

## Introduction

Purpose of this document is to give a generic picture about the main Network management Interfaces used to drive physical devices.

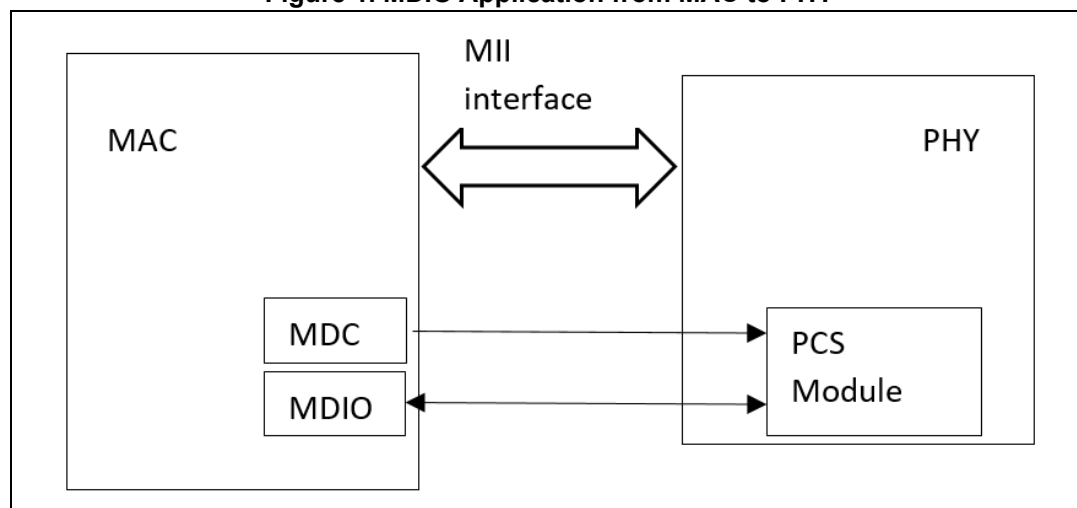
This technical note tries to cover the protocol internals, specification and hardware connection. It also provides some implementations as examples available for SPC5x automotive micro-controllers.

Media Access Controller (MAC) embedded into the SPC5x MCUs need to be configured and managed during normal usage, for example to handle connection link speed, to start auto-negotiation or fixing capabilities.

Usually a Media Access Controller (MAC) can be interfaced to standard physical transceiver or external switches. This is done by using the media independent interface (MII). In the [Section 2: Media Independent Interface](#) it will be detailed how this interface family is used to transfer data from/to Ethernet.

Media Management interfaces are also required to control and get the status from a stand-alone PHY transceiver or integrated PHYs inside a switch device. Some of them will be analyzed in this document in the [Section 1: MIIM, MII Management Interface](#).

**Figure 1. MDIO Application from MAC to PHY**



# Contents

<b>1</b>	<b>MIIM, MII Management Interface</b>	<b>5</b>
1.1	Serial Management Interface (SMI)	6
1.2	I2C and SPI for Switch Connection to an internal MAC	6
1.3	An example of basic PHY programming	7
<b>2</b>	<b>Media Independent Interface</b>	<b>9</b>
2.1	MII(10/100M) Interface	9
2.1.1	MII Lite mode	10
2.2	RMII(10/100M) Interface	10
2.3	GMII Interface	11
2.4	RGMII Interface	12
2.5	SGMII Interface	12
<b>3</b>	<b>SPC5x Media Independent Interface</b>	<b>14</b>
3.1	10/100 MII/RMII Transceiver	14
3.2	MII interface on SPC58NG Discovery board	15
3.3	RMII interface on SPC574Kxx Discovery board	16
3.4	RGMII interface on SPC58EHx/SPC58NHx reference boards	18
3.4.1	RGMII clock selection	19
<b>Appendix A</b>	<b>Further information</b>	<b>21</b>
A.1	Acronyms and abbreviations	21
A.2	Reference documents	21
	<b>Revision history</b>	<b>22</b>

List of tables

Table 1. MDIO frame format . . . . . 5

Table 2. MII PIN connections . . . . . 10

Table 3. Speed and clock selection for RGMII mode . . . . . 12

Table 4. Acronyms and abbreviations . . . . . 21

Table 5. Document revision history . . . . . 22



## List of figures

Figure 1.	MDIO Application from MAC to PHY . . . . .	1
Figure 2.	MDIO / MDC read/write operations . . . . .	6
Figure 3.	High level L2 switch connection . . . . .	7
Figure 4.	Control, status and ID PHY registers . . . . .	8
Figure 5.	MII connections . . . . .	9
Figure 6.	RMII connections . . . . .	11
Figure 7.	Example of MAC block in SGMII mode . . . . .	13
Figure 8.	DP83848C pins . . . . .	14
Figure 9.	PHY connection on a SPC57xxMB . . . . .	15
Figure 10.	SPC58N discovery board . . . . .	15
Figure 11.	Example of configured PINs . . . . .	16
Figure 12.	SPC57K-Discovery board . . . . .	17
Figure 13.	SPC574Kxx clocking . . . . .	17
Figure 14.	SPC574Kxx Ethernet PIN configurations . . . . .	18
Figure 15.	SPC58EHx/ SPC58NHx Reference board . . . . .	18
Figure 16.	SPC58EHx example of RGMII pin configuration . . . . .	19
Figure 17.	SPC58EHx RGMII clock schema . . . . .	19

# 1 MIIM, MII Management Interface

The MIIM should not be confused with the MII interface which is used to exchange data with a PHY device or an embedded switch.

The MIIM is also known as **MDIO/MDC Interface**.

The **Management Data Input/output (MDIO)** is a serial bus defined for the Ethernet family of IEEE 802.3 standards for the Media Independent Interface.

MDIO was defined in Clause 22 of IEEE 802.3; a MDIO bus is able to access up to 32 registers in 32 different PHY devices. Through the MDIO is possible, in a glance, to read and write to the PHY internal registers. These registers provide status and control information such as: link status, speed and duplex mode, low power conditions and restarting auto-negotiation process.

In a realistic application it is mandatory to get the status of the physical layer from upper layer protocols; MAC (Medium Address Control Layer) to tune parameters at run-time decisions, e.g. like clock settings or Software fix-up routines. In the same way, user can decide to change or invoking dedicated fix-up routines.

From the hardware point of view, the MIIM consists of the following two signals:

- MDC clock
- MDIO data

The MDC is the Management Data clock that is sourced from the Ethernet part. It is 25MHz clock.

