

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願い申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

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、誤りが無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、海底中継機器、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。

SH7619 グループ

イーサネットコンプライアンステスト用プログラム例

要旨

本アプリケーションノートは、SH7619 を使用したボードにおいて、イーサネットコンプライアンステストを実施するために必要なプログラム例を説明します。

動作確認デバイス

SH7619

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.....	14
4. 参考ドキュメント.....	29

1. はじめに

1.1 仕様

10BASE-T または 100BASE-TX のイーサネットコンプライアンステストを実施するために必要な送信波形を SH7619 から出力します。

テストパラメータの操作に RS-232C とスイッチを使用して、テストに対応する設定変更を簡単に行うことができます。SH7619 の波形調整レジスタの設定値も変更できます。

1.2 使用機能

- イーサネットフィジカルレイヤトランシーバ (PHY)
- PHY インタフェース (PHY-IF)
- ピンファンクションコントローラ (PFC)
- 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)
- FIFO 付きシリアルコミュニケーションインタフェース (SCIF)
- I/O ポート

1.3 適用条件

マイコン	SH7619
動作周波数	内部クロック : 125 MHz バスクロック : 62.5 MHz 周辺クロック : 31.25 MHz
統合開発環境	ルネサステクノロジ製 High-performance Embedded Workshop Ver.4.03.00
C コンパイラ	ルネサステクノロジ製 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=sh2 -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。

- SH7619 グループ 初期設定例
- SH7619 グループ 内蔵 PHY モジュールの初期設定例
- SH7619 グループ 内蔵 PHY モジュール PHY モード設定例
- SH7619 グループ 内蔵 PHY モジュール 波形調整レジスタ設定例
- SH7619 グループ イーサネット送信設定例
- SH7619 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、PHY モードの設定値、テストプログラムの種類および波形調整レジスタの設定値を RS-232C 経由で入力します。入力した PHY モードと波形調整レジスタ値を使用して、内蔵 PHY モジュール、EtherC および E-DMAC を初期化します。その後、テストプログラムに応じたデータを繰り返し送信します。データ送信の開始と停止は、スイッチによって制御します。

以下にイーサネット物理層コンプライアンステストの概要を示します。コンプライアンステスト用測定器の設定については、ご利用になる測定機器メーカーへお問い合わせください。

2.1 100BASE-TX コンプライアンステストの概要

100BASE-TXコンプライアンステストにおけるテスト項目一覧と必要なテストパターン（送信プログラム）を表 1 に示します。100BASE-TX ではアイドル状態（送信待ち状態）時も信号が出力されるため、本 LSI 側に特別なテストパターンは必要ありません。

表1 100BASE-TX コンプライアンステスト

テスト項目		説明	テストパターン
テンプレート		送信波形がテンプレートに対してパスすることを確認する。	IDLE ^{※2}
振幅	差動電圧出力	送信波形の正負差動電圧の振幅が規定の範囲内であることを確認する。	IDLE ^{※2}
	オーバシュート	送信波形のオーバシュートが規定の範囲内であることを確認する。	IDLE ^{※2}
	振幅シンメトリ	送信波形の正負差動電圧の比が規定の範囲内であることを確認する。	IDLE ^{※2}
時間	立ち上がり時間および立ち下がり時間	送信波形の立ち上がり時間および立ち下がり時間が規定の範囲内であることを確認する。	IDLE ^{※2}
	立ち上がり時間および立ち下がりシンメトリ	送信波形の立ち上がり時間および立ち下がり時間の比が規定の範囲内であることを確認する。	IDLE ^{※2}
ジッタ		送信波形のジッタが規定の範囲内であることを確認する。	IDLE ^{※2}
デューティ・サイクル歪み		送信波形の立ち上がり、立ち下がりの 50%ポイントでエッジのずれが規定の範囲内であることを確認する。	IDLE ^{※2}
リターン・ロス ^{※1}	送信	Tx における入射信号と反射信号の比が規定の範囲内であることを確認する。	—
	受信	Rx における入射信号と反射信号の比が規定の範囲内であることを確認する。	—

【注】 ^{※1} 本応用例では対応しません。

^{※2} アイドル時の波形を利用して測定を行うテストパターンです。

2.2 10BASE-T コンプライアンステストの概要

10BASE-Tコンプライアンステストにおけるテスト項目一覧と必要なテストパターン（送信プログラム）を表2に示します。10BASE-Tではテスト項目に応じて、本LSI側で実行するテストパターンを切替える必要があります。

表2 10BASE-TX コンプライアンステスト

テスト項目	説明	テストパターン
テンプレート		
MAU	バースト的に送信される波形がテンプレートに対してパスすることを確認する。	擬似 RANDOM ^{※2}
TP_IDL	フレームの終わり(アイドル状態移行前)に送信される波形がテンプレートに対してパスすることを確認する。	擬似 RANDOM ^{※2}
リンクパルス	送信するリンク信号がテンプレートに対してパスすることを確認する。	IDLE ^{※3}
差動電圧	送信波形の正負差動電圧の振幅が規定の範囲内であることを確認する。	擬似 RANDOM ^{※2}
ジッタ	送信波形のジッタが規定の範囲内であることを確認する。	擬似 RANDOM ^{※2}
高調波	連続1または連続0送信における高調波ノイズが規定の範囲内であることを確認する。	ALL ONE ^{※4}
リターン・ロス ^{※1}	送信	Txにおける入射信号と反射信号の比が規定の範囲内であることを確認する。
	受信	Rxにおける入射信号と反射信号の比が規定の範囲内であることを確認する。
コモンモード電圧 ^{※1}	コモンモード波形のピーク電圧が規定の範囲内であることを確認する。	擬似 RANDOM ^{※2}

