SH7216 Group

Configuring the Ethernet PHY-LSI Auto-Negotiation

R01AN0053EJ0200
Rev. 2.00
Sep. 17, 2010

Summary

This application note describes the configuration example of the SH7216 microcomputers (MCUs) to connect with an Ethernet PHY-LSI to auto negotiate with the link partner.

Target Device

SH7216 MCU

Contents

1. Introduction

1.1 Specifications

- This application retrieves the auto-negotiation result (duplex mode) from the Ethernet PHY-LSI connected to the SH7216. Its duplex mode (full-duplex or half-duplex) is set in the Ethernet Controller.
- This application uses the RTL8201 Ethernet PHY-LSI (Realtek Semiconductor Corp.).
- This application uses the auto-negotiation function to link with the Ethernet PHY-LSI.

1.2 Modules Used

- Pin Function Controller (PFC)
- Ethernet Controller (EtherC)

1.3 Applicable Conditions

MCU	SH7216
Operating Frequency	Internal clock: 200 MHz Bus clock: 50 MHz Peripheral clock: 50 MHz AD clock: 50 MHz
Integrated Development Environment	Renesas Electronics High-performance Embedded Workshop Ver.4.07.00
C Compiler	Renesas Electronics SuperH RISC engine Family C/C++ compiler package Ver.9.03 Release 00
Compiler Options	Default setting in the High-performance Embedded Workshop (-cpu=sh2afpu -fpu=single -debug -gbr=auto -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1)

1.4 Related Application Notes

For more information, refer to the following application notes:

- SH7216 Group Example of Initialization
- SH7216 Group Configuration to Receive Ethernet Frames
- SH7216 Group Configuration to Transmit Ethernet Frames

2. Applications

This application uses the Ethernet PHY-LSI to auto negotiate with the other device. The auto-negotiation result (duplex mode) is retrieved via the PHY interface register (PIR) on the EtherC.

2.1 EtherC Overview

As the Ethernet PHY-LSI configures the link with the physical layer, the SH7216 EtherC reads the Ethernet PHY-LSI to retrieve the linking results. This application sets the PHY-LSI pins to enable the auto-negotiation. For more information on the Ethernet PHY-LSI, refer to the datasheet of the Ethernet PHY-LSI.

Interface between the EtherC and the Ethernet PHY-LSI is specified as the MII (Media Independent Interface) by the IEEE 802.3. Figure 1 shows the connection between the SH7216 and the RTL8201CP.

Ethernet PHY-LSI registers store the retrieved duplex mode to read the mode by the serial interface of the MDC and MDIO pins. Use the SH7216 PIR register to read both MDC and MDIO pins. Refer to 2.2 How to Access MII Registers for the procedures to access the PHY-LSI registers.

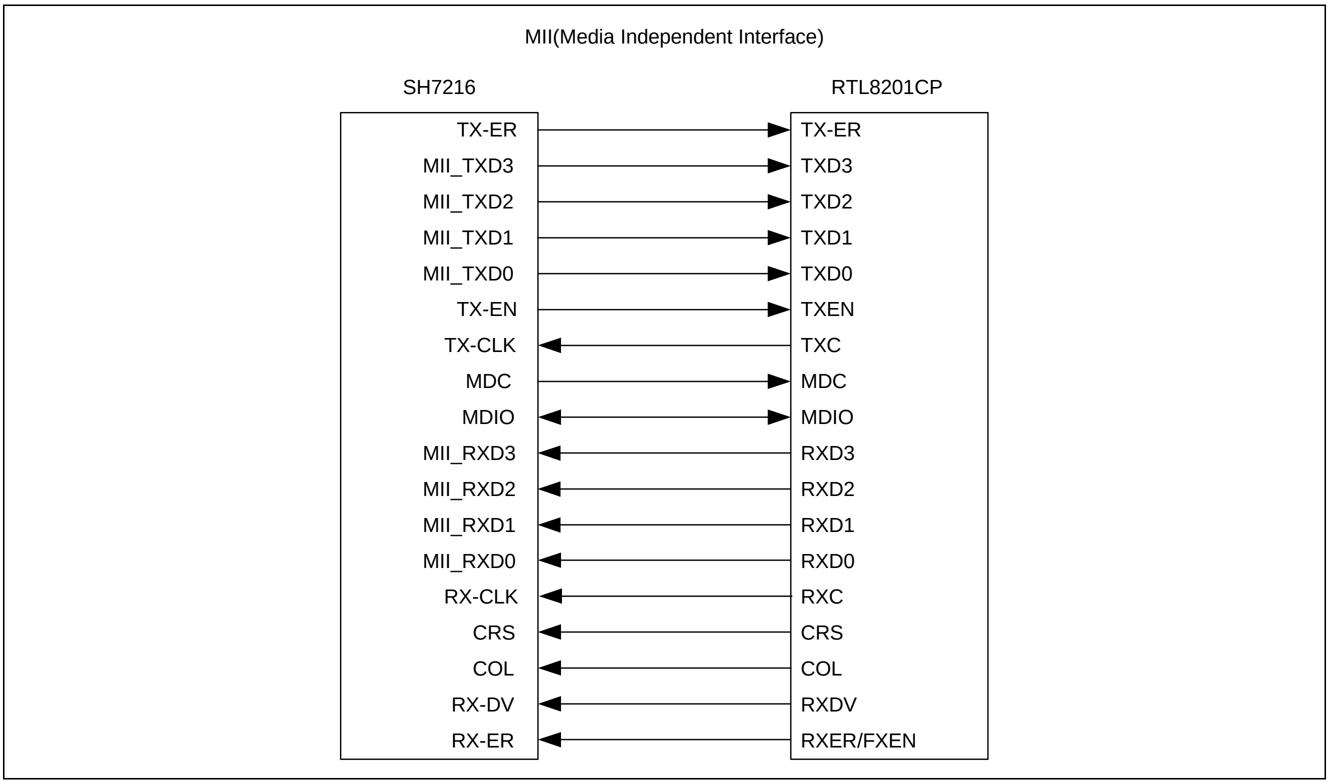


Figure 1 Connection between the SH7216 and RTL8201CP

2.2 How to Access MII Registers

This section describes the procedure for accessing MII registers on the Ethernet PHY-LSI.

Use MDC and MDIO pins (EtherC) to access MII registers. The MDC is the synchronizing clock pin, and the MDIO is the data I/O pin. Use the EtherC PIR register to refer to or change the pin state. As the MII does not include control pins, the MII must output data using the specified format (MII management frame). Figure 2 shows the MII management frame. This sample program outputs high-impedance (phase Z) for one bit period in the idle condition. The IEEE 802.3 does not specify the clock input, however, some PHY-LSIs may not be able to connect correctly. This processing is included in the sample program to make sure the PHY-LSI connect correctly.

Input or output data to/from the MII management frame one bit at a time from the PRE (preamble). Figure 3 to Figure 5 show the I/O flow charts in units of one bit. Pins MDC and MDIO I/O timing must satisfy the IEEE 802.3

IEEE 802.3 I/O timing is described in Table 1 and Figure 6.

Access type	MII Management Frame Fields							
Items	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
Number of bits	32	2	2	5	5	2	16	1
Read	1..1	01	10	AAAAA	RRRRR	Z0	D..D	Z
Write	1..1	01	01	AAAAA	RRRRR	10	D..D	Z

Legend:

PRE (preamble): Outputs a sequence of 32 contiguous one bits to establish synchronization.

ST (start of frame): Outputs 01 to indicate the start of the frame.

OP (operation code): Specifies reading or writing. A 10 pattern indicates reading, and a 01 pattern indicates writing.

PHYAD (PHY address): Identification address when connecting multiples of PHY-LSIs. Outputs in MSB, often specifies the address, depending on the PHY-LSI pin settings.

REGAD (Register address): Outputs in MSB to specify the number of the MII register.

TA (turnaround): Switches the MDIO destination to avoid contention between signals.
(a) Releases bus for one-bit (outputs Z)
As the PHY-LSI outputs 0, it is indicated as Z0.
(b) Outputs 10 when writing

DATA (data): Read or write value in register in 16-bit data. Writes or reads in MSB, sequentially.

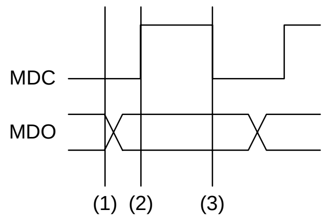
IDLE (IDLE condition): Waits until the next MII management format is input, releases bus (outputs Z).