The MDIO, Management Data Input/output is a bidirectional open-drain pin with an appropriate pull-up resistor (e.g. 4.7 Ohm).<sup>(a)</sup>

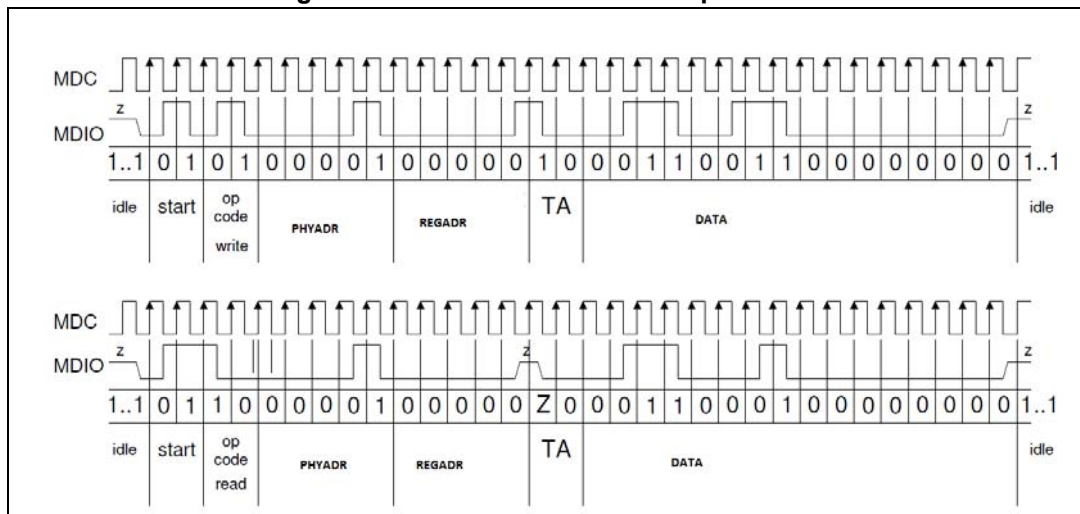
The following table shows the basic frame format that consists of a header plus 16-bit of data that can be driven by Station Management Entity (STA) and MDIO Manageable Devices (MMD).

**Table 1. MDIO frame format**

Clause 22 MDIO frame format		
ST	2 bits	Start of Frame (01 for Clause 22)
OP	2 bits	OP Code 01: Write, 10: read
PHYADR	5 bits	0 – 31 PHY Address
REGADR	5 bits	Register Address
TA	2 bits	Turnaround time to change bus ownership
16-bit of DATA: Write operation is driven by STA and Read is driven by MMD		

a. Resistor selection can also depend on the transceiver so refer to vendor's recommendation.

Figure 2. MDIO / MDC read/write operations



The frame format described above is compliant with the Clause 22 that has some limitations, for example, it does not have a low voltage option. The clause 45 has been introduced to the standard to extend features and register set.

Although the main registers are standardized, vendors usually add a small set of registers to setup extra features; for example some PHY device extends the low power capability. Anyway, this extra register must not be confused with the Clause 45.

For example, reference boards embed the DP83848C Single Port 10/100 Mb/s Ethernet Physical Layer Transceiver from National Semiconductor. This device is basically compliant with clause 22. The DP83848C also implements the Energy Detect Mode in order to stay in low-power if there is no activity on the cable.

Using the Clause 45 it is possible to access 65.536 registers available for supporting extended features. Some of these allow Wake-Up from PHY and EEE (Energy Ethernet Efficient defined in the 803.az).

This document doesn't detail the Clause 45 but, in summary, inside the frame, instead of *REGADR* there is the *DEVTYPE* (5 bits) used to specify the targeted device type. The address space is extended from 5 to 16 bits so the latest 16 bits (only used as data in the Clause 22) are for Address or Data. A new protocol is used to select extended registers and read/write data. Then it is possible to access basic registers and be compatible with Clause 22.

## 1.1 Serial Management Interface (SMI)

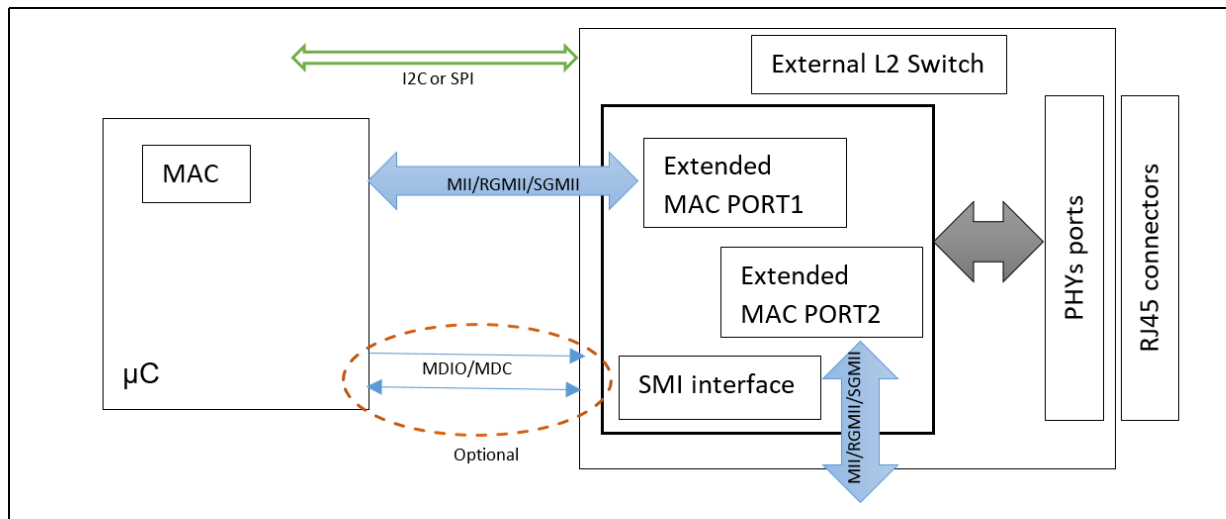
This is used to read and write registers in the devices and it uses the same MDIO/MDC pins but with small differences on opcode and address fields. On some MACs that do not support SMI this is software emulated (bit-banging).

## 1.2 I<sup>2</sup>C and SPI for Switch Connection to an internal MAC

External switch can be connected to the main MAC embedded in the microcontroller. To dialog with the switch, different interfaces can be used. Switch programming can be done by I<sup>2</sup>C or SPI depending on the switch that has been integrated on the PCB.