【注】 ^{※1} 本応用例では対応しません。

^{※2} ランダムなイーサネットフレームの送信を行うテストパターンです。

^{※3} イーサネットフレーム送信を行わず、アイドル時のリンクパルス出力を利用して測定を行うテストパターンです。

^{※4} 一定期間連続した"1" (H'FF) 送信を行うテストパターンです。

2.3 システム構成

図 1に本応用例のシステム構成を示します。

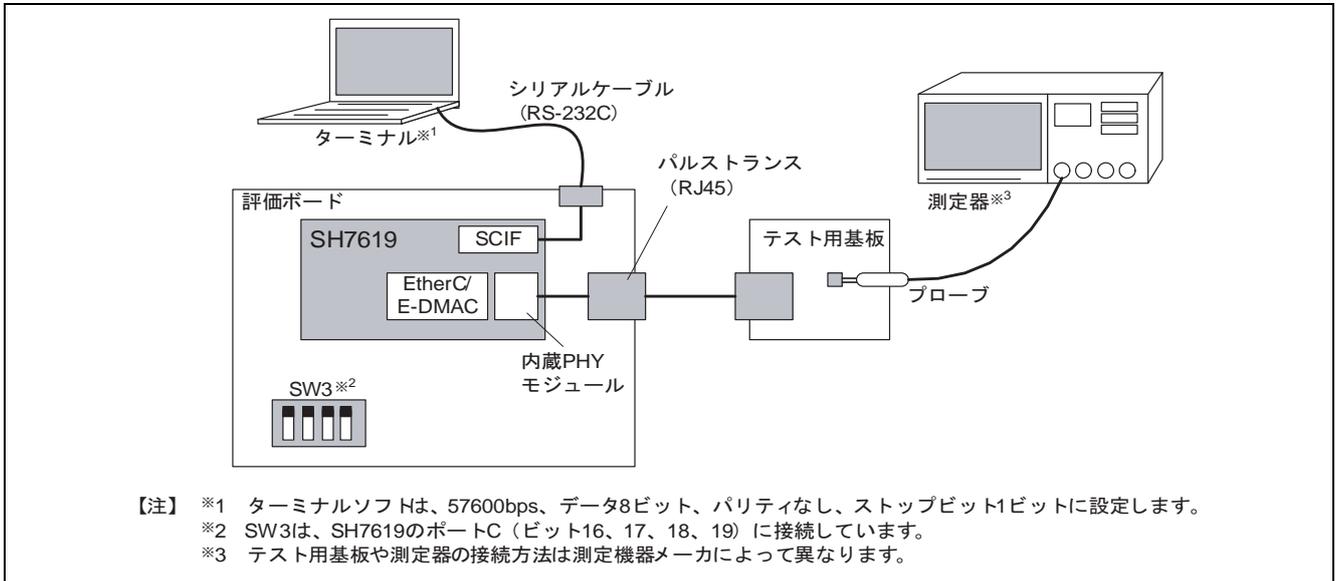


図1 システム構成

2.4 参考プログラムの説明

参考プログラムは

- main.c
- ether.c
- phy.c
- intprg_eth.c
- dbsct_eth.c

の5つのソースプログラムと、アプリケーションノート「SH7619 グループ SH7619 初期設定例」で作成した各初期化用ファイルから構成されています。

- main.c**
 メイン関数 (main関数) と、テストパラメータ設定関数 (set_param関数)、テストプログラム実行関数 (go_test関数) が記述されています。
 メイン関数は、スイッチ 3 (SW3) の状態に応じて、テストパラメータの設定またはテストプログラムの実行を行います。図 2にmain関数のフローを示します。テストパラメータ設定関数は、RS-232C経由でテストパラメータ (テストパターン、PHYモード、波形調整レジスタ設定値) を取得します。図 3にset_param関数のフローを示します。テストプログラム実行関数は、取得したテストパラメータを反映したイーサネットフレームの送信を行います。図 4にgo_test関数のフローを示します。
- ether.c**
 イーサネット初期化関数 (lan_open関数) や送信関数 (lan_send関数) などが記述されています。図 5にlan_open関数とlan_send関数のフローを示します。その他の関数の処理フローはアプリケーションノート「SH7619 グループ イーサネット送信設定例」または「SH7619 グループ イーサネット受信設定例」で解説しています。合わせて参照してください。
- phy.c**
 波形調整レジスタ設定関数 (phy_set_ampctrl関数)、PHYIFの初期化関数 (phyif_init関数)、交渉結果のリンクモードを取得する関数 (phy_read_linkmode関数) およびSMIレジスタアクセス関数群が記述されています。図 6にphy_set_ampctrl関数のフローを示します。図 7にphyif_init関数のフローを示します。図 8にphy_read_linkmode関数のフローを示します。SMIレジスタアクセス関数群については、アプリケーションノート「SH7619 グループ 内蔵PHYモジュールの初期設定例」を参照してください。
- intrpg_eth.c**
 イーサネット関連の割り込み関数 (INT_EDMAC_EINT0 関数) が記述されています。INT_EDMAC_EINT0関数のフローはアプリケーションノート「SH7619 グループ イーサネット送信設定例」または「SH7619 グループ イーサネット受信設定例」を参照してください。
- dbstc_eth.c**
 dbstc.cのメモリ配置をイーサネット送信用に変更したものです。アプリケーションノート「SH7619 グループ イーサネット送信設定例」または「SH7619 グループ イーサネット受信設定例」を参照してください。

