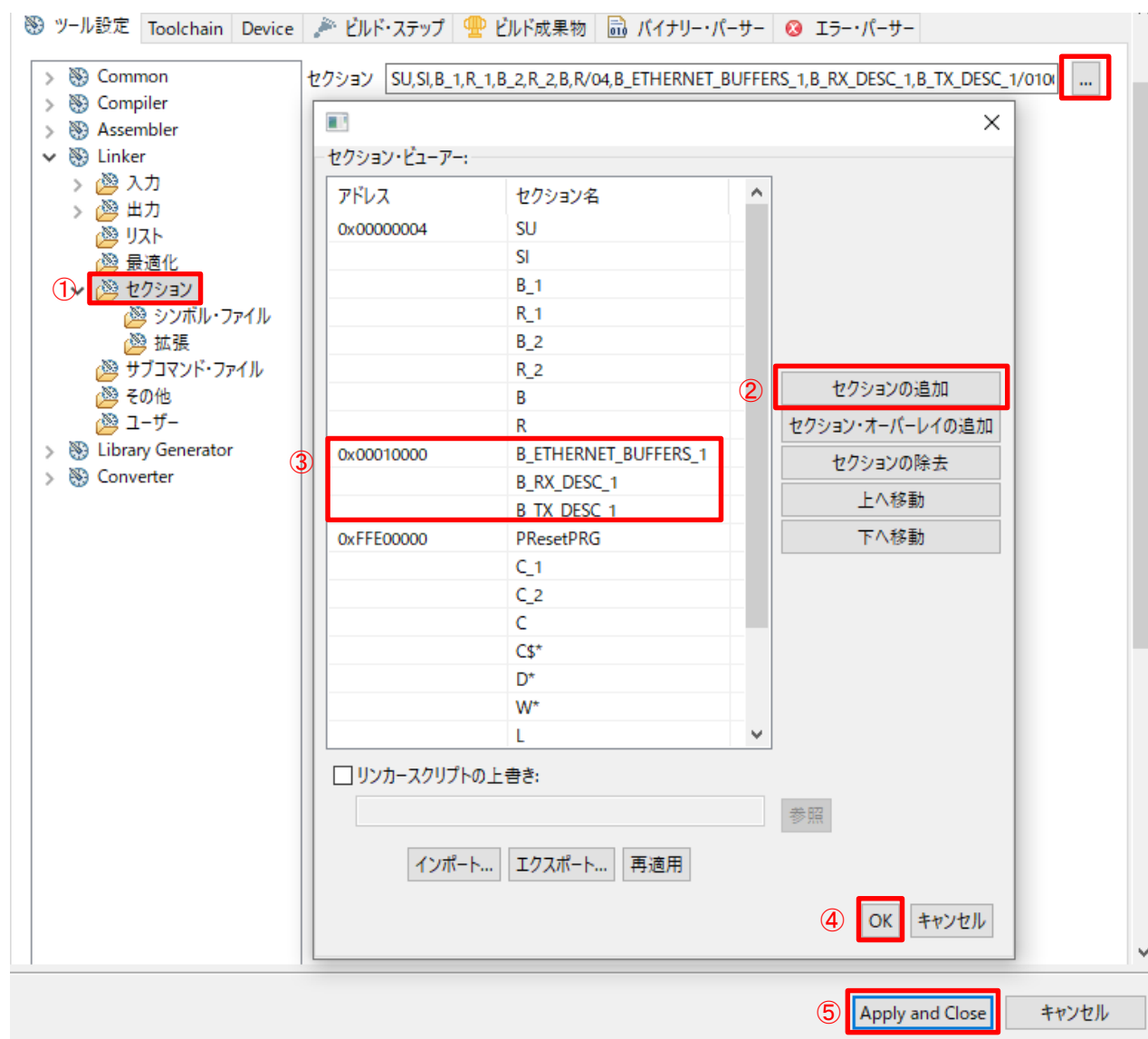
e² studio のリンクの設定で、以下の設定をしてください。

- セクションの追加

セクションの追加

RAM に B_ETHERNET_BUFFERS_1、B_RX_DESC_1、B_TX_DESC_1 のセクションを追加します。

- ① 「セクション」をクリック
- ② 「セクションの追加」をクリック
- ③ 「アドレス」欄と「セクション名」欄にアドレスとセクション名を入力
- ④ 「OK」をクリック
- ⑤ 「Apply and Close」をクリック