The application can configure the internal registers for basic and complex user-case. Software should configure the MII interface, perform reset, and configure link property and setup VLAN. Via I2C or SPI the switch can optionally get PHY status although some configurations also provide SMI or MIIC pins.

**Figure 3. High level L2 switch connection**



### 1.3 An example of basic PHY programming

Figure 4 shows some of the main registers of a 10/100 PHY transceiver; in a basic PHY programming, after resetting the device, the Control register is programmed according to the application needs. For example, auto-negotiation can be restarted or a fixed link can be setup. The bit 15 in the Control register must be 0 and bit 11 can be used to enter in power-down state. The PHY ID registers are used to get the device ID. Often at MAC layer, after resetting the PHY, the ID is read to address the desired device. An Ethernet driver can fail if there is a broken ID (usually 0xffff means that the PHY is not properly reset or missing pull-down resistor generates issues on the bus).

Some Ethernet devices can stuck in case of the PHY clock configuration is not properly set.

For sure the status register is used to understand if the auto-negotiation process is completed, or if the link is up etc. Some devices implement an interrupt service to get the status (usually extra registers are provided to enable/disable IRQ and report the state). Often it is also used polling mechanism. In that case, a software timer routine is used to periodically (e.g. 1s) read the PHY status and report this to the upper layer.

Figure 4. Control, status and ID PHY registers

Control Register															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	Loopback	Speed Select	A/N Enable	Power Down	Isolate	Restart A/N	Duplex Mode	Collision Test	Reserved						

Status Register															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
100Base-T4	100Base-TX Full Duplex	100Base-TX Half Duplex	10Base-T Full Duplex	10Base-T Half Duplex	Reserved					A/N Complete	Remote Fault	A/N Ability	Link Status	Jabber Detect	Extended Capability

PHY ID 1 Register															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHY ID Number (Bits 3-18 of the Organizationally Unique Identifier - OUI)															

PHY ID 2 Register															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHY ID Number (Bits 19-24 of the Organizationally Unique Identifier - OUI)						Manufacturer Model Number						Manufacturer Revision Number			



## 2 Media Independent Interface

The media-independent interface (MII) is used to connect an Ethernet block to a PHY.

The next sections provide an overview about all the different MII modes:

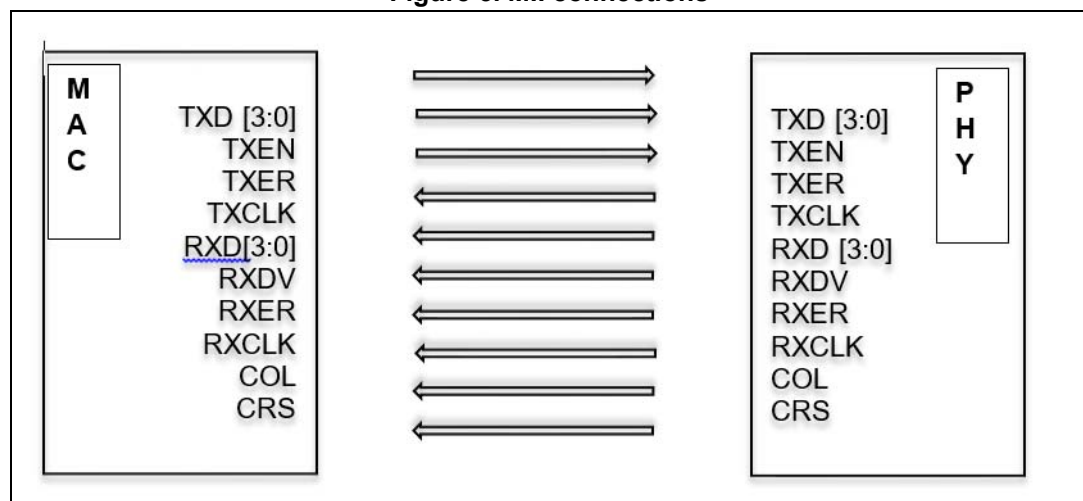
- **MII** - Media independent interface
- **RMII** - Reduced media independent interface
- **GMII** - gigabit media independent interface
- **RGMII** - Reduced gigabit media independent interface
- **SGMII** – Serial gigabit media independent interface

### 2.1 MII(10/100M) Interface

In MII mode there are 16 signals as shown in the picture below plus two other ones for MDIO and MDC. In this mode, both TXCLK and RXCLK provided by PHY. Clock rate is 2.5 MHz for 10Mbps and 25MHz for 100Mbps. Clock can be provided to the PHY by either an internal XOSC or external clock source.

The main limit of this mode consists of the high number of signals and just MDIO/MDC can be shared among multiple PHYs; so this mode is not preferred for multiport configurations.

Figure 5. MII connections



From software point of view, GPIO lines need to be setup following the RM manual for the specific microcontroller. For each port, direction and alternate functions have to be programmed. The table below shows the port direction for SPC58NN84xx  $\mu$ C to connect the Ethernet\_0 device to the PHY in MII mode.

Table 2. MII PIN connections

Port	Function	Description	Direction
PC[2]	ETH_MDIO	Ethernet 0 Management Data	i
PC[3]	ETH_MDC	Ethernet 0 Management Data Clock	o
PC[11]	ETH_TX_ER	Ethernet 0 Transmit Error	o
PC[11]	ETH_CRS	Ethernet 0 Carrier Sense	i
PC[12]	ETH_RDATA0	Ethernet 0 Receive Data 0	i
PC[13]	ETH_RDATA1	Ethernet 0 Receive Data 1	i
PC[14]	ETH_TX_EN	Ethernet 0 Transmit Enable	o
PC[15]	ETH_TDATA0	Ethernet 0 Transmit Data 0	o
PE[12]	ETH_TDATA1	Ethernet 0 Transmit Data 1	o
PK[14]	ETH_TX_CLK_0	Ethernet 0 Transmit Clock 0	i
PK[15]	ETH_RDATA3	Ethernet 0 Receive Data 3	i
PM[0]	ETH_RX_DV	Ethernet 0 Receive Data Valid	i
PM[1]	ETH_RDATA2	Ethernet 0 Receive Data 2	i
PM[2]	ETH_COL	Ethernet 0 Collision	i
PM[3]	ETH_RX_ER	Ethernet 0 Receive Error	i
PM[4]	ETH_TDATA2	Ethernet 0 Transmit Data 2	o
PM[5]	ETH_TDATA3	Ethernet 0 Transmit Data 3	o
PM[8]	ETH_RX_CLK_0	Ethernet 0 Receive Clock 0	i

### 2.1.1 MII Lite mode

Some devices support MII-Lite mode that has not to be confused with RMII one. MII Lite is used to save pins compared with the standard MII mode keeping the clock to 25MHz.

