Renesas RA Family

RA Ethernet Design and Custom PHY Setup using FSP

Introduction

This application note describes Ethernet designs in general, provides a brief introduction to the RA Ethernet controller and interface to the PHY peripheral. It provides design guidelines when using the RA MCU with RMII modes for Ethernet specific applications.

The application note also addresses utilizing the FSP configurator to incorporate the Ethernet module and ensure proper configuration. Furthermore, it delves into integrating a new PHY into custom boards and providing software support through FSP. It encompasses communication via MDIO from the MCU to the PHY.

Additionally, the application note also includes guidance on utilizing the Wake-on-LAN (WOL) feature and implementing Low Power Mode (LPM) in Ethernet-based designs. Lastly, it provides instructions for debugging an Ethernet design based on RA.

Applies to:

- RA MCU Group with Ethernet Peripherals

Contents

1. Introduction to Ethernet Connectivity

Ethernet connectivity stands as a widely utilized technology within the realm of data communications, typically employed within Local Area Networks (LANs). Governed by the IEEE 802.3 standards, Ethernet accommodates various speeds and mediums for its operation. In recent times, Ethernet communication has found application across diverse sectors such as home automation, industrial automation, and consumer electronics. Additionally, Ethernet serves as a prevalent choice in IoT applications, emerging as the default connectivity option. Particularly noteworthy is its utilization in Power over Ethernet (PoE) systems, wherein system power is drawn directly from the communication line, negating the need for additional power connections. This characteristic renders Ethernet an ideal connectivity solution across a myriad of applications.

1.1 General Overview

For networking connectivity challenges, Renesas Microcontrollers (MCUs) and Microprocessors (MPUs) with Ethernet support make it easy to implement Ethernet solution in your applications.

The Renesas RA group of microcontrollers (MCUs) uses the high-performance Arm® Cortex® core and offers Ethernet MAC with built in DMA, to ensure high data throughput.

Renesas RA Ethernet provides the following functionality:

- 10BASE-T/100BASE-TX IEEE 802.3 Compliant Ethernet.
- Half- and full-duplex support.
- Transmit/receive processing (Blocking and Non-Blocking).
- Auto-negotiation support.
- Magic packet (Magic Pattern) detection mode support.
- Flow control compliant with IEEE802.3x.
- Media Independent Interface (MII), Reduced Media Independent Interface (RMII), compliant with the IEEE802.3u standard.
- MDC/MDIO Management Interface for PHY Register Configuration.
- Wake-on-LAN (WOL) signal output.
- Hardware filtering of received multicast packets with a MAC address.

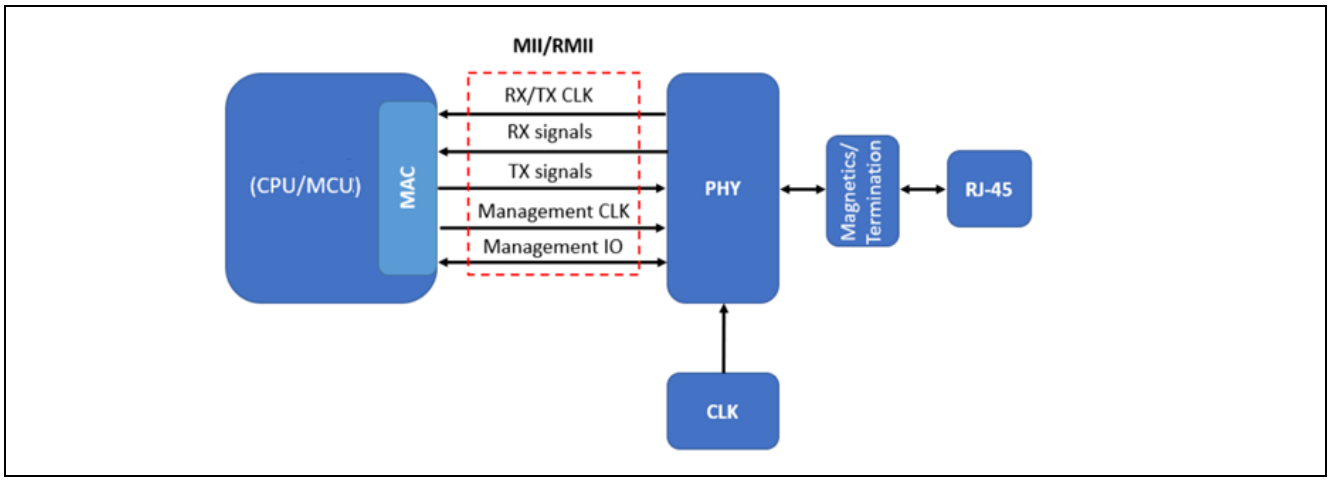

Figure 1. Ethernet Architecture Block Diagram

1.2 Ethernet MAC Controller

RA MCUs provide a one or two channel Ethernet Controller (ETHERC) compliant with the Ethernet or IEEE802.3 Media Access Control (MAC) layer protocol. Each ETHERC channel has one channel of the MAC layer interface. Connecting the MCU to the physical layer LSI (PHY-LSI) allows transmission and reception of frames compliant with the Ethernet/IEEE802.3 standard. The ETHERC is connected to the Ethernet DMA Controller (EDMAC), so data can be transferred without much intervention of the CPU. Figure 2 shows the RA Ethernet Module blocks.

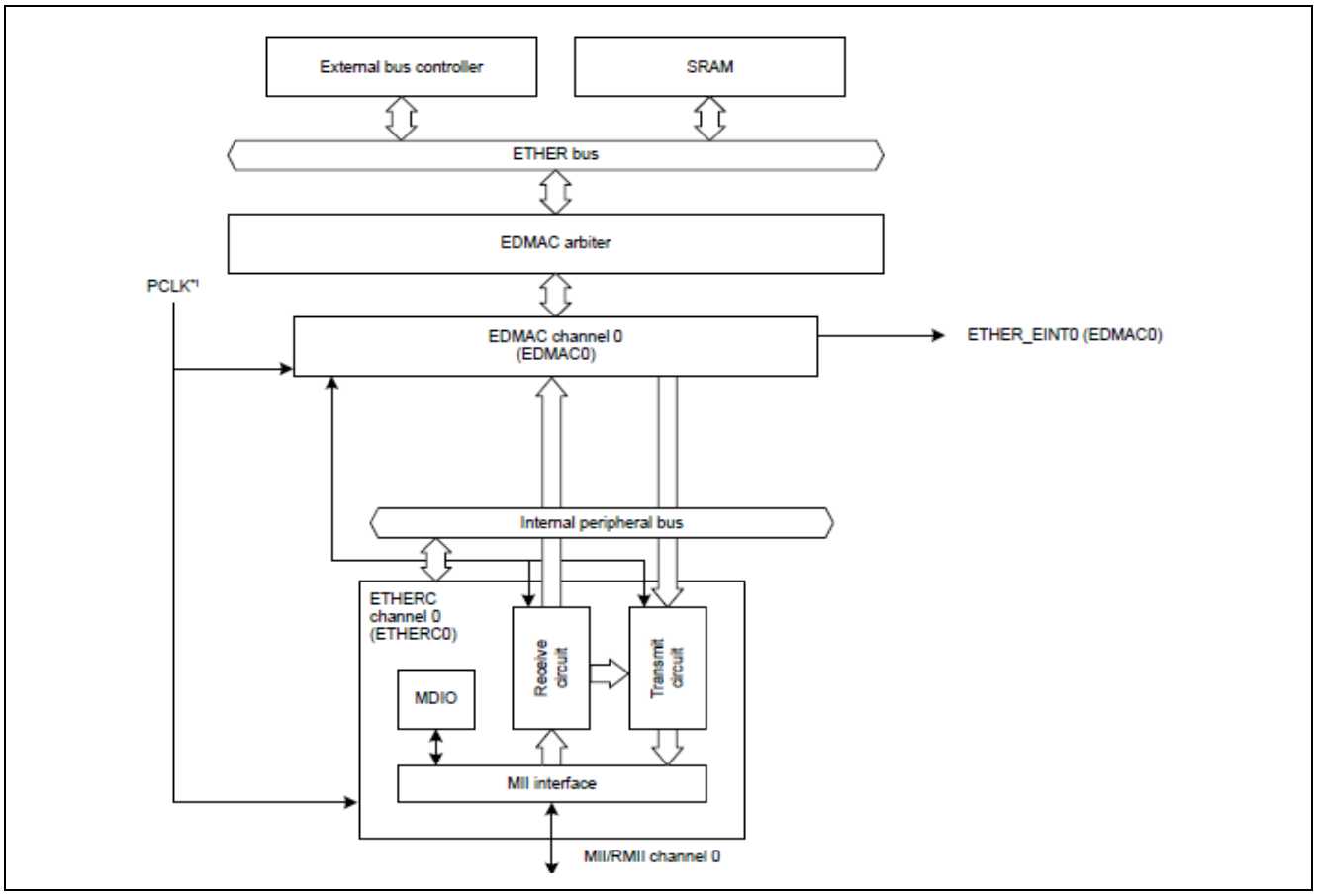

Figure 2. RA MCU Ethernet Controller Block Diagram

The Ethernet Media Access Controller (MAC) meets the essential protocol requirements for operating an Ethernet/IEEE 802.3-compliant node and provides an interface between the MCU host subsystem and the external Ethernet PHY.

The Ethernet MAC can operate in either 100-Mbps or 10-Mbps mode by supporting both half-duplex and full-duplex modes. When operating in half-duplex mode, the MAC complies fully with Section 4 of ISO/IEC 8802-3 (ANSI/IEEE standard) and ANSI/IEEE 802.3 standards. When operating in full-duplex mode, the MAC complies with IEEE 802.3x full-duplex operation standard.

The MAC also provides programmable enhanced features designed to minimize host supervision, bus utilization, and pre- or post-message processing. These features include the ability to disable retries after a collision, dynamic FCS (Frame Check Sequence) generation on a frame-by-frame basis, automatic pad field insertion and deletion to enforce minimum frame size attributes, and automatic retransmission and detection of collision frames.

The primary functionalities of the MAC are:

- Transmission and reception of encapsulated data messages.
- Frame boundary delimitation, frame synchronization.
- Error detection (physical medium transmission errors).
- Media control and access management.
- Medium allocation (collision detection, except in full-duplex operation).
- Contention resolution (collision handling, except in full-duplex operation).
- Flow control during full-duplex mode.
- Decoding of control frames (PAUSE command) and disabling the transmitter.
- Generation of control frames.
- Interfacing to the external PHY.

1.3 Ethernet DMA Controller

RA MCUs provide a dedicated channel to the Ethernet DMA Controller (EDMAC) for the Ethernet Controller (ETHERC). The EDMAC controls most of the transmit and receive buffer management for communications. This reduces the load on the CPU and allows efficient data transmission and reception.