Figure 2 MII Management Frame Format

(1) Write to the PHY interface register (PIR)
MMD = 1
MDO = write data
MDC = 0

(2) Write to the PHY interface register (PIR)
MMD = 1
MDO = write data
MDC = 1

(3) Write to the PHY interface register (PIR)
MMD = 1
MDO = write data
MDC = 0



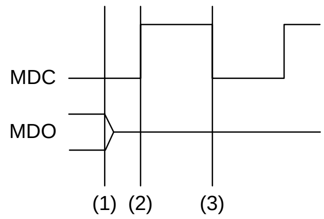
Relationship between 1-bit data writing and the timing

Figure 3 1-bit Data Write Flow Chart

(1) Write to the PHY interface register (PIR)
MMD = 0
MDC = 0

(2) Write to the PHY interface register (PIR)
MMD = 0
MDC = 1

(3) Write to the PHY interface register (PIR)
MMD = 0
MDC = 0



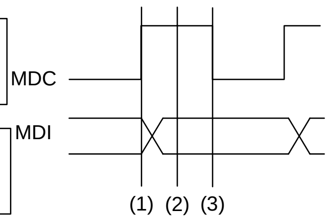
Relationship between the bus release and the timing

Figure 4 Bus Release Flow Chart (phase Z output)

(1) Write to the PHY interface register (PIR)
MMD = 0
MDC = 1

(2) Read from the PHY interface register (PIR)
Specifies the MDI as the read data

(3) Write to the PHY interface register (PIR)
MMD = 0
MDC = 0



Relationship between the 1-bit data reading and the timing

Figure 5 1-bit Data Read Flow Chart

Table 1 MDI/MDIO I/O Timing

Item	Symbol	Min.	Max.	Unit
MDC high level pulse width	t^1	160	–	ns
MDC low level pulse width	t^2	160	–	ns
MDC cycle time	t^3	400	–	ns
MDIO setup time	t^4	10	–	ns
MDIO hold time	t^5	10	–	ns
MDIO output delay time	t^6	0	300	ns

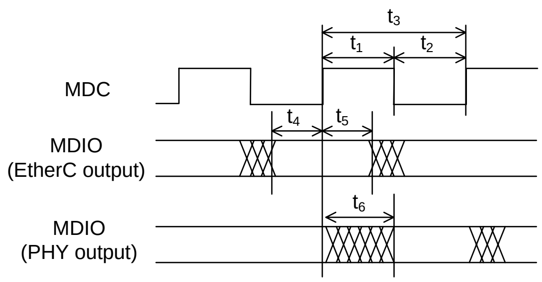


Figure 6 MDC/MDIO I/O Timing

2.3 Sample Program

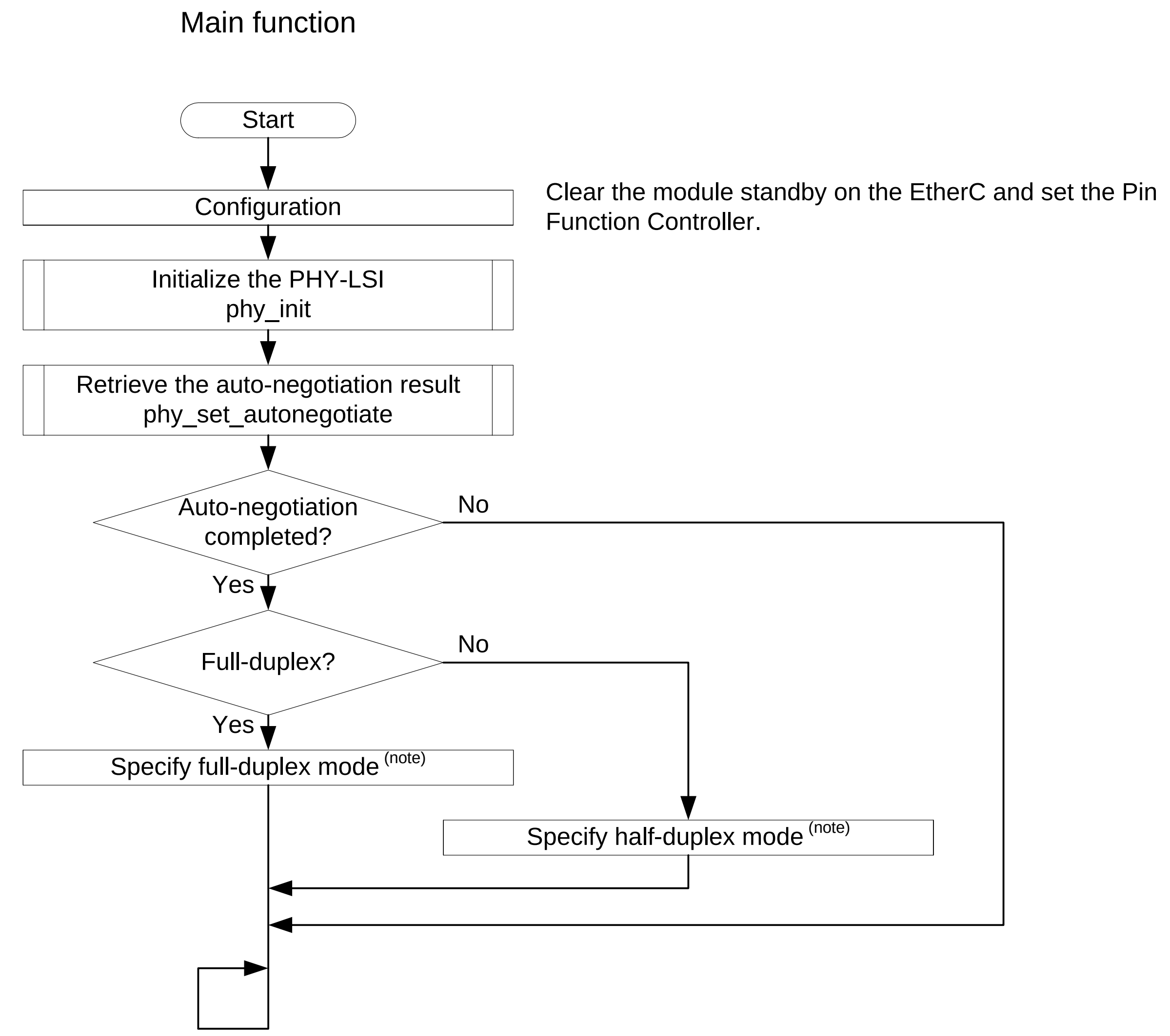
The sample program consists of two source code files (main.c and phy.c), and initialization files created according to the application note "SH7216 Group Example of Initialization".

- main.c

This file describes the main function to reset the PHY-LSI by software, and retrieve the auto-negotiation result. Figure 7 shows the main function flow chart.

- phy.c

This file describes the function to initialize the PHY-LSI (phy_init function) and to retrieve the auto-negotiation result (phy_autonegotiate function). Figure 8 shows the phy_init function flow chart. Figure 9 shows the phy_autonegotiate function flow chart. Figure 10 to Figure 15 show flow charts of the lower functions which are executed within phy_init function and phy_set_autonegotiate function.



Note: Specify the duplex mode (full-duplex or half-duplex) in DM bit in the EtherC mode register (ECMR).