In MII-Lite mode the following signals are omitted: TXER, RXER, COL, and CRS.

Malformed frames could be detected by looking at the internal descriptor status at MAC layer. Missing the COL, Half duplex could not be supported.

## 2.2 RMII(10/100M) Interface

This mode reduces the number of signals to connect a PHY to a MAC. There is a single 50MHz clock (REF\_CLK) for 10/100Mbps that is an input to the PHY.

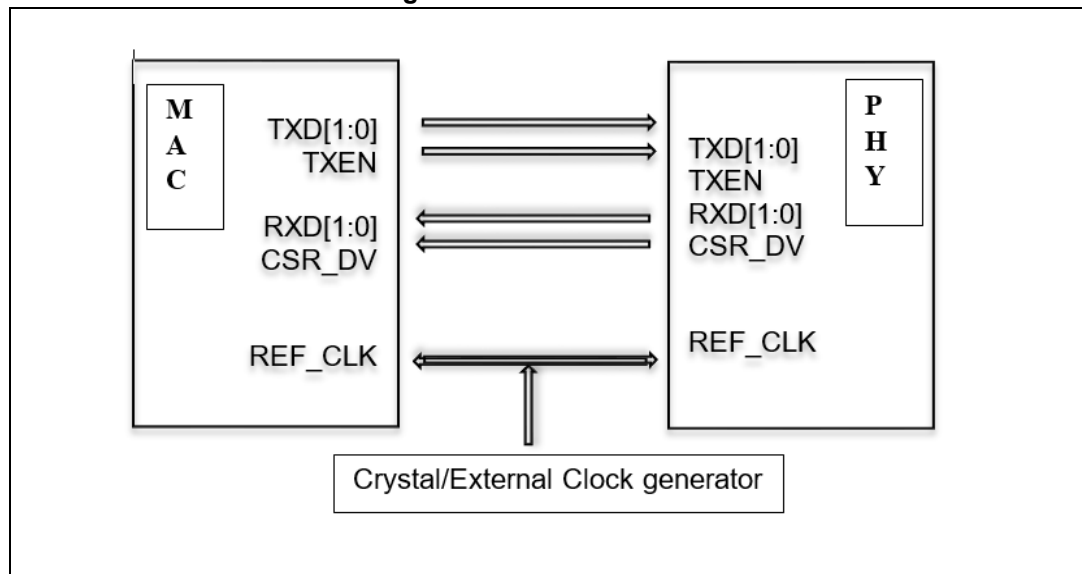
At MAC side, the RX clock is by-passed and the TX\_CLK is connected to the 50MHz.

The REF\_CLK can be provided from either a crystal/external clock or internal PLL (if supported). The latter case usually needs some extra software configuration to setup the clock on the related pins.

In RMII mode, the RXDV and CSR are multiplexed and COL removed.

Other signals are removed e.g. TXD[2:3], RXD[2:3] due to the doubled frequency, from 25 to 50MHz. Last but not least, that 50 MHz is for 10Mb/s and for 100Mb/s as well.

Figure 6. RMII connections



## 2.3 GMII Interface

Gigabit media-independent Interface (GMII) is an interface between the Media Access Control (MAC) device and the physical layer (PHY). The maximum speed is 1 Gbps using clock rate at 125MHz. GMII is compatible with MII specifications.

Below the signals used in this mode.

Data signals

- TXD[7:0]
- RXD[7:0]

Control signals

- TXER
- RXER
- COL
- TXEN
- CRS
- RXDV

Plus

- MDIO/MDC

Clock signals

- RXCLK: receive data clock and is provided by the PHY to the MAC
- TXCLK: transmit clock for MII mode and is provided by the PHY
- GTXCLK: 125MHZ for supporting giga speed and is supplied to the PHY.

PHY has its own external crystal for 25MHz.

*Note: Actually the GMII is not supported by STM Automotive micro-controllers but It is mentioned as a basis for other supported modes in order to see the differences and usages.*

## 2.4 RGMII Interface

Reduced gigabit media-independent interface (RGMII) is one of the most preferred interfaces used for giga configurations, it uses half signals compared to GMII as shown below:

Data signals:

- TXD[3:0]
- RXD[3:0]

Control signals

- RX\_CTL: RXDV and RXER are multiplexed.
- TX\_CTL: TXEN and TXER are multiplexed

Clock signals

- TXCLK
- RXCLK

The TXCLK is provided by the MAC to the PHY and this must be set according to the speed:

**Table 3. Speed and clock selection for RGMII mode**

Mbps	MHz
10	2.5
100	25
1000	125 (double edge)

On some configurations, where the TXCLK is internally provided by PLL, a PHY can communicate the speed to the Ethernet driver that, at run-time, can scale the TXCLK according to the speed negotiated.

RXCLK is managed by the PHY and configured as input in the MAC.

RGMII data is sampled on both edges of the clock so, considering the PCB path delays, some devices provide extra delay (typically 2ns) to keep the sync. Some GiPHY has extra vendor registers to add the delay with more granularity.

*Note:* On STMicroelectronics Automotive Microcontroller the adopted standard is: RGMII v2.6 from HP/Marvell.

## 2.5 SGMII Interface

This chapter just introduces the SGMII that is actually not supported on SPC5x families.

Serial gigabit media-independent interface serializes GMII interface over a single line operating at 1.25Gbps in 1x mode and 3.125Gbps in 2.5x mode.

