

# SH7670グループ

## イーサネット送信設定例

R01AN0302JJ0101  
Rev. 1.01  
2010.10.15

### 要旨

本アプリケーションノートでは、SH7670/SH7671/SH7672/SH7673 のイーサネット送信設定例を説明しています。

### 動作確認デバイス

SH7670

### 目次

1. はじめに.....	2
2. 応用例の説明.....	3
3. 参考プログラムリスト.....	18
4. 参考ドキュメント.....	39

## 1. はじめに

### 1.1 仕様

- 本応用例ではイーサネットフレームを 10 フレーム送信します。1 フレームの送信が完了してから次の送信を開始します。
- フレーム送信完了はフレーム送信完了割り込みにより判断します。

### 1.2 使用機能

- イーサネットコントローラ(EtherC)
- イーサネットコントローラ用ダイレクトメモリアクセスコントローラ(E-DMAC)
- 割り込みコントローラ(INTC)
- I<sup>2</sup>C バスインタフェース 3(IIC3)
- ピンファンクションコントローラ(PFC)

### 1.3 適用条件

マイコン	SH7670/SH7671/SH7672/SH7673 (R5S76700/R5S76710/R5S76720/R5S76730)
動作周波数	内部クロック : 200 MHz バスクロック : 66.67 MHz 周辺クロック : 33.33MHz
統合開発環境	ルネサスエレクトロニクス製 High-performance Embedded Workshop Ver.4.03.00
C コンパイラ	ルネサスエレクトロニクス製 SuperH RISC engine ファミリ C/C++コンパイラパッケージ Ver.9.01 Release01
コンパイルオプション	High-performance Embedded Workshop でのデフォルト設定 (-cpu=sh2afpu -fpu=single -object="\$(CONFIGDIR)¥\$(FILELEAF).obj" -debug -gbr=auto -chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 -nologo)

### 1.4 関連アプリケーションノート

本アプリケーションノートに関連するアプリケーションノートを以下に示します。合わせて参照してください。

- SH7670 グループ 初期設定例
- SH7670 グループ イーサネット PHY-LSI 自動交渉設定例
- SH7670 グループ イーサネット受信設定例

## 2. 応用例の説明

本応用例では、イーサネットコントローラ(EtherC)、およびイーサネットコントローラ用ダイレクトメモリアクセスコントローラ(E-DMAC)を使用します。

### 2.1 使用機能の動作概要

本 LSI では、イーサネット通信を行う場合必ず EtherC と E-DMAC を使用します。EtherC は送受信制御を行います。E-DMAC はその送信/受信 FIFO とユーザが指定するデータ格納先(バッファ)間の DMA 転送を専用に行います。

#### 2.1.1 EtherC の概要

本 LSI は、イーサネットあるいは IEEE802.3 の MAC(Media Access Control)層規格に準拠したイーサネットコントローラ(EtherC)を内蔵しています。EtherC は、同規格に準拠した物理層 LSI(PHY-LSI)と接続することにより、イーサネット/IEEE802.3 フレームの送受信を行うことができます。本 LSI 内蔵の EtherC は MAC 層インタフェースを 1 系統内蔵しています。また EtherC は、本 LSI 内部で E-DMAC に接続されており、メモリとの高速アクセスが可能です。

図 1 に EtherC の構成を示します。

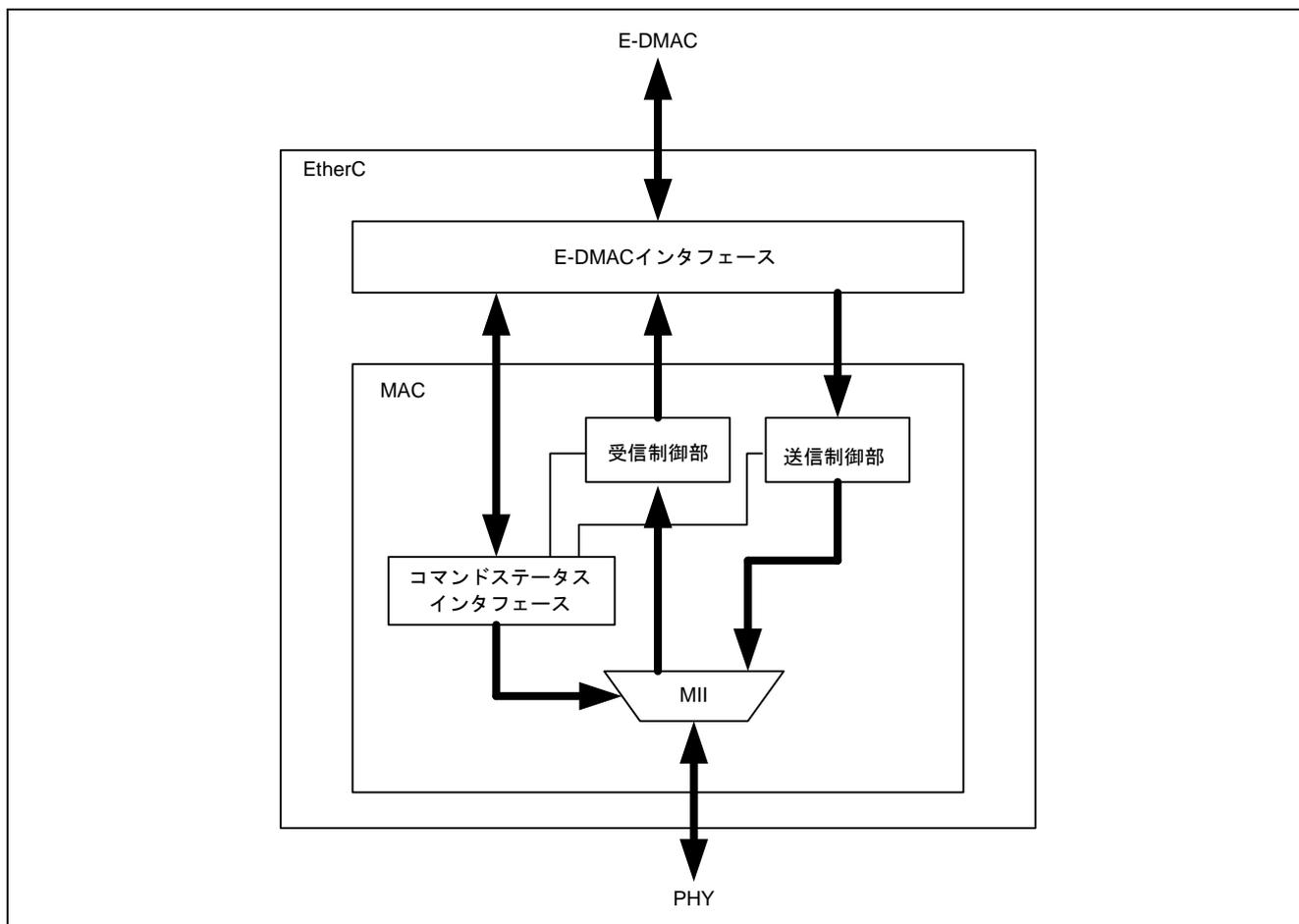


図1 EtherC の構成

2.1.2 EtherC 送信部の概要

EtherC 送信部は、E-DMAC 送信部から送信要求があると送信データをフレームに組み立てて MII(Media Independent Interface)に出力します。MII を経由した送信データは、PHY-LSI によって回線上に送出されます。図 2に EtherC 送信部の状態遷移図を示します。送信動作のフローは以下のようになります。

1. EtherC は、送信許可ビット(EtherC モードレジスタ(ECMR)の TE ビット)がセットされると送信アイドル状態に遷移します。
2. (A)半二重転送方式(HDPX)時  
E-DMAC 送信部から送信要求があると EtherC はキャリア検出を行い、未検出であればフレーム間隔時間の送信延期を経てプリアンプルを MII に送出します。キャリアを検出した場合は、キャリアがなくなつてからフレーム間隔時間の送信延期を経てプリアンプルを MII に送出します。  
(B)全二重転送方式(FDPX)時  
キャリア検出を必要とせず、E-DMAC 送信部から送信要求があると即座にプリアンプルを送出します。ただし連続送信時は、直前に送信したフレームから必ずフレーム間隔時間の送信延期を経てプリアンプルを送出します。
3. SFD(Start Frame Delimiter)、データ、CRC(Cyclic Redundancy Check)を順次送信します。送信を終了するとフレーム送信完了割り込み(TC)が発生します。データ送信中に衝突あるいはキャリア未検出状態になるとそれぞれの割り込みが発生します。
4. アイドル状態に遷移し、以後送信データがあれば送信を継続します。

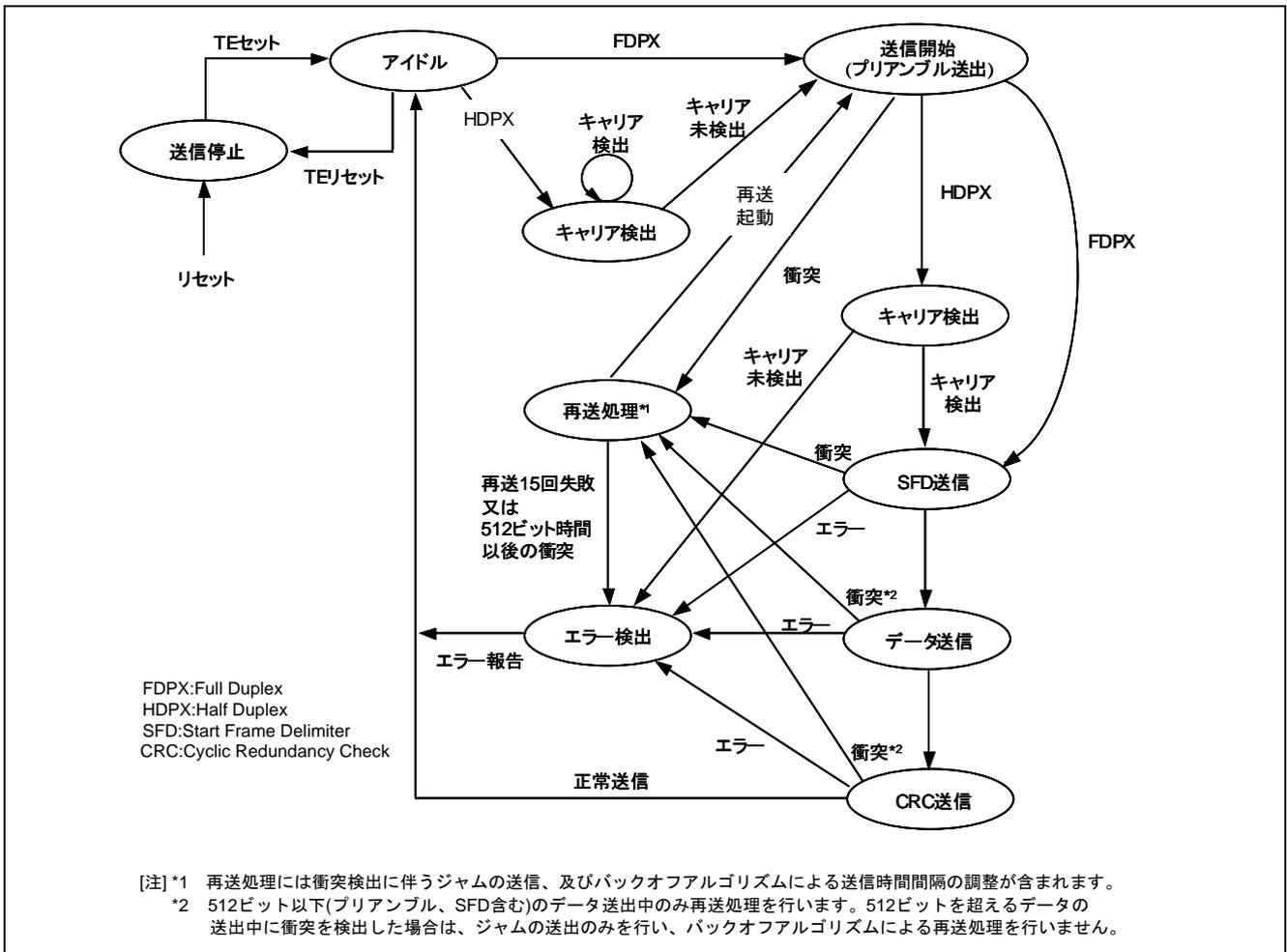


図2 EtherC 送信部状態遷移図

### 2.1.3 E-DMAC の概要

本 LSI は、EtherC に直結したダイレクトメモリアクセスコントローラ(E-DMAC)を内蔵しています。E-DMAC は、E-DMAC 内蔵の DMAC を使用し、E-DMAC 内の送信/受信 FIFO とユーザが指定するデータ格納先(送信/受信バッファ)との間で送受信データの DMA 転送を行います。CPU により直接送信/受信 FIFO のデータを読み書きすることはできません。この DMA 転送時に、E-DMAC が参照する情報を送信/受信ディスクリプタ(次章で詳述)と呼び、ユーザがメモリ上に配置します。E-DMAC は、イーサネットフレーム送受信に先立ちディスクリプタの情報を読み込み、その内容にしたがって送信データを送信バッファから読み込み、または受信データを受信バッファへ書き込みます。このディスクリプタを複数個並べ、ディスクリプタ列(リスト)とすることで、複数のイーサネットフレームの送受信を連続的に行うことができます。

