

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ひ申し上げます。

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。

# SH7710 グループ

## イーサネット PHY-LSI 自動交渉設定例

### 要旨

この資料は、SH7710/7712/7713 にイーサネット PHY-LSI を接続し、相手先と自動交渉するための設定例を示します。

### 動作確認デバイス

SH7712

DP83847: National Semiconductor Corporation 製 Ethernet PHY-LSI

### 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 .....	10
4. 参考ドキュメント .....	25
5. ホームページとサポート窓口 .....	25

## 1. はじめに

### 1.1 仕様

- 本応用例では、PHY-LSI の自動交渉結果を取得します。その結果に基づき、接続モード(全二重モード又は半二重モード)を EtherC に設定します。
- PHY 自動交渉結果の取得では、PHY-LSI の自動交渉機能により決定したモードを取得します。
- イーサネット PHY-LSI には、National Semiconductor Corporation の DP83847 を使用しています。

### 1.2 使用機能

- イーサネットコントローラ (EtherC)

### 1.3 適用条件

- マイコン: SH7712(HD6417712)
- 動作周波数:
  - 内部クロック 198.00MHz
  - バスクロック 66.00MHz
  - 周辺クロック 33.00MHz
- 統合開発環境: ルネサステクノロジ  
High-performance Embedded Workshop Ver.4.03.00.001
- 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 V.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=sh3dsp -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

- SH7710 グループ アプリケーションノート SH7710/7712/7713 初期設定例  
本資料の参考プログラムは、上記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。こちらも合わせて参照ください。

## 2. 応用例の説明

本応用例では、イーサネット PHY-LSI により自動交渉を行います。自動交渉結果は EtherC の PHY 部インタフェースレジスタ(PIR)を経由して読み出されます。

### 2.1 使用機能の動作概要

Ethernet PHY-LSIの各種MIIレジスタへのアクセスは、EtherCのPIRを経由して行います。図 1にMII管理フレームフォーマット、図 2～図 4にMIIレジスタへのアクセス方法を示します。パルス幅やロックサイクル時間に制限がありますので、ご注意ください。

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	00001	RRRRR	Z0	D..D	Z
ライト	1..1	01	01	00001	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。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1 MII 管理フレームフォーマット

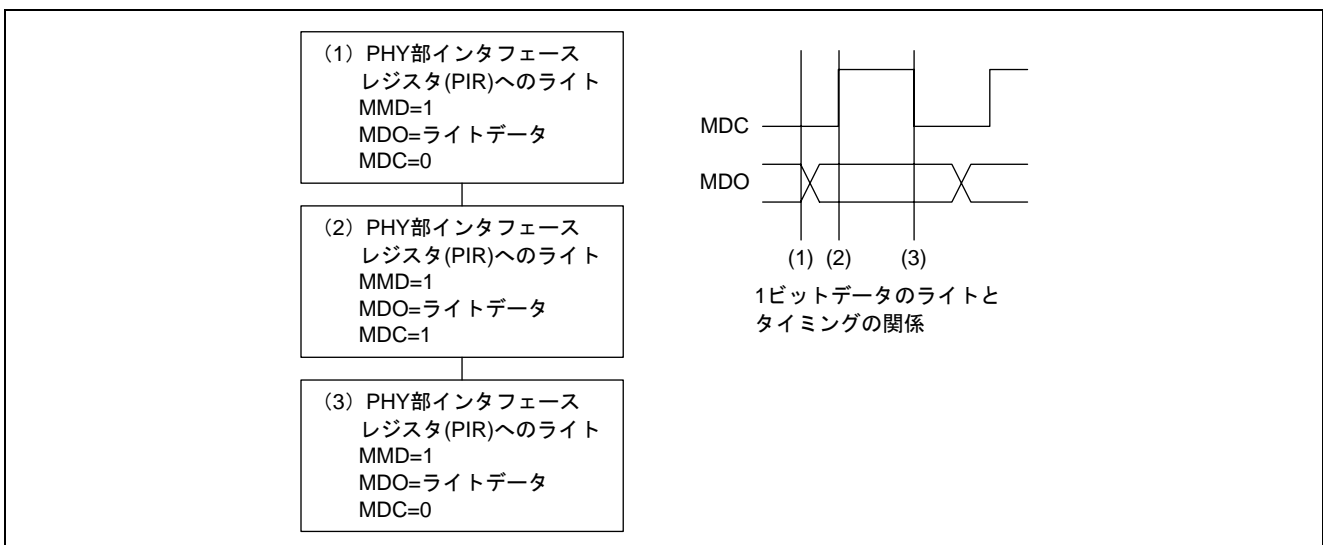


図2 1ビットデータのライトフロー

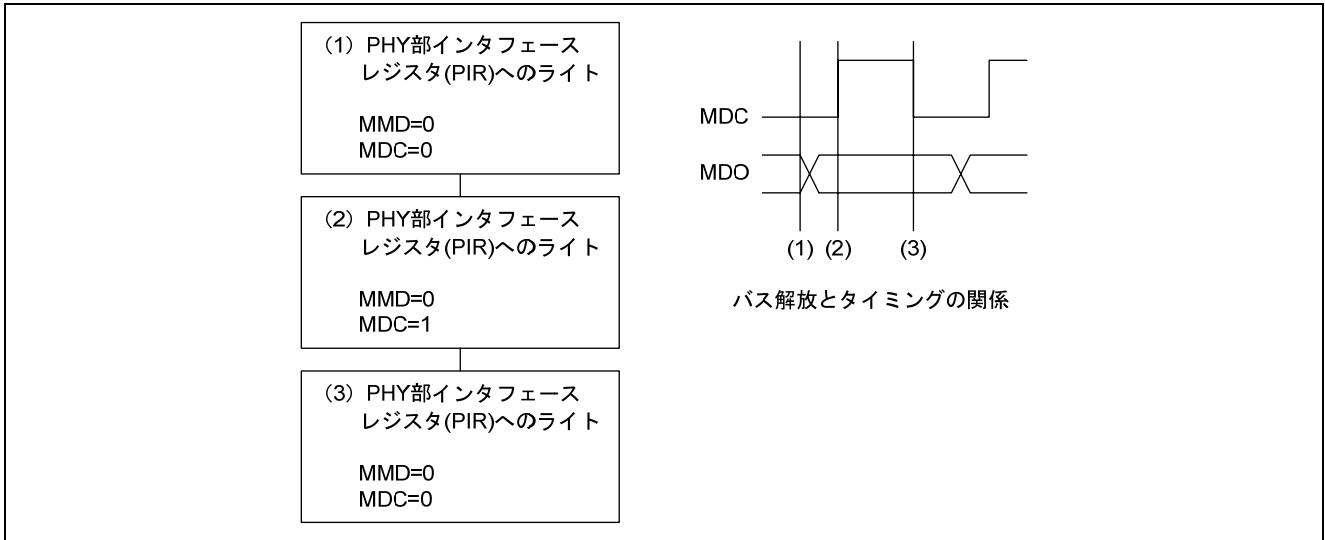


図3 バス解放フロー (Z 出力)