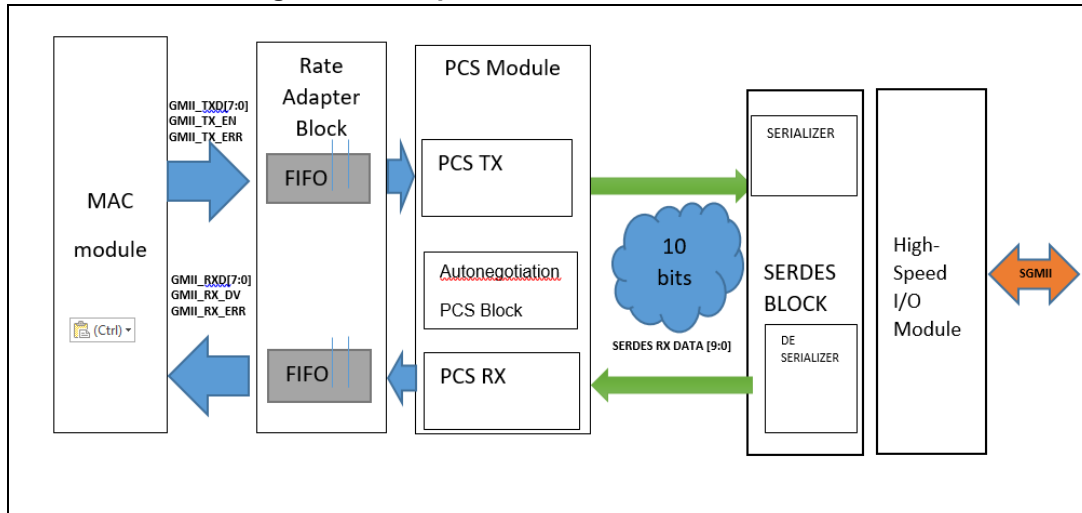
Gigabit operation uses GMII and 10/100 Megabit operation uses MII.

The 8B/10B encoding is used on transmit/receive data for each lane.

The reduced pin count comes at the cost of higher power consumption, mainly because the SGMII interface maintains a constant clock rate regardless of the operating speed of the MAC.

In the SGMII standard, the specification implementations show the MAC and PHY functional blocks and on PCS layer, SerDes blocks are inside the interface.

**Figure 7. Example of MAC block in SGMII mode**



### 3 SPC5x Media Independent Interface

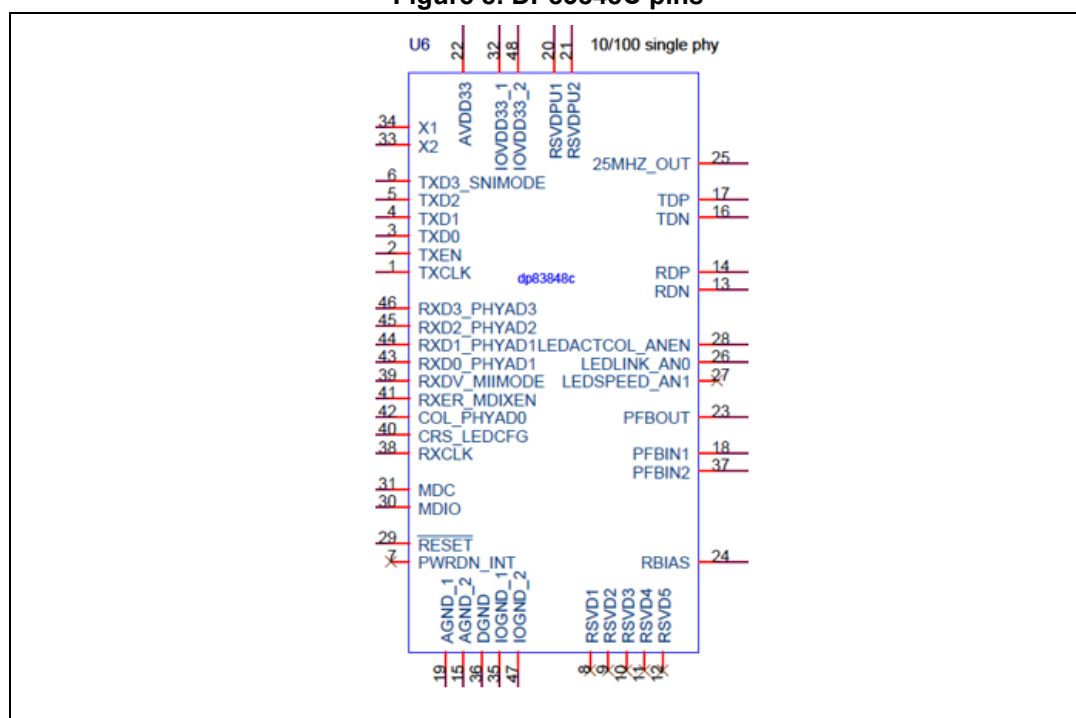
The following paragraphs show some MII connections adopted on different SPC5x reference boards.

#### 3.1 10/100 MII/RMII Transceiver

Reference and development boards integrated the DP83848C device, that is a standard transceiver that supports both MII and RMII modes.

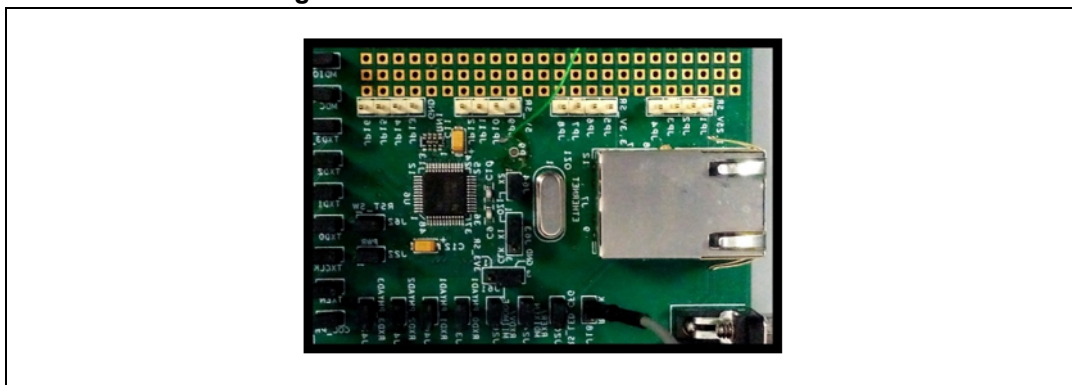
All PHY pins on the board are connected to PIO pin ports that have to be setup according to the platform.

Figure 8. DP83848C pins



The PHY device is then connected to a magnetic and RJ45 plug. PCD has to follow PHY device application note for integrating components to avoid malfunctions on final product.

Figure 9. PHY connection on a SPC57xxMB



### 3.2 MII interface on SPC58NG Discovery board

The SPC58NG discovery board is designed to address car body applications and Ethernet car networking.

Both instances of the Ethernet controllers are wired to the DP83848C transceivers in MII mode. The 25MHz clocks come from on board oscillator.

While programming the SPC58Gx MCU, the Ethernet interface depends on the settings of the SIUL2\_SCR0 (in the SIUL2), in case of MII, SIUL2\_SCR0[Ethernet\_MODE] = 1, must be set.

