

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

ルネサスマイクロコンピュータ

本資料に記載の全ての情報は本資料発行時点のものであり、ルネサス エレクトロニクスは、予告なしに、本資料に記載した製品または仕様を変更することがあります。
ルネサス エレクトロニクスのホームページなどにより公開される最新情報をご確認ください。

ご注意書き

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

このマニュアルの使い方

このマニュアルは、組み込み用 TCP/IP M3S-T4-Tiny の Ethernet ドライバインタフェースの仕様を説明します。

1. 本マニュアルの構成

このマニュアルは、大きく分けて次の内容で構成しています。

- 1 章 概要
- 2 章 前提条件
- 3 章 Ethernet ドライバインタフェースの内部構成
- 4 章 変数仕様
- 5 章 関数仕様

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。

目次

1. 概要.....	5
2. 前提条件	6
3. Ethernet ドライバインタフェースの内部構成	7
4. 変数仕様	9
4.1 グローバル変数	9
5. 関数仕様	10
5.1 lan_open.....	11
5.2 lan_close	12
5.3 lan_read	13
5.4 rcv_buff_release	14
5.5 lan_write	15
5.6 lan_reset.....	16
5.7 tcp_api_slp.....	17
5.8 tcp_api_wup.....	18
5.9 udp_api_slp.....	19
5.10 udp_api_wup.....	20
5.11 ena_int	21
5.12 dis_int	22
5.13 tcpudp_act_cyc	23
5.14 tcpudp_get_time	24
5.15 lan_inthdr.....	25
5.16 get_random_number	26
5.17 report_error.....	27
5.18 register_callback_linklayer	30
5.19 lan_check_link.....	31

1. 概要

本仕様書では、Tiny な TCP/IP ライブラリ M3S-T4-Tiny（以下、T4 と呼びます）の Ethernet ドライバインタフェース仕様について説明します。T4 ライブラリでは、Ethernet による通信をサポートしていますが、LAN コントローラの仕様に依存した部分については、ライブラリ本体からドライバ部として分離し、カスタマイズできるようにしました。

本仕様書では、お客様がお使いになる LAN コントローラの仕様に合わせて、Ethernet ドライバを開発される際に必要になる関数の仕様について説明しています。

2. 前提条件

前提条件は以下の通りです。

1. Ethernetの手順で、Ethernet形式のパケットの送受信を行います。
2. 送受信パケットデータは、CRCを除くEthernetのパケットとします。
3. 送信パケットデータは、ヘッダ部とデータ部に分けられて扱われます。ヘッダ部はグローバル変数に格納され、データ部は1 byte整数型(char)の配列に格納されてドライバに渡されます。
4. 受信パケットデータは、1バイトごとに、Ethernetにより転送される順に、ネットワークバイトオーダー（ビッグエンディアン）で受信バッファに格納されます。
5. 受信パケットデータの最大長は、Ethernetでの制限である、1520バイトとします。ただし、ドライバで確保できる受信バッファ長が1520バイトよりも小さい場合は、受信バッファ長が受信パケットデータの最大長となります。
6. 受信バッファはドライバにより管理され、グローバル変数にそのポインタを格納します。バッファ数はユーザが定義します。

3. Ethernet ドライバインタフェースの内部構成

図 3.1に各プロトコル処理間の関係と T4 ライブラリと Ethernet ドライバとの関係を示します。Ethernet ドライバインタフェースは表 3.1に示します。詳細は「5 関数仕様」を参照してください。