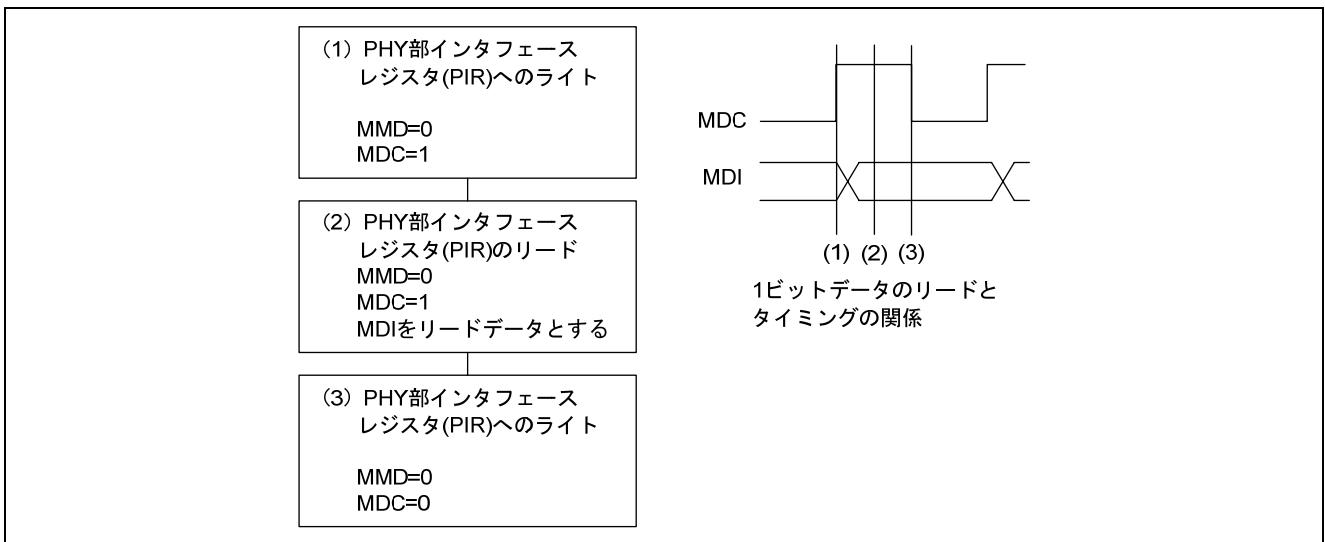


図4 1ビットデータのリードフロー

### 2.2 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「SH7710 グループ アプリケーションノート SH7710/7712/7713 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、PHY-LSIの自動交渉結果に基づきEtherCに接続モード(全二重モードまたは半二重モード)を設定します。図5にmain関数のフローを示します。

- phy.c

自動交渉結果の取得関数（phy\_autonego関数）が記述されています。図6にphy\_autonego関数のフローを示します。またphy\_autonego関数内で呼び出されているMIIレジスタのリード関数（phy\_reg\_read関数）とライト関数(phy\_reg\_write関数)、それらの下位関数のフローを図7～図11に示します。

