RX ファミリ

組み込み用 TCP/IP M3S-T4-Tiny 導入ガイド
Firmware Integration Technology

要旨

本資料は、RX ファミリ組み込み用 TCP/IP M3S-T4-Tiny V.2.10 (以下、T4 と略します)を導入し、使用するために必要な情報をまとめています。本資料を導入ガイドと呼びます。

T4 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。通信に使用するマイコンの周辺機能は、Ethernet の場合、内蔵 Ethernet コントローラ、または外部 Ethernet コントローラ IC と接続するための外部バスです。RX ファミリで Ethernet を実現する場合、Ethernet コントローラを内蔵している RX64M または RX71M または RX65N を推奨します。

各種 Renesas Starter Kit+同梱の CPU ボードや、がじぇっとるねさすの RX65N 搭載ボード（GR-ROSE）や、サードパーティ製ボードで簡単に 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

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/jp/ja/software-tool/fit

[実機動作確認における注意事項]

本アプリケーションノートは TCP/IP 機能を FIT モジュール化したものです。実機動作確認可能なサンプルは含まれませんのでご注意ください。T4 FIT モジュールを用いて実機動作確認可能な状態に組み上げたサンプルは順次下記 URL 内にアップロードしていきます。

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

以下の図は T4 を使用したソフトウェア構造、4 種類の例です。以下の条件により、ソフトウェア構造が異なります。

- ITRON TCP/IP APIs Style / Socket APIs style
- TSIP(Trusted Secure IP)使用(注) / 未使用

    注：MCU が TSIP を搭載している場合に使用できます。

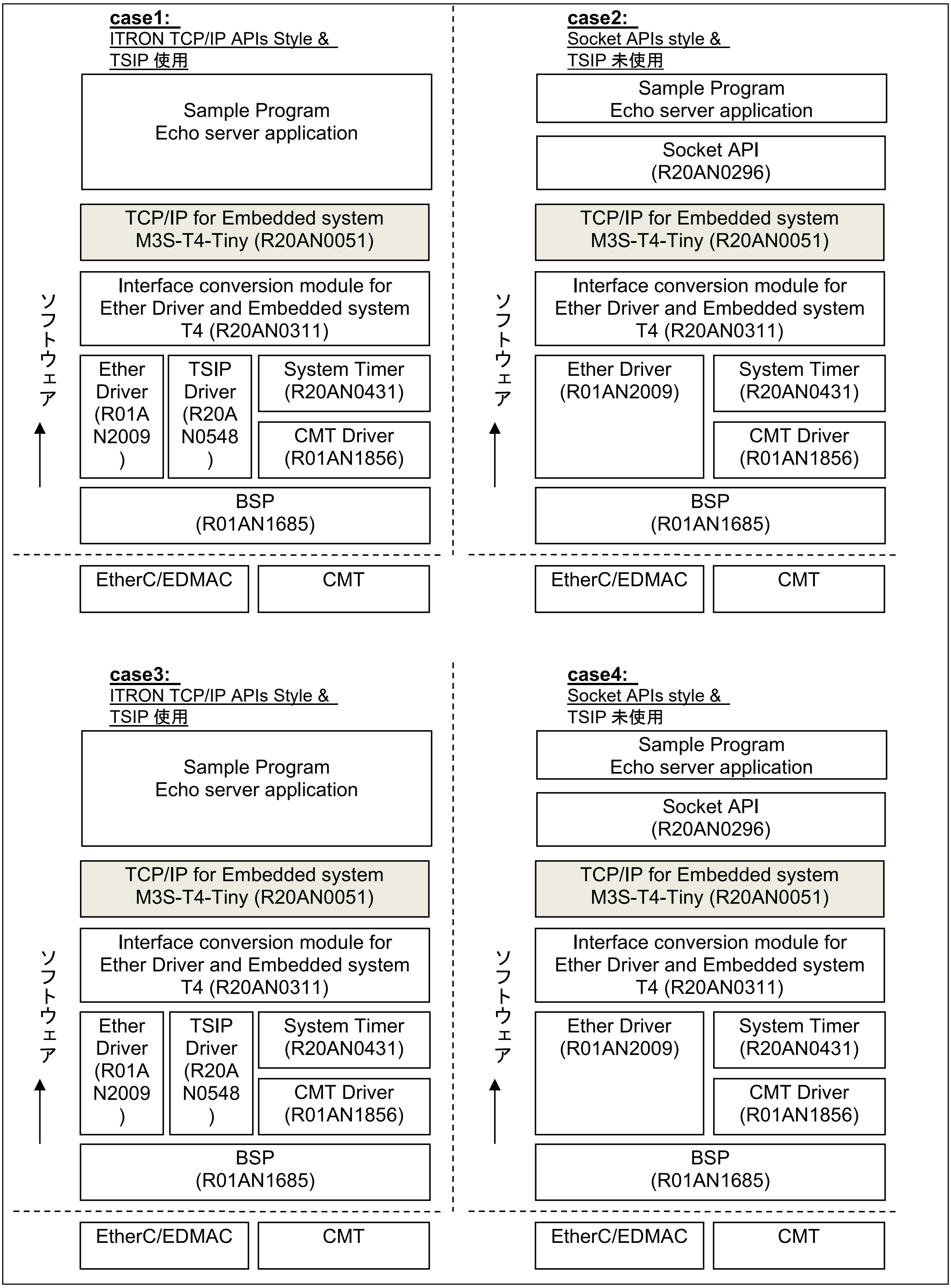


図 1-1 T4 ソフトウェア構成

[T4 仕様上の注意事項]

T4 は、比較的簡易なアプリケーションを搭載することを想定しています。Linux 用のネットワークアプリケーションの移植を想定したソケットインタフェースや、IPSec や IPv6 などの次世代 IP 技術、ICMP によるエラー通知やルーティングプロトコルなどのルータ用機能は搭載しておりません。

T4 は最新の脆弱性対応を実施しない場合があります。そのため、T4 はインターネット接続ではなく、ローカル接続を目的とした通信に適しています。

動作確認デバイス

RX ファミリ

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 7.1 動作確認環境を参照してください。

GCC for Renesas RX にはバージョンによる使用制限があります。1.6 制限事項を参照してください。

関連ドキュメント

- Firmware Integration Technology ユーザーズマニュアル(R01AN1833)
- ボードサポートパッケージモジュール Firmware Integration Technology (R01AN1685)
- e^2 studio に組み込む方法 Firmware Integration Technology (R01AN1723)
- CS+に組み込む方法 Firmware Integration Technology (R01AN1826)
- RX スマート・コンフィグレータ ユーザーガイド: e^2 studio 編(R20AN0451)
- RX スマート・コンフィグレータ ユーザーガイド: CS+編(R20AN0470)
- RX スマート・コンフィグレータ ユーザーガイド: IAREW 編(R20AN0535)
- RX ファミリ Ethernet ドライバと組み込み用 TCP/IP M3S-T4-Tiny のインタフェース変換モジュール Firmware Integration Technology(R20AN0311)
- RX ファミリ イーサネットモジュール Firmware Integration Technology(R01AN2009)
- RX ファミリ システムタイマモジュール Firmware Integration Technology(R20AN0431)
- RX Family CMT Module Using Firmware Integration Technology(R01AN1856)
- RX ファミリ TSIP(Trusted Secure IP)モジュール Firmware Integration Technology（バイナリ版）(R20AN0548)
- 組み込み用 TCP/IP M3S-T4-Tiny ユーザーズマニュアル(R20UW0031)
- 組み込み用 TCP/IP M3S-T4-Tiny: Ethernet ドライバインタフェース仕様書(R20UW0032)

目次

1. 概要

1.1 T4 FIT モジュールとは

本モジュールは API として、プロジェクトに組み込んで使用します。本モジュールの組み込み方については、「2.12FIT モジュールの追加方法」を参照してください。

1.2 T4 FIT モジュールの概要

T4 FIT モジュールは、RX ファミリマイコン用 TCP/IP 通信プロトコルソフトウェアライブラリです。

RX マイコンを搭載したあらゆる機器で TCP/IP 通信を実現できます。

ソフトウェア名：RX ファミリ 組み込み用 TCP/IP M3S-T4-Tiny V.2.10

1.3 API の概要

表 1.1 に本モジュールに含まれる API 関数を示します。

表 1.1 API 関数一覧

関数	関数説明
tcp_acp_cep	TCP 接続要求(受動オープン)
tcp_con_cep	TCP 接続要求（能動オープン）
tcp_rcv_dat	TCP データの受信
tcp_snd_dat	TCP データの送信
tcp_sht_cep	TCP データ送信の終了
tcp_cls_cep	TCP 通信端点のクローズ
tcp_can_cep	TCP ペンディングしている処理のキャンセル
udp_rcv_dat	UDP データの受信
udp_snd_dat	UDP データの送信
udp_can_cep	UDP ペンディングしている処理のキャンセル
tcpudp_get_ramsize	T4 が使用するワーク領域のサイズの計算
tcpudp_open	T4 ライブラリの初期化を行います。
_process_tcpip	TCP/IP の処理を行います。
tcpudp_close	T4 ライブラリの終了処理を行います。
tcpudp_reset	T4 ライブラリのリセット
igmp_join_group	マルチキャストグループへの参加
igmp_leave_group	マルチキャストグループからの離脱

API の詳細はユーザーズマニュアル(R20UW0031)をご参照ください。ユーザーズマニュアルはライブラリの使用方法および API を説明します。

また、Ethernet ドライバインタフェース仕様書(R20UW0032)はライブラリから呼び出されるユーザ定義関数の作成方法について説明します。

1.4 ファイル構成

本アプリケーションノートは、以下の表 1-2 のファイルが含まれます。

表 1.2　ファイル構成 1

ファイル/ディレクトリ名	内容
r20an0051xx0210-rx-t4-connectivity.zip	
r20an0051jj0210-rx-t4.pdf	導入ガイド(日本語、本書)
r20an0051ej0210-rx-t4.pdf	導入ガイド(英語)
FITDemos	
rskrx65n_2mb_tcp_blocking	[e2 studio](cc-rx) (rskrx65n-2mb)TCP ブロッキングコール
rskrx65n_2mb_tcp_nonblocking_cancel	[e2 studio](cc-rx) (rskrx65n-2mb)TCP ノンブロッキングキャンセルコール
rskrx65n_2mb_tcp_nonblocking	[e2 studio](cc-rx) (rskrx65n-2mb)TCP ノンブロッキングコール
rskrx65n_2mb_udp_blocking	[e2 studio](cc-rx) (rskrx65n-2mb)UDP ブロッキングコール
rskrx65n_2mb_udp_nonblocking	[e2 studio](cc-rx) (rskrx65n-2mb)UDP ノンブロッキングコール
FITModules	
r_t4_rx_v2.10.xml	xml ファイル
r_t4_rx_v2.10.zip	FIT モジュール本体
r_t4_rx_v2.10_extend.mdf	mdf ファイル

r_t4_rx_v2.10.zip を解凍したフォルダには、以下の表 1-3 のファイルが含まれます。

表 1.3 T4 FIT Modules 構成

ファイル/ディレクトリ名	内容
T4 FIT Modules (r_t4_rx_v.2.10.zip)	
T4 コンフィグ (r_config)	
r_t4_rx_config.h	T4 コンフィグヘッダ
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)
r20uw0032jj0109-t4tiny.pdf	ドライバインタフェース仕様書(日本語)
r20an0051jj0210-rx-t4.pdf	導入ガイド(日本語)
en	
r20uw0031ej0111-t4tiny.pdf	ユーザーズマニュアル(英語)
r20uw0032ej0109-t4tiny.pdf	ドライバインタフェース仕様書(英語)
r20an0051ej0210-rx-t4.pdf	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

1.5 処理例

図 5.1 main 関数の処理フロー参照

1.6 制限事項

GCC for Renesas RX に関して、T4 は GCC for Renesas RX 4.8.4.201803 以前のバージョンのみサポートします。これより後の GCC バージョンは”in_addr_t”型が追加されており、T4 との間で型の二重定義によるビルドエラーが発生するため、使用できません。