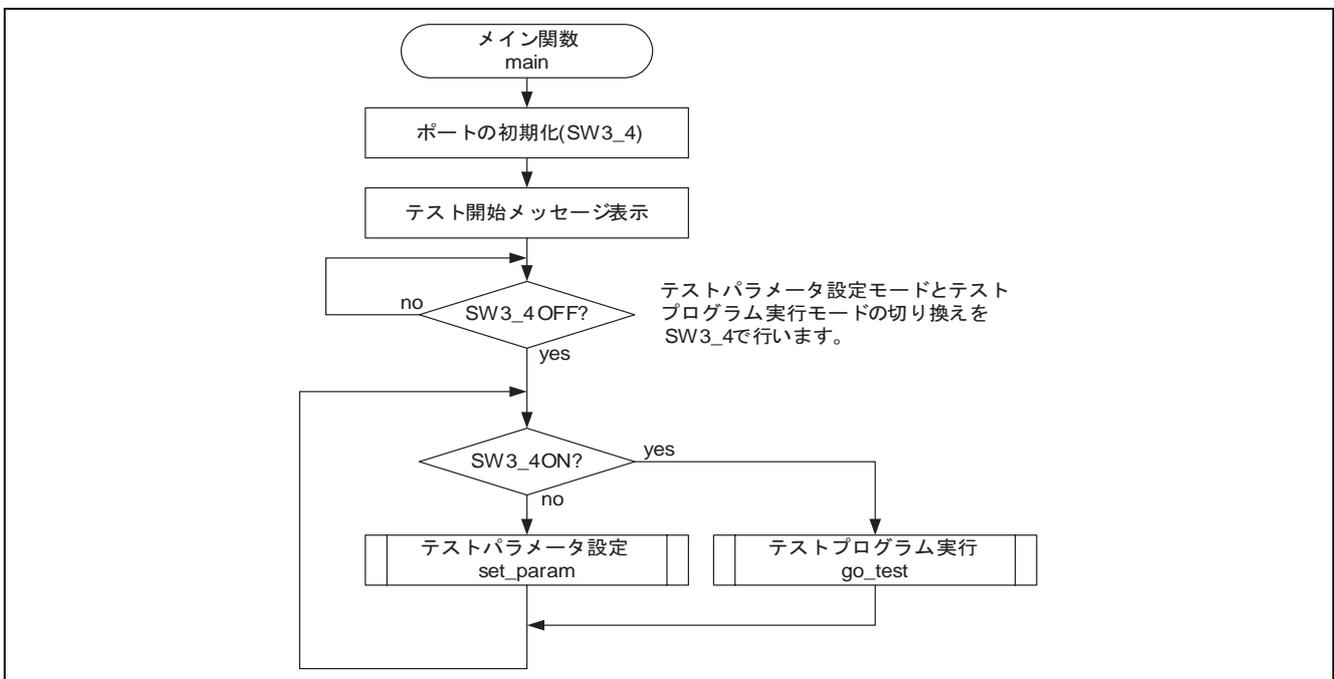


図2 メイン関数の処理フロー

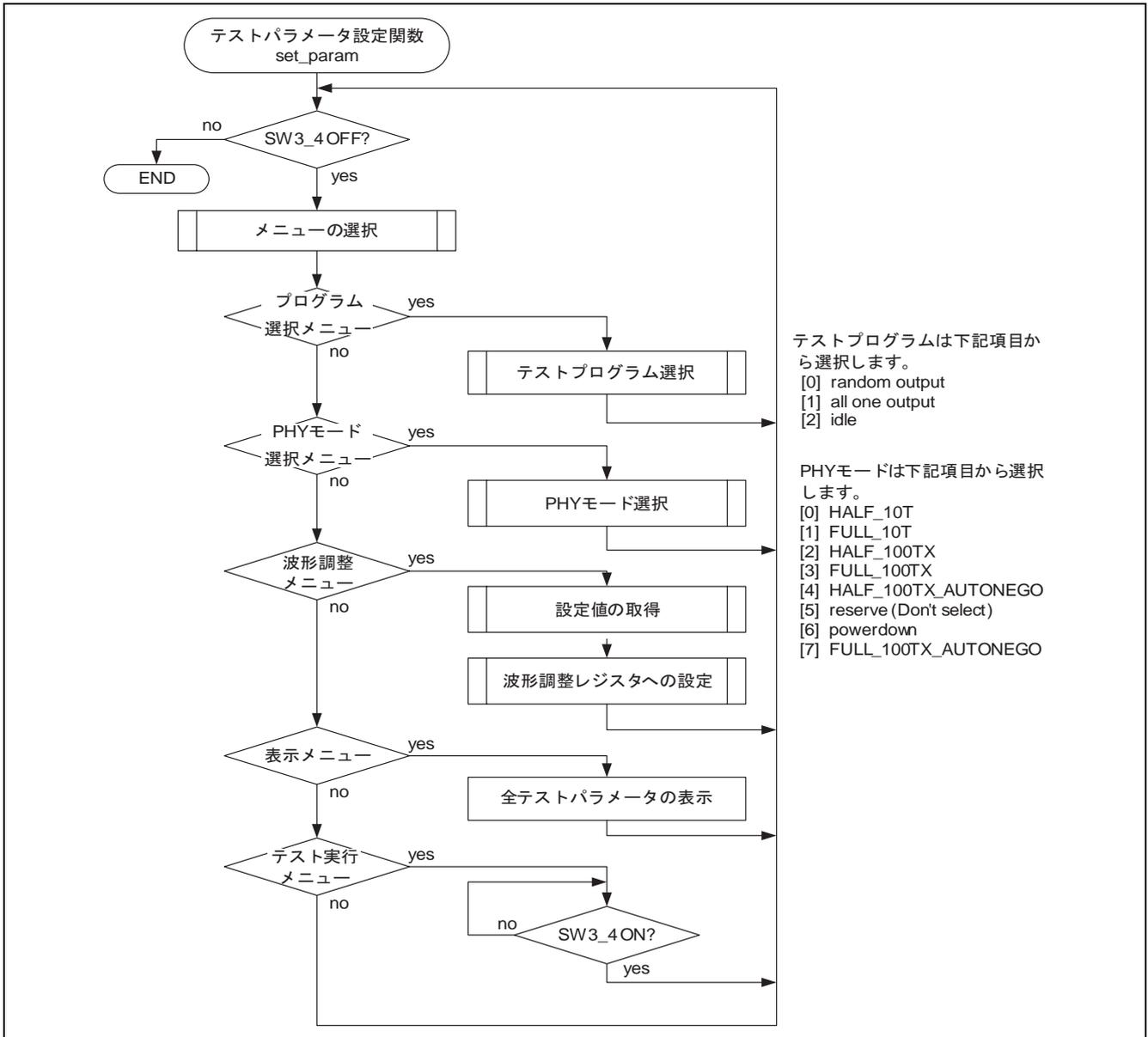


図3 テストパラメータ設定関数の処理フロー

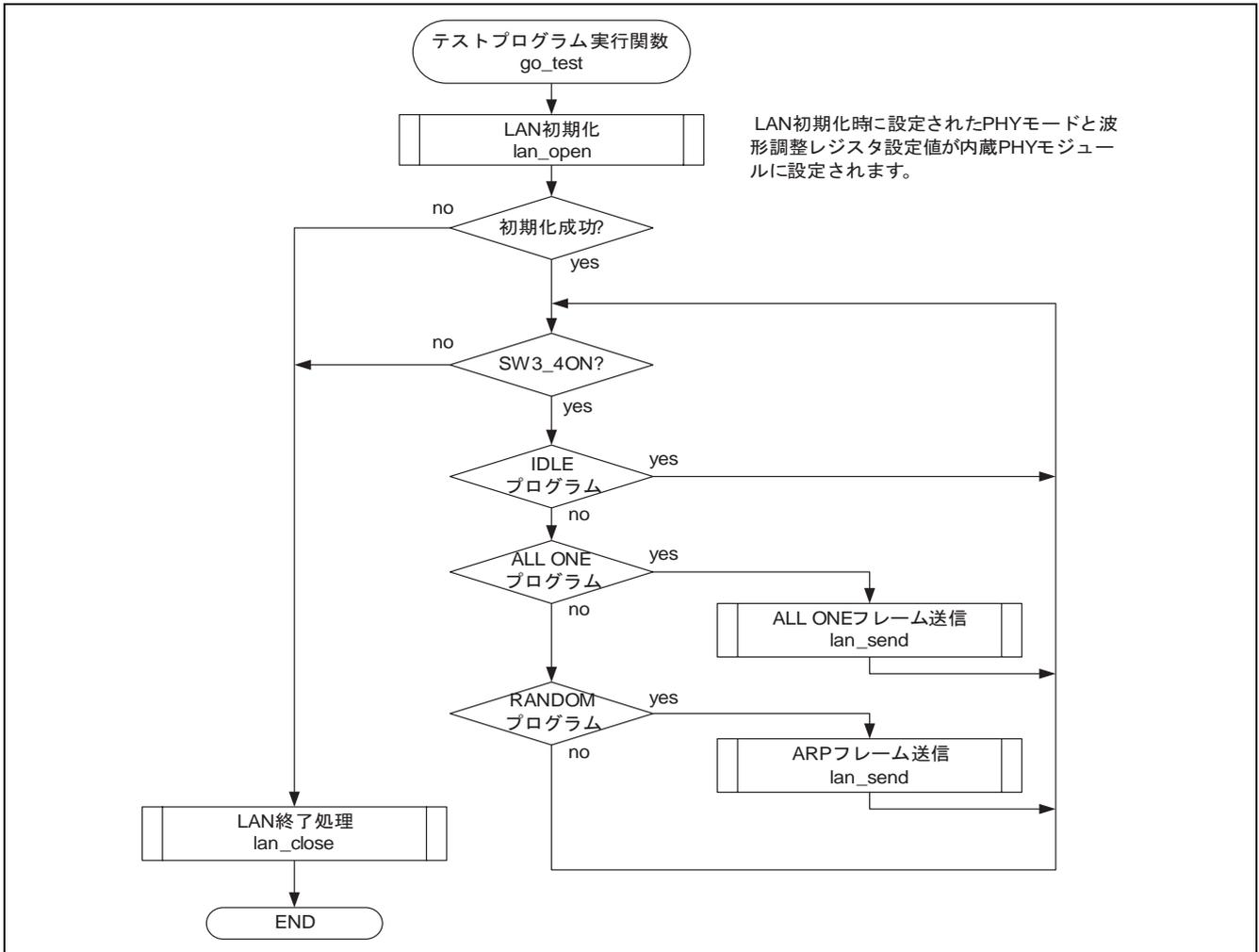


図4 テストプログラム実行関数の処理フロー

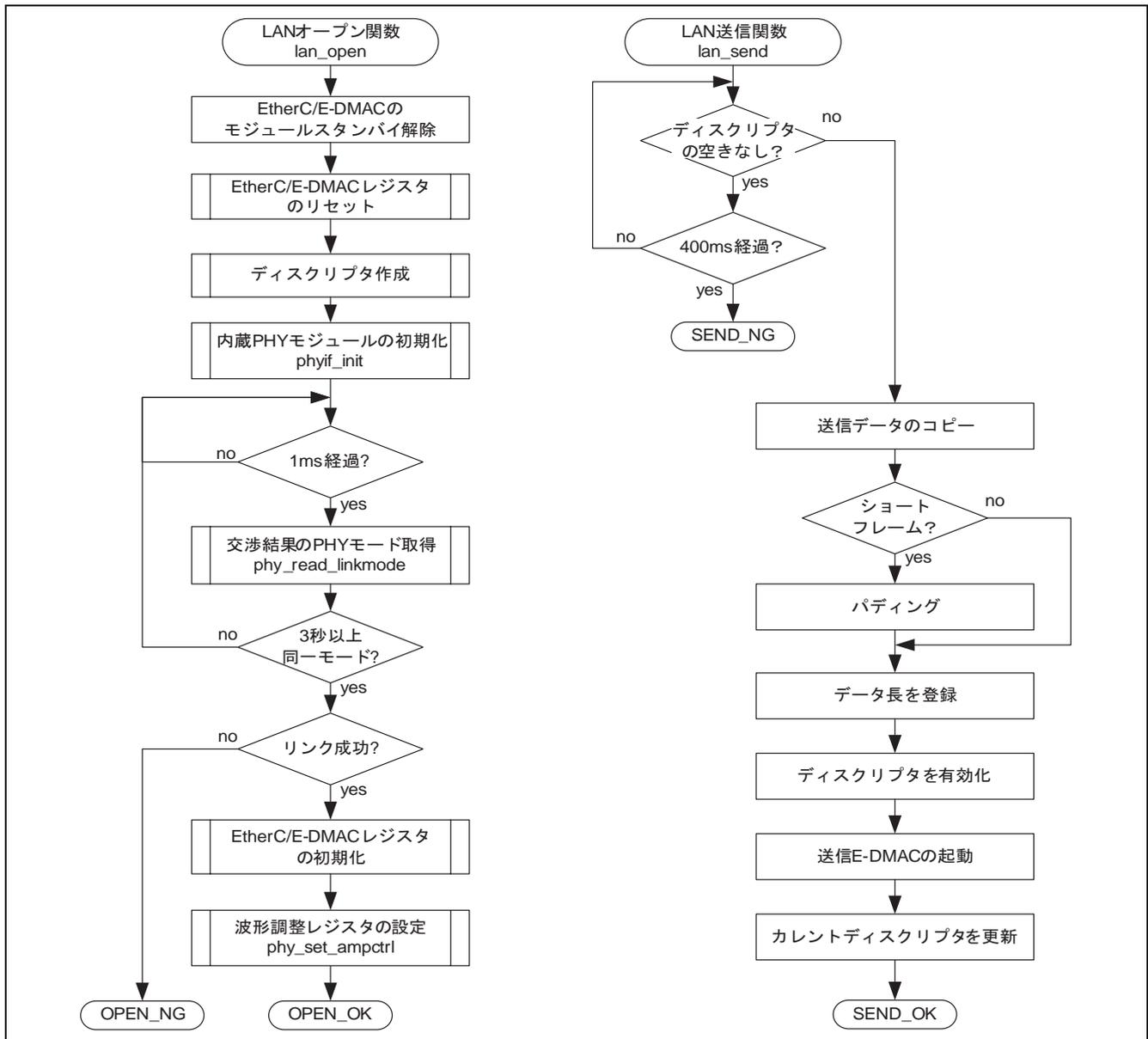


図5 LANオープン関数とLAN送信関数の処理フロー

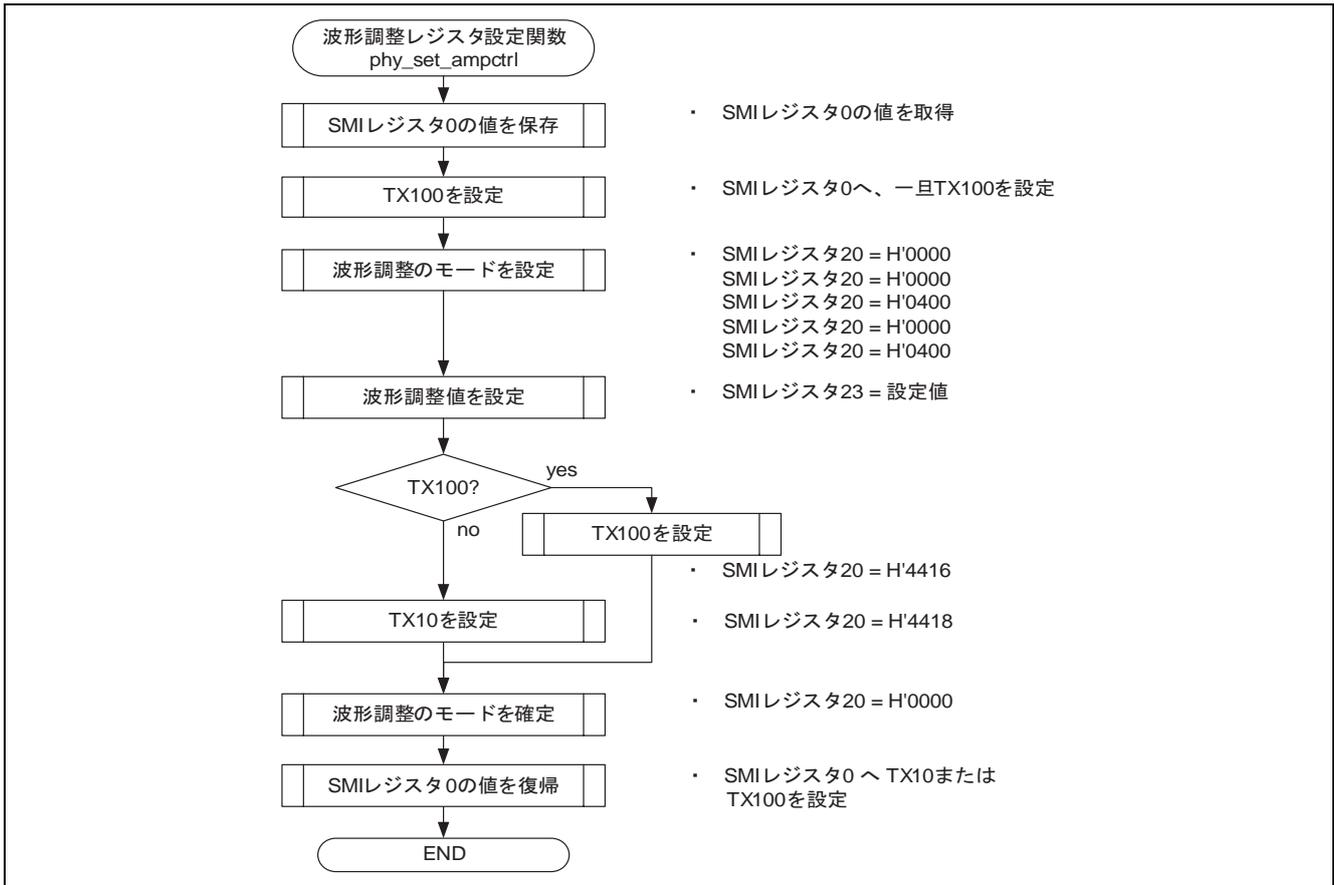


図6 波形調整レジスタ設定関数の処理フロー

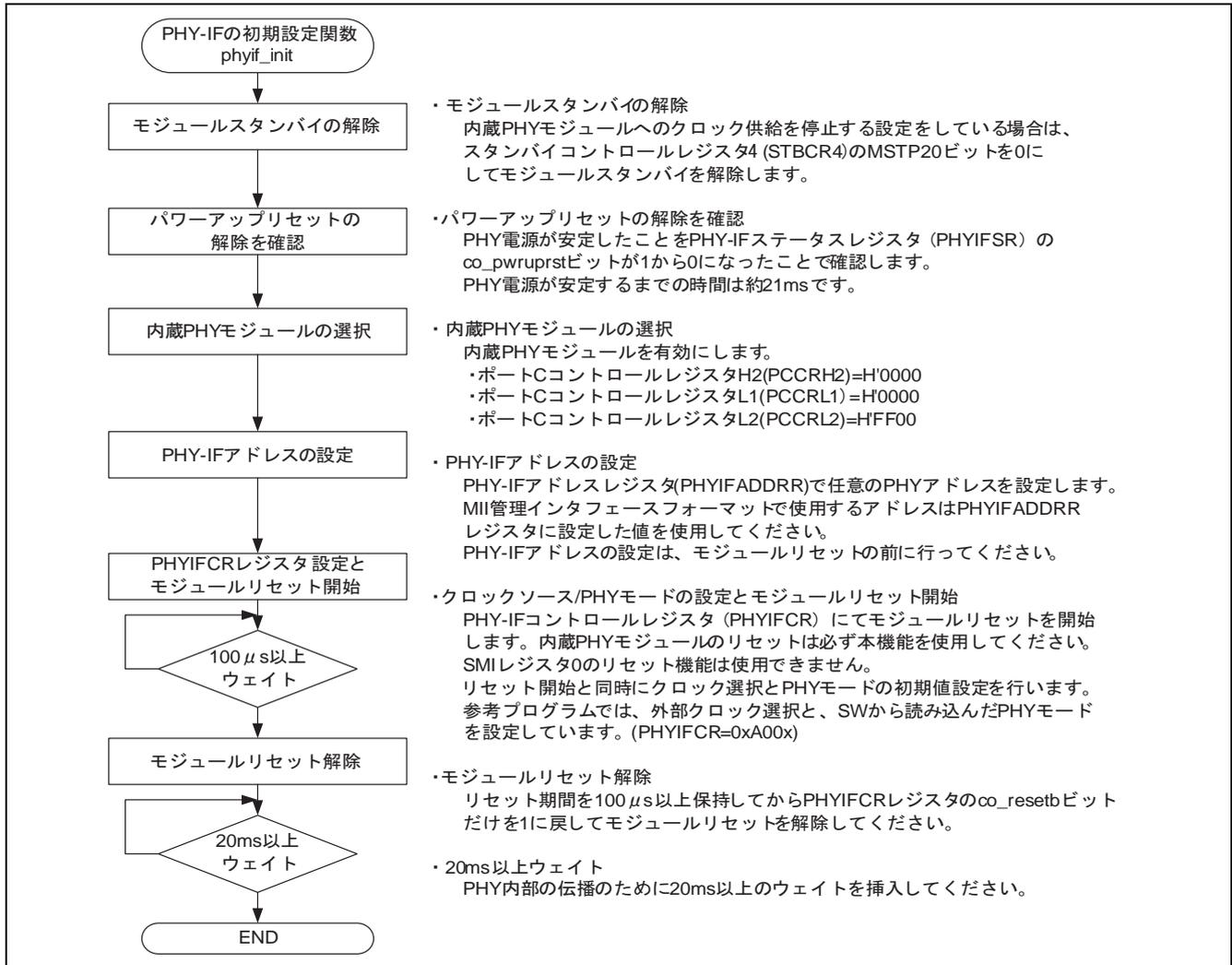


図7 PHYIF の初期設定関数の処理フロー

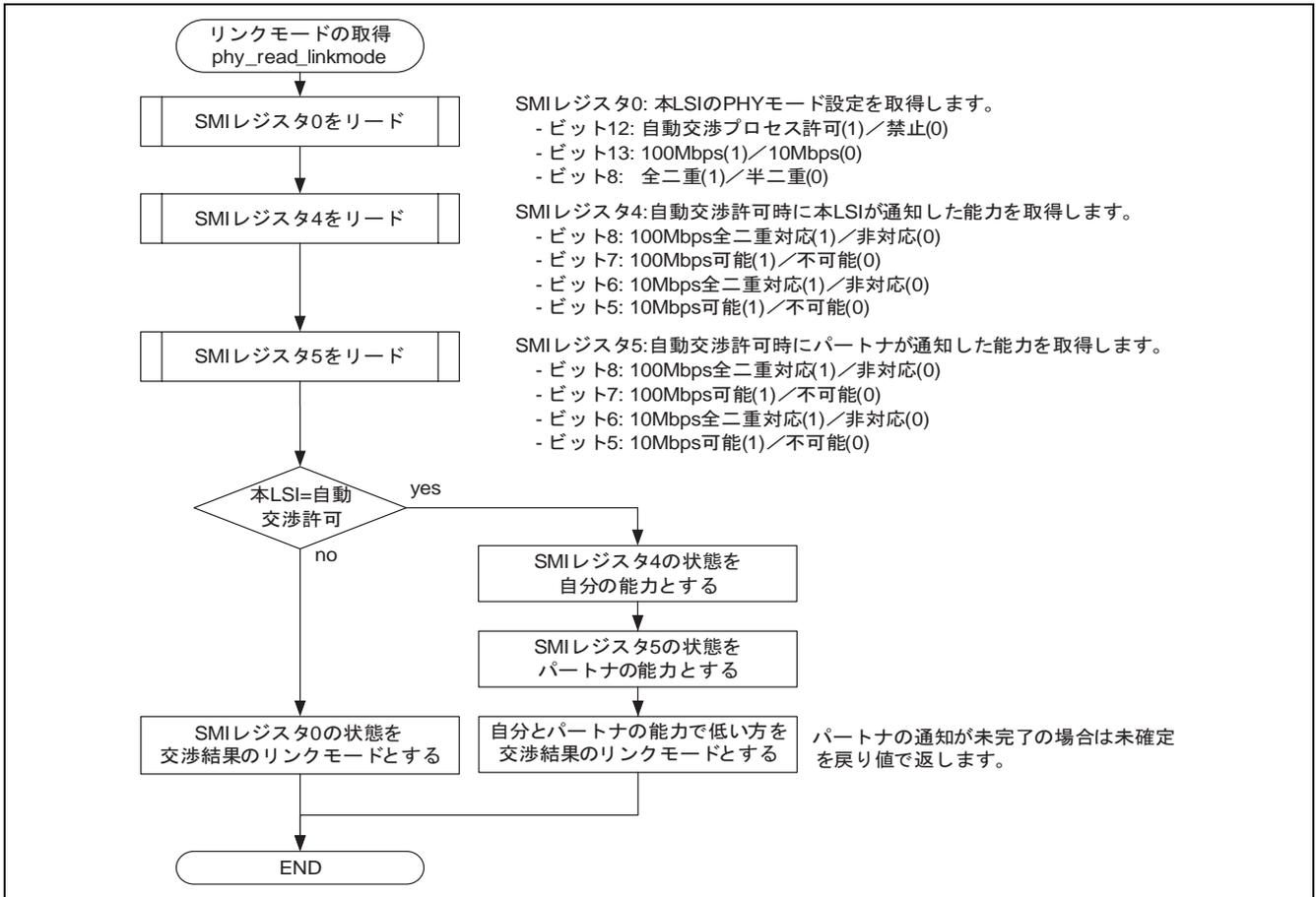


図8 リンクモードの取得関数の処理フロー

2.5 参考プログラムにおける設定内容

表 3に参考プログラムでの設定内容を示します。

表3 参考プログラムでの設定

項目	設定内容
EtherC	
送信許可／受信許可	送信許可（受信禁止）
フロー制御	未使用
CRC エラー制御	エラーありとして受信
MagicPacket	許可しない
内部ループバック	未使用
デュプレックスモード	確定したリンクモードから設定
プロミスキャスモード	未使用
割込み	リンク信号変化
MAC アドレス	00-01-02-03-04-05（必要に応じて、機器固有の MAC アドレスに変更してください。）