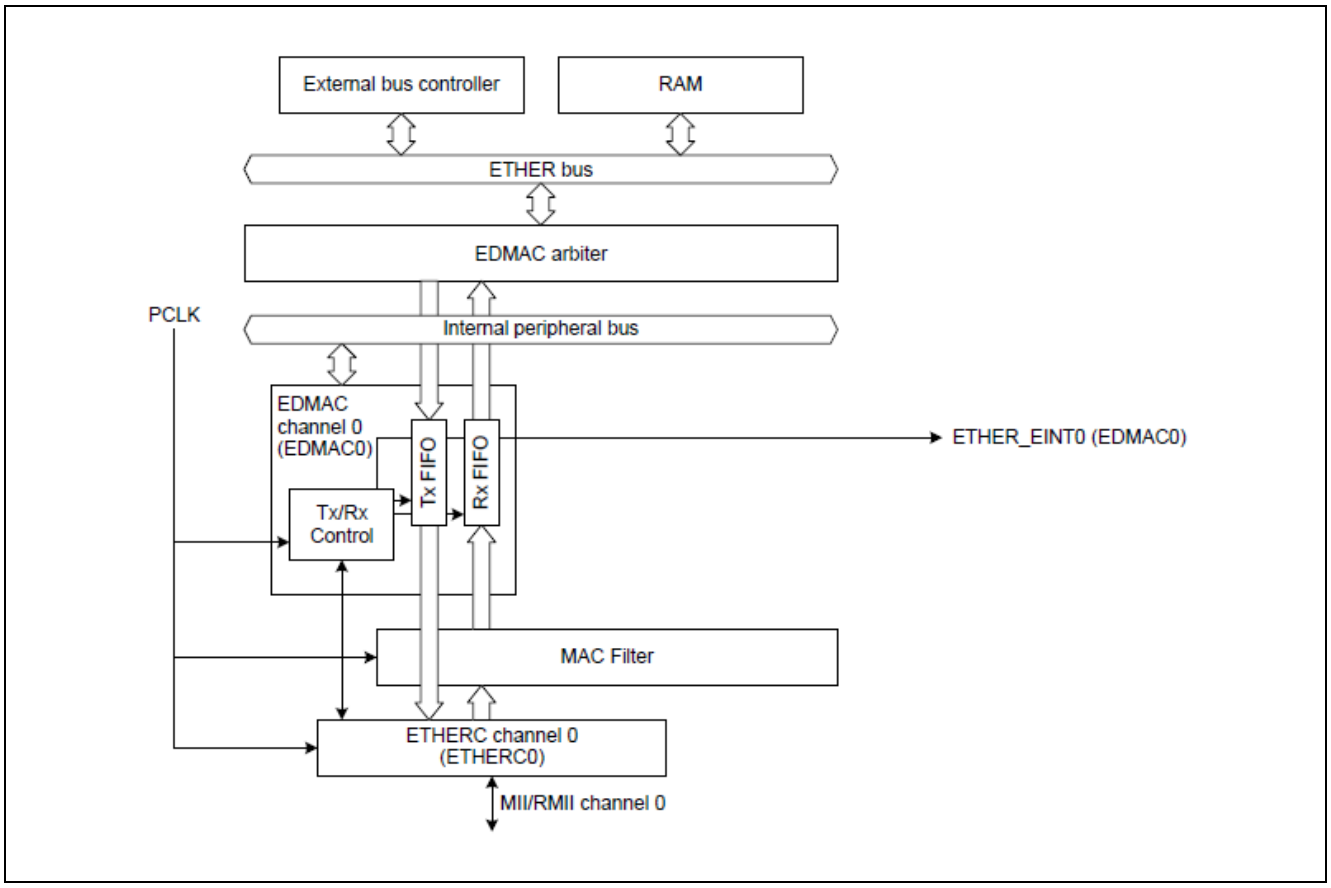

Figure 3. RA MCU Ethernet DMA Controller Block Diagram

The DMA controller uses descriptors to move data from the source address to the destination with little CPU intervention. DMA is designed for packet-oriented data transfer, such as frames in Ethernet. It supports single buffer frame transmission and reception (1 buffer per frame) and multi-buffer frame transmission and reception (multiple buffers per frame).

It minimizes system bus occupancy time using block transfer (32-byte units). It also writes back the transmit or receive frame state to descriptors and inserts padding in the receive data.

The controller can be programmed to interrupt the CPU, such as when the frame sending and receiving operations are completed, and under other normal / error conditions.

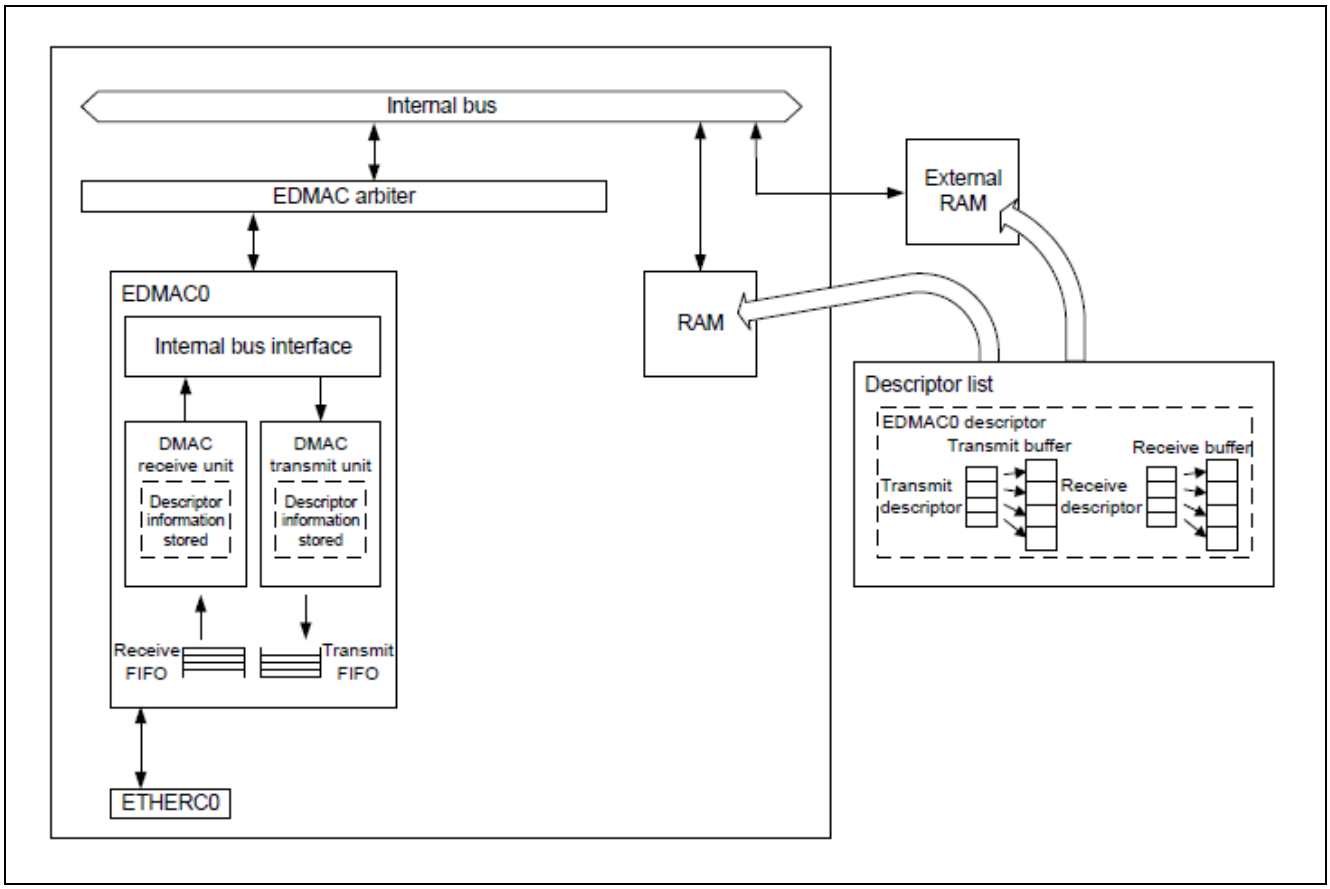

Figure 4. RA MCU Ethernet DMA Controller Block Diagram

1.4 Ethernet PHY

The PHY is the physical interface transceiver that implements the physical layer. The IEEE-802.3 standard defines the Ethernet PHY. It complies with the IEEE-802.3 specifications for 10BaseT (clause 14) and 100BaseTX (clauses 24 and 25).

The PHY sits between the media access controller (MAC) device, and the network connection. The MAC device can either be a microcontroller or a processor. The connection to the MAC layer can be a Media Independent Interface (MII) or a Reduced Media Independent Interface (RMII). These interfaces are defined in the IEEE 802.3 standard and offer either reduced number of signal lines, higher data speeds, or both.

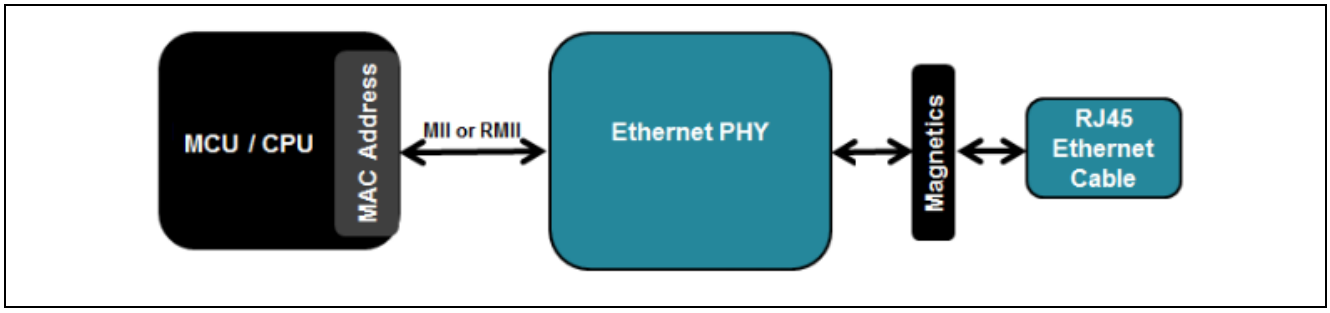

Figure 5. RA MCU Ethernet Interfacing to PHY Block Diagram

Functionally, the PHY can be divided into the following sections:

- 100Base-TX transmit and receive.
- 10Base-T transmit and receive.
- Internal MII/RMII interface to the Ethernet Media Access Controller.
- Auto-negotiation to automatically determine the best speed and duplex possible.
- Management Control (MDIO) to read status registers and write control registers.