5.6.2 GCC for Renesas RX のセクション設定例

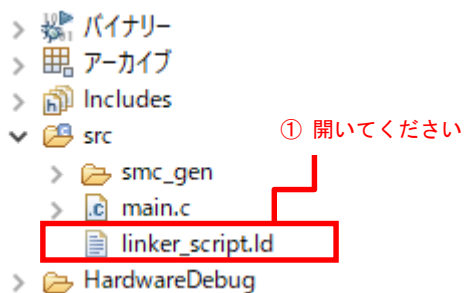
linker_script.ld ファイルを編集して、セクションおよびシンボルを追加してください。

セクションおよびシンボルの追加

下記のコードを追加。

```
B_ETHERNET_BUFFERS_1 0x00010000 (NOLOAD) : AT(0x00010000)
{
    _B_ETHERNET_BUFFERS_1_start = .;
    *(B_ETHERNET_BUFFERS_1)
    _B_ETHERNET_BUFFERS_1_end = .;
} >RAM
B_RX_DESC_1 (NOLOAD) :
{
    _B_RX_DESC_1_start = .;
    *(B_RX_DESC_1)
    _B_RX_DESC_1_end = .;
} >RAM
B_TX_DESC_1 (NOLOAD) :
{
    _B_TX_DESC_1_start = .;
    *(B_TX_DESC_1)
    _B_TX_DESC_1_end = .;
} >RAM
```

①プロジェクトエクスプローラーから「linker_script.ld」を開く。



② 「linker_script.ld」 をクリック。

③ コードを入力。

```
125 {
126     *(.gcc_exc)
127 } > RAM
128 .bss :
129 {
130     _bss = .;
131     *(.bss)
132     *(.bss.**
133     *(COMMON)
134     *(B)
135     *(B_1)
136     *(B_2)
137     _ebss = .;
138     _end = .;
139 } > RAM
140 B_ETHERNET_BUFFERS_1 0x00010000 (NOLOAD) : AT(0x00010000)
141 {
142     _B_ETHERNET_BUFFERS_1_start = .;
143     *(B_ETHERNET_BUFFERS_1)
144     _B_ETHERNET_BUFFERS_1_end = .;
145 } >RAM
146 B_RX_DESC_1 (NOLOAD) :
147 {
148     _B_RX_DESC_1_start = .;
149     *(B_RX_DESC_1)
150     _B_RX_DESC_1_end = .;
151 } >RAM
152 B_TX_DESC_1 (NOLOAD) :
153 {
154     _B_TX_DESC_1_start = .;
155     *(B_TX_DESC_1)
156     _B_TX_DESC_1_end = .;
157 } >RAM
158 .ofs1 0xFE7F5D40: AT(0xFE7F5D00)
159 {
160     KEEP(*(.ofs1))
161 } > OFS
```

③ 入力してください

② クリックしてください

5.6.3 IAR C/C++ Compiler for Renesas RX のセクション設定例

icf ファイルを編集して、セクション設定を追加してください。

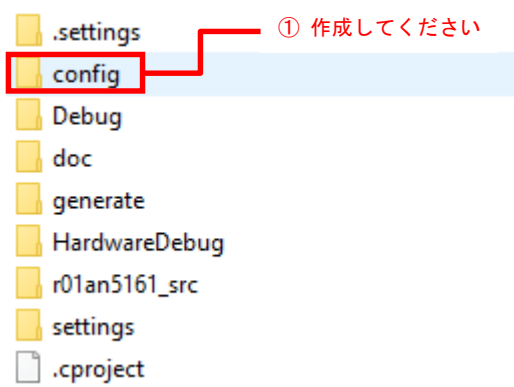
編集する icf ファイルはプロジェクトのターゲットデバイスにより異なりますので、使用するデバイスの型名上 8 桁を確認して編集してください。

例として RX63N(R5F563NFDxFC)では「Inkr5f563nf.icf」を編集します。

以下に RX63N(R5F563NFDxFC)での編集例を説明します。

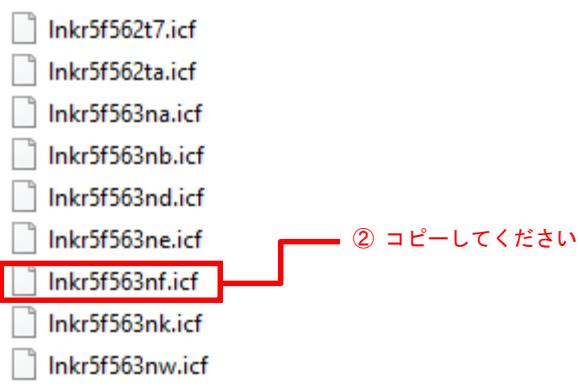
セクション設定の追加

- ① プロジェクトフォルダに「config」フォルダを作成。



- ② IAR C/C++ Compiler for Renesas RX(以下「EWRX」と記載)をインストールしたフォルダの「¥rx¥config」から「Inkr5f563nf.icf」をプロジェクトフォルダの「config」フォルダにコピー。