2. API 情報

本 FIT モジュールは、下記の条件で動作を確認しています。

2.1 ハードウェアの要求

ご使用になる MCU が以下の機能をサポートしている必要があります。

- 内蔵 Ethernet コントローラ、または外部 Ethernet コントローラ IC と接続された外部バス

2.2 ソフトウェアの要求

このドライバは以下の FIT モジュールに依存しています。

- r_t4_driver_rx

2.3 サポートされているツールチェーン

本 FIT モジュールは「7.1 動作確認環境」に示すツールチェーンで動作確認を行っています。

GCC for Renesas RX の使用には制限事項があります。1.6 制限事項を参照してください。

2.4 使用する割り込みベクタ

なし

2.5 ヘッダファイル

すべての API 呼び出しとそれをサポートするインタフェース定義は r_t4_itcpip.h に記載しています。

2.6 整数型

このドライバは ANSI C99 を使用しています。これらの型は stdint.h で定義されています。

2.7 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 で行います。

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 r_ t4_rx_config.h	
T4_CFG_SYSTEM_CHANNEL_NUMBER ※デフォルト値は 1	LAN ソケットの個数を設定します。 本設定は、_t4_channel_num に反映されます。
T4_CFG_SYSTEM_DHCP ※デフォルト値は 1	DHCP 機能の有効/無効を設定します。 1=有効、0=無効 本設定は、_t4_dhcp_enable に反映されます。
T4_CFG_FIXED_IP_ADDRESS_CHx ※x=0～(T4_CFG_SYSTEM_CHANNEL_NUMBER-1)	(T4_CFG_SYSTEM_DHCP が 1 の場合は設定不要です。) LAN ソケットごとの IP アドレスを設定します。 バイト間はコンマ(,)で区切ってください 記述例: 192,168,0,3 本設定は、tcpudp_env[].ipaddr に反映されます。
T4_CFG_FIXED_SABNET_MASK_CHx ※x=0～(T4_CFG_SYSTEM_CHANNEL_NUMBER-1)	(T4_CFG_SYSTEM_DHCP が 1 の場合は設定不要です。) LAN ソケットごとのサブネットマスクを設定します。 バイト間はコンマ(,)で区切ってください 記述例: 255,255,255,0 本設定は、tcpudp_env[].maskaddr に反映されます。
T4_CFG_FIXED_GATEWAY_ADDRESS_CHx ※x=0～(T4_CFG_SYSTEM_CHANNEL_NUMBER-1)	(T4_CFG_SYSTEM_DHCP が 1 の場合は設定不要です。) LAN ソケットごとのゲートウェイアドレスを設定します。 バイト間はコンマ(,)で区切ってください 記述例: 0,0,0,0 本設定は、tcpudp_env[].gwaddr に反映されます。
T4_CFG_ETHER_CHx_MAC_ADDRESS ※x=0～(T4_CFG_SYSTEM_CHANNEL_NUMBER-1)	LAN ソケットごとの MAC アドレスを設定します。 バイト間はコンマ(,)で区切ってください 記述例: 0x74,0x90,0x50,0x10,0xFE,0x77 本設定は、_myethaddr に反映されます。
T4_CFG_SYSTEM_CALLBACK_FUNCTION_USE ※デフォルト値は 1	システムコールバックを登録するか否かを設定します。 1=有効、0=無効 本設定は g_fp_user に反映されます。
T4_CFG_SYSTEM_CALLBACK_FUNCTION_NAME_TMP	(T4_CFG_SYSTEM_CALLBACK_FUNCTION_USE が 0 の場合は設定不要です。) システムコールバックの関数名を登録します。 本設定は g_fp_user に反映されます。
T4_CFG_TCP_REPIDx_PORT_NUMBER ※x=1～4	TCP の受付口のポート番号を設定します。 本設定は tcp_crep[].myaddr.portno に反映されます。
T4_CFG_TCP_CEPIDx_CHANNEL ※x = (1~6)	TCP 端点毎に紐づけする LAN ソケット番号を設定します。 範囲は 0～(T4_CFG_SYSTEM_CHANNEL_NUMBER-1)です。 本設定は tcp_ccep[].lan_port_number に反映されます。
T4_CFG_TCP_CEPIDx_RECEIVE_WINDOW_SIZE ※x = (1~6)	TCP 端点毎の受信ウィンドウサイズを設定します。 本設定は tcp_ccep[].rbufsz に反映されます。
T4_CFG_TCP_CEPIDx_CALLBACK_FUNCTION_USE ※x = (1~6) デフォルト値は 0	TCP 端点毎のコールバックルーチンの使用有無を設定します。 端点動作をブロッキングコールで実行する場合は 0 を設定してください。 端点動作をノンブロッキングコールで実行する場合は 1 を設定してください。 本設定は tcp_ccep[].callback に反映されます。

T4_CFG_TCP_CEPIDx_CALLBACK_FUNCTION_NAME_TMP ※x = (1~6) ※デフォルト値は 0	TCP 端点動作をノンブロッキングコールで実行する場合のコールバックルーチン関数の名前を登録して下さい。 本設定は tcp_ccep[].callback に反映されます。
T4_CFG_TCP_CEPIDx_KEEPALIVE_ENABLE ※x = (1~6) ※デフォルト値は 0	TCP 端点毎の KEEPALIVE 機能の有効/無効を設定します。 0=無効、1=有効 本設定は tcp_ccep[].keepalive_enable に反映されます。
T4_CFG_TCP_MSS ※デフォルト値は 1460 (byte)	TCP 通信の MSS 値を設定します。 本設定は_tcp_mss に反映されます。
T4_CFG_TCP_2MSL_TIME ※デフォルト値は 60 (秒)	TCP 通信の 2MSL 時間を設定します。 本設定は_tcp_2msl に反映されます。
T4_CFG_TCP_MAX_TIMEOUT_PERIOD ※デフォルト値は 600 (秒)	TCP 通信の再送タイムアウト時間を設定します。 本設定は_tcp_rt_tmo_rst に反映されます。
T4_CFG_TCP_DIVIDE_SENDING_PACKET ※デフォルト値は 1	TCP 通信の多重送信機能を設定します。 1=有効、0=無効 本設定は_tcp_dack に反映されます。
T4_CFG_TCP_KEEPALIVE_START ※デフォルト値は 7200 (秒)	Keepalive の送信開始時間を設定します。 設定可能な範囲は 1~86400 です。 本設定は_tcp_keepalive_start に反映されます。
T4_CFG_TCP_KEEPALIVE_INTERVAL ※デフォルト値は 10 (秒)	Keepalive の再送信間隔時間を設定します。 設定可能な範囲は 1~86400 です。 本設定は_tcp_keepalive_interval に反映されます。
T4_CFG_TCP_KEEPALIVE_COUNT ※デフォルト値は 10 (回)	Keepalive パケット送信の最大回数を設定します。 設定可能な範囲は 0~0xFFFFFFFF です。 本設定は_tcp_keepalive_count に反映されます。
T4_CFG_UDP_CEPIDx_CHANNEL ※x=(1~6)	UDP 端点毎に紐づけする LAN ソケット番号を設定します。 範囲は 0～(T4_CFG_SYSTEM_CHANNEL_NUMBER-1)です。 本設定は udp_ccep[].lan_port_number に反映されます。
T4_CFG_UDP_CEPIDx_PORT_NUMBER ※x=(1~6)	UDP 端点毎にポート番号を設定します。 本設定は udp_ccep[].myaddr.portno に反映されます。
T4_CFG_UDP_CEPIDx_CALLBACK_FUNCTION_USE ※x=(1~6)	UDP 端点毎のコールバックルーチンの使用有無を設定します。 端点動作をブロッキングコールで実行する場合は 0 を設定してください。 端点動作をノンブロッキングコールで実行する場合は 1 を設定してください。 本設定は udp_ccep[].callback に反映されます。
T4_CFG_UDP_CEPIDx_CALLBACK_FUNCTION_NAME_TMP ※x=(1~6)	UDP 端点動作をノンブロッキングコールで実行する場合のコールバックルーチン関数の名前を登録して下さい。 本設定は udp_ccep[].callback に反映されます。
T4_CFG_UDP_MULTICAST_TTL ※デフォルト値は 1	マルチキャスト宛に送信する IP データグラムの TTL を設定します。 本設定は__multi_TTL に反映されます。
T4_CFG_UDP_BEHAVIOR_OF_RECEIVED_ZERO_CHECKSUM ※デフォルト値は 0	UDP ゼロチェックサムを受信したときの動作を設定します。 0:正常パケットとして扱う 1:異常パケットとして破棄する 本設定は_udp_enable_zerochecksum に反映されます。
T4_CFG_IP_ARP_CACHE_TABLE_COUNT ※デフォルト値は 3	ARP のキャッシュエントリ数を設定します。 本設定は_ip_tblcnt に反映されます。

2.8 コードサイズ

本モジュールの ROM サイズ、RAM サイズ、最大使用スタックサイズを下表に示します。

ROM (コードおよび定数) と RAM (グローバルデータ) のサイズは、ビルド時の「2.7 コンパイル時の設定」のコンフィギュレーションオプションによって決まります。

下表の値は下記条件で確認しています。

モジュールリビジョン: r_t4_rx Rev.2.10

コンパイラバージョン: Renesas Electronics C/C++ Compiler Package for RX Family V3.02.00

(統合開発環境のデフォルト設定に"-lang = c99"オプションを追加)

GCC for Renesas RX 4.8.4.201803

(統合開発環境のデフォルト設定に"-std=gnu99"オプションを追加)

IAR C/C++ Compiler for Renesas RX version 4.14.1

(統合開発環境のデフォルト設定)

コンフィグレーションオプション: デフォルト設定

ROM、RAM コードサイズ				
デバイス	分類	使用メモリ		
		Renesas Compiler	GCC	IAR Compiler
RX65N (TSIP 非搭載版)	ROM	**25,435 バイト**	**23,345 バイト**	**24,827 バイト**
	RAM	**227 バイト**	**227 バイト**	**213 バイト**
	スタック	**436 バイト**	**-**	**764 バイト**

ドライバ層やコールバックルーチンの実装、MCU の TSIP 機能の有無により、
スタックサイズは変化しますので、ユーザは CallWalker 等のスタック算出ツールを使用し、
スタックサイズの確認を行ってください。

2.9 引数

API 関数の引数である構造体の詳細はユーザーズマニュアルを参照してください。構造体は、API 関数のプロトタイプ宣言とともに r_t4_itcpip.h に記載されています。

2.10 戻り値

API 関数の戻り値の詳細はユーザーズマニュアルを参照してください。戻り値は、API 関数のプロトタイプ宣言とともに r_t4_itcpip.h で記載されています。

2.11 コールバック関数

コールバック関数の詳細はユーザーズマニュアルを参照してください。

2.12 FIT モジュールの追加方法

本モジュールは、使用するプロジェクトごとに追加する必要があります。ルネサスでは、Smart Configurator を使用した(1)、(3)、(5)の追加方法を推奨しています。ただし、Smart Configurator は、一部の RX デバイスのみサポートしています。サポートされていない RX デバイスについては(2)、(4)の方法を使用してください。

(1) e² studio 上で Smart Configurator を使用して FIT モジュールを追加する場合
e² studio の Smart Configurator を使用して、自動的にユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「Renesas e² studio スマート・コンフィグレータ ユーザーガイド (R20AN0451)」を参照してください。

(2) e² studio 上で FIT Configurator を使用して FIT モジュールを追加する場合
e² studio の FIT Configurator を使用して、自動的にユーザプロジェクトに FIT モジュールを追加することができます。詳細は、アプリケーションノート「RX ファミリ e² studio に組み込む方法 Firmware Integration Technology (R01AN1723)」を参照してください。

(3) CS+上で Smart Configurator を使用して FIT モジュールを追加する場合
CS+上で、スタンドアロン版 Smart Configurator を使用して、自動的にユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「Renesas e² studio スマート・コンフィグレータ ユーザーガイド (R20AN0451)」を参照してください。

(4) CS+上で FIT モジュールを追加する場合
CS+上で、手動でユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「RX ファミリ CS+に組み込む方法 Firmware Integration Technology (R01AN1826)」を参照してください。

2.13 for 文、while 文、do while 文について

本モジュールでは、レジスタの反映待ち処理等で for 文、while 文、do while 文（ループ処理）を使用しています。これらループ処理には、「WAIT_LOOP」をキーワードとしたコメントを記述しています。そのため、ループ処理にユーザがフェイルセーフの処理を組み込む場合は、「WAIT_LOOP」で該当の処理を検索できます。

以下に記述例を示します。

```
while 文の例：
/* WAIT_LOOP */
while(0 == SYSTEM.OSCOVFSR.BIT.PLOVF)
{
    /* The delay period needed is to make sure that the PLL has stabilized. */
}

for 文の例：
/* Initialize reference counters to 0. */
/* WAIT_LOOP */
for (i = 0; i < BSP_REG_PROTECT_TOTAL_ITEMS; i++)
{
    g_protect_counters[i] = 0;
}

do while 文の例：
/* Reset completion waiting */
do
{
    reg = phy_read(ether_channel, PHY_REG_CONTROL);
    count++;
} while ((reg & PHY_CONTROL_RESET) && (count < ETHER_CFG_PHY_DELAY_RESET)); /* WAIT_LOOP */
```

3. T4 ライブラリの情報

RX ファミリ用の TCP/IP ライブラリです。

3.1 コンパイルオプション

下記に示すコンパイルオプションにてライブラリを生成しています。

3.1.1 CC-RX

(リトルエンディアンの場合)

-isa=rxv1 -nofpu -lang=c99 -output=obj -obj_path=DefaultBuild -nologo

(デバッグ情報付きリトルエンディアンの場合)

-isa=rxv1 -nofpu -lang=c99 -output=obj -obj_path=Debug -debug -optimize=0 -nologo

(ビッグエンディアンの場合)

-isa=rxv1 -nofpu -endian=big -lang=c99 -output=obj -obj_path=DefaultBuild -nologo

(デバッグ情報付きビッグエンディアンの場合)

-isa=rxv1 -nofpu -endian=big -lang=c99 -output=obj -obj_path=Debug -debug -optimize=0 -nologo

3.1.2 GCC

(リトルエンディアンの場合)

-Os -ffunction-sections -fdata-sections -Wstack-usage=100 -misa=v1 -mlittle-endian-data -std=gnu99

(ビッグエンディアンの場合)