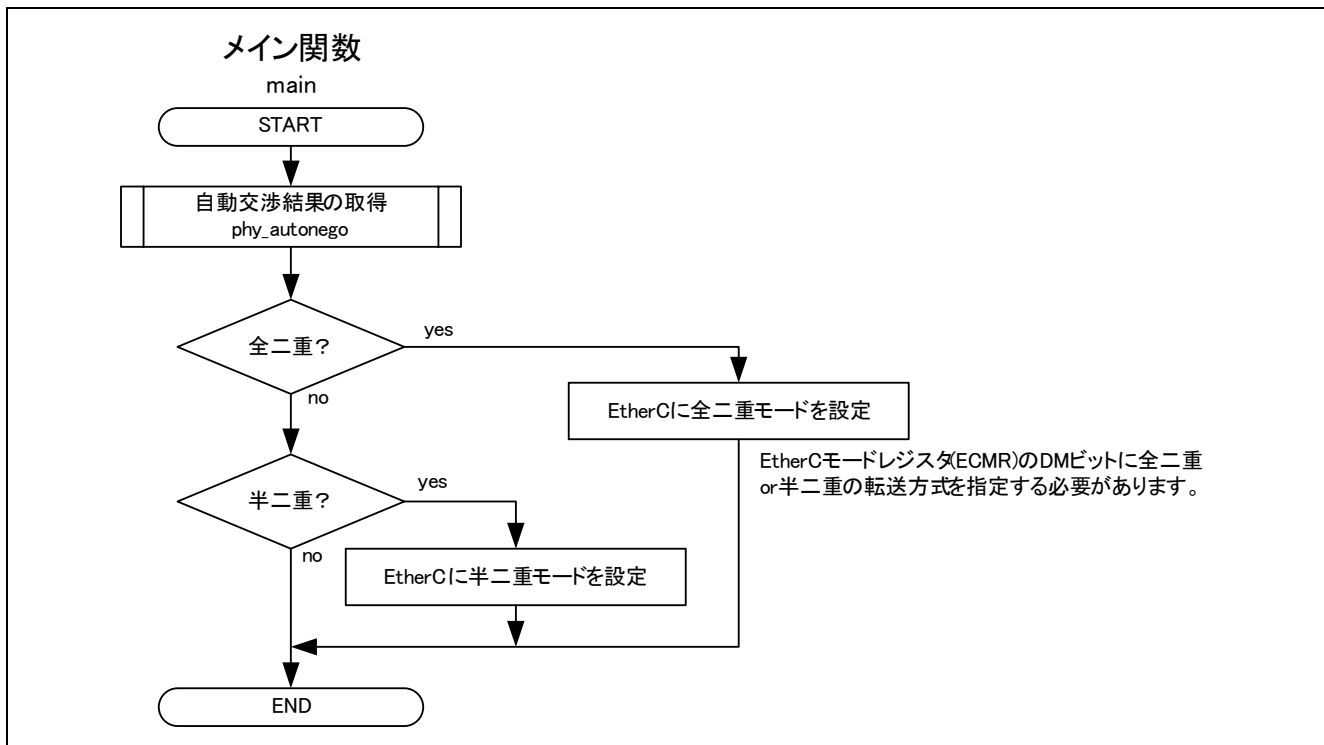


図5 参考プログラムの処理フロー

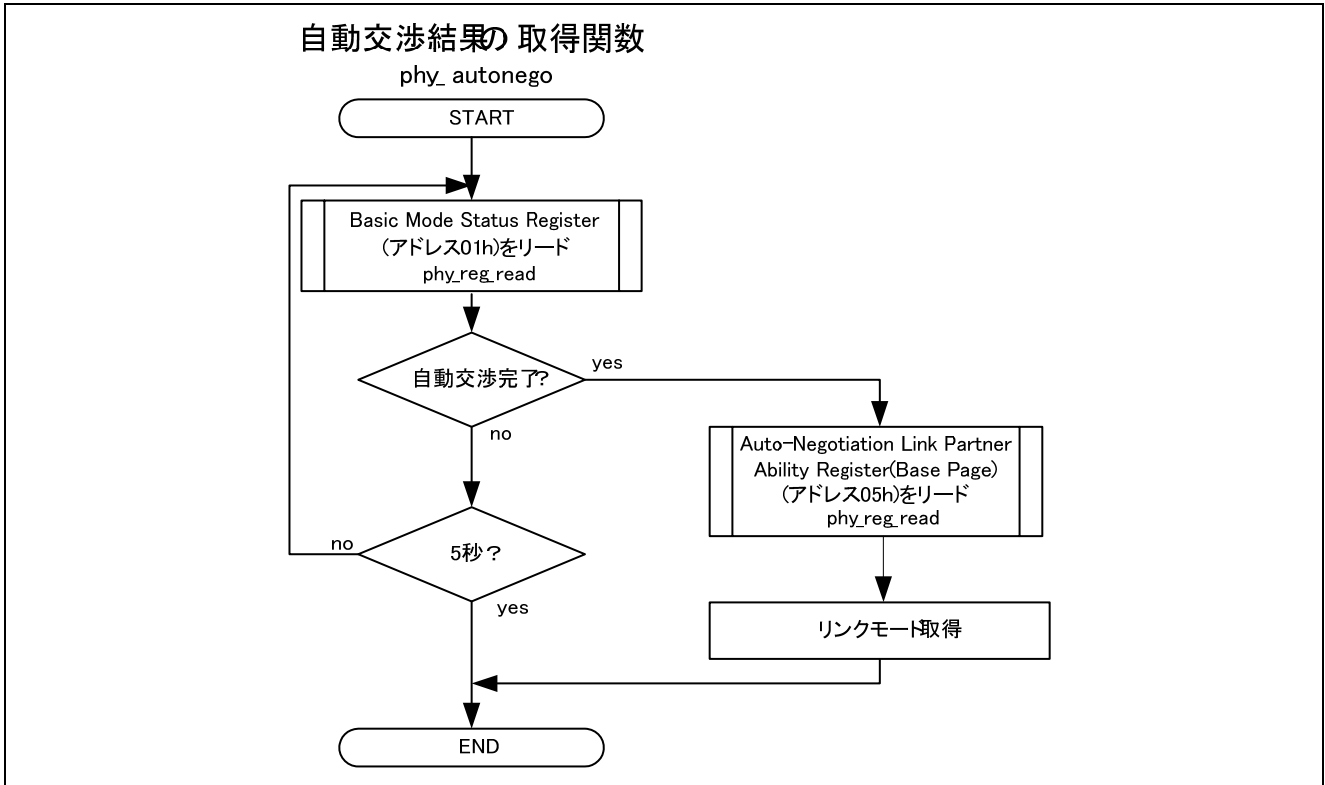


図6 参考プログラムの処理フロー(2)

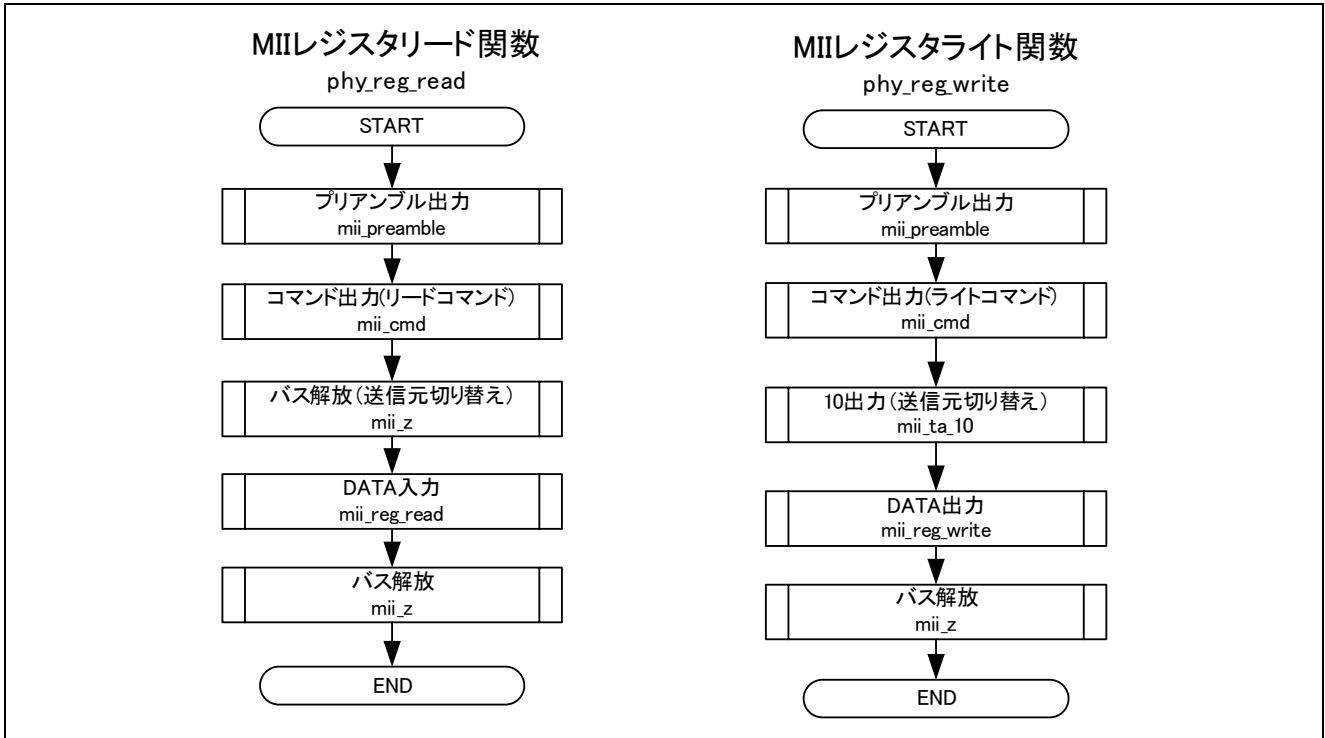


図7 MII レジスタアクセスの処理フロー (1)