1.5 Management Data Input/Output (MDIO) Interface

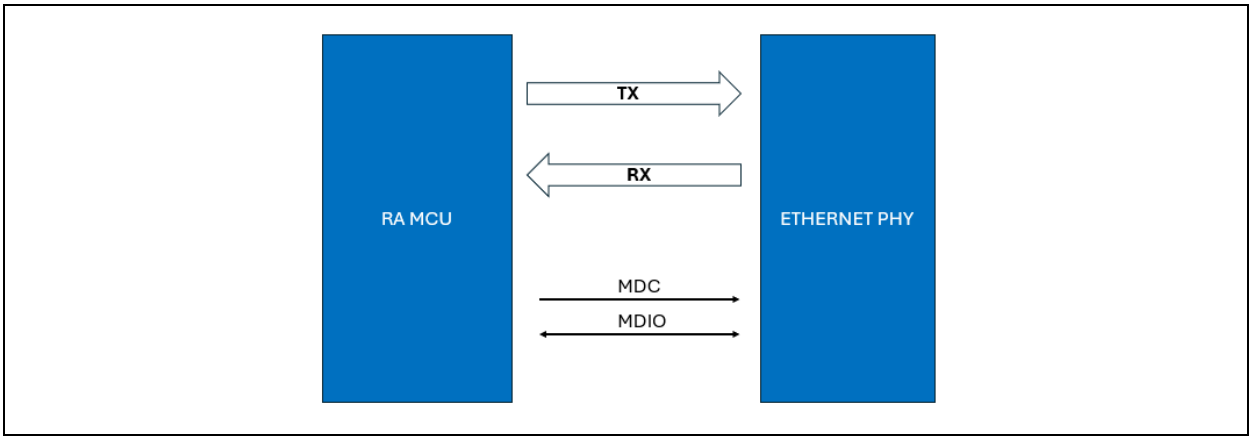

Figure 6. RA MCU Interfacing to PHY using MDIO interface.

The MDIO (Management Data Input/Output) interface is a standardized communication protocol used for configuring and monitoring Ethernet PHY (Physical Layer) devices in networking equipment such as switches, routers, and network interface cards. The MDIO interface enables the MCU or Ethernet MAC (Media Access Control) controller to communicate with the PHY to perform tasks such as setting PHY configuration registers, retrieving status information, and managing link negotiation.

MDIO operates as a serial interface, typically consisting of two signals - MDIO (Management Data Input/Output) and MDC (Management Data Clock). The MDIO line carries data between the host and the PHY, while the MDC line provides the clock signal for synchronous communication.

The MDIO interface allows the RA MCU or MAC controller to access the internal registers of the Ethernet PHY. These registers control various aspects of PHY operation, including link speed, duplex mode, auto-negotiation, power management, and diagnostics. MDIO facilitates the auto-negotiation process between the host and the PHY to establish the optimal link parameters, such as speed and duplex mode. During auto-negotiation, the host and PHY exchange information about their capabilities and negotiate the highest common denominator for link configuration.

The MDIO interface enables the host to query the PHY for status information, including link status, link speed, duplex mode, and error conditions. This allows the host to monitor the health and performance of the Ethernet connection in real-time. Through the MDIO interface, the host can configure various parameters of the PHY to adapt to different network environments and requirements. This includes setting the PHY's operating mode, enabling or disabling features, and adjusting performance settings.

MDIO operates using a primary-secondary device protocol, where the MCU (master) communicates with one or more PHY devices (slaves) on the same MDIO bus. Communication occurs in half-duplex mode, with the host transmitting commands or data to the PHY devices and receiving responses or status information.

Each PHY device on the MDIO bus is assigned a unique 5-bit address, allowing the host to select and communicate with specific PHY devices. The address space typically ranges from 0 to 31, with addresses 0 and 31 reserved for special purposes (e.g., broadcast and all-slave operations).

Ethernet PHY devices have a set of standard registers, as defined by the IEEE 802.3 specification, which are used to control and monitor the operation of the PHY. These registers allow for configuration, status reporting, and control of various PHY functions. The Register Address from 0 to 15 (0x00 to 0x0F) maps to the standard register as required by Clause 22 of the 802.3 standard.

The register addresses 16 to 31 (0x10 to 0x1F) are reserved for vendor-specific registers according to the IEEE 802.3 specification. These registers allow PHY manufacturers to implement additional features and provide extra configuration options that go beyond the standard set of registers defined by the specification.

MDIO communication consists of frames transmitted serially over the MDIO line. Each frame includes a start bit, followed by the 5-bit PHY address, a 5-bit register address, a 16-bit data field, and an end bit. The register address specifies the internal register of the PHY device being accessed, allowing the host to read from or write to specific registers.

The host sends a read command frame to the PHY device, specifying the PHY address and the register address to read from. The PHY responds with a data frame containing the contents of the specified register.

The host sends a write command frame to the PHY device, specifying the PHY address, the register address to write to, and the data to be written. The PHY acknowledges the write operation.

MDIO communication is synchronous, with data being sampled on the rising edge of the MDC clock signal. The MDC signal is generated by the host and shared with all PHY devices on the MDIO bus.

MDIO communication follows a strict timing specification, with specific requirements for setup and hold times, clock frequency, and data transmission rates. Handshake signals, such as the start and end bits, ensure proper synchronization and framing of data during communication.

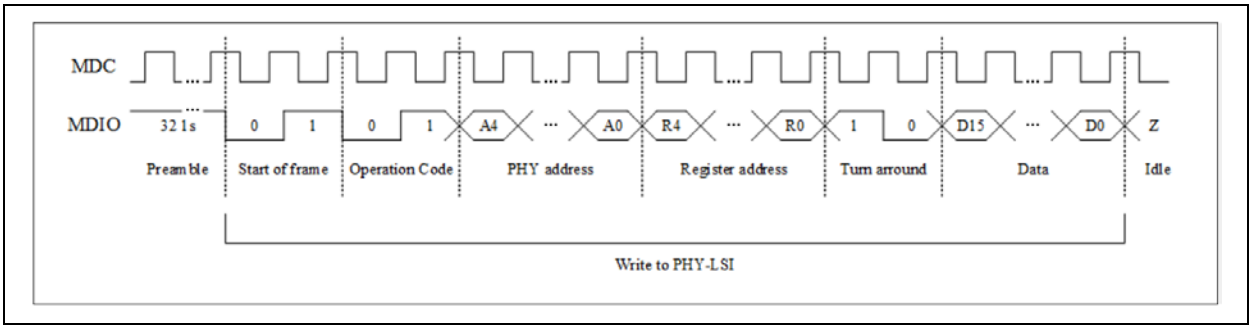

Figure 7. RA MCU PHY Write using MDIO interface.

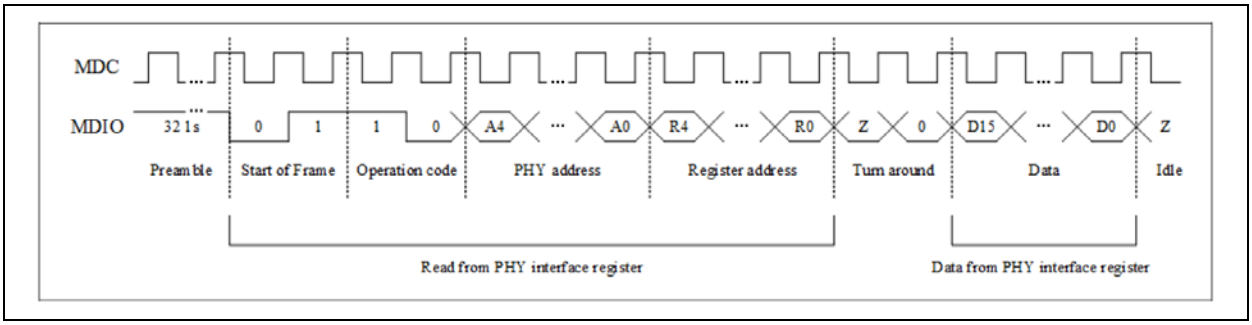

Figure 8. RA MCU PHY Read using MDIO interface.

2. General Design Guidelines

While designing an application using ETHERC and EDMAC on an RA MCU, there are some Hardware and Software guidelines that need to be followed. These guidelines are recommended best practices.

Due to the board size and number of pins required for Ethernet interface, users can choose RMII or MII. RMII uses 10 pins whereas MII uses 18 pins. This document focuses on RMII as it is more commonly used.

Proper clock source recommended for the MII and RMII is necessary with recommended accuracy of +/- 50 ppm.

The MCU also needs to select the proper clock source for the Ethernet controller; this can be accomplished for RA by selecting the proper clock source using FSP configurator within the allowed range. When using the ETHERC, set the clock to ICLK = PCLKA, and PCLKA needs to meet 12.5 MHz ≤ PCLKA ≤ 120 MHz.

It is recommended to use the Link status pin in the design, so that a change in the link status can be handled in the software without much CPU intervention and implementation in the user code. Refer to the section 5.1 for details about configuring the link status handling using FSP.

Renesas has a separate application note that provides guidelines for interfacing from the Ethernet controller to a PHY transceiver, placement, signal routing, and different layers to manage the signals for isolation. This *Ethernet Hardware Design Guide* (R01AN3342EJ0101) document can be used as a reference during board designs.

Following are some hardware layouts recommended guidelines:

- Although the trace length between the PHY and MAC is generally short enough to ignore transmission line effects, you should pay attention to impedance when routing the signal and clock traces when the traces are electrically long. The trace impedance for RMII/MII is 50 Ohms +/- 10%. This is the normal standard for most of the routing. Additionally in some designs, you may also notice series resistors of 30 Ohms.
- The differential Ethernet signals from the PHY to the connector must be impedance controlled, and those need to be 100-ohm differential impedance.
- Since MII and RMII require clock signals between the PHY and MAC, best practices for routing clock signals should be used with shorter trace length. Clock lines should also be shielded with GND lines to prevent crosstalk through capacitive coupling, especially when longer traces are necessary.
- All MII/RMII signal traces should be routed as short as possible on a single layer, and traces should be routed in a straight path. If you must turn a corner with a signal trace, the trace should bend by no more than 45 degrees. Clock lines should also be shielded with GND lines to prevent crosstalk through capacitive coupling, especially when longer traces are necessary.
- Because the TX and RX data signals are triggered by the rising edge of a clock, communication in MII and RMII is synchronous. Thus, the data lines and the clock line between the MAC and PHY should be length matched. The allowed deviation in length matching depends on the rise/fall time for digital signals between these two elements, although it is generally recommended that any deviation be less than 10 mm as MII and RMII use TTL logic. Again, the allowed trace length mismatch depends on the rise/fall time of digital signals.
- The final important point in distinguishing MII and RMII routing relates to the number of signals used in each standard. Some PHY devices support either standard, and so some of the pins on the PHY will be unused if you are using RMII. Your datasheets will tell you which pins need to be pulled-down and which can be safely left open. PHY Pin strapping is also required to by use of external pullup resistors to attain the required latch-in during the device powerup and reset. Refer to PHY peripheral device user's manual for the available options.

2.1 Design Guidelines while using Arm TrustZone MCU

The design guidelines in this section are specific to some of the Arm® TrustZone® MCUs which needs special configurations to access the non-secure buffers.

For the RA MCU with EDMAC and Arm TrustZone support, the EDMAC peripheral can only be configured with the non-secure attribute, which is described by the Peripheral Security Attribution section in the User's Manual. Therefore, the EDMAC on these target devices always requires the non-secure buffers to be located in non-secure attributed RAM area.

