

M3S-T2-Tiny V.1.01

M3S-T2-30 V.1.01

M3S-T2-308 V.1.01

R8C/Tiny, M16C/Tiny, H8/Tiny シリーズ用 TCP/IP ライブラリ

M16C/62 グループ用 TCP/IP ライブラリ

M32C/83 グループ用 TCP/IP ライブラリ

Ethernet ドライバ・インタフェース 仕様書

- Microsoft、MS-DOS、Windows および Windows NT は、米国 Microsoft Corporation の米国およびその他の国における登録商標です。
- HP-UX は、米国 Hewlett-Packard Company のオペレーティングシステムの名称です。
- Sun 、 Java およびすべての Java 関連の商標およびロゴは、米国およびその他の国における米国 Sun Microsystems, Inc.の商標または登録商標です。
- UNIX は、X/Open Company Limited が独占的にライセンスしている米国ならびに他の国における登録商標です。
- IBM および AT は、米国 International Business Machines Corporation の登録商標です。
- HP 9000 は、米国 Hewlett-Packard Company の商品名称です。
- SPARC および SPARCstation は、米国 SPARC International, Inc.の登録商標です。
- Intel, Pentium は、米国 Intel Corporation の登録商標です。
- Adobe および Acrobat は、Adobe Systems Incorporated (アドビシステムズ社) の登録商標です。
- Netscape および Netscape Navigator は、米国およびその他の諸国の Netscape Communications Corporation 社の登録商標です。
- その他すべてのブランド名および製品名は個々の所有者の登録商標もしくは商標です。

安全設計に関するお願い

- 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご留意ください。

本資料ご利用に際しての留意事項

- 本資料は、お客様が用途に応じた適切なルネサス テクノロジ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株式会社ルネサス テクノロジおよび株式会社ルネサス ソリューションズが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
- 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の使用に起因する損害、第三者所有の権利に対する侵害に関し、株式会社ルネサス テクノロジおよび株式会社ルネサス ソリューションズは責任を負いません。
- 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全ての情報は本資料発行時点のものであり、株式会社ルネサス テクノロジおよび株式会社ルネサス ソリューションズは、予告なしに、本資料に記載した製品又は仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりましては、事前に株式会社ルネサス テクノロジ、株式会社ルネサス ソリューションズ、株式会社ルネサス販売又は特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ (<http://www.renesas.com>) などを通じて公開される情報に常にご注意ください。
- 本資料に記載した情報は、正確を期すため、慎重に制作したものです。万一本資料の記述誤りに起因する損害がお客様に生じた場合には、株式会社ルネサス テクノロジおよび株式会社ルネサス ソリューションズはその責任を負いません。
- 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。株式会社ルネサス テクノロジおよび株式会社ルネサス ソリューションズは、適用可否に対する責任を負いません。
- 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、株式会社ルネサス テクノロジ、株式会社ルネサス ソリューションズ、株式会社ルネサス販売又は特約店へご照会ください。
- 本資料の転載、複製については、文書による株式会社ルネサス テクノロジおよび株式会社ルネサス ソリューションズの事前の承諾が必要です。
- 本資料に関し詳細についてのお問い合わせ、その他お気付きの点がございましたら株式会社ルネサス テクノロジ、株式会社ルネサス ソリューションズ、株式会社ルネサス販売又は特約店までご照会ください。

[目次]

1.	概要	3
2.	前提条件	3
3.	Ethernet ドライバの内部構成	4
4.	関数仕様	5
4.1	グローバル変数	5
4.2	各関数機能詳細	6
	lan_open Ethernet ドライバの初期化	6
	lan_close Ethernet ドライバの機能停止	7
	lan_read データを受信	8
	rcv_buff_release 受信バッファ解放許可	9
	lan_write データを送信	10
	lan_reset Ethernet コントローラのリセット	11
	api_slp API 完了待ち	12
	api_wup API 完了待ちの解除	13
	tcpudp_act_cyc TCP/IP 処理関数の周期起動の制御	14
	tcpudp_get_time 時刻の取得	15
5.	付録	16
5.1	ドライバ内で使用するグローバル変数	16
5.2	割込みハンドラ	17
	lan_inthdr 割込みハンドラ	17

1. 概要

本仕様書では、Tiny な TCP/IP ライブラリ M3S-T2-xxx（以下、T2 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。T2 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

