

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

内蔵 PHY モジュールの初期設定例

要旨

この資料は、SH7619 の内蔵 PHY モジュールを利用するために必要な設定項目の例を示します。

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.....	28

1. はじめに

1.1 仕様

- メイン関数では、PHYIF の初期設定と PHY 自動交渉を実行します。
- PHYIF 初期設定では、内蔵 PHY モジュールを使用するための PHYIF の初期設定を行います。
- PHY 自動交渉では、PHY の自動交渉機能（または並列検出機能）により決定したモードを取得します。

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 (PHYIF)
- 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)

1.3 適用条件

- マイコン: SH7619 (R4S76190)
- 動作周波数:
 - 内部クロック 125MHz
 - バスクロック 62.5MHz
 - 周辺クロック 31.25MHz
- 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.9.01Release01
- 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

本資料の参考プログラムは、SH7619 初期設定アプリケーションノートの設定条件で動作確認をしています。そちらも合わせてご参照ください。

2. 応用例の説明

本応用例では、内蔵モジュールを利用した自動交渉を行います。内蔵 PHY モジュールの設定は PHY インタフェース (PHYIF) およびピンファンクションコントローラ (PFC) で行い、自動交渉結果は内蔵 PHY モジュール内の PHY 管理制御部 (SMI) を使用して読み出します。

2.1 使用機能の動作概要

内蔵 PHY の初期設定は、ピンファンクションコントローラと PHY インタフェース、および SMI (Serial Management Interface) で行います。

ピンファンクションコントローラでは内蔵 PHY の有効/無効を設定し、PHY インタフェースではモジュールリセットや動作クロックの選択、PHY モードの設定などを行います。その他の制御や状態参照は SMI レジスタ群で行います。SMI レジスタ群へのアクセスは SMI を用いて行います。

SMI はイーサネットコントローラ (EtherC) と内蔵 PHY モジュールをシリアルで双方向につないだものです。IEEE802.3 規格の MII 管理インタフェースに準拠していますので、規格に沿った手順でアクセスすることができます。SH7619 では EtherC の PHY 部インタフェースレジスタ (PIR) を使用して SMI を制御します。

SMI の制御手順とアクセス可能な SMI レジスタ群については、「2.2 SMI レジスタ群のアクセス手順」と「SH7619 グループ ハードウェアマニュアル 22.4.2 レジスタの説明」をご参照ください。

表 1、表 2 に PHY と PHYIF の概要を、図 1、図 2 に PHYIF と SMI の概略図を示します。

表1 内蔵 PHY モジュールの概要

項目	概要
IEEE802.3/802.3u 準拠	10/100Mbps イーサネット PHY 仕様準拠
クロックソース	内部クロック (内蔵 PHY 用クロック M ϕ) 外部クロック (25MHz)
リンクモード	100Mbps (全二重、半二重) 10Mbps (全二重、半二重)
リンク決定方式	自動交渉 (Auto-Negotiation) 並列検出 (パラレル・ディテクション) 手動設定
MAC インタフェース	MII (Media Independent Interface)
管理制御部	SMI (Serial Management Interface) IEEE802.3 規格準拠レジスタ 0~6

表2 PHYIF の概要

項目	概要
内蔵 PHY 有効/無効	ピンファンクションコントローラ (PFC) により設定可能
内蔵 PHY 設定	モジュールリセット 動作クロック選択 PHY モード PHY アドレス

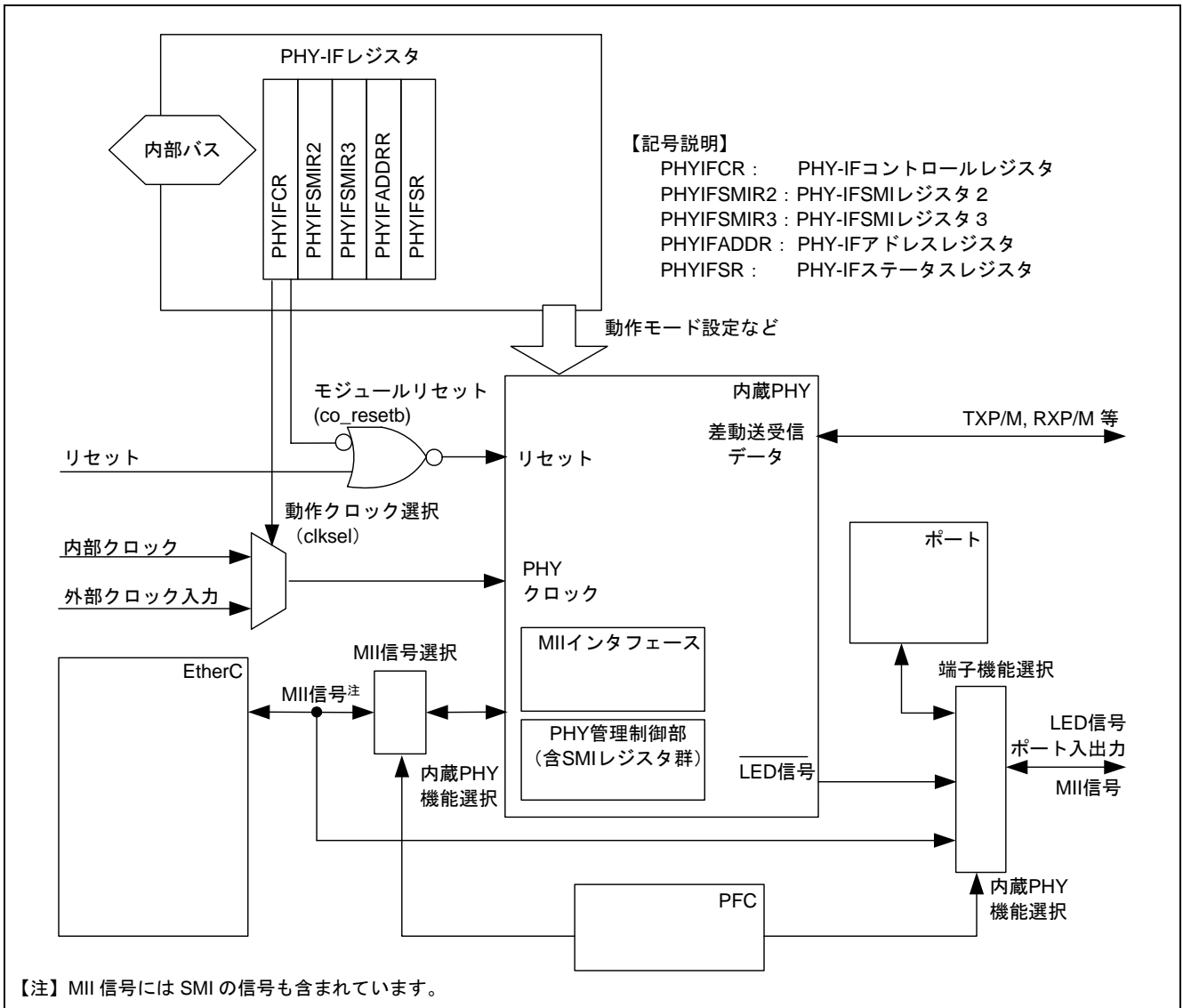


図1 PHYIF 概略図

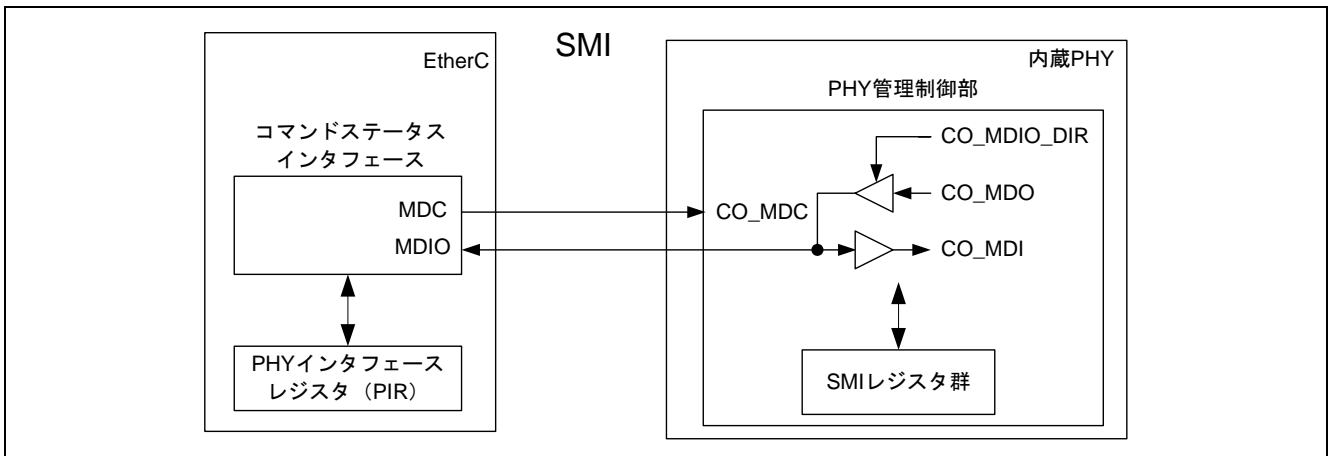


図2 SMI 概略図

2.2 SMI レジスタ群のアクセス手順

SMI レジスタ群をアクセスする手順を説明します。

SMIは、MDCとMDIO（いずれもEtherC側の端子名）の2本からなり、MDCが同期用クロック端子、MDIOがデータ入出力端子です。各端子の状態はEtherCのPIRレジスタで参照または変更できます。制御用端子がないため、SMIでは必ず規定されたフォーマット（MII管理フレーム）どおりにデータを出力しなければなりません。MII管理フレームを図3に示します。本サンプルプログラムではIDLE状態でZ出力を1ビット実行します。IEEE802.3規格ではクロック入力について言及されていませんが、本LSIの内蔵PHYを使用する場合には必要となります。