E-DMAC	
エンディアン変換	ビッグエンディアン
割込み	ライトバック完了割込み（送信）、受信完了割込み 他
ディスクリプタへのステータスコピー	全て反映
送信 FIFO しきい値	ストア&フォワード
送信 FIFO 容量	送信 512 バイト／受信 512 バイト
受信方式	連続受信
送信ディスクリプタ	16 バイト×4
受信ディスクリプタ	16 バイト×4
送受信バッファ	1520 バイト／ディスクリプタ（1 フレーム／1 ディスクリプタ）
内蔵 PHY モジュール	
使用する PHY	内蔵 PHY モジュール
クロックソース	外部クロック（25MHz）
PHY モード	ターミナルから指定
PHY アドレス	0
波形調整	TX100 および TX10 波形の出力調整
使用する SMI レジスタ	レジスタ 0（Basic Control） レジスタ 4（Auto-Negotiation Advertisement） レジスタ 5（Auto-Negotiation Link Partner Ability） レジスタ 20（モード変更用レジスタ） レジスタ 23（波形調整用レジスタ）
SCIF2(RS-232C)	
ボーレート	57600bps
データ	8bit
パリティ	なし
フロー制御	なし

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 Technology Corp. and is only
5  *   intended for use with Renesas products. No other uses are authorized.
6  *
7  *   This software is owned by Renesas Technology Corp. 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 *   TECHNOLOGY CORP. 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 THE 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) 2009. Renesas Technology Corp., All Rights Reserved.
29 * "FILE COMMENT"***** Technical reference data *****
30 *   System Name : SH7619 Sample Program
31 *   File Name   : main.c
32 *   Abstract    : イーサネットコンプライアンステスト用プログラム例
33 *   Version     : 1.00.00
34 *   Device      : SH7619
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-HS19 (cpu board)
40 *   Description :
41 *****/
42 *   History     : Apr.22,2009 Ver.1.00.00
43 * "FILE COMMENT END"*****/
    
```

3.2 サンプルプログラムリスト"main.c" (2)

