iEthernet W5500
Datasheet

Version 0.58e

This is a preliminary document release. All specifications are subject to change without notice.

http://www.wiznet.co.kr

W5500

The W5500 chip is a Hardwired TCP/IP embedded Ethernet controller that enables easier internet connection for embedded systems using SPI (Serial Peripheral Interface). W5500 suits best for users who need Internet connectivity for application that uses a single chip to implement TCP/IP Stack, 10/100 Ethernet MAC and PHY. Hardwired TCP/IP stack supports TCP, UDP, IPv4, ICMP, ARP, IGMP, and PPPoE, which has been proven in various applications for many years. W5500 uses a 32Kbytes internal buffer as its data communication memory. By using W5500, users can implement the Ethernet application they need by using a simple socket program instead of handling a complex Ethernet Controller. Users can use 8 independent hardware sockets simultaneously.

SPI (Serial Peripheral Interface) is provided for easy integration with the external MCU. The W5500 SPI supports 80 MHz speed and new efficient SPI protocol, so user can implement high speed network communication. In order to reduce power consumption of the system, W5500 provides WOL (Wake on LAN) and power down mode.

Features

- Support Hardwired TCP/IP Protocols : TCP, UDP, ICMP, IPv4, ARP, IGMP, PPPoE
- Support 8 independent sockets simultaneously
- Support Power down mode
- Support Wake on LAN over UDP
- Support High Speed Serial Peripheral Interface(SPI MODE 0, 3)
- Internal 32Kbytes Memory for TX/RX Buffers
- 10BaseT/100BaseTX Ethernet PHY embedded
- Support Auto Negotiation (Full and half duplex, 10 and 100-based)
- Not support IP Fragmentation
- 3.3V operation with 5V I/O signal tolerance
- Lead-Free Package
- LED outputs (Full/Half duplex, Link, Speed, Active)
- 48 Pin LQFP Lead-Free Package (7x7mm, 0.5mm pitch)

Target Applications

W5500 is suitable for Embedded application, such as:

- Home Network Devices: Set-Top Boxes, PVRs, Digital Media Adapters
- Serial-to-Ethernet: Access Controls, LED displays, Wireless AP relays, etc.
- Parallel-to-Ethernet: POS / Mini Printers, Copiers
- USB-to-Ethernet: Storage Devices, Network Printers
- GPIO-to-Ethernet: Home Network Sensors
- Security Systems: DVRs, Network Cameras, Kiosks
- Factory and Building Automations
- Medical Monitoring Equipments
- Embedded Servers