Therefore, the RA serial boot interface (SCI/USB Boot Mode) needs to be accessible on the customer's board to program the IDAU register and setup the Arm® TrustZone® boundaries. This allows use of the Renesas Device Partition Manager to properly size the required non-secure memory partition in RAM.

2.2 Sample Design while using RA MCU with RMII

RMII is one of the most commonly used interfaces for Ethernet design due to the usage of lesser number of pins. Figure 9 shows the RA MCU used with RMII interface to connect with ICS1894-32 PHY transceiver device. For RMII, the REF50CK0 (50 MHz) clock is a required input signal instead of 25 MHz for MII.

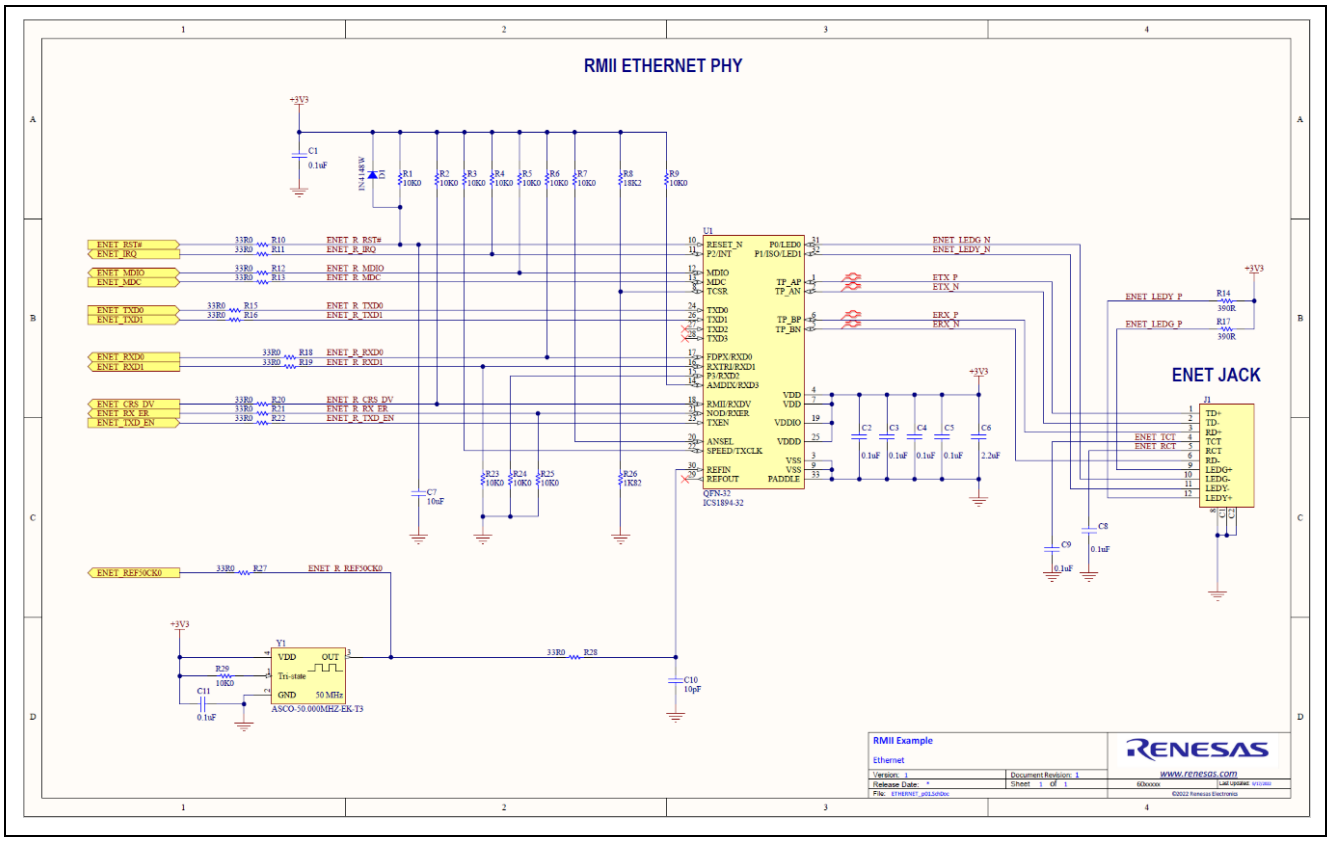

Figure 9. Sample RA MCU with RMII Interface for Ethernet

3. FSP Configuration for Ethernet Module

3.1 Ethernet Driver Configuration

This section has the FSP Ethernet configuration. Users are required to configure the selection based on the required modes. Details of each configuration is explained in detail in the RA Flexible Software Package Documentation: Introduction (renesas.github.io) under **Modules** > **Networking** > **r_ether** and **r_ether_phy**

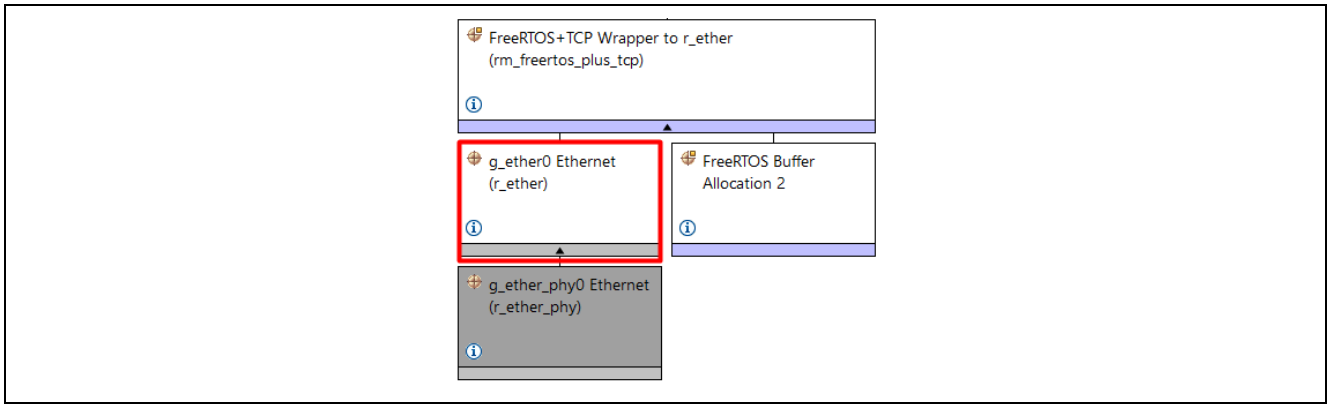

Figure 10. FSP Configuration for Ethernet

Table 1. FSP Configuration for Ethernet

Configuration	Options	Default	Description
Channel	0	0	Selects the ethernet channel number depending on the channel used in the design.
MAC address	Must be a valid MAC address	00:11:22:33:44:55	MAC address of this channel.
Zero-copy Mode	• Disable • Enable	Disable	Enable or disable zero-copy mode.
Flow control functionality	• Disable • Enable	Disable	Enable or disable flow control.
Multicast Mode	• Disable • Enable	Enable	Enable or disable multicast frame reception.
Promiscuous Mode	• Disable • Enable	Disable	Enable or disable this option to receive packets addressed to other NICs.
Broadcast filter	Must be a valid non-negative integer with maximum configurable value of 65535.	0	Limit of the number of broadcast frames received continuously.
Number of TX buffer	Must be an integer from 1 to 8	1	Number of transmit buffers.
Number of RX buffer	Must be an integer from 1 to 8	1	Number of receive buffers.
Allocate RX buffer	• Disable • Enable	Enable	Enable or disable the allocation of the RX buffer when generating the configuration structure.
Buffer size	Must be at least 1514 which is the maximum Ethernet frame size.	1514	Size of Ethernet buffer.
Padding size	• Disable • 1 Byte • 2 Bytes • 3 Bytes	Disable	The padding size that is automatically inserted into the received packets.
Padding offset	Must be less than 64 bytes.	0	The offset into a receive buffer to insert padding bytes.
Interrupt priority	MCU Specific Options	Priority 12	Select the EDMAC interrupt priority.
Callback	Name must be a valid C symbol	NULL	Callback provided when an ISR occurs.

3.2 PHY Configuration

This section has the FSP configuration of the Ethernet PHY. The specific configuration details are shown in the table. The FSP provides a Default configuration for the ether PHY. For Custom boards, this needs to be configured based on the board design.

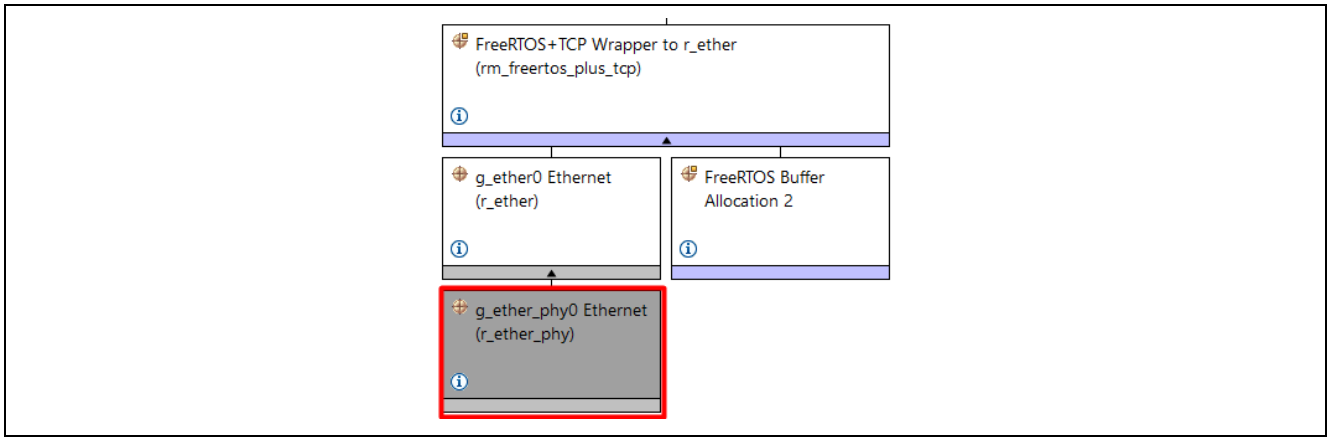

Figure 11. FSP Configuration for Ethernet PHY

Build Time Configurations for r_ether_phy

The following build time configurations are defined in `fsp_cfg/r_ether_phy_cfg.h`:

Table 2. Build Time Configuration for Ethernet PHY

Configuration	Options	Default	Description
Parameter Checking	Default (BSP) • Enabled • Disabled	Default (BSP)	If the selected code for parameter checking is included in the build.
Select PHY (DEPRECATED)[1]	Default • Other • KSZ8091RNB • KSZ8041 • DP83620 • ICS1894	Default	Config -> Select PHY is deprecated. Enable support for a PHY in the Config section and select it with Module -> PHY LSI type. Set Select PHY to Default to hide this message.
KSZ8091RNB Target	• Disabled • Enabled	Disabled	Select whether to use KSZ8091RNB Phy LSI or not.
KSZ8041 Target	• Disabled • Enabled	Disabled	Select whether to use KSZ8041 Phy LSI or not.
DP83620 Target	• Disabled • Enabled	Disabled	Select whether to use DP83620 Phy LSI or not.
ICS1894 Target	• Disabled • Enabled	Disabled	Select whether to use ICS1894 Phy LSI or not.
Reference Clock	Default • Enabled • Disabled	Default	Select whether to use the RMII reference clock. Selecting 'Default' will automatically choose the correct option when using a Renesas development board.