この E-DMAC の機能によって CPU の負荷を軽減し、効率の良いデータ送受信制御を行うことができます。

図 3 に E-DMAC とディスクリプタおよびバッファの構成を示します。

E-DMAC の特長は以下のようになります。

特長

- ・送信/受信 2 系統の独立した DMAC 内蔵
- ・ディスクリプタ管理方式による CPU 負荷の軽減
- ・送受信フレームステータスのディスクリプタへの反映
- ・DMA ブロック転送(16 バイト単位)によるシステムバスの効率使用
- ・1 フレーム/1 ディスクリプタ、1 フレーム/複数フレーム(マルチバッファ)方式対応可能(2.1.5 参照)

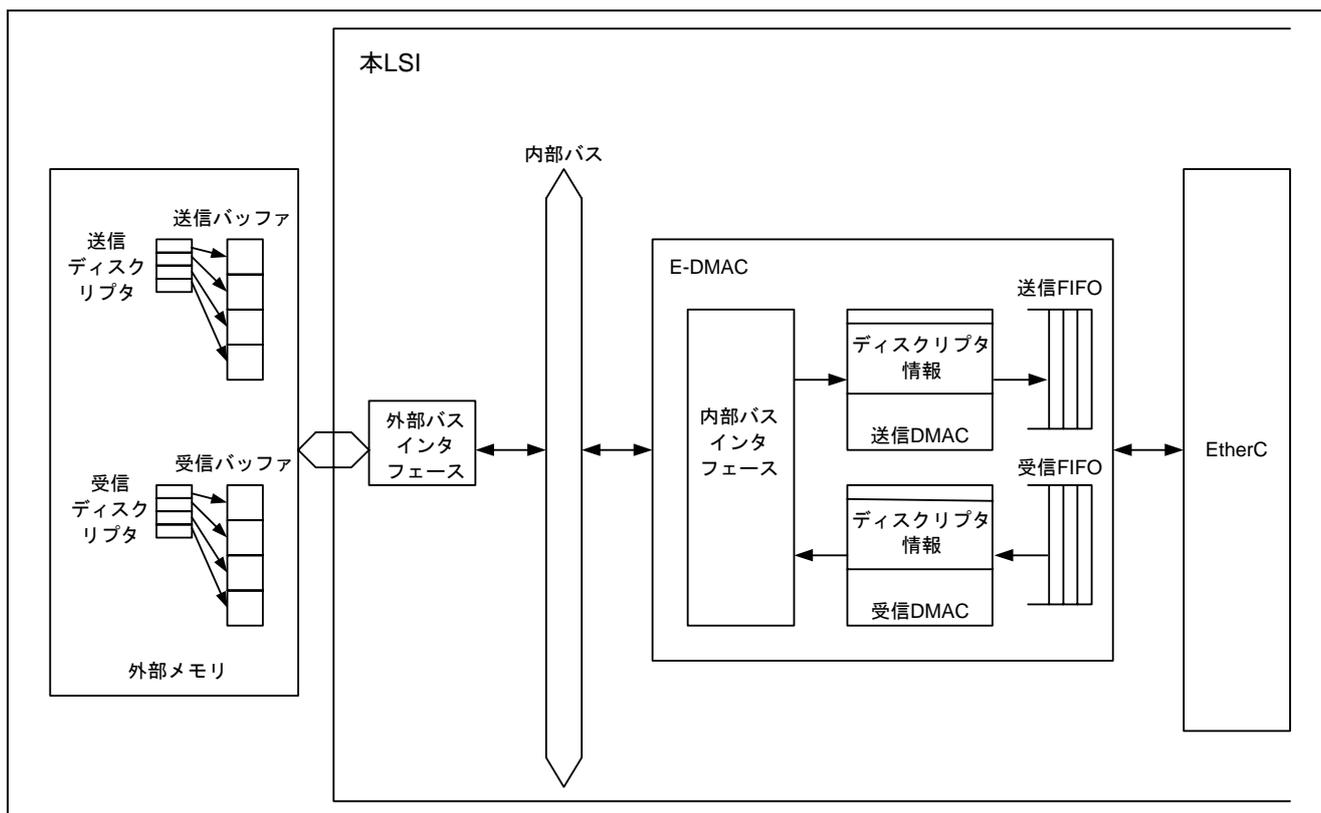


図3 E-DMAC とディスクリプタおよびバッファの構成



### 2.1.6 送信ディスクリプタの設定例

図5に送信ディスクリプタおよび送信バッファを3面使用した場合の例(1フレーム/1ディスクリプタ)を示します。ここでは1回の送信要求で1フレームだけ送信するものとします。図では各送信ディスクリプタをTD0部分のみに簡略して記載しています。図中の番号①、②等は実行順を示します。

設定は以下のようになります。

1. フレーム/1ディスクリプタ方式のため、全ディスクリプタ面のTFP1,TFP0ビットにB'11を設定します。
2. 全ディスクリプタ面のTACTビット、TFEビット、TFS26~TFS0ビットには初期値としてすべて0を設定します。
3. 第1面と第2面のディスクリプタのTDLEビットに0を設定します。第3面のディスクリプタのTDLEビットに1を設定することにより、第3面の処理を終了すると第1面のディスクリプタを読み込みます。このような設定によりディスクリプタをリング構造にすることができます。
4. 図5では省略していますが、当該ディスクリプタが参照している送信バッファのデータ長をTDLに、送信バッファの先頭アドレスをTBAに設定します。
5. この例では1回の送信要求で1フレームだけ送信するため、最初の送信では第1面のディスクリプタのTACTビットにだけ1を設定します。次の送信では第2面のディスクリプタのTACTビットにだけ1を設定します。送信手順の詳細は次章で説明します。

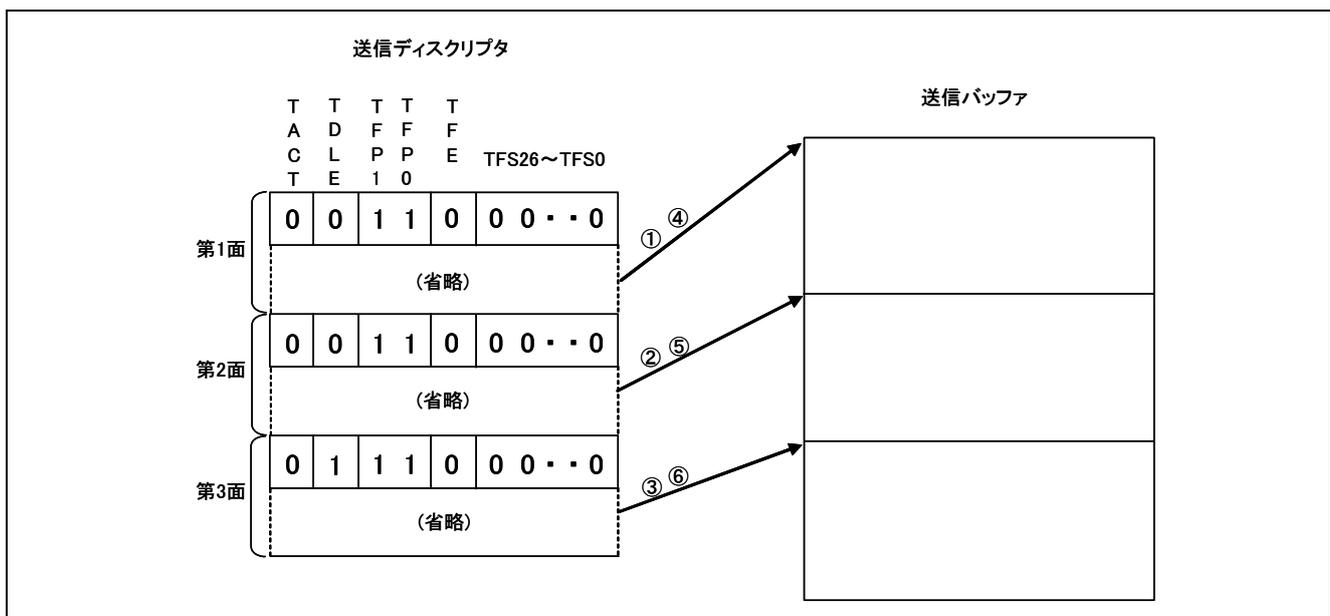


図5 送信ディスクリプタと送信バッファの関係

### 2.1.7 使用機能の動作手順(送信時)

EtherC モードレジスタ(ECMR)の TE ビットが 1 の状態で E-DMAC 送信要求レジスタ(EDTRR)の送信要求ビット(TR)に 1 を書き込むと E-DMAC 送信部が起動します。E-DMAC は EtherC/E-DMAC のソフトウェアリセット後、送信ディスクリプタ先頭アドレスレジスタ(TDLAR)で示すディスクリプタを読み込みます。読み込んだディスクリプタの TACT ビットが 1(有効)の場合は、E-DMAC は送信ディスクリプタの TD2 で指定される送信バッファ先頭アドレスから順次送信フレームデータを読み出して EtherC に転送します。EtherC は送信フレームを作成し MII に向けて送信を開始します。ディスクリプタ内で指示されるバッファ長分の DMA 転送後、送信ディスクリプタの TFP の値によって以下の処理を行います。

- TFP=B'00 or B'10(フレーム継続)

DMA 転送後、ディスクリプタのライトバック(TACT ビットの 0 書き込み)を行います。その後、次のディスクリプタの TACT ビットを読み込みます。

- TFP=B'01 or B'11(フレーム終了)

フレームの送信完了後、ディスクリプタのライトバック(TACT ビットの 0 およびステータスの書き込み)を行います。その後、次のディスクリプタの TACT ビットを読み込みます。

読み込んだディスクリプタの TACT ビットが 1 のときは、フレームの送信を継続し次のディスクリプタを読み込みます。TACT ビットが 0(無効)のディスクリプタを読み込むと、E-DMAC は EDTRR の TR ビットを 0 にして送信処理を完了します。TR ビットが 0 になった後 TR ビットに 1 を書き込むと再度 E-DMAC 送信部が起動しますが、この場合は最後に送信を行ったディスクリプタの次のディスクリプタを読み込みます。

図 6 に送信フローの例(1 フレーム/1 ディスクリプタ、複数ディスクリプタ面の場合)を示します。

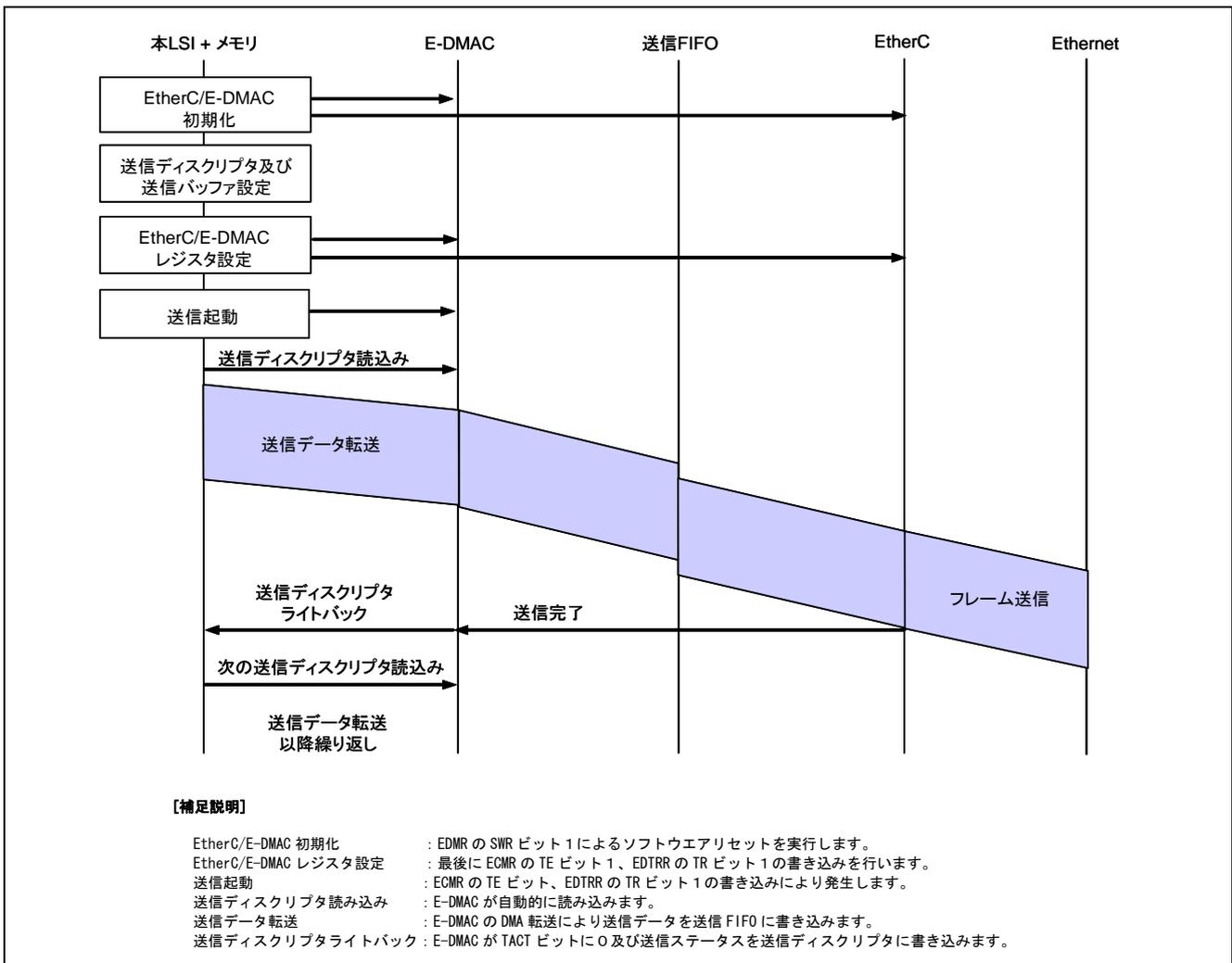


図6 送信フローの例(1 フレーム/1 ディスクリプタ)