MII管理フレームの入出力は、PREから順に1ビット単位で行います。1ビット単位の入出力フローを図4～図6に示します。MDCおよびMDIOの入出力タイミングはIEEE802.3規格を満たしてください。表3および図7にIEEE802.3規格の入出力タイミングを示します。

アクセス種別	MII管理フレーム							
項目	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
ビット数	32	2	2	5	5	2	16	1
リード	1..1	01	10	AAAAA	RRRRR	Z0	D..D	Z
ライト	1..1	01	01	AAAAA	RRRRR	10	D..D	Z

[記号説明]

- PRE (preamble) : 同期を取る。32個の連続した1を出力。
- ST(start of frame) : フレームの先頭を示す。01の出力。
- OP(operation code) : リード/ライトを選択。リードの場合は10、ライトの場合は01を出力。
- PHYAD(PHY Address) : 複数のPHY-LSIを接続する場合の識別アドレス。内蔵PHYモジュールではPHYは単独で接続されるが、PHYIFのPHYIFADDRRで設定した値を出力すること。MSBから出力。
- REGAD(Register Address) : MIIレジスタの番号を指定。MSBから出力。
- TA(turn around) : 信号の衝突を防ぐため、MDIOの送信元を切り替える。
(a) リード時は、1ビット分のバス解放（Z出力）を行う。
(PHYからは0が出力されるので、Z0と表記)
(b) ライト時は10を出力。
- DATA(data) : レジスタのリード値またはライト値。16ビットデータ。MSBから順次ライトまたはリード。
- IDLE(IDLE condition) : 次のMII管理フォーマット入力までの待機処理。バス解放(Z出力)を行う。

図3 MII 管理フレームフォーマット

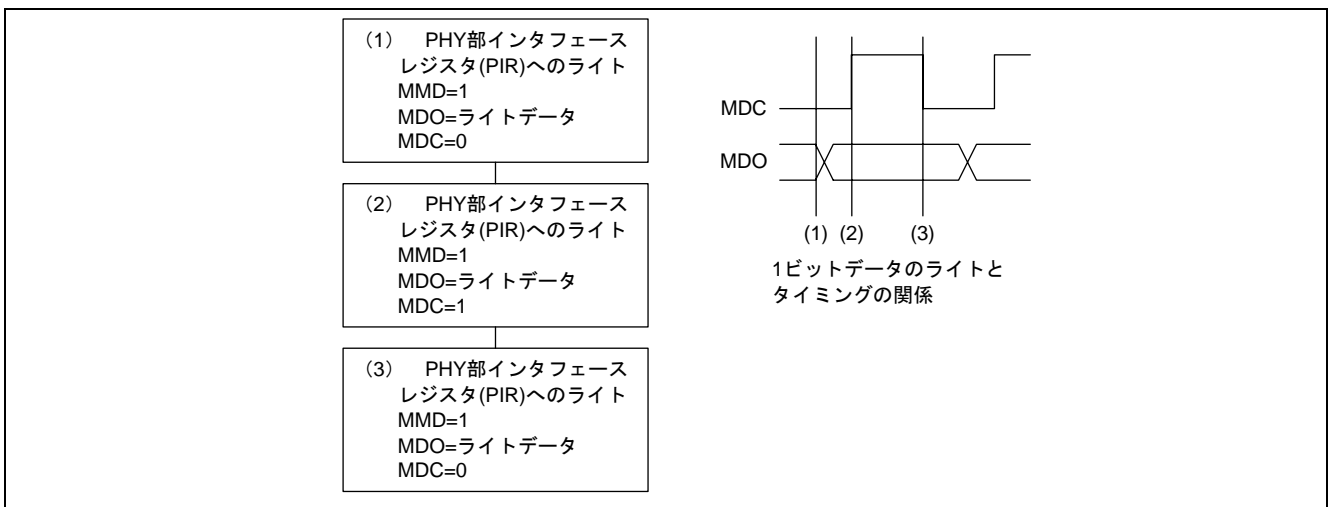


図4 1ビットデータのライトフロー

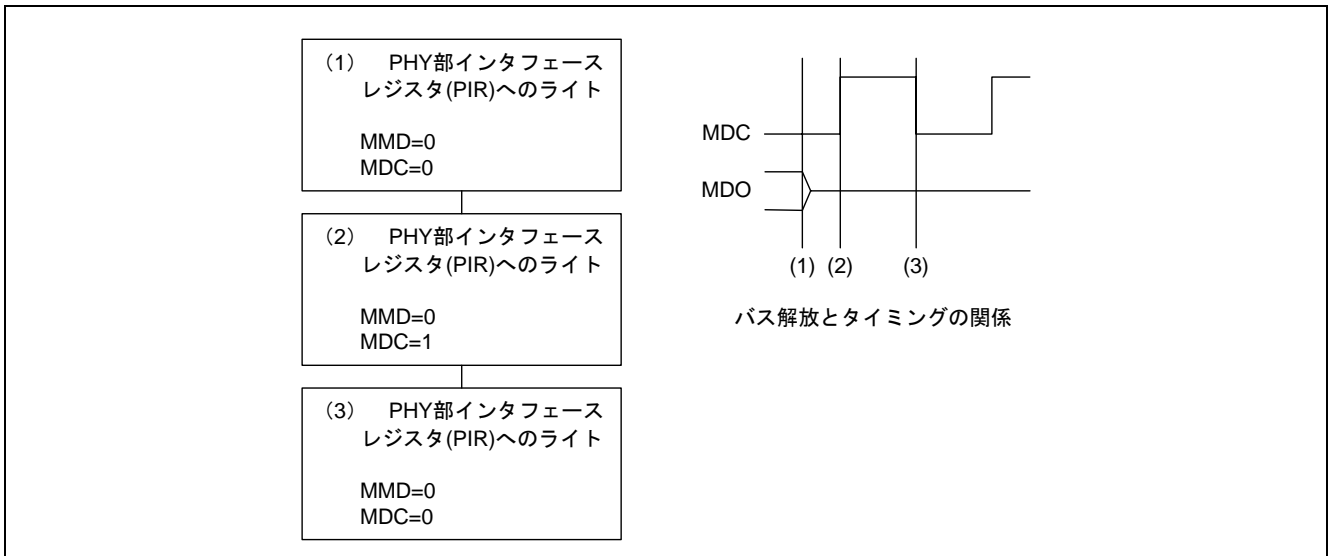


図5 バス解放フロー (Z 出力)

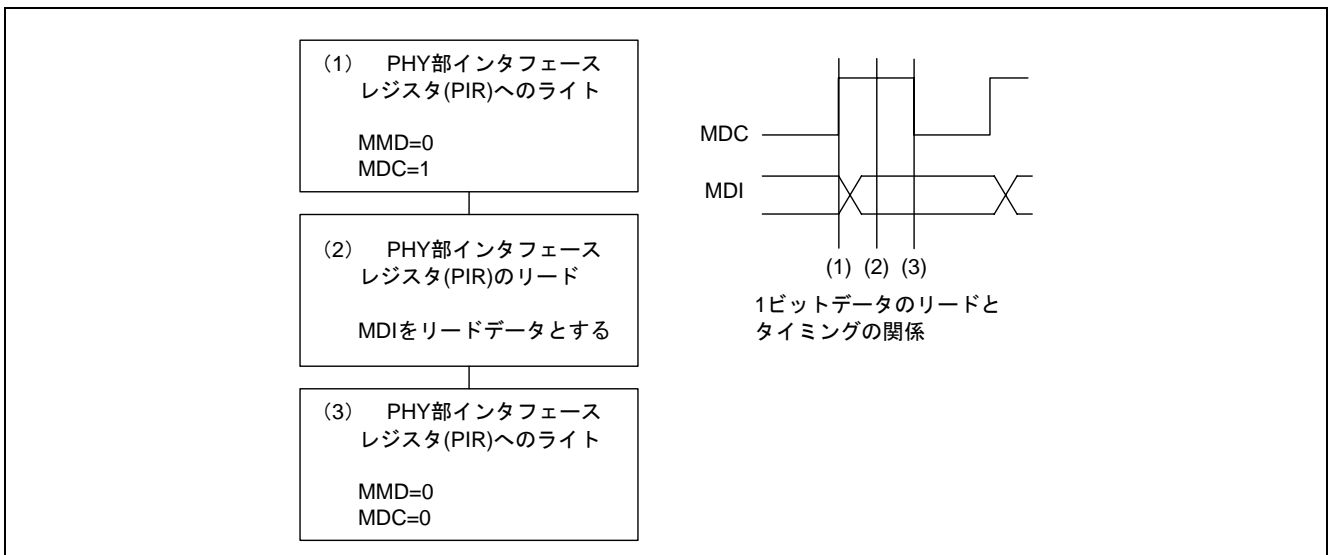


図6 1ビットデータのリードフロー

表3 MDC/MDIO入出力タイミング

項目	記号	min	max	単位
MDC High レベルパルス幅	t_1	160		ns
MDC Low レベルパルス幅	t_2	160		ns
MDC サイクル時間	t_3	400		ns
MDIO セットアップ時間	t_4	10		ns
MDIO ホールド時間	t_5	10		ns
MDIO 出力遅延時間	t_6	0	300	ns