Figure 7 Main Function Flow Chart

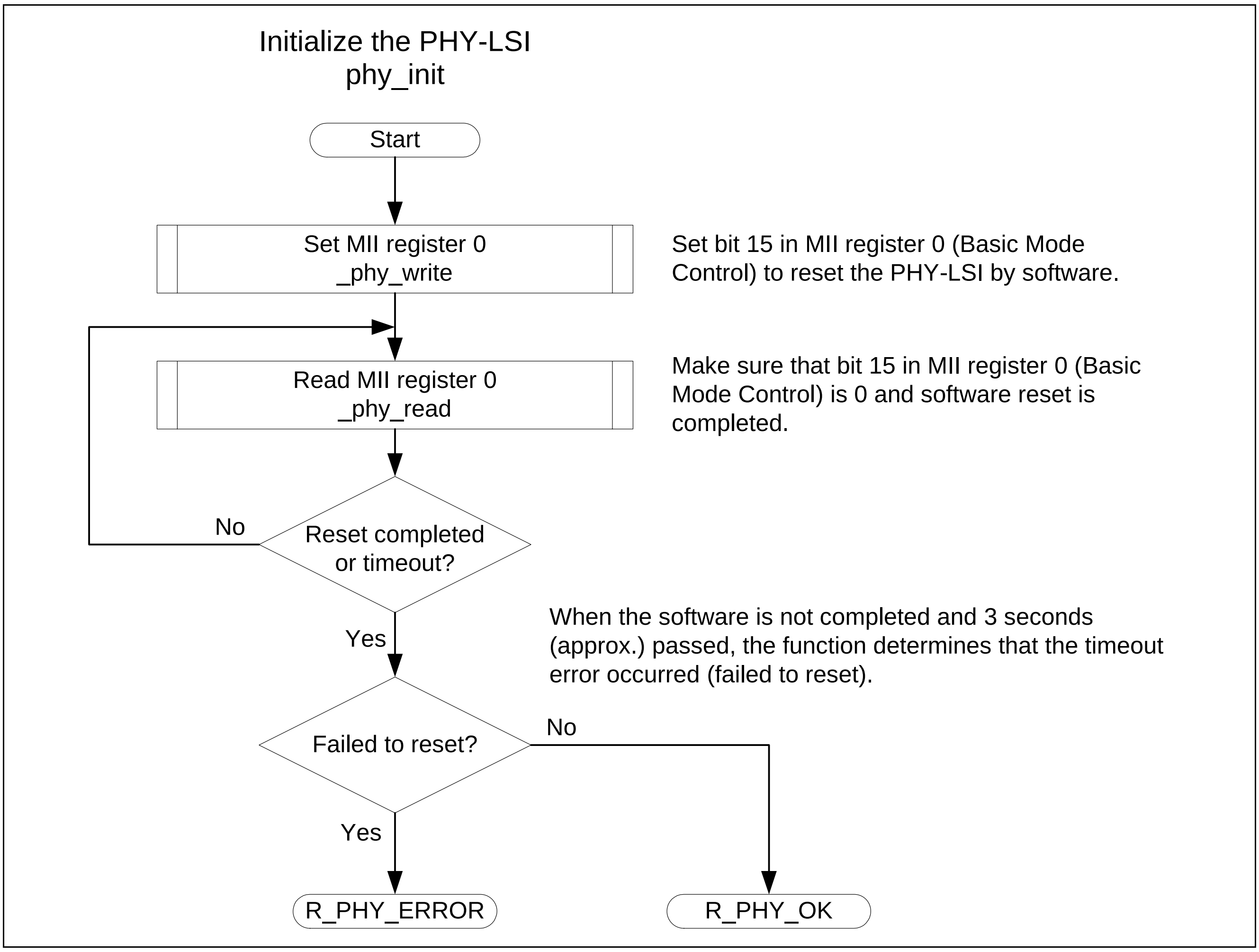


Figure 8 Flow Chart of the phy_init Function

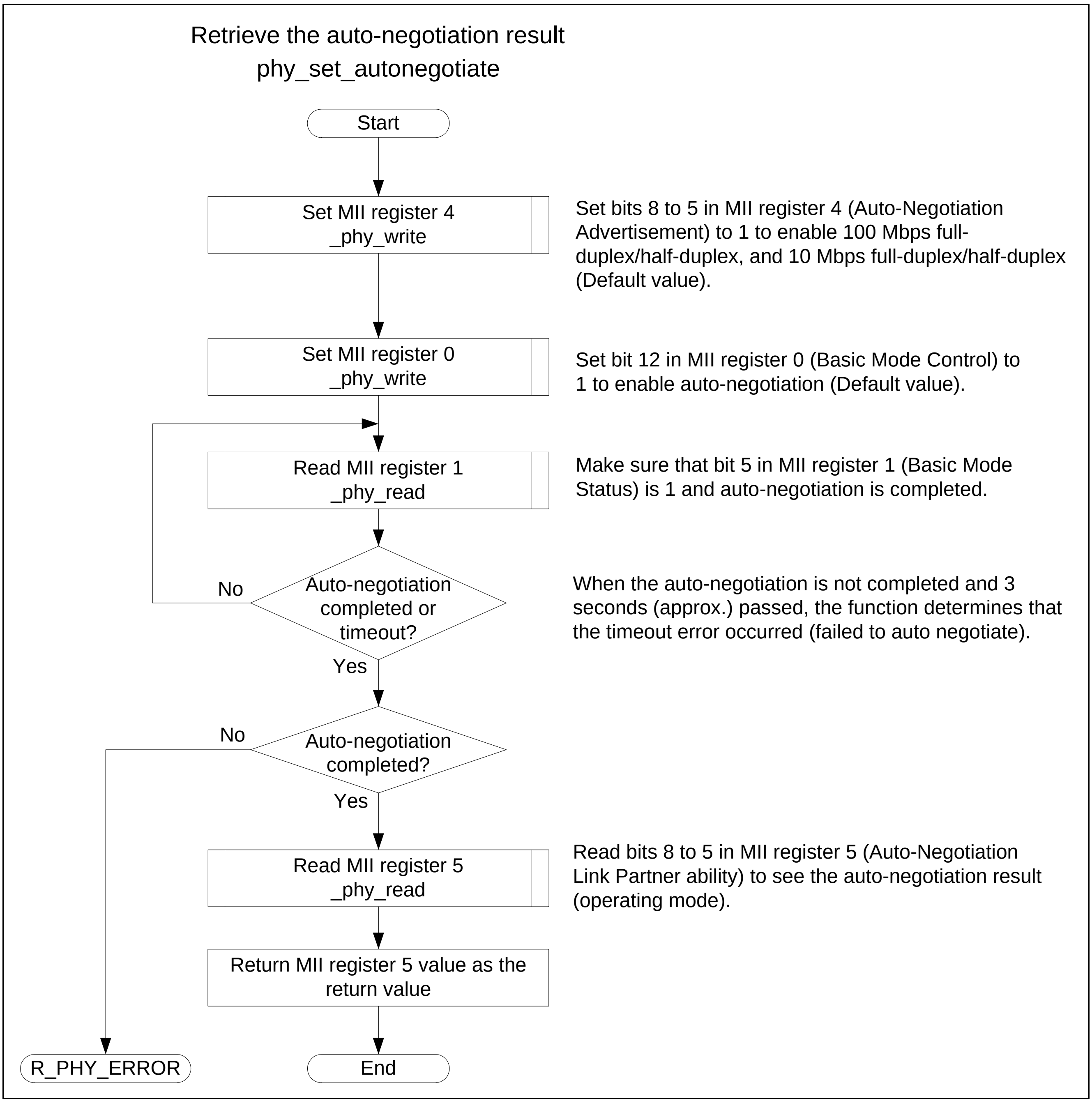
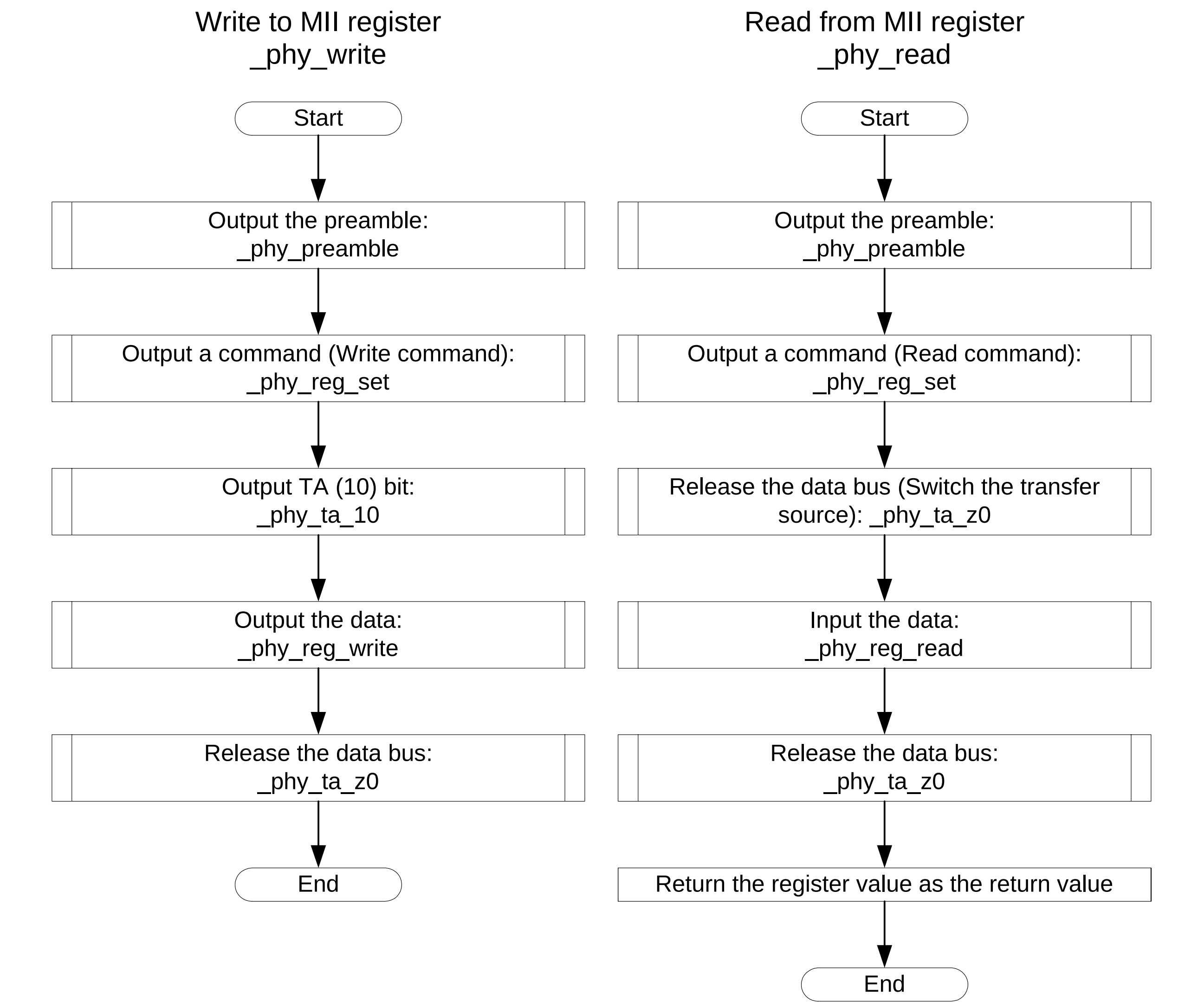


Figure 9 Flow Chart of the phy_set_autonegotiate Function



Note: This sample program defines ST, OP, PHYAD, and REGAD of the MII management frame as commands. For more information about the MII management frame, refer to 2.2 How to Access MII Registers.