Block Diagram

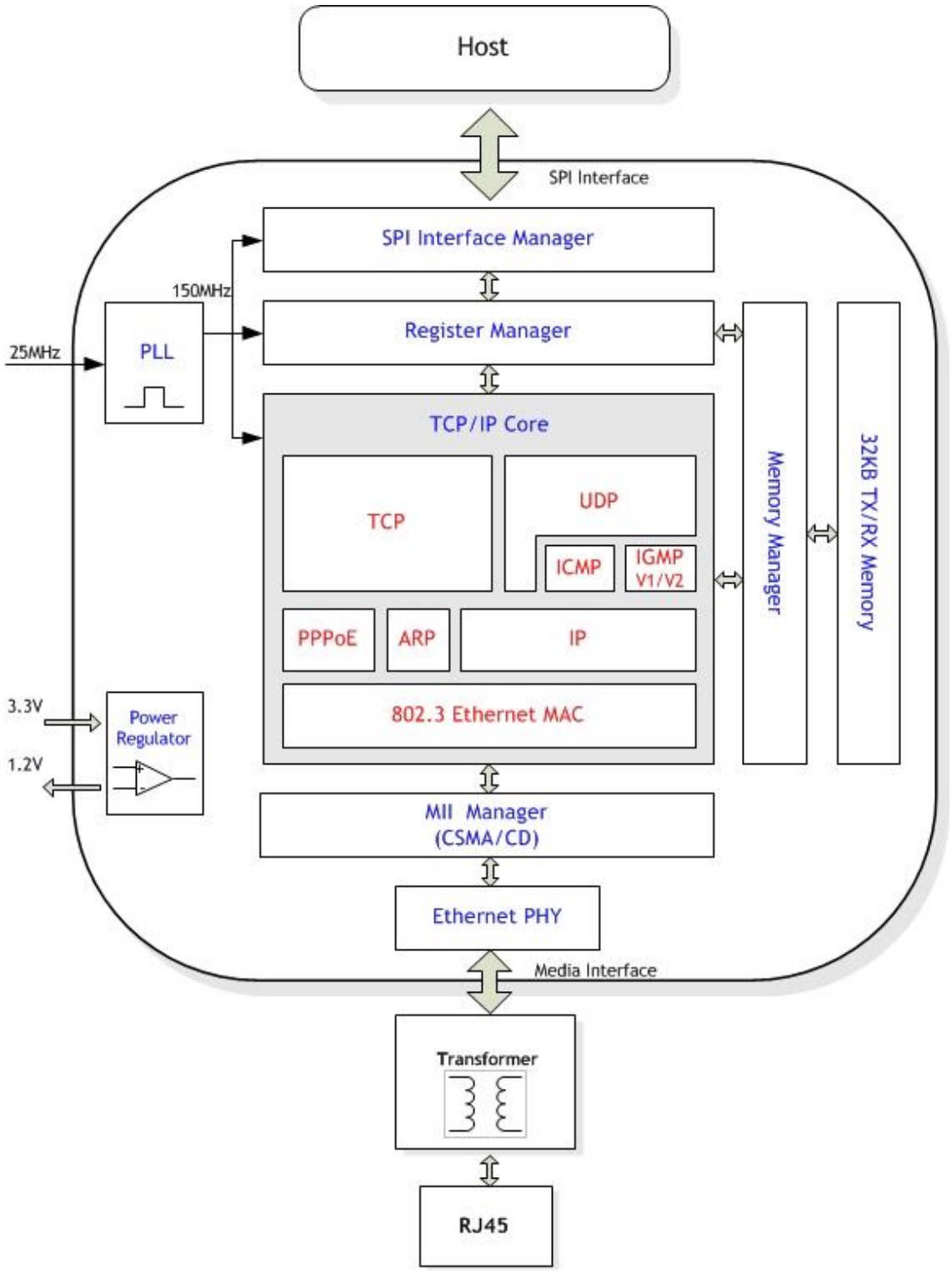


Table of contents

1 Pin Assignment

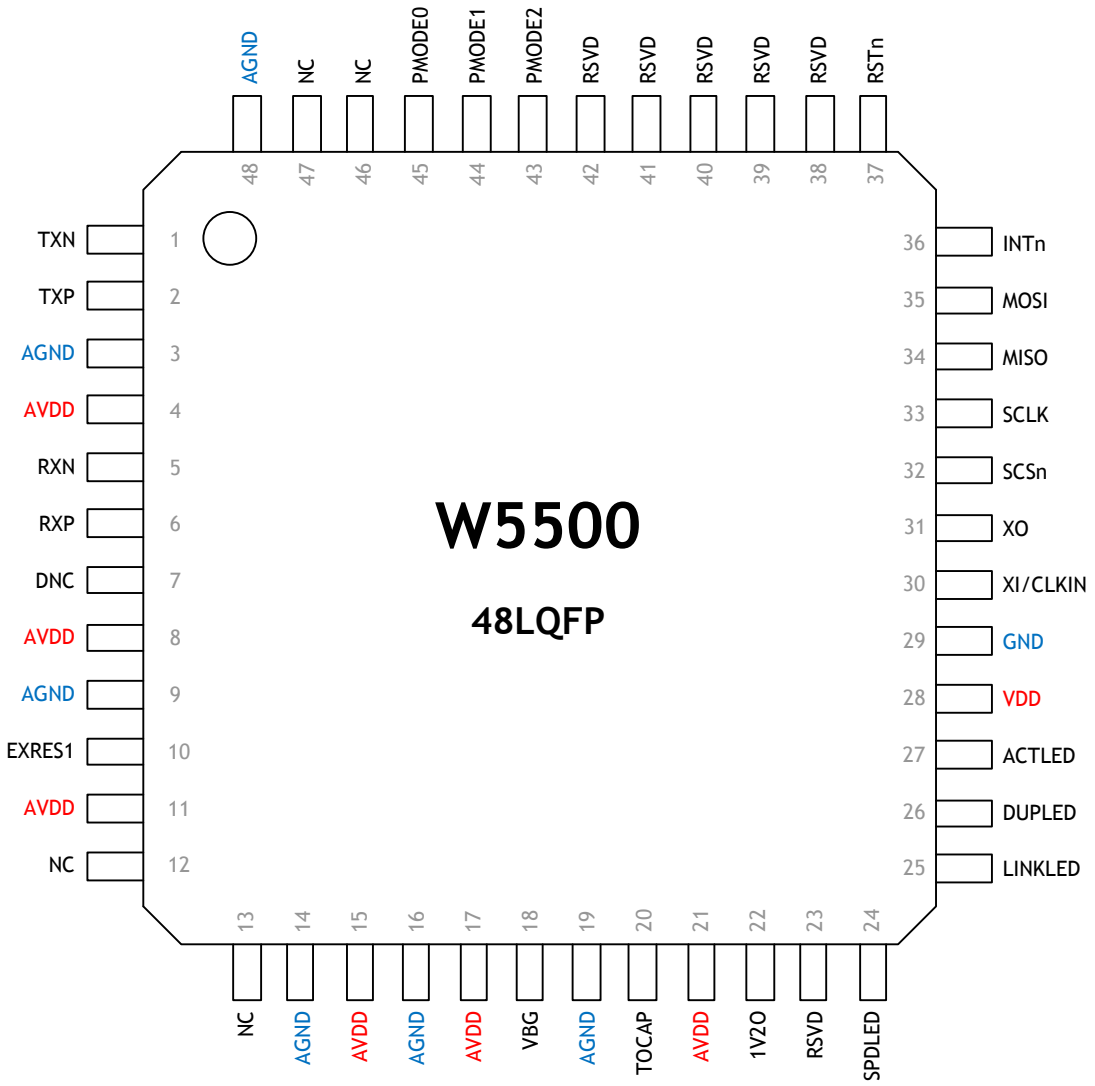


Figure 1. W5500 Pin Layout

1.1 Pin Descriptions

Table 1. Pin Type Notation

Type	Description
I	Input
O	Output
I/O	Input / Output
A	Analog
PWR	3.3V power
GND	Ground

Table 2. W5500 Pin Description

Pin No	Symbol	Internal Bias[1]	Type	Description
1	TXN	-	AO	**TXP/TXN Signal Pair**
2	TXP	-	AO	The differential data is transmitted to the media on the TXP/TXN signal pair.
3	AGND	-	GND	**Analog ground**
4	AVDD	-	PWR	**Analog 3.3V power**
5	RXN	-	AI	**RXP/RXN Signal Pair**
6	RXP	-	AI	The differential data from the media is received on the RXP/RXN signal pair.
7	DNC	-	AI/O	**Do Not Connect Pin**
8	AVDD	-	PWR	**Analog 3.3V power**
9	AGND	-	GND	**Analog ground**
10	EXRES1	-	AI/O	**External Reference Resistor** It should be connected to an external resistor (12.4KΩ, 1%) needed for biasing of internal analog circuits. Refer to the 'Analog layout guide' for details.
11	AVDD	-	PWR	**Analog 3.3V power**
12	-		-	**NC**
13	-		-	**NC**
14	AGND	-	GND	**Analog ground**
15	AVDD	-	PWR	**Analog 3.3V power**
16	AGND	-	GND	**Analog ground**
17	AVDD	-	PWR	**Analog 3.3V power**
18	VBG	-	AO	**Band Gap Output Voltage** This pin will be measured as 1.2V at 25℃. It must be left floating.
19	AGND	-	GND	**Analog ground**
20	TOCAP	-	AO	**External Reference Capacitor** This pin must be connected to a 4.7uF capacitor. The trace length to capacitor should be short to stabilize the internal signals.
21	AVDD	-	PWR	**Analog 3.3V power**
22	1V2O	-	AO	**1.2V Regulator output voltage** This pin must be connected to a 10nF capacitor.

[1] Internal Bias after hardware reset

				This is the output voltage of the internal regulator.
23	RSVD	-	I	It must be tied to GND.
24	SPDLED	-	O	**Speed LED** This shows the Speed status of the connected link. Low: 100Mbps High: 10Mbps
25	LINKLED	-	O	**Link LED** This shows the Link status. Low: Link is established High: Link is not established
26	DUPLED	-	O	**Duplex LED** This shows the Duplex status for the connected link. Low: Full-duplex mode High: Half-duplex mode
27	ACTLED	-	O	**Active LED** This shows that there is Carrier sense (CRS) from the active Physical Medium Sub-layer (PMD) during TX or RX activity. Low: Carrier sense from the active PMD High: No carrier sense
28	VDD	-	PWR	**Digital 3.3V Power**
29	GND	-	GND	**Digital Ground**
30	XI/CLKIN	-	AI	**Crystal input / External Clock input** External 25MHz Crystal Input. This pin can also be connected to single-ended TTL oscillator (CLKIN). 3.3V clock should be applied for the External Clock input. If this method is implemented, XO should be left unconnected. Refer to the 'Crystal reference schematic' for details.
31	XO	-	AO	**Crystal output** External 25MHz Crystal Output Note: Float this pin if using an external clock being driven through XI/CLKIN
32	SCSn	Pull-up	I	**Chip Select for SPI bus** This pin can be asserted low to select W5500 in SPI interface. Low: selected High: deselected
33	SCLK	-	I	**SPI clock input**

<table>
<tr><td></td><td></td><td></td><td></td><td>This pin is used to receive SPI Clock from SPI master.</td></tr>
<tr><td>34</td><td>MISO</td><td>-</td><td>O</td><td>SPI master input slave(W5500) output</td></tr>
<tr><td>35</td><td>MOSI</td><td>-</td><td>I</td><td>SPI master output slave(W5500) input</td></tr>
<tr><td>36</td><td>INTn</td><td>-</td><td>O</td><td>Interrupt output
(Active low)
Low: Interrupt asserted from W5500
High: No interrupt</td></tr>
<tr><td>37</td><td>RSTn</td><td>Pull-up</td><td>I</td><td>Reset
(Active low)
RESET should be held low at least 500 us for W5500 reset.</td></tr>
<tr><td>38</td><td>RSVD</td><td>-</td><td>I</td><td>It must be tied to GND.</td></tr>
<tr><td>39</td><td>RSVD</td><td>-</td><td>I</td><td>It must be tied to GND.</td></tr>
<tr><td>40</td><td>RSVD</td><td>-</td><td>I</td><td>It must be tied to GND.</td></tr>
<tr><td>41</td><td>RSVD</td><td>-</td><td>I</td><td>It must be tied to GND.</td></tr>
<tr><td>42</td><td>RSVD</td><td>-</td><td>I</td><td>It must be tied to GND.</td></tr>
<tr><td>43</td><td>PMODE2</td><td>Pull-up</td><td>I</td><td rowspan="3">PHY Operation mode select pins
These pins determine the network mode. Refer to the below table for details.
<table>
<tr><th colspan="3">PMODE [2:0]</th><th rowspan="2">Description</th></tr>
<tr><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>10BT Half-duplex, Auto-negotiation disabled</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>10BT Full-duplex, Auto-negotiation disabled</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>100BT Half-duplex, Auto-negotiation disabled</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>100BT Full-duplex, Auto-negotiation disabled</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>100BT Half-duplex, Auto-negotiation enabled</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Not used</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Power Down mode</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>All capable, Auto-negotiation enabled</td></tr>
</table></td></tr>
<tr><td>44</td><td>PMODE1</td><td>Pull-up</td><td>I</td></tr>
<tr><td>45</td><td>PMODE0</td><td>Pull-up</td><td>I</td></tr>
<tr><td>46</td><td>-</td><td>-</td><td>-</td><td>NC</td></tr>
<tr><td>47</td><td>-</td><td>-</td><td>-</td><td>NC</td></tr>
<tr><td>48</td><td>AGND</td><td>-</td><td>GND</td><td>Analog ground</td></tr>
</table>

EXRES1 pin and 12.4KΩ Resistor connection, picture

Clock pin connection, picture

2 HOST Interface

W5500 provides SPI (Serial Peripheral Interface) Bus Interface with 4 signals (SCSn, SCLK, MOSI, MISO) for External HOST Interface, and can operate as a SPI Slave.