## 2.1.8 使用機能の設定手順(送信時)

ここでは、イーサネット送信するための基本的な設定例について説明します。図7、図8にイーサネット送信設定フロー例を示します。

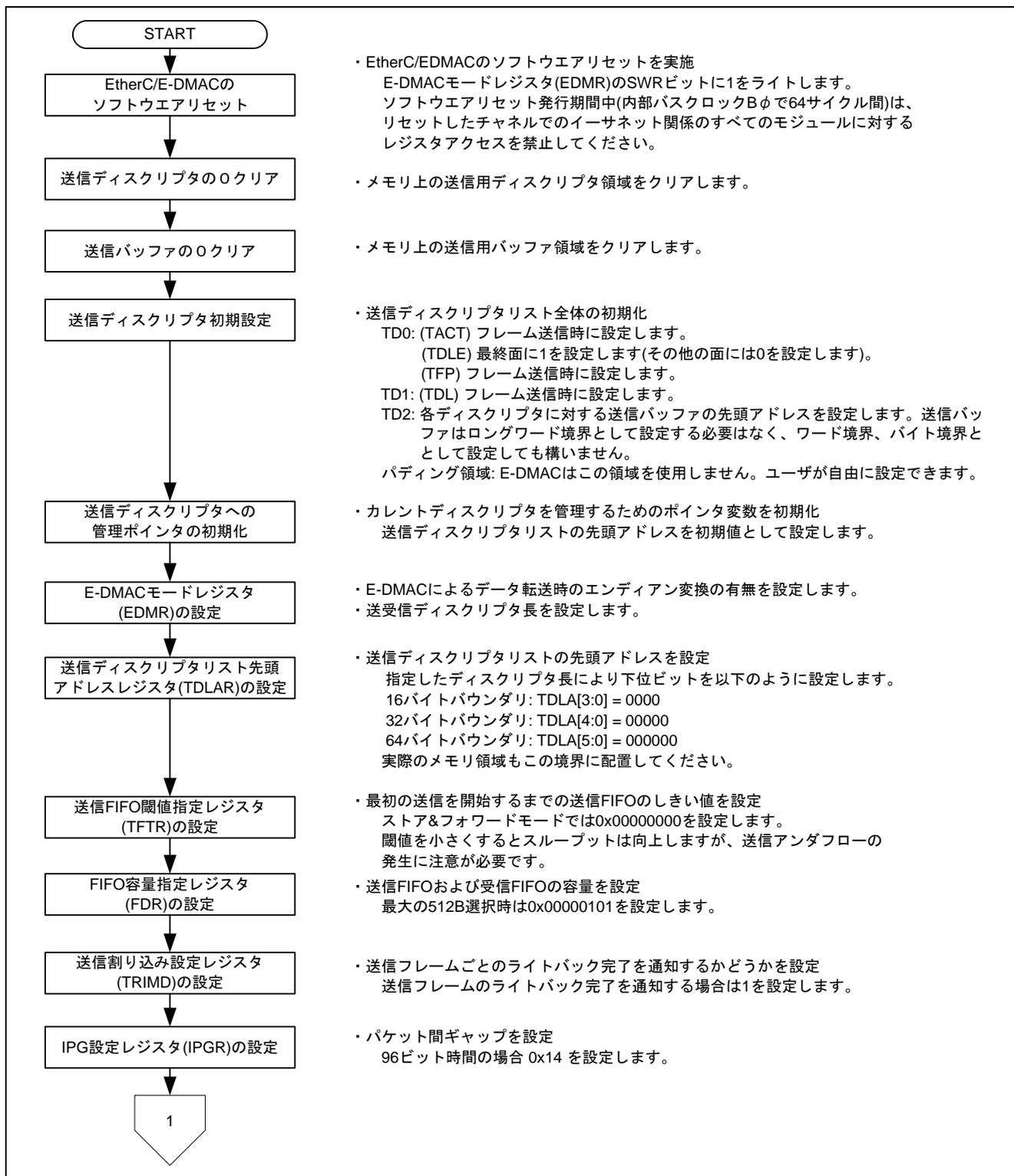


図7 イーサネット送信設定フロー例(1)

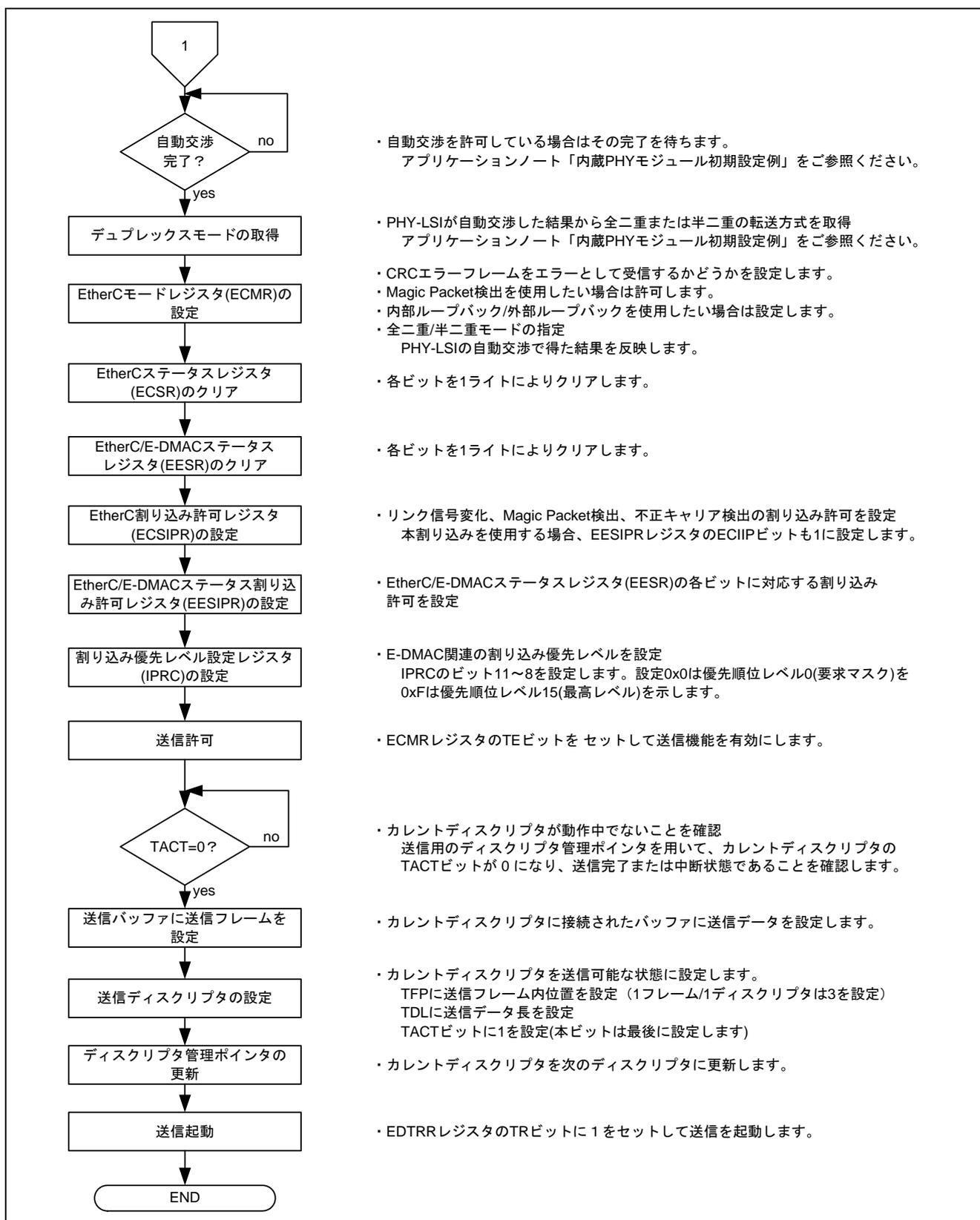


図8 イーサネット送信設定フロー例(2)

## 2.2 参考プログラムの動作

参考プログラムでは、EtherC のおよび E-DMAC を使用し、対向ホストに向けて 10 フレーム送信します。送信ディスクリプタと 1520 バイトの送信バッファを 4 面用意(1 フレーム/1 ディスクリプタ)しています。送信ディスクリプタをリング状にして使用しています。フレーム送信完了割り込み(TC)により 1 フレームの送信が完了したと判断し、次の送信を開始します。

送信データについては、イーサネットフレームのうちプリアンプル、スタートフレームデリミタ(SFD)、および CRC 部を除いた部分を用意する必要があります。ヘッダ部の宛先 MAC アドレス及び送信元 MAC アドレスは、ご使用になる製品の MAC アドレスに変更していただく必要があります。なお、EtherC は送信元 MAC アドレスのチェックは行いません。

図 9 に参考プログラムの動作環境を、図 10 にイーサネットフレームフォーマットを示します。

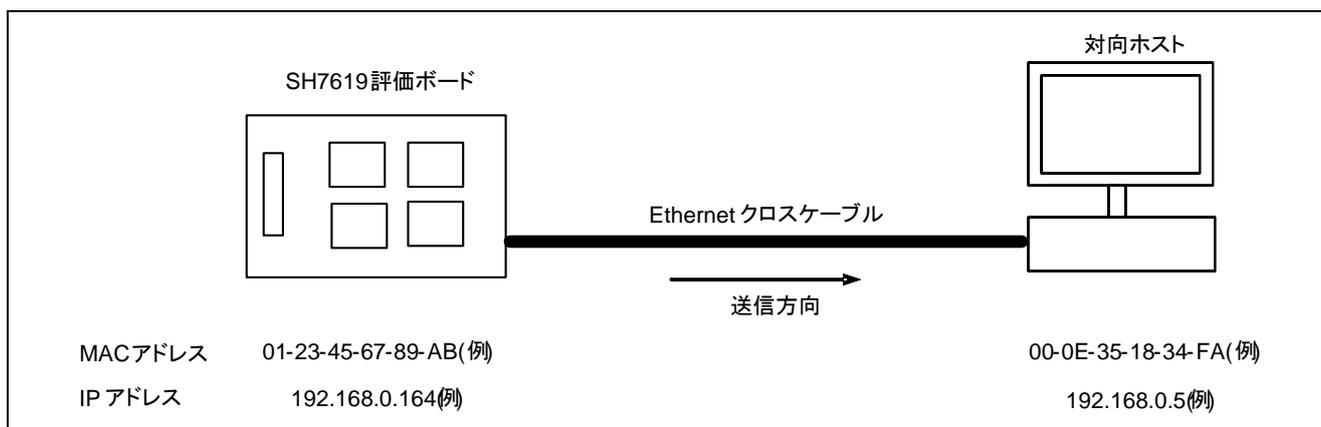


図9 参考プログラムの動作環境



図10 イーサネットフレームフォーマット

## 2.3 参考プログラムのディスクリプタ定義

E-DMAC ではディスクリプタのパディング領域を使用しません。ユーザが自由に使用できます。本プログラムではこの領域に次のディスクリプタの先頭アドレスを設定し、ソフトウェアにてもリング構造を実現しています。図 11に参考プログラムでの送信ディスクリプタ構造体の定義と送信ディスクリプタ列の使用例を示します。