Figure 10 Flow Chart for Accessing MII Register (1/6)

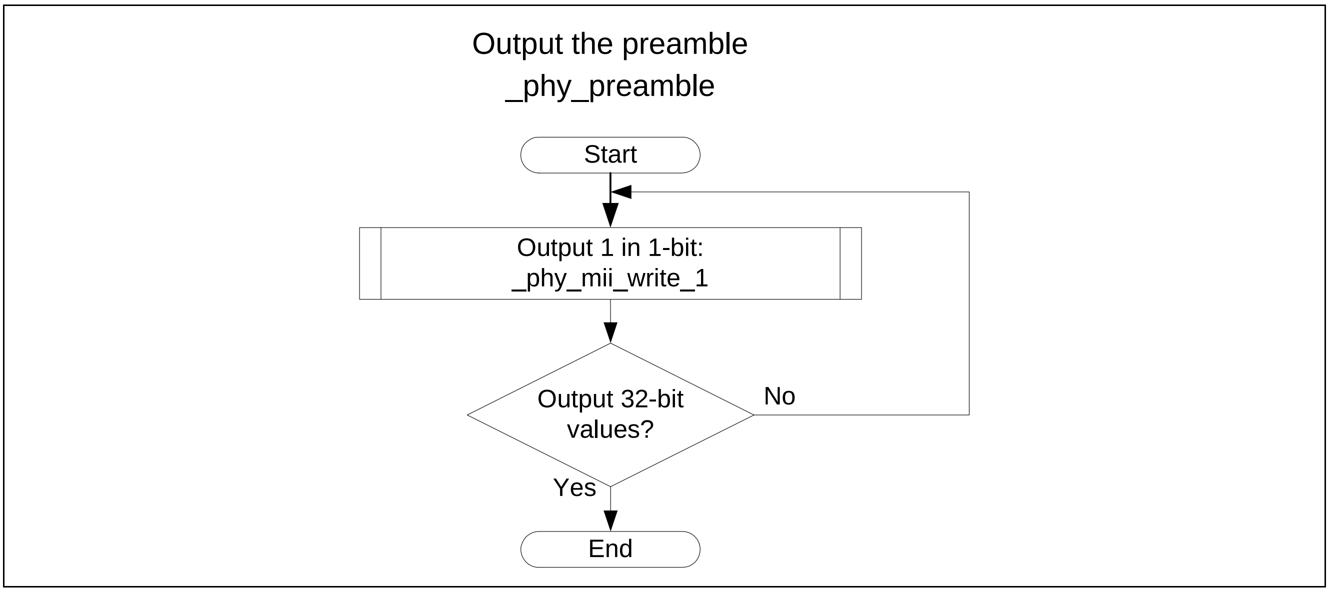
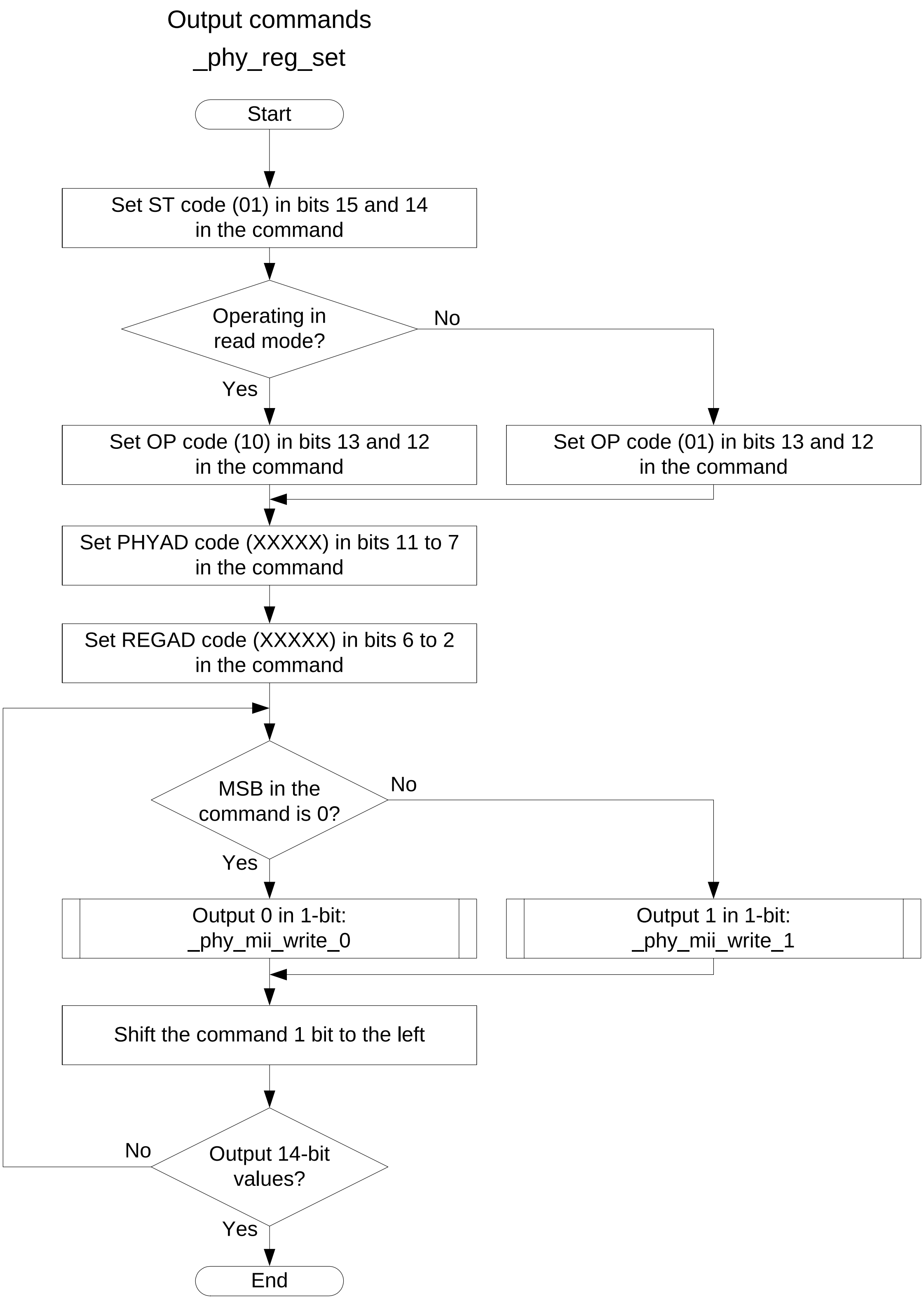


Figure 11 Flow Chart for Accessing the MII Register (2/6)



Note: This sample program defines ST, OP, PHYAD, and REGAD of the MII management frame as commands. For more information about the MII management frame, refer to 2.2 How to Access MII Registers.

Figure 12 Flow Chart for Accessing the MII Register (3/6)

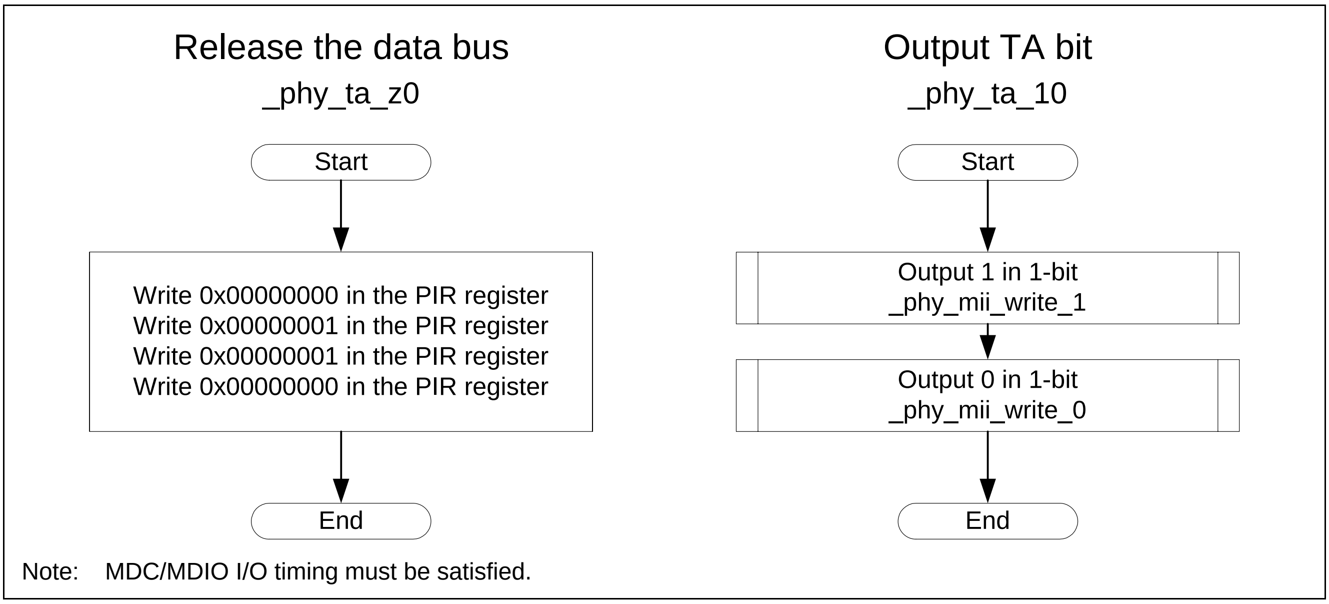


Figure 13 Flow Chart for Accessing the MII Register (4/6)