-Os -ffunction-sections -fdata-sections -Wstack-usage=100 -misa=v1 -mbig-endian-data -std=gnu99

3.1.3 IAR

(リトルエンディアンの場合)

--suppress_core_attribute -e -Oh --no_cross_call
--dlib_config "C:¥Program Files (x86)¥IAR Systems¥Embedded Workbench 8.2¥rx¥LIB¥dlrxflln.h"
--double=32 --data_model=f --endian l -D NDEBUG --core rxv1 --int=32 --fpu=none --align_func=1

(ビッグエンディアンの場合)

--suppress_core_attribute -e -Oh --no_cross_call
--dlib_config "C:¥Program Files (x86)¥IAR Systems¥Embedded Workbench 8.2¥rx¥LIB¥dlrxflbn.h"
--double=32 --data_model=f --endian b -D NDEBUG --core rxv1 --int=32 --fpu=none --align_func=1

3.2 バージョン情報

T4 では、R_t4_version 変数の library メンバに文字列でバージョン情報を格納しています。R_t4_version 変数は r_t4_itcpip.h に定義されています。また、本製品のライブラリに格納されているデータは以下の通りです。

```
extern const mw_version_t R_t4_version;
```

3.2.1 CC-RX

- RXV 1 コア(little endian)用ライブラリファイル(Ethernet 対応) :

```
compiler = 0x03020000
library = "M3S-T4-Tiny(Ethernet) version 2.10 for RXV1 LITTLE endian.(Mar 10
2021, 14:44:15)"
```

- RXV 1 コア(little endian)用デバッグ情報付きライブラリファイル(Ethernet 対応) :

```
compiler = 0x03020000
library = "M3S-T4-Tiny(Ethernet) version 2.10 for RXV1 LITTLE endian.(Mar 10
2021, 14:46:54)"
```

- RXV 1 コア(big endian)用ライブラリファイル(Ethernet 対応) :

```
compiler = 0x03020000
library = "M3S-T4-Tiny(Ethernet) version 2.10 for RXV1 BIG endian.(Mar 10
2021, 14:44:29)"
```

- RXV 1 コア(big endian)用デバッグ情報付きライブラリファイル(Ethernet 対応) :

```
compiler = 0x03020000
library = "M3S-T4-Tiny(Ethernet) version 2.10 for RXV1 BIG endian.(Mar 10
2021, 14:47:06)"
```

3.2.2 GCC

- RXV 1 コア(little endian)用ライブラリファイル(Ethernet 対応) :

```
compiler = 0x00040804
library = "M3S-T4-Tiny(Ethernet) version 2.10 for GCC RX LITTLE endian.(Mar 10
2021, 16:33:31)"
```

- RXV 1 コア(big endian)用ライブラリファイル(Ethernet 対応) :

```
compiler = 0x00040804
library = "M3S-T4-Tiny(Ethernet) version 2.10 for GCC RX BIG endian.(Mar 10
2021, 16:33:43)"
```

3.2.3 IAR

- RXV 1 コア(little endian)用ライブラリファイル(Ethernet 対応) :

```
compiler = 0x0000019E
library = "M3S-T4-Tiny(Ethernet) version 2.10 for IAR RXV1 LITTLE endian.(Mar
10 2021, 16:41:44)"
```

- RXV 1 コア(big endian)用ライブラリファイル(Ethernet 対応) :

```
compiler = 0x0000019E
library = "M3S-T4-Tiny(Ethernet) version 2.10 for IAR RXV1 BIG endian.(Mar 10
2021, 16:42:00)"
```

4. QE for TCP/IP

QE for TCP/IP は e2 studio のプラグインとして T4 のデバッグ情報を提供します。各通信端点に対し以下情報をリアルタイムに表示することができます。これによりユーザのデバッグ効率を上げることができます。

・TCP 状態遷移のどの状態か

・どの API が実行されているか

・どのようなエラーが発生しているかなど

以下からダウンロードすることができます。e2 studio に組み込んで使用することができます。

https://www.renesas.com/jp/ja/products/software-tools/tools/solution-toolkit/qe-qe-for-tcp-ip.html

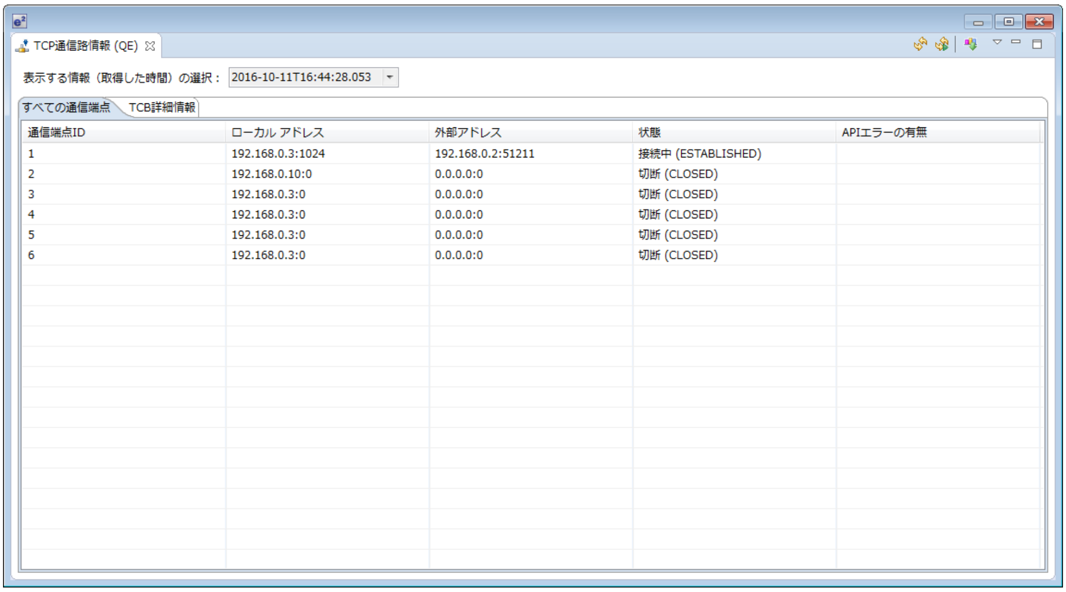


図 4-1　全通信端点のリスト表示例

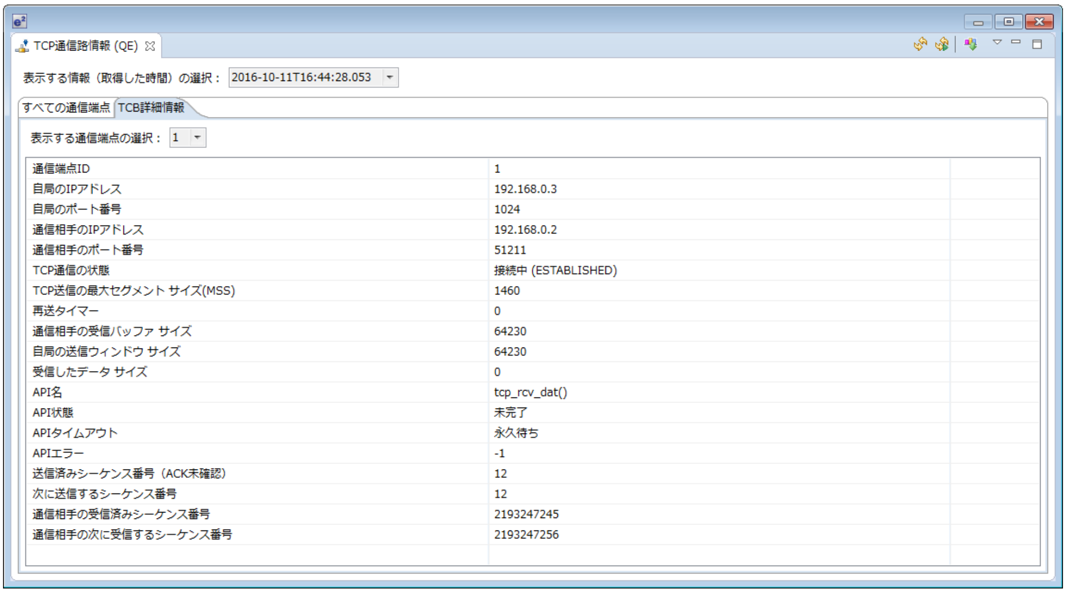


図 4-2　通信端点の詳細表示例

5. サンプルプログラム

以下の 5 パターンのエコーバックサーバのソースコードを含むプロジェクトを用意しています。

- 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。

5.1 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

5.1.1 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 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