[1] Note: Select PHY (DEPRECATED) will be removed from the FSP 4.1.0 release and all future releases. Instead, the “KSZ8091RNZ Target” or “KSZ8041 Target” or “DP83620 Target” or “ICS1894 Target” will be used based on the PHY LSI used on the board. For user specific PHY a new XML needs to be added to support the selection of new PHY.

Configuration	Options	Default	Description
Channel	• 0 • 1	0	Select the Ethernet controller channel number.
PHY-LSI Address	• Specify a value between 0 and 31.	0	Specify the address of the PHY-LSI used.
PHY-LSI Reset Completion Timeout	• Specify a value between 0x1 and 0xFFFFFFFF.	0x00020000	Specify the number of times to read the PHY-LSI control register while waiting for reset completion. This value should be adjusted experimentally based on the PHY-LSI used.
Select MII type	• MII • RMII	RMII	Specify whether to use MII or RMII.
PHY LSI type	• Kit Component • DEFAULT • KSZ8091RNB • KSZ8041 • DP83620 • ICS1894	Kit Component	Select the PHY LSI target. Selecting 'Kit Component' will automatically choose the correct option when using a Renesas development board.
MII/RMII Register Access Wait-time	• Specify a value between 0x1 and 0x7FFFFFFF.	8	Specify the bit timing for MII/RMII register accesses during PHY initialization. This value should be adjusted experimentally based on the PHY-LSI used.
Flow Control	• Disable • Enable	Disable	Select whether to enable or disable flow control.

3.3 Adding a New PHY Device

Adding a new PHY device to an RA design requires changes in the hardware and software (FSP).

Hardware:

Follow and adopt the aforementioned design guidelines for the selected RMII interface. Reference designs from Renesas can be used as a basis while developing the custom boards. Below are some of the important considerations.

PHY pins need to be strapped to configure the board to the required modes of operation and for PHY address selection. Strapping is the use of external pullup resistors to attain the required latch-in during the device powerup and reset. Refer to PHY peripheral device user's manual for the available options.

MDIO line typically requires an external pull-up resistor. This is necessary because the MDIO line is an open-drain or open-collector bus. In such configurations, the line can only be pulled low by the connected devices (the PHY or the MAC) but cannot be driven high by them. Instead, the high state is achieved through an external pull-up resistor.

- PHY address configuration – FSP Ethernet Controller HAL treats the default PHY address as 0
- Mode configuration – The default operation mode of PHY after the Power-On Reset should support auto-negotiation(The speed and Mode are selected based on the negotiation with peer end).

Note: With reference to the Figure 9Figure 6, for the PHY ICS-1894-32, we can see how the PHY LSI address is derived based on the PHY address Bit pins configured in the Hardware (P3, P2, P1, P0). Here P3

is tied to GND which makes it read as logic 0, P2 is tied to 3.3V, and which is read as logic 1, P1 and P0 are logic 0. This makes the LSI PHY address as 0100b = 4

Identify if the target RA device has the Arm® TrustZone® security features. Design needs to include the serial boot interface for configuring the IDAU (Implementation Defined Attribution Unit) registers.

When using a crystal oscillator on a PHY device, verify the drive level of the crystal clock. The drive level should be a in accordance with PHY specification. To obtain the best performance from a clock source, it requires a clock accuracy of better than +/- 50 PPM over the entire temperature range of operation. Refer to the crystal oscillator data sheet- and the PHY Data sheet for the details while selecting the device clock and its clock drive level.

For the PCB layout, the serial termination resistors on RMII interface should be placed as close as possible with all drive pins of RMII.

For the differential pairs routing from PHY to transformer inside RJ45, it is better to try as much as possible to keep the impedance for TX pair and RX pair the same in the PCB layout.

Software:

Based on the existing PHY support provided by FSP, when the user adds new PHY device there are modifications required from the FSP end as well.

Generally, the PHY devices have the common set of General registers and the FSP can access them with the existing code with minimal addition of wrapper code. If the bits for some of these registers are different, then it is important to handle them correctly in code.

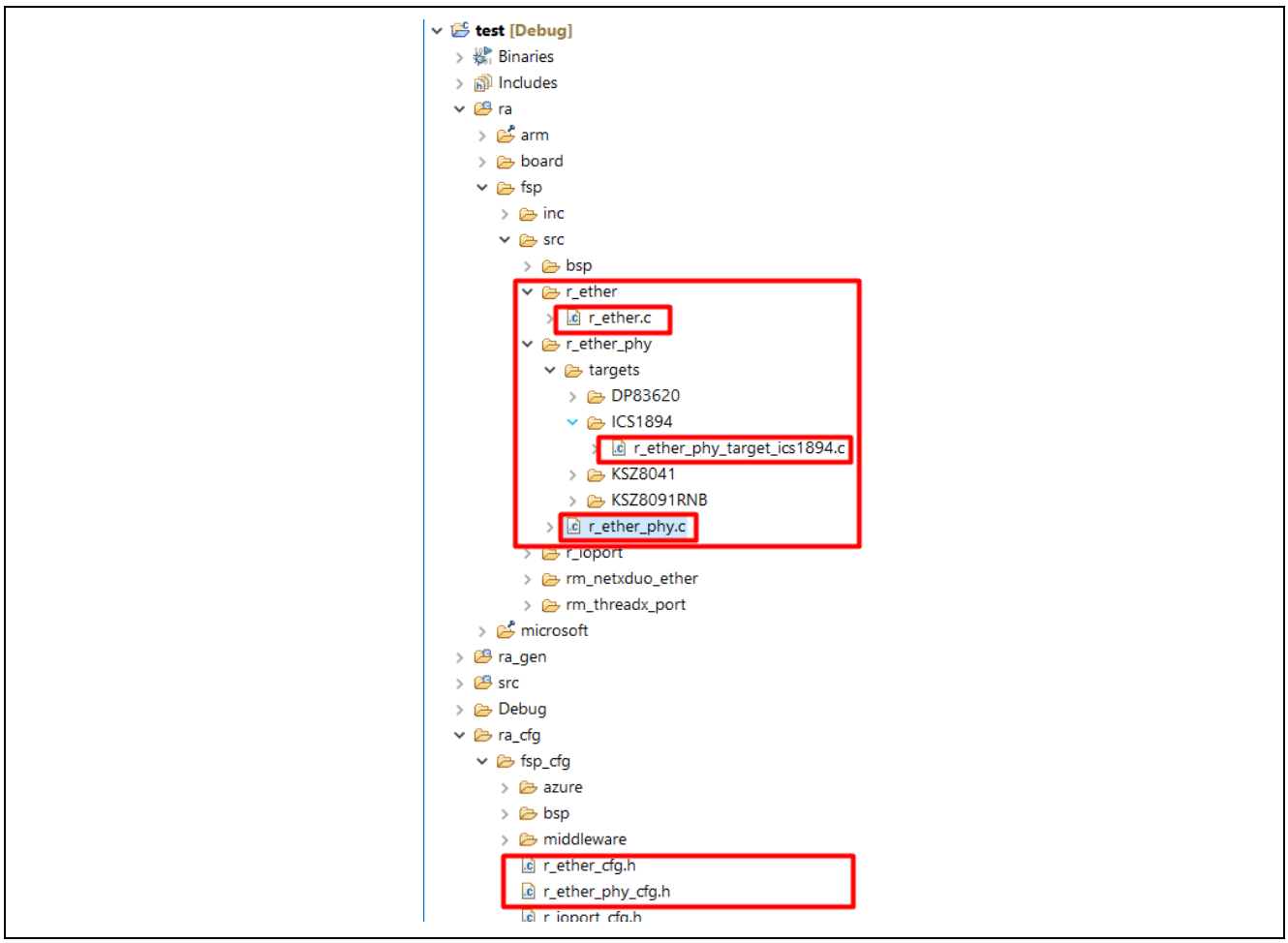

Figure 12. FSP Ethernet source code details

The FSP Ethernet file structure of the RA Ethernet and the supported PHY drivers are shown in the Figure 12. Fsp **>** src **>** r_ether **>** r_ether.c has the Ethernet driver and function call to the PHY driver.

Fsp **>** src **>** r_ether_phy **>** r_ether_phy.c has the generic API for the PHY and reading the PHY registers through the Management Data Input/Output (MDIO) interface. The code under these files remains the same even when you add the new PHY to the board.

```
const ether_phy_api_t g_ether_phy_on_ether_phy =
{
    .open                  = R_ETHER_PHY_Open,
    .close                 = R_ETHER_PHY_Close,
    .startAutoNegotiate    = R_ETHER_PHY_StartAutoNegotiate,
    .linkPartnerAbilityGet = R_ETHER_PHY_LinkPartnerAbilityGet,
    .linkStatusGet         = R_ETHER_PHY_LinkStatusGet,
    .chipInit              = R_ETHER_PHY_ChipInit,
    .read                  = R_ETHER_PHY_Read,
    .write                 = R_ETHER_PHY_Write
};
```

Figure 13. Ethernet PHY API details

R_ETHER_PHY_Open API, Resets the PHY device by writing into the PHY Control register (ETHER_PHY_REG_CONTROL) and checking the status of the Reset by reading it back.

R_ETHER_PHY_LinkPartnerAbilityGet API reads the ETHER_PHY_REG_STATUS bit to get the other side Link Physical Capability.

Similarly, R_ETHER_PHY_StartAutoNegotiate API Starts the Auto negotiation by setting the link ability and configuring what the Ethernet and PHY supports. These are standard Functionality which every PHY implements as part of the standard Registers.