```

44  #include <stdio.h>
45  #include "iodefine.h"
46  #include "defs.h"
47  #include "ether.h"
48  #include "phy.h"
49
50  /* **** マクロ定義 **** */
51  #define IS_MODESW_ON() ((~PC.DRH.WORD) & 0x0008u)/* sw3_4 */
52
53  /* ---- main menu ---- */
54  enum tag_main{
55      TEST_PROG=0, PHY_MODE, ADJUST_WAVE, VIEW, GO_TEST,
56      NUM_OF_MAIN_MSG
57  };
58  /* ---- program set menu ---- */
59  enum tag_prog{
60      RANDOM_OUTPUT=0, ALLONE_OUTPUT, IDLE,
61      NUM_OF_PROG_MSG
62  };
63  /* ---- phy mode set menu ---- */
64  enum tag_phynd{
65      HALF_10T=0, FULL_10T, HALF_100TX, FULL_100TX, HALF_100TX_AUTONEGO, RESERVE,
66      POWERDOWN, FULL_100TX_AUTONEGO,
67      NUM_OF_PHYND_MSG
68  };
69  /* ---- test parameter type ---- */
70  typedef struct{
71      int prog;
72      int phynd;
73      unsigned short wv_100;
74      unsigned short wv_10;
75  }TEST_TYPE;
76
77  /* **** 変数宣言 **** */
78  /* ---- random frame ---- */
79  static unsigned char s_frame[] = {
80      0xff,0xff,0xff,0xff,0xff,0xff, /* あて先 MAC アドレス */
81      0x00,0x01,0x02,0x03,0x04,0x05, /* 送信元 MAC アドレス (00:01:02:03:04:05)*/
82      0x08,0x06, /* タイプ(ARP) */
83      0x00,0x01, /* ---H/Wタイプ= Ethernet */
84      0x08,0x00, /* ---プロトコルタイプ= IP */
85      0x06,0x04, /* ---HW/プロトコルアドレス長 */
86      0x00,0x01, /* ---OPCODE= request */
87      0x00,0x01,0x02,0x03,0x04,0x05, /* ---送信元 MAC アドレス(00:01:02:03:04:05) */
88      0xc0,0xa8,0x00,0x03, /* ---送信元 IP アドレス(192.168.0.3) */
89      0x00,0x00,0x00,0x00,0x00,0x00, /* ---問合せ先 MAC アドレス */
90      0xc0,0xa8,0x00,0x05, /* ---問合せ先 IP アドレス(192.168.0.5) */
91  };
92  /* ---- all one frame ---- */
93  static unsigned char allone[1514];
94  /* ---- test parameter ---- */
95  TEST_TYPE test = {IDLE, FULL_100TX, 0x81C8, 0x4000};
    
```

3.3 サンプルプログラムリスト"main.c" (3)

```

96     /* ---- console message string ---- */
97     static char *str_prog[]={
98         "RANDOM_OUTPUT", "ALLONE_OUTPUT", "IDLE",
99     };
100    static char *str_phynd[]={
101        "HALF_10T", "FULL_10T", "HALF_100TX", "FULL_100TX", "HALF_100TX_AUTONEGO",
102        "RESERVE", "POWERDOWN", "FULL_100TX_AUTONEGO",
103    };
104    static char *main_msg={
105        "\n\n"
106        "*****\n"
107        "<MAIN MENU>\n"
108        "Please choose the number of the parameter.\n"
109        " [0] test program setting\n"
110        " [1] phy mode setting\n"
111        " [2] adjustment of waveform\n"
112        " [3] view\n"
113        " [4] go test\n"
114        "*****\n"
115        ">> "
116    };
117    static char *sub_msg[]={
118        /* sub_msg[0] */
119        "-----\n"
120        " <TEST PROGRAM Setting>\n"
121        " [0] random output\n"
122        " [1] all one output\n"
123        " [2] idle\n"
124        "-----\n"
125        ">> ",
126        /* sub_msg[1] */
127        "-----\n"
128        " <PHY MODE Setting>\n"
129        " [0] HALF_10T\n"
130        " [1] FULL_10T\n"
131        " [2] HALF_100TX\n"
132        " [3] FULL_100TX\n"
133        " [4] HALF_100TX_AUTONEGO\n"
134        " [5] reserve(Don't select)\n"
135        " [6] powerdown\n"
136        " [7] FULL_100TX_AUTONEGO\n"
137        "-----\n"
138        ">> ",
139        /* sub_msg[2] */
140        "-----\n"
141        " <ADJUSTMENT OF THE WAVEFORM>\n",
142        /* sub_msg[3] */
143        "-----\n"
144        " <VIEW>\n",
145        /* sub_msg[4] */
146        "-----\n"
147        " <GO TEST>\n",
148    };

```

3.4 サンプルプログラムリスト"main.c" (4)

```

149  /* **** プロトタイプ宣言 **** */
150  void main(void);
151  static void set_param(void);
152  static void get_phy_ampval(int link, unsigned short *ampval);
153  static void input_val(int *val, int max, char *msg);
154  static void go_test(void);
155
156  /*"FUNC COMMENT"*****
157  * ID      :
158  * Outline : イーサネットコンプライアンステスト用プログラム例 メイン関数
159  *-----
160  * Include : "iodefine.h", "ether.h"
161  *-----
162  * Declaration : void main(void);
163  *-----
164  * Description : RS-232C からの入出力と SW3_4 のレベルをユーザインタフェースとして
165  *              : テスト用のパラメータ(送信プログラム種別、PHY モード初期値、波形
166  *              : 調整レジスタ設定値)を設定します。
167  *              : テスト実行を設定すると、指定のパラメータに従って送信プログラムが
168  *              : 実行され、コンプライアンステストを実施することができます。
169  *              : SW3_4 の ON/OFF によってテストを停止し、パラメータを再設定できます。
170  *-----
171  * Argument  : void
172  *-----
173  * Return Value : void
174  *-----
175  * Note      : Note
176  *"FUNC COMMENT END"*****
177  void main(void)
178  {
179      int i,j;
180      /* ==== data set ==== */
181      for(i=0; i<sizeof(allone); i++){
182          allone[i]=0xff;
183      }
184      /* ==== sw3 initialize ==== */
185      PFC.PCCR2.BYTE.L = 0x00;          /* PC16-19 端子を入出力ポートに設定 */
186      PFC.PCIORH.WORD &= 0xffff;      /* PC16-19 を入力ポートに設定 */
187
188      /* ==== test parameter initialize ==== */
189      printf("¥n¥n");
190      printf("=====¥n");
191      printf(" SH7619 Ethernet Compliance test program start! ¥n");
192      printf(" SW3_4 = ON   :Test program execution mode¥n");
193      printf(" SW3_4 = OFF  :Parameter setting mode¥n");
194      printf(" Please set SW3_4 to turning off. >> ");
195      fflush(stdout);
196      while( IS_MODESW_ON() ){
197          ;
198      }
199      printf("OK!¥n");
200      printf("=====¥n");
201      fflush(stdout);

```

3.5 サンプルプログラムリスト"main.c" (5)

```

202
203     /* ==== program start ==== */
204     while(1){
205         if( !IS_MODESW_ON() ){
206             set_param();
207         }
208         else{
209             go_test();
210         }
211     }
212 }
213
214 /*"FUNC COMMENT"*****
215 * ID      :
216 * Outline  : テストパラメータ設定
217 *-----
218 * Include  : <stdio.h>
219 *-----
220 * Declaration : static void set_param(void);
221 *-----
222 * Description : テスト用パラメータを設定します。
223 *              : メインメニューを RS-232C ターミナルに表示し、設定番号をユーザから
224 *              : 取得します。取得した番号に従ってサブメニューを表示しパラメータを
225 *              : ユーザから取得します。
226 *              : -テストプログラム(test program)が指定された場合は、実行する送信
227 *              : プログラムを選択します。
228 *              : -PHY モード(phy mode)が指定された場合は、PHY モードの初期設定値を
229 *              : 選択します。
230 *              : -波形調整(adjust wave)が指定された場合は、SMI レジスタ 23 に設定する
231 *              : 波形調整設定値を指定します。
232 *              : -表示(view)が指定された場合は、現在の設定値を表示します。
233 *              : -実行(go test)が指定された場合は、現在の設定値を表示し SW3_4 が
234 *              : ON になるのを待って本関数を終了します。
235 *-----
236 * Argument  : void
237 *-----
238 * Return Value : void
239 *-----
240 * Note      : Note
241 *"FUNC COMMENT END"*****

```

3.6 サンプルプログラムリスト"main.c" (6)