5.2.1 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

5.2.2 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

5.3 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

5.3.1 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

5.4 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

5.4.1 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

5.4.2 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

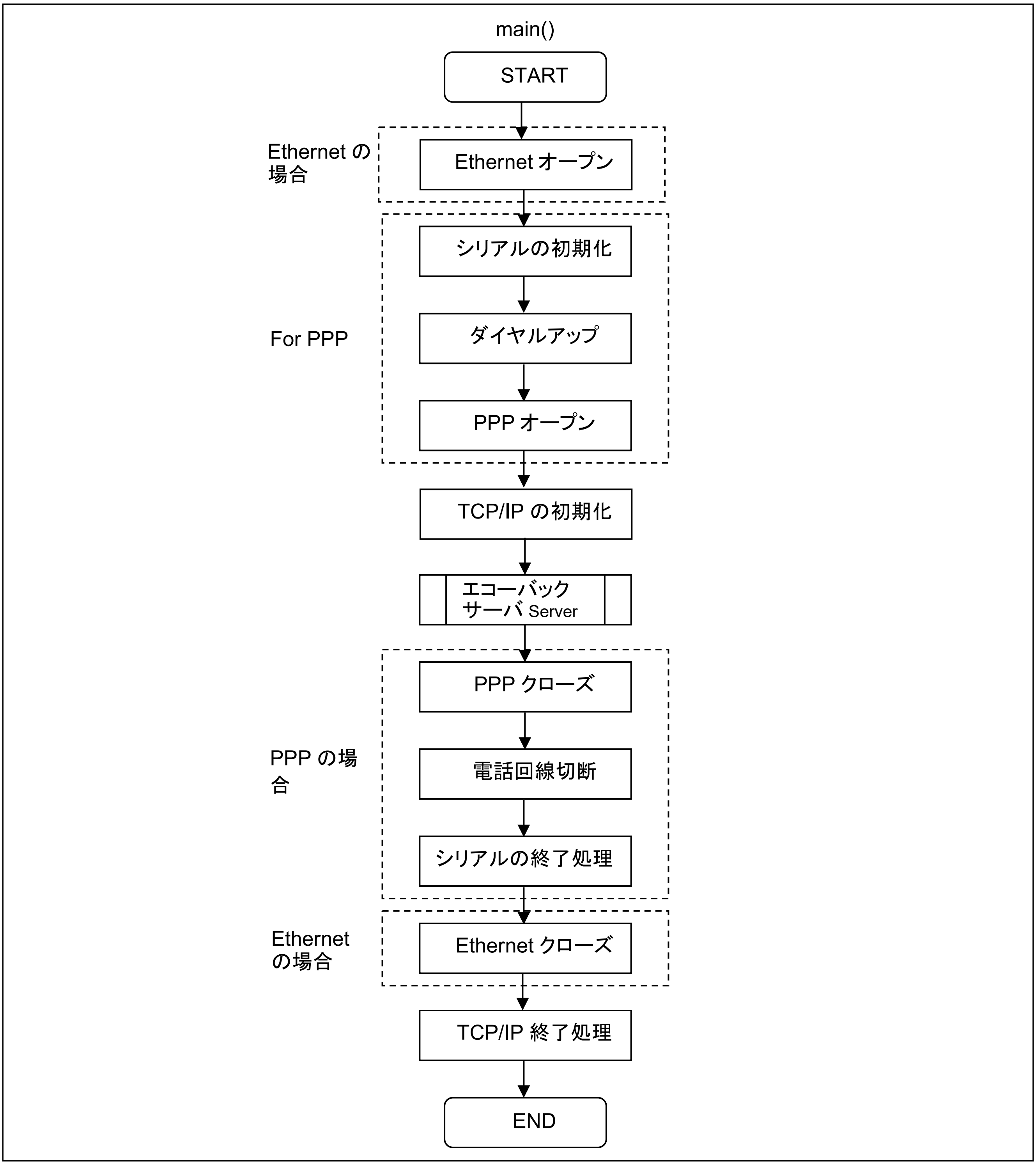


図 5.1　main 関数の処理フロー

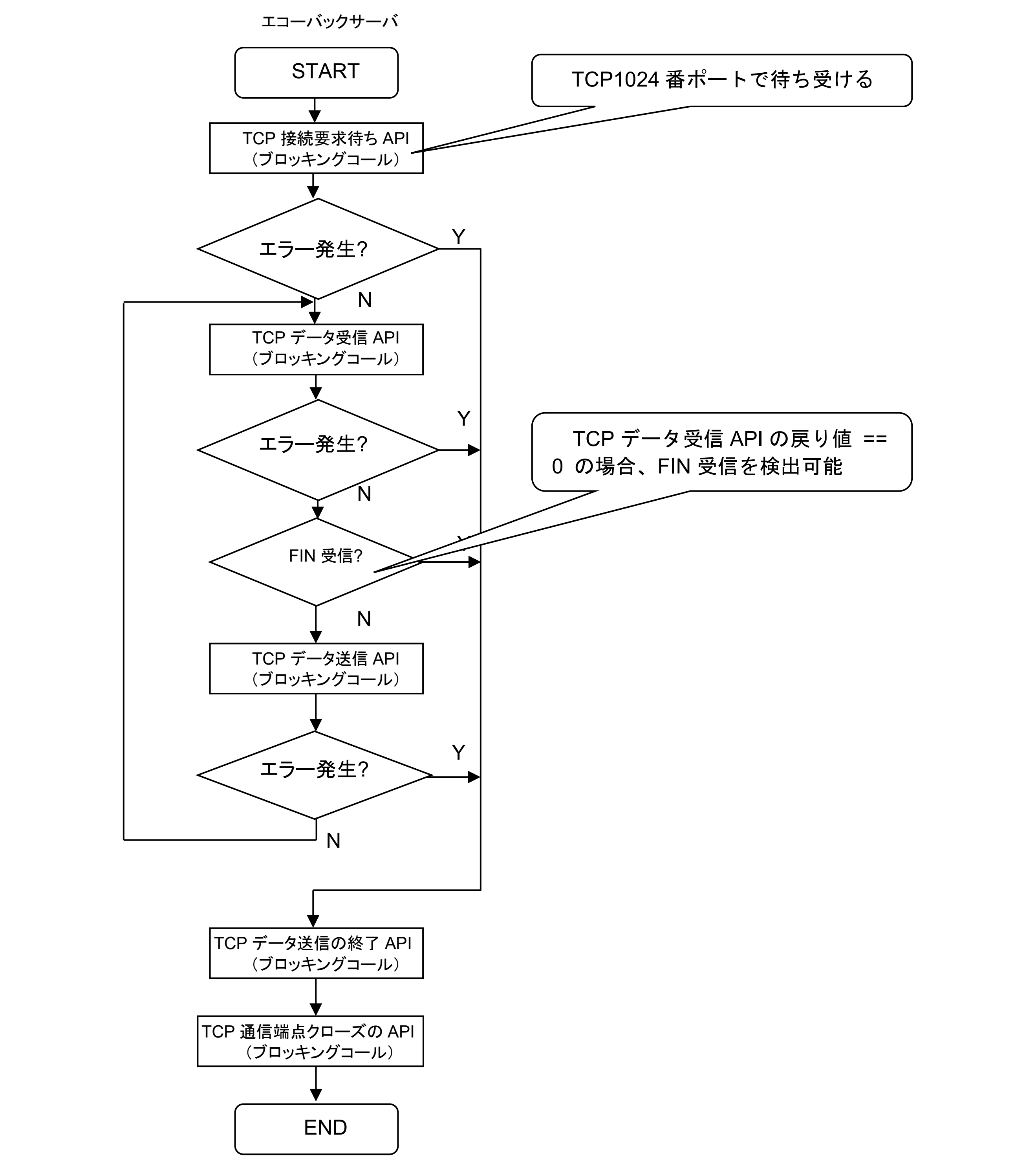


図 5.2　TCP のエコーバックサーバ（ブロッキングコール）の処理フロー

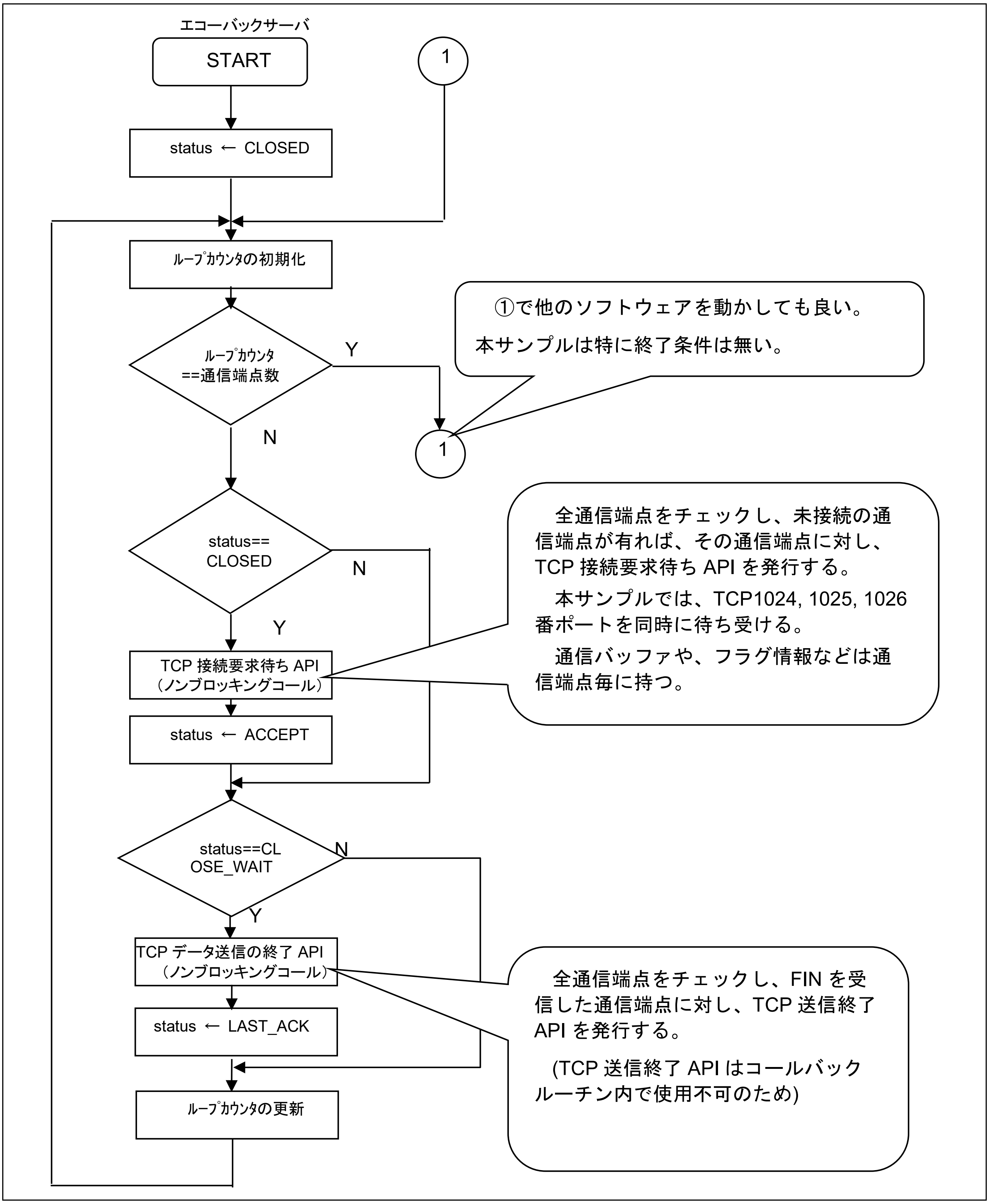


図 5.3　TCP のエコーバックサーバ（ノンブロッキングコール）の処理フロー

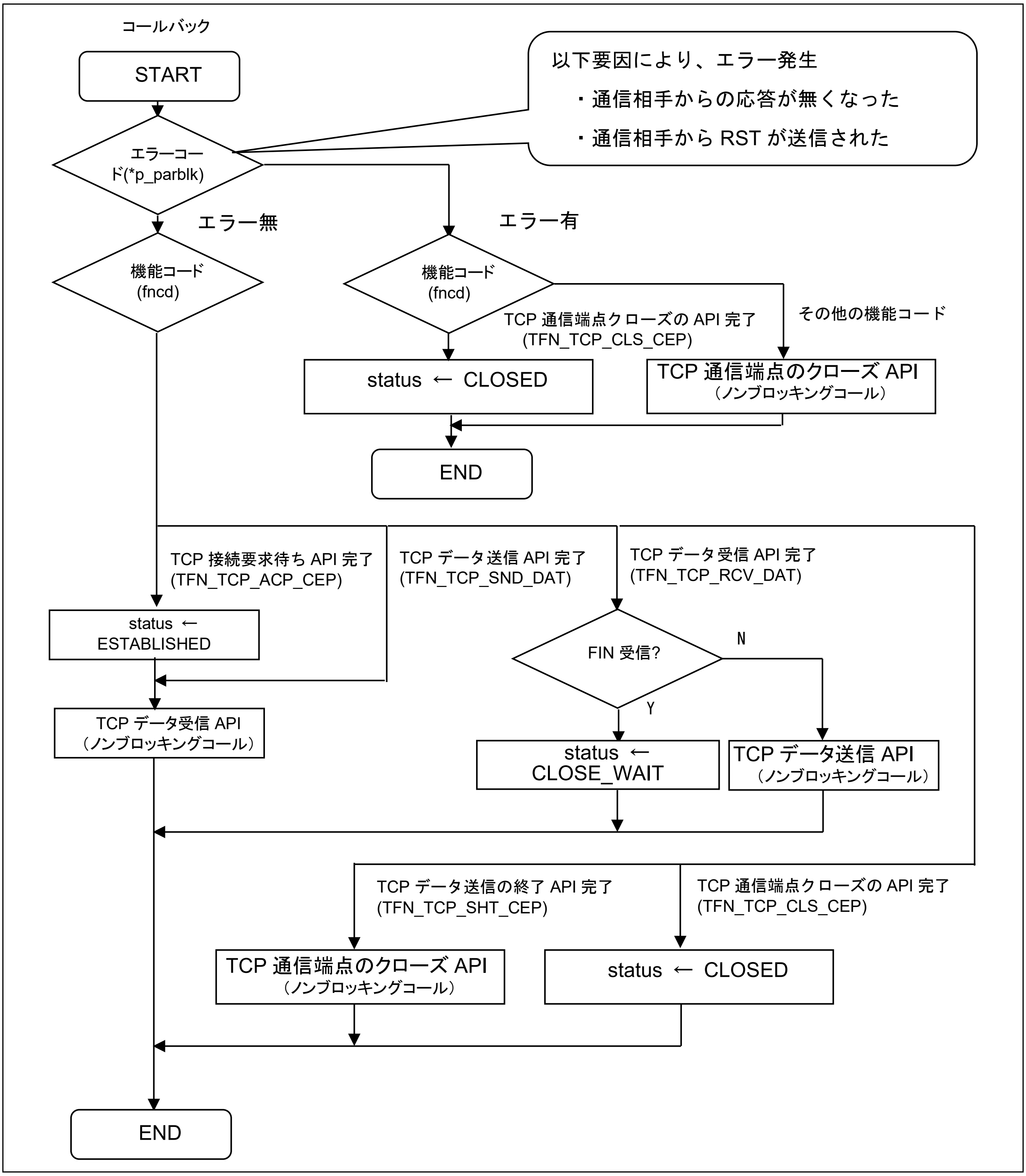


図 5.4　TCP のコールバック（ノンブロッキングコール）の処理フロー

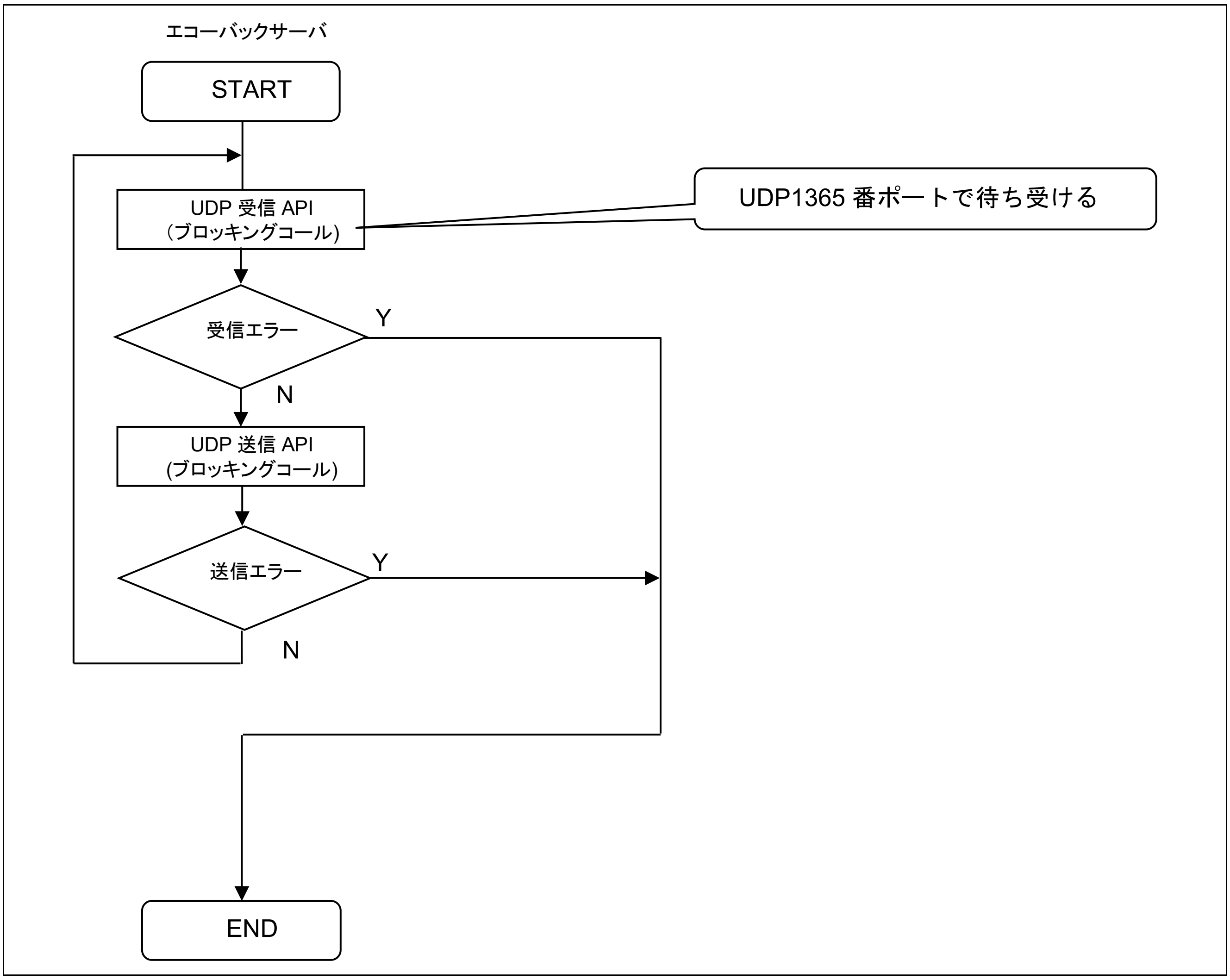


図 5.5　UDP のエコーバックサーバ（ブロッキングコール）の処理フロー

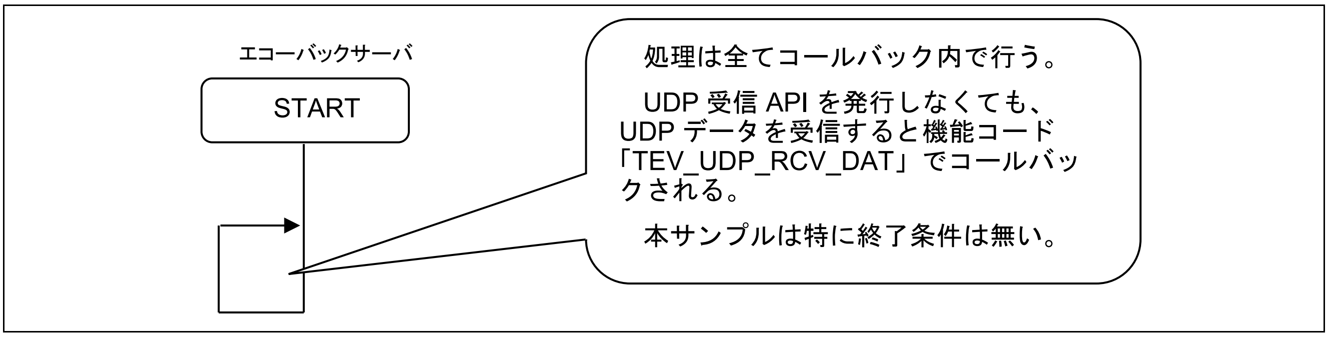


図 5.6　UDP のエコーバックサーバ（ノンブロッキングコール）の処理フロー

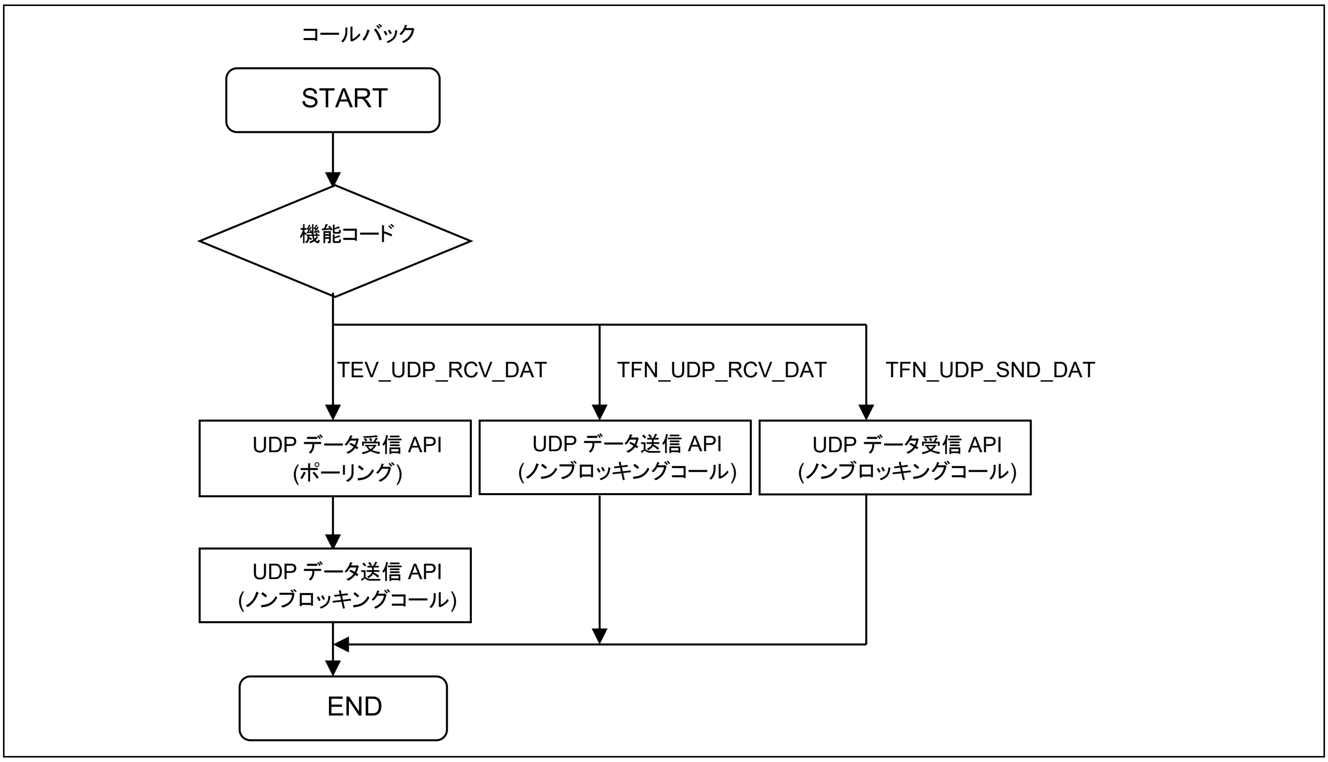


図 5.7　UDP のコールバック（ノンブロッキングコール）の処理フロー

5.5 コンポーネントの設定値

スマート・コンフィグレータでプロジェクトを組み立てた時、[コンポーネント]タブで設定を行います。

表 5.1 に初期値と設定例を記載します。

設定例は Renesas Starter Kit+ for RX64M を使ったサンプルプログラムで端点 3～6 をノンブロッキングコールで使用する場合を示します。

TCP は[TCP CEPIDx callback function name]にコールバック関数を登録してください。UDP は[UDP CEPIDy callback function name]にコールバック関数を登録して下さい。

※x は(1～6)の整数です。y は(1～6)の整数です。

表 5.1 コンポーネント設定値

Configurations	初期値	設定例	
		UDP ノンブロッキング	TCP ノンブロッキング
Channel number your system has.	1	1	1
Enable/Disable DHCP function.	1	1	1
IP address for ch0, when DHCP disable.	192,168,0,3	-	-
Subnet mask for ch0, when DHCP disable.	255,255,255,0	-	-
Gateway address for ch0, when DHCP disable.	0,0,0,0	-	-
IP address for ch1,when DHCP disable.	192,168,0,10	-	-
Subnet mask for ch1, when DHCP disable.	255,255,255,0	-	-
Gateway address for ch1, when DHCP disable.	0,0,0,0	-	-
Ether ch0 MAC address.	0x74,0x90,0x50 ,0x10,0xFE,0x77	0x74,0x90,0x50 ,0x10,0xFE,0x77	0x74,0x90,0x50 ,0x10,0xFE,0x77
Ether ch1 MAC address.	0x74,0x90,0x50 ,0x00,0x79,0x79	-	-
SYSTEM callback function use.	1	1	1
SYSTEM callback function name.	system_callback	system_callback	system_callback
TCP REPID1 port number.	1024	1024	1024
TCP REPID2 port number.	1025	1025	1025
TCP REPID3 port number.	1026	1026	1026
TCP REPID4 port number.	1027	1027	1027
TCP CEPID1 channel number.	0	0	0
TCP CEPID1 receive window size.	1460	1460	1460
TCP CEPID1 callback function use.	0	0	0
TCP CEPID1 callback function name.	0	-	-
TCP CEPID1 Keep-alive enable.	0	0	0
TCP CEPID2 channel number.	0	0	0
TCP CEPID2 receive window size.	1460	1460	1460
TCP CEPID2 callback function use.	0	0	0
TCP CEPID2 callback function name.	0	-	-
TCP CEPID2 Keep-alive enable.	0	0	0
TCP CEPID3 channel number.	0	0	0