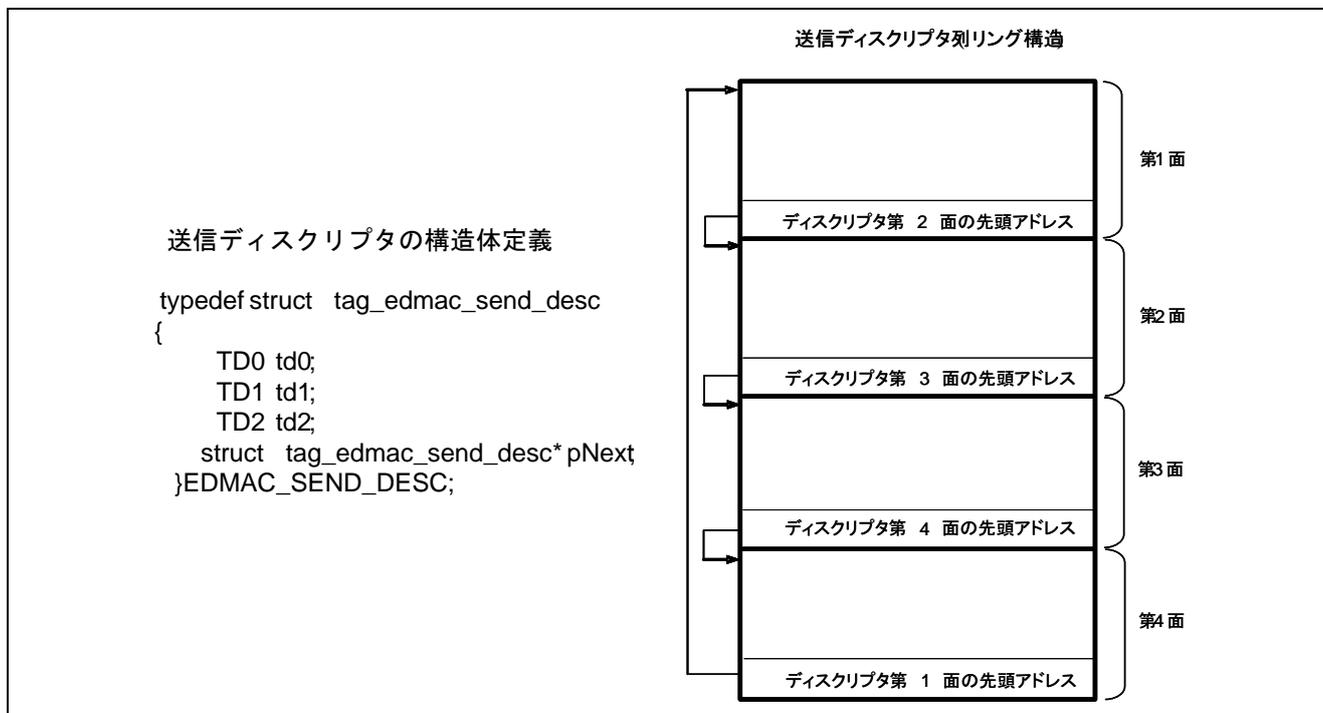


図11 送信ディスクリプタの構造体定義と送信ディスクリプタ列使用例

## 2.4 参考プログラムの処理手順

図 12～図 15に参考プログラムの処理フローを示します。なお、EtherC/E-DMAC の各種レジスタおよびデバイスクリプタの初期設定では受信の設定も行ってはいますが、受信処理自体は行っていません。

PHY 自動交渉関数 `phy_autonego` の詳細は「SH7670 グループ アプリケーションノート イーサネット PHY-LSI 自動交渉設定例」をご参照ください。

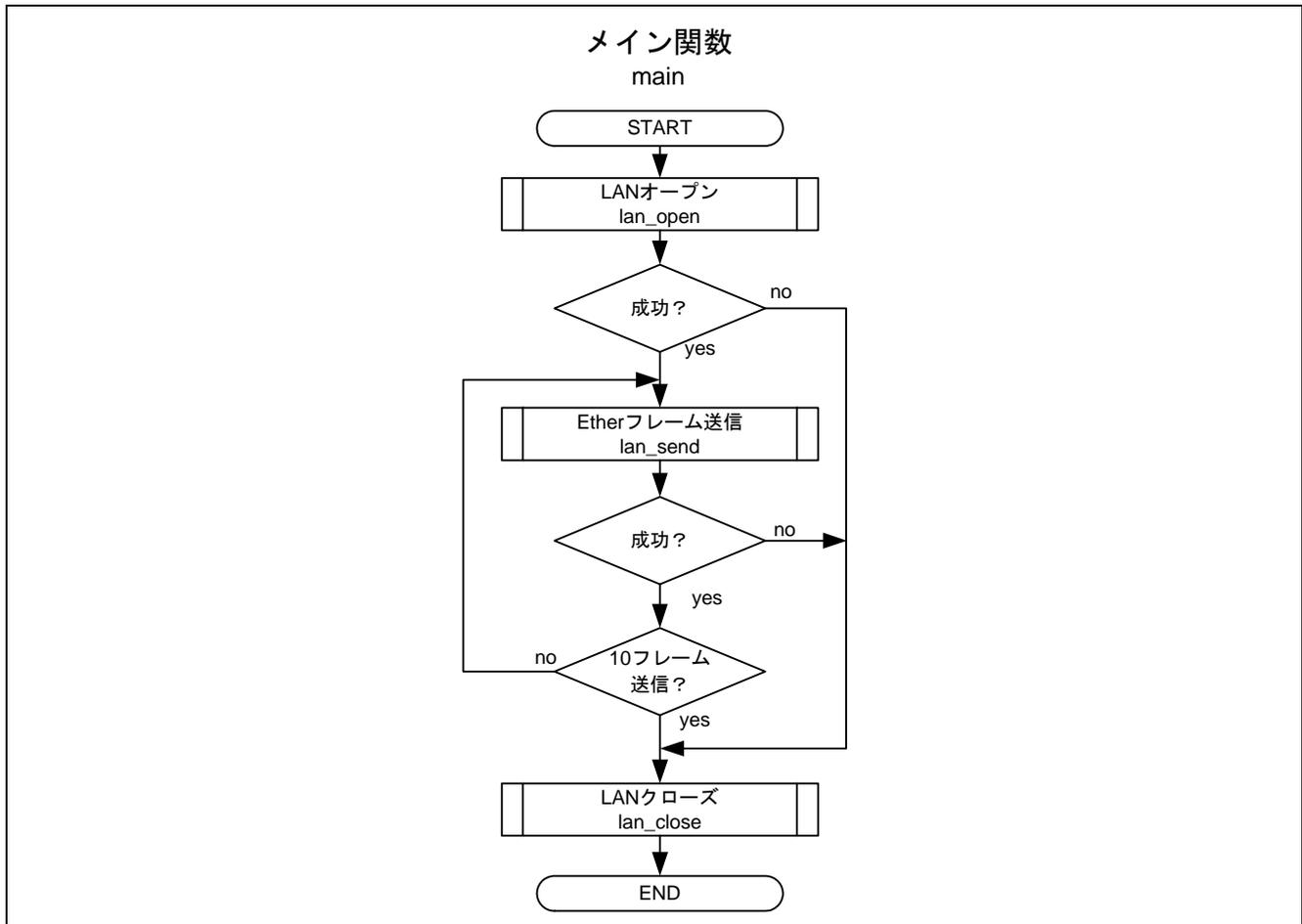


図12 メインプログラムの処理フロー例(1)

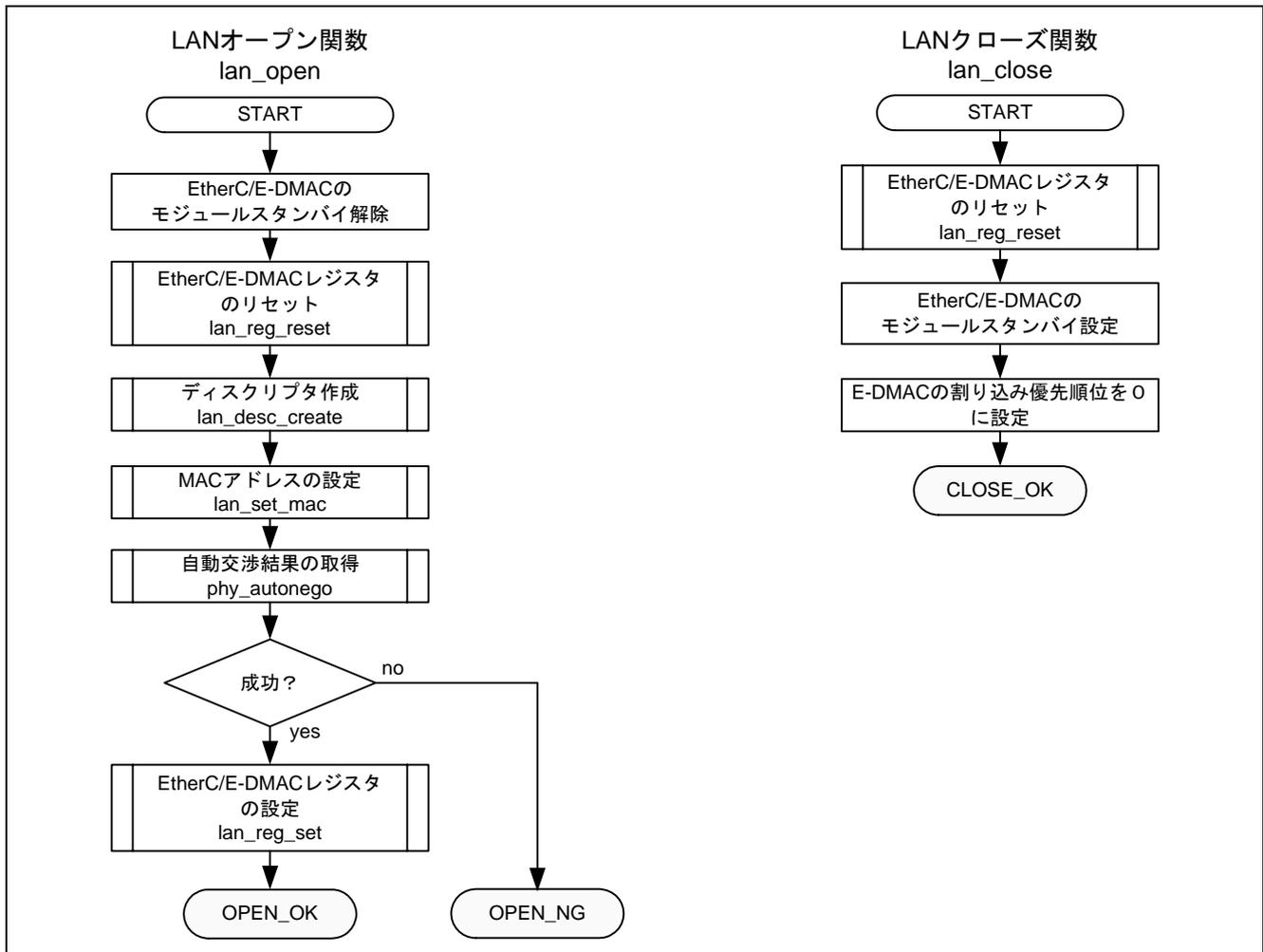


図13 参考プログラムの処理フロー例(2)