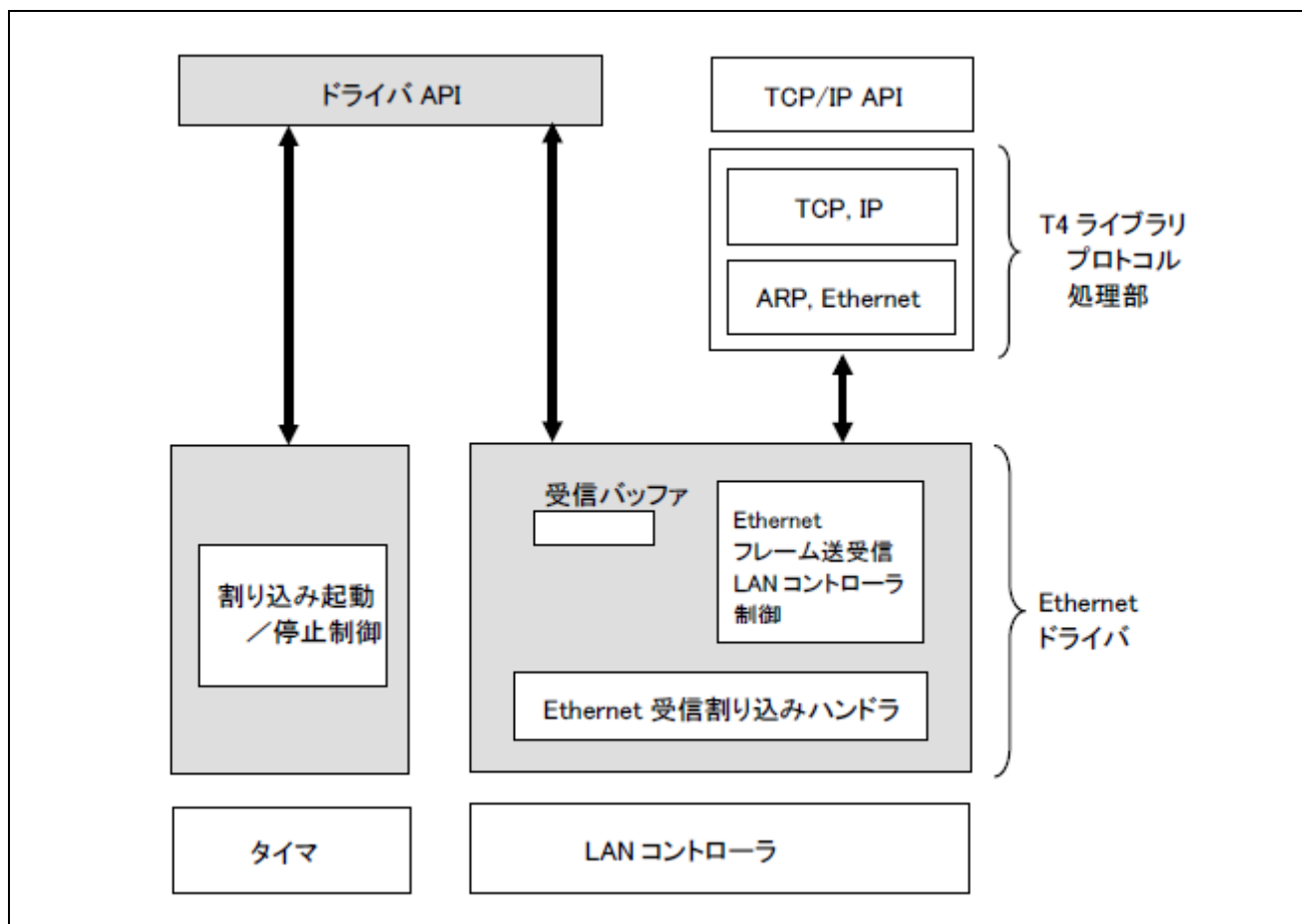


図3.1 Ethernet ドライバインタフェースの構成

Ether ドライバは表 3.1に示す関数をインタフェースとし、初期化、送受信、その他の処理を行います。

表3.1 Ether ドライバインタフェース

関数名	機能
ER lan_open(void)	LAN コントローラの初期化／起動
ER lan_close(void)	LAN コントローラの停止
H lan_read(UB, B**)	データ受信
H rcv_buff_release(UB)	受信バッファ解放
H lan_write(UB, B*, H, B*, H)	データ送信
void lan_reset(UB)	LAN コントローラのリセット
void tcp_api_slp(ID)	TCP API 完了待ち
void tcp_api_wup(ID)	TCP API 完了待ち解除
void udp_api_slp(ID)	UDP API 完了待ち
void udp_api_wup(ID)	UDP API 完了待ち解除
void ena_int(void)	TCP/IP 処理関数の周期起動の再開
void dis_int(void)	TCP/IP 処理関数の周期起動の一時停止
void tcpudp_act_cyc(UB)	TCP/IP 処理関数の周期起動の制御
UH tcpudp_get_time(void)	時刻の取得
void lan_inthdr(void)	割込みハンドラ
void report_error(UB, H, UB*)	エラー通知
void get_random_number(UB *, UW)	乱数取得
void register_callback_linklayer(callback_from_system_t)	Ethernet リンク切断/接続時に呼ばれるコールバック関数の登録
H lan_check_link(UB)	Ethernet のリンク状態の確認

4. 変数仕様

4.1 グローバル変数

イーサネットアドレス

UB _myethaddr[6]

LAN コントローラの MAC アドレスを格納する変数です。T4 のコンフィグレーションファイルでユーザによる設定が可能です。

ユーザにより、すべて 0 が設定された場合、ROM 等から MAC アドレスを読み出して LAN コントローラに設定します。

5. 関数仕様

各 API の詳細は、以下のフォーマットに沿っています。

< API 書式 >

API の書式を示します。

< 機能 >

各 API の機能・動作、使用上の注意点を示します。

< 引数 >

API に与える引数の意味、値の制限を示します。

< 戻り値・エラーコード >

API からの戻り値やエラーコードの種類、発生する条件を示します。

5.1 lan_open

< API 書式 >