With reference to the Figure 7 and Figure 8, in which the MDIO read and MDIO write signal flow diagram are captured. Below code R_ETHER_PHY_Write and R_ETHER_PHY_Read performs the same function for the MDIO read and write as per the IEEE 802.3, standard(22-12 of 22.2.4.5).

```
fsp_err_t R_ETHER_PHY_Write (ether_phy_ctrl_t * const p_ctrl, uint32_t reg_addr, uint32_t data)
{
    ether_phy_instance_ctrl_t * p_instance_ctrl = (ether_phy_instance_ctrl_t *) p_ctrl;

#if (ETHER_PHY_CFG_PARAM_CHECKING_ENABLE)
    FSP_ASSERT(p_instance_ctrl);
    ETHER_PHY_ERROR_RETURN(ETHER_PHY_ADDRESS_SIZE >= reg_addr, FSP_ERR_INVALID_ARGUMENT);
    ETHER_PHY_ERROR_RETURN(ETHER_PHY_REGISTER_DATA_SIZE >= data, FSP_ERR_INVALID_ARGUMENT);
    ETHER_PHY_ERROR_RETURN(ETHER_PHY_INTERFACE_STATUS_INITIALIZED == p_instance_ctrl->interface_status,
                           FSP_ERR_NOT_INITIALIZED);
#endif

    /*
     * The value is read from the PHY register by the frame format of MII Management Interface provided
     * for by Table 22-12 of 22.2.4.5 of IEEE 802.3-2008_section2.
     */
    ether_phy_preamble(p_instance_ctrl);
    ether_phy_reg_set(p_instance_ctrl, reg_addr, ETHER_PHY_MII_WRITE);
    ether_phy_trans_1to0(p_instance_ctrl);
    ether_phy_reg_write(p_instance_ctrl, data);
    ether_phy_trans_idle(p_instance_ctrl);

    return FSP_SUCCESS;
}                                   /* End of function R_ETHER_PHY_Write() */
```

Figure 14. Ethernet PHY Write code details.

```
fsp_err_t R_ETHER_PHY_Read (ether_phy_ctrl_t * const p_ctrl, uint32_t reg_addr, uint32_t * const p_data)
{
    ether_phy_instance_ctrl_t * p_instance_ctrl = (ether_phy_instance_ctrl_t *) p_ctrl;
    uint32_t data;
#if (ETHER_PHY_CFG_PARAM_CHECKING_ENABLE)
    FSP_ASSERT(p_instance_ctrl);
    ETHER_PHY_ERROR_RETURN(NULL != p_data, FSP_ERR_INVALID_POINTER);
    ETHER_PHY_ERROR_RETURN(ETHER_PHY_ADDRESS_SIZE >= reg_addr, FSP_ERR_INVALID_ARGUMENT);
    ETHER_PHY_ERROR_RETURN(ETHER_PHY_INTERFACE_STATUS_INITIALIZED == p_instance_ctrl->interface_status,
                           FSP_ERR_NOT_INITIALIZED);
#endif

    /*
     * The value is read from the PHY register by the frame format of MII Management Interface provided
     * for by Table 22-12 of 22.2.4.5 of IEEE 802.3-2008_section2.
     */
    ether_phy_preamble(p_instance_ctrl);
    ether_phy_reg_set(p_instance_ctrl, reg_addr, ETHER_PHY_MII_READ);
    ether_phy_trans_zto0(p_instance_ctrl);
    ether_phy_reg_read(p_instance_ctrl, &data);
    ether_phy_trans_idle(p_instance_ctrl);

    (*p_data) = data;

    return FSP_SUCCESS;
}                                       /* End of function R_ETHER_PHY_Read() */
```

Figure 15. Ethernet PHY Read code details.

The file fsp**>** src **>** r_ether_phy **>** targets **>** xxxxxxx **>** r_ether_phy_target_xxxxxxx.c is the vendor specific driver related to the PHY under the target selection. When the user adds a new PHY to the board, the corresponding FSP driver needs to be created. For the new PHY support, create a new folder under r_ether_phy **>** target; refer to the existing supported target boards used in the FSP, and add the necessary code under this folder to operate new PHY device. Make them read-only so that FSP will not overwrite these files on Project Generation. r_ether_phy_target_xxxxxxx.c has the PHY LSI specific initialization and it will be different for different PHY modules used in the design. Since the initialization is vendor specific Hence the code will be part of the r_ether_phy_target_xxxxxxx.c rather than common code.

Also, FSP allows to configure the ether_phy, based on the configurators, and it creates the r_ether_phy_cfg.h, which has the user desired configurations made under the properties window of the ether_phy module. These properties can be used when configuring the new PHY.

After the PHY related code is finalized, if the finalized PHY code needs to be used across different versions of FSP, it is recommended to make it configurable using the FSP configurator. Therefore, a pack file needs to be created along with an xml for the user configuration. This can be accomplished by creating/modifying the FSP pack files. The creation of the FSP pack files are covered in different application note (***Getting Started with the Wi-Fi Modules on FSP*** (r11an0486eu)) that users can refer to for more details.

Note: This Application Note does not have a bundled application to show case the reference code for the New PHY. Users can choose the different PHY from (**r_ether_phy** > **Select PHY**) using the FSP configurator and generate the code. Create 2 sets of projects and select different PHYs to compare the differences in the code under the folders listed in Figure 12. This also gives the changes required in the source code for the different PHYs.

Note: Adding a new PHY to the board does not require any changes to the Data path at the Ethernet and upper layers. FSP should work independent of the new PHY.

Note: Most of the HW reference designs provided by Renesas do not handle the LINK status pin. If the new design requires LINK status to be handled by the FSP, the HW needs to have the design in place connecting the LINK status pin from the PHY to the Ethernet Controllers ET0_LINKSTA. FSP has the associated configuration and code to handle this feature.

Note: In scenarios where the PHY does not provide the capability through hardware pin to report the Link status to the Ethernet controller, the users are required to handle the change in the link status by polling for the link status change in the application code. This can be done by periodically reading the Link status bit from the PHY register via the API (**R_ETHER_PHY_LinkStatusGet).**

3.4 Signal Drive Strength Configuration

The signal strength of the Ethernet pins used in pins configuration based on the Channel 0/1 needs to have the selection as drive capacity of “High”. This can be done using the pin tab of the **FSP configurator Pins** > **Pin Function** > **Peripherals** > **Connectivity: ETHRC →ETHERRC_RMII0/1**.

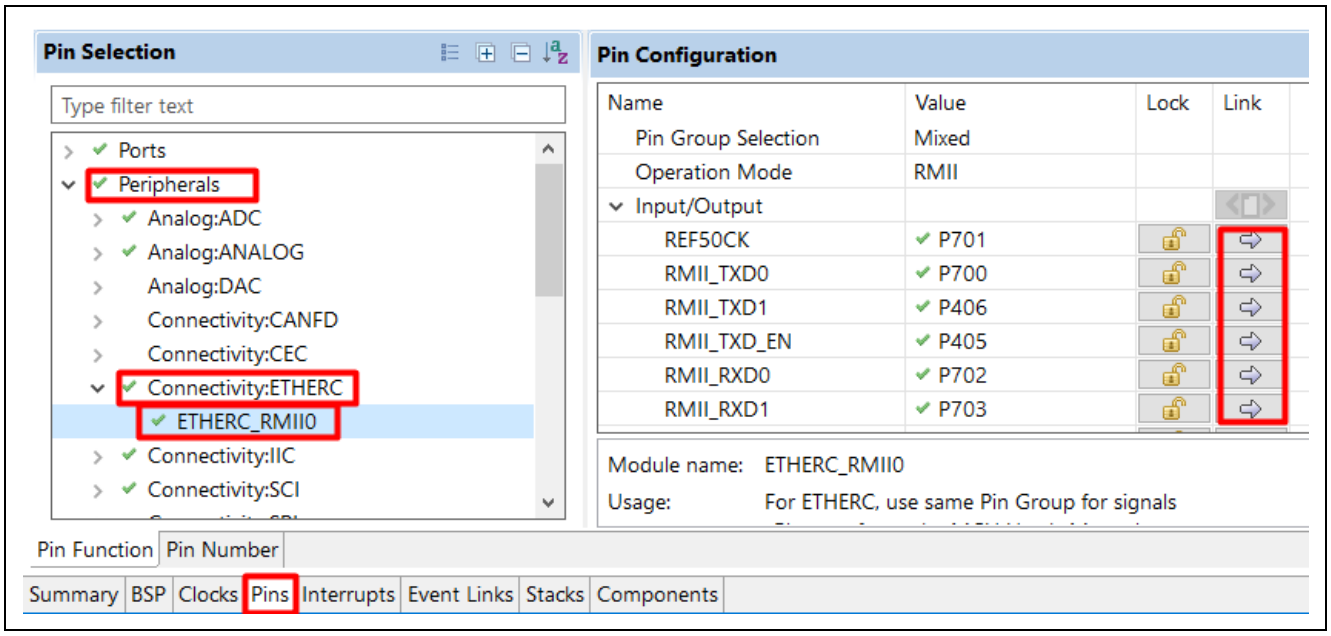

Figure 16. RA Ethernet Pin Drive Capacity Settings

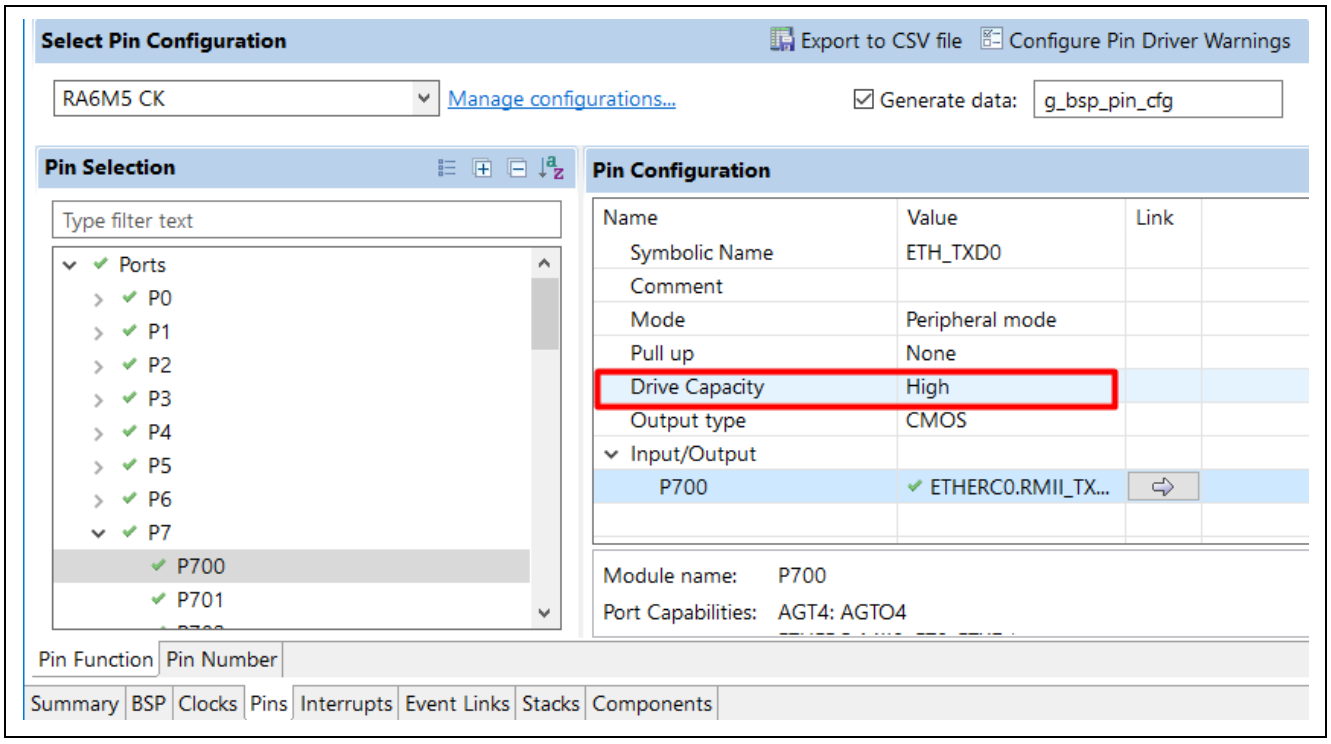

Figure 17. RA Ethernet Pin Drive Capacity Settings

4. Support of Low Power Modes and Wake-on-LAN

In RA-based Ethernet designs that require lower power implementations, we have two options to make it available for the customers.