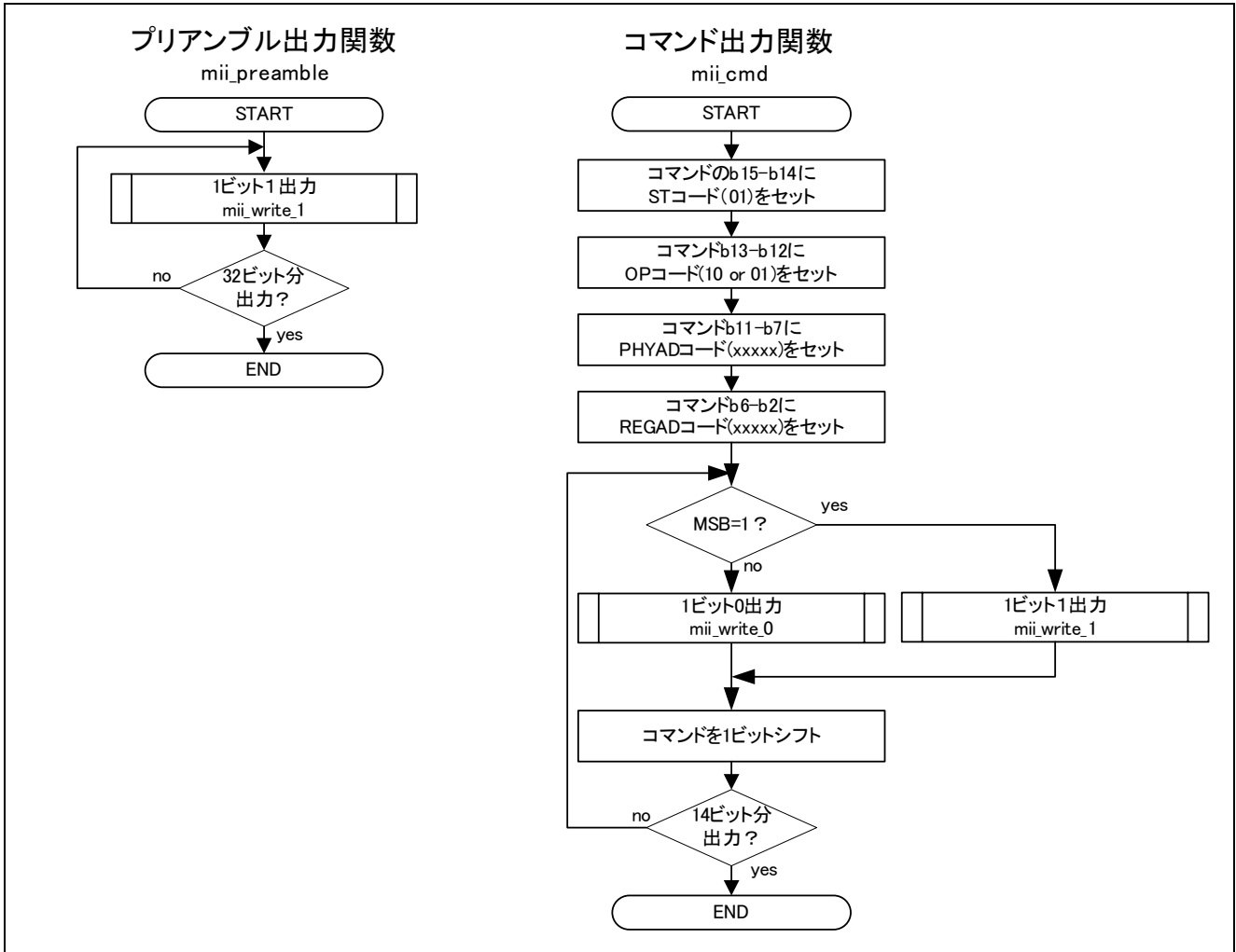


図8 MII レジスタアクセスの処理フロー (2)

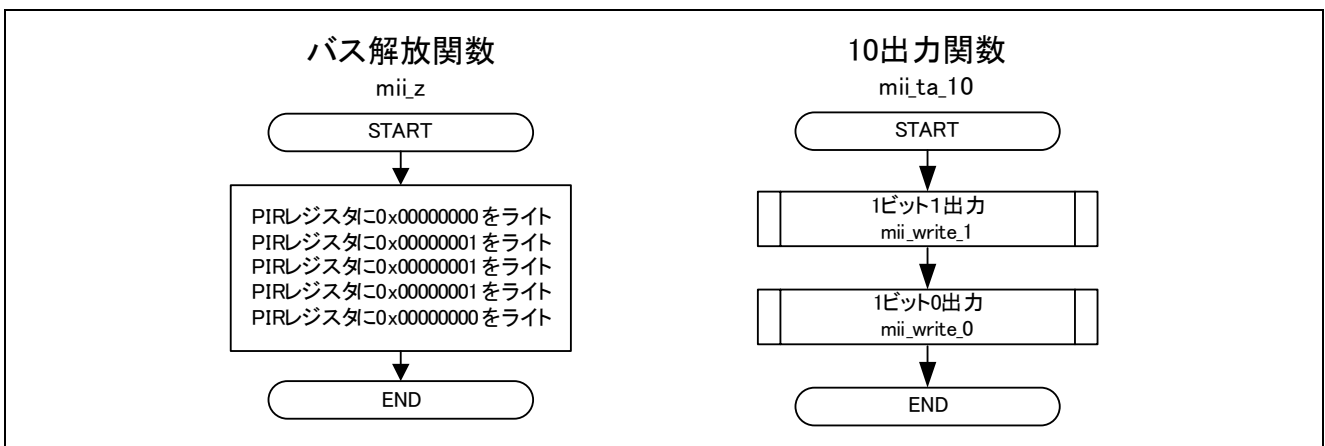


図9 MII レジスタアクセスの処理フロー (3)

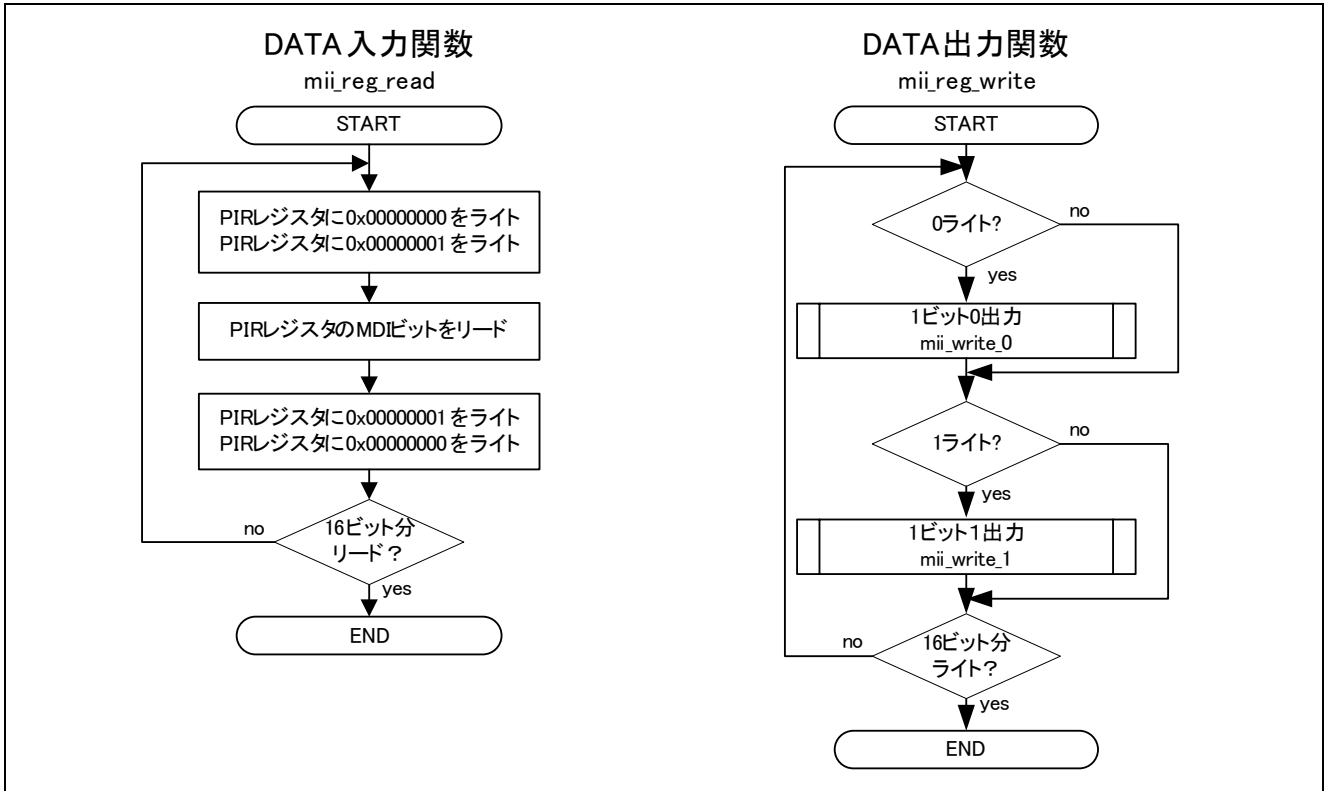


図10 MII レジスタアクセスの処理フロー (4)