```

242 static void set_param(void)
243 {
244     int cmd;
245     int md;
246
247     while( !IS_MODESW_ON() ){
248         /* ==== input parameter ==== */
249         input_val(&cmd, NUM_OF_MAIN_MSG-1, main_msg);
250         switch(cmd){
251             case TEST_PROG:
252                 input_val(&test.prog, NUM_OF_PROG_MSG-1, sub_msg[TEST_PROG]);
253                 break;
254             case PHY_MODE:
255                 input_val(&test.phymd, NUM_OF_PHYMD_MSG-1, sub_msg[PHY_MODE]);
256                 break;
257             case ADJUST_WAVE:
258                 printf(sub_msg[ADJUST_WAVE]);
259                 get_phy_ampval(HALF_TX, &test.wv_100);
260                 get_phy_ampval(HALF_10M, &test.wv_10);
261                 break;
262             case VIEW:
263                 printf(sub_msg[VIEW]);
264                 printf("  This is a present setting.¥n");
265                 printf("    Program          -> %s¥n", str_prog[test.prog]);
266                 printf("    Phy mode          -> %s¥n", str_phymd[test.phymd]);
267                 printf("    Tx100 wave adjust -> 0x%04x¥n", test.wv_100);
268                 printf("    Tx10  wave adjust -> 0x%04x¥n", test.wv_10);
269                 fflush(stdout);
270                 break;
271             case GO_TEST:
272                 printf(sub_msg[GO_TEST]);
273                 printf("    %s, %s, (Tx100)=0x%04x, (Tx10)=0x%04x¥n",
274                     str_prog[test.prog], str_phymd[test.phymd], test.wv_100, test.wv_10);
275                 printf("    Please set SW3_4 to turning on. >> ");
276                 fflush(stdout);
277                 while( !IS_MODESW_ON() ){
278                     ;
279                 }
280                 printf("OK!¥n");
281                 fflush(stdout);
282                 break;
283             default:
284                 break;
285         }
286     }
287 }

```

3.7 サンプルプログラムリスト"main.c" (7)

```

288  /*"FUNC COMMENT"*****
289  * ID      :
290  * Outline : 波形調整レジスタ設定値の入力
291  *-----
292  * Include : <stdio.h>
293  *-----
294  * Declaration : static void get_phy_ampval(int link, unsigned short *ampval);
295  *-----
296  * Description : RS-232 経由でターミナルから波形調整レジスタの設定値を入力します。
297  *-----
298  * Argument  : int link          ; I : リンクモード(phy_read_linkmode()の戻り値)
299  *            : unsigned short *ampval ; 0 : 設定値を格納するバッファのアドレス
300  *-----
301  * Return Value : void
302  *-----
303  * Note       : Note
304  *"FUNC COMMENT END"*****
305  static void get_phy_ampval(int link, unsigned short *ampval)
306  {
307      int tmp[3];
308
309      printf("\n");
310      printf("=====\n");
311
312      /* ---- 100TX ---- */
313      if( link == FULL_TX || link == HALF_TX ){
314          printf("The value of SMI registerr 23 (for 100TX)\n");
315          input_val(&tmp[0], 3, " Please input the value of DnCMP.[0-3] > "); /* DnCMP */
316          input_val(&tmp[1], 7, " Please input the value of DnA. [0-7] > "); /* DnA */
317          input_val(&tmp[2], 3, " Please input the value of DnSL. [0-3] > "); /* DnSL */
318          printf("OK! (DnCMP=%d, DnA=%d, DnSL=%d)\n", tmp[0], tmp[1], tmp[2]);
319          *ampval = 0x8000 | (tmp[0] << 7) | (tmp[1] << 4) | (tmp[2] << 2);
320      }
321      /* ---- 10T ---- */
322      else{
323          printf("The value of SMI registerr 23 (for 10T)\n");
324          input_val(&tmp[0], 3, " Please input the value of DnTAMP.[0-3] > "); /* DnTAMP */
325          input_val(&tmp[1], 3, " Please input the value of DnTCMP.[0-3] > "); /* DnTCMP */
326          printf("OK! (DnTAMP=%d, DnTCMP=%d)\n", tmp[0], tmp[1]);
327          *ampval = (tmp[0] << 14) | (tmp[1] << 12);
328      }
329      printf("=====\n");
330      fflush(stdout);
331  }
    
```

3.8 サンプルプログラムリスト"main.c" (8)

```

332  /*"FUNC COMMENT"*****
333  * ID      :
334  * Outline : ユーザインタフェース(数値入力)
335  *-----
336  * Include : <stdio.h>
337  *-----
338  * Declaration : static void input_val(int *val, int max, char *msg);
339  *-----
340  * Description : ターミナルに入力メッセージを表示し、0 から指定された範囲内の数値
341  *              : を入力します。
342  *-----
343  * Argument  : int *val ; 0 : 入力した数値の格納先
344  *              : int max ; I : 入力する数値の最大値
345  *              : char *msg; I : 入力メッセージ
346  *-----
347  * Return Value : void
348  *-----
349  * Note      : 数値以外の入力には対応していません。
350  /*"FUNC COMMENT END"*****/
351  static void input_val(int *val, int max, char *msg)
352  {
353      int n;
354
355      while(1){
356          printf(msg);
357          fflush(stdout);
358          scanf(" %d",&n);
359          if( 0 <= n && n <= max ){
360              *val = n;
361              break;
362          }
363          else{
364              printf(" ....illegal value!¥n");
365          }
366      }
367  }
    
```

3.9 サンプルプログラムリスト"main.c" (9)

```

368  /*"FUNC COMMENT"*****
369  * ID      :
370  * Outline : テストプログラムの実行
371  *-----
372  * Include : <stdio.h>, "ether.h"
373  *-----
374  * Declaration : static void go_test(void);
375  *-----
376  * Description : テスト用パラメータの設定値に従って、送信処理を行う。
377  *              : SW3_4 が OFF になるまで連続して送信処理を行う。
378  *-----
379  * Argument  : void
380  *-----
381  * Return Value : void
382  *-----
383  * Note      : None
384  *"FUNC COMMENT END"*****/
385  static void go_test(void)
386  {
387      int cnt=0;
388      int i,j;
389
390      /* ==== lan open ==== */
391      printf("\nTest program start!!\n");
392      printf("lan_open.....");
393      fflush(stdout);
394
395      if( lan_open(test.phymd, test.wv_100, test.wv_10) != OPEN_OK ){
396          printf("error!\n");
397          fflush(stdout);
398          lan_close();
399          return;
400      }
401      printf("OK!\n");
402      printf("(It is possible to stop by turning off SW3_4)\n\n");
403      fflush(stdout);

```

3.10 サンプルプログラムリスト"main.c" (10)

```

404
405     /* ==== transfer start ==== */
406     while( IS_MODESW_ON() ){
407         switch( test.prog ){
408             /* ==== for 10BASE-T Linkpulse && 100BASE-TX ==== */
409             case IDLE:
410                 break;
411             /* ==== for 10BASE-T Harmonic ==== */
412             case ALLONE_OUTPUT:
413                 if( lan_send(allone, sizeof(allone)) != SEND_OK ){
414                     printf("error: lan_send()¥n");
415                     fflush(stdout);
416                     while(1){
417                         /* wait */
418                     }
419                 }
420                 break;
421             /* ==== for 10BASE-T else==== */
422             case RANDOM_OUTPUT:
423                 if( lan_send(s_frame, sizeof(s_frame)) != SEND_OK ){
424                     printf("error: lan_send()¥n");
425                     fflush(stdout);
426                     while(1){
427                         /* wait */
428                     }
429                 }
430                 break;
431             default:
432                 break;
433         }
434     }
435     lan_close();
436     printf("Test program is stopped.¥n");
437     fflush(stdout);
438 }
439 /* End of file */

```

3.11 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 Technology Corp. and is only
5      *   intended for use with Renesas products. No other uses are authorized.
6      *
7      *   This software is owned by Renesas Technology Corp. 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     *   TECHNOLOGY CORP. 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 THE 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) 2009. Renesas Technology Corp., All Rights Reserved.
29     *   "FILE COMMENT" ***** Technical reference data *****
30     *   System Name : SH7619 Sample Program
31     *   File Name   : ether.c
32     *   Abstract    : イーサネットコンプライアンステスト用プログラム例
33     *   Version     : 1.00.00
34     *   Device      : SH7619
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-HS19 (cpu board)
40     *   Description :
41     *****/
42     *   History     : Apr.22,2009 Ver.1.00.00
43     *   "FILE COMMENT END" ***** /