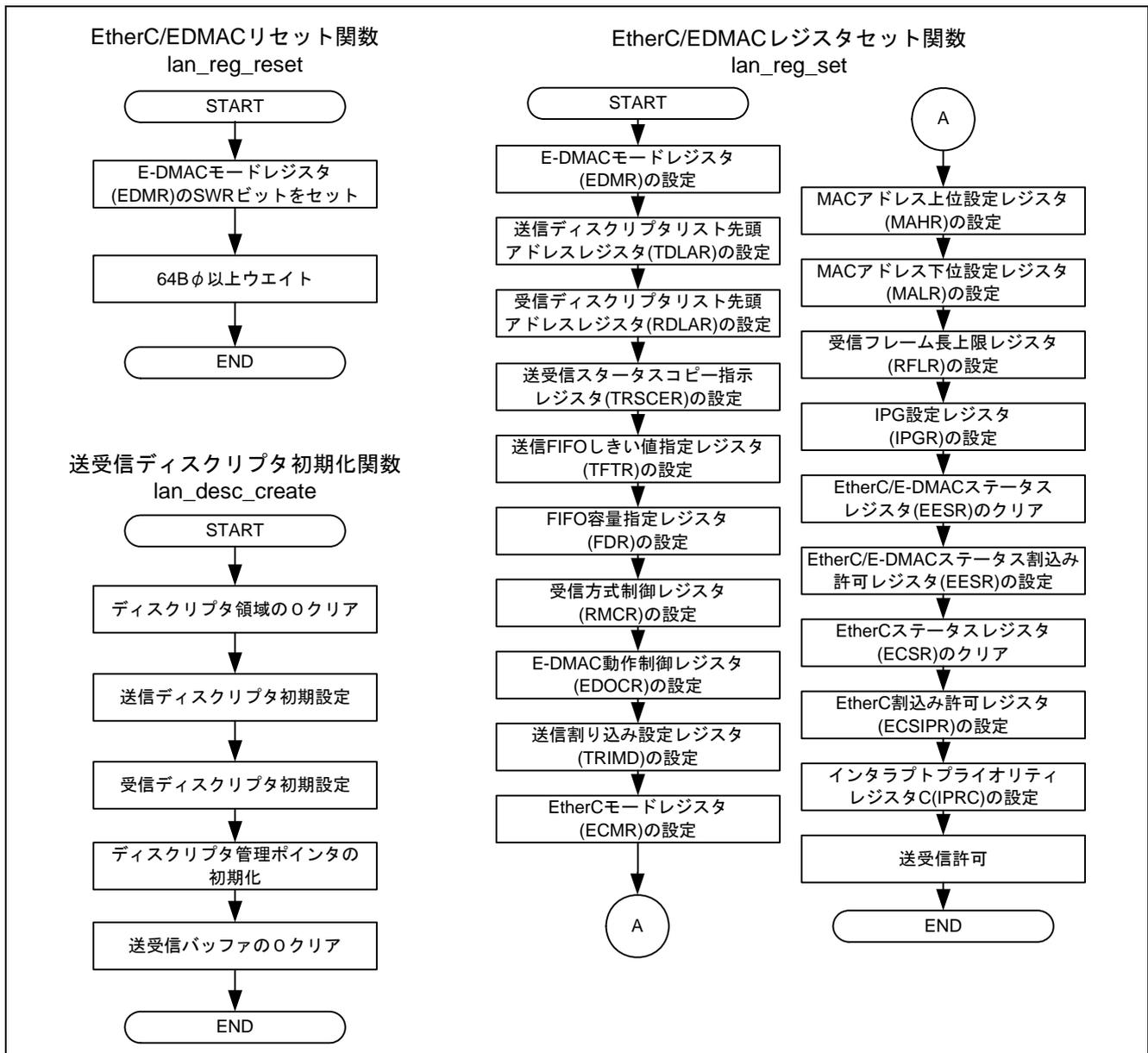


図14 参考プログラムの処理フロー例(3)

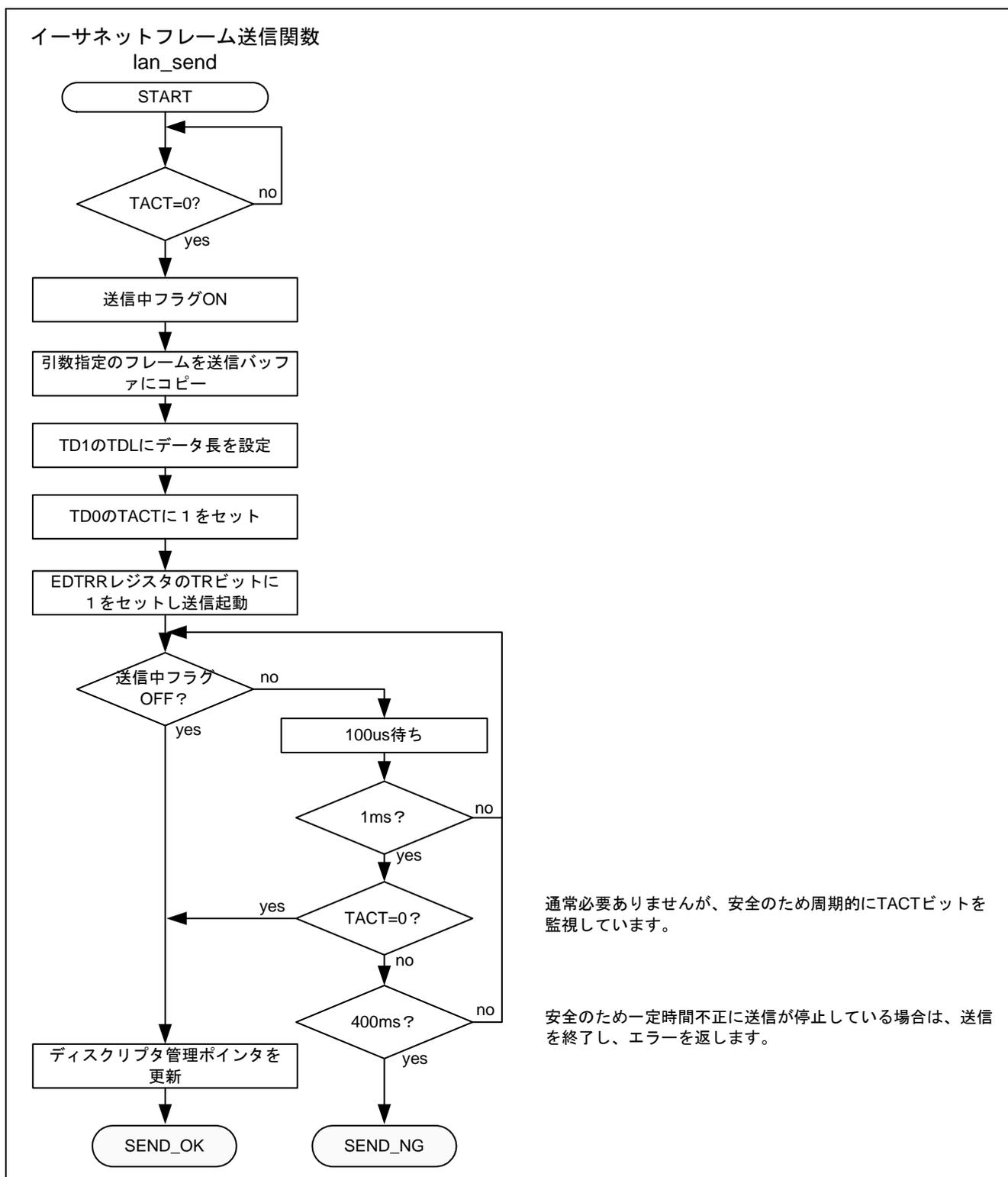


図15 参考プログラムの処理フロー例(4)

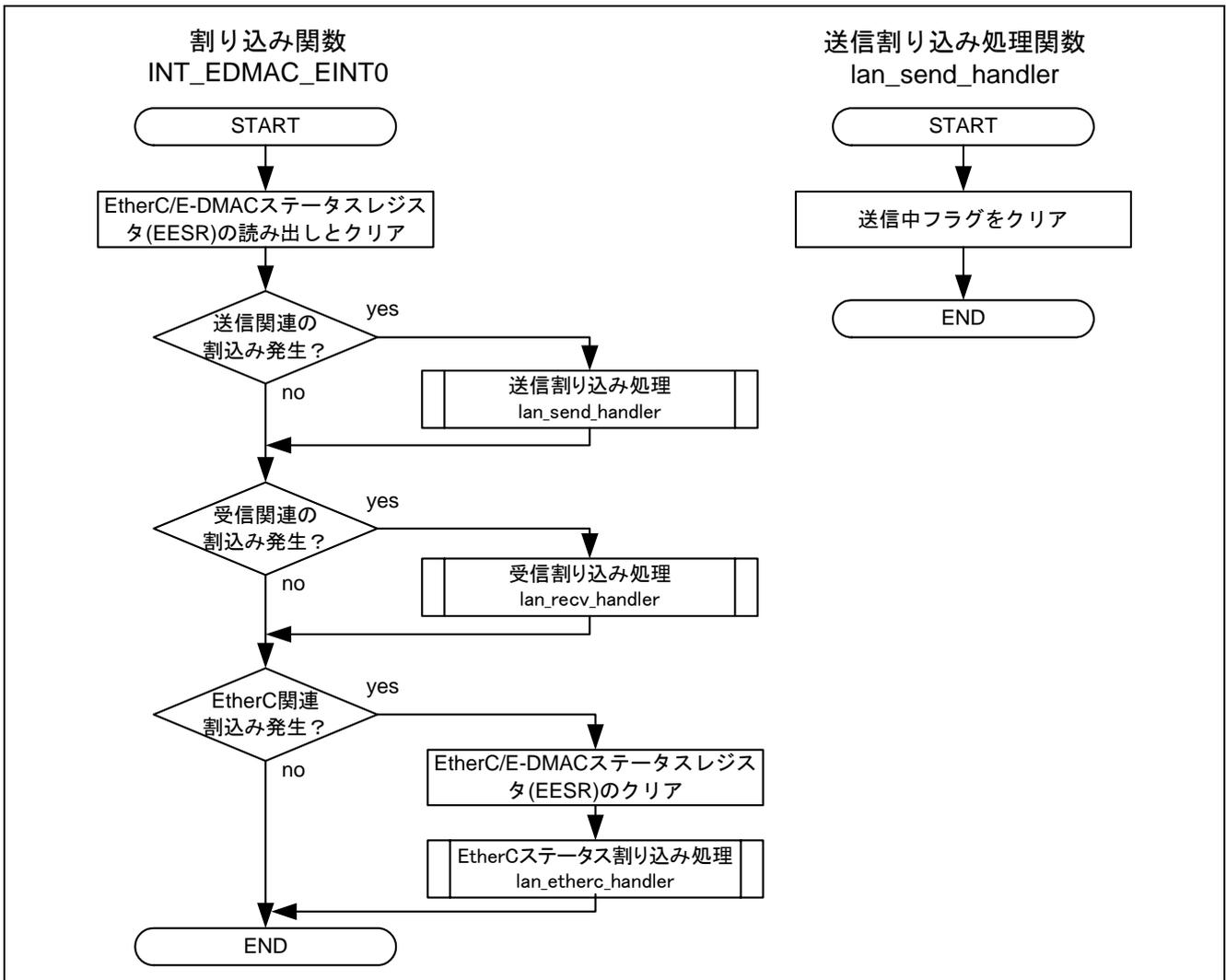


図16 参考プログラムの処理フロー例(5)

### 3. 参考プログラムリスト

#### 3.1 サンプルプログラムリスト"main.c" (1)

```
1  /*****
2  *   DISCLAIMER
3  *
4  *   This software is supplied by Renesas Electronics Corporation and is only
5  *   intended for use with Renesas products. No other uses are authorized.
6  *
7  *   This software is owned by Renesas Electronics Corporation and is protected under
8  *   all applicable laws, including copyright laws.
9  *
10 *   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 *   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 *   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 *   PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 *   DISCLAIMED.
15 *
16 *   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 *   ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 *   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 *   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 *   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 *
22 *   Renesas reserves the right, without notice, to make changes to this
23 *   software and to discontinue the availability of this software.
24 *   By using this software, you agree to the additional terms and
25 *   conditions found by accessing the following link:
26 *   http://www.renesas.com/disclaimer
27 *****/
28 * Copyright (C) 2007(2010) Renesas Electronics Corporation. All rights reserved.
29 * "FILE COMMENT"***** Technical reference data *****
30 * System Name : SH7671 Sample Program
31 * File Name   : main.c
32 * Abstract    : イーサネット送信設定例
33 * Version     : 1.00.01
34 * Device      : SH7671
35 * Tool-Chain  : High-performance Embedded Workshop (Ver.4.03.00).
36 *              : C/C++ compiler package for the SuperH RISC engine family
37 *              :                               (Ver.9.01 Release01).
38 * OS          : None
39 * H/W Platform: M3A-HS71(CPU board)
40 * Description :
41 *****/
42 * History     : Jul.04,2007 ver.1.00.00
43 *              : Oct.08,2010 ver.1.00.01 Changed the company name and device name
44 * "FILE COMMENT END"*****/
```

## 3.2 サンプルプログラムリスト"main.c" (2)

```
45  #include "iodefine.h"
46  #include "defs.h"
47  #include "ether.h"
48
49  /* **** プロトタイプ宣言 **** */
50  void main(void);
51
52  /* **** 変数宣言 **** */
53  static unsigned char s_frame[] = {
54      0xff,0xff,0xff,0xff,0xff,0xff, /* あて先 MAC アドレス          */
55      0x00,0x01,0x02,0x03,0x04,0x05, /* 送信元 MAC アドレス (00:01:02:03:04:05)*/
56      0x08,0x06, /* タイプ(ARP)                */
57      0x00,0x01, /* +--H/Wタイプ= Ethernet     */
58      0x08,0x00, /* +--プロトコルタイプ= IP    */
59      0x06,0x04, /* +--HW/プロトコルアドレス長 */
60      0x00,0x01, /* +--OPCODE= request         */
61      0x00,0x01,0x02,0x03,0x04,0x05, /* +--送信元 MAC アドレス(00:01:02:03:04:05) */
62      0xc0,0xa8,0x00,0x03, /* +--送信元 IP アドレス(192.168.0.3) */
63      0x00,0x00,0x00,0x00,0x00,0x00, /* +--問合せ先 MAC アドレス */
64      0xc0,0xa8,0x00,0x05, /* +--問合せ先 IP アドレス(192.168.0.5) */
65  };
66
```