ER lan_open (void)

< 機能 >

Ethernet コントローラの初期化を行い、他のドライバ関数が使える状態にします。また、受信バッファを初期化します。

グローバル変数_myethaddr が 0 の場合には、EEPROM に格納されているイーサネットアドレスを Ethernet コントローラに設定し、かつ、_myethaddr にコピーします。

グローバル変数_myethaddr が 0 でない場合には、その値を Ethernet コントローラに設定します。

ユーザによって呼び出されます。

< 引数 >

名称	I/O	型	機能
なし	-	-	-

< 戻り値・エラーコード >

型	機能
0	正常
負の値	エラー(起動失敗)

5.2 lan_close

< API 書式 >

ER lan_close (void)

< 機能 >

Ethernet コントローラの動作を停止します。
ユーザによって呼び出されます。

< 引数 >

名称	I/O	型	機能
なし	-	-	-

< 戻り値・エラーコード >

型	機能
0	正常
-1	エラー

5.3 lan_read

< API 書式 >

H lan_read (UB lan_port_no , B **buf)

< 機能 >

LAN ポート番号が示す受信バッファへのポインタをパラメータが指すポインタ変数に格納し、状態に応じて戻り値を返します。

< 引数 >

名称	I/O	型	機能
lan_port_no	Input	UB	読み出し LAN ポート番号
buf	Output	B**	受信バッファ領域へのポインタの格納先

< 戻り値・エラーコード >

型	機能
正の値	受信したパケットのサイズ
-1	データなし
-2	コントローラ停止中
-5	Ethernet コントローラ異常、または、Ethernet コントローラのリセットが必要
-6	受信パケット CRC エラー

5.4 rcv_buff_release

< API 書式 >

H rcv_buff_release (UB lan_port_no)

< 機能 >

LAN ポート番号が示す現在 T4 で使用中の受信バッファ (関数 lan_read のパラメータで渡した受信バッファ) を解放 (使用できる状態に変更) します。

受信バッファ解放許可 (本関数) が返されるタイミングは不定であることを前提に、受信バッファの管理を行ってください。

< 引数 >

名称	I/O	型	機能
lan_port_no	Input	UB	開放する受信バッファに対応する LAN ポート番号

< 戻り値・エラーコード >

型	機能
0	完了

5.5 lan_write

< API 書式 >

H lan_write (UB lan_port_no, B *header, H header_len, B *data, H data_len)