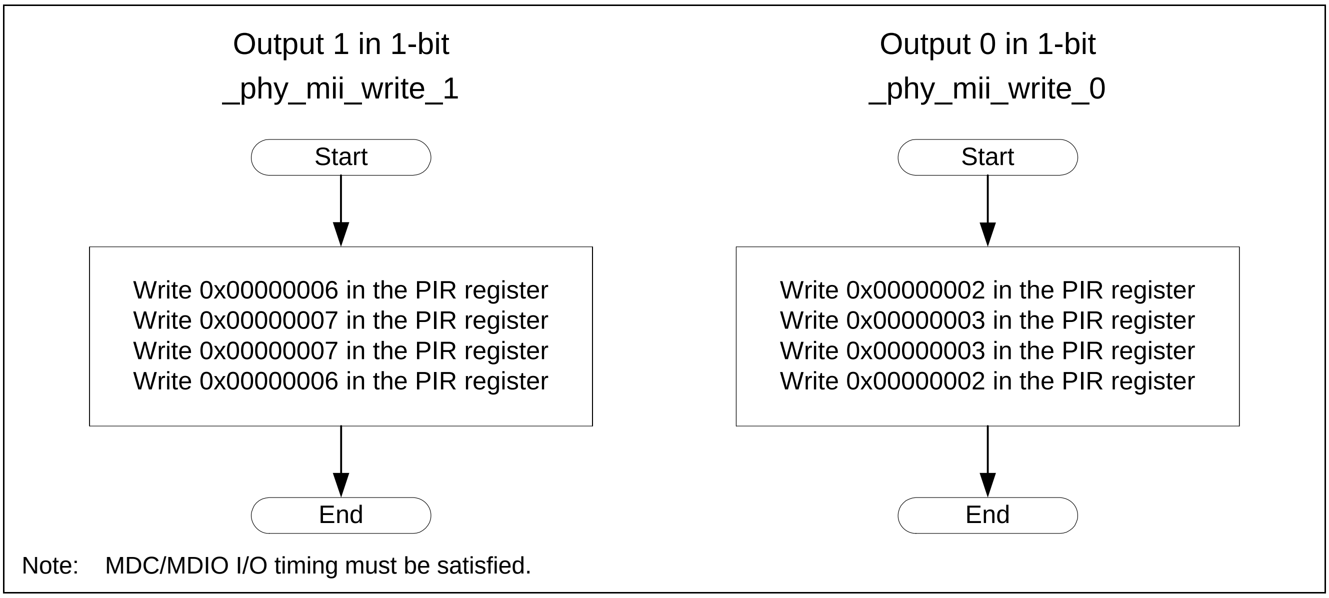
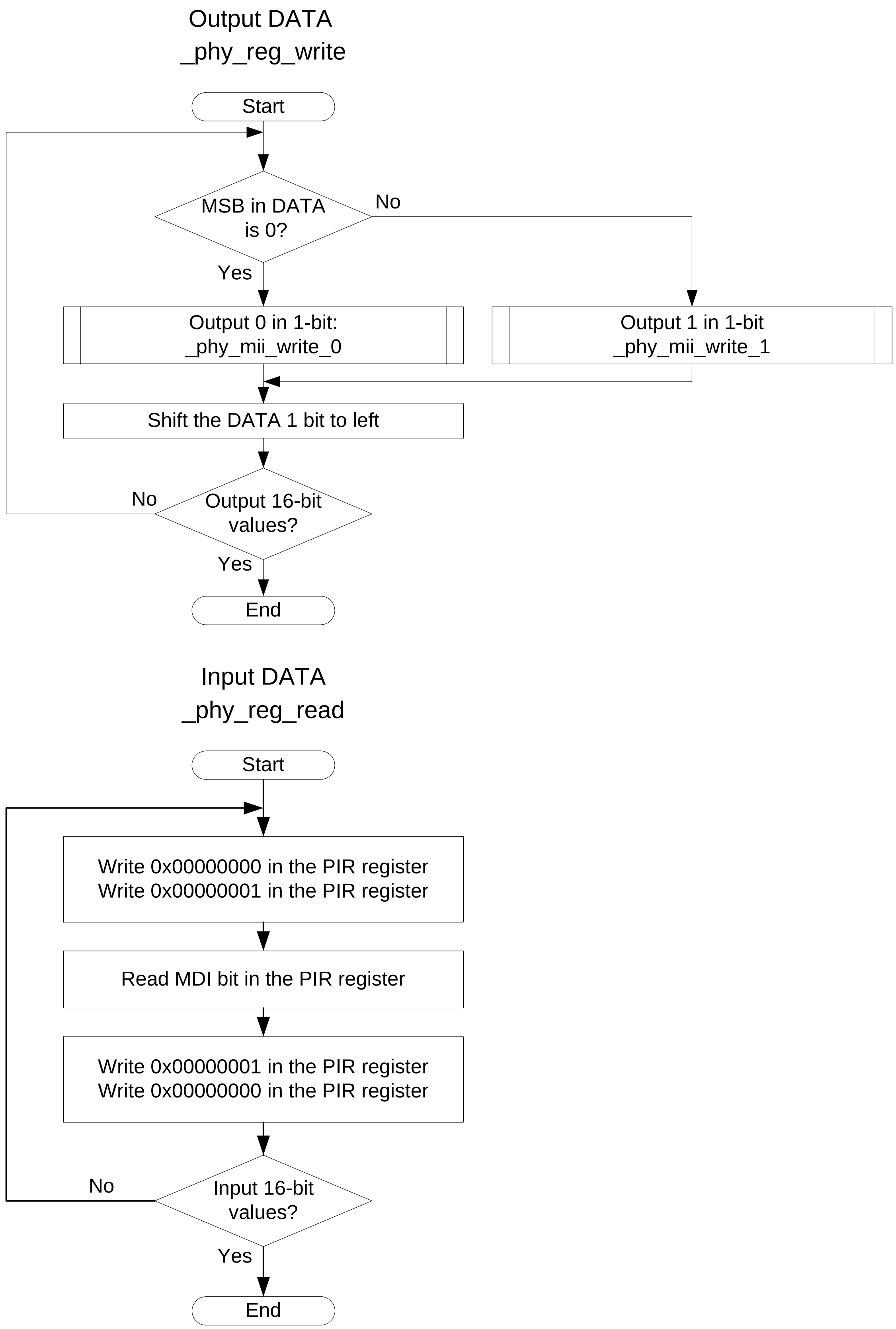


Figure 14 Flow Chart for Accessing the MII Register (5/6)



Note: MDC/MDIO I/O timing must be satisfied.

Figure 15 Flow Chart for Accessing the MII Register (6/6)

2.4 Sample Program Setting

Table 2 lists the setting in the sample program.

Table 2 Sample Program Setting

Item	Description
PHY-LSI to use	RTL8201CP (Manufactured by the Realtek Semiconductor Corp.)
Link mode	100 Mbps (Full-duplex and half-duplex modes) and 10 Mbps (Full-duplex and half-duplex modes)
Link option	Auto-negotiation
MII register to use	• Basic Mode Control (Address: H'00) • Basic Mode Status (Address: H'01) • Auto-Negotiation Advertisement (Address: H'04) • Auto-Negotiation Link Partner Ability (Address: H'05)

2.5 Notes on Using the Sample Program

- The sample program assumes to use the auto-negotiation mode to link with the PHY-LSI.
- When the link partner device is connected in the auto-negotiation mode, the link mode with the SH7216 is decided according to the priority listed in Table 3.
- When waiting until the reset is completed in the phy_init function, and the auto-negotiation is completed in the phy_set_autonegotiate function, these functions do not return while they are referring to register values for about 3 seconds (when the internal clock is at 200 MHz).