Wake-on-LAN
Wake-on-LAN (WOL) keeps the Ethernet PHY in an active state but saves power by enabling the microcontroller (MCU), other components to power down selectively. These components are made to wake up when the Ethernet PHY receives a known “magic pattern” on the network. The *magic pattern is* a frame that is most often sent as a broadcast and contains anywhere within its payload 6 bytes of all 255 (0xFF 0xFF 0xFF 0xFF 0xFF 0xFF in hexadecimal), followed by sixteen repetitions of the target computer's 48-bit MAC address, for a total of 102 bytes.

The WOL feature enables users to save on overall system power while keeping the Ethernet transceiver awake when the system has well-defined trigger for wake ups.

Figure 18 shows a magic pattern scheme for which the Ethernet PHY will monitor ingress packets before signaling a wake up.

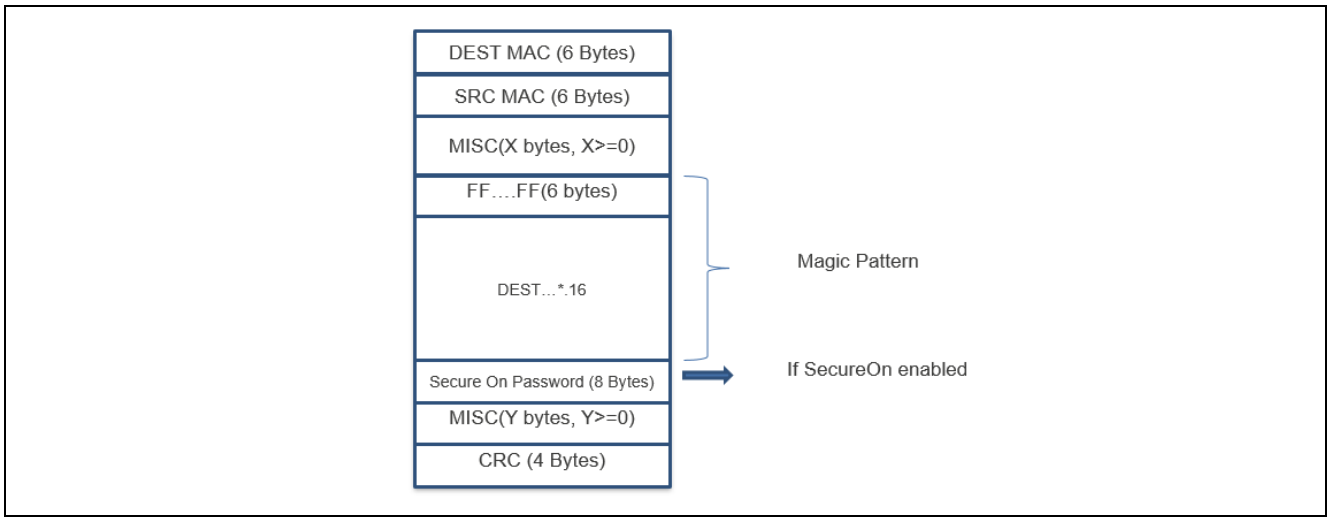

Figure 18. WOL Magic Pattern

The RA Ethernet controller supports Wake-on-LAN (WOL). WOL is a function to detect a Magic Pattern transmitted from a host device or other device, and to wake the MCU from a low power mode such as Sleep mode.

When the ETHERC detects a magic pattern, it outputs high on the ET0_WOL pin. Write 1 to the EDMAC0.EDMR.SWR bit to drive the ET0_WOL pin low.

Because a magic pattern is transmitted in broadcast mode, it is received regardless of the destination MAC address selected in the format. The ETHERC outputs high on the ET0_WOL pin only when the destination MAC address matches its own MAC address.

WOL implementation in RA can be accomplished by enabling this feature. This feature is available in the FSP Ethernet driver. Customers who want to use this feature need to configure the Ethernet with the API “wakeOnLANEnable”. Refer to FSP User’s manual for more details (7).

Energy-efficient Ethernet (IEEE 802.az):
This covers the details of Energy efficient Ethernet (EE) designed to save power when there is no traffic on the link. In Energy Efficient Ethernet, power is reduced when the port is down. Power is also reduced when the port is up and there is no traffic on it.

5. Debugging

5.1 Debugging RA Ethernet Controller using FSP

RA devices have integrated software support using the FSP. The devices are configured through the FSP configurator as mentioned in the previous sections. It is good practice to review the required Ethernet and PHY configurations are made for the device before debugging it. Refer the section 3 for more details for the Ethernet and PHY configurations.

Requirements for RA Ethernet Controller

For Arm® Cortex-M4 devices, when the Ethernet controller (ETHERC) and Ethernet DMA Controller (EDMAC) are used, PCLKA (Ethernet) must be in the range of 12.5 MHz to 120 MHz.

Ethernet (r_ether)

- If your Ethernet design has the Link status signal option available on the board, make sure you specify the LINK STATUS signal level coming from the PHY device and configure the appropriate signal level transition selection based on the PHY chip which is being used on the board.

- This is one of the important features which needs to be in the Ethernet design/application as it takes care of alerting the Ethernet controller when the link is up/down. Using this link status signal, Ethernet controller can send event/notification to the upper layer so that the data can be stopped or reinitiated at the different layers. Also, when the link comes up after going down, the controller can initiate transmission of the data from the buffer to the wire.

- Renesas RA Ethernet has the code support for this feature in the FSP. In the absence of this feature in your hardware design, users are required to use the polling method to check the status of the Link and thereby take appropriate action on the data transmission and its control.

Usage of this feature can be turned ON/OFF using Link Signal Change Flag.

Table 4. RA Ethernet Controller Debug

Configuration	Options	Description
ET0_LINKSTA Pin Status Flag	Fall -> Rise / Rise -> Fall	Specify the polarity of the link signal coming from the output by the PHY-LSI. When 0 is specified, link-up and link-down correspond respectively to the fall and rise of the LINKSTA signal. When 1 is specified, link-up and link-down correspond respectively to the rise and fall of the LINKSTA signal.
Link Signal Change Flag	Used / Unused	Use LINKSTA signal for detect link status changes: 0 = unused (use PHY-LSI status register) 1 = use (use LINKSTA signal)

- Channel selection for the Ethernet controller needs to be correct. By default, it is 0 and in the MCUs which have 2 Ethernet controllers, an appropriate selection needs to be made based on the hardware design.
- Valid MAC address selection is very important and needs to be unique to the devices on the network. Duplicate MAC address on the same network may result in undesired network behaviors. Renesas

provides a test MAC address by default, and users are required to change it to their registered MAC address in the production boards.

- By default, FSP uses the zero-copy mode option for the data copied to the upper layers. This greatly improves the performance as there is no duplication of the data. Enough buffer memory needs to be allocated to handle the incoming continuous data stream.
- The ETHERC supports flow control compliant with IEEE802.3x and can transmit and receive PAUSE frames. By using the flow control, the packet loss can be minimized in the congested networks. FSP provides the default built-in configuration in the code.
- When the Multicast mode is enabled, multicast packets are received and passed to the upper layers. When disabled, multicast frames are not received. This is an important property when developing applications which require reception of multicast packets. Also, by using the broadcast filter, the number of broadcast frames received can be limited with the configurable value.
- When the ETHERC operates in promiscuous mode, it receives all valid frames regardless of whether the address matches the destination address, broadcast address, and multicast bit settings. This mode is used for the sniffing of packets, debugging the network as it receives all the packets on the network. For normal application, this can be disabled to reduce the unnecessary processing.
- While using the ETHERC, it is good practice to optimize the size of the buffer memory required for transmitting and receiving Ethernet frames. FSP provides a way to allocate the TX and RX buffers required for the application. These can be selected by configuring the number of buffers (1 to 8) for both TX and RX and setting the desired size for each buffer. The default value is set to 1514 byte per buffer. Both values need to be modified to meet the application requirements.

5.2 Debugging Ethernet PHY using FSP

Some of the PHY specific configurations which need to be taken into consideration are given as follows. Refer to these sections for more details and depending on whether your board has these configurations.

Ethernet PHY (r_ether_phy)

- Channel selection to match the ETHERC channel being used in the design: by default, it is 0. If the MCU supports 2 Ethernet channels, the proper selections need to be made to suit the design and for the proper communication to the device.
- PHY-LSI Address: All PHYs support the serial MDIO management interface. The MDIO pin is a bidirectional shared serial bus. Up to 31 PHYs can share this bus. Typically, only one PHY is connected to the bus. The selection of a PHY address (with which the PHY device is configured to operate) out of the 31 addresses, is required for proper 2-way communication.
- PHY-LSI Reset Completion Timeout: configure the datasheet recommended timeout value for the PHY.
- Selection Interface from MAC to the PHY: proper selection of RMII or MII needs to be in place as per the design.
- PHY LSI Type: It is important for custom boards to have the proper PHY LSI type selection. For Renesas evaluation boards, the default selection is made based on the “Kit Component” selection.
- MII/RMII Register Access Wait-time: Specify the bit timing for MII/RMII register accesses during PHY initialization. This value should be adjusted experimentally based on the PHY-LSI used. This is a critical value and should be selected as recommended in the datasheet of the custom PHY.
- Flow Control: If the flow control option is required, it must be selected accordingly.
- Consider adding temporary debugger breakpoints after each PHY API call, then validate the return code for success or failure, which may indicate a problem.
- Instrument the PHY driver code, by adding debug messages during debugging.

MDIO

- Ensure that the MDIO bus connections between the MCU and the PHY device are properly connected. Verify the wiring, and solder joints for any physical damage or loose connections.
- Confirm that the Management Data Clock (MDC) signal is being generated and transmitted correctly by the MCU. Use an oscilloscope or logic analyzer to check the MDC signal for proper frequency, amplitude, and timing.
- Inspect Timing Requirements, review the timing specifications for MDIO transactions in the PHY's datasheet or user manual. Ensure that the MCU meets the timing requirements for MDIO communication, including setup and hold times, clock frequency, and data transmission rates.