- Ethernet インタフェースは、1 つのみ対象とします（ループバックを除く）。
- Ethernet の手順で、Ethernet 形式のパケットの送受信を行います。
- 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。
- 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 バイト整数型(char)の配列に格納されてドライバに渡されます。
- 受信パケットデータは、1 オクテットごとに、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。
- 受信パケットデータの最大長は、Ethernet での制限である、1520 オクテットとします。ただし、ドライバで確保できる受信バッファ長が 1520 オクテッ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。
- 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。
- 受信バッファは、2 バイト境界に合わせて配置するものとします。

3. Ethernet ドライバの内部構成

T2 ライブラリに含まれるプロトコル処理部とドライバとの関係を図 1 に、ドライバ・インタフェース（詳細は4.2節参照）を表 1 に示します。

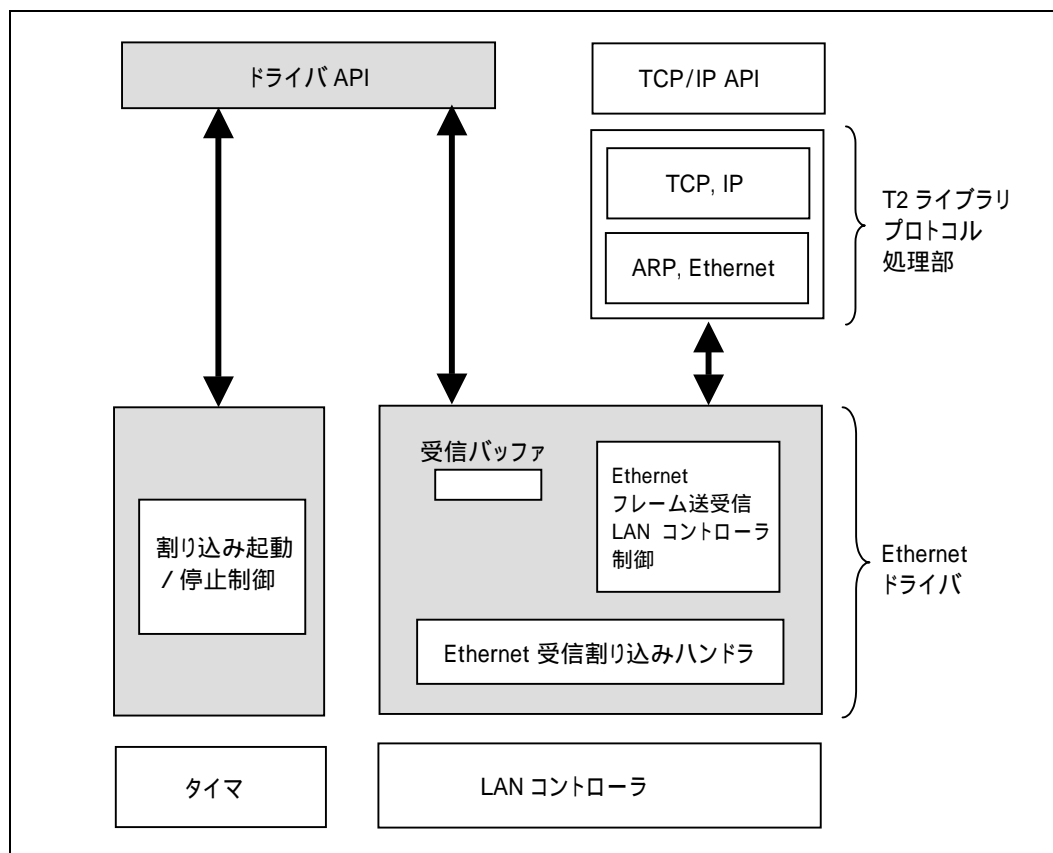


図 1 Ethernet ドライバの構成図

表 1 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(B**)	データ受信
H rcv_buff_release(void)	受信バッファ解放
H lan_write(B*, H, B*, H)	データ送信
void lan_reset(void)	LAN コントローラのリセット
void api_slp(void)	API 完了待ち
void api_wup(void)	API 完了待ち解除
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)	割り込みハンドラ

ドライバは表 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。

4. 関数仕様

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。T2 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。ユーザにより全て 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。

4.2 各関数機能詳細

lan_open	Ethernet ドライバの初期化
----------	-------------------

【形式】

ER lan_open (void)

【パラメータ】

なし

【機能説明】

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。

【戻り値】

0	正常
負数	エラー(起動できなかった)

【呼び出し箇所】

ユーザによる呼び出し

lan_close

Ethernet ドライバの機能停止

【形式】

ER lan_close (void)