The W5500 SPI can be connected to MCU like Figure 2 and Figure 3 according to the SPI Operation Mode (Variable Length Data / Fixed Length Data Mode) which will be explained in Chapter 1.3 and Chapter 1.4.

In Figure 2, SPI Bus can be shared with other SPI Devices.
Since the SPI Bus is dedicated to W5500, SPI Bus cannot be shared with other SPI Devices. It is shown in Figure 3.

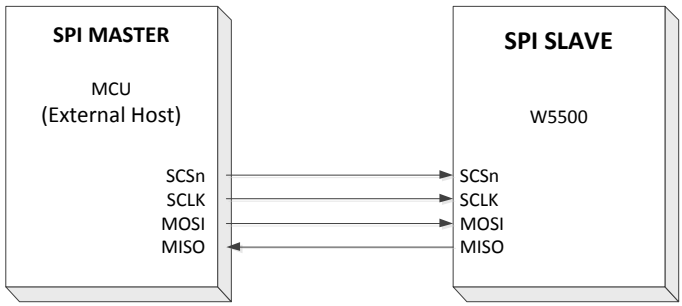


Figure 2. Variable Length Data Mode (SCSn controlled by the host)

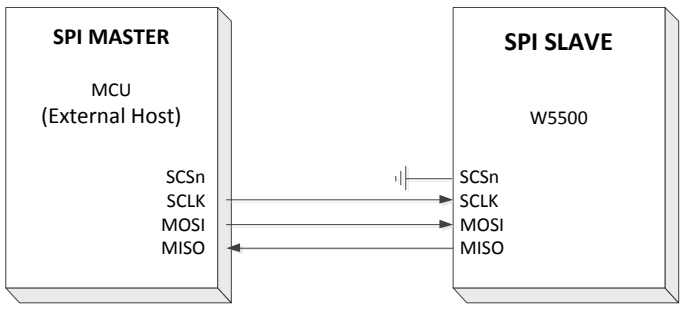


Figure 3. Fixed Length Data Mode (SCSn is always connected by Ground)

The SPI protocol defines four modes for its operation (Mode 0, 1, 2, 3).
Each mode differs according the SCLK polarity and phase.
The only difference between SPI Mode 0 and SPI Mode 3 is the polarity of the SCLK signal at the inactive state.
With SPI Mode 0 and 3, data is always latched in on the rising edge of SCLK and always output on the falling edge of SCLK.

The W5500 supports SPI Mode 0 and Mode 3.

Both MOSI and MISO signals use transfer sequence from Most Significant Bit (MSB) to Least Significant Bit (LSB) when MOSI signal transmits and MISO signal receives.

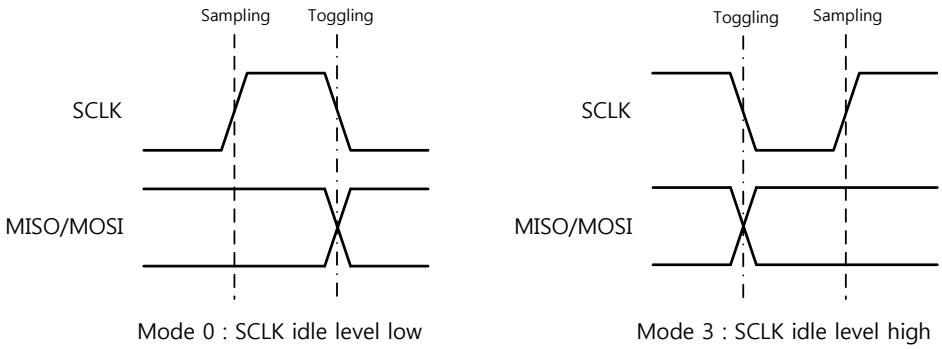


Figure 4. SPI Mode 0 & 3

2.1 SPI Operation Mode

W5500 is controlled by SPI Frame (Refer to the Chapter 2.2 SPI Frame) which communicates with the External Host. W5500 SPI Frame consists 3 phases, Address Phase, Control Phase and Data Phase.

Address Phase specifies 16 bits Offset Address for W5500 Register or TX/RX Memory.

Control Phase specifies Block which belongs to Offset Address set by Address Phase, and specifies Read/Write Access Mode and SPI Operation Mode (Variable Length Data / Fixed Length Data Mode).

And Data Phase specifies random length (N-bytes, 1 ≤ N) Data or 1 byte, 2 bytes and 4 bytes Data.

If SPI Operation Mode is set as Variable Length Data Mode (VDM), SPI Bus Signal SCSn must be controlled by the External Host with SPI Frame step.

SCSn Control Start (Assert (High-to-Low)) in Variable Length Data Mode informs SPI Frame Start (Address Phase) to W5500

SCSn Control End (De-assert (Low-to-High)) informs the SPI Frame End (Data Phase End of random N bytes) to W5500.

2.2 SPI Frame

W5500 SPI Frame consists of 16bits Offset Address in Address Phase, 8bits Control Phase and N bytes Data Phase in Figure 5.

The 8bits Control Phase is reconfigured with Block Select bits (BSB[4:0]), Read/Write Access Mode bit (RWB) and SPI Operation Mode (OM[1:0]).

Block Select bits select the block which belongs to the Offset Address.

Block Select bits select the block for Offset Address.

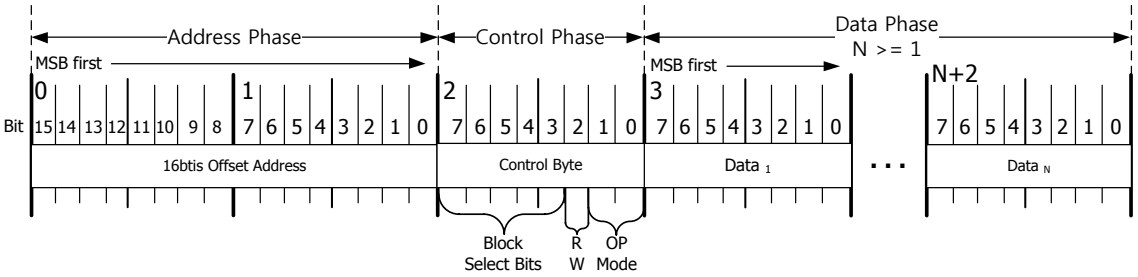


Figure 5. SPI Frame Format

W5500 supports Sequential Data Read/Write.

Data is transferred from the Base. The Base means the Offset Address which is set for 2/4/N bytes Sequential Data processing.

And Data is transferred with Offset Address which is automatically increased by 1. (auto increment addressing).

2.2.1 Address Phase

This Address Phase specifies the 16 bits Offset Address for the W5500 Registers and TX/RX Buffer Blocks.

The 16-bit Offset Address value is transferred from MSB to LSB sequentially.

When the SPI Frame has the Data Phase with 2/4/N Bytes, it supports Sequential Data Read/Write in which the SPI Frame starts from the Base and the Offset Address is increased by 1.

2.2.2 Control Phase

The Control Phase specifies the Block which belongs to the Offset Address set by Address Phase, and specifies the Read/Write Access Mode and the SPI Operation Mode.

7	6	5	4	3	2	1	0
BSB4	BSB3	BSB2	BSB1	BSB0	RWB	OM1	OM0

Bit	Symbol	Description
7~3	BSB [4:0]	**Block Select Bits** W5500 has Common Register, 8 Socket Register, TX/RX Buffer Block for each Socket. The next table shows the Block selected by BSB[4:0].