インストール時のデフォルトは「C:¥Program Files (x86)¥IAR Systems¥Embedded Workbench 8.1」です。



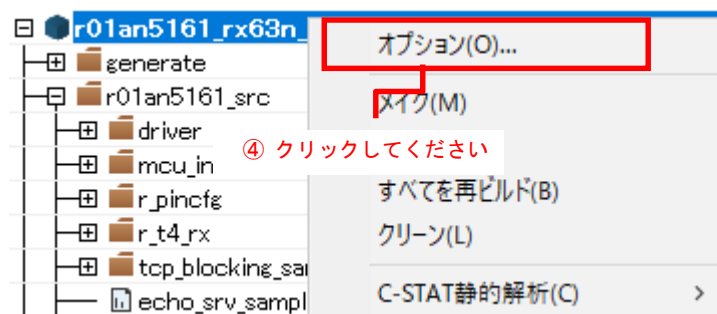
- ③ コピーした「Inkr5f563nf.icf」ファイルを開き以下のコードを追加。

```
place at address mem:0x00010000 { rw section B_ETHERNET_BUFFERS_1, rw section
B_RX_DESC_1, rw section B_TX_DESC_1 };
```

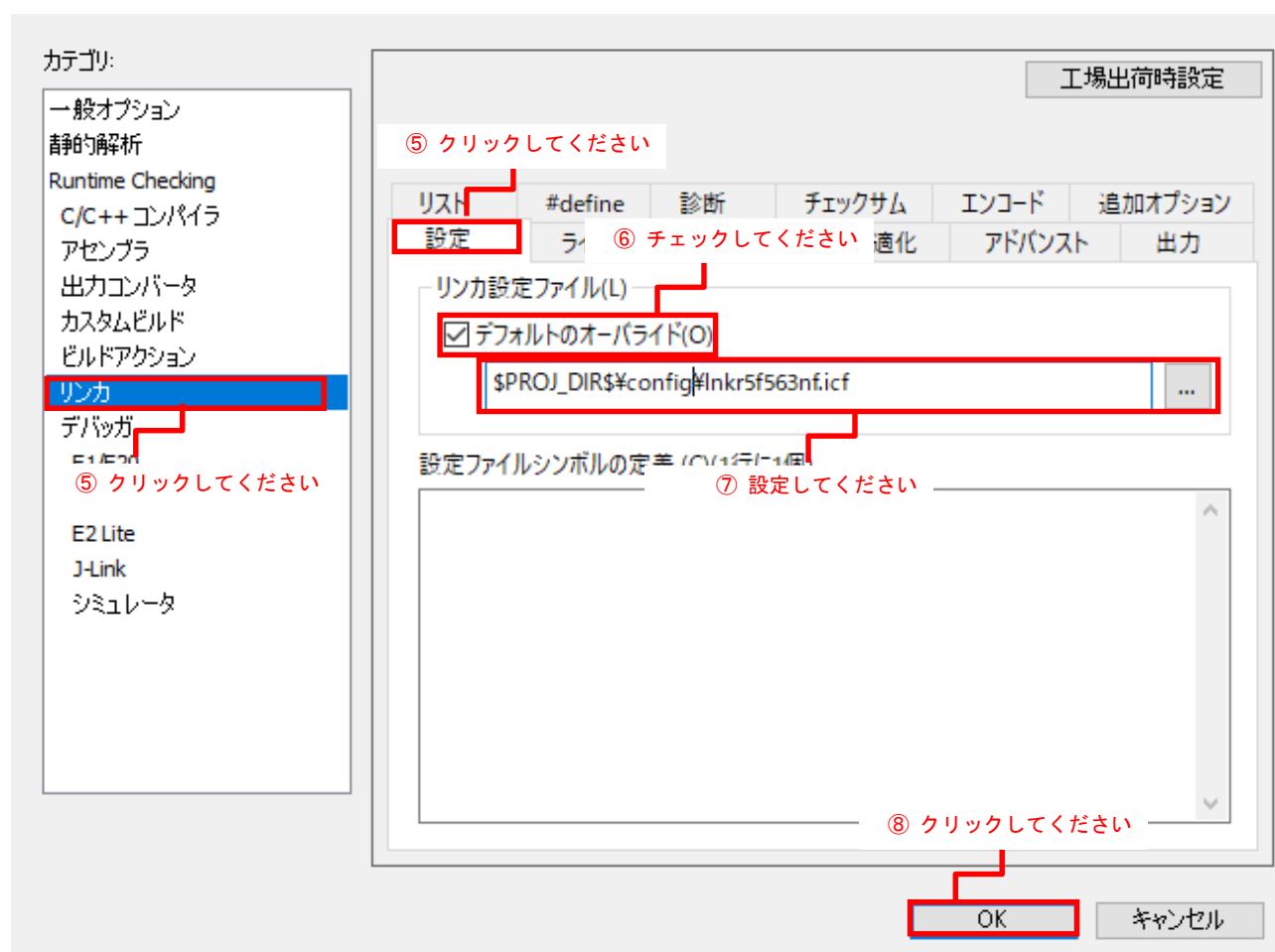
```
place at address mem:0xFFFFF800 { .ro section .resetvect.};
place at address mem:0xFFFFF800 { .ro section .exceptvect.};
place at address mem:0x00010000 { .rw section B_ETHERNET_BUFFERS_1, .rw section B_RX_DESC_1, .rw section B_TX_DESC_1.};
"ROM16":place in ROM_region16 { .rw section .code16*,
.ro section .data16*};
"RAM16":place in RAM_region16 { .rw section .code16*,
.ro section .data16*};
```