## 3.3 サンプルプログラムリスト"main.c" (3)

```
67  /*"FUNC COMMENT"*****
68  * ID      :
69  * Outline  : イーサネット送信サンプルプログラム メイン関数
70  *-----
71  * Include  : #include "iodefine.h"
72  *-----
73  * Declaration : void main(void)
74  *-----
75  * Function  : 内蔵イーサネットコントローラ (EtherC) とイーサネットコント
76  *            : ローラ用ダイナミックメモリアクセスコントローラ (E-DMAC) を
77  *            : 使用して、イーサネットフレームを送信します。
78  *            : また PHY モジュールには、REALTEK 社製 RTL8201CP を使用します。
79  *            : 送信ディスクリプタを複数面用意して連続送信を行っています。
80  *-----
81  * Argument  : void
82  *-----
83  * ReturnValue : void
84  *-----
85  * Notice    : EEPROM から取得した Mac アドレスは送信フレームに反映させていません。
86  *"FUNC COMMENT END"*****/
87 void main(void)
88 {
89     int i;
90     int ret;
91
92     /* ==== イーサネット初期設定 ==== */
93     ret = lan_open();
94     if( ret == OPEN_OK ){
95         /* ==== 10 フレーム送信開始 ==== */
96         for(i=0; i<10; i++){
97             /* ---- 送信 ---- */
98             ret = lan_send( s_frame, sizeof(s_frame) );
99             if( ret != SEND_OK ){
100                break;
101            }
102        }
103    }
104    /* ==== イーサネット送受信停止 ==== */
105    lan_close();
106 }
107 /* End of file */
```

## 3.4 サンプルプログラムリスト"ether.c"(1)

```

1  /*****
2  *   DISCLAIMER
3  *
4  *   This software is supplied by Renesas Electronics Corporation and is only
5  *   intended for use with Renesas products. No other uses are authorized.
6  *
7  *   This software is owned by Renesas Electronics Corporation and is protected under
8  *   all applicable laws, including copyright laws.
9  *
10 *   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 *   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 *   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 *   PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 *   DISCLAIMED.
15 *
16 *   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 *   ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 *   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 *   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 *   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 *
22 *   Renesas reserves the right, without notice, to make changes to this
23 *   software and to discontinue the availability of this software.
24 *   By using this software, you agree to the additional terms and
25 *   conditions found by accessing the following link:
26 *   http://www.renesas.com/disclaimer
27 *****/
28 * Copyright (C) 2007(2010) Renesas Electronics Corporation. All rights reserved.
29 * "FILE COMMENT"***** Technical reference data *****
30 *   System Name : SH7671 Sample Program
31 *   File Name   : ether.c
32 *   Abstract    : イーサネット送信設定例
33 *   Version     : 1.02.01
34 *   Device      : SH7671
35 *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.03.00).
36 *               : C/C++ compiler package for the SuperH RISC engine family
37 *               :                               (Ver.9.01 Release01).
38 *   OS          : None
39 *   H/W Platform: M3A-HS71(CPU board)
40 *   Description :
41 *****/
42 *   History     : Jul.04,2007 ver.1.00.00
43 *               : Sep.20,2007 ver.1.01.00 MACをEEPROMから取得するコードを追加
44 *               : Oct.12,2007 ver.1.01.01 lan_reg_set() レジスタ設定順序変更
45 *               :                               lan_close() 割込みマスク追加
46 *               :                               lan_desc_create() メモリクリア追加
47 *               : Oct.30,2007 ver.1.01.02 lan_send() ウェイトfor文の条件判定修正
48 *               : Nov.07,2007 ver.1.01.03 lan_reg_set() ライトバック通知する
49 *               : Nov.26,2007 ver.1.01.04 周期監視処理のコメントを変更
50 *               : Mar.07,2008 ver.1.02.00 lan_set_mac 関数変更
51 *               : Oct.08,2010 ver.1.02.01 Changed the company name and device name
52 * "FILE COMMENT END"*****/

```

## 3.5 サンプルプログラムリスト” ether.c”(2)

```
53  #include "machine.h"
54  #include "string.h"
55  #include "iodefine.h"
56  #include "defs.h"
57  #include "phy.h"
58  #include "ether.h"
59  #include "siic.h"
60
61  /* **** マクロ宣言 **** */
62  #define DEVADDR_EEPROM 0                /* EEPROM の PIN 設定に依存 */
63  #define ROMADDR_MAC      0             /* EEPROM 内の MAC アドレス格納場所 */
64  #define DEFAULT_MAC_H  0x00010203     /* デバッグ用 */
65  #define DEFAULT_MAC_L  0x00000405
66  #define MACSET_OK      0
67  #define MACSET_NG     -1
68
69  /* **** プロトタイプ宣言 **** */
70  void main(void);
71  void lan_send_handler( unsigned long status );
72  static void lan_desc_create( void );
73  static void lan_reg_reset( void );
74  static void lan_reg_set( int link );
75  static int lan_set_mac( void );
76
77  /* **** 変数宣言 **** */
78  /* ---- ディスクリプタ ---- */
79  #pragma section ETH_DESC                /* 16 バイト境界へ配置 */
80  static volatile TXRX_DESCRIPTOR_SET desc; /* ディスクリプタ領域 */
81  #pragma section
82  /* ---- バッファ ---- */
83  #pragma section ETH_BUFF                /* 16 バイト境界へ配置 */
84  static volatile TXRX_BUFFER_SET buf;    /* 送受信バッファ領域 */
85  #pragma section
86  /* ---- MAC アドレス ---- */
87  static unsigned long my_macaddr_h;
88  static unsigned long my_macaddr_l;
89  /* ---- その他 ---- */
90  static volatile int f_send = 0;         /* 送信中フラグ */
91
```

## 3.6 サンプルプログラムリスト” ether.c”(3)

```
92  /*"FUNC COMMENT"*****
93  * ID      :
94  * Outline : イーサネットオープン関数
95  *-----
96  * Include : #include "iodefine.h"
97  *         : #include "phy.h"
98  *         : #include "ether.h"
99  *-----
100 * Declaration : int lan_open(void)
101 *-----
102 * Function   : E-DMAC, EtherC, PHY, バッファメモリの初期化を行う。
103 *           : 関数内でイーサネットに必要な初期化を行い、送受信可能な状態にする。
104 *           : 送受信可能な状態に設定することができないときはエラーを返す。
105 *-----
106 * Argument   : void
107 *-----
108 * ReturnValue : OPEN_OK(0) :オープン成功
109 *           : OPEN_NG(-1):オープン失敗
110 *-----
111 * Notice     :
112 *"FUNC COMMENT END"*****/
113 int lan_open(void)
114 {
115     int link;
116
117     /* ==== PFC の設定 ==== */
118     //PORT.PBCRL1.BIT.PB6MD = 1;      /* DK30686 基板使用時の設定 */
119     PORT.PCCRHL.WORD = 0x0155;      /* EtherC 機能 */
120     PORT.PCCRL1.WORD = 0x5555;
121     PORT.PCCRL2.WORD = 0x5555;
122     /* ==== EtherC/EDMAC のモジュールスタンバイ解除 ==== */
123     CPG.STBCR4.BIT.MSTP40 = 0;
124     /* ==== EtherC,E-DMAC の停止 === */
125     lan_reg_reset();
126     /* ==== バッファメモリの初期化 ==== */
127     lan_desc_create();
128     /* ==== MAC アドレス取得 ==== */
129     lan_set_mac();
130     /* ==== EtherC,E-DMAC の設定 ==== */
131     link = phy_autonego();          /* デュプレックスモード確認 */
132     if( link == NEGO_FAIL ){
133         return OPEN_NG;            /* OPEN 失敗 */
134     }
135     else{
136         lan_reg_set(link);
137     }
138     return OPEN_OK;
139 }
```

## 3.7 サンプルプログラムリスト” ether.c”(4)

```
140 /*"FUNC COMMENT"*****
141 * ID      :
142 * Outline : イーサネットクローズ関数
143 *-----
144 * Include : #include "iodefine.h"
145 *         : #include "ether.h"
146 *-----
147 * Declaration : int lan_close(void)
148 *-----
149 * Function    : EDMAC/EtherC を停止する。
150 *             : EDMAC/EtherC はクロックの供給も停止する。
151 *-----
152 * Argument    : void
153 *-----
154 * ReturnValue : int  CLOSE_OK( 0):クローズ成功
155 *             :      CLOSE_NG(-1):クローズ失敗
156 *-----
157 * Notice     :
158 *"FUNC COMMENT END"*****/
159 int lan_close( void )
160 {
161     int i;
162
163     /* ==== EtherC,E-DMAC のリセット ==== */
164     lan_reg_reset();
165     /* ==== EtherC,E-DMAC の停止 ==== */
166     CPG.STBCR4.BIT.MSTP40 = 1;
167     /* ==== E-DMAC 関連の割り込みを禁止 ==== */
168     INTC.IPR12.BIT._ETC = 0;
169
170     return CLOSE_OK;
171 }
```

## 3.8 サンプルプログラムリスト” ether.c”(5)

```

172
173 /*"FUNC COMMENT"*****
174 * ID      :
175 * Outline : フレーム送信関数
176 *-----
177 * Include : #include "ether.h"
178 *         : #include "iodefine.h"
179 *-----
180 * Declaration : int lan_send( unsigned char *addr, int flen )
181 *-----
182 * Function    : 指定されたフレームを送信ディスクリプタに登録されたバッファに
183 *              : コピーして送信します。送信が完了するまで本関数内で待ちます。
184 *              : 安全のため周期的に EDMAC を監視しています。
185 *-----
186 * Argument   : なし
187 *-----
188 * ReturnValue : SEND_OK(0) : 登録成功
189 *             : SEND_NG(-1): 登録失敗
190 *-----
191 * Notice     :
192 /*"FUNC COMMENT END"*****/
193 int lan_send( unsigned char *addr, int flen )
194 {
195     int i;
196     int tlms = 0;
197     int t400ms = 0;
198
199     /* ==== 送信中ではないことを確認 ==== */
200     while( desc.pSend_top->td0.BIT.TACT == 1 ){
201         /* wait */
202     }
203     /* ==== 送信中フラグをセット ==== */
204     f_send = 1;                               /* 送信完了割込みでクリア */
205
206     /* ==== 送信ディスクリプタの更新 ==== */
207     memcpy( desc.pSend_top->td2.TBA, addr, flen ); /* 送信データ */
208     if( flen < 60 ){                             /* 最小フレーム 60 バイト */
209         memcpy( (desc.pSend_top->td2.TBA)+flen, 0, 60-flen ); /*パディング*/
210         flen = 60;
211     }
212     desc.pSend_top->td1.TDL = flen;                /* データ長 */
213     desc.pSend_top->td0.BIT.TACT = 1;             /* 送信許可 */
214
215     /* ==== 送信停止中であれば起動 ==== */
216     if( EDMAC.EDTRR.BIT.TR == 0 ){               /* 必ずリードして確認する*/
217         EDMAC.EDTRR.BIT.TR = 1;
218     }

```

## 3.9 サンプルプログラムリスト” ether.c”(6)

```
219     /* ==== 送信完了の確認 ==== */
220     while( f_send ){
221         for( i=LOOP_100us; i>0; i-- ){
222             /* 100us wait */
223         }
224         /* ---- 1ms 経過時はディスクリプタも確認 ---- */
225         if( ++t1ms > 10 ){
226             t1ms = 0;
227             if( desc.pSend_top->td0.BIT.TACT == 0 ){
228                 break;
229             }
230         }
231         /* ---- 400ms 経過したら EDMAC 動作停止と判断 ---- */
232         if( ++t400ms > 4000 ){
233             t400ms = 0;
234             return SEND_NG;
235         }
236     }
237     /* ==== カレントポインタの更新 ==== */
238     desc.pSend_top = desc.pSend_top->pNext;
239
240     return SEND_OK;
241 }
242
243 /* "FUNC COMMENT"*****
244 * ID      :
245 * Outline : ディスクリプタ構築関数
246 *-----
247 * Include : #include "ether.h"
248 *-----
249 * Declaration : static void lan_desc_create( void )
250 *-----
251 * Function   : イーサネットで必要な送受信バッファとディスクリプタの初期化を
252 *             : 行います。1 フレーム/1 バッファを前提としています。
253 *-----
254 * Argument   : void
255 *-----
256 * ReturnValue : void
257 *-----
258 * Notice     :
259 * "FUNC COMMENT END"*****
```