BSB [4:0]	Meaning
00000	Selects Common Register.
00001	Selects Socket 0 Register
00010	Selects Socket 0 TX Buffer
00011	Selects Socket 0 RX Buffer
00100	Reserved
00101	Selects Socket 1 Register
00110	Selects Socket 1 TX Buffer
00111	Selects Socket 1 RX Buffer
01000	Reserved
01001	Selects Socket 2 Register
01010	Selects Socket 2 TX Buffer
01011	Selects Socket 2 RX Buffer
01100	Reserved
01101	Selects Socket 3 Register
01110	Selects Socket 3 TX Buffer
01111	Selects Socket 3 RX Buffer
10000	Reserved
10001	Selects Socket 4 Register
10010	Selects Socket 4 TX Buffer
10011	Selects Socket 4 RX Buffer
10100	Reserved
10101	Selects Socket 5 Register

		10110 Selects Socket 5 TX Buffer 10111 Selects Socket 5 RX Buffer 11000 Reserved 11001 Selects Socket 6 Register 11010 Selects Socket 6 TX Buffer 11011 Selects Socket 6 RX Buffer 11100 Reserved 11101 Selects Socket 7 Register 11110 Selects Socket 7 TX Buffer 11111 Selects Socket 7 RX Buffer If the Reserved Bits are selected, the W5500 may go to malfunction.
2	RWB	**Read/Write Access Mode Bit** This sets Read/Write Access Mode. '0' : Read '1' : Write
1~0	OM [1:0]	**SPI Operation Mode Bits** This sets the SPI Operation Mode. SPI Operation Mode supports two modes, the Variable Length Data Mode and the Fixed Length Data Mode. **- Variable Length Data Mode (VDM)** : Data Length is controlled by SCSn. External Host makes SCSn Signal Assert (High-to-Low) and informs the start of the SPI Frame Address Phase to W5500. Then the external host transfers the Control Phase with OM[1:0]='00'. After N-Bytes Data Phase transfers, SCSn Signal is De-asserted (Low-to-High) and informs the end of the SPI Frame Data Phase to W5500. In VDM Mode, the SCSn must be controlled with SPI Frame unit by the External Host. (Refer to the Figure 2) **- Fixed Length Data Mode (FDM)** : In FDM, the Data Length is set by OM[1:0], these are not '00' value. So, the SCSn signal should be Low state, and has one Length type (among 1 Bytes, 2 Bytes, 4 Bytes) according to the

OM[1:0] value. (Refer to the Figure 3.)

The next table shows the SPI Operation Mode according to the OM[1:0].

OM[1:0]	Meaning
00	Variable Data Length Mode, N-Bytes Data Phase (1 ≤ N)
01	Fixed Data Length Mode , 1 Byte Data Length (N = 1)
10	Fixed Data Length Mode , 2 Byte Data Length (N = 2)
11	Fixed Data Length Mode , 4 Byte Data Length (N = 4)

2.2.3 Data Phase

With the Control Phase set by the SPI Operation Mode Bits OM[1:0], the Data Phase is set by two types of length, one type is the N-Bytes length (VDM mode) and the other type is 1/2/4 Bytes (FDM mode).

At this time, 1 byte data is transferred through MOSI or MISO signal from MSB to LSB sequentially.

2.3 Variable Length Data Mode (VDM)

In VDM mode, the SPI Frame Data Phase Length is determined by SCSn Control of the External Host. That means that the Data Phase Length can be random length (Any length from 1 Byte to N Bytes) according to the SCSn Control.

The OM[1:0] of the Control Phase should be '00' value in VDM mode.

2.3.1 Write Access in VDM

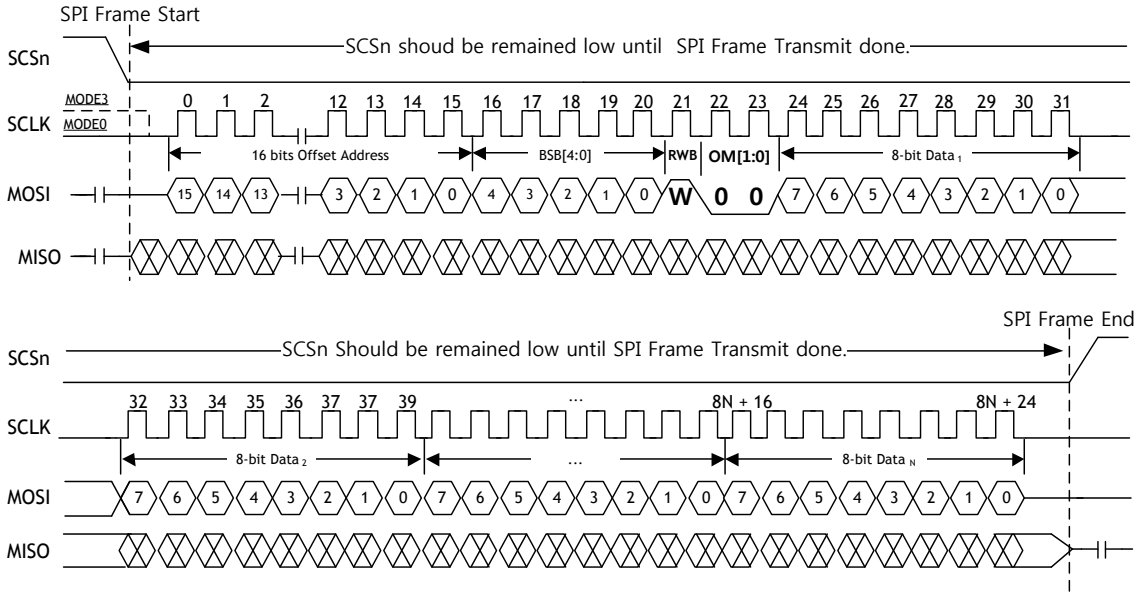


Figure 6. Write SPI Frame in VDM mode

The External Host SPI Frame during write access to W5500 is shown in Figure 6.

In VDM mode, the RWB signal is '1' (Write), OM[1:0] is '00' in SPI Frame Control Phase.

At this time the External Host assert (High-to-Low) SCSn signal before transmitting SPI Frame.

Then the Host transmits SPI Frame's all bits to W5500 through MOSI signal. All bits are synchronized with the falling edge of the SCLK.

After finishing the SPI Frame transmit, the Host deasserts SCSn signal (Low-to-High).

When SCSn is Low and the Data Phase continues, the Sequential Data Write can be supported.

1 Byte WRITE Access Example

When the Host writes Data 0xAA to 'Socket Interrupt Mask Register (SIMR) of Common Register Block by using VDM mode, the data is written with the SPI Frame below.

```
Offset Address = 0x0018
BSB[4:0]       = '00000'
RWB            = '1'
OM[1:0]        = '00'
1st Data       = 0xAA
```

The External Host asserts (High-to-Low) SCSn before transmitting SPI Frame, then the Host transmits 1 bit with synchronizing the Toggle SCLK. The External Host de-asserts (Low-to-High) the SCSn at the end of SPI Frame transmit. (Refer to the Figure 7)

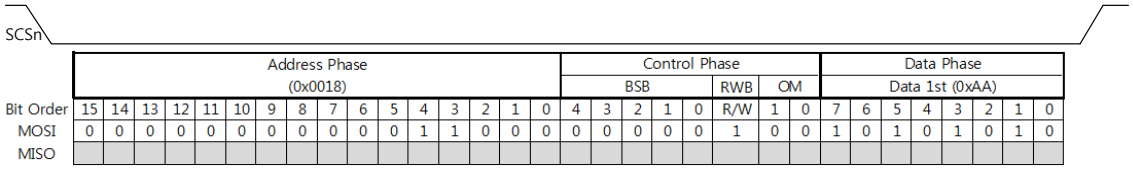


Figure 7. SIMR Register Write in VDM Mode

N-Bytes WRITE Access Example

When the Host writes 5 Bytes Data (0x11, 0x22, 0x33, 0x44, 0x55) to Socket 1's TX Buffer Block 0x0040 Address by using VDM mode, 5 bytes data are written with the SPI Frame below.

```
Offset Address = 0x0040
BSB[4:0]       = '00110'
RWB            =  '1'
OM[1:0]        = '00'
1st Data       = 0x11
2nd Data       = 0x22
3rd Data       = 0x33
4th Data       = 0x44
5th Data       = 0x55
```

The N-Bytes Write Access is shown in Figure 8.

The 5 bytes of Data (0x11, 0x22, 0x33, 0x44, 0x55) are written sequentially to Socket 1's Tx Buffer Block Address 0x0040 ~ 0x0044.

The External Host asserts (High-to-Low) SCSn before transmitting SPI Frame.

The External Host de-asserts (Low-to-High) the SCSn at the end of SPI Frame transmit.