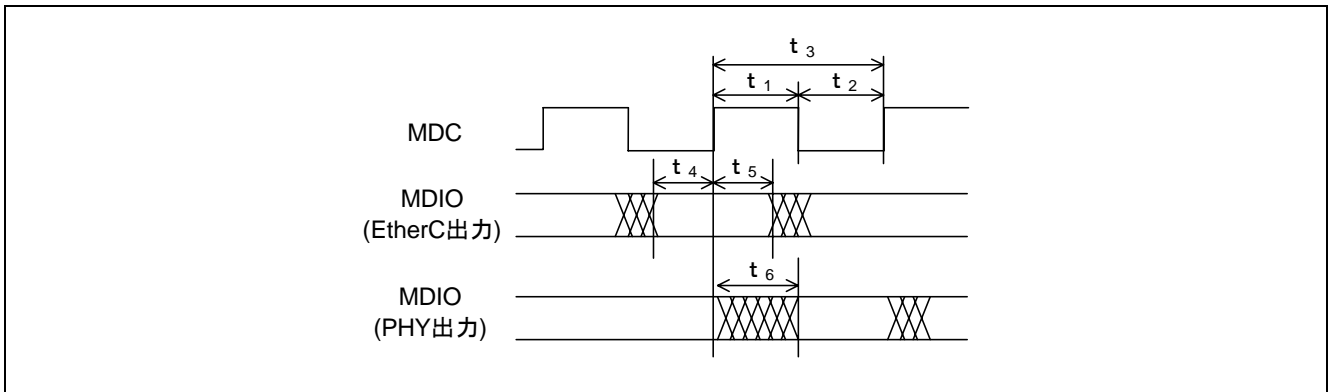


図7 MDC/MDIO 入出力タイミング

2.3 使用機能の設定手順

図 8にPHYIFの初期設定フロー例を示します。

各レジスタ設定の詳細は、「SH7619 グループ ハードウェアマニュアル」を参照ください。

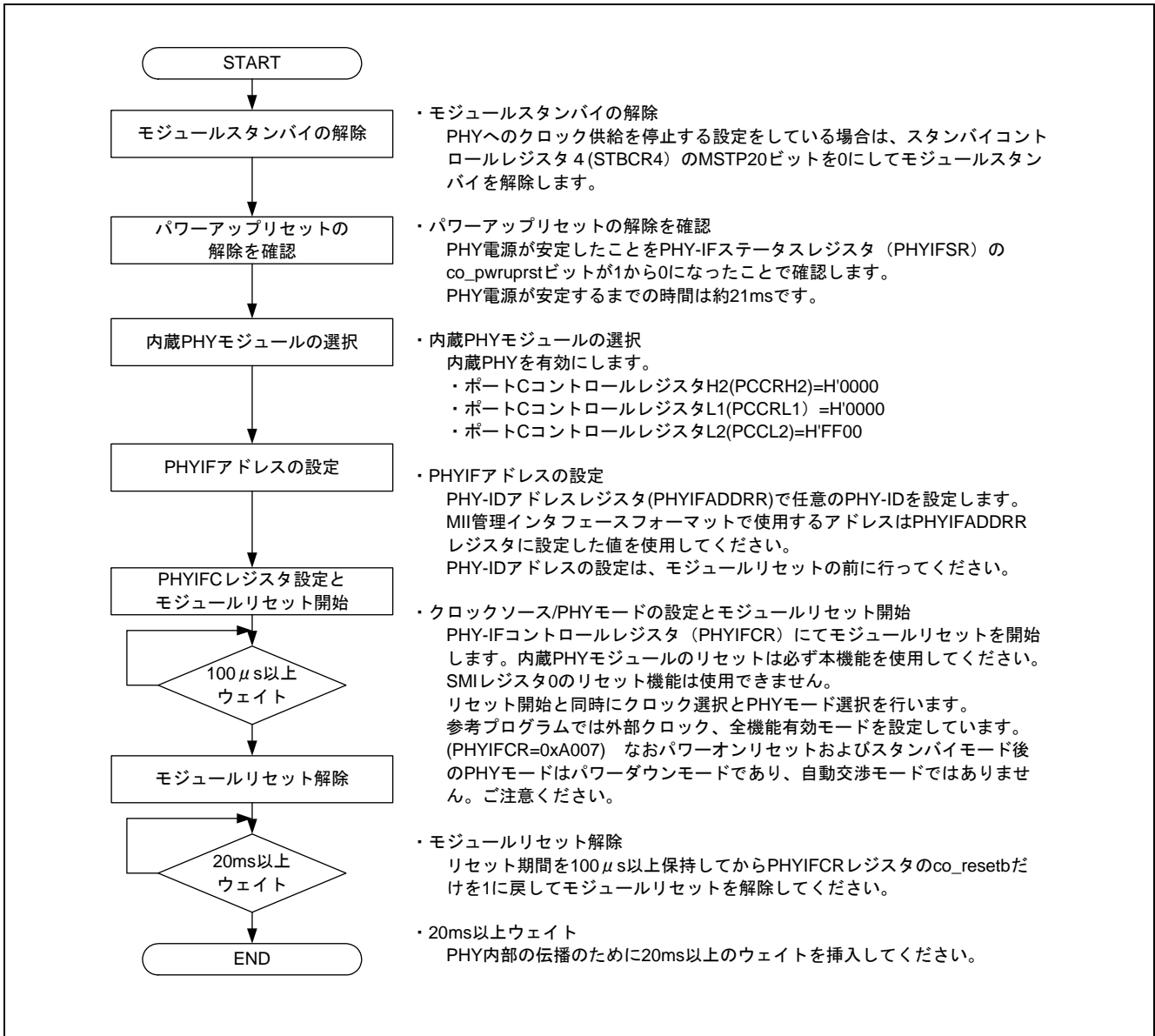


図8 PHYIF の初期設定フロー例

2.4 参考プログラムの説明

参考プログラムは

- main.c
- phy.c

の2つのソースプログラムと「SH7619 グループアプリケーションノート 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) が記述されており、PHYIFの初期化と自動交渉結果取得を実行します。図 9にmain関数のフローを示します。
- phy.c
PHYIFの初期化 (phyif_init関数) と自動交渉結果の取得 (phy_autonego関数) が記述されています。phyif_init関数のフローは「図 8 PHYIFの初期設定フロー例」と同一ですので、そちらをご参照ください。図 10にphy_autonego関数のフローを示します。図 11～図 14にphy_autonego関数内で実行されるSMIレジスタリード関数 (phy_reg_read関数) とその下位関数のフローを示します。