- Verify that you can write and read from the PHY device with the PHY APIs provided by FSP. This can be verified using the protocol Analyzer (such as the Saleae logic analyzer) with a MDIO analyzer connecting the MDC and MDIO line (Reference **Figure 19** and **Figure 20**) .
- Monitor MDIO Traffic, use debugging tools or software utilities to monitor MDIO bus traffic between the MCU and the PHY device. Capture MDIO transactions, including read and write operations, to identify any communication errors or unexpected behavior.
- Check PHY Address: Ensure that the PHY device's address configured in the MCU matches the physical address of the PHY device on the MDIO bus. Verify that the PHY address is correctly set in the host device's configuration registers as per the HW design.
- Test read and write operations to PHY registers using the MDIO interface. Verify that the host device can successfully read from and write to the PHY's control and status registers. Check for any errors or inconsistencies in register access.
- Check PHY Status Registers: Monitor the status registers of the PHY device to check for any error conditions or abnormal behavior. Look for indications of link status, auto-negotiation status, and error flags that may provide insights into the issue. If auto-negotiation is enabled, verify that the MCU and the PHY device can successfully negotiate link parameters such as speed and duplex mode. Check for any auto-negotiation failures or inconsistencies.
- Consult the PHY's user manual to identify any unique configurations necessary for its operation. Special vendor-specific commands may be required to enable the PHY's functionality. Modify the code accordingly to transmit these vendor-specific commands to the PHY. Ensure that the code is adjusted to accommodate these specific commands, allowing the PHY to operate as intended.

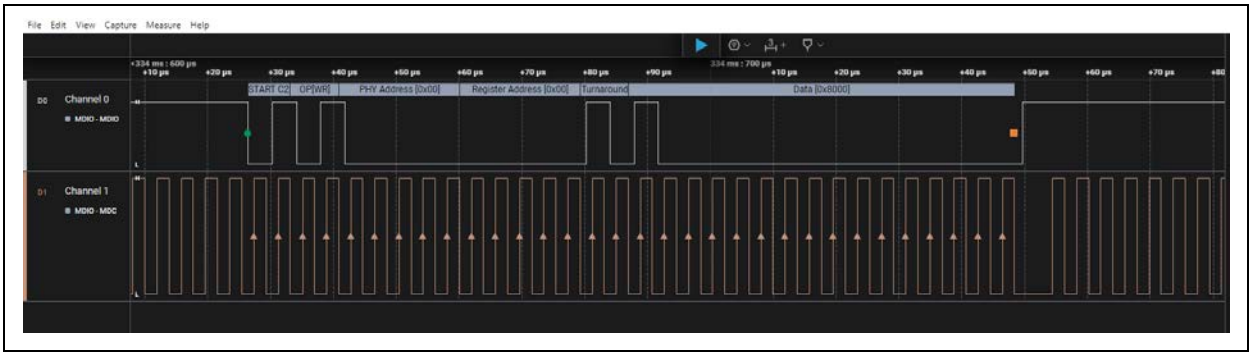

Figure 19. Snapshot of RA (EK-RA6M5 board) PHY Write using Salae Logic analyzer

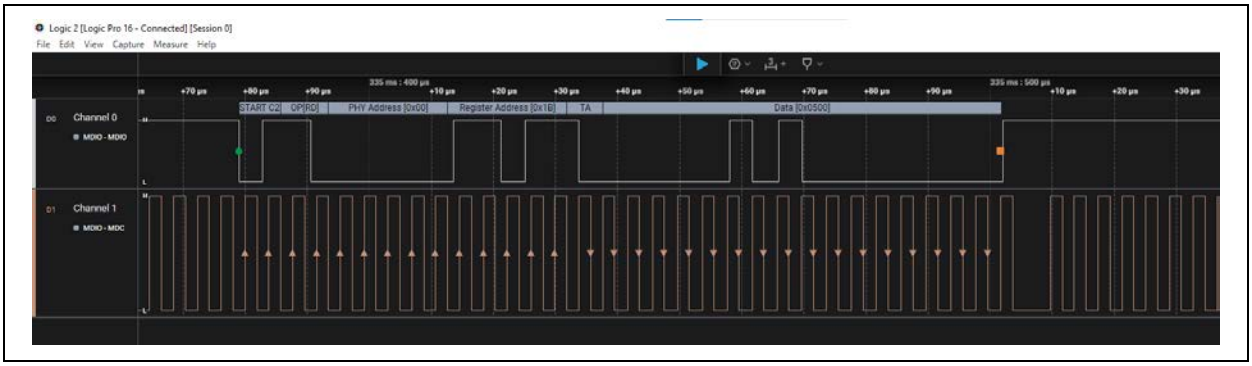

Figure 20. Snapshot of RA (EK-RA6M5 board) PHY Read using Salae Logic analyzer

By connecting the MDC and MDIO pins to Salae logic analyzer the MDIO Write and read operations can be captured. In the **Figure 19**, you can see the snapshot of PHY write to the PHY address 0x00 and Register Address 0x00 with the Data 0x8000 can be seen. Similarly, in **Figure 20**, the PHY read for the PHY address 0x00 and register address 0x1B with the data 0x0500 are shown.

5.3 Debugging RA Ethernet Designs

Some considerations for debugging RA Ethernet designs are as follows:

- Review board designs with reference to the schematics. Verify the usage of the correct component values (resistors, capacitors, and other passive components).
- Check for bad soldering (short or open), jumpers, damaged traces, improper component mounting, and so forth.
- Verify the VCC and GND for Ethernet PHY controller, also verify the Reset and Clock signals to the PHY.
- For prototype boards, it is best practice to evaluate the pins and signals before running any IP level traffic tests.
- During the initial development stages. It is important to ensure the TX/RX signals at the RJ-45 connector are compliant with (802.3 Physical Layer Compliance). This can be done by using a logic analyzer or oscilloscope.
- Identify the selection of RMII/MII modes in the hardware and verify that the software (FSP) is configured to use the correct mode.
- Basic network connectivity can be checked using the proper cable connected between the RA board and host. If the Link LED turns ON, that means the connectivity is present.
- Ethernet LEDs: Green and Amber. These LEDS indicate the status of the Ethernet at the Physical layer. If the Green LED is solid, it indicates the Link presence. Blinking of the Amber LED indicates network activity.
- Loopback tests can be performed to validate the proper HW connections.
- Perform Ethernet MAC local loopback to verify the functionality of the MAC using this internal loopback. Set the ILB bit (Internal Loopback Mode) of the ECMR (ETHERC Mode Register) to perform this internal loop.
- Perform MAC+PHY loopback to verify the functionality of the PHY using this internal loopback. Refer to the PHY User's manual for setting the loopback bit under Device Mode Control register.
- When Physical layer is up, the Ethernet connection can be debugged using TCP/IP and the Wireshark tools for data traffic. This can be performed by connecting the Ethernet port to a switch and connecting a PC with Wireshark tool to a switch and connecting uplink of the switch to a LAN/router.
- Basic Ping operation from the PC can be initiated on the board's IP address. Similarly, the ping utility can also be developed on the device side for debugging purposes.
- Run and analyze the Wireshark logs to review the ARP, and ICMP packets exchanged between the two devices. This will ensure the connectivity between the devices.
- Ping and trace route utility are a simple and best way to debug the connectivity. Ping tests with varying data sizes can be an effective way to test and measure the performance and reliability of the code.
- Packet Generator tools like Spirent Network analyzer (https://www.spirent.com/products/ethernet-ip-test-solutions) can also to be used to test the Ethernet. This can also be automated for stress testing with varying packet sizes. It also keeps a log of the number of packets sent, received, dropped and other errors.
- Without auto negotiation enabled, Ethernet mode and speed selection mismatch can also be an issue. For example, if the RA device is configured as half duplex or full duplex with 10/100 Mbps speed, but the other side does not use the same settings then there will be a problem during the communication.
- All the RA evaluation boards by default have an Auto Negotiation (AN) mode. Check that the AN bit is set after exchanging the link capability between the 2 devices. After the link negotiates the mode and speed, the AN bit will be set. This can be verified on the PHY side by monitoring the AN bit in the PHY registers.

6. FAQs

6.1 RA Ethernet FAQs

- Q: Does FSP support adding different PHYs to RA based Ethernet designs?
  - A: FSP supports adding new PHYs to RA based Ethernet designs. The FSP has built-in PHY code wrapper functions to interface with the new PHY device.
- Q: Does RA support adding 2 Ethernet channels and 2 different PHYs?
  - A: Some of the RA MCUs have 2 Ethernet channels and both can be used with different PHYs. FSP provides support for this.
- Q: Do I need to change the Ethernet driver code when a new PHY is used in the design?
  - A: There is no need to change the Ethernet driver on the FSP side. However, when adding the new PHY, the PHY drivers will need small modifications depending on the PHY device.
- Q: Can I configure the mode and speed of the Ethernet devices using FSP Configurator?
  - A: FSP configurator does not provide the option of configuring the Speed selection or Mode selection. It is by default in the auto negotiation mode.
- Q: Can I use the RA Ethernet with POE (Power over Ethernet) mode?
  - A: RA Ethernet can be used in POE mode.
- Q: Can I use the RA Ethernet along with FSP for PPPoE (PPP over Ethernet) ?
  - A: FSP does not support PPPoE. Adding support for PPPoE can be done in the software by using RA Ethernet as a PPPoE interface.
- Q: Does Renesas RA provide integrated PHYs for Ethernet designs?
  - A: RA Ethernet only provides an Ethernet controller with MAC support. PHY support needs to be added separately. Please refer to the RA Evaluation kit design files and datasheets for more details.
- Q: Does Renesas RA provide support for MII interfaces?
  - A: RA Ethernet provides support for MII/RMII and can be used with MII/RMII interface designs. However, users are recommended to thoroughly evaluate the MII interface.
- Q: Does Renesas RA provide support for Low Power modes while using the Ethernet controller?
  - A: Low Power modes are supported on Renesas RA MCUs while using the Ethernet controller. In addition, the Wake-on-LAN feature is also supported on RA MCUs. FSP has the support for the same.

6.2 Renesas Rulz

Refer to the following link for some of the answered questions from the Renesas Support teams:
https://community.renesas.com/search?q=Ethernet&group=250#serp=2

7. References

- RA Flexible Software Package Documentation: Introduction (renesas.github.io)https://www.renesas.com/us/en/document/apn/rx-family-ethernet-hardware-design-guide-rev101
- https://www.renesas.com/us/en/document/dst/1894-32-datasheet

8. Known Issues

This section talks about the known FSP and e²studio tool related issues. More details for any Ethernet related issues can be found at this link (https://github.com/renesas/fsp/issues).

9. Website and Support

Visit the following vanity URLs to learn about key elements of the RA family, download components and related documentation, and get support.

RA Product Information	renesas.com/ra
RA Product Support Forum	renesas.com/ra/forum
RA Flexible Software Package	renesas.com/FSP
Renesas Support	renesas.com/support

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Oct.06.22	—	Initial Release
1.01	Jun.24.24	—	Added MDIO support

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance 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 the 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.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is 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. Additionally, 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.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.