## 3.10 サンプルプログラムリスト” ether.c”(7)

```
260 static void lan_desc_create( void )
261 {
262     int i;
263     /* ==== ディスクリプタ領域の構築 ==== */
264     /* ---- メモリクリア ---- */
265     memset(&desc, 0, sizeof(desc) );
266     /* ---- 送信ディスクリプタ ---- */
267     for(i=0; i<NUM_OF_TX_DESCRIPTOR; i++){
268         desc.send[i].td2.TBA = buf.send[i];          /* TD2 */
269         desc.send[i].td1.TDL = 0;                    /* TD1 */
270         desc.send[i].td0.LONG= 0x30000000;          /* TD0:1frame/1buf、送信無効*/
271         if( i != (NUM_OF_TX_DESCRIPTOR-1) ){        /* pNext */
272             desc.send[i].pNext = &desc.send[i+1];
273         }
274     }
275     desc.send[i-1].td0.BIT.TDLE = 1;
276     desc.send[i-1].pNext = &desc.send[0];
277     /* ---- 受信ディスクリプタ ---- */
278     for(i=0; i<NUM_OF_RX_DESCRIPTOR; i++){
279         desc.recv[i].rd2.RBA = buf.recv[i];          /* RD2 */
280         desc.recv[i].rd1.RBL = SIZE_OF_BUFFER;      /* RD1 */
281         desc.recv[i].rd0.LONG= 0xb0000000;          /* RD0:1frame/1buf、受信有効*/
282         if( i != (NUM_OF_RX_DESCRIPTOR-1) ){        /* pNext */
283             desc.recv[i].pNext = &desc.recv[i+1];
284         }
285     }
286     desc.recv[i-1].rd0.BIT.RDLE = 1;                 /* 最終ディスクリプタの設定 */
287     desc.recv[i-1].pNext = &desc.recv[0];
288
289     /* ---- ディスクリプタ管理情報の初期化 ---- */
290     desc.pSend_top = &desc.send[0];
291     desc.pRecv_end = &desc.recv[0];
292
293     /* ==== バッファ領域の構築 ==== */
294     /* ---- 領域クリア ---- */
295     memset(&buf, 0, sizeof(buf) );
296 }
```

## 3.11 サンプルプログラムリスト” ether.c”(8)

```

297  /*"FUNC COMMENT"*****
298  * ID      :
299  * Outline : EtherC,E-DMAC レジスタ初期化関数
300  *-----
301  * Include : #include "iodefine.h"
302  *-----
303  * Declaration : static void lan_reg_reset( void )
304  *-----
305  * Function   : EtherC と E-DMAC のレジスタをリセットします。
306  *           : 関数内でリセット期間を Bφ 64 サイクル以上確保しています。
307  *-----
308  * Argument   : void
309  *-----
310  * ReturnValue : void
311  *-----
312  * Notice     :
313  *"FUNC COMMENT END"*****/
314  static void lan_reg_reset( void )
315  {
316     volatile int j = 100;          /* Bφ 64 サイクルウェイト */
317
318     /* ---- ソフトウェアリセット ---- */
319     EDMAC.EDMR.BIT.SWR = 1;
320
321     /* ---- リセット期間の確保 ---- */
322     while(j--){
323         /* Bφ 64 サイクル ウェイト */
324     }
325 }
326 /*"FUNC COMMENT"*****
327 * ID      :
328 * Outline : EtherC,E-DMAC レジスタ設定
329 *-----
330 * Include : #include "iodefine.h"
331 *           : #include "phy.h"
332 *           : #include "ether.h"
333 *-----
334 * Declaration : void lan_reg_set(int link)
335 *-----
336 * Function   : E-DMAC, EtherC の初期化を行います。
337 *           : 送受信ともに有効に設定しています。
338 *-----
339 * Argument   : int link : I :EtherC に設定するデュプレックスモード。
340 *           :           :phy_autonego 関数の戻り値を利用。
341 *-----
342 * ReturnValue : void
343 *-----
344 * Notice     : 本関数は EDMAC ソフトウェアリセット後の送受信停止状態で実行ください。
345  *"FUNC COMMENT END"*****/

```

## 3.12 サンプルプログラムリスト” ether.c”(9)

```

346 static void lan_reg_set( int link )
347 {
348     /* ==== EDMAC ==== */
349     EDMAC.EDMR.LONG = 0x00000000; /* エンディアン変換なし(ビッグエンディアン) */
350     /* ディスクリプタ長 16 バイト */
351     EDMAC.TDLAR = &desc.send[0]; /* 送信ディスクリプタ先頭 */
352     EDMAC.RDLAR = &desc.recv[0]; /* 受信ディスクリプタ先頭 */
353     EDMAC.TRSCER.LONG = 0x00000000; /* 全ステータスをディスクリプタにコピー */
354     EDMAC.TFTR = 0x00; /* 送信 FIFO しきい値(ストア&フォワード) */
355     EDMAC.FDR.BIT.TFD = 1; /* 送信 FIFO 容量(512 バイト) */
356     EDMAC.FDR.BIT.RFD = 1; /* 受信 FIFO 容量(512 バイト) */
357     EDMAC.RMCR.BIT.RNC = 1; /* 連続受信可能 */
358     EDMAC.EDOCR.LONG = 0x00000000; /* FIFO エラー時も動作継続 */
359     EDMAC.FCFTR.LONG = 0x00070000; /* フロー制御しきい値設定 ※EtherC で禁止*/
360     EDMAC.TRIMD.BIT.TIS = 1; /* ライトバック完了通知をする */
361     /* ==== EtherC ==== */
362     EtherC.ECMR.LONG = 0x00000000; /* フロー制御無効 */
363     /* CRC フレームを「エラーあり」と認識 */
364     /* Magic Packet 検出を許可しない */
365     /* 受信禁止 */
366     /* 送信禁止 */
367     /* 内部ループバックを行わない */
368     /* 外部ループバックを行わない */
369     /* デュプレックスモード (半二重) */
370     /* プロミスキャスモードを行わない */
371     if( link == FULL_TX || link == FULL_10M ){
372         EtherC.ECMR.BIT.DM = 1; /* 全二重モードに設定 */
373     }
374     EtherC.MAHR = my_macaddr_h; /* MAC アドレスの設定 */
375     EtherC.MALR = my_macaddr_l;
376     EtherC.RFLR = 0x000; /* 最大受信フレーム長(1518 バイト) */
377     EtherC.IPGR = 0x14; /* パケット間ギャップ(96 ビット時間) */
378     /* ==== 割り込み関連 ==== */
379     EDMAC.EESR.LONG = 0x47FF0F9F; /* 全ステータスクリア(1 ライトでクリア) */
380     EDMAC.EESIPR.LONG = EDMAC_EESIPR_INI_SEND | EDMAC_EESIPR_INI_RECV |
EDMAC_EESIPR_INI_EtherC;
381     /* 送受信+EtherC の割り込みを許可 */
382     EtherC.ECSR.LONG = 0x00000017; /* 全ステータスクリア(1 ライトでクリア) */
383     EtherC.ECSIPR.LONG = EtherC_ECSIPR_INI; /* 割り込み許可 */
384     INTC.IPR12.BIT._ETC = 5; /* E-DMAC(EINT0)割り込み優先順位 5 */
385     /* ==== 送受信の許可設定 ==== */
386     /* ---- EtherC ---- */
387     EtherC.ECMR.BIT.RE = 1; /* 受信機能有効 */
388     EtherC.ECMR.BIT.TE = 1; /* 送信機能有効 */
389     /* ---- E-DMAC ---- */
390     if(EDMAC.EDRRR.BIT.RR == 0){
391         EDMAC.EDRRR.BIT.RR = 0; /* 受信無効 */
392     }
393 }

```

## 3.13 サンプルプログラムリスト” ether.c”(10)

```
394 /*"FUNC COMMENT"*****
395 * ID      :
396 * Outline : 送信割込み関数
397 *-----
398 * Include : #include "iodefine.h"
399 *         : #include "ether.h"
400 *-----
401 * Declaration : void lan_send_handler( unsigned long status )
402 *-----
403 * Function    : EDMAC(EESR)に関する送信関連の割込みハンドラ。
404 *             :
405 *-----
406 * Argument    : unsigned long status : I : EESR の状態 (割込み許可したビットのみ)
407 *-----
408 * ReturnValue : なし
409 *-----
410 * Notice     :
411 *"FUNC COMMENT END"*****/
412 void lan_send_handler( unsigned long status )
413 {
414     /* ==== 送信中フラグをクリア ==== */
415     f_send = 0;
416 }
417 /*"FUNC COMMENT"*****
418 * ID      :
419 * Outline : 受信割込み関数
420 *-----
421 * Include : #include "iodefine.h"
422 *         : #include "ether.h"
423 *-----
424 * Declaration : void lan_recv_handler( unsigned long status )
425 *-----
426 * Function    : EDMAC(EESR)に関する受信関連の割込みハンドラ。
427 *             :
428 *-----
429 * Argument    : unsigned long status : I : EESR の状態 (割込み許可したビットのみ)
430 *-----
431 * ReturnValue : なし
432 *-----
433 * Notice     : 本サンプルでは何も行いません。
434 *"FUNC COMMENT END"*****/
435 void lan_recv_handler( unsigned long status )
436 {
437 }
```

## 3.14 サンプルプログラムリスト” ether.c”(11)

```

438 /*"FUNC COMMENT"*****
439 * ID      :
440 * Outline : EtherC 割込み関数
441 *-----
442 * Include : #include "iodefine.h"
443 *         : #include "ether.h"
444 *-----
445 * Declaration : void lan_etherc_handler( unsigned long status )
446 *-----
447 * Function    : EtherC(ECSR)に関する割込みハンドラ。
448 *             :
449 *-----
450 * Argument    : unsigned long status : I : ECSR の状態 (割込み許可したビットのみ)
451 *-----
452 * ReturnValue : なし
453 *-----
454 * Notice     : 本サンプルでは何も行いません。
455 *"FUNC COMMENT END"*****/
456 void lan_etherc_handler( unsigned long status )
457 {
458 }
459
460 /*"FUNC COMMENT"*****
461 * ID      :
462 * Outline : MAC アドレス設定
463 *-----
464 * Include : #include "siic.h"
465 *-----
466 * Declaration : static int lan_set_mac( void )
467 *-----
468 * Function    : EEPROM から MAC アドレスを取得する関数
469 *-----
470 * Argument    : なし
471 *-----
472 * ReturnValue : MACSET_OK(0) :取得成功
473 *             : MACSET_NG(-1):取得失敗
474 *-----
475 * Notice     :
476 *"FUNC COMMENT END"*****/
477 static int lan_set_mac( void )
478 {
479     volatile int ret, i;
480     unsigned char buf[10];
481
482     /* ==== EEPROM driver 初期設定 ==== */
483     siic_Init_Driver();
484

```

【注】 iic3 関連関数(siic\_xxx)の詳細は、本アプリケーションノート付属のサンプルプログラムソースファイルをご参照ください。

## 3.15 サンプルプログラムリスト” ether.c”(12)

```
485     /* ==== EEPROMをリード ==== */
486     ret = siic_EepRomRW(DEVADDR_EEPROM, ROMADDR_MAC, 6, buf, SIIC_MODE_EEP_READ);
487     if (ret < SIIC_OK) {
488         /* ---- リード失敗 ---- */
489         my_macaddr_h = DEFAULT_MAC_H;
490         my_macaddr_l = DEFAULT_MAC_L;
491         return MACSET_NG;
492     }
493     do{
494         ret = siic_Chk_Eep();
495         if( ret < SIIC_OK ){
496             /* ---- リード失敗 ---- */
497             my_macaddr_h = DEFAULT_MAC_H;
498             my_macaddr_l = DEFAULT_MAC_L;
499             return MACSET_NG;
500         }
501     }while( ret != SIIC_OK);
502     /* ---- リード成功 ---- */
503     for(i=0; i<6; i++){
504         if( buf[i] != 0xff ){
505             break;
506         }
507     }
508     if( i == 6 ){
509         /* ---- Eeprom が未設定の場合はデフォルト値を設定 ---- */
510         my_macaddr_h = DEFAULT_MAC_H;
511         my_macaddr_l = DEFAULT_MAC_L;
512     }
513     else{
514         /* ---- リードしたアドレスを設定 ---- */
515         my_macaddr_h = buf[0];
516         my_macaddr_h <<= 8;
517         my_macaddr_h |= buf[1];
518         my_macaddr_h <<= 8;
519         my_macaddr_h |= buf[2];
520         my_macaddr_h <<= 8;
521         my_macaddr_h |= buf[3];
522         my_macaddr_l = buf[4];
523         my_macaddr_l <<= 8;
524         my_macaddr_l |= buf[5];
525     }
526     return MACSET_OK;
527 }
528
529 /* End of file */
```