TCP CEPID3 receive window size.	1460	1460	1460
TCP CEPID3 callback function use.	0	0	1
TCP CEPID3 callback function name.	0	-	tcp_nonblocking_callback
TCP CEPID3 Keep-alive enable.	0	0	0
TCP CEPID4 channel number.	0	0	0
TCP CEPID4 receive window size.	1460	1460	1460
TCP CEPID4 callback function use.	0	0	1
TCP CEPID4 callback function name.	0	-	tcp_nonblocking_callback

TCP CEPID4 Keep-alive enable.	0	0	0
TCP CEPID5 channel number.	0	0	1
TCP CEPID5 receive window size.	1460	1460	1460
TCP CEPID5 callback function use.	0	0	1
TCP CEPID5 callback function name.	0	-	tcp_nonblocking_callback
TCP CEPID5 Keep-alive enable.	0	0	0
TCP CEPID6 channel number.	0	0	1
TCP CEPID6 receive window size.	1460	1460	1460
TCP CEPID6 callback function use.	0	0	1
TCP CEPID6 callback function name.	0	-	tcp_nonblocking_callback
TCP CEPID6 Keep-alive enable.	0	0	0
TCP max segment size.	1460	1460	1460
TCP max segment life time.	60	60	60
TCP max timeout period time.	600	600	600
TCP divide the sending packet.	1	1	1
UDP CEPID1 channel number.	0	0	0
UDP CEPID1 port number.	1365	1365	1365
UDP CEPID1 callback function use	0	0	0
UDP CEPID1 callback name.	0	-	-
UDP CEPID2 channel number.	0	0	0
UDP CEPID2 port number.	1366	1366	1366
UDP CEPID2 callback function use.	0	0	0
UDP CEPID2 callback name.	0	-	-
UDP CEPID3 channel number.	0	0	0
UDP CEPID3 port number.	1367	1367	1367
UDP CEPID3 callback function use.	0	1	0
UDP CEPID3 callback name.	0	udp_nonblocking_callback	-
UDP CEPID4 channel number.	0	1	0
UDP CEPID4 port number.	1368	1368	1368
UDP CEPID4 callback function use.	0	1	0
UDP CEPID4 callback name.	0	udp_nonblocking_callback	-
UDP CEPID5 channel number.	0	1	0
UDP CEPID5 port number.	1369	1369	1369
UDP CEPID5 callback function use.	0	1	0
UDP CEPID5 callback name.	0	udp_nonblocking_callback	-
UDP CEPID6 channel number.	0	1	0
UDP CEPID6 port number.	1370	1370	1370
UDP CEPID6 callback function use	0	1	0
UDP CEPID6 callback name.	0	udp_nonblocking_callback	-
UDP multicast TTL value.	1	1	1
UDP behavior of received zero checksum.	0	0	0
IP ARP cache table.	3	3	3

5.6 RX65N 用サンプルプログラム環境

サンプルプログラム rskrx65n のプロジェクトは、e² studio に含まれている FIT モジュール組み込み機能を用いてプロジェクトを作成しています。

5.6.1 使用する FIT モジュール

使用している FIT モジュールは以下の通りです。

- RX ファミリ 組み込み用 TCP/IP M3S-T4-Tiny 導入ガイド Firmware Integration Technology (R20AN0051)
- RX ファミリ Ethernet ドライバと組み込み用 TCP/IP M3S-T4-Tiny のインタフェース変換モジュール Firmware Integration Technology (R20AN0311)
- RX ファミリ イーサネットモジュール Firmware Integration Technology (R01AN2009)
- RX ファミリ システムタイマモジュール Firmware Integration Technology (R20AN0431)
- RX Family CMT Module Using Firmware Integration Technology (R01AN1856)
- RX ファミリ ボードサポートパッケージモジュール Firmware Integration Technology (R01AN1685)