Figure 10. SPC58N discovery board

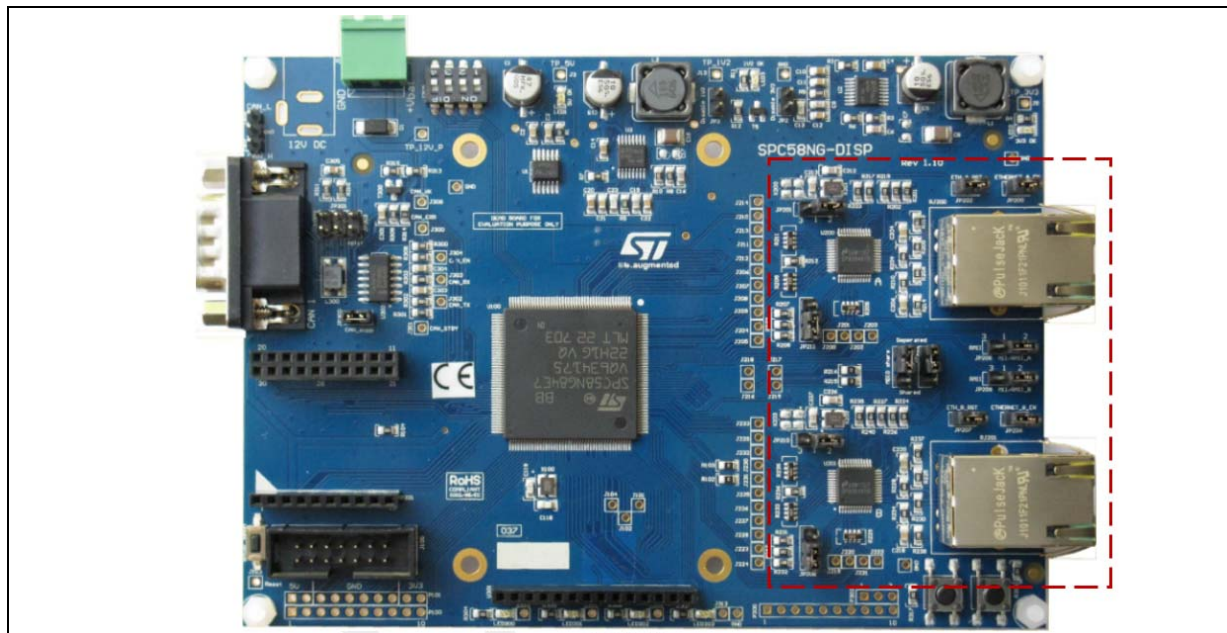


Figure 11 shows the list of configured signals for the Ethernet0 where the MII interface is configured.

In fact, the TX\_CLK and RX\_CLK are in INPUT so provided by the PHY transceiver as expected for this mode. The Output Edge Rate Control for TX\_DATA signals should be VERY\_STRONG so SIUL2\_MSCR\_Ion[OERC] = 0x2.

MDIO pin is configured as IN/OUT as required by specifications.

**Figure 11. Example of configured PINs**

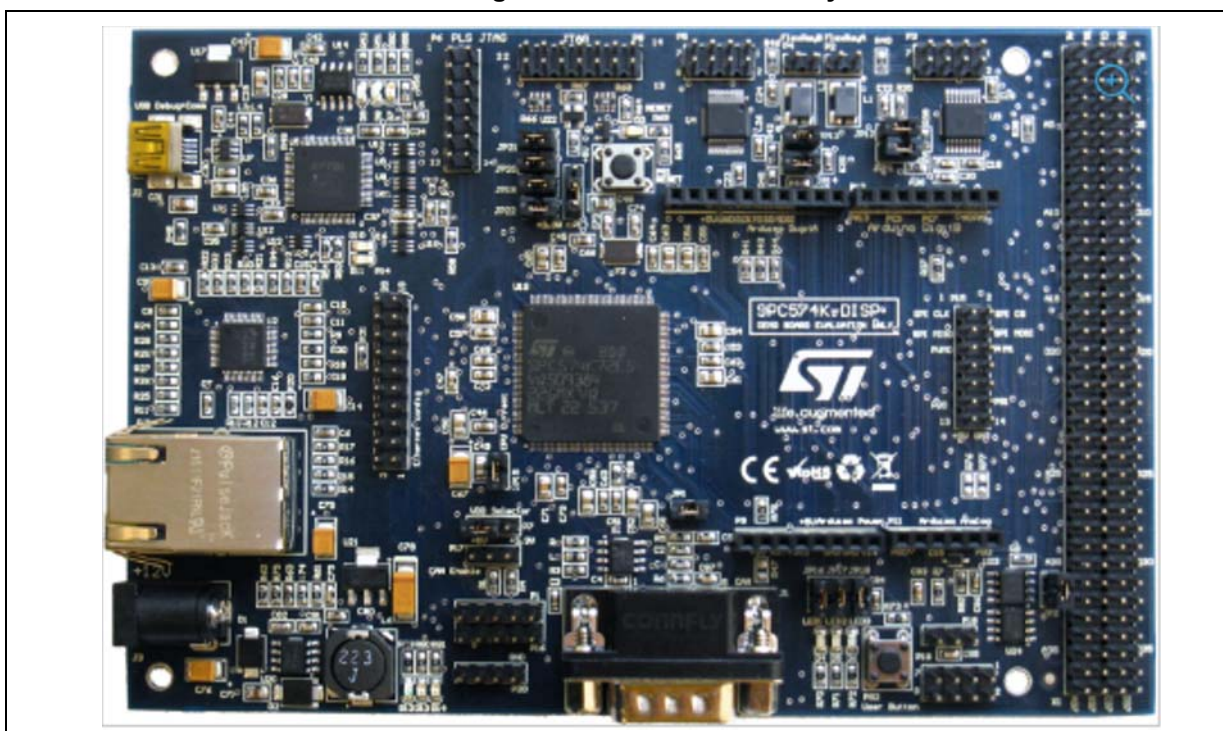
#	Identifier	Port	Bit	Pin Mode	Latched State	MSCR Index	SSS	MSCR Index	SSS
1	PIN_MDIO_0	PORT_C	0	MODE_IO	LOW	32	13	924	8
2	PIN_RDATA1_0	PORT_C	10	MODE_INPUT	LOW	42	0	928	4
3	PIN_RDATA0_0	PORT_C	11	MODE_INPUT	LOW	43	0	927	3
4	PIN_TDATA3_0	PORT_C	12	MODE_OUTPUT	LOW	44	14	512	0
5	PIN_TX_EN_0	PORT_C	13	MODE_OUTPUT	LOW	45	16	512	0
6	PIN_TX_CLK_0	PORT_C	14	MODE_INPUT	LOW	46	0	923	3
7	PIN_TDATA0_0	PORT_C	15	MODE_OUTPUT	LOW	47	1	512	0
8	PIN_COL_0	PORT_C	5	MODE_INPUT	LOW	37	0	931	2
9	PIN_TDATA2_0	PORT_D	0	MODE_OUTPUT	LOW	48	5	512	0
10	PIN_CRS_0	PORT_D	14	MODE_INPUT	LOW	62	0	926	3
11	PIN_RX_ER_0	PORT_D	15	MODE_INPUT	LOW	63	0	932	3
12	PIN_RX_CLK_0	PORT_D	3	MODE_INPUT	LOW	51	0	922	1
13	PIN_MDC_0	PORT_E	0	MODE_OUTPUT	LOW	64	13	512	0
14	PIN_TDATA1_0	PORT_E	12	MODE_OUTPUT	LOW	76	1	512	0
15	PIN_RX_DV_0	PORT_E	4	MODE_INPUT	LOW	68	0	925	5
16	PIN_RDATA2_0	PORT_F	4	MODE_INPUT	LOW	84	0	929	2
17	PIN_RDATA3_0	PORT_F	5	MODE_INPUT	LOW	85	0	930	2