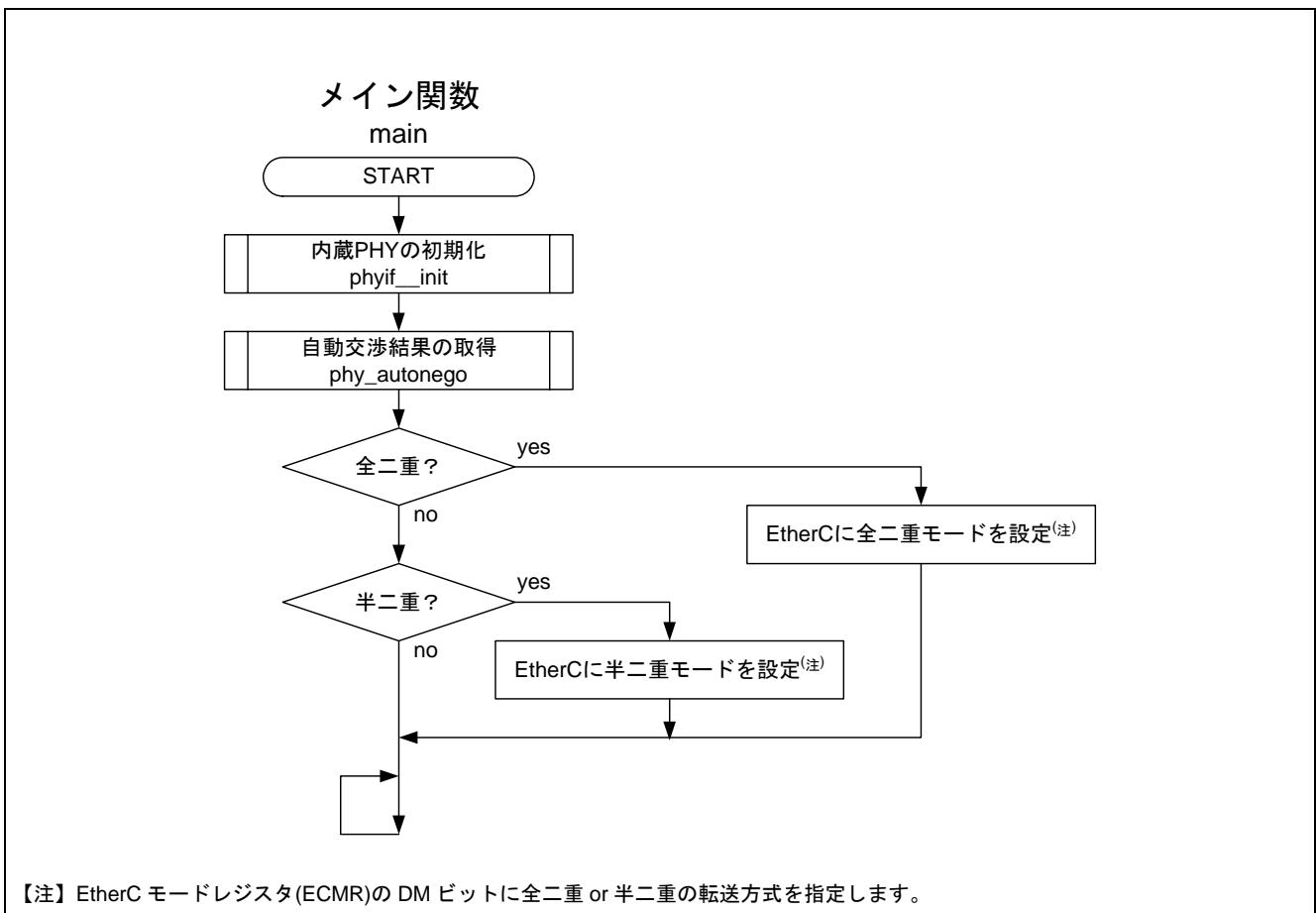


図9 メイン関数の処理フロー

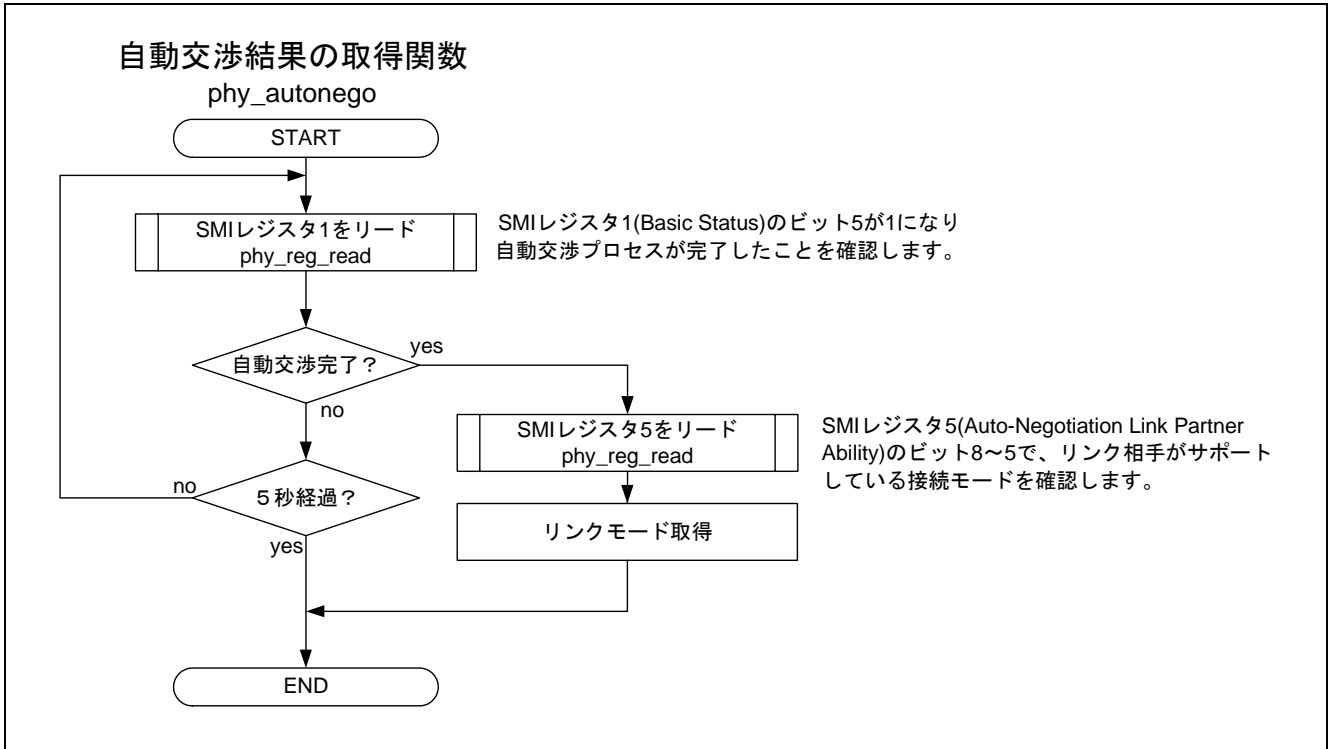


図10 自動交渉結果取得関数の処理フロー

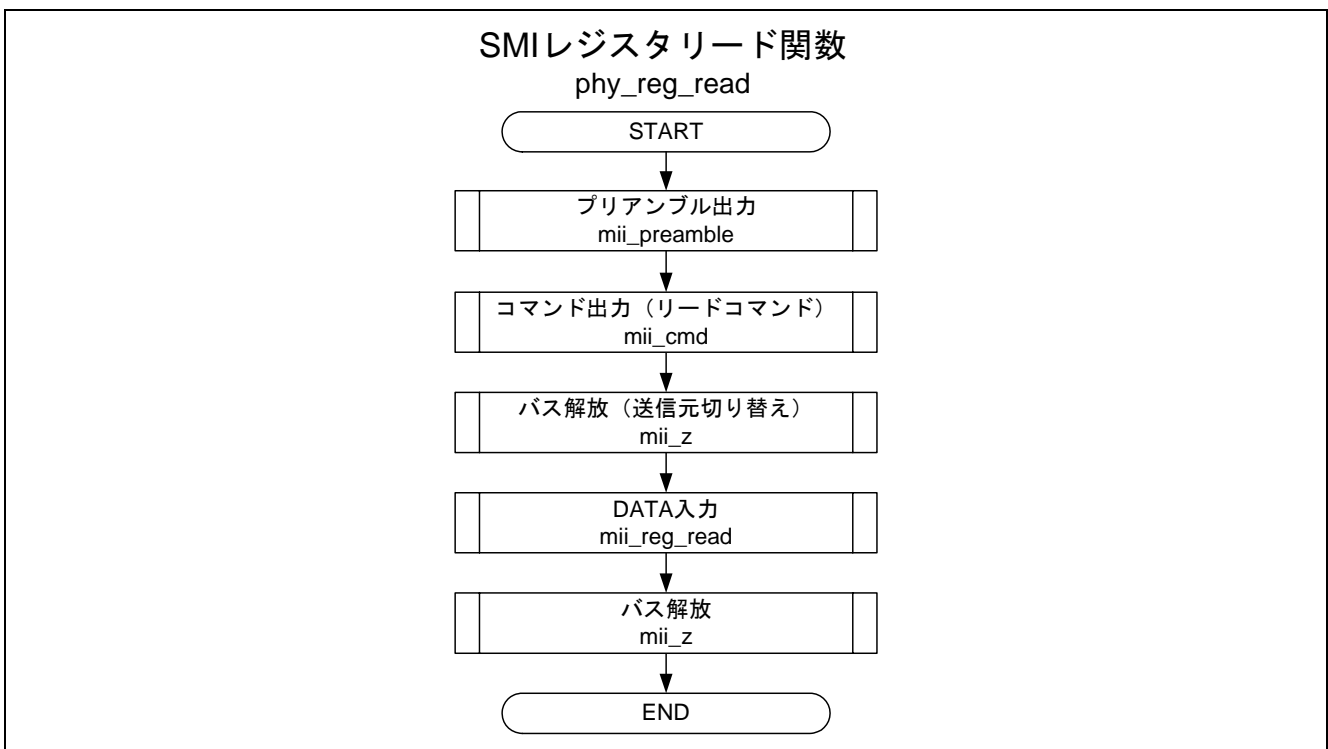


図11 SMI レジスタアクセスの処理フロー (1)