【注】 iic3 関連関数(siic\_xxx)の詳細は、本アプリケーションノート付属のサンプルプログラムソースファイルをご参照ください。

## 3.16 サンプルプログラムリスト” ether.h”(1)

```
1  /*****
2  *  DISCLAIMER
3  *
4  *  This software is supplied by Renesas Electronics Corporation and is only
5  *  intended for use with Renesas products. No other uses are authorized.
6  *
7  *  This software is owned by Renesas Electronics Corporation and is protected under
8  *  all applicable laws, including copyright laws.
9  *
10 *  THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 *  REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 *  INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 *  PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 *  DISCLAIMED.
15 *
16 *  TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 *  ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 *  FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 *  FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 *  AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 *
22 *  Renesas reserves the right, without notice, to make changes to this
23 *  software and to discontinue the availability of this software.
24 *  By using this software, you agree to the additional terms and
25 *  conditions found by accessing the following link:
26 *  http://www.renesas.com/disclaimer
27 *****/
28 * Copyright (C) 2007(2010) Renesas Electronics Corporation. All rights reserved.
29 * "FILE COMMENT"***** Technical reference data *****
30 * System Name : SH7671 Sample Program
31 * File Name   : ether.h
32 * Abstract    : イーサネット送信設定例
33 * Version     : 1.00.02
34 * Device      : SH7671
35 * Tool-Chain  : High-performance Embedded Workshop (Ver.4.03.00).
36 *              : C/C++ compiler package for the SuperH RISC engine family
37 *              :                               (Ver.9.01 Release01).
38 * OS          : None
39 * H/W Platform: M3A-HS71(CPU board)
40 * Description :
41 *****/
42 * History     : Jul.04,2007 ver.1.00.00
43 *              : Nov.07,2007 ver.1.00.01 送信完了割込→ライトバック完了割込
44 *              : Oct.08,2010 ver.1.00.02 Changed the company name and device name
45 * "FILE COMMENT END"*****/
46 #ifndef _ETHER_H
47 #define _ETHER_H
48
```

## 3.17 サンプルプログラムリスト” ether.h”(2)

```

49  /* **** マクロ定義 **** */
50  #define NUM_OF_TX_DESCRIPTOR 4
51  #define NUM_OF_RX_DESCRIPTOR 4
52  #define NUM_OF_TX_BUFFER 4
53  #define NUM_OF_RX_BUFFER 4
54  #define SIZE_OF_BUFFER 1520 /* 16 バイトの整数倍にすること */
55
56  #define OPEN_OK 0
57  #define OPEN_NG -1
58  #define SEND_OK 0
59  #define SEND_NG -1
60  #define CLOSE_OK 0
61  #define CLOSE_NG -1
62  #define MIN_FRAME_SIZE 60
63  #define MAX_FRAME_SIZE 1514
64
65  #define EDMAC_EESIPR_INI_SEND 0x44080F00 /* 0x40000000 : ライトバック完了 *
66                                          * 0x04000000 : 送信中断検出 *
67                                          * ->未使用 0x00200000 : フレーム送信完了*
68                                          * 0x00200000 : フレーム送信完了 *
69                                          * 0x00080000 : 送信 FIFO アンダフロー*
70                                          * 0x00000800 : キャリア未検出 *
71                                          * 0x00000400 : キャリア消失検出 *
72                                          * 0x00000200 : 遅延衝突検出 *
73                                          * 0x00000100 : 送信リトライオーバ */
74  #define EDMAC_EESIPR_INI_RECV 0x0205001F /* 0x02000000 : 受信中断検出 *
75                                          * 0x00040000 : フレーム受信 *
76                                          * 0x00010000 : 受信 FIFO オーバフロー *
77                                          * 0x00000010 : 端数ビットフレーム受信 *
78                                          * 0x00000008 : ロングフレーム受信 *
79                                          * 0x00000004 : ショートフレーム受信 *
80                                          * 0x00000002 : PHY-LSI 受信エラー *
81                                          * 0x00000001 : 受信フレーム CRC エラー */
82  #define EDMAC_EESIPR_INI_EtherC 0x00400000 /* 0x00400000 : EtherC ステータスレジスタ*/
83  #define EtherC_ECSIPR_INI 0x00000004 /* 0x00000004 : リンク信号変化 */
84
85  /* **** 型定義 **** */
86
87  /* ==== 送信ディスクリプタ ==== */
88  typedef union{
89      unsigned long LONG;
90      struct{
91          unsigned int TACT:1; /* 送信ディスクリプタ有効 */
92          unsigned int TDLE:1; /* 送信ディスクリプタ最終 */
93          unsigned int TFP :2; /* 送信フレーム内位置 1,0 */
94          unsigned int TFE :1; /* 送信フレームエラー */
95          unsigned int reserved :23; /* TFS26~4 : 予約 */
96          unsigned int TFS3:1; /* ノーキャリア検出 (EESR-CND ビット) */
97          unsigned int TFS2:1; /* キャリア消失検出 (EESR-DLC ビット) */
98          unsigned int TFS1:1; /* 送信中の遅延衝突検出 (EESR-CD ビット) */
99          unsigned int TFS0:1; /* 送信リトライオーバ (EESR-TRO ビット) */
100     }BIT;
101     }TD0;

```

## 3.18 サンプルプログラムリスト” ether.h”(3)

```

102 typedef struct{
103     unsigned short TDL;                /* 送信バッファデータ長          */
104     unsigned short reserved;
105 }TD1;
106 typedef struct{
107     unsigned char *TBA;                /* 送信バッファのアドレス        */
108 }TD2;
109 typedef struct tag_edmac_send_desc{
110     TD0 td0;
111     TD1 td1;
112     TD2 td2;
113     struct tag_edmac_send_desc *pNext;
114 }EDMAC_SEND_DESC;
115
116 /* ==== 受信ディスクリプタ ==== */
117 typedef union{
118     unsigned long LONG;
119     struct{
120         unsigned int RACT:1;          /* 受信ディスクリプタ有効          */
121         unsigned int RDLE:1;          /* 受信ディスクリプタ最終          */
122         unsigned int RFP :2;          /* 受信フレーム内位置 1,0          */
123         unsigned int RFE :1;          /* 受信フレームエラー              */
124         unsigned int reserved1:17;    /* TFS26~10 : 予約                  */
125         unsigned int RFS9:1;          /* 受信 FIFO オーバフロー (EESR-RFOF ビット) */
126         unsigned int reserved2:1;     /*          : 予約                    */
127         unsigned int RFS7:1;          /* マルチキャストフレームを受信 (EESR-RMAF ビット) */
128         unsigned int reserved3:1;     /*          : 予約                    */
129         unsigned int reserved4:1;     /*          : 予約                    */
130         unsigned int RFS4:1;          /* 端数ビットフレーム受信エラー (EESR-RRF ビット) */
131         unsigned int RFS3:1;          /* ロングフレーム受信エラー (EESR-RTLE ビット) */
132         unsigned int RFS2:1;          /* ショートフレーム受信エラー (EESR-RTSF ビット) */
133         unsigned int RFS1:1;          /* PHY-LSI 受信エラー (EESR-PRE ビット) */
134         unsigned int RFS0:1;          /* 受信フレーム CRC エラー検出 (EESR-CERF ビット) */
135     }BIT;
136 }RD0;
137 typedef struct{
138     unsigned short RBL;                /* 受信バッファ長                  */
139     unsigned short RDL;                /* 受信データ長                    */
140 }RD1;
141 typedef struct{
142     unsigned char *RBA;                /* 受信バッファのアドレス          */
143 }RD2;
144 typedef struct tag_edmac_recv_desc{
145     RD0 rd0;
146     RD1 rd1;
147     RD2 rd2;
148     struct tag_edmac_recv_desc *pNext;
149 }EDMAC_RECV_DESC;

```

## 3.19 サンプルプログラムリスト” ether.h”(4)

```
150
151 /* ==== 送受信ディスクリプタ全体(16バイト境界に配置すること) ==== */
152 typedef struct{
153     EDMAC_SEND_DESC send[NUM_OF_TX_DESCRIPTOR];
154     EDMAC_RECV_DESC recv[NUM_OF_RX_DESCRIPTOR];
155     EDMAC_SEND_DESC *pSend_top;          /* 送信ディスクリプタの登録位置 */
156     EDMAC_RECV_DESC *pRecv_end;        /* 受信ディスクリプタの受信完了/登録位置 */
157 }TXRX_DESCRIPTOR_SET;
158
159 /* ==== 送受信バッファ(16バイト境界に配置すること) ==== */
160 /* ---- 全送受信バッファ領域の定義 ---- */
161 typedef struct{
162     unsigned char send[NUM_OF_TX_BUFFER][SIZE_OF_BUFFER];
163     unsigned char recv[NUM_OF_RX_BUFFER][SIZE_OF_BUFFER];
164 }TXRX_BUFFER_SET;
165
166 /* **** プロトタイプ宣言 **** */
167 int lan_open( void );
168 int lan_close( void );
169 int lan_send( unsigned char *addr, int flen );
170
171
172 #endif
173
174 /* End of File */
```

## 3.20 サンプルプログラムリスト”intprg\_eth.c”(1)

```
1  /*****
2  *  DISCLAIMER
3  *
4  *  This software is supplied by Renesas Electronics Corporation and is only
5  *  intended for use with Renesas products. No other uses are authorized.
6  *
7  *  This software is owned by Renesas Electronics Corporation and is protected under
8  *  all applicable laws, including copyright laws.
9  *
10 *  THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 *  REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 *  INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 *  PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 *  DISCLAIMED.
15 *
16 *  TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 *  ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 *  FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 *  FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 *  AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 *
22 *  Renesas reserves the right, without notice, to make changes to this
23 *  software and to discontinue the availability of this software.
24 *  By using this software, you agree to the additional terms and
25 *  conditions found by accessing the following link:
26 *  http://www.renesas.com/disclaimer
27 *****/
28 * Copyright (C) 2007(2010) Renesas Electronics Corporation. All rights reserved.
29 * "FILE COMMENT" ***** Technical reference data *****
30 * System Name : SH7671 Sample Program
31 * File Name   : intprg_eth.c
32 * Abstract    : interrupt entry function
33 * Version     : 1.01.01
34 * Device      : SH7671
35 * Tool-Chain  : High-performance Embedded Workshop (Ver.4.03.00).
36 *              : C/C++ compiler package for the SuperH RISC engine family
37 *              :                               (Ver.9.01 Release01).
38 * OS          : None
39 * H/W Platform: M3A-HS71(CPU board)
40 * Description :
41 *****/
42 * History     : Jul.04,2007 ver.1.00.00
43 *              : Sep.18,2007 ver.1.01.00 セクション変更 P->PIntPRG
44 *              : Oct.08,2010 ver.1.01.01 Changed the company name and device name
45 * "FILE COMMENT END" *****/
```

(途中省略)

## 3.21 サンプルプログラムリスト”intprg\_eth.c”(2)

```
654 // 171 ETC EINT0
655 void INT_ETC_EINT0(void)
656 {
657     unsigned long stat_edmac;
658     unsigned long stat_EtherC;
659
660     /* ---- 割込み要求フラグのクリア ---- */
661     stat_edmac = EDMAC.EESR.LONG & EDMAC.EESIPR.LONG; /* 許可した割込みのみ対象 */
662     EDMAC.EESR.LONG = stat_edmac;
663     /* ==== 送信関連 ==== */
664     if( stat_edmac & EDMAC_EESIPR_INI_SEND ){
665         lan_send_handler( stat_edmac & EDMAC_EESIPR_INI_SEND );
666     }
667     /* ==== 受信関連 ==== */
668     if( stat_edmac & EDMAC_EESIPR_INI_RECV ){
669         lan_recv_handler( stat_edmac & EDMAC_EESIPR_INI_RECV );
670     }
671     /* ==== EtherC 関連 ==== */
672     if( stat_edmac & EDMAC_EESIPR_INI_EtherC ){
673         /* ---- 割込み要求フラグのクリア ---- */
674         stat_EtherC = EtherC.ECSR.LONG & EtherC.ECSIPR.LONG; /* 許可した割込みのみ対象 */
675         EtherC.ECSR.LONG = stat_EtherC;
676         lan_etherc_handler( stat_EtherC );
677     }
678 }
```

(以下省略)

#### 4. 参考ドキュメント

- ソフトウェアマニュアル  
SH-2A/SH-2A-FPU ソフトウェアマニュアル Rev3.00  
(最新版をルネサス エレクトロニクスホームページから入手してください。)
- ハードウェアマニュアル  
SH7670 グループ ハードウェアマニュアル Rev.2.00  
(最新版をルネサス エレクトロニクスホームページから入手してください。)

## ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

<http://japan.renesas.com/>

お問合せ先

<http://japan.renesas.com/inquiry>

すべての商標および登録商標は、それぞれの所有者に帰属します。

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2008.02.28	—	初版発行
1.01	2010.10.15	—	フォーマット変更 参考プログラムの修正（AC 特性切り替え処理を削除）

## 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、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

### 1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

### 2. 電源投入時の処置

【注意】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. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレスがあります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

### 4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。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。

### 5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違っていると、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。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。

## ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連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の侵害等に関し、当社は、一切その責任を負いません。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。
3. 当社製品を改造、改変、複製等しないでください。
4. 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。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続きを行ってください。本資料に記載されている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の目的で使用しないでください。また、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に使用することができません。
6. 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。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されてい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生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。  
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  
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、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）  
特定水準： 航空機器、航空宇宙機器、海中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社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。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、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、社会的損害など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、機器またはシステムとしての出荷保証をお願いいたします。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、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、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、当社は、一切その責任を負いません。
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固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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と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。



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・電話番号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 エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

(03)5201-5307

■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/inquiry>