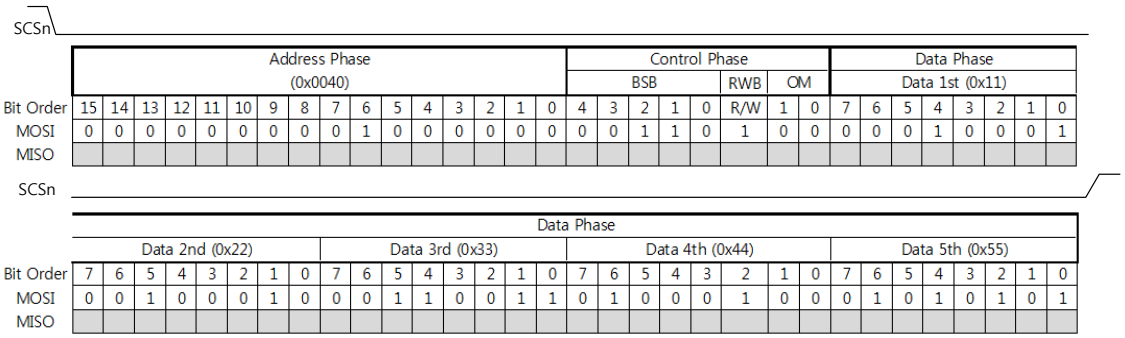


Figure 8. 5 Byte Data Write at 1th Socket's TX Buffer Block 0x0040 in VDM mode

2.3.2 Read Access in VDM

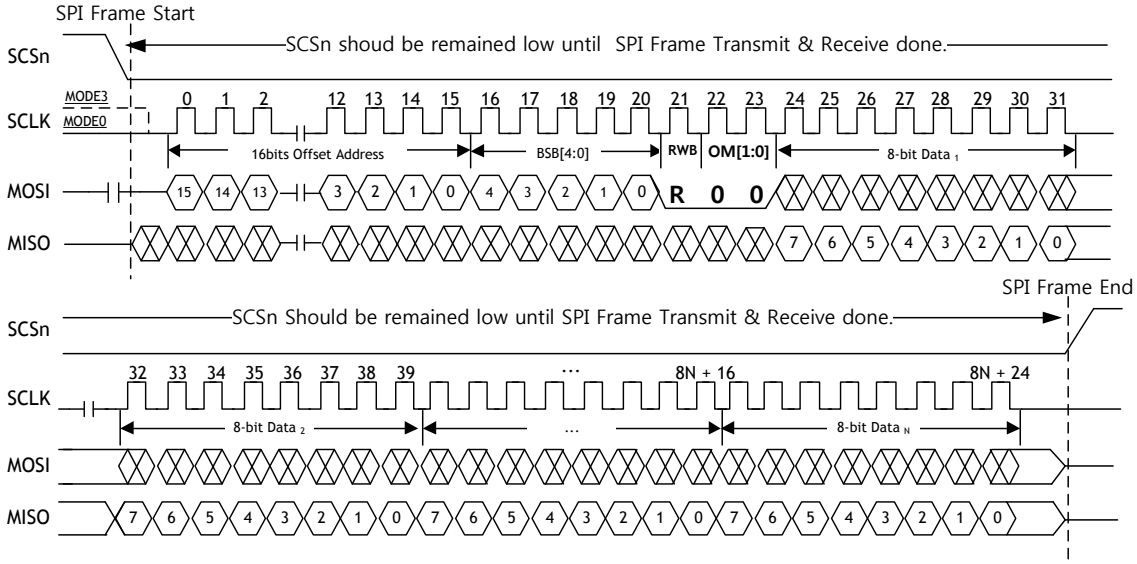


Figure 9. Read SPI Frame in VDM mode

The External Host SPI Frame during Read access to W5500 is shown in Figure 9.

In VDM mode, the RWB signal is '0' (Write), OM[1:0] is '00' in SPI Frame Control Phase.

At this time the External Host assert (High-to-Low) SCSn signal before transmitting SPI Frame.

Then the Host transmits Address and Control Phase all bits to W5500 through MOSI signal.

All bits are synchronized with the falling edge of the SCLK.

Then the Host receives all bits of Data Phase with synchronizing the rising edge of Sampling SCLK through MISO signal.

After finishing the Data Phase receive, the Host deasserts SCSn signal (Low-to-High).

When SCSn is Low and the Data Phase continues to receive, the Sequential Data Read can be supported.

1 Byte READ Access Example

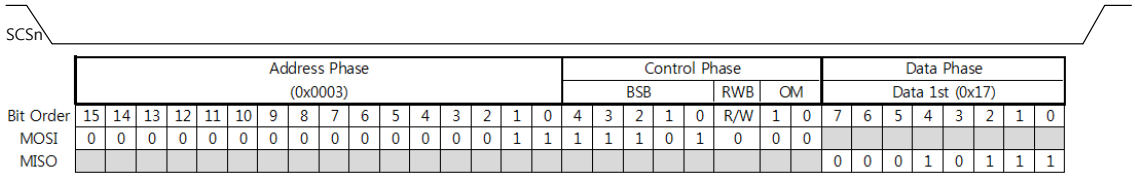
When the Host reads the 'Socket Status Register(S7_SR) of the Socket 7's Register Block by using VDM mode, the data is read with the SPI Frame below. Let's S7_SR to 'SOCK_ESTABLISHED (0x17)'.

```
Offset Address = 0x0003
BSB[4:0]       = '11101'
RWB            = '0'
OM[1:0]        = '00'
1st Data       = 0x17
```

The External Host asserts (High-to-Low) SCSn signal before transmitting SPI Frame, then the Host transmits Address and Control Phase to W5500 through the MOSI signal.

Then the Host receives Data Phase from the MISO signal.

After finishing the Data Phase receives, the Host deasserts SCSn signal (Low-to-High). (Refer to the Figure 10.)



	Address Phase (0x0003)																Control Phase								Data Phase							
																	BSB					RWB	OM		Data 1st (0x17)							
Bit Order	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	4	3	2	1	0	R/W	1	0	7	6	5	4	3	2	1	0
MOSI	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0								
MISO																									0	0	0	1	0	1	1	1

Figure 10. S7_SR Read in VDM Mode

N-Bytes Read Access Example

When the Host reads 5 Bytes Data (0xAA, 0xBB, 0xCC, 0xDD, 0xEE) from the Socket 3's RX Buffer Block 0x0100 Address by using VDM mode, 5 bytes data are read with the SPI Frame below.

```
Offset Address = 0x0100
BSB[4:0]       = '01111'
RWB            = '0'
OM[1:0]        = '00'
1st Data       = 0xAA
2nd Data       = 0xBB
3rd Data       = 0xCC
4th Data       = 0xDD
5th Data       = 0xEE
```

The N-Bytes Read Access is shown in Figure 11.

The 5 bytes of Data (0xAA, 0xBB, 0xCC, 0xDD, 0xEE) are read sequentially from the Socket 3's Rx Buffer Block Address 0x0100 ~ 0x0104.

The External Host asserts (High-to-Low) SCSn before transmitting SPI Frame.

The External Host de-asserts (Low-to-High) the SCSn at the end of the SPI Frame Data Phase.

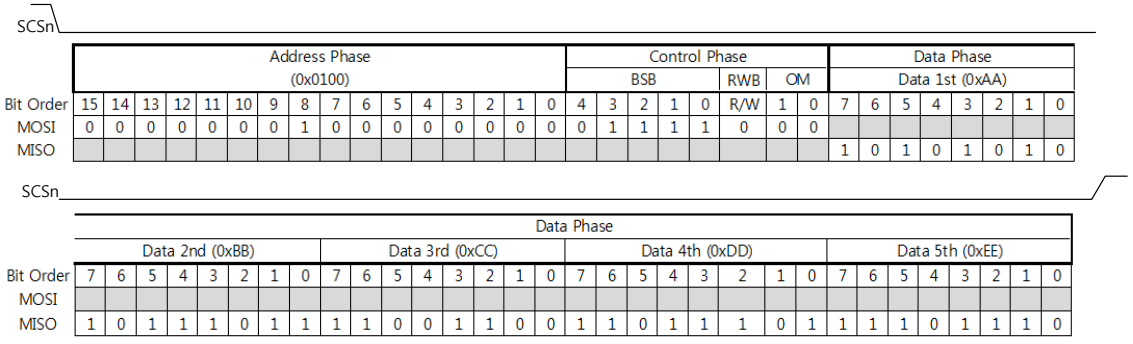


Figure 11. 5 Byte Data Read at 3rd Socket's RX Buffer Block 0x0100 in VDM mode

2.4 Fixed Length Data Mode (FDM)

The FDM mode can be used when the External Host cannot control SCSn signal.