Table 3 Link Mode Priorities

Priority		Link Mode
High	1	100 Mbps, full-duplex
	2	100 Mbps, half-duplex
	3	10 Mbps, full-duplex
Low	4	10 Mbps, half-duplex

3. Sample Program Listing

3.1 Sample Program List "main.c" (1/3)

```
/*******************************************************************************
*   DISCLAIMER
*
*   This software is supplied by Renesas Electronics Corp. and is only
*   intended for use with Renesas products.  No other uses are authorized.
*
*   This software is owned by Renesas Electronics Corp. and is protected under
*   all applicable laws, including copyright laws.
*
*   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
*   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
*   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
*   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
*   DISCLAIMED.
*
*   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
*   ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
*   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
*   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
*   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
*
*   Renesas reserves the right, without notice, to make changes to this
*   software and to discontinue the availability of this software.
*   By using this software, you agree to the additional terms and
*   conditions found by accessing the following link:
*   http://www.renesas.com/disclaimer
********************************************************************************
*   Copyright (C) 2010. Renesas Electronics Corporation. All Rights Reserved.
*""FILE COMMENT""*********** Technical reference data *************************
*   System Name : SH7216 Sample Program
*   File Name   : main.c
*   Abstract    : Retrieve the auto-negotiation result by the Ethernet PHY-LSI
*   Version     : 2.00.00
*   Device      : SH7216
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
*               : C/C++ compiler package for the SuperH RISC engine family
*               :                             (Ver.9.03 Release00).
*   OS          : None
*   H/W Platform: R0K572167 (CPU board)
*   Description : Configures the MCU for the Ethernet transmission and transmits
*               : Ethernet frames.
********************************************************************************
*   History     : Nov.18,2009 Ver.1.00.00
*               : Jul.23,2010 Ver.2.00.00 Comply with the Renesas API
*""FILE COMMENT END""**********************************************************/
#include "iodefine.h"
#include "stdint.h"
#include "phy.h"
```

3.2 Sample Program List "main.c" (2/3)

```

/* ==== Prototype Declaration ==== */
void main(void);

/* ==== Variable Declaration ==== */

/*""FUNC COMMENT""*************************************************************
 * ID            :
 * Outline       : Sample program main
 *-----------------------------------------------------------------------------
 * Include       : "iodefine.h", "stdint.h", and "phy.h"
 *-----------------------------------------------------------------------------
 * Declaration   : void main(void);
 *-----------------------------------------------------------------------------
 * Description   : Uses the internal Ethernet Controller (EtherC) to read the PHY
 *               : link mode. The PHY link mode is decided by the PHY-LSI
 *               : auto-negotiation in phy_init and phy_set_autonegotiate in the
 *               : R_Ether_Open function.
 *               : Sets the duplex mode (full-/half-duplex) in the EtherC.
 *               : Uses the RTL8201CP (Realtek. Corp) as the PHY module.
 *-----------------------------------------------------------------------------
 * Argument      : void
 *-----------------------------------------------------------------------------
 * Return Value  : void
 *-----------------------------------------------------------------------------
 * Note          : None
 *""FUNC COMMENT END""**********************************************************/
void main(void)
{
  uint32_t ch = 0;
  uint16_t phydata;

  /* ==== Clears the EtherC/E-DMAC module standby ==== */
  STB.CR4.BIT._ETHER = 0;
  /* ==== Sets the PFC (For the EtherC) ==== */
  PFC.PACRL4.BIT.PA12MD = 7; /* TX_CLK (input) */
  PFC.PACRL3.WORD = 0x7777;  /* TX_EN,MII_TXD0,MII_TXD1,MII_TXD2 (output) */
  PFC.PACRL2.BIT.PA7MD = 7;  /* MII_TXD3 (output) */
  PFC.PACRL2.BIT.PA6MD = 7;  /* TX_ER (output) */
  PFC.PDCRH4.WORD = 0x7777;  /* RX_DV,RX_ER,MII_RXD3,MII_RXD2 (input) */
  PFC.PDCRH3.WORD = 0x7777;  /* MII_RXD1,MII_RXD0,RX_CLK,CRS (input) */
  PFC.PDCRH2.WORD = 0x7777;  /* COL (input),WOL,EXOUT,MDC (input) */
  PFC.PDCRH1.BIT.PD19MD = 7; /* LINKSTA (input) */
  PFC.PDCRH1.BIT.PD18MD = 7; /* MDIO (input/output) */
```

3.3 Sample Program List "main.c" (3/3)

```

  /* ==== Initializes the PHY-LSI ==== */
  phy_init();

  /* ==== Retrieves the PHY-LSI auto-negotiation result ==== */
  phydata = phy_set_autonegotiate();

  /* ---- Determines whether to auto-negotiate or not ---- */
  if(phydata == R_PHY_ERROR){
    /* Failed to auto-negotiate */
  }
  /* ---- Detects the performance of the link partner ---- */
  else if(phydata & 0x0100){          /* Detects PHY-LSI register 0 */
                                  /* bit8 : DuplexMode : 1 ---- */
                                  /* Supports full-duplex mode */

    EtherC.ECMR.BIT.DM = 1;           /* Full-duplex communication */
  }
  else{
    EtherC.ECMR.BIT.DM = 0;           /* Half-duplex communication */
  }

  while(1){
    /* sleep */
  }
}

/* End of file */
```

3.4 Sample Program List "phy.c" (1/13)

```
/*******************************************************************************
*   DISCLAIMER
*
*   This software is supplied by Renesas Electronics Corp. and is only
*   intended for use with Renesas products.  No other uses are authorized.
*
*   This software is owned by Renesas Electronics Corp. and is protected under
*   all applicable laws, including copyright laws.
*
*   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
*   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
*   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
*   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
*   DISCLAIMED.
*
*   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
*   ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
*   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
*   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
*   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
*
*   Renesas reserves the right, without notice, to make changes to this
*   software and to discontinue the availability of this software.
*   By using this software, you agree to the additional terms and
*   conditions found by accessing the following link:
*   http://www.renesas.com/disclaimer
********************************************************************************
*   Copyright (C) 2009(2010). Renesas Electronics Corporation. All Rights Reserved.
*""FILE COMMENT""*********** Technical reference data *************************
*   System Name : SH7216 Sample Program
*   File Name   : phy.c
*   Version     : 2.00.00
*   Device      : SH7216
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
*               : C/C++ compiler package for the SuperH RISC engine family
*               :                             (Ver.9.03 Release00).
*   OS          : None
*   H/W Platform: R0K572167 (CPU board)
*   Description : Ethernet PHY device driver
******************************************************************************
*   History     : Jun.10.2009 Ver.1.00.00
*               : Jul.23,2010 Ver.2.00.00 Comply with the Renesas API
*""FILE COMMENT END""**********************************************************/
#include "iodefine.h"
#include "stdint.h"
#include "phy.h"

```

3.5 Sample Program List "phy.c" (2/13)

```
/* ==== Prototype declaration ==== */
uint16_t _phy_read( uint16_t reg_addr );
void _phy_write( uint16_t reg_addr, uint16_t data );
void _phy_preamble( void );
void _phy_reg_set( uint16_t reg_addr, int32_t option );
void _phy_reg_read( uint16_t *data );
void _phy_reg_write( uint16_t data );
void _phy_ta_z0( void );
void _phy_ta_10( void );
void _phy_mii_write_1( void );
void _phy_mii_write_0( void );

/*""FUNC COMMENT""***************************************************************
 * ID           :
 * Outline      : Reset the PHY module by software
 *-------------------------------------------------------------------------------
 * Include      :
 *-------------------------------------------------------------------------------
 * Declaration  : int16_t phy_init( void );
 *-------------------------------------------------------------------------------
 * Description  : Initializes the Ethernet PHY-LSI.
 *-------------------------------------------------------------------------------
 * Argument     : void
 *-------------------------------------------------------------------------------
 * Return Value : R_PHY_OK;    Succeeded to initialize the PHY-LSI
 *              : R_PHY_ERROR; Failed to initialize the PHY-LSI
 *-------------------------------------------------------------------------------
 * Note         : None
 *""FUNC COMMENT END""**********************************************************/
int16_t phy_init( void )
{
  uint16_t reg;
  uint16_t count;

  /* ==== Initializes the PHY-LSI ==== */
  _phy_write(BASIC_MODE_CONTROL_REG, 0x8000);

  count = 0;

  do{
    reg = _phy_read(BASIC_MODE_CONTROL_REG);
    count++;
  }while((reg & 0x8000) && (count < 0x4000));

  if(count >= 0x4000){
    return R_PHY_ERROR;
  }
  else{
    return R_PHY_OK;
  }
}

```

3.6 Sample Program List "phy.c" (3/13)

```
/*""FUNC COMMENT""**************************************************************
 * ID           :
 * Outline      : Set the PHY-LSI in 100 Mbps, full-duplex mode
 *------------------------------------------------------------------------------
 * Include      :
 *------------------------------------------------------------------------------
 * Declaration  : void phy_set_100full( void );
 *------------------------------------------------------------------------------
 * Description  : Sets the PHY-LSI in 100 Mbps, full-duplex mode.
 *------------------------------------------------------------------------------
 * Argument     : void
 *------------------------------------------------------------------------------
 * Return Value : void
 *------------------------------------------------------------------------------
 * Note         : None
 *""FUNC COMMENT END""**********************************************************/
void phy_set_100full( void )
{
  _phy_write(BASIC_MODE_CONTROL_REG, 0x2100);
}

/*""FUNC COMMENT""**************************************************************
 * ID           :
 * Outline      : Set the PHY-LSI in 10 Mbps, half-duplex mode
 *------------------------------------------------------------------------------
 * Include      :
 *------------------------------------------------------------------------------
 * Declaration  : void phy_set_10half( void );
 *------------------------------------------------------------------------------
 * Description  : Sets the PHY-LSI in 10 Mbps, half-duplex mode.
 *------------------------------------------------------------------------------
 * Argument     : void
 *------------------------------------------------------------------------------
 * Return Value : void
 *------------------------------------------------------------------------------
 * Note         : None
 *""FUNC COMMENT END""**********************************************************/
void phy_set_10half( void )
{
  _phy_write(BASIC_MODE_CONTROL_REG, 0x0000);
}