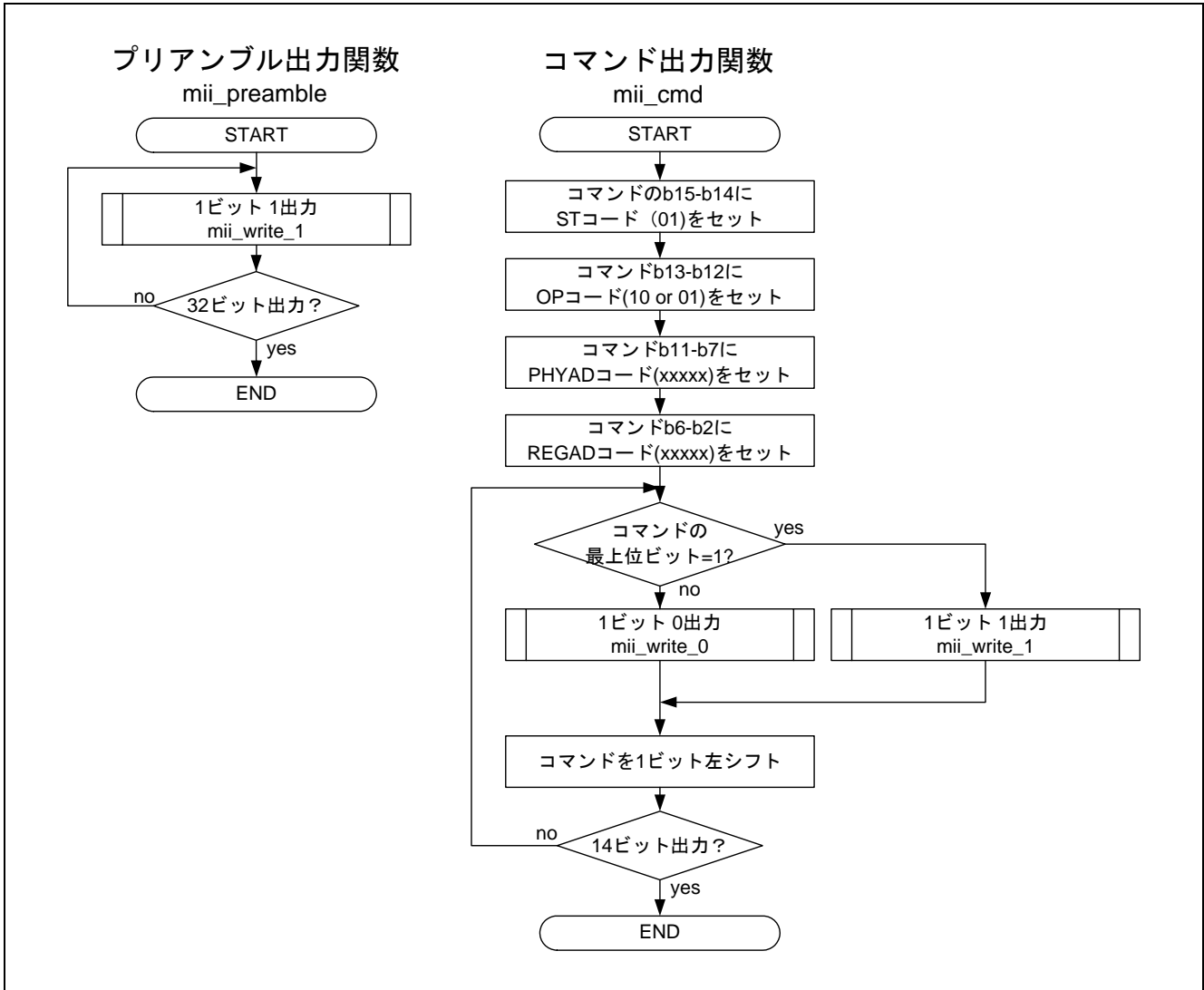


図12 SMI レジスタアクセスの処理フロー (2)

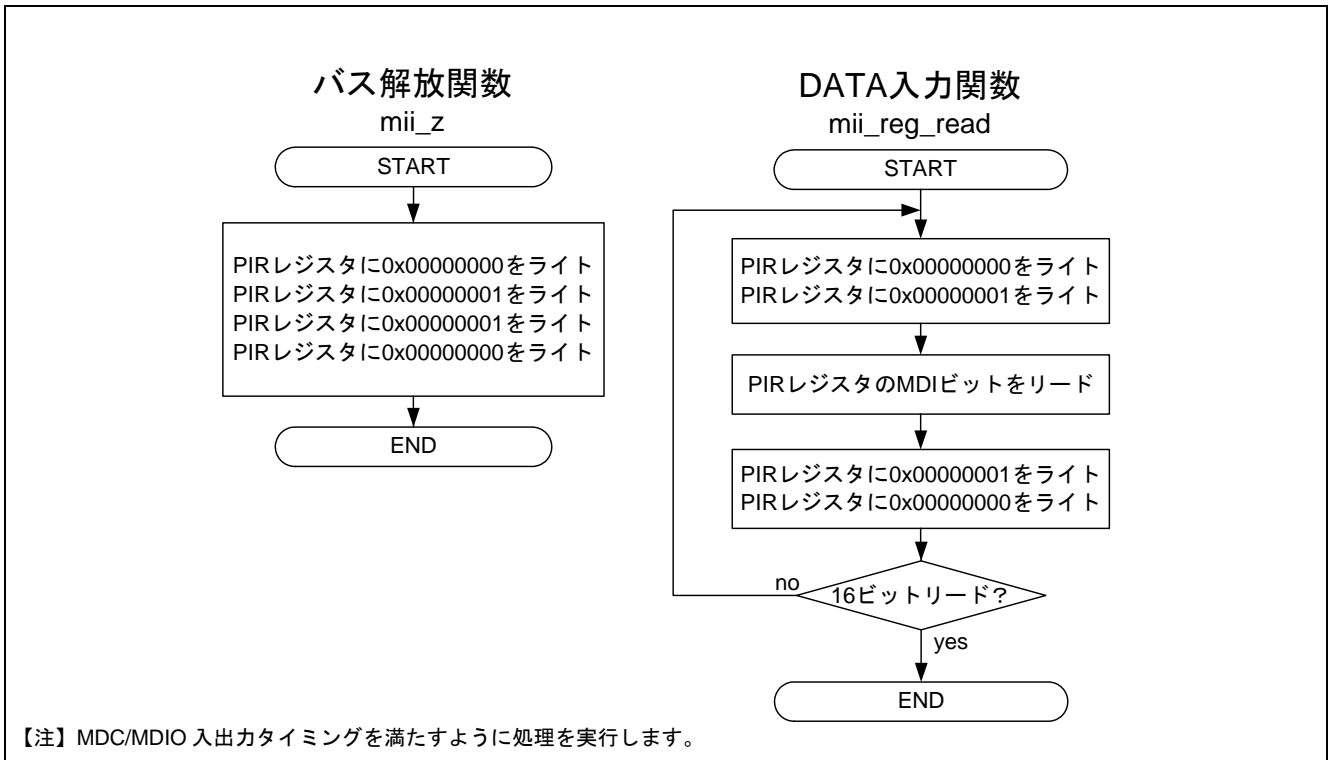


図13 SMI レジスタアクセスの処理フロー (3)

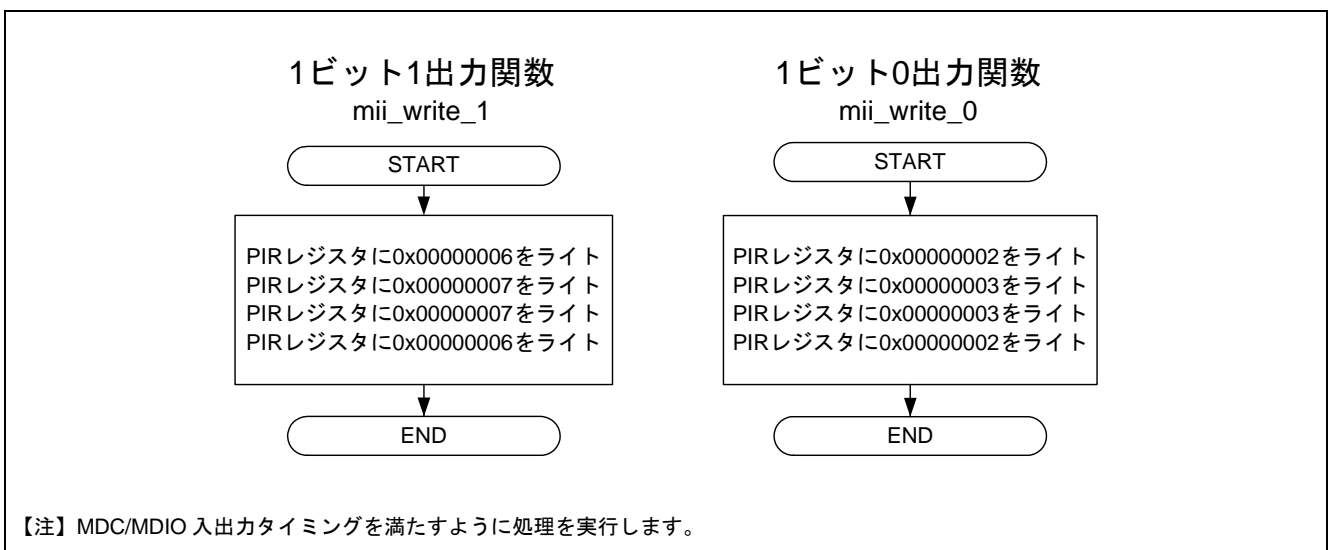


図14 SMI レジスタアクセスの処理フロー (4)