上記 FIT モジュールの詳細は、各 FIT モジュールに含まれるドキュメントを参照してください。

5.6.2 ソフトウェア構成

サンプルプログラムのソフトウェアおよび FIT モジュールの構成図を示します。

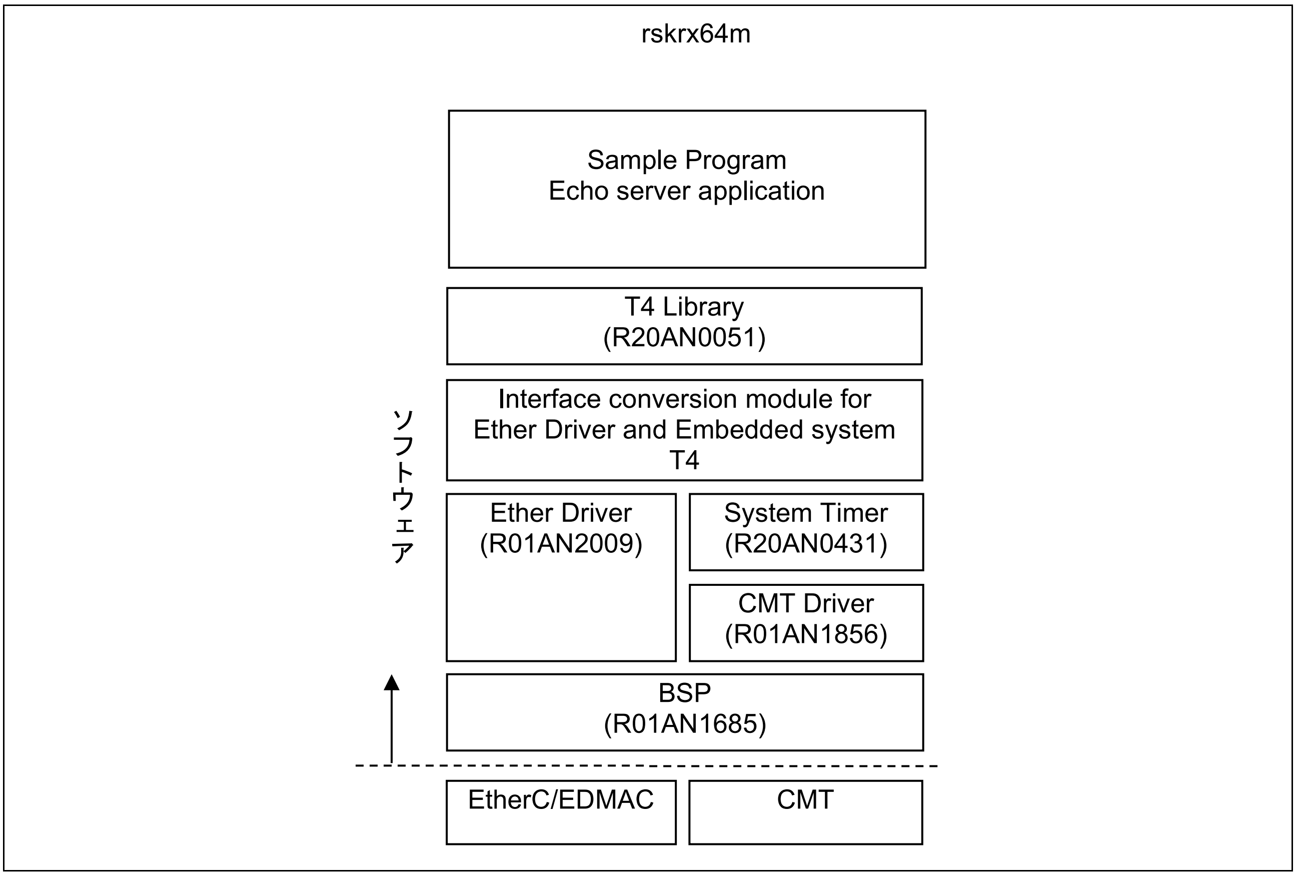


図 5-8　FIT モジュール構成図

5.6.3 ライブラリファイルのリンク方法

T4 ライブラリファイルをリンクする方法を示します。

FIT モジュールを読み込んだ時、コンパイラによっては正しくリンク設定ができません。

そのため、ユーザは T4 ライブラリの FIT モジュールを読み込んだ後、手動で T4 ライブラリファイルのリンク設定を行う必要があります。以下にリンク方法を示します。

CC-RX の場合：

スマート・コンフィグレータから T4 ライブラリの FIT モジュールを読み込むと、自動で T4 ライブラリファイルのリンク設定を行います。

GCC の場合：

ユーザは T4 ライブラリの FIT モジュールを読み込んだ後、手動で T4 ライブラリファイルのリンク設定を行う必要があります。

- 「プロジェクト・エクスプローラー」ウィンドウでプロジェクト名を右クリックし、「プロパティ」をクリックします。
- 「プロパティ」画面で左のツリーに「設定」が選択されている状態で、"Linker" -> "Source"をクリックします。"Additional Input files"の右上の「追加」アイコンをクリックします。
- 「ファイル・パスの追加」画面で「ワークスペース」をクリックします。
- 「ファイルの選択」画面で r_t4_rx -> lib -> gcc -> libT4_Library_ether_gcc_rxv1_little.a を選択し、「OK」をクリックします。
- 「ファイル・パスの追加」画面にもどりますので、「OK」をクリックし、「プロパティ」画面にもどします。
- 「プロパティ」画面で左のツリーに「設定」が選択されている状態で、"Linker" -> "Archives"をクリックします。"User defined archive (library) files"に T4 ライブラリ（どのコンパイラ用のライブラリかは問いません）が追加されている場合、右上の「削除」アイコンをクリックし削除してください。なお、スマート・コンフィグレータ V.20.10.0 にてコードの生成を実行するたびに"User defined archive (library) files"へ T4 ライブラリが自動で追加されますので、コードの生成を実行後に削除が必要です。
- 「プロパティ」画面にもどりますので、「OK」をクリックします。

IAR の場合：

ユーザは T4 ライブラリの FIT モジュールを読み込んだ後、手動で T4 ライブラリファイルのリンク設定を行う必要があります。

- 「プロジェクト・エクスプローラー」ウィンドウでプロジェクト名を右クリックし、「プロパティ」をクリックします。
- 「プロパティ」画面で左のカテゴリに「リンカ」が選択されている状態で、"ライブラリ"タブ -> "追加ライブラリ"のテキストボックスに r_t4_rx -> lib -> iar -> T4_Library_ether_iar_rxv1_little.a を追加して「OK」をクリックします。

5.6.4　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。(RX64M 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」ウィンドウが開き、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「RX64M」->「R5F564MLDxFC」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。

5.7 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

5.7.1 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。

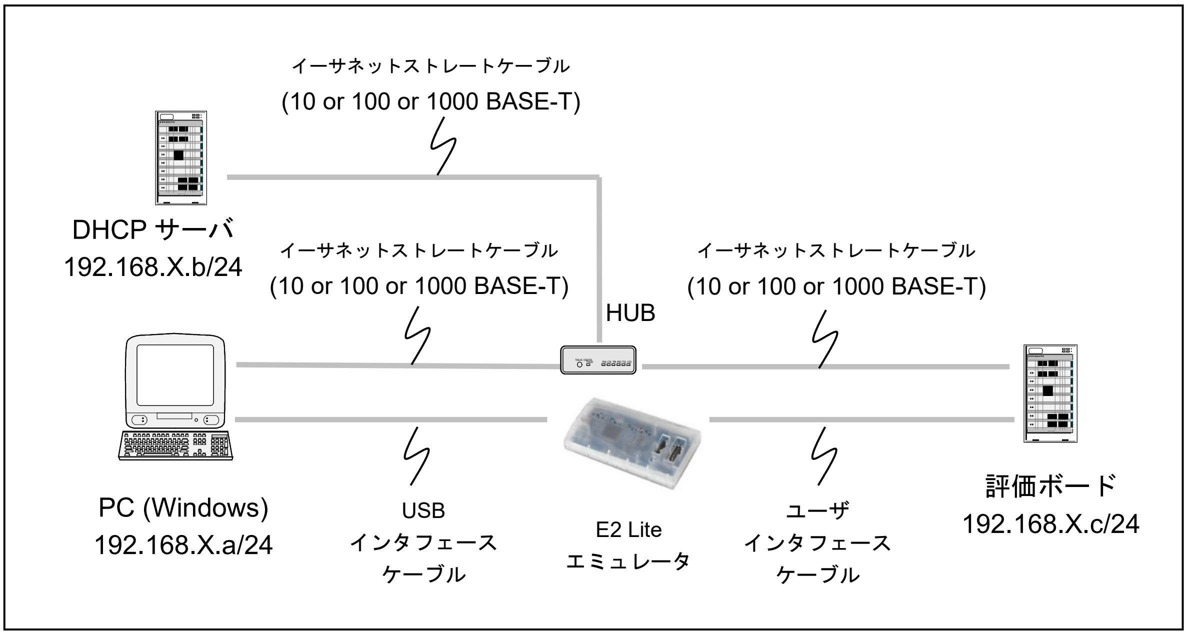


図 5-9 Ethernet サンプルプログラムの実行環境

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力しませんが、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。