```

3.12 サンプルプログラムリスト"ether.c" (2)

```

44  #include "machine.h"
45  #include "string.h"
46  #include "iodefine.h"
47  #include "defs.h"
48  #include "phy.h"
49  #include "ether.h"
50
51  /* **** プロトタイプ宣言 **** */
52  static void lan_desc_create( void );
53  static void lan_reg_reset( void );
54  static void lan_reg_set( int link );
55
56  /* **** 変数宣言 **** */
57  /* ---- ディスクリプタ ---- */
58  #pragma section ETH_DESC                /* 16バイト境界へ配置 */
59  static volatile TXRX_DESCRIPTOR_SET desc; /* ディスクリプタ領域 */
60  #pragma section
61  /* ---- バッファ ---- */
62  #pragma section ETH_BUFF                /* 16バイト境界へ配置 */
63  static volatile TXRX_BUFFER_SET buf;    /* 送受信バッファ領域 */
64  #pragma section
65  /* ---- MAC アドレス ---- */
66  static const unsigned long my_macaddr_h = 0x00010203;
67  static const unsigned long my_macaddr_l = 0x0405; /* 00:01:02:03:04:05 */
68  /* ---- その他 ---- */
69
70  /*"FUNC COMMENT"*****
71  * ID          :
72  * Outline     : イーサネットオープン関数
73  *-----
74  * Include     : "iodefine.h", "ether.h", "phy.h"
75  *-----
76  * Declaration : int lan_open(int mode, unsigned short wv_100, unsigned short wv_10);
77  *-----
78  * Description : E-DMAC, EtherC, PHY, バッファメモリの初期化を行う。
79  *             : 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。
80  *             : 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。
81  *             : リンク結果に応じて指定された波形調整を行う。
82  *-----
83  * Argument    : int mode          ; I : PHY モードの初期値(PHYIFCR レジスタ bit2-0)
84  *             : unsigned short wv_100 ; I : Tx100 用 波形調整レジスタ値
85  *             : unsigned short wv_10  ; I : Tx10 用 波形調整レジスタ値
86  *-----
87  * Return Value : OPEN_OK(0) : オープン成功
88  *             : OPEN_NG(-1): オープン失敗
89  *-----
90  * Note        : Note
91  *"FUNC COMMENT END"*****
    
```

3.13 サンプルプログラムリスト"ether.c" (3)

```

92     int lan_open(int mode, unsigned short wv_100, unsigned short wv_10)
93     {
94         int link, pre_link;
95         int i,cnt;
96
97         /* ==== EtherC/EDMAC のモジュールスタンバイ解除 ==== */
98         SBY.CR4.BIT._EtherC = 0;
99         /* ==== EtherC,E-DMAC の停止 === */
100        lan_reg_reset();
101        /* ==== バッファメモリの初期化 ==== */
102        lan_desc_create();
103        /* ==== PHY-IF の初期化 ==== */
104        phyif_init(mode);
105        /* ==== リンクの確立を確認 ==== */
106        do{
107            for(i=LOOP_100us*10; i>0; i--){
108                /* 1ms wait */
109            }
110            /* ---- SMI レジスタから交渉結果の PHY モードをリード ---- */
111            link = phy_read_linkmode();
112            if( link == pre_link ){
113                /* 3 秒以上同じモードであることを確認 */
114                cnt++;
115            }
116            else{
117                pre_link = link;
118                cnt = 0;
119            }
120        }while( cnt < 3000 );
121
122        if( link == NEGO_FAIL ){
123            return OPEN_NG;                /* OPEN 失敗 */
124        }
125        /* ==== EtherC,E-DMAC の初期化 ==== */
126        lan_reg_set(link);
127
128        /* ==== 内蔵 PHY モジュールの波形調整 ==== */
129        if( (link == HALF_TX) || (link == FULL_TX) ){
130            phy_set_ampctrl(link, wv_100);
131        }
132        else{
133            phy_set_ampctrl(link, wv_10);
134        }
135
136        return OPEN_OK;
137    }

```

3.14 サンプルプログラムリスト"ether.c" (4)

```

        ~ 省略 ~

193  /*"FUNC COMMENT"*****
194  * ID      :
195  * Outline : フレーム送信関数
196  *-----
197  * Include : "iodefine.h", "ether.h"
198  *-----
199  * Declaration : int lan_send( unsigned char *addr, int flen );
200  *-----
201  * Description : 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
202  *              : 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。
203  *              : EESR レジスタの割込み要求ビットがセットされない現象と、送信 FIFO
204  *              : アンダフロー時の EDMAC 動作停止に対する対応を実施しています。
205  *-----
206  * Argument  : unsigned char *addr; I : 送信フレームの先頭アドレス
207  *            : int flen           ; I : 送信フレームのデータ長
208  *-----
209  * Return Value : SEND_OK(0) : 登録成功
210  *              : SEND_NG(-1): 登録失敗
211  *-----
212  * Note      : Note
213  *"FUNC COMMENT END"*****/
214  int lan_send( unsigned char *addr, int flen )
215  {
216      volatile int w;
217      volatile int t1ms = 0;
218      volatile int t400ms = 0;
219
220      /* ==== 送信中ではないことを確認 ==== */
221      while( desc.pSend_top->td0.BIT.TACT == 1 ){
222          for( w=LOOP_100us; w>0; w-- ){
223              /* 100us wait */
224          }
225          /* ---- 400ms 経過したら EDMAC 動作停止と判断 ---- */
226          /* 送信 FIFO アンダフロー発生時の EDMAC 動作停止への対応 */
227          if( ++t400ms > 4000 ){
228              t400ms = 0;
229              return SEND_NG;
230          }
231      }
232      /* ==== 送信ディスクリプタの更新 ==== */
233      memcpy( desc.pSend_top->td2.TBA, addr, flen ); /* 送信データ */
234      if( flen < 60 ){                             /* 最小フレーム 60 バイト */
235          memcpy( (desc.pSend_top->td2.TBA)+flen, 0, 60-flen ); /*パディング*/
236          flen = 60;
237      }
238      desc.pSend_top->td1.TDL = flen;                /* データ長 */
239      desc.pSend_top->td0.BIT.TACT = 1;             /* 送信許可 */
    
```

3.15 サンプルプログラムリスト"ether.c" (5)

```

240
241     /* ==== 送信停止中であれば起動 ==== */
242     if( EDMAC.EDTRR.BIT.TR == 0 ){                               /* 必ずリードして確認する*/
243         EDMAC.EDTRR.BIT.TR = 1;
244     }
245     /* ==== カレントポインタの更新 ==== */
246     desc.pSend_top = desc.pSend_top->pNext;
247
248     return SEND_OK;
249 }
250
     ~ 省略 ~
    
```

4. 参考ドキュメント

- ソフトウェアマニュアル
SH-1/SH-2/ SH-DSP ソフトウェアマニュアル Rev.7.00
(最新版をルネサス テクノロジ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
SH7619 グループハードウェアマニュアル Rev.6.00
(最新版をルネサス テクノロジ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

- ルネサステクノロジホームページ
<http://japan.renesas.com/>
- お問合せ先
<http://japan.renesas.com/inquiry>
csc@renesas.com

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2009.4.22	—	初版発行

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。

本資料ご利用に際しての留意事項

1. 本資料は、お客様に用途に応じた適切な弊社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弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証するものではありません。
2. 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など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の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、あるいはその他軍事用途の目的で使用しないでください。また、輸出に際しては、「外国為替および外国貿易法」その他輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。
4. 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などの全ての情報は本資料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがあります。弊社の半導体製品のご購入およびご使用に当たりましては、事前に弊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、弊社ホームページ (<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。
5. 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のですが、万一本資料の記述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、弊社はその責任を負いません。
6. 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を流用する場合は、流用する情報を単独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。弊社は、適用可否に対する責任を負いません。
7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海底中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会ください。なお、上記用途に使用されたことにより発生した損害等について弊社はその責任を負いかねますのでご了承願います。
8. 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないでください。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。
 - 1) 生命維持装置。
 - 2) 人体に埋め込み使用するもの。
 - 3) 治療行為（患部切り出し、薬剤投与等）を行うもの。
 - 4) その他、直接人命に影響を与えるもの。
9. 本資料に記載された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、実装条件およびその他諸条件につきましては、弊社保証範囲内でご使用ください。弊社保証値を越えて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、弊社はその責任を負いません。
10. 弊社は製品の品質および信頼性の向上に努めておりますが、特に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。弊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、誤動作防止設計などの安全設計（含むハードウェアおよびソフトウェア）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。
11. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなきよう、お客様の責任において十分な安全設計をお願いします。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。
12. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断りいたします。
13. 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弊社営業窓口までご照会ください。

D039444