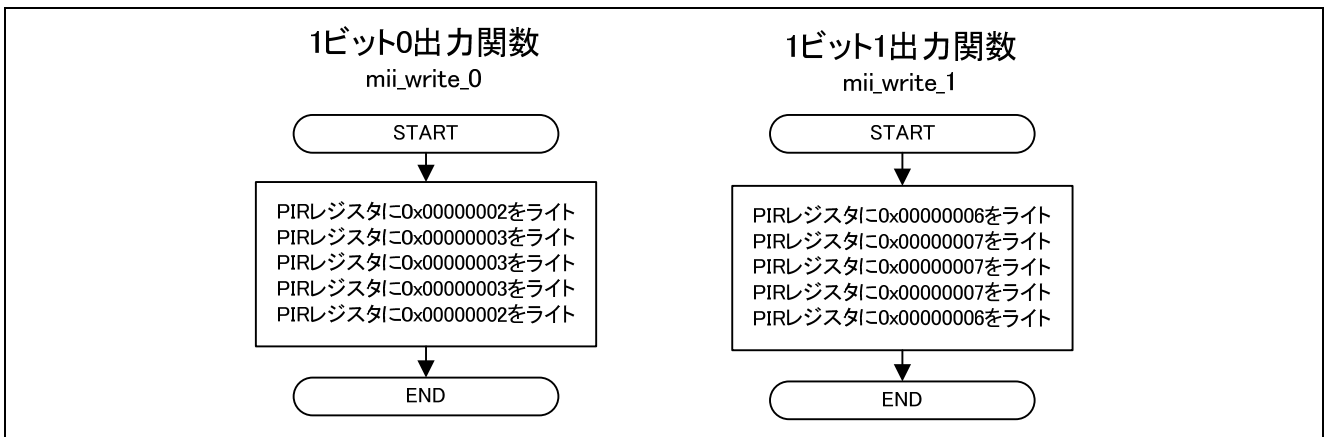


図11 MII レジスタアクセスの処理フロー (5)

## 2.3 参考プログラムにおける設定内容

表 1に参考プログラムでの設定を示します。

表1 参考プログラムでの設定

項目	設定内容
使用する PHY-LSI	National Semiconductor Corporation 製 DP83847
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) および並列検出 (パラレル・ディテクション)
使用する MII レジスタ	Basic Mode Status Register(アドレス:01h) Auto-Negotiation LinkPartner Ability Register(アドレス:05h)

## 2.4 参考プログラム使用時の注意点

参考プログラムでは、`phy_autonego` 関数内で自動交渉が完了するまで、最大で 5 秒の待ちが発生します。(通常、自動交渉は数秒で完了します。) ご注意ください。

### 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   : SH7712 Sample Program
4  *   File Name    : main.c
5  *   Contents     : PHY-LSI の自動交渉設定例
6  *   Version      : 1.00.00
7  *   Model        : MS7712SE01
8  *   CPU          : SH7712
9  *   Compiler     : SHC9.1.1.0
10 *   OS           : None
11 *
12 *   note         : PHY-LSI を使い自動交渉を行います。
13 *                : PHY-LSI は自動交渉を行いリンクモード
14 *                : 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。
15 *                : 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。
16 *
17 *
18 *                <注意事項>
19 *                本サンプルプログラムはすべて参考資料であり、
20 *                その動作を保証するものではありません。
21 *                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
22 *                技術参考資料としてご利用ください。
23 *
24 *   Copyright (C) 2007 Renesas Technology Corp. All Rights Reserved
25 *   and Renesas Solutions Corp. All Rights Reserved
26 *
27 *   history      : 2007.11.07 ver.1.00.00
28 *                :
29 *"FILE COMMENT END"*****/
30 #include "iodefine.h"
31 #include "phy.h"
32
33 /* ==== プロトタイプ宣言 ==== */
34 void main(void);
    
```

### 3.2 サンプルプログラムリスト"main.c"(2)

```

34  /*"FUNC COMMENT"*****
35  * ID          :
36  * Outline     : PHY-LSI 自動交渉メイン関数
37  *-----
38  * Include     : #include "iodefine.h"
39  *-----
40  * Declaration : void main(void)
41  *-----
42  * Function    : PHY-LSI を使い自動交渉を行います。
43  *             : PHY-LSI は自動交渉を行いリンクモードを決定しますので、
44  *             : 本サンプルでは結果を読み出しています。
45  *             : 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。
46  *-----
47  * Argument    : void
48  *-----
49  * ReturnValue : void
50  *-----
51  * Notice      :
52  *"FUNC COMMENT END"*****/
53  void main(void)
54  {
55      int link ;
56
57      /* ==== PHY 交渉結果の取得(0 チャンネル) ==== */
58      link = phy_autonego(0); /* channel 0 を選択 */
59
60      /* ==== EtherC デュプレックスモードの設定 ==== */
61      if(link == FULL_TX || link == FULL_10M){
62          MAC0.ECMR.BIT.DM = 1;
63      }
64      else if(link == HALF_TX || link == HALF_10M){
65          MAC0.ECMR.BIT.DM = 0;
66      }
67      else{
68          ; /* error */
69      }
70
71      /* ==== メインループ ==== */
72      while(1){
73          ;
74      }
75  }
76  }
77
78  /* 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   : SH7712 Sample Program
4  *   File Name    : phy.c
5  *   Contents     : PHY-LSI 自動交渉初期設定例
6  *   Version      : 1.00.00
7  *   Model        : MS7712SE01
8  *   CPU          : SH7712
9  *   Compiler     : SHC9.1.1.0
10 *   OS           : None
11 *
12 *   note         : 自動交渉を行い、結果を読み出すための関数を用意してあります。
13 *               :
14 *
15 *               <注意事項>
16 *               本サンプルプログラムはすべて参考資料であり、
17 *               その動作を保証するものではありません。
18 *               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
19 *               技術参考資料としてご利用ください。
20 *
21 *   Copyright (C) 2007 Renesas Technology Corp. All Rights Reserved
22 *   and Renesas Solutions Corp. All Rights Reserved
23 *
24 *   history      :2007.11.07 ver.1.00.00
25 *"FILE COMMENT END"*****/
26 #include "iodefine.h"
27 #include "phy.h"
28
29 /* **** プロトタイプ宣言 **** */
29 static unsigned short  phy_reg_read(int ch, unsigned short reg_addr);
30 static void phy_reg_write(int ch, unsigned short reg_addr, unsigned short data);
31 static void mii_preamble(int ch);
32 static void mii_cmd(int ch, unsigned short reg_addr, int option);
33 static void mii_reg_read(int ch, unsigned short *data);
34 static void mii_reg_write(int ch, unsigned short data);
35 static void mii_z(int ch);
36 static void mii_ta_10(int ch);
37 static void mii_write_1(int ch);
38 static void mii_write_0(int ch);
39
40 /* **** マクロ定義 **** */
41 /* PHY-LSI の MII レジスタ */
42 #define BASIC_MODE_CONTROL_REG      (0x0000)
43 #define BASIC_MODE_STATUS_REG       (0x0001)
44 #define PHY_IDENTIFIER1_REG         (0x0002)
45 #define PHY_IDENTIFIER2_REG         (0x0003)
46 #define AN_ADVERTISEMENT_REG        (0x0004)
47 #define AN_LINK_PARTNER_ABILITY_REG (0x0005)
48 #define AN_EXPANSION_REG             (0x0006)

```

### 3.4 サンプルプログラムリスト"phy.c"(2)