2.5 参考プログラムにおける設定内容

表 4に参考プログラムでの設定を示します。

表4 参考プログラムでの設定

項目	設定内容
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)
リンクモード	100Mbps (全二重、半二重) および 10Mbps (全二重、半二重)
リンク決定方式	自動交渉 (Auto-Negotiation)
使用する SMI レジスタ	レジスタ 5 (Auto-Negotiation LinkPartner Ability)

2.6 参考プログラム使用時の注意点

- 参考プログラムは、PHYのリンク決定方式として自動交渉モードを選択しています。内蔵PHYは 10Mbps/半二重から 100Mbps/全二重まで全てのリンクモードをサポートしているため、相手先が自動交渉モードの場合は表 5の優先順位に従い、リンクモードが決定します。
- 相手先が自動交渉モードをサポートしていない場合も、SH7619 内蔵 PHY の並列検出機能によりリンクを検出します。その場合のリンクモードは半二重モードになります。接続相手が全二重モード固定の場合は接続できません。
- 通常、自動交渉は数秒で完了しますが、参考プログラムでは phy_autonego 関数内で最大 5 秒間、自動交渉の完了を確認します。

表5 リンクモードの優先順位

優先順位		リンクモード
高	1	100Mbps 全二重
	2	100Mbps 半二重
低	3	10Mbps 全二重
	4	10Mbps 半二重

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  /*"FILE COMMENT"*****
2  *
3  * System Name : SH7619 Sample Program
4  * File Name   : main.c
5  * Contents    : 内蔵 PHY モジュールの初期設定例
6  * Version     : 1.00.00
7  * Model       : M3A-HS19
8  * CPU         : SH7619
9  * Compiler    : SHC9.1.1.0
10 * note        :
11 *             <注意事項>
12 *             本サンプルプログラムはすべて参考資料であり、
13 *             その動作を保証するものではありません。
14 *             本サンプルプログラムはお客様のソフトウェア開発時の
15 *             技術参考資料としてご利用ください。
16 *
17 * Copyright (C) 2007 Renesas Technology Corp. All Rights Reserved
18 * and Renesas Solutions Corp. All Rights Reserved
19 *
20 * history      :2007.06.29 ver.1.00.00
21 *"FILE COMMENT END"*****/
22 #include "iodefine.h"
23 #include "defs.h"
24 #include "phy.h"
25
26 /* ==== Prototype declaration ==== */
27 void main(void);
28

```


3.2 サンプルプログラムリスト”main.c”(2)

```

29  *"FUNC COMMENT"*****
30  * ID      :
31  * Outline : PHY モジュール初期設定サンプルプログラム メイン関数
32  *-----
33  * Include : #include "iodefine.h"
34  *-----
35  * Declaration : void main(void)
36  *-----
37  * Function  : 内蔵 PHY モジュールを利用するための PHYIF の設定を行います。
38  *           : PHYIF のモジュールリセット後、PHY は自動交渉を行いリンクモード
39  *           : を決定しますので、本サンプルでは結果を読み出しています。
40  *           : なお全二重/半二重の結果は、EtherC に設定する必要があります。
41  *-----
42  * Argument  : void
43  *-----
44  * ReturnValue : void
45  *-----
46  * Notice    :
47  *"FUNC COMMENT END"*****/
48 void main(void)
49 {
50     int link;
51
52     /* ==== PHY-IF 初期設定 ==== */
53     phyif_init();
54
55     /* ==== PHY 交渉結果の取得 ==== */
56     link=phy_autonego();
57
58     /* ==== EtherC デュプレックスモードの設定 ==== */
59     if( link == FULL_TX || link == FULL_10M ){
60         EtherC.ECMR.BIT.DM = 1;          /* 全二重 */
61     }
62     else if(link == HALF_TX || link == HALF_10M){
63         EtherC.ECMR.BIT.DM = 0;          /* 半二重 */
64     }
65     else{
66         /* リンク失敗 */
67     }
68
69     /* ==== メインループ ==== */
70     while(1){
71         ;
72     }
73 }
74
75 /* End of file */

```

3.3 サンプルプログラムリスト"phy.c"(1)

```

1  /*"FILE COMMENT"*****
2  *
3  *   System Name   : SH7619 Sample Program
4  *   File Name    : phy.c
5  *   Contents     : 内蔵 PHY モジュールの初期設定例
6  *   Version      : 1.00.01
7  *   Model        : M3A-HS19
8  *   CPU          : SH7619
9  *   Compiler     : SHC9.1.1.0
10 *   note         :
11 *
12 *               <注意事項>
13 *               本サンプルプログラムはすべて参考資料であり、
14 *               その動作を保証するものではありません。
15 *               本サンプルプログラムはお客様のソフトウェア開発時の
16 *               技術参考資料としてご利用ください。
17 *
18 *   Copyright (C) 2007 Renesas Technology Corp. All Rights Reserved
19 *   and Renesas Solutions Corp. All Rights Reserved
20 *
21 *   history      :2007.07.03 ver.1.00.00
22 *                :2007.07.13 ver.1.00.01 関数ヘッダコメント誤記修正(mii_write_1/0)
23 *"FILE COMMENT END"*****/
24 #include "iodefine.h"
25 #include "defs.h"
26 #include "phy.h"
27
28 /* **** プロトタイプ宣言 **** */
29 static unsigned short phy_reg_read( unsigned short reg_addr );
30 static void phy_reg_write( unsigned short reg_addr, unsigned short data );
31 static void mii_preamble( void );
32 static void mii_cmd( unsigned short reg_addr, int option );
33 static void mii_ta10( void );
34 static void mii_reg_read( unsigned short *data );
35 static void mii_reg_write( unsigned short data );
36 static void mii_write_1( void );
37 static void mii_write_0( void );
38 static void mii_z( void );
39
40
41 /* **** マクロ定義 **** */
42 /* SMI レジスタ */
43 #define BASIC_MODE_CONTROL_REG    0
44 #define BASIC_MODE_STATUS_REG    1
45 #define PHY_IDENTIFIER1_REG      2
46 #define PHY_IDENTIFIER2_REG      3
47 #define AN_ADVERTISEMENT_REG     4
48 #define AN_LINK_PARTNER_ABILITY_REG 5
49 #define AN_EXPANSION_REG         6
50 /* PHY アドレス */
51 #define PHY_ADDR                  0
    
```

3.4 サンプルプログラムリスト"phy.c"(2)

```

52  /* SMI アクセス用 */
53  #define PHY_ST          1
54  #define PHY_WRITE      1
55  #define PHY_READ       2
56  #define MDC_WAIT       3          /* 400ns/4 < (Pφ*2)*MDC_WAIT */
57
58  /*"FUNC COMMENT"*****
59  * Outline      : PHY-IF モジュールの初期設定
60  *-----
61  * Include      : #include "iodefine.h"
62  *-----
63  * Declaration : void phyif_init(void);
64  *-----
65  * Function    : PHY モジュールを使用するために、ピンファンクションコント
66  *              : ローラ (PFC) と PHY インタフェース (PHY-IF) を初期化します。
67  *              : PHY のクロックは外部クロック 25MHz を使用しています。
68  *              : PHY モジュールを使用する前に本関数を実行してください。
69  *              : また内蔵 PHY モジュールは、SMI レジスタ 0 のリセット機能が使用でき
70  *              : ません。PHY-IF のモジュールリセット機能を使用してください。
71  *-----
72  * Argument    : void
73  *-----
74  * Return Value: void
75  *-----
76  * Notice      : ウェイト時間はシステムに応じて変更してください。
77  *"FUNC COMMENT END"*****/
78  void phyif_init( void )
79  {
80      volatile int t100us = LOOP_100us;
81      volatile int t20ms = LOOP_100us * 200;
82
83      /* ==== モジュールストップの解除 ==== */
84      SBY.CR4.BIT._PHYIF = 0;
85
86      /* ==== パワーアップリセット ==== */
87      while( PHYIF.SR.BIT.co_pwruprst != 0 ){
88          /* sleep */
89      }
90      /* ==== 内蔵 PHY モジュールの選択 ==== */
91      PFC.PCCR2.WORD = 0x0000;          /* 外付け PHY 用 MII 端子を無効にする。*/
92      PFC.PCCRL1.WORD = 0x0000;        /*              //              */
93      PFC.PCCRL2.WORD = 0xFF00;        /* 内蔵 PHY 用 出力端子を有効にする。 */
94
95      /* ==== クロックの設定 ==== */
96                                  /* 内部クロック利用時はここで CPG 設定*/
97      /* ==== 内蔵 PHY のリセット ==== */
98      /* ---- SMI 初期値設定 ---- */
99      PHYIF.ADDRR.BIT.co_st_phyadd = PHY_ADDR;
100                                  /* PHYLSI のアドレス。SMI の利用に必要*/
101                                  /* モジュールリセット前に設定する */

```

3.5 サンプルプログラムリスト"phy.c"(3)