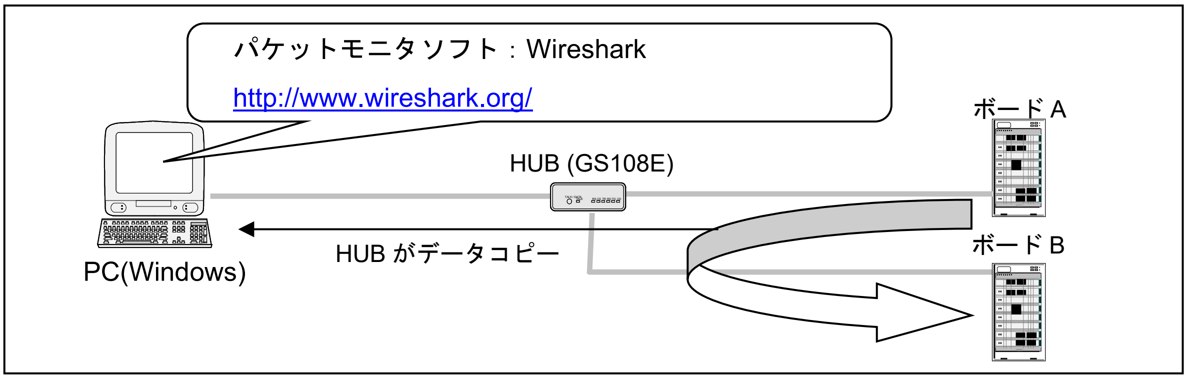


図 5-10 Ethernet サンプルプログラムの動作確認環境

(2)　PC の設定変更

Windows 10 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。

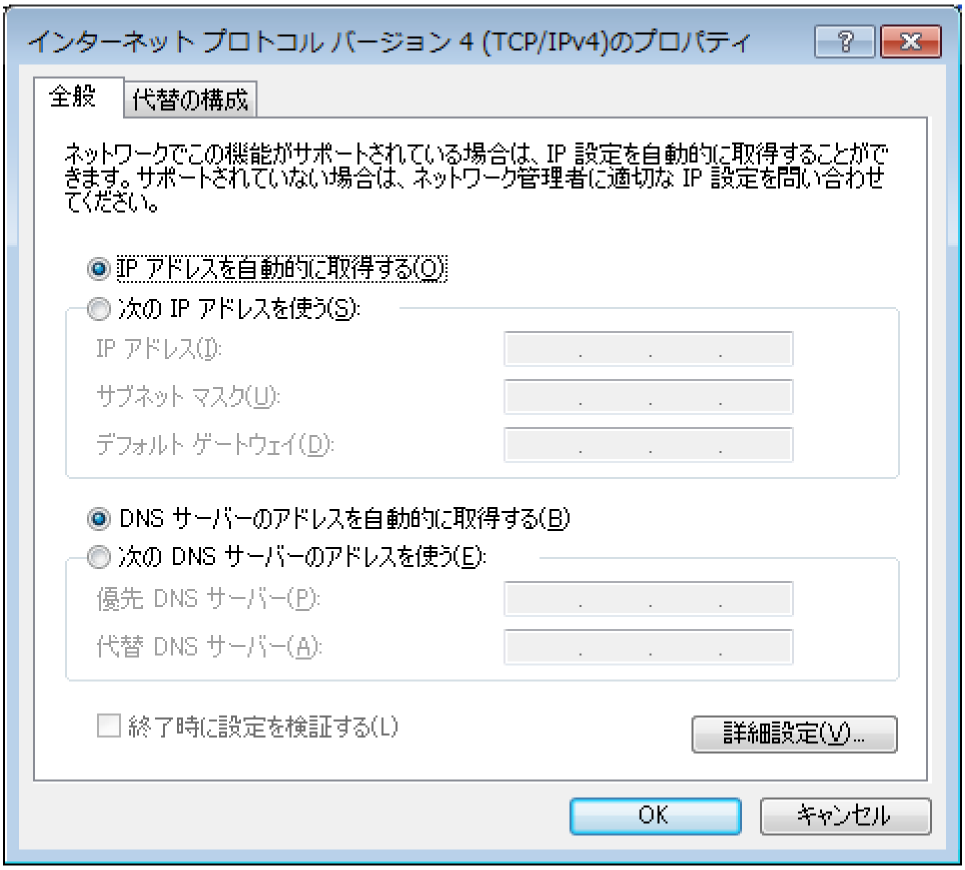


図 5-11 インターネットプロトコルバージョン 4 (TCP/IPv4)のプロパティ

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。

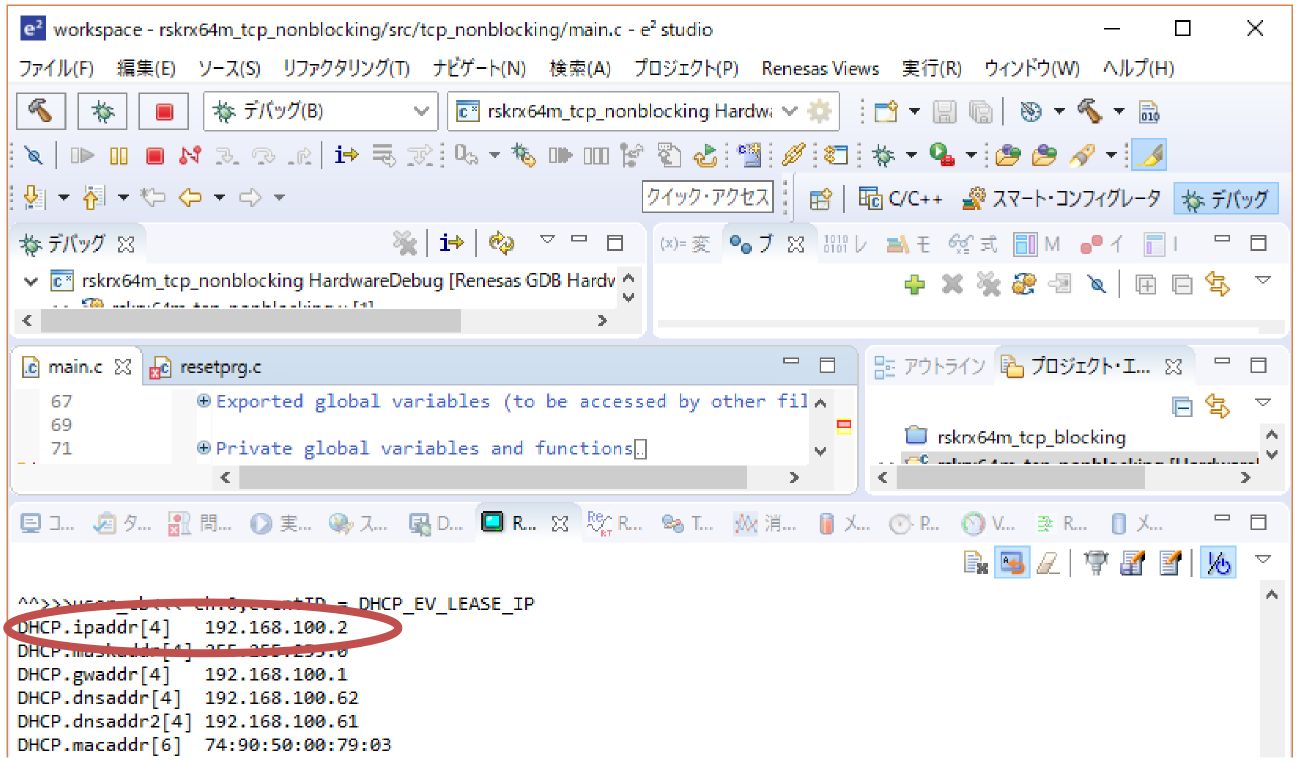


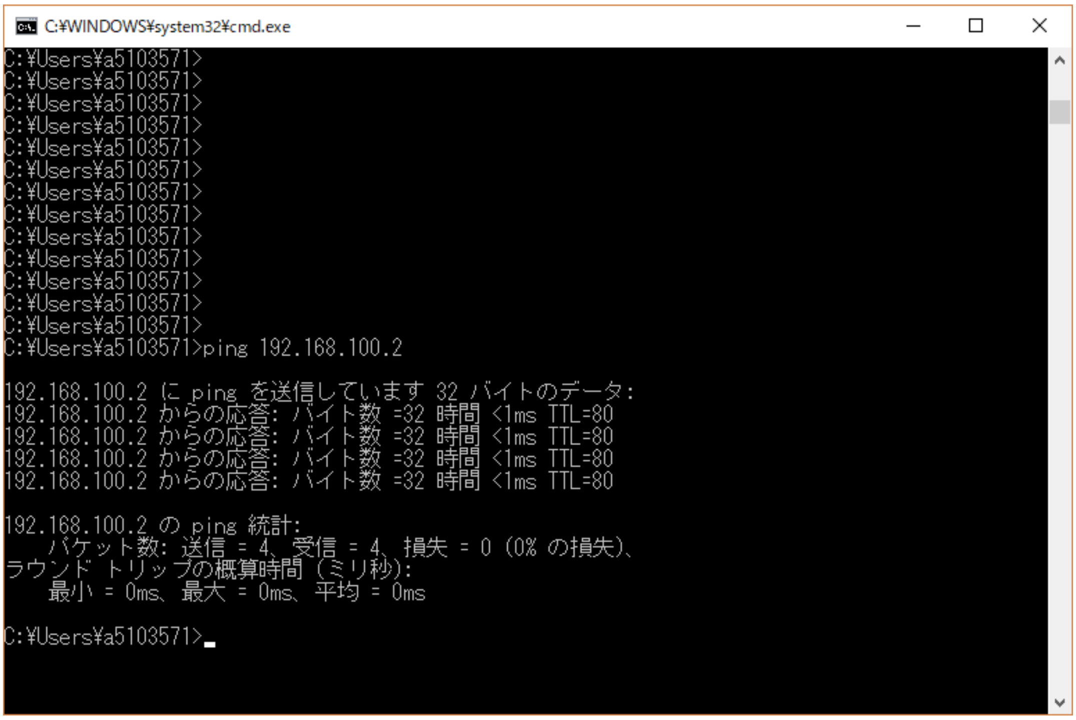
図 5-12 IP アドレスの表示例

図の例は、マイコンに割り当てられた IP アドレスが 192.168.100.2 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:¥WINDOWS¥system32¥cmd.exe
C:¥Users¥a5103571>
C:¥Users¥a5103571>
C:¥Users¥a5103571>
C:¥Users¥a5103571>
C:¥Users¥a5103571>
C:¥Users¥a5103571>
C:¥Users¥a5103571>
C:¥Users¥a5103571>
C:¥Users¥a5103571>
C:¥Users¥a5103571>
C:¥Users¥a5103571>
C:¥Users¥a5103571>
C:¥Users¥a5103571>
C:¥Users¥a5103571>ping 192.168.100.2

192.168.100.2 に ping を送信しています 32 バイトのデータ:
192.168.100.2 からの応答: バイト数 =32 時間 <1ms TTL=80
192.168.100.2 からの応答: バイト数 =32 時間 <1ms TTL=80
192.168.100.2 からの応答: バイト数 =32 時間 <1ms TTL=80
192.168.100.2 からの応答: バイト数 =32 時間 <1ms TTL=80

192.168.100.2 の ping 統計:
    パケット数: 送信 = 4、受信 = 4、損失 = 0 (0% の損失)、
ラウンド トリップの概算時間 (ミリ秒):
    最小 = 0ms、最大 = 0ms、平均 = 0ms