```

50  /* MII アクセス用 */
51  #define PHY_ST                1
52  #define PHY_WRITE             1
53  #define PHY_READ              2
54  #define PHY_ADDR0            1
55  #define PHY_ADDR1            1
56  #define MDC_WAIT              3 /* 2xPφxMDC_WAIT > 400ns/4 ;Pφは周辺クロック */
57
58  /*"FUNC COMMENT"*****
59  * Outline      : PHY リンクの交渉結果検出
60  *-----
61  * Include      : #include "iodefine.h"
62  *-----
63  * Declaration  : int phy_autonego(int ch);
64  *-----
65  * Function     : 自動交渉結果を読み込み、戻り値で返します。
66  *               : 最大 5 秒間、自動交渉が終わるまで本関数内で待ちます。
67  *               :
68  *-----
69  * Argument     : int ch : I : PHY-LSI のチャンネル番号(0 または 1)
70  *-----
71  * Return Value : 4(FULL_TX) :100Mbps 全二重
72  *               : 3(HALF_TX) :100Mbps 半二重
73  *               : 2(FULL_10M):10Mbps 全二重
74  *               : 1(HALF_10M):10Mbps 半二重
75  *               : 0(NEGO_FAIL) :交渉失敗
76  *-----
77  * 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。
78  *               :
79  *               :
80  *               :
81  *-----
82  *"FUNC COMMENT END"*****/
83  int phy_autonego(int ch)
84  {
85      unsigned short data0,data1;
86      int link = NEGO_FAIL;
87      volatile int t;
88      int i;

```

### 3.5 サンプルプログラムリスト”phy.c”(3)

```

80     /* ==== 自動交渉の終了待ちループ ==== */
81     for( i=0; i<500; i++){
82
83         /* ---- 10ms ウェイト ---- */
84         t=290000;
85         while( --t){           /* 10ms ウェイトカウンタ @200MHz */
86             ;
87         }
88         /* ---- 自動交渉完了かを確認 ---- */
89         data0 = phy_reg_read(ch, BASIC_MODE_STATUS_REG);
90
91         if(data0 & 0x0020){ /* 自動交渉完了 */
92
93             /* ---- 自動交渉結果取得 ---- */
94             data1 = phy_reg_read(ch, AN_LINK_PARTNER_ABILITY_REG);
95                 /* AN_LINK_PARTNER_ABILITY_REG 自動交渉結果 *
96                 * 相手が自動交渉に対応していない場合は*
97                 * 並列検出が行われるが、結果はこのレジ*
98                 * スタに反映される。                */
99             /* ---- 結果判別->交渉終了で break ---- */
100            if( data1&0x0100 ){
101                link = FULL_TX;
102            }
103            else if( data1&0x0080 ){
104                link = HALF_TX;
105            }
106            else if( data1&0x0040 ){
107                link = FULL_10M;
108            }
109            else if( data1&0x0020 ){
110                link = HALF_10M;
111            }
112            else{
113                link = NEGO_FAIL;
114            }
115            break;
116        }
117    }
118    return link;
119 }

```



### 3.6 サンプルプログラムリスト"phy.c"(4)

```

120  /*"FUNC COMMENT"*****
121  * Outline      : PHY-LSI の各種 MII レジスタリード
122  *-----
123  * Include      :
124  *-----
125  * Declaration : static unsigned short  phy_reg_read(int ch, unsigned short reg_addr)
126  *-----
127  * Function     : PHY-LSI の各種 MII レジスタ値を取得します。
128  *-----
129  *
130  * Argument     : int ch: I : PHY-LSI のチャンネル番号(0 または 1)
131  *               : unsigned short reg_addr : I : 値を読み込む PHY-LSI レジスタアドレス
132  *-----
133  * Return Value : 取得したレジスタの値
134  *-----
135  * Notice       :
136  *"FUNC COMMENT END"*****/
137  static unsigned short  phy_reg_read(int ch, unsigned short reg_addr)
138  {
139      unsigned short  data;
140
141      mii_preamble(ch);
142      mii_cmd(ch, reg_addr, PHY_READ);
143      mii_z(ch);
144      mii_reg_read(ch, &data);
145      mii_z(ch);
146
147      return data;
148  }
149  /*"FUNC COMMENT"*****
150  * Outline      : PHY-LSI の各種 MII レジスタライト
151  *-----
152  * Include      :
153  *-----
154  * Declaration : static void phy_reg_write(int ch, unsigned short reg_addr,
155  *               :               unsigned short data )
156  *-----
157  * Function     : PHY-LSI の各種 MII レジスタに値を設定します。
158  *-----
159  * Argument     : int ch: I : PHY-LSI のチャンネル番号(0 または 1)
160  *               : unsigned short reg_addr : I : 値を書き込む PHY-LSI レジスタアドレス
161  *               : unsigned short data   : I : PHY-LSI のレジスタに設定する値
162  *-----
163  * Return Value: void
164  *-----
165  * Notice       :
166  *"FUNC COMMENT END"*****/
167  static void phy_reg_write(int ch, unsigned short reg_addr, unsigned short data)
168  {
169      mii_preamble(ch);
170      mii_cmd(ch, reg_addr, PHY_WRITE);
171      mii_ta_10(ch);
172      mii_reg_write(ch, data);
173      mii_z(ch);
174  }

```

### 3.7 サンプルプログラムリスト"phy.c"(5)

```

174  /*"FUNC COMMENT"*****
175  * Outline      : PHY-LSI の各種 MII レジスタへのアクセス準備
176  *-----
177  * Include      :
178  *-----
179  * Declaration : static void mii_preamble(int ch)
180  *-----
181  * Function     : PHY-LSI レジスタにアクセスするための前準備として、
182  *               : MII ブロックに 32 ビット分 1 を出力します。
183  *-----
184  * Argument     : int ch: I : PHY-LSI のチャンネル番号(0 または 1)
185  *-----
186  * Return Value: void
187  *-----
188  * Notice      :
189  *"FUNC COMMENT END"*****/
190  static void mii_preamble(int ch)
191  {
192      short i;
193
194      i = 32;
195      while( i > 0 ) {
196          /* MII(Media Independent Interface)ブロックに 1 を出力。*/
197          mii_write_1(ch);
198          i--;
199      }
200  }
    
```

### 3.8 サンプルプログラムリスト"phy.c"(6)