【パラメータ】

なし

【機能説明】

Ethernet コントローラの動作を停止します。

【戻り値】

0	正常
-1	エラー

【呼び出し箇所】

ユーザによる呼び出し

lan_read**データを受信****【形式】**

H lan_read (B **buf)

【パラメータ】

B **buf 受信バッファ領域へのポインタの格納先

【機能説明】

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。

【戻り値】

0 以上	受信したパケットの大きさ
-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

【呼び出し箇所】

プロトコル処理部で使用

lan_write**データを送信****【形式】**

H lan_write (B *hdr , H hlen, B *buf, H dlen)

【パラメータ】

B *hdr	送信するヘッダ領域へのポインタ
H hlen	送信するヘッダの長さ
B *buf	送信するデータ領域へのポインタ
H dlen	送信するデータの長さ

【機能説明】

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。

【戻り値】

0	送信成功
-5	送信失敗

【呼び出し箇所】

プロトコル処理部で使用

lan_reset

Ethernet コントローラのリセット

【形式】

void lan_reset (void)

【パラメータ】

なし

【機能説明】

以下の手順で Ethernet コントローラのリセットを行います。受信バッファ等の変数の初期化はしません。

Ethernet コントローラの停止 (lan_close()使用)

Ethernet コントローラのレジスタ類を再設定

Ethernet コントローラを再スタート

【戻り値】

なし

【呼び出し箇所】

プロトコル処理部で使用

api_slp**API 完了待ち****【形式】**

void api_slp (void)

【パラメータ】

なし

【機能説明】

T2 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、μITRON 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。

【戻り値】

なし

【呼び出し箇所】

プロトコル処理部で使用

api_wup

API 完了待ちの解除

【形式】

void api_wup (void)

【パラメータ】

なし

【機能説明】

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 `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 `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 `wup_tsk` を呼び出して待ちを解除することができます。

関数 `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 `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 `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。

【戻り値】

なし

【呼び出し箇所】

プロトコル処理部で使用

tcpudp_act_cyc

TCP/IP 処理関数の周期起動の制御

【形式】

void tcpudp_act_cyc (UB cycact)

【パラメータ】

UB cycact	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

【機能説明】

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。

【戻り値】

なし

【呼び出し箇所】

プロトコル処理部で使用

【注意事項】

- ・関数_process_tcpip の周期起動間隔は 10msec 以下に設定して下さい。
- ・本関数は T2 ライブラリ・オープン関数 tcpudp_open、T2 ライブラリ・クローズ関数 tcpudp_close から呼び出されます。

tcpudp_get_time

時刻の取得

【形式】

UH tcpudp_get_time (void)

【パラメータ】

なし

【機能説明】

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。

【戻り値】

現在時刻

【呼び出し箇所】

プロトコル処理部で使用

5. 付録

この章では、ユーザが定義する変数や関数・ハンドラについて、サンプルプログラムでの例を説明します。サンプルですので、この通りに作成する必要はありません。要否も含めてユーザが決定・定義してください。

5.1 ドライバ内で使用するグローバル変数

受信バッファ構造体

```
typedef struct {
    UB  stat;      // データを受信したときの LAN コントローラの状態
    UB  page;      // LAN コントローラのリングバッファ中の位置
    UH  size;      // 受信パケット長
    UB  data[n];   // 受信バッファ本体
} ETHER_RCV_BUF;
```

受信バッファ本体を含む構造体です。サンプルでは受信バッファ数は 1 にしており、構造体の中に受信バッファが含まれた形になっています。

サンプルでは、関数 `lan_read` で、Ethernet コントローラのバッファ（リングバッファ）からこの受信バッファへのデータ転送を行い、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ています。

5.2 割込みハンドラ

lan_inthdr	割込みハンドラ
-------------------	----------------

【形式】

void lan_inthdr (void)

【パラメータ】

なし

【機能説明】

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、割り込み要因レジスタのクリアや Ethernet コントローラのステータスチェックの他には、特に処理は行いません。

【戻り値】

なし

【呼び出し箇所】

サンプルでは INT1 割込みハンドラとして使用

Ethernet ドライバ・インタフェース仕様書

Rev. 1.01
04.07.01
RJJ10J0630-0101Z

COPYRIGHT ©2004 RENESAS TECHNOLOGY CORPORATION
AND RENESAS SOLUTIONS CORPORATION ALL RIGHTS RESERVED



株式会社ルネサス テクノロジ

東京都千代田区大手町2-6-2 日本ビル 〒100-0004