The SCSn signal should be tied to Low (Always connected to GND) and it cannot be shared the SPI Bus with other SPI Devices. (Refer to the Figure 3.)

In VDM mode, Data Phase length is controlled by SCSn control.

But in FDM mode, Data Phase length is controlled by OM[1:0] value (‘01’ / ‘10’ / ‘11’) which is the SPI Operation Mode Bits of the Control Phase.

Because SPI Frame of FDM mode is the same as SPI Frame of VDM mode (1Byte, 2 Bytes, 4 Bytes SPI Frame) except the SCSn signal control and OM[1:0] setting, detail description will be omiited.

It is not recommended to use the FDM mode unless it is inevitable. In addition, we use only 1/2/4 Bytes SPI Frame, as described in ‘Chapter 2.4.1’ & ‘Chapter 2.4.2’. Using SPI Frame with other length of Data will cause malfunction of W5500.

2.4.1 Write Access in FDM

1 Bytes WRITE Access

	Address Phase (Any)																Control Phase								Data Phase							
																	BSB (Any)					RWB	OM		Data 1st (any)							
Bit Order	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	4	3	2	1	0	R/W	1	0	7	6	5	4	3	2	1	0
MOSI	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	1	**0**	**1**	*	*	*	*	*	*	*	*
MISO																																

Figure 12. 1 Byte Data Write SPI Frame in FDM mode

2 Bytes WRITE Access

	Address Phase (Any)																Control Phase								Data Phase							
																	BSB					RWB	OM		Data 1st (any)							
Bit Order	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	4	3	2	1	0	R/W	1	0	7	6	5	4	3	2	1	0
MOSI	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	1	**1**	**0**	*	*	*	*	*	*	*	*
MISO																																

	Data Phase							
	Data 2nd (any)							
Bit Order	7	6	5	4	3	2	1	0
MOSI	*	*	*	*	*	*	*	*
MISO								

Figure 13. 2 Bytes Data Write SPI Frame in FDM mode

4 Bytes WRITE Access

	Address Phase (Any)																Control Phase								Data Phase							
																	BSB					RWB	OM		Data 1st (any)							
Bit Order	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	4	3	2	1	0	R/W	1	0	7	6	5	4	3	2	1	0
MOSI	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	1	**1**	**1**	*	*	*	*	*	*	*	*
MISO																																

	Data Phase								Data Phase								Data Phase							
	Data 2nd (any)								Data 3rd (any)								Data 4th (any)							
Bit Order	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
MOSI	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
MISO																								

Figure 14. 4 Bytes Data Write SPI Frame in FDM mode

2.4.2 Read Access in FDM

1 Byte READ Access

	Address Phase (Any)																Control Phase								Data Phase							
																	BSB (Any)					RWB	OM		Data 1st (Any)							
Bit Order	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	4	3	2	1	0	R/W	1	0	7	6	5	4	3	2	1	0
MOSI	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0	**0**	**1**								
MISO																									*	*	*	*	*	*	*	*

Figure 15. 1 Byte Data Read SPI Frame in FDM mode

2 Bytes READ Access

	Address Phase (Any)																Control Phase								Data Phase							
																	BSB (Any)					RWB	OM		Data 1st (Any)							
Bit Order	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	4	3	2	1	0	R/W	1	0	7	6	5	4	3	2	1	0
MOSI	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0	**1**	**0**								
MISO																									*	*	*	*	*	*	*	*

	Data Phase							
	Data 2nd (Any)							
Bit Order	7	6	5	4	3	2	1	0
MOSI								
MISO	*	*	*	*	*	*	*	*

Figure 16. 2 Bytes Data Read SPI Frame in FDM mode

4 Bytes READ Access

	Address Phase (Any)																Control Phase								Data Phase							
																	BSB (Any)					RWB	OM		Data 1st (Any)							
Bit Order	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	4	3	2	1	0	R/W	1	0	7	6	5	4	3	2	1	0
MOSI	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0	**1**	**1**								
MISO																									*	*	*	*	*	*	*	*

	Data Phase								Data Phase								Data Phase							
	Data 2nd (Any)								Data 3rd (Any)								Data 4th (Any)							
Bit Order	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
MOSI																								
MISO	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

Figure 17. 4 Bytes Data Read SPI Frame in FDM mode

3 Register and Memory Organization

W5500 has 1 Common Register Block, 8 Socket Register Blocks, and TX/RX Buffer Blocks allocated at each Socket. Each block is selected by the BSB[4:0](Block Select Bit) of SPI Frame.<Figure 18> shows the selected block by the BSB[4:0] and the available offset address range of Socket TX/RX Buffer Blocks. Each Socket TX Buffer Block exists in one 16KB TX memory physically and is allocated with 2KB initially. Also, Each Socket RX Buffer Block exists in one 16KB RX Memory physically and is allocated with 2KB initially.

Regardless of the allocated size of each Socket TX/RX Buffer, it can be accessible within the 16 bits offset address range (From 0x0000 to 0xFFFF).

Refer to ‘Chapter 3.3’ for more information about 16KB TX/RX Memory organization and access method.

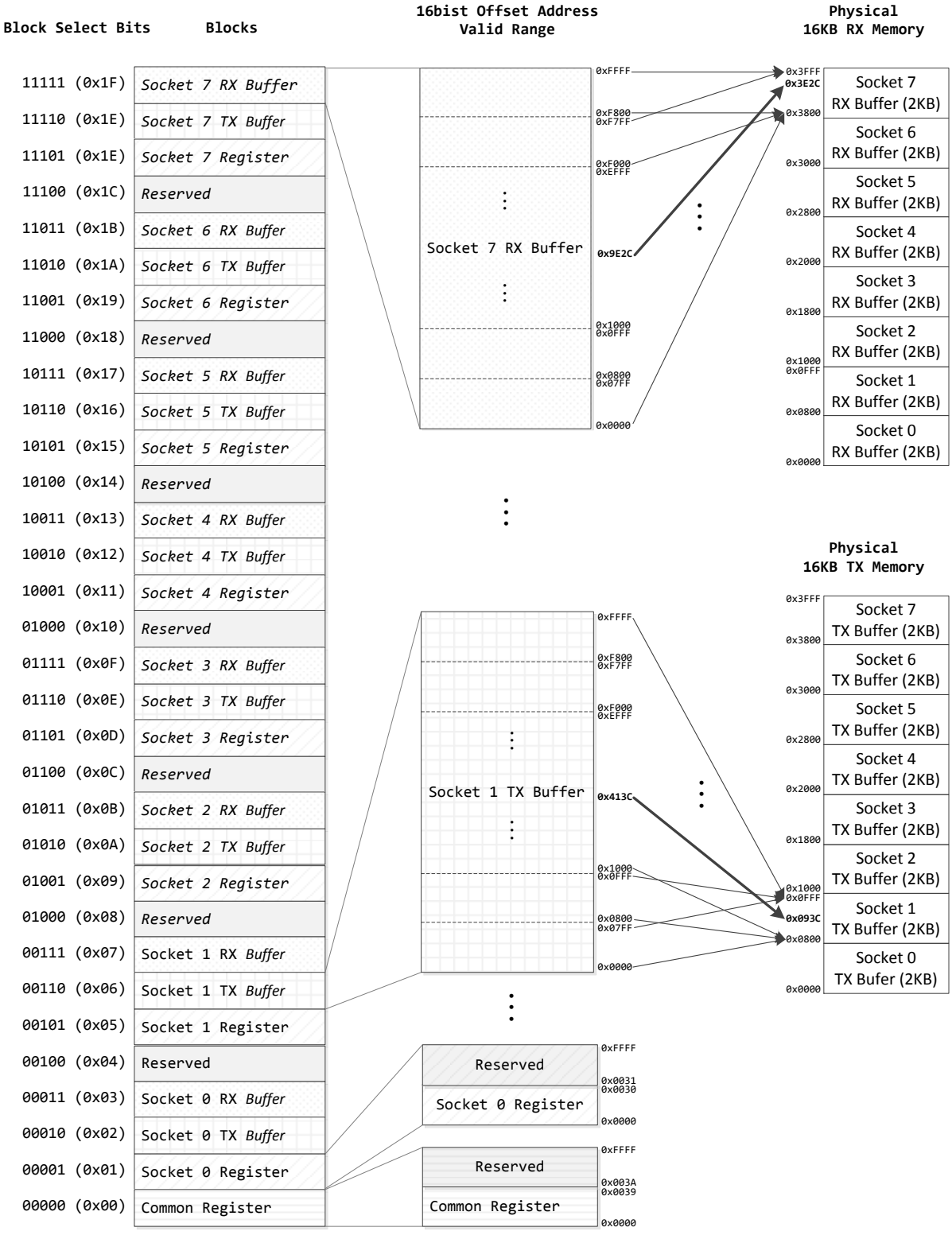


Figure 18. Register & Memory Organization

3.1 Common Register Block

Common Register Block configures the general information of W5500 such as IP and MAC address. This block can be selected by BSB[4:0] of SPI Frame. <Table 3> defines the offset address of registers in this block. Refer to 'Chapter 4.1' for the detail description of each register.