< 機能 >

パラメータで渡されたヘッダ領域とデータ領域の内容を Ethernet コントローラの送信バッファに 1 つのパケットとして書き込み、パケットを送信します。

< 引数 >

名称	I/O	型	機能
lan_port_no	Input	UB	送信する LAN ポート番号
header	Input	B*	送信するヘッダ領域へのポインタ
header_len	Input	H	送信するヘッダの長さ
data	Input	B*	送信するデータ領域へのポインタ
data_len	Input	H	送信するデータの長さ

< 戻り値・エラーコード >

型	機能
0	送信成功
-5	送信失敗

5.6 lan_reset

< API 書式 >

```
void lan_reset ( UB lan_port_no )
```

< 機能 >

以下の手順で Ethernet コントローラの指定されたチャネルをリセットします。受信バッファ等の変数の初期化はしません。

- (1) Ethernet コントローラの停止(lan_close())を呼び出します)
- (2) Ethernet コントローラのレジスタ類を再設定
- (3) Ethernet コントローラを再スタート

< 引数 >

名称	I/O	型	機能
lan_port_no	Input	UB	リセットする LAN ポート番号

< 戻り値・エラーコード >

型	機能
なし	-

5.7 tcp_api_slp

< API 書式 >

```
void tcp_api_slp ( ID cepid )
```

< 機能 >

T4 では、TCP の各 API 発行後、その API が完了したかどうかを繰り返しチェックしています。

本関数は、そのチェックのたびに呼び出されます。ユーザ定義により、チェックを行う間隔の変更や、API 完了までの間別タスクへの切り替えが可能です。

例えば、 μ ITRON OS を使用する場合には、本関数中で `tslp_tsk()` や `dly_tsk()` を呼び出すことにより別タスクに切り替えて、CPU を有効に利用することができます。

その他、MCU がウェイトモード(割り込みが発生するまで CPU クロックを停止する動作モード)をサポートしている場合、本関数中でウェイトモードに遷移させることにより、消費電力を低減することができます。

本関数を空(処理なし)にした場合、比較的短い間隔で完了チェックを行うようになりますが、機能的には問題ありません。

< 引数 >

名称	I/O	型	機能
cepid	Output	ID	待ち状態に入る API に指定した端点 ID

< 戻り値・エラーコード >

型	機能
なし	-

5.8 tcp_api_wup

< API 書式 >

```
void tcp_api_wup ( ID cepid)
```

< 機能 >

本関数は API が完了した時点で呼び出され、API 完了待ち関数 tcp_api_slp による待ち状態を解除します。例えば μ ITRON OS を使用する場合、関数 tcp_api_slp でシステムコール slp_tsk を呼び出して API が完了するまで待ち、API 完了時には本関数中でシステムコール iwup_tsk を呼び出して待ちを解除することができます。本関数の引数には完了した API で指定した端点 ID が格納されます。これによりユーザはどのタスクの待ち状態を解除するべきかを知ることができます。

関数 tcp_api_slp でウェイトモードに遷移させた場合には、割り込みにより関数 tcp_api_slp による待ちは解除されるため、本関数で必ずしも待ちを解除する必要はありません。このように他の要因により関数 tcp_api_slp の待ちが解除される場合には、本関数は空(処理なし)でも問題ありません。

< 引数 >

名称	I/O	型	機能
cepid	Output	ID	完了した API に指定した端点 ID

< 戻り値・エラーコード >

型	機能
なし	-

5.9 udp_api_slp

< API 書式 >

```
void udp_api_slp ( ID cepid )
```

< 機能 >

T4 では、UDP の各 API 発行後、その API が完了したかどうかを繰り返しチェックしています。
その他説明は tcp_api_slp()関数と同等です。

< 引数 >

名称	I/O	型	機能
cepid	Output	ID	待ち状態に入る API に指定した端点 ID

< 戻り値・エラーコード >

型	機能
なし	-

5.10 udp_api_wup

< API 書式 >

```
void udp_api_wup ( ID cepid)
```

< 機能 >

本関数は UDP の各 API が完了した時点で呼び出され、API 完了待ち関数 `udp_api_slp` による待ち状態を解除します。その他説明は `tcp_api_wup()` 関数と同等です。