```

3.7 Sample Program List "phy.c" (4/13)

```
/*""FUNC COMMENT""*************************************************************
 * ID             :
 * Outline        : Detecte the PHY link auto-negotiation result
 *-----------------------------------------------------------------------------
 * Include        :
 *-----------------------------------------------------------------------------
 * Declaration    : int16_t phy_set_autonegotiate( void );
 *-----------------------------------------------------------------------------
 * Description    : Retrieves the auto-negotiation result by the MII management
 *                : interface and returns it as the return value.
 *                : The PHY-LSI used in this sample program supports up to 100 Mbps,
 *                : full-duplex mode. When the partner device supports the
 *                : auto-negotiation, the PHY-LSI is connected with the device in
 *                : its fastest mode. When the device does not support the
 *                : auto-negotiation, this sample program detects the actual link
 *                : speed in half-duplex mode by the parallel detection.
 *-----------------------------------------------------------------------------
 * Argument       : void
 *-----------------------------------------------------------------------------
 * Return Value   : bit 8   - Full duplex 100 Mbps
 *                : bit 7   - Half duplex 100 Mbps
 *                : bit 6   - Full duplex 10 Mbps
 *                : bit 5   - Half duplex 10 Mbps
 *                : bit 4-0 - Always set to 00001 (IEEE 802.3)
 *                : -1      - error
 *-----------------------------------------------------------------------------
 * Note           : None
 *""FUNC COMMENT END""**********************************************************/
int16_t phy_set_autonegotiate( void )
{
  uint16_t reg;
  uint16_t count;

  _phy_write(AN_ADVERTISEMENT_REG, 0x01E1);
  _phy_write(BASIC_MODE_CONTROL_REG, 0x1000);

  count = 0;

  do{
    reg = _phy_read(BASIC_MODE_STATUS_REG);
    count++;
  }while(!(reg & 0x0020) && count < 0x4000);

  if(count >= 0x4000){
    return R_PHY_ERROR;
  }
  else{
    return ((int16_t)_phy_read(AN_LINK_PARTNER_ABILITY_REG));
  }
}

```

3.8 Sample Program List "phy.c" (5/13)

```
/*""FUNC COMMENT""*****************************************************************
 * ID            :
 * Outline       : Read the PHY module register
 *------------------------------------------------------------------------------
 * Include       :
 *------------------------------------------------------------------------------
 * Declaration   : uint16_t _phy_read( uint16_t reg_addr );
 *------------------------------------------------------------------------------
 * Description   : Retrieves the PHY module register value.
 *------------------------------------------------------------------------------
 * Argument      : uint16_t reg_addr ; I : PHY register address to retrieve
 *------------------------------------------------------------------------------
 * Return Value : Retrieved register value
 *------------------------------------------------------------------------------
 * Note          : None
 *""FUNC COMMENT END""**********************************************************/
uint16_t _phy_read( uint16_t reg_addr )
{
  uint16_t data;

  _phy_preamble();
  _phy_reg_set( reg_addr, PHY_READ );
  _phy_ta_z0();
  _phy_reg_read( &data );
  _phy_ta_z0();

  return ( data );
}

/*""FUNC COMMENT""*****************************************************************
 * ID            :
 * Outline       : Write the PHY module register
 *------------------------------------------------------------------------------
 * Include       :
 *------------------------------------------------------------------------------
 * Declaration   : void _phy_write( uint16_t reg_addr, uint16_t data );
 *------------------------------------------------------------------------------
 * Description   : Sets the value in the PHY module register.
 *------------------------------------------------------------------------------
 * Argument      : unsigned short reg_addr ; I : PHY register address to write
 *               : unsigned short data     ; I : Value to set in the PHY register
 *------------------------------------------------------------------------------
 * Return Value : void
 *------------------------------------------------------------------------------
 * Note          : None
 *""FUNC COMMENT END""**********************************************************/
void _phy_write( uint16_t reg_addr, uint16_t data )
{
  _phy_preamble();
  _phy_reg_set( reg_addr, PHY_WRITE );
  _phy_ta_10();
  _phy_reg_write( data );
  _phy_ta_z0();
}
```

3.9 Sample Program List "phy.c" (6/13)

```

/*""FUNC COMMENT""**************************************************************
 * ID           :
 * Outline      : Prepare to access the PHY module register
 *-----------------------------------------------------------------------------
 * Include      :
 *-----------------------------------------------------------------------------
 * Declaration  : void  _phy_preamble( void );
 *-----------------------------------------------------------------------------
 * Description  : Outputs 1 to the MII management interface to prepare to access
 *              : the PHY module register.
 *-----------------------------------------------------------------------------
 * Argument     : void
 *-----------------------------------------------------------------------------
 * Return Value : void
 *-----------------------------------------------------------------------------
 * Note         : None
 *""FUNC COMMENT END""**********************************************************/
void _phy_preamble( void )
{
  int16_t i;

  i = 32;
  while( i > 0 ){
    /* Outputs 1 to the MII (Media Independent Interface) block */
    _phy_mii_write_1();
    i--;
  }
}

```

3.10 Sample Program List "phy.c" (7/13)

```
/*""FUNC COMMENT""*****************************************************************
 * ID             :
 * Outline        : Set the PHY module register mode
 *----------------------------------------------------------------------------
 * Include        :
 *----------------------------------------------------------------------------
 * Declaration  : void _phy_reg_set( uint16_t reg_addr, int32_t option );
 *----------------------------------------------------------------------------
 * Description  : Sets the PHY module register R/W mode.
 *----------------------------------------------------------------------------
 * Argument      : unsigned short reg_addr ; I : PHY register address
 *               : int option              ; I : Specify the R/W mode
 *----------------------------------------------------------------------------
 * Return Value : void
 *----------------------------------------------------------------------------
 * Note          : None
 *""FUNC COMMENT END""**********************************************************/
void _phy_reg_set( uint16_t reg_addr, int32_t option )
{
  int32_t i;
  uint16_t data;

  data = 0;
  data = (PHY_ST << 14);          /* ST code     */

  if( option == PHY_READ ){
    data |= (PHY_READ << 12);  /* OP code(RD) */
  }
  else{
    data |= (PHY_WRITE << 12); /* OP code(WT) */
  }

  data |= (PHY_ADDR << 7);     /* PHY Address */
  data |= (reg_addr << 2);     /* Reg Address */

  i = 14;
  while( i > 0 ){
    if( (data & 0x8000) == 0 ){
        _phy_mii_write_0();
    }
    else{
        _phy_mii_write_1();
    }
    data <<= 1;
    i--;
  }
}
```

3.11 Sample Program List "phy.c" (8/13)

```
/*""FUNC COMMENT""*************************************************************
 * ID            :
 * Outline       : Retrieve the PHY module register value
 *-----------------------------------------------------------------------------
 * Include       :
 *-----------------------------------------------------------------------------
 * Declaration   : void  _phy_reg_read( uint16_t *data );
 *-----------------------------------------------------------------------------
 * Description   : Retrieves the PHY module register value one bit at a time.
 *               : Inputs or outputs signals to satisfy the following conditions;
 *               :  - MDC High level pulse width: 160 ns (min.)
 *               :  - MDC Low level pulse width: 160 ns (min.)
 *               :  - MDC cycle time: 400 ns (min.)
 *               :  - MDIO output delay time (from PHY): 300 ns (max.)
 *-----------------------------------------------------------------------------
 * Argument      : uint16_t *data ; O : Address storing the retrieved value
 *-----------------------------------------------------------------------------
 * Return Value : void
 *-----------------------------------------------------------------------------
 * Note          : Alter the wait time according to the system.
 *""FUNC COMMENT END""*********************************************************/
void  _phy_reg_read( uint16_t *data )
{
  int32_t  i,j;
  uint16_t reg_data;

  reg_data = 0;
  i = 16;
  /* ---- Reads data one bit at a time ---- */
  while( i > 0 ){
    for(j = MDC_WAIT; j > 0; j--){
        EtherC.PIR.LONG = 0x00000000;
    }
    for(j = MDC_WAIT; j > 0; j--){
        EtherC.PIR.LONG = 0x00000001;
    }

    reg_data <<= 1;
    reg_data |= (uint16_t)((EtherC.PIR.LONG & 0x00000008) >> 3); /* MDI read*/

    for(j = MDC_WAIT; j > 0; j--){
        EtherC.PIR.LONG = 0x00000001;
    }
    for(j = MDC_WAIT; j > 0; j--){
        EtherC.PIR.LONG = 0x00000000;
    }
    i--;
  }
  *data = reg_data;
}
```

3.12 Sample Program List "phy.c" (9/13)

```
/*""FUNC COMMENT""*************************************************************
 * ID            :
 * Outline       : Set the PHY module register value
 *-----------------------------------------------------------------------------
 * Include       :
 *-----------------------------------------------------------------------------
 * Declaration  : void _phy_reg_write( uint16_t data );
 *-----------------------------------------------------------------------------
 * Description  : Sets the PHY module register value one bit at a time.
 *-----------------------------------------------------------------------------
 * Argument     : uint16_t data ; I : Value to set in the register
 *-----------------------------------------------------------------------------
 * Return Value : void
 *-----------------------------------------------------------------------------
 * Note         : None
 *""FUNC COMMENT END""*********************************************************/
void _phy_reg_write( uint16_t data )
{
  int32_t  i;

  i = 16;
  while( i > 0 ){
    if( (data & 0x8000) == 0 ){
    _phy_mii_write_0();
    }
    else{
        _phy_mii_write_1();
    }
    i--;
    data <<= 1;
  }
}