C:¥Users¥a5103571>
```

図 5-13　ping の実行例

5.7.2　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(Windows10 のみ)

・スタート→Windows システム ツール→コントロールパネル→プログラムと機能

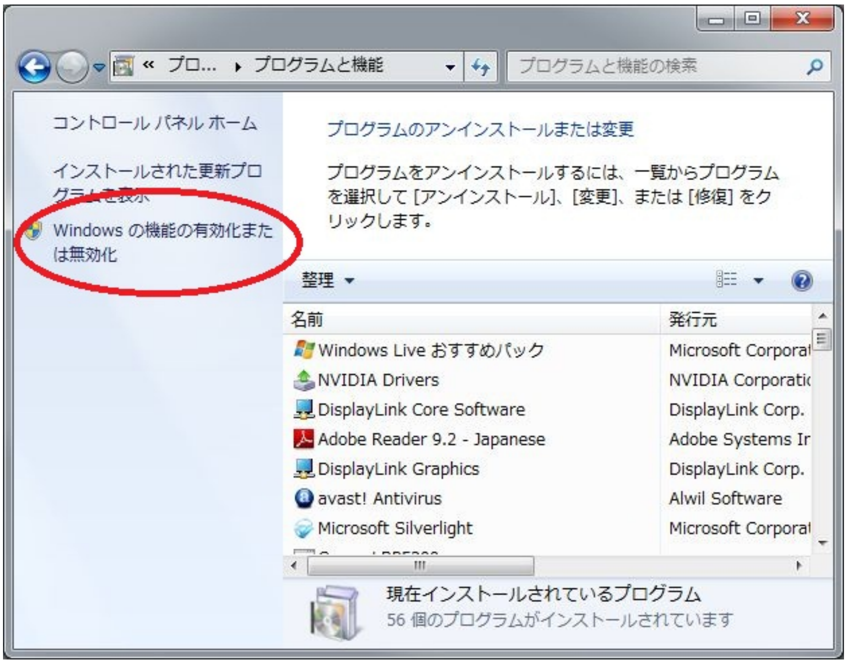


図 5-14　プログラムと機能

・Telnet クライアントにチェックを入れてください。

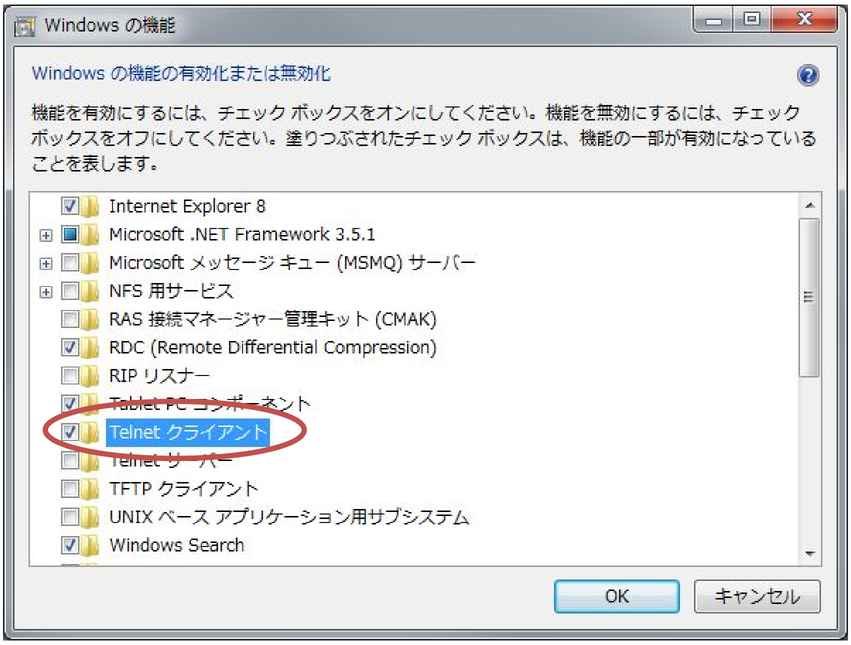


図 5-15　Telnet クライアント

(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

5.7.3 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。

6. 注意事項

6.1 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。

6.2 スマート・コンフィグレータ

T4 ソフトウェアコンポーネント設定画面でユーザは 6 端点(TCP/UDP)をカスタマイズできます。

端点数そのものを変更したい時は r_t4_rx_config.h と config_tcpudp.c を編集してください。

6.3 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。
(3) サンプルプログラムは TSIP 非搭載版 RX65N 用に作成されていますが、TSIP 搭載版 RX65N 用にカスタマイズすることができます。カスタマイズ方法は 2 つあります。
- スマート・コンフィグレータを使用する方法
  a. スマート・コンフィグレータのボード設定画面から TSIP 搭載版 RX65N の
  ボード種別・デバイス型名に変更。
  b. スマート・コンフィグレータのコンポーネント設定画面から TSIP FIT モジュールを追加。
  c. スマート・コンフィグレータからコードの生成を実行。
- 手動で編集する方法 (注)
  a. BSP の r_bsp_config.h 内にあるマクロ定義
  BSP_CFG_MCU_PART_ENCRYPTION_INCLUDED の値を true に変更。
  b. Web ページからダウンロードした TSIP ドライバを e2 studio プロジェクトに手動追加。

注 1：スマート・コンフィグレータでコードの生成を再実行した場合、編集箇所がリセットされる可能性があります。
注 2：対象のマクロ定義については r_t4_driver_rx Rev.1.09 の t4_driver.c の 64～76 行目を参照してください。

7. 付録

7.1 動作確認環境

本 FIT モジュールの動作確認環境を以下に示します。

表 7.1 動作確認環境 (Rev.2.09)

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e^2 studio V7.3.0 ルネサス エレクトロニクス製 CS+ V8.01.00(CC-RX T4 ライブラリ生成環境) IAR Embedded Workbench for Renesas RX 4.11.1
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 4.8.4.201801 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 4.11.1 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのリビジョン	Rev.2.09
ソフトウェアツール (オプション)	QE for TCP/IP V1.0.1
使用ボード	Renesas Starter Kit+ for RX65N-2MB (RTK50565Nxxxxxx) Renesas Starter Kit+ for RX64M (R0K50564Mxxxxxx)

表 7.2 動作確認環境 (Rev.2.10)

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e^2 studio V20.10.0 ルネサス エレクトロニクス製 CS+ V8.04.00(CC-RX T4 ライブラリ生成環境) IAR Embedded Workbench for Renesas RX 4.14.1
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 V3.02.00 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 4.8.4.201803 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 4.14.1 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのリビジョン	Rev.2.10
ソフトウェアツール (オプション)	QE for TCP/IP V1.0.1
使用ボード	Renesas Starter Kit+ for RX65N-2MB (RTK50565Nxxxxxx) Renesas Starter Kit+ for RX64M (R0K50564Mxxxxxx)

7.2 ソフトウェア更新履歴

ソフトウェアバージョン	変更点	リリース日時
V.2.10	セキュリティ強化対応 ・TCP シーケンス番号予測攻撃への対応のため、Initial sequence number (ISN)生成アルゴリズムを変更しました。 制限事項 ・GCC のバージョンに関する制限事項を追加しました。	2021/04/01
V.2.09	対象コンパイラを追加 ・GCC for Renesas RX, IAR C/C++ Compiler for Renesas RX 機能追加 ・TCP Keep-Alive 機能を追加しました。 以下バグ修正 ・ゲートウェイアドレス有効時、一部の宛先 IP アドレスに対して TCP/UDP 通信ができない問題を修正しました。 ・コールバックルーチンから API を呼び出した時、E_QOVR が返ることがある問題を修正しました。 ・DHCP 有効時、IP アドレスが確定する前にマルチキャストアドレス宛またはブロードキャストアドレス宛の UDP パケットを受信すると UDP のコールバックルーチンが呼び出される問題を修正しました。	2019/06/28
V.2.08	以下バグ修正 ・DHCP 無効かつ 3ch 以上使用する設定にすると、ワーク領域内の一部チャネルの IP 情報が壊れる問題を修正。 ・DHCP 有効で複数ボード同時にリスタートすると IP 取得できない問題を修正。 ・UDP 送信完了時、コールバック関数が呼び出されないことがある問題を修正。 ・リピータハブを接続するときに IP アドレスが競合する問題を修正。	2018/12/10
V.2.07	以下バグ修正 ・異常な Ping 応答パケットを送信する問題を修正。 ・UDP 送受信処理キャンセル時の UDP 送受信ができない問題を修正。	2017/12/31
V.2.06 Release 00	機能追加 ・DHCP 機能追加しました。 以下バグ修正 ・TCP で受信ウィンドウにデータが残っている状態で受信 API キャンセルが効かない。 ・同時に複数の SYN を受信したとき一部の SYN/ACK を返さない場合がある。 ・TCP 受信で再送パケットに追加データが含まれる場合に、破棄される。 ・相手からの TCP 受信と T4 からの TCP 再送が重なった時に送信 TCP パケットが不正になる。	2016/12/15
V.2.05 Release 00	機能追加 ・IGMP 機能追加 API の igmp_join_group()と igmp_leave_group()を追加しました。 igmp_join_group()を使用することでマルチキャストグループに参加することができます。 igmp_leave_group()を使用することでマルチキャストグループから離脱することができます。	2015/12/01

V.2.03 Release 00	以下バグ修正 ・T4 と通信相手の FIN パケット送信タイミングが重なった場合に発生する以下 2 点の問題点を修正。 -T4 側からの送信に対する ACK を返さない機器への対策を実施 -クローズ後も受信データを参照しゼロウィンドウ状態を検出してしまい、以降の TCP 接続を受け付けない場合がある問題を修正。	2015/08/07
V.2.02 Release 00	・RX のライブラリ生成環境を CC-RX V1.02 から V.2.03 に変更しました。	2015/01/05
V.2.01 Release 00	・ソースコードをルネサスコーディングルールに適用しました。	2014/07/01
V.2.00 Release 00	機能追加 ・複数 LAN ポート対応し、それぞれの LAN ポートに MAC アドレス、IP アドレスを設定出来るようになりました。 ・ソースコードを公開しました。 ・FIT(Firmware Integration Technology)に対応しました ・北斗電子製 RX63N ボードをサポートしました。 以下バグ修正 ・ITRON V.4 とエラーコードが異なる個所を修正 ・LAN ケーブル断時に tcp_cls_cep()をキャンセルできない問題を修正 ・UDP 送信時に ARP が解決しない場合に UDP 送信処理が完了しない問題を修正 ・tcp_sht_cep()が TFN_TCP_ALL 指定でキャンセルできてしまう問題を修正	2014/04/01
V.1.06 Release 00	機能追加 ・UDP ブロードキャスト受信 (宛先 IP アドレス 255.255.255.255) ・UDP ディレクテッドブロードキャスト受信 (宛先 IP アドレス 192.168.0.0/24 の場合、192.168.0.255) ・UDP ブロードキャスト送信 (宛先 IP アドレス 255.255.255.255) ・UDP ディレクテッドブロードキャスト送信 (宛先 IP アドレス 192.168.0.0/24 の場合、192.168.0.255) 以下バグ修正 ・RI600/4 と併用した場合、r_t4_itcpip.h の型定義と itron.h の型定義とが衝突するのを修正 ・TCP ウィンドウサイズが 0 の SYN パケットを受信した際、通常の SYN+ACK ではなく、SYN フラグのない ACK のみが送信される不具合を修正 ・接続待受け中の tcp_acp_cep()が異常な戻り値を伴いコールバックされる不具合を修正 ・特定の IP アドレス、サブネットマスクを設定すると通信パケットが送信されない不具合を修正 ・PPP 再接続時に PPP サーバから IP アドレス割り当てに失敗する不具合を修正 ・RX210 の PPP サンプルプログラムで、SCI チャネル 1 選択時に通信ができない不具合を修正	2013/06/21
1.05	以下機能追加 ・PPP 用の T4 ライブラリを追加しました。 ・api_wup() を tcp_api_wup()と udp_api_wup()に分割しました	2012/04/01

	・api_slp() を tcp_api_slp()と udp_api_slp()に分割しました 以下性能向上 ・チェックサム演算をアセンブラ化して、通信を高速化しました。 ・Ethernet ドライバの送信割り込みを許可にして、通信を高速化しました。 以下バグ修正 ・UDP チェックサム演算結果がゼロになった場合、受信バッファのチェックサム格納領域を破壊していたのを修正。 ・APR 要求受信後、ARP 応答送信までの間にブロードキャストパケットを受信した場合に不正パケットが送信される不具合を修正。	
1.04	以下機能追加 Ethernet ドライバ関数「report_error」を追加しました。 UDP チェックサムの処理切り替え設定用変数、 「_udp_enable_zerochecksum」を追加しました。 T4 付属のサンプルソフト"t4_driver.c"において、FR フラグを クリアするタイミングを修正して、EDMAC 転送が不正に止まる現象を改善しました。	2011/08/30
1.03	以下バグ修正 (不具合現象) RI600/4 と併用した場合、ユーザ定義関数 api_wup() がどの通信端点による呼び出しかが不明なため、起床するタスク ID も不明になる。 (対策) api_wup()の引数で、処理が終了した通信端点 ID を指定するよう変更。	2011/01/25
1.02	以下バグ修正 (不具合現象) RI600/4 と併用した場合、r_t4_itcpip.h の型定義と itron.h の型定義とが衝突する (対策) r_t4_itcpip.h を修正。	内部バージョン
1.01	以下バグ修正 (不具合現象) T4 が tcp_snd_dat で通信相手に受信ウィンドウ目一杯に送信した後、 相手が受信失敗した等で、受信ウィンドウが十分な状態の ACK 済みの ACK を送信してきた場合、 送信側がゼロウィンドウプローブ、 受信側が ACK 済みの ACK 送信を繰り返し通信不可になる場合がある (対策) T4 が「通信相手はゼロウィンドウ」と判定し、 通信相手が通知してきたウィンドウサイズがゼロでない場合、 ゼロウィンドウプローブではなくデータ再送する	2010/11/10
1.00	新規リリース	2010/09/01

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
2.10	2021.04.01	-	RX ファミリ　組み込み用 TCP/IP M3S-T4-Tiny V.2.10 に合わせてリリース 以下更新しました。 ・要旨 　・ソフトウェア構造に TSIP FIT モジュールを追加 　・PPP 機能に関する内容を削除 　・FIT モジュールではサポートしていない RX62N および RX63N に関する内容を削除 ・5.6 章 　・RX64M サンプルプログラムに関する内容を削除 以下追加しました。 ・1.6 章 　・GCC バージョンに関する制限事項
2.09	2019.06.28	-	RX ファミリ　組み込み用 TCP/IP M3S-T4-Tiny V.2.09 に合わせてリリース 以下更新しました。 1. 概要 2. API 情報 3. T4 ライブラリの情報 7. 付録
2.08	2018.12.10	-	r20uw0031 から次の章を追加。 ・4 章　サンプルプログラム r20an0312 を次の章に統合。 ・4 章　サンプルプログラム 次の章を追加 ・4.6 章　コンポーネントの設定値
2.07	2017.12.31	-	RX ファミリ　組み込み用 TCP/IP M3S-T4-Tiny V.2.07 に合わせてリリース →パッケージバージョン表記を Ver 表記に変更 以下更新しました。 概要 2.3 T4 Ethernet サンプルアプリケーション　ROM/RAM/スタックサイズ 2.4 バージョン情報 4.　注意事項 5.　付録 以下追加しました。 2.5 FIT モジュールの追加方法 5.1 動作確認環境
2.06	2016.12.15	-	RX ファミリ　組み込み用 TCP/IP M3S-T4-Tiny V.2.06 Release 00 に合わせてリリース RX65N に対応しました。 7 章を追加しました。 デバッグ情報付きライブラリを追加しました。 コンパイルオプションを変更しました。

			ライブラリ開発環境を変更しました。
2.05	2015.12.01	-	RX ファミリ 組み込み用 TCP/IP M3S-T4-Tiny V.2.05 Release 00 に合わせてリリース 以下更新しました。 概要 ファイル構成から rxv2 コアライブラリを削除 4. 開発環境のコンパイラバージョン 5. T4 Ethernet サンプルアプリケーション ROM/RAM/スタックサイズ 6. バージョン情報
2.03	2015.08.07	-	RX ファミリ 組み込み用 TCP/IP M3S-T4-Tiny V.2.03 Release 00 に合わせてリリース
2.02	2015.01.05	- p4,p5 p6	RX ファミリ 組み込み用 TCP/IP M3S-T4-Tiny V.2.02 Release 00 に合わせてリリース RX71M に対応しました ライブラリファイル名とコンパイルオプションを変更しました。 ライブラリ開発環境を変更しました。
2.01	2014.07.01	- - p1 p2	RX ファミリ 組み込み用 TCP/IP M3S-T4-Tiny V.2.01 Release 00 に合わせてリリース 本書のタイトルを変更しました FIT モジュールの URL を修正しました。 ソフトウェアの構成例を追加しました。
2.00	2014.04.01		RX ファミリ 組み込み用 TCP/IP M3S-T4-Tiny V.2.00 Release 00 に合わせてリリース 開発環境に北斗電子 RX63N ボード追加 スタックサイズの表を修正 スタックサイズの表を修正 スタックサイズの設定値を修正
1.06	2013.06.21	- p6 p10 p12 p13 p14 p15	RX ファミリ 組み込み用 TCP/IP M3S-T4-Tiny V.1.06 Release 00 に合わせてリリース ・ライブラリ更新履歴をソフトウェア更新履歴に変更 →Ver 表記をパッケージバージョン表記に変更 動作環境に北斗電子製の RX62N ボードを追加しました 動作環境にがじぇっとるねさすの RX63N ボードを追加しました スタックサイズの表を修正 スタックサイズの表を修正 スタックサイズの設定値を修正 Ethernet サンプルドライバ・パッチプログラムの項を追加 サンプルプログラムの動作確認方法を追加
1.05	2012.11.09	 p1 p4	M3S-T4-Tiny for the RX Family V.1.05 Release01 に合わせてリリース 要旨に RX63N の記述を追加 開発環境の[ボード]に RX63N と型名を追加
1.04	2012.09.30	全体	内部評価用としてリリース RX63N 用のサンプルプログラムを追加 RX62N 用のサンプルプログラムを更新 RX62N 用 Ether ドライバをバージョンアップ

			ゼロコピーAPI に対応し、パフォーマンスが向上 LAN ケーブル活線挿抜に対応 Wake on LAN に対応
1.03	2012.04.01	全体 全体 p2 p6	M3S-T4-Tiny for the RX Family V.1.05 Release00 に合わせてリリース T4 の PPP に関する情報を追加 以下誤記修正 略称 HEW の名称を記載。 サンプルプログラムに関する注意事項を 1 点追加。 マルチキャストに関する注意事項を 1 点追加。
1.02	2011.08.30	全体	T4 ライブラリ Ver1.04 に合わせてリリース
1.01	2011.01.25	全体	T4 ライブラリ Ver1.03 に合わせてリリース
1.00	2010.11.10	—	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うと、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。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、お客様の責任において、お客様の機器・システムを設計ください。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みます。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. 当社製品または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. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 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。
6. 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。
7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因しまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 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。
9. 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。
10. 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て、当社は、一切その責任を負いません。
11. 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。
12. 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。
13. 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。
14. 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.5.0-1　2020.10)

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

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。

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/