< 引数 >

名称	I/O	型	機能
cepid	Output	ID	完了した API に指定した端点 ID

< 戻り値・エラーコード >

型	機能
なし	-

5.11 ena_int

< API 書式 >

```
void ena_int ( void )
```

< 機能 >

本関数は、キャンセル API 内部で呼び出され、キャンセル処理の排他制御に使用します。本関数は、周期起動に用いるタイマ割り込みを有効にします。

< 引数 >

名称	I/O	型	機能
なし	-	-	-

< 戻り値・エラーコード >

型	機能
なし	-

5.12 dis_int

< API 書式 >

```
void dis_int ( void )
```

< 機能 >

本関数は、キャンセル API 内部で呼び出され、キャンセル処理の排他制御に使用します。本関数は、周期起動に用いるタイマ割り込みを無効にします。

< 引数 >

名称	I/O	型	機能
なし	-	-	-

< 戻り値・エラーコード >

型	機能
なし	-

5.13 tcpudp_act_cyc

< API 書式 >

```
void tcpudp_act_cyc ( UB cycact )
```

< 機能 >

パラメータ cycact にしたがって、TCP/IP 処理関数_process_tcpip の周期起動を制御します。関数_process_tcpip の周期起動間隔は、10msec 以下に設定します。

< 引数 >

名称	I/O	型	機能
cycact	Input	UB	TCP/IP 処理関数の周期起動の開始／停止を指定 0 : TCP/IP 処理関数の周期起動を停止する 1 : TCP/IP 処理関数の周期起動を開始する

< 戻り値・エラーコード >

型	機能
なし	-

5.14 tcpudp_get_time

< API 書式 >

UH tcpudp_get_time (void)

< 機能 >

現在時刻を返します。現在時刻の分解能は 10ms で、端数は切り捨てて返します。

現在時刻はシステム起動時を 0 とし、10ms 毎にインクリメントします。

オーバーフローした場合は 0 に戻ります。

< 引数 >

名称	I/O	型	機能
なし	-	-	-

< 戻り値・エラーコード >

型	機能
UH	現在時刻

5.15 lan_inthdr

< API 書式 >

```
void lan_inthdr ( void )
```

< 機能 >

Ethernet コントローラの割込み信号により起動します。

サンプルでは Ethernet 割込みハンドラとして使用しています。

< 引数 >

名称	I/O	型	機能
なし	-	-	-

< 戻り値・エラーコード >

型	機能
なし	-

5.16 get_random_number

< API 書式 >

```
void get_random_number(UB *data, UW len)
```

< 機能 >

指定されたポインタに指定されたデータ長の乱数を設定します。

< 引数 >

名称	I/O	型	機能
data	Output	UB *	乱数を書き込むデータポインタ
len	Input	UW	必要な乱数の byte 長

< 戻り値・エラーコード >

型	機能
なし	-

5.17 report_error

< API 書式 >

```
void report_error ( UB lan_port_no , H error_code, UB *buf )
```

< 機能 >

lan_read()で読み込んだデータにより、T4 ライブラリ内部で発生したエラー情報をエラーコードとして引数に指定し Ethernet ドライバに通知します。また、エラーを検出したパケットが格納された領域の先頭アドレスを引数に指定し Ethernet ドライバに通知します。