```

102     /* ---- モジュールリセット ---- */
103     PHYIF.CR.WORD = 0xa007;          /* 外部 CLK,全機能モード,自動交渉 */
104     while( --t100us ){              /* ※SMI レジスタ 0 のリセットは使用禁止*/
105         /* sleep */                  /* リセット期間確保 */
106     }
107     /* ==== 内蔵 PHY のリセット解除 ==== */
108     PHYIF.CR.BIT.co_resetb = 1;
109     while( --t20ms ){
110         /* sleep */                  /* PHY 内部のリセット伝播のため */
111     }
112 }
113
114 /*"FUNC COMMENT"*****
115 * Outline      : PHY リンクの自動交渉結果検出
116 *-----
117 * Include      : #include "iodefine.h"
118 *              : #include "defs.h"
119 *-----
120 * Declaration  : int phy_autonego(void);
121 *-----
122 * Function     : 自動交渉結果を SMI を使って読み込み、戻り値で返します。
123 *              : 内蔵 PHY は 100Mbps 全二重モードまで全てサポートしていますので、
124 *              : リンク先が自動交渉に対応している場合は、リンク先の最高性能と
125 *              : なるリンクモードで接続されます。
126 *              : リンク先が自動交渉に対応していない場合も、内蔵 PHY の並列検出機
127 *              : 能によりリンク速度が検出されます。その場合は半二重モードとなり
128 *              : ます。
129 *              : 自動交渉は通常約 1200ms で完了しますが、本関数は自動交渉完了を
130 *              : 最大 5 秒間確認します。
131 *-----
132 * Argument     : void
133 *-----
134 * Return Value: 4(FULL_TX) :100Mbps 全二重
135 *              : 3(HALF_TX) :100Mbps 半二重
136 *              : 2(FULL_10M):10Mbps 全二重
137 *              : 1(HALF_10M):10Mbps 半二重
138 *              : 0(NEGO_FAIL) :交渉失敗
139 *-----
140 * Notice       : 本サンプルプログラムは並列検出機能を考慮していません。
141 *"FUNC COMMENT END"*****
    
```

3.6 サンプルプログラムリスト”phy.c”(4)

```

142     int phy_autonego( void )
143     {
144         unsigned short data;
145         int link = NEGO_FAIL;
146         volatile int t;
147         int i;
148
149         /* ==== 自動交渉の終了待ちループ(Max5 秒) ==== */
150         for(i=0; i<500; i++){
151             /* ==== 経過時間計測の基準とするために 10ms ウェイト ==== */
152             t = LOOP_100us*100;
153             while( --t){
154                 ;
155             }
156             /* ==== 自動交渉完了確認 ==== */
157             data = phy_reg_read(BASIC_MODE_STATUS_REG);
158             if( data & 0x0020 ){                /* SMI レジスタ 1 :Basic Status      *
159                                                 * ビット 5 :(1)自動交渉プロセス完了 *
160                                                 *          :(0)自動交渉プロセス未完 */
161                 /* ---- 自動交渉が完了したので相手の能力を取得 ---- */
162                 data = phy_reg_read(AN_LINK_PARTNER_ABILITY_REG);
163                 /* SMI レジスタ 5 :Auto-Nego Link Partner Ability *
164                 * ビット 8 :(1)100Mbps/全二重 可能 *
165                 * ビット 7 :(1)100Mbps/半二重 可能 *
166                 * ビット 6 :(1) 10Mbps/全二重 可能 *
167                 * ビット 5 :(1) 10Mbps/半二重 可能 */
168                 /* ---- 結果判別->交渉終了で break ---- */
169                 if( data&0x0100 ){
170                     link = FULL_TX;
171                 }
172                 else if( data&0x0080 ){
173                     link = HALF_TX;
174                 }
175                 else if( data&0x0040 ){
176                     link = FULL_10M;
177                 }
178                 else if( data&0x0020 ){
179                     link = HALF_10M;
180                 }
181                 else{
182                     link = NEGO_FAIL;        /* ここにはこないはず */
183                 }
184                 /* ---- 自動交渉検出処理 完了 ---- */
185                 break;
186             }
187         }
188         return link;
189     }
190

```

3.7 サンプルプログラムリスト”phy.c”(5)

```

191  /*"FUNC COMMENT"*****
192  * Outline      : PHY モジュール SMI レジスタリード
193  *-----
194  * Include      :
195  *-----
196  * Declaration : static unsigned short phy_reg_read(unsigned short reg_addr)
197  *-----
198  * Function     : PHY モジュール SMI レジスタ値を取得します。
199  *-----
200  * Argument     : unsigned short reg_addr : I : 値を読み込む SMI レジスタアドレス
201  *-----
202  * Return Value: 取得したレジスタの値
203  *-----
204  * Notice       :
205  *"FUNC COMMENT END"*****/
206  static unsigned short phy_reg_read( unsigned short reg_addr )
207  {
208     unsigned short  data;
209
210     mii_preamble();
211     mii_cmd( reg_addr, PHY_READ );
212     mii_z();
213     mii_reg_read( &data );
214     mii_z();
215
216     return data;
217 }
218
219 /*"FUNC COMMENT"*****
220 * Outline      : PHY モジュール SMI レジスタライト
221 *-----
222 * Include      :
223 *-----
224 * Declaration : static void phy_reg_write( unsigned short reg_addr,
225 *          :          unsigned short data )
226 *-----
227 * Function     : PHY モジュール SMI レジスタに値を設定します。
228 *-----
229 * Argument     : unsigned short reg_addr : I : 値を書き込む SMI レジスタアドレス
230 *          : unsigned short data      : I : SMI レジスタに設定する値
231 *-----
232 * Return Value: void
233 *-----
234 * Notice       :
235 *"FUNC COMMENT END"*****/
236  static void phy_reg_write( unsigned short reg_addr, unsigned short data )
237  {
238     mii_preamble();
239     mii_cmd( reg_addr, PHY_WRITE );
240     mii_ta10();
241     mii_reg_write( data );
242     mii_z();
243
244 }

```

3.8 サンプルプログラムリスト"phy.c"(6)

```

245
246 /*"FUNC COMMENT"*****
247 * Outline      : PHY モジュール SMI レジスタへのアクセス準備
248 *-----
249 * Include      :
250 *-----
251 * Declaration : static void mii_preamble( void )
252 *-----
253 * Function    : PHY モジュール SMI レジスタにアクセスするための前準備として、
254 *              : SMI に 1 を出力します。
255 *-----
256 * Argument    : void
257 *-----
258 * Return Value: void
259 *-----
260 * Notice      :
261 /*"FUNC COMMENT END"*****/
262 static void mii_preamble( void )
263 {
264     short i;
265
266     i = 32;
267     while( i > 0 ) {
268         /* MII(Media Independent Interface)ブロックに 1 を出力。*/
269         mii_write_1();
270         i--;
271     }
272 }
273
274 /*"FUNC COMMENT"*****
275 * Outline      : PHY モジュール SMI レジスタモード設定
276 *-----
277 * Include      :
278 *-----
279 * Declaration : static void mii_cmd( unsigned short reg_addr,
280 *              :                          int option )
281 *-----
282 * Function    : PHY モジュール SMI レジスタの R/W モードを設定します。
283 *-----
284 * Argument    : unsigned short reg_addr : I : SMI レジスタアドレス
285 *              : int option           : I : R/W モード指定
286 *-----
287 * Return Value: void
288 *-----
289 * Notice      :
290 /*"FUNC COMMENT END"*****/

```

3.9 サンプルプログラムリスト"phy.c"(7)

```

291 static void mii_cmd( unsigned short reg_addr, int option )
292 {
293     int i;
294     unsigned short data;
295
296     data = 0;
297     data = (PHY_ST << 14); /* ST code */
298     if( option == PHY_READ ) {
299         data |= (PHY_READ << 12); /* OP code(RD) */
300     }
301     else {
302         data |= (PHY_WRITE << 12); /* OP code(WT) */
303     }
304     data |= (PHY_ADDR << 7); /* PHY Address */
305     data |= (reg_addr << 2); /* Reg Address */
306
307     for(i=14; i>0; i--){
308         if( (data & 0x8000) == 0 ) {
309             mii_write_0();
310         }
311         else {
312             mii_write_1();
313         }
314         data <<= 1;
315     }
316 }
317
318 /*"FUNC COMMENT"*****
319 * Outline : PHY モジュール SMI レジスタ値取得
320 *-----
321 * Include :
322 *-----
323 * Declaration : static void mii_reg_read( unsigned short *data )
324 *-----
325 * Function : PHY モジュール SMI レジスタ値を1ビットずつ取得します。
326 * : 下記条件を満たすように信号を入出力します。
327 * : ・MDC-High パルス幅:160ns(min)
328 * : ・MDC-Low パルス幅:160ns(min)
329 * : ・MDC-サイクル時間:400ns(min)
330 * : ・MDIO-出力遅延時間(from PHY):300ns(max)
331 *-----
332 * Argument : unsigned short *data : 0 : 取得した値の格納先アドレス
333 *-----
334 * Return Value: void
335 *-----
336 * Notice : システムに応じてウェイト時間を変更してください。
337 *"FUNC COMMENT END"*****

```


3.10 サンプルプログラムリスト"phy.c"(8)