### 3.3 RMII interface on SPC574Kxx Discovery board

The SPC57K-Discovery is designed to address SPC57K line Power Architecture® Microcontrollers with full access to several peripherals such as the Fast Ethernet Controller interfaced in RMII mode.



Figure 12. SPC57K-Discovery board



In RMII mode the REF\_CLK must be 50MHz and it is provided by programming the AUX Clock 10 Selector:

Figure 13. SPC574Kxx clocking

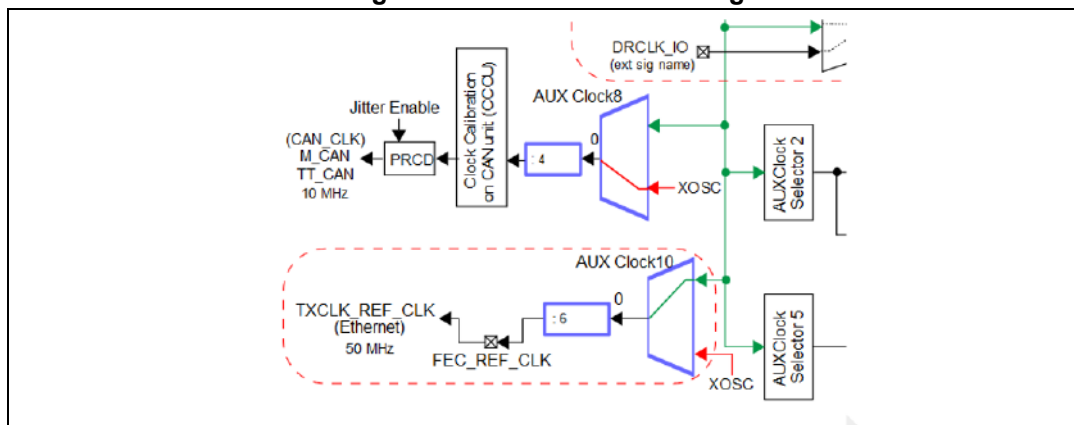
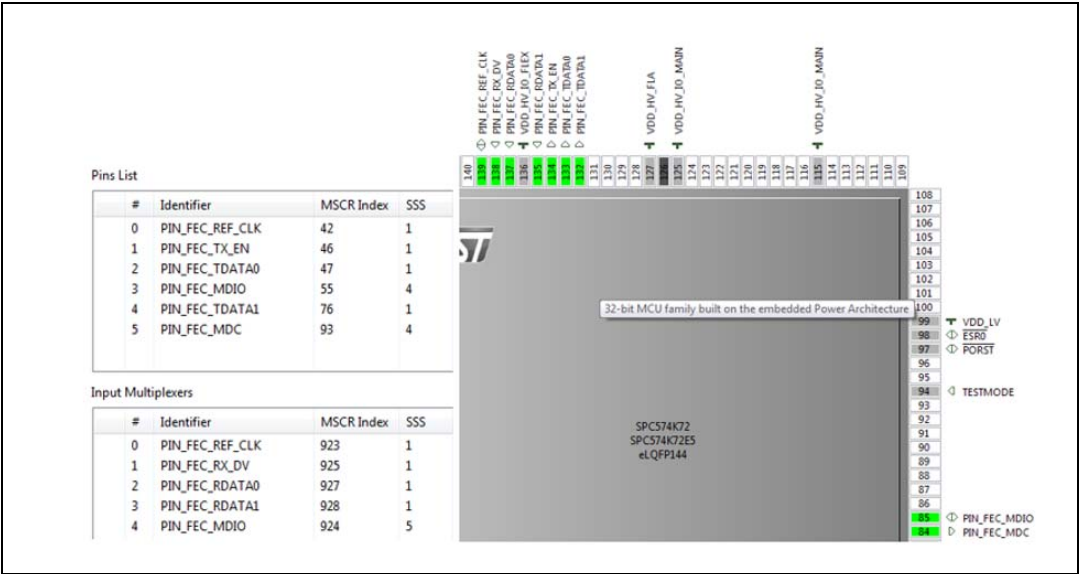


Figure 14 shows the PIN mapping used for this interface. Reduced signal numbers are used for DATA lines, the REF\_CLK is an INPUT signal and others are multiplexed as required by this standard.

Figure 14. SPC574Kxx Ethernet PIN configurations



### 3.4 RGMII interface on SPC58EHx/SPC58NHx reference boards

The available reference boards that support the 32-bit SPC58H line of STMicroelectronics' automotive microcontrollers, also provide a 1GB Ethernet\_1 interface with RJ45.

A secondary 10/100 Mbps Ethernet controller is also available. This can be configured for example in MII mode.

By default, the PHY transceiver on the series of reference boards for the GiGa Ethernet just supports the RGMII mode.

Figure 15. SPC58EHx/ SPC58NHx Reference board



To select the RGMII mode, during the MCU configuration, the SIUL2\_SCR0[ETH1\_REGPHY\_IF\_SELECT] must be set to 0x1.