```

201  /*"FUNC COMMENT"*****
202  * Outline      : PHY-LSI の各種 MII レジスタモード設定
203  *-----
204  * Include      :
205  *-----
206  * Declaration : static void mii_cmd(int ch, unsigned short reg_addr, int option)
207  *             :
208  *-----
209  * Function     : PHY-LSI の各種 MII レジスタの R/W モードを設定します。
210  *-----
211  * Argument    : int ch: I : PHY-LSI のチャンネル番号(0または1)
212  *              : unsigned short reg_addr : I : PHY-LSI のレジスタアドレス
213  *              : int option              : I : R/W モード指定
214  *-----
215  * Return Value: void
216  *-----
217  * Notice      :
218  *"FUNC COMMENT END"*****/
219  static void mii_cmd(int ch, unsigned short reg_addr, int option)
220  {
221      int i;
222      unsigned short data;
223
224      data = 0;
225      data = (PHY_ST << 14);          /* ST code */
226      if( option == PHY_READ ) {
227          data |= (PHY_READ << 12);  /* OP code(RD) */
228      }
229      else {
230          data |= (PHY_WRITE << 12); /* OP code(WT) */
231      }
232
233      if(ch==0){
234          data |= (PHY_ADDR0 << 7);  /* PHY Address */
235      }
236      else if(ch==1){
237          data |= (PHY_ADDR1 << 7);  /* PHY Address */
238      }
239      else{
240          /* DO NOTHING */
241      }
242      data |= (reg_addr << 2);       /* Reg Address */
243
244      for(i=14; i>0; i--){
245          if( (data & 0x8000) == 0 ) {
246              mii_write_0(ch);
247          }
248          else {
249              mii_write_1(ch);
250          }
251          data <<= 1;
252      }
253  }
    
```

### 3.9 サンプルプログラムリスト"phy.c"(7)

```

254  /*"FUNC COMMENT"*****
255  * Outline      : PHY-LSI の各種 MII レジスタ値取得
256  *-----
257  * Include      :
258  *-----
259  * Declaration : static void mii_reg_read(int ch, unsigned short *data)
260  *-----
261  * Function     : PHY-LSI の各種 MII レジスタ値を 1 ビットずつ取得します。
262  *              : 下記条件を満たすように信号を入出力します。
263  *              : ・MDC-High パルス幅 : 160ns(min)
264  *              : ・MDC-Low パルス幅 : 160ns(min)
265  *              : ・MDC-サイクル時間 : 400ns(min)
266  *              : ・MDIO-出力遅延時間(from PHY) : 300ns(max)
267  *-----
268  * Argument     : int ch: I : PHY-LSI のチャンネル番号(0 または 1)
269  *              : unsigned short *data : 0 : 取得した値の格納先アドレス
270  *-----
271  * Return Value: void
272  *-----
273  * Notice       : システムに応じてウェイト時間を変更してください。
274  *"FUNC COMMENT END"*****/
275  static void mii_reg_read(int ch, unsigned short *data)
276  {
277      int i,j;
278      unsigned short reg_data;
279
280      /* 1 ビットずつデータを読み込む。*/
281      reg_data = 0;
282      if(ch==0){
283          for(i=16; i>0; i--){
284              for(j=MDC_WAIT; j>0; j--){
285                  MAC0.PIR.LONG = 0x00000000;
286              }
287              for(j=MDC_WAIT; j>0; j--){
288                  MAC0.PIR.LONG = 0x00000001;
289              }
290              reg_data <<= 1;
291              reg_data |= (MAC0.PIR.LONG & 0x00000008) >> 3; /* MDI read */
292
293              for(j=MDC_WAIT; j>0; j--){
294                  MAC0.PIR.LONG = 0x00000001;
295              }
296              for(j=MDC_WAIT; j>0; j--){
297                  MAC0.PIR.LONG = 0x00000000;
298              }
299          }
300      }
    
```

## 3.10 サンプルプログラムリスト"phy.c"(8)

```

301     else if(ch==1){
302         for(i=16; i>0; i--){
303             for(j=MDC_WAIT; j>0; j--){
304                 MAC1.PIR.LONG = 0x00000000;
305             }
306             for(j=MDC_WAIT; j>0; j--){
307                 MAC1.PIR.LONG = 0x00000001;
308             }
309             reg_data <<= 1;
310             reg_data |= (MAC1.PIR.LONG & 0x00000008) >> 3; /* MDI read */
311
312             for(j=MDC_WAIT; j>0; j--){
313                 MAC1.PIR.LONG = 0x00000001;
314             }
315             for(j=MDC_WAIT; j>0; j--){
316                 MAC1.PIR.LONG = 0x00000000;
317             }
318         }
319     }
320     else{
321         /* DO NOTHING */
322     }
323     *data = reg_data;
324 }
325
326 /*"FUNC COMMENT"*****
327 * Outline      : PHY -LSI の各種 MII レジスタ値設定
328 *-----
329 * Include      :
330 *-----
331 * Declaration  : static void mii_reg_write(int ch,unsigned short data)
332 *-----
333 * Function     : PHY-LSI の各種 MII レジスタ値を 1 ビットずつ設定します。
334 *-----
335 * Argument    : int ch: I : PHY-LSI のチャンネル番号(0または1)
336 *              : unsigned short data : I : レジスタに設定する値
337 *-----
338 * Return Value: void
339 *-----
340 * Notice      :
341 *"FUNC COMMENT END"*****/
342 static void mii_reg_write(int ch, unsigned short data)
343 {
344     int i;
345
346     /* 1 ビットずつデータを書き込む。*/
347     for(i=16; i>0; i--){
348         if( (data & 0x8000) == 0 ) {
349             mii_write_0(ch);
350         }
351         else {
352             mii_write_1(ch);
353         }
354         data <<= 1;
355     }
356 }
    
```

### 3.11 サンプルプログラムリスト"phy.c"(9)

```

357  /*"FUNC COMMENT"*****
358  * Outline      : MII バス開放
359  *-----
360  * Include      :
361  *-----
362  * Declaration : static void mii_z(int ch)
363  *-----
364  * Function     : MII へのアクセスをライトに設定します。
365  *              : 下記条件を満たすように信号を出力します。
366  *              : ・MDC-High パルス幅 : 160ns(min)
367  *              : ・MDC-Low パルス幅 : 160ns(min)
368  *              : ・MDC-サイクル時間 : 400ns(min)
369  *              : ・MDIO-セットアップ時間 : 10s(min)
370  *              : ・MDIO-ホールド時間 : 10ns(min)
371  *-----
372  * Argument     : int ch : I : PHY-LSI のチャンネル番号(0 または 1)
373  *-----
374  * Return Value: void
375  *-----
376  * Notice       : システムに応じてウエイト時間を変更してください。
377  *"FUNC COMMENT END"*****/
378  static void mii_z(int ch)
379  {
380      int j;
381
382      if(ch==0){
383          for(j=MDC_WAIT;j>0;j--){
384              MAC0.PIR.LONG = 0x00000000;
385          }
386          for(j=MDC_WAIT;j>0;j--){
387              MAC0.PIR.LONG = 0x00000001;
388          }
389          for(j=MDC_WAIT;j>0;j--){
390              MAC0.PIR.LONG = 0x00000001;
391          }
392          for(j=MDC_WAIT;j>0;j--){
393              MAC0.PIR.LONG = 0x00000000;
394          }
395      }
396      else if(ch==1){
397          for(j=MDC_WAIT;j>0;j--){
398              MAC1.PIR.LONG = 0x00000000;
399          }
400          for(j=MDC_WAIT;j>0;j--){
401              MAC1.PIR.LONG = 0x00000001;
402          }
403          for(j=MDC_WAIT;j>0;j--){
404              MAC1.PIR.LONG = 0x00000001;
405          }
406          for(j=MDC_WAIT;j>0;j--){
407              MAC1.PIR.LONG = 0x00000000;
408          }
409      }

```

## 3.12 サンプルプログラムリスト"phy.c"(10)