```

338 static void mii_reg_read( unsigned short *data )
339 {
340     int i,j;
341     unsigned short reg_data;
342
343     /* 1 ビットずつデータを読み込む。 */
344     reg_data = 0;
345     for(i=16; i>0; i--){
346         for(j=MDC_WAIT; j>0; j--){
347             EtherC.PIR.LONG = 0x00000000;
348         }
349         for(j=MDC_WAIT; j>0; j--){
350             EtherC.PIR.LONG = 0x00000001;
351         }
352         reg_data <<= 1;
353         reg_data |= (EtherC.PIR.LONG & 0x00000008) >> 3;    /* MDI read */
354
355         for(j=MDC_WAIT; j>0; j--){
356             EtherC.PIR.LONG = 0x00000001;
357         }
358         for(j=MDC_WAIT; j>0; j--){
359             EtherC.PIR.LONG = 0x00000000;
360         }
361     }
362     *data = reg_data;
363 }
364
365 /*"FUNC COMMENT"*****
366 * Outline      : PHY モジュール SMI レジスタ値設定
367 *-----
368 * Include      :
369 *-----
370 * Declaration : static void mii_reg_write( unsigned short data )
371 *-----
372 * Function     : PHY モジュール SMI レジスタ値を1ビットずつ設定します。
373 *-----
374 * Argument     : unsigned short data : I : レジスタに設定する値
375 *-----
376 * Return Value: void
377 *-----
378 * Notice       :
379 *"FUNC COMMENT END"*****

```

3.11 サンプルプログラムリスト"phy.c"(9)

```

380 static void mii_reg_write( unsigned short data )
381 {
382     int i;
383
384     /* 1ビットずつデータを書き込む。*/
385     for(i=16; i>0; i--){
386         if( (data & 0x8000) == 0 ) {
387             mii_write_0();
388         }
389         else {
390             mii_write_1();
391         }
392         data <<= 1;
393     }
394 }
395
396 /*"FUNC COMMENT"*****
397 * Outline      : PHY モジュール SMI バス解放
398 *-----
399 * Include      :
400 *-----
401 * Declaration : static void mii_z( void )
402 *-----
403 * Function    : PHY モジュール SMI へのアクセスをデータ読込みに設定します
404 *              : 下記条件を満たすように信号を出力します。
405 *              : ・MDC-High パルス幅:160ns(min)
406 *              : ・MDC-Low パルス幅:160ns(min)
407 *              : ・MDC-サイクル時間:400ns(min)
408 *              : ・MDIO-セットアップ時間:10ns(min)
409 *              : ・MDIO-ホールド時間:10ns(min)
410 *-----
411 * Argument    : void
412 *-----
413 * Return Value: void
414 *-----
415 * Notice      : システムに応じてウェイト時間を変更してください。
416 "FUNC COMMENT END"*****/
417 static void mii_z( void )
418 {
419     int j;
420     for(j=MDC_WAIT; j>0; j--){
421         EtherC.PIR.LONG = 0x00000000;
422     }
423     for(j=MDC_WAIT; j>0; j--){
424         EtherC.PIR.LONG = 0x00000001;
425     }
426     for(j=MDC_WAIT; j>0; j--){
427         EtherC.PIR.LONG = 0x00000001;
428     }
429     for(j=MDC_WAIT; j>0; j--){
430         EtherC.PIR.LONG = 0x00000000;
431     }
432 }
433

```

3.12 サンプルプログラムリスト"phy.c"(10)

```

434  /*"FUNC COMMENT"*****
435  * Outline      : PHY モジュール SMI TA(10)ビット出力
436  *-----
437  * Include      :
438  *-----
439  * Declaration : static void mii_ta10( void )
440  *-----
441  * Function     : PHY モジュール SMI へ1,0 を出力します
442  *-----
443  * Argument     : void
444  *-----
445  * Return Value: void
446  *-----
447  * Notice      :
448  *"FUNC COMMENT END"*****/
449  static void mii_ta10( void )
450  {
451      mii_write_1();
452      mii_write_0();
453  }
454
455  /*"FUNC COMMENT"*****
456  * Outline      : PHY モジュール SMI 1 ビット(1) 出力
457  *-----
458  * Include      :
459  *-----
460  * Declaration : static void mii_write_1( void )
461  *-----
462  * Function     : PHY モジュール SMI に 1 を出力します
463  *               : 下記条件を満たすように信号を出力します。
464  *               :   ・MDC-High パルス幅:160ns(min)
465  *               :   ・MDC-Low パルス幅:160ns(min)
466  *               :   ・MDC-サイクル時間:400ns(min)
467  *               :   ・MDIO-セットアップ時間:10ns(min)
468  *               :   ・MDIO-ホールド時間:10ns(min)
469  *-----
470  * Argument     : void
471  *-----
472  * Return Value: void
473  *-----
474  * Notice      : システムに応じてウェイト時間を変更してください。
475  *"FUNC COMMENT END"*****/
    
```

3.13 サンプルプログラムリスト"phy.c"(11)

```

476 static void mii_write_1( void )
477 {
478     int j;
479     for(j=MDC_WAIT; j>0; j--){
480         EtherC.PIR.LONG = 0x00000006;
481     }
482     for(j=MDC_WAIT; j>0; j--){
483         EtherC.PIR.LONG = 0x00000007;
484     }
485     for(j=MDC_WAIT; j>0; j--){
486         EtherC.PIR.LONG = 0x00000007;
487     }
488     for(j=MDC_WAIT; j>0; j--){
489         EtherC.PIR.LONG = 0x00000006;
490     }
491 }
492
493 /*"FUNC COMMENT"*****
494 * Outline      : PHY モジュール SMI 1 ビット(0) 出力
495 *-----
496 * Include      :
497 *-----
498 * Declaration : static void mii_write_0( void )
499 *-----
500 * Function     : PHY モジュール SMI に 0 を出力します。
501 *               : 下記条件を満たすように信号を出力します。
502 *               :   ・MDC-High パルス幅:160ns(min)
503 *               :   ・MDC-Low パルス幅:160ns(min)
504 *               :   ・MDC-サイクル時間:400ns(min)
505 *               :   ・MDIO-セットアップ時間:10ns(min)
506 *               :   ・MDIO-ホールド時間:10ns(min)
507 *-----
508 * Argument     : void
509 *-----
510 * Return Value: void
511 *-----
512 * Notice       : システムに応じてウェイト時間を変更してください。
513 *"FUNC COMMENT END"*****/
514 static void mii_write_0( void )
515 {
516     int j;
517     for(j=MDC_WAIT; j>0; j--){
518         EtherC.PIR.LONG = 0x00000002;
519     }
520     for(j=MDC_WAIT; j>0; j--){
521         EtherC.PIR.LONG = 0x00000003;
522     }
523     for(j=MDC_WAIT; j>0; j--){
524         EtherC.PIR.LONG = 0x00000003;
525     }
526     for(j=MDC_WAIT; j>0; j--){
527         EtherC.PIR.LONG = 0x00000002;
528     }
529 }
530 /* End of File */

```

3.14 サンプルプログラムリスト"phy.h"(1)

```

1  /*"FILE COMMENT"*****
2  *
3  *   System Name   : SH7619 Sample Program
4  *   File Name    : phy.h
5  *   Contents     : 内蔵 PHY モジュール初期設定例
6  *   Version      : 1.00.00
7  *   Model        : M3A-HS19
8  *   CPU          : SH7619
9  *   Compiler     : SHC9.01.00
10 *   note         :
11 *               <注意事項>
12 *               本サンプルプログラムはすべて参考資料であり、
13 *               その動作を保証するものではありません。
14 *               本サンプルプログラムはお客様のソフトウェア開発時の
15 *               技術参考資料としてご利用ください。
16 *
17 *   Copyright (C) 2007 Renesas Technology Corp. All Rights Reserved
18 *   and Renesas Solutions Corp. All Rights Reserved
19 *
20 *   history      :2007.06.29 ver.1.00.00
21 *"FILE COMMENT END"*****/
22 #ifndef _PHY_H
23 #define _PHY_H
24
25 /* **** マクロ定義 **** */
26 #define NEGO_FAIL      0
27 #define HALF_10M      1
28 #define FULL_10M      2
29 #define HALF_TX       3
30 #define FULL_TX       4
31 /* **** プロトタイプ宣言 **** */
32 void phyif_init( void );
33 int phy_autonego( void );
34
35 #endif
36
37 /* End of File */
    
```

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 Rev7.00
(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.5.00
(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/>
- お問い合わせ先
<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	2007.12.28	—	新規作成

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、「外国為替及び外国貿易法」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頂きますとともに、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下さい。
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下さい。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下さい。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下さい。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）を行なうもの。
 - 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致します。
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致します。
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下さい。