Table 3. Offset Address for Common Register

Offset	Register	Offset	Register	Offset	Register
0x0000	Mode (MR)	0x0013	Interrupt Low Level Timer (INTLEVEL0)	0x0021	(PHA3)
				0x0022	(PHA4)
		0x0014	(INTLEVEL1)	0x0023	(PHA5)
0x0001	Gateway Address (GAR0)				
0x0002	(GAR1)	0x0015	Interrupt (IR)		PPP Session Identification
0x0003	(GAR2)			0x0024	(PSID0)
0x0004	(GAR3)	0x0016	Interrupt Mask (IMR)	0x0025	(PSID1)
					PPP Maximum Segment Size
0x0005	Subnet Mask Address (SUBR0)	0x0017	Socket Interrupt (SIR)	0x0026	(PMRU0)
0x0006	(SUBR1)			0x0027	(PMRU1)
0x0007	(SUBR2)	0x0018	Socket Interrupt Mask (SIMR)		Unreachable IP address
0x0008	(SUBR3)			0x0028	(UIPR0)
		0x0019	Retry Time (RTR0)	0x0029	(UIPR1)
0x0009	Source Hardware Address (SHAR0)	0x001A	(RTR1)	0x002A	(UIPR2)
0x000A	(SHAR1)			0x002B	(UIPR3)
0x000B	(SHAR2)	0x001B	Retry Count (RCR)		Unreachable Port
0x000C	(SHAR3)			0x002C	(UPORT0)
0x000D	(SHAR4)	0x001C	PPP LCP Request Timer (PTIMER)	0x002D	(UPORT1)
0x000E	(SHAR5)				PHY Configuration
		0x001D	PPP LCP Magic number (PMAGIC)	0x002E	(PHYCFGR)
0x000F	Source IP Address (SIPR0)			0x002F ~ 0x0038	Reserved
0x0010	(SIPR1)	0x001E	PPP Destination MAC Address (PHA0)		
0x0011	(SIPR2)	0x001F	(PHA1)		Chip version
0x0012	(SIPR3)	0x0020	(PHA2)	0x0039	(VERSIONR)
0x003A ~ 0xFFFF		Reserved			

3.2 Socket Register Block

W5500 supports 8 Sockets for communication channel. Each Socket is controlled by Socket n Register Block(when 0≤n≤7). The n value of Socket n Register can be selected by BSB[4:0] of SPI Frame. <Table 4> defines the 16bits Offset Address of registers in Socket n Register Block. Refer to 'Chapter 4.3' for the detail information about each register.

Table 4. Offset Address in Socket n Register Block (0≤n≤7)

Offset	Register	Offset	Register	Offset	Register
0x0000	Socket n Mode (Sn_MR)	0x0010 0x0011	Socket n Destination Port (Sn_DPORT0) (Sn_DPORT1)	0x0024 0x0025	Socket n TX Write Pointer (Sn_TX_WR0) (Sn_TX_WR1)
0x0001	Socket n Command (Sn_CR)				
0x0002	Socket n Interrupt (Sn_IR)	0x0012 0x0013	Socket n Maximum Segment Size (Sn_MSSR0) (Sn_MSSR1)	0x0026 0x0027	Socket n RX Received Size (Sn_RX_RSR0) (Sn_RX_RSR1)
0x0003	Socket n Status (Sn_SR)	0x0014	Reserved	0x0028 0x0029	Socket n RX Read Pointer (Sn_RX_RD0) (Sn_RX_RD1)
0x0004 0x0005	Socket n Source Port (Sn_PORT0) (Sn_PORT1)	0x0015	Socket n IP TOS (Sn_TOS)		
0x0006 0x0007 0x0008 0x0009 0x000A 0x000B	Socket n Destination Hardware Address (Sn_DHAR0) (Sn_DHAR1) (Sn_DHAR2) (Sn_DHAR3) (Sn_DHAR4) (Sn_DHAR5)	0x0016	Socket n IP TTL (Sn_TTL)	0x002A 0x002B	Socket n RX Write Pointer (Sn_RX_WR0) (Sn_RX_WR1)
		0x0017 ~ 0x001D	Reserved	0x002C	Socket n Interrupt Mask (Sn_IMR)
		0x001E	Socket n Receive Buffer Size (Sn_RXBUF_SIZE)	0x002D 0x002E	Socket n Fragment Offset in IP header (Sn_FRAG0) (Sn_FRAG1)
		0x001F	Socket n Transmit Buffer Size (Sn_TXBUF_SIZE)	0x002F	Keep alive timer (Sn_KPALVTR)
0x000C 0x000D 0x000E	Socket n Destination IP Address (Sn_DIPR0) (Sn_DIPR1) (Sn_DIPR2)	0x0020 0x0021	Socket n TX Free Size (Sn_TX_FSR0) (Sn_TX_FSR1)	0x0030 ~ 0xFFFF	Reserved
		0x0022	Socket n TX Read Pointer (Sn_TX_RD0)		

0x000F	(Sn_DIPR3)	0x0023	(Sn_TX_RD1)		

3.3 Memory

W5500 has one 16KB TX memory for Socket n TX Buffer Blocks and one 16KB RX memory for Socket n RX buffer Blocks.

16KB TX memory is initially allocated 2KB size for each Socket TX Buffer Block (2KB X 8 = 16KB). The initial allocated 2KB size of Socket n TX Buffer can be re-allocated by using 'Socket n TX Buffer Size Register (Sn_TXBUF_SIZE)'.

When all Sn_TXBUF_SIZE registers have been configured, Socket TX Buffer is allocated with the configured size in 16KB TX Memory and is assigned sequentially from Socket 0 to Socket 7. And its physical memory address is automatically determined in 16KB TX memory. Therefore, the total sum of Sn_TXBUF_SIZE should be not to exceed to 16, and if not, data transmission error is occurred.

The 16KB RX memory allocation method is the same as the 16KB TX memory allocation method. 16KB RX memory is initially allocated 2KB size for each Socket RX Buffer Block (2KB X 8 = 16KB). The initial allocated 2KB size of Socket n RX Buffer can be re-allocated by using 'Socket n RX Buffer Size Register (Sn_RXBUF_SIZE)'.

When all Sn_RXBUF_SIZE registers have been configured, Socket RX Buffer is allocated with the configured size in 16KB RX Memory and is assigned sequentially from Socket 0 to Socket 7. And its physical memory address is automatically determined in 16KB RX memory. Therefore, the total sum of Sn_RXBUF_SIZE should be not to exceed to 16, and if not, data reception error is occurred.

For 16KB TX/RX memory allocation, refer to Sn_TXBUF_SIZE & Sn_RXBUF_SIZE in 'Chapter 4.2'.

The Socket n TX Buffer Block allocated in 16KB TX memory is buffer for saving data to be transmitted by host. The 16bits Offset Address of Socket n TX Buffer Block has 64KB address space ranged from 0x0000 to 0xFFFF, and it is configured with reference to 'Socket n TX Write Pointer Register (Sn_TX_WR)' & 'Socket n TX Read Pointer Register(Sn_RX_RD)'. However, the 16bits Offset Address automatically converted into the physical address to be accessible in 16KB TX memory such as <Figure 18>. Refer to 'Chapter 4.2' for Sn_TX_WR & Sn_TX_RD.