ユーザは必要に応じてエラーコードに応じた処理を記述します。

エラーが同時に発生した場合、エラーコードが最もゼロに近いものが出力されます。

< 引数 >

名称	I/O	型	機能
lan_port_no	Output	UB	エラーが発生した LAN ポート番号を指定する
error_code	Output	H	-1 : 受信データ長異常 lan_read()の戻り値が、60byte 未満、または 1514byte より大きい場合
			-2 : ネットワーク層プロトコル異常 lan_read()で受信したデータが、IP パケット、または ARP パケットでない場合
			-3 : トランスポート層プロトコル異常 lan_read()で受信したデータが、IP パケットだったが、TCP パケット、UDP パケット、または ICMP、IGMP パケットでない場合 IP アドレスは T4 のものと一致しており、パケットはトランスポート層まで到達している
			-21 : ARP メッセージ異常① lan_read()で受信したデータが、ARP パケットだったが、ARP メッセージに含まれる IP アドレスが T4 のものと一致しない場合
			-22 : ARP メッセージ異常② lan_read()で受信したデータが、ARP パケットだったが、ARP メッセージに含まれる以下データが異常だった場合 ・ ハードウェアタイプ (0x0001 : Ethernet) ・ 上位プロトコルタイプ (0x0800 : IP) ・ ハードウェアアドレス長 (0x06 : MAC アドレス) ・ プロトコルアドレス長 (0x04 : IP アドレス)
			-41 : IP ヘッダ異常① lan_read()で受信したデータが、IP パケットだったが、IP ヘッダに含まれる送信先 IP アドレスが T4 のものと一致しない場合
			-42 : IP ヘッダ異常② lan_read()で受信したデータが、IP パケットだったが、IP ヘッダに含まれる送信元 IP アドレスがマルチキャスト、またはブロードキャストの場合
			-43 : IP ヘッダ異常③ lan_read()で受信したデータが、IP パケットだったが、IP ヘッダに含まれる送信元 IP アドレスがループバックアドレスの場合

			-44 : IP ヘッダ異常④ lan_read()で受信したデータが、IP パケットだったが、IP ヘッダに含まれる IP バージョンが「4」以外の場合
			-45 : IP ヘッダ異常⑤ lan_read()で受信したデータが、IP パケットだったが、IP ヘッダにオプションが含まれる場合
			-46 : IP ヘッダ異常⑥ lan_read()で受信したデータが、IP パケットだったが、IP ヘッダに含まれるチェックサム値が異常な場合
			-47 : IP ヘッダ異常⑦ lan_read()で受信したデータが、IP パケットだったが、IP ヘッダに含まれる IP データ長が lan_read()の戻り値(受信データ長)よりも大きい、または IP ヘッダ最小長(20byte)に満たない場合
			-48 : IP ヘッダ異常⑧ lan_read()で受信したデータが、IP パケットだったが、IP ヘッダに含まれる送信元 IP アドレスがネットワークアドレス(例 : xxx.xxx.xxx.0/24)、またはネットワークブロードキャストアドレスの場合(例 : xxx.xxx.xxx.255/24)
			-49 : IP ヘッダ異常⑨ lan_read()で受信したデータが、IP パケットだったが、IP ヘッダに含まれる IP フラグメントフラグが ON になっている場合
			-61 : TCP ヘッダ異常① lan_read()で受信したデータが、TCP/IP パケットだったが、TCP ヘッダに含まれるポート番号が、接続中、または接続待ちのどの端点のものでもない場合 また、すべての TCP 通信端点が接続中でさらに TCP 接続要求があった場合 例1.通信相手が指定した TCP ポート番号が 80 番、T4 が tcp_acp_cep()で待受けている TCP ポート番号が 20 番のみの場合、本エラーコードが返る 例2.通信相手が指定した TCP ポート番号が 80 番、T4 が tcp_acp_cep()を呼び出していない状態の場合、本エラーコードが返る 例3. T4 が使用している TCP 通信端点が 5 個あり、その 5 個全てが接続中で、さらに TCP 接続要求があった場合、本エラーコードが返る
			-62 : TCP ヘッダ異常② lan_read()で受信したデータが、TCP/IP パケットだったが、TCP ヘッダに含まれるチェックサム値が異常な場合
			-81 : UDP ヘッダ異常① lan_read()で受信したデータが、UDP/IP パケットだったが、UDP ヘッダに含まれるチェックサム値が異常な場合
			-82 : UDP ヘッダ異常② lan_read()で受信したデータが、UDP/IP パケットだったが、UDP ヘッダに含まれるチェックサム値がゼロ、かつ、udp_enable_zerochecksum 変数が 0 以外に設定されている場合
			-83 : UDP ヘッダ異常③ lan_read()で受信したデータが、UDP/IP パケットだったが、UDP ヘッダに含まれるポート番号がどの端点のものでもない場合