```

410     else{
411         /* DO NOTHING */
412     }
413 }
414
415 /*"FUNC COMMENT"*****
416 * Outline      : MII TA(1,0)ビット出力
417 *-----
418 * Include      :
419 *-----
420 * Declaration  : static void mii_ta_10(int ch)
421 *-----
422 * Function     : MII に1,0 を出力します。
423 *-----
424 * Argument     : int ch: I : PHY-LSI のチャンネル番号(0 または1)
425 *-----
426 * Return Value: void
427 *-----
428 * Notice      :
429 *"FUNC COMMENT END"*****/
430 static void mii_ta_10(int ch)
431 {
432     mii_write_1(ch);
433     mii_write_0(ch);
434 }
    
```

### 3.13 サンプルプログラムリスト”phy.c”(11)

```

435 /*"FUNC COMMENT"*****
436 * Outline      : MII 1ビット(1) 出力
437 *-----
438 * Include      :
439 *-----
440 * Declaration  : static void mii_write_1(int ch)
441 *-----
442 * Function     : MII に 1 を出力します。
443 *               : 下記条件を満たすように信号を出力します。
444 *               : ・MDC-High パルス幅 : 160ns(min)
445 *               : ・MDC-Low パルス幅 : 160ns(min)
446 *               : ・MDC-サイクル時間 : 400ns(min)
447 *               : ・MDIO-セットアップ時間 : 10s(min)
448 *               : ・MDIO-ホールド時間 : 10ns(min)
449 *-----
450 * Argument     : int ch: I : PHY-LSI のチャンネル番号(0 または 1)
451 *-----
452 * Return Value: void
453 *-----
454 * Notice       : システムに応じてウェイト時間を変更してください。
455 *"FUNC COMMENT END"*****/
456 static void mii_write_1(int ch)
457 {
458     int j;
459
460     if(ch==0){
461         for(j=MDC_WAIT;j>0;j--){
462             MAC0.PIR.LONG = 0x00000006;
463         }
464         for(j=MDC_WAIT;j>0;j--){
465             MAC0.PIR.LONG = 0x00000007;
466         }
467         for(j=MDC_WAIT;j>0;j--){
468             MAC0.PIR.LONG = 0x00000007;
469         }
470         for(j=MDC_WAIT;j>0;j--){
471             MAC0.PIR.LONG = 0x00000006;
472         }
473     }
474     else if(ch==1){
475         for(j=MDC_WAIT;j>0;j--){
476             MAC1.PIR.LONG = 0x00000006;
477         }
478         for(j=MDC_WAIT;j>0;j--){
479             MAC1.PIR.LONG = 0x00000007;
480         }
481         for(j=MDC_WAIT;j>0;j--){
482             MAC1.PIR.LONG = 0x00000007;
483         }
484         for(j=MDC_WAIT;j>0;j--){
485             MAC1.PIR.LONG = 0x00000006;
486         }
487     }

```



## 3.14 サンプルプログラムリスト"phy.c"(12)

```

472     else{
473         /* DO NOTHING */
474     }
475 }
476
477 /*"FUNC COMMENT"*****
478 * Outline      : MII 1ビット(0) 出力
479 *-----
480 * Include      :
481 *-----
482 * Declaration  : static void mii_write_0(int ch)
483 *-----
484 * Function     : MII に 0 を出力します。
485 *               : 下記条件を満たすように信号を出力します。
486 *               : ・MDC-High パルス幅 : 160ns(min)
487 *               : ・MDC-Low パルス幅 : 160ns(min)
488 *               : ・MDC-サイクル時間 : 400ns(min)
489 *               : ・MDIO-セットアップ時間 : 10s(min)
490 *               : ・MDIO-ホールド時間 : 10ns(min)
491 *-----
492 * Argument     : int ch: I : PHY-LSI のチャンネル番号(0 または 1)
493 *-----
494 * Return Value : void
495 *-----
496 * Notice       : システムに応じてウェイト時間を変更してください。
497 *"FUNC COMMENT END"*****/
498 static void mii_write_0(int ch)
499 {
500     int j;
501
502     if(ch==0){
503         for(j=MDC_WAIT;j>0;j--){
504             MAC0.PIR.LONG = 0x00000002;
505         }
506         for(j=MDC_WAIT;j>0;j--){
507             MAC0.PIR.LONG = 0x00000003;
508         }
509         for(j=MDC_WAIT;j>0;j--){
526             MAC0.PIR.LONG = 0x00000003;
527         }
528         for(j=MDC_WAIT;j>0;j--){
529             MAC0.PIR.LONG = 0x00000002;
530         }
531     }
532     else if(ch==1){
533         /* 省略 */
534     }
535     else{
536         /* DO NOTHING */
537     }
538 }
539
540 /* End of File */

```

### 3.15 サンプルプログラムリスト”phy.h”

```

1  /*"FILE COMMENT"*****
2  *
3  *   System Name   : SH7712 Sample Program
4  *   FILE Name    : phy.h
5  *   Version      : 1.00.00
6  *   Contents     : ヘッダファイル
7  *   Model        : MS7712SE01
8  *   CPU          : SH7712
9  *   Compiler     : SHC9.1.1.0
10 *   OS           : None
11 *
12 *   note         : PHYのヘッダプログラムです。
13 *                <注意事項>
14 *                本サンプルプログラムはすべて参考資料であり
15 *                その動作を保証するものではありません。
16 *                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
17 *                技術参考資料としてご利用ください。
18 *
19 *
20 *   Copyright (C) 2007 Renesas Technology Corp. All Rights Reserved
21 *                AND Renesas Solutions Corp. All Rights Reserved
22 *
23 *   history      : 2007.11.07 ver 1.00.00
24 *"FILE COMMENT END"*****/
25 #ifndef _PHY_H
26 #define _PHY_H
27
28 /* **** プロトタイプ宣言 **** */
29 int phy_autonego(int ch);
30
31 /* Link 結果 */
32 #define NEGO_FAIL          0
33 #define HALF_10M          1
34 #define FULL_10M         2
35 #define HALF_TX           3
36 #define FULL_TX           4
37
38 #endif
39
40 /* End of File */
    
```

#### 4. 参考ドキュメント

- ソフトウェアマニュアル  
SH3、SH3E、SH3-DSP ソフトウェアマニュアル 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)。
- ハードウェアマニュアル  
SH7710 グループハードウェアマニュアル Rev.2.00  
SH7712 ハードウェアマニュアル Rev.1.00  
SH7713 ハードウェアマニュアル Rev.1.00  
(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)。

#### 5. 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/>

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2007.11.07	—	初版発行

### 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すは、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ものです。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. 弊社は製品の品質及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。