```

3.13 Sample Program List "phy.c" (10/13)

```
/*""FUNC COMMENT""*****************************************************************
 * ID             :
 * Outline        : Access the PHY module register and release the bus
 *------------------------------------------------------------------------------
 * Include        :
 *------------------------------------------------------------------------------
 * Declaration    : void _phy_ta_z0( void );
 *------------------------------------------------------------------------------
 * Description    : Sets the access to the PHY module as reading te data.
 *                : Outputs signals to satisfy the following conditions;
 *                :  - MDC High level pulse width: 160 ns (min.)
 *                :  - MDC Low level pulse width: 160 ns (min.)
 *                :  - MDC cycle time: 400 ns (min.)
 *                :  - MDIO setup time: 10 ns (min.)
 *                :  - MDIO hold time: 10 ns (min.)
 *------------------------------------------------------------------------------
 * Argument       : void
 *------------------------------------------------------------------------------
 * Return Value : void
 *------------------------------------------------------------------------------
 * Note           : Alte the wait time according to the system.
 *""FUNC COMMENT END""************************************************************/
void _phy_ta_z0( void )
{
 int32_t j;

 for(j = MDC_WAIT; j > 0; j--){
   EtherC.PIR.LONG = 0x00000000;
 }
 for(j = MDC_WAIT; j > 0; j--){
   EtherC.PIR.LONG = 0x00000001;
 }
 for(j = MDC_WAIT; j > 0; j--){
   EtherC.PIR.LONG = 0x00000001;
 }
 for(j = MDC_WAIT; j > 0; j--){
   EtherC.PIR.LONG = 0x00000000;
 }
}
```

3.14 Sample Program List "phy.c" (11/13)

```
/*""FUNC COMMENT""*************************************************************
 * ID           :
 * Outline      : Access the PHY module register, output the TA (10) bit
 *-----------------------------------------------------------------------------
 * Include      :
 *-----------------------------------------------------------------------------
 * Declaration  : void _phy_ta_10(void);
 *-----------------------------------------------------------------------------
 * Description  : Outputs 1 and 0 to the PHY module MII management interface.
 *-----------------------------------------------------------------------------
 * Argument     : void
 *-----------------------------------------------------------------------------
 * Return Value : void
 *-----------------------------------------------------------------------------
 * Note         : None
 *""FUNC COMMENT END""*********************************************************/
void _phy_ta_10(void)
{
  _phy_mii_write_1();
  _phy_mii_write_0();
}

```

3.15 Sample Program List "phy.c" (12/13)

```
/*""FUNC COMMENT""***************************************************************
 * ID            :
 * Outline       : Access the PHY module register, output 1 in 1-bit
 *-------------------------------------------------------------------------------
 * Include       :
 *-------------------------------------------------------------------------------
 * Declaration   : void _phy_mii_write_1( void );
 *-------------------------------------------------------------------------------
 * Description   : Outputs 1 to the PHY module MII management interface.
 *               : Outputs signals to satisfy the following conditions;
 *               :  - MDC High level pulse width: 160 ns (min.)
 *               :  - MDC Low level width: 160 ns (min.)
 *               :  - MDC cycle time: 400 ns (min.)
 *               :  - MDIO setup time: 10 ns (min.)
 *               :  - MDIO hold time: 10 ns (min.)
 *-------------------------------------------------------------------------------
 * Argument      : void
 *-------------------------------------------------------------------------------
 * Return Value : void
 *-------------------------------------------------------------------------------
 * Note          : Alter the wait time according to the system.
 *""FUNC COMMENT END""**********************************************************/
void _phy_mii_write_1( void )
{
  int32_t j;

  for(j = MDC_WAIT; j > 0; j--){
    EtherC.PIR.LONG = 0x00000006;
  }
  for(j = MDC_WAIT; j > 0; j--){
    EtherC.PIR.LONG = 0x00000007;
  }
  for(j = MDC_WAIT; j > 0; j--){
    EtherC.PIR.LONG = 0x00000007;
  }
  for(j = MDC_WAIT; j > 0; j--){
    EtherC.PIR.LONG = 0x00000006;
  }
}
```

3.16 Sample Program List "phy.c" (13/13)

```
/*""FUNC COMMENT""*************************************************************
 * ID             :
 * Outline        : Access the PHY module register, output 0 in 1-bit
 *------------------------------------------------------------------------------
 * Include        :
 *------------------------------------------------------------------------------
 * Declaration    : void _phy_mii_write_0( void );
 *------------------------------------------------------------------------------
 * Description    : Outputs 0 to the PHY module MII management interface.
 *                : Outputs signals to satisfy the following conditions;
 *                :  - MDC High level pulse width: 160 ns (min.)
 *                :  - MDC Low level pulse width: 160 ns (min.)
 *                :  - MDC cycle time: 400 ns (min.)
 *                :  - MDIO setup time: 10 ns (min.)
 *                :  - MDIO hold time: 10 ns (min.)
 *------------------------------------------------------------------------------
 * Argument       : void
 *------------------------------------------------------------------------------
 * Return Value : void
 *------------------------------------------------------------------------------
 * Note           : Alter the wait time according to the system.
 *""FUNC COMMENT END""**********************************************************/
void _phy_mii_write_0( void )
{
  int32_t j;

  for(j = MDC_WAIT; j > 0; j--){
    EtherC.PIR.LONG = 0x00000002;
  }
  for(j = MDC_WAIT; j > 0; j--){
    EtherC.PIR.LONG = 0x00000003;
  }
  for(j = MDC_WAIT; j > 0; j--){
    EtherC.PIR.LONG = 0x00000003;
  }
  for(j = MDC_WAIT; j > 0; j--){
    EtherC.PIR.LONG = 0x00000002;
  }
}

/* End of File */
```

3.17 Sample Program List "phy.h" (1/2)

```
/*******************************************************************************
*   DISCLAIMER
*
*   This software is supplied by Renesas Electronics Corp. and is only
*   intended for use with Renesas products.  No other uses are authorized.
*
*   This software is owned by Renesas Electronics Corp. and is protected under
*   all applicable laws, including copyright laws.
*
*   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
*   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
*   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
*   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
*   DISCLAIMED.
*
*   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
*   ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
*   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
*   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
*   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
*
*   Renesas reserves the right, without notice, to make changes to this
*   software and to discontinue the availability of this software.
*   By using this software, you agree to the additional terms and
*   conditions found by accessing the following link:
*   http://www.renesas.com/disclaimer
********************************************************************************
*   Copyright (C) 2009(2010). Renesas Electronics Corporation. All Rights Reserved.
*""FILE COMMENT""*********** Technical reference data **************************
*   System Name : SH7216 Sample Program
*   File Name   : phy.h
*   Version     : 2.00.00
*   Device      : SH7216
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
*               : C/C++ compiler package for the SuperH RISC engine family
*               :                             (Ver.9.03 Release00).
*   OS          : None
*   H/W Platform: R0K572167 (CPU board)
*   Description : Ethernet PHY device driver
******************************************************************************
*   History     : Jun.10.2009 Ver.1.00.00
*               : Jul.23,2010 Ver.2.00.00 Comply with the Renesas API
*""FILE COMMENT END""**********************************************************/
#ifndef PHY_H
#define PHY_H

```

3.18 Sample Program List "phy.h" (2/2)

```
/* ==== Macro definition ==== */
/* ---- Standard PHY Registers ---- */
#define BASIC_MODE_CONTROL_REG       0
#define BASIC_MODE_STATUS_REG        1
#define PHY_IDENTIFIER1_REG          2
#define PHY_IDENTIFIER2_REG          3
#define AN_ADVERTISEMENT_REG         4
#define AN_LINK_PARTNER_ABILITY_REG  5
#define AN_EXPANSION_REG             6

/* ---- Media Independent Interface ---- */
#define  PHY_ST    1
#define  PHY_READ  2
#define  PHY_WRITE 1
#define  PHY_ADDR  1

#define  MDC_WAIT  3

/* ---- PHY return definitions ---- */
#define R_PHY_OK    0
#define R_PHY_ERROR -1

/* ====  Prototype declaration ==== */
/* ----External prototypes ---- */
int16_t phy_init( void );
void      phy_set_100full( void );
void      phy_set_10half( void );
int16_t phy_set_autonegotiate( void );

#endif /* PHY_H */

```

4. References

- Software Manual
  SH-2A/SH2A-FPU Software Manual Rev. 3.00
  The latest version of the software manual can be downloaded from the Renesas Electronics website.

- Hardware Manual
  SH7214 Group, SH7216 Group Hardware User's Manual Rev. 2.00
  The latest version of the hardware user's manual can be downloaded from the Renesas Electronics website.

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Jan.30.10	—	First edition issued
2.00	Sep.17.10	All	Updated to support the Renesas API in the sample program

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.
   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.
   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.
5. Differences between Products
   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

Notice

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan, R.O.C.
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141