Figure 16 shows an example of the PIN configuration used for RGMII mode. Note that the SIUL2\_MSCR[OERC] should be ULTRA STRONG for the OUTPUT signals, e.g. TXDATA and TX\_CLK.

**Figure 16. SPC58EHx example of RGMII pin configuration**

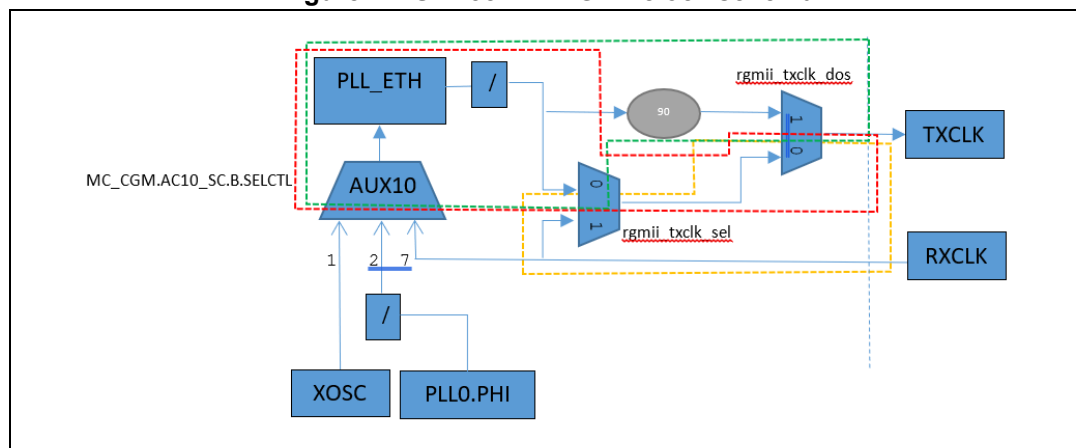
#	Identifier	Port	Bit	Pin Mode	Latched State	MSCR Index	SSS	MSCR Index	SSS
19	PIN_ETH1_MDC	PORT_Q	15	MODE_OUTPUT	LOW	271	1	512	0
20	PIN_ETH1_MDIO	PORT_Q	14	MODE_IO	LOW	270	1	971	1
21	PIN_ETH1_RX_DATA0	PORT_S	0	MODE_INPUT	LOW	288	0	974	3
22	PIN_ETH1_RX_DATA1	PORT_O	2	MODE_INPUT	LOW	226	0	975	3
23	PIN_ETH1_RX_DATA2	PORT_S	3	MODE_INPUT	LOW	291	0	976	3
24	PIN_ETH1_RX_DATA3	PORT_S	5	MODE_INPUT	LOW	293	0	977	3
25	PIN_ETH1_RX_CLK	PORT_R	14	MODE_INPUT	LOW	286	0	968	2
26	PIN_ETH1_RX_DV	PORT_O	1	MODE_INPUT	LOW	225	0	972	2
27	PIN_ETH1_PHY_CLK_125	PORT_R	4	MODE_INPUT	LOW	276	0	980	1
28	PIN_ETH1_TX_DATA0	PORT_R	12	MODE_OUTPUT	LOW	284	2	512	0
29	PIN_ETH1_TX_DATA1	PORT_R	10	MODE_OUTPUT	LOW	282	2	512	0
30	PIN_ETH1_TX_DATA2	PORT_R	8	MODE_OUTPUT	LOW	280	2	512	0
31	PIN_ETH1_TX_DATA3	PORT_R	6	MODE_OUTPUT	LOW	278	2	512	0
32	PIN_ETH1_TX_CLK	PORT_R	0	MODE_OUTPUT	LOW	272	4	512	0
33	PIN_ETH1_TX_EN	PORT_R	2	MODE_OUTPUT	LOW	274	1	512	0

### 3.4.1 RGMII clock selection

For this micro-controller, the TXCLK can be selected by using either an internal PLL\_ETH clock or the receive clock (that is routed from the PHY transceiver).

The Figure 17 shows the clock schema for the RGMII clock selections.

**Figure 17. SPC58EHx RGMII clock schema**



The TXCLK can be selected with:

- Delay on Source (DoS)<sup>(b)</sup>
  - rgmii\_txclk\_dos = 1
  - rgmii\_txclk\_sel = 0
- Delay on Destination (DoD)
  - Selecting the RXCLK as TXCLK (both for MAC / PHY)<sup>(c)</sup>
    - rgmii\_txclk\_dos=0
    - rgmii\_txclk\_sel=1
  - Select the ETH\_PLL divided output as TXCLK<sup>(d)</sup>
    - rgmii\_txclk\_dos=0
    - rgmii\_txclk\_sel=0

The DoS selection is the advised configuration.

*Note: In some circumstances, w/o DoS selection it could be needed to provide the external delay timing (to adjust and correct TX and RX timing paths) from PHY layer.*

*In fact, many PHY transceivers offer this kind of feature by programming internal extended registers.*

When select the PLL\_ETH as reference clock for the TXCLK, the Aux10 must be programmed to configure the reference clock for this source: there are three possibilities:

- XOSC
- A divided version of PLL0
- A divided version of RXCLK.

---

b. Refer to green box in the [Figure 17](#).

c. Refer to yellow box in the [Figure 17](#).

d. Refer to Red box in the [Figure 17](#).

## Appendix A Further information

### A.1 Acronyms and abbreviations

Table 4. Acronyms and abbreviations

Terms	Meaning
MII	Media independent interface
RMII	Reduced Media independent interface
GMII	GiGa Media independent interface
RGMII	Reduced GiGa Media independent interface
SGMII	Serial gigabit media independent interface
STA	Station Management Entity
MMD	MDIO Manageable Devices
PCS	Physical Coding SubLayer
SerDes	Serializer/Deserializer
QoS	Quality Of Service
LVDS	low-voltage differential signaling

### A.2 Reference documents

- IEEE 802.3u standard
- IEEE 802.az standard
- IEEE1596.3-1996 standard
- National Semiconductor DP83848C PHYTER® - Commercial Temperature Single Port 10/100 Mb/s Ethernet Physical Layer Transceiver datasheet
- STMicroelectronics SPC57xxMB schematics
- Cisco System Serial-GMII Specification Revision 1.7

## Revision history

**Table 5. Document revision history**

Date	Revision	Changes
27-Mar-2020	1	Initial release.

**IMPORTANT NOTICE – PLEASE READ CAREFULLY**

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to [www.st.com/trademarks](http://www.st.com/trademarks). All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2020 STMicroelectronics – All rights reserved