			-101 : ICMP ヘッダ異常① lan_read()で受信したデータが、ICMP/IP パケットだったが、ICMP ヘッダに含まれるタイプがエコー要求(0x08)以外の場合
			-121 : IGMP ヘッダ異常① lan_read()で受信したデータが、IGMP/IP パケットだったが、IGMP ヘッダのチェックサムが異常だった場合
			-122 : IGMP ヘッダ異常② lan_read()で受信したデータが、IGMP/IP パケットだったが、IGMPtype が IGMPv1Report(0x12)、IGMPv2Report(0x16)以外だった場合
			-131 : DHCP 処理異常
			-132 : DHCP 送信タイムアウト
buf	Output	UB *	エラーを検出したパケットが格納された領域の先頭アドレス

< 戻り値・エラーコード >

型	機能
なし	-

5.18 register_callback_linklayer

< API 書式 >

```
void register_callback_linklayer ( callback_from_system_t call_fp )
```

< 機能 >

本関数はリンクレイヤーからのイベントを T4 に通知するコールバック関数を登録する関数です。

callback_from_system_t 型の関数ポインタを登録し、通知を受けます。コールバック関数は常に 1 つしか登録できません。複数回発行した場合、常に上書き登録となります。

コールバック関数は下記の通り。

```
typedef ER(*callback_from_system_t)(UB channel, UW eventid, VP param);
```

関数ポインタ未登録の場合、状態通知は行われません。引数 channel は LAN ポート番号を表し、eventid は通知の種類を表します。param はパラメータをユーザに渡すためのポインタ(現在未使用のためゼロ)です。

ETHER_EV_LINK_OFF : Ethernet リンク切断しました。

ETHER_EV_LINK_ON : Ethernet リンク接続しました。

ETHER_EV_COLLISION_IP : IP アドレス衝突を検出しました。

< 引数 >

名称	I/O	型	機能
call_fp	Input	callback_from_system_t	コールバック関数登録

< 戻り値・エラーコード >

型	機能
なし	-

5.19 lan_check_link

< API 書式 >

H lan_check_link (UB lan_port_no)

< 機能 >

Ethernet のリンク状態を確認します。

< 引数 >

名称	I/O	型	機能
lan_port_no	Input	UB	LAN ポート番号

< 戻り値・エラーコード >

型	機能
0	リンクオフ
1	リンクオン

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2010.10.07	—	初版発行
1.01	2011.01.06	8	api_wup()のプロトタイプ宣言を変更しました。
1.02	2011.08.23	8-9	report_error()関数を追加しました。
1.03	2012.04.01	-	以下仕様変更しました。 api_slp(void) -> tcp_api_slp(ID), udp_api_slp(ID) api_wup(ID) -> tcp_api_wup(ID), udp_api_slp(ID) 以下誤植修正しました。 lan_read の引数説明 bif -> buf rev1.02 の改定記録での関数名誤記 error_report() -> report_error()
1.04	2013.06.21	6-16	ドキュメントのテンプレートを更新しました
1.05	2014.04.01	改訂記録	表 1. ドライバインタフェース一覧を更新しました 以下仕様変更しました。 lan_read(B**) -> lan_read(UB, B**) rcv_buff_release(void) -> rcv_buff_release(UB) lan_write(B*, H, B*, H) -> lan_write(UB, B*, H, B*, H) lan_reset(void) -> lan_reset(UB)
1.06	2015.08.07	3,16	ドキュメントのテンプレートを更新しました
1.07	2015.12.01	-	各関数の引数に、I/O 情報を追記しました。 get_random_number()を追記しました。
1.08	2016.11.30	8,30,31	ドキュメントのテンプレートを更新しました。 L2LinkOff/On 時にコールバック通知される関数登録関数 register_callback_linklayer を追記しました。 lan_check_link を追記しました。



ルネサスエレクトロニクス株式会社

営業お問合せ窓口

<http://www.renesas.com>

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

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

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

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