③ 追加してください

- ④EWRX から使用するプロジェクトを開き、ワークスペースのプロジェクトを右クリックし
オプションを開く。



- ⑤「カテゴリ：リンカ」を選択し、「設定」をクリック。
⑥「デフォルトのオーバーライド」にチェックを入れる。
⑦ ③で編集したファイルを参照先に設定する。
⑧ 「OK」をクリック。



5.7 オプション設定メモリ

表 5-5 にサンプルコードで使用するオプション設定メモリの状態を示します。必要に応じて、お客様のシステムに最適な値を設定してください。

表 5-5 サンプルコードで使用するオプション設定メモリ（注 1）

シンボル	アドレス	設定値	内容
OFS0	FFFF FF8Fh to FFFF FF8Ch	FFFF FFFFh	リセット後、IWDT は停止 リセット後、WDT は停止
OFS1	FFFF FF8Bh to FFFF FF88h	FFFF FFFFh	リセット後、電圧監視 0 リセット無効 リセット後、HOCO 発振が無効
MDES	FFFF FF83h to FFFF FF80h	FFFF FFFFh	リトルエンディアン

注 1：ただ RX63N のオプション設定メモリを記載しています。

6. サンプルコード

サンプルコードは、ルネサス エレクトロニクスホームページから入手してください。

7. 注意事項

7.1 T4 ライブラリ

- (1) tcp_rcv_dat()および、tcp_snd_dat()の第三引数「INT len」には 15bit 以内のサイズを指定してください。
- (2) tcp_acp_cep()、tcp_con_cep()、tcp_cls_cep()、tcp_rcv_dat()、tcp_snd_dat()、udp_snd_dat()、および udp_rcv_dat()の引数「TMO tmout」に正の値を指定する場合、15bit 以内のサイズを指定してください。
- (3) 本ライブラリは、マイコンオプション fint_register=0 (高速割り込み専用レジスタ [なし]) で使用してください。本オプションの省略時解釈は、fint_register=0 です。

7.2 サンプルプログラム

- (1) サンプルプログラムはリトルエンディアンのみ付属しています。
- (2) サンプルプログラムの MAC アドレスは config_tcpudp.c の_myethaddr 変数に格納されています。
_myethaddr 変数(MAC アドレス)の初期値は必要に応じてシステムに合わせて変更してください。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2019.12.22	-	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. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

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. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「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。

5. クロックについて

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。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 V_{IL} (Max.) から V_{IH} (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 V_{IL} (Max.) から V_{IH} (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

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、将来の拡張機能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。

8. 製品間の相違について

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、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ幅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生じた損害（お客様または第三者いずれに生じた損害も含まれます。以下同じです。）に関し、当社は、一切その責任を負いません。
 2. 当社製品、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またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
 3. 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、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング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）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等
当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、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（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
 6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合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、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他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。かかる法令を遵守しないこと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。
 9. 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。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
 10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
 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。
 12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 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直接的、間接的に支配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。

(Rev.4.0-1 2017.11)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。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。