The Socket n RX Buffer Block allocated in 16KB RX memory is buffer for saving the received data through the Ethernet. The 16bits Offset Address of Socket n RX Buffer Block has 64KB address space ranged from 0x0000 to 0xFFFF, and it is configured with reference to 'Socket n RX RD Pointer Register (Sn_RX_RD)' & 'Socket n RX Write Pointer Register (Sn_RX_WR)'. However, the 16bits Offset Address automatically

converted into the physical address to be accessible in 16KB RX memory such as <Figure 18>. Refer to 'Chapter 4.2' for Sn_RX_RD & Sn_RX_WR.

4 Register Descriptions

4.1 Common Registers

MR (Mode Register) [R/W] [0x0000] [0x00][2]

MR is used for S/W reset, ping block mode and PPPoE mode.

7	6	5	4	3	2	1	0
RST		WOL	PB	PPPoE		FARP	

Bit	Symbol	Description
7	RST	If this bit is '1', All internal registers will be initialized. It will be automatically cleared as '0' after S/W reset.
6	Reserved	Reserved
5	WOL	**Wake on LAN** 0 : Disable WOL mode 1 : Enable WOL mode Notice: W5500는 WOL을 위해 Magic Packet over UDP를 지원한다. 즉,Magic Packet over UDP는 UDP Payload에 6 Bytes의 Synchronization Stream (0xFFFFFFFFFFFF), Target MAC Address의 16 번의 반복으로 구성되며, Password와 같은 옵션필드들은 무시된다. W5500에서 사용하는 Magic Packet over UDP는 source port Number의 제한이 없다. If WOL mode is enabled and the received magic packet over UDP has been normally processed, the Interrupt PIN (INTn) asserts to low. For using WOL mode, the UDP Socket should be opened with any source port number. (Refer to Socket n Mode Register (Sn_MR) for opening Socket.) Notice: The magic packet over UDP supported by W5500 consists of 6 bytes synchronization stream ('0xFFFFFFFFFFFF') and 16 times Target MAC address stream in UDP payload. The options such like password are ignored. You can use any UDP source port number for WOL mode.
4	PB	**Ping Block Mode** 0 : Disable Ping block 1 : Enable Ping block If the bit is '1', it blocks the response to a ping request.

[2] Register Notation : [Read/Write] [Address] [Reset value]

3	PPPoE	**PPPoE Mode** 0 : DisablePPPoE mode 1 : EnablePPPoE mode If you use ADSL, this bit should be '1'. For more detail, refer to the 'PPPoE Application note'
2	-	Reserved
1	FARP	**Force ARP** 0 : Disable Force ARP mode 1 : Enable Force ARP mode In Force ARP mode, It forces on sending ARP Request whenever data is sent.
0	-	Reserved

GAR (Gateway IP Address Register) [R/W] [0x0001 – 0x0004] [0x00]

GAR configures the default gateway address.

Ex) In case of "192.168.0.1"

0x0001	0x0002	0x0003	0x0004
192 (0xC0)	168 (0xA8)	0 (0x00)	1 (0x01)

SUBR (Subnet Mask Register) [R/W] [0x0005 – 0x0008] [0x00]

SUBR configures the subnet mask address.

Ex) In case of "255.255.255.0"

0x0005	0x0006	0x0007	0x0008
255 (0xFF)	255 (0xFF)	255 (0xFF)	0 (0x00)

SHAR (Source Hardware Address Register) [R/W] [0x0009 – 0x000E] [0x00]

SHAR configures the source hardware address.

Ex) In case of "00.08.DC.01.02.03"

0x0009	0x000A	0x000B	0x000C	0x000D	0x000E
0x00	0x08	0xDC	0x01	0x02	0x03

SIPR (Source IP Address Register) [R/W] [0x000F – 0x0012] [0x00]

SIPR configures the source IP address.

Ex) In case of “192.168.0.2”

0x000F	0x0010	0x0011	0x0012
192 (0xC0)	168 (0xA8)	0 (0x00)	2 (0x02)

INTLEVEL (Interrupt Low Level Timer Register) [R/W] [0x0013 – 0x0014] [0x0000]

INTLEVEL configures the Interrupt Assert Time (I_{AWT}). When the next interrupt will be occurred, Interrupt PIN (INTn) will be asserted to low after INTLEVEL time.

$$I_{AWT} = (INTLEVEL + 1) \times PLL_{CLK} \times 4 \quad \text{(when INTLEVEL > 0)}$$

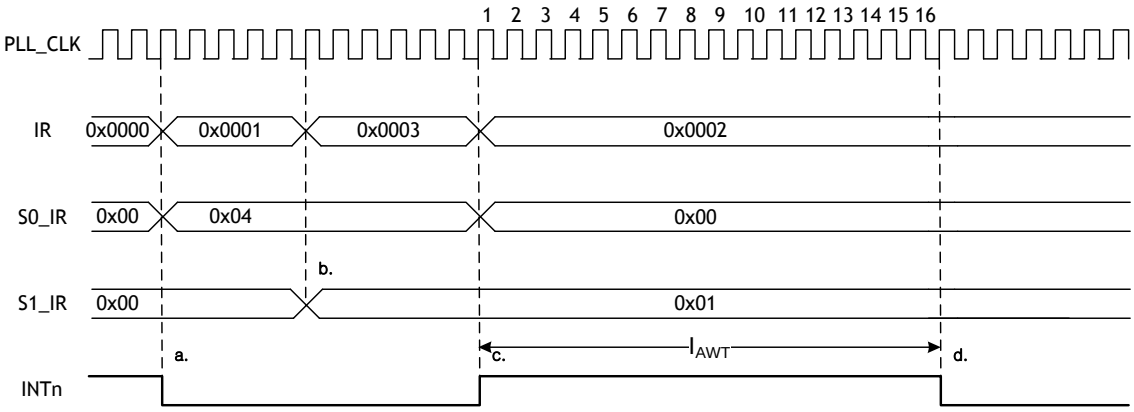


Figure 19. INTLEVEL Timing

a. When Timeout Interrupt of Socket 0 is occurred, S0_IR[3] & IR[2] bit set as '1' and then INTn PIN is asserted to low.

b. When the connection interrupt of Socket 1 is occurred before the previous interrupt processing is not completed, S1_IR[0] & IR2[1] bits set as '1' and INTn PIN is still low.

c. If the host did process the previous interrupt completely by clearing the S0_IR[3] bit, INTn PIN is de-asserted to high but S1_IR[0] & IR2[1] is still set as '1'.

d. Although S1_IR[0] & IR[2] bit are set as '1', the INTn can't be asserted to low during INTLEVEL time. After INTLEVEL time is expired, the INTn will be asserted to low.

Refer to IR2 & Sn_IR in this chapter.

IR (Interrupt Register) [R/W] [0x0015] [0x00]

IR indicates the interrupt status. Each bit of IR will be still '1' until the bit will be written to '1' by the host. When IR is not equal to '0x00', INTn PIN is asserted to low until it will be '0x00'.

7	6	5	4	3	2	1	0
CONFLICT	UNREACH	PPPoE	MP	Reserved	Reserved	Reserved	Reserved

Bit	Symbol	Description
7	CONFLICT	**IP Conflict** It is set as '1' when own source IP address is same with the sender ip address in the received ARP request.
6	UNREACH	**Destination unreachable** When receiving the ICMP (Destination port unreachable) packet, this bit is set as '1'. When this bit is '1', Destination Information such as IP address and Port number may be checked with the corresponding UIPR & UPORTR.
5	PPPoE	**PPPoE Connection Close** When PPPoE mode is enabled and the PPPoE is disconnected, this bit is set.
4	MP	**Magic Packet** When WOL mode is enabled and receives the magic packet over UDP, this bit is set.
3~0	Reserved	Reserved

IMR (Interrupt Mask Register) [R/W][0x0016][0x00]

IMR is used to mask interrupts. Each bit of IMR corresponds to each bit of IR. When a bit of IMR is '1' and the corresponding bit of IR is '1', an interrupt will be issued. In other words, when a bit of IMR is '0', an interrupt will not be issued even if the corresponding bit of IR is '1'.

7	6	5	4	3	2	1	0
IM_IR7	IM_IR6	IM_IR5	IM_IR4	Reserved	Reserved	Reserved	Reserved

Bit	Symbol	Description
7	IM_IR7	**IP Conflict Interrupt Mask** **0: Disable IP Conflict Interrupt** **1: Enable IP Conflict Interrupt**
6	IM_IR6	**Destination unreachable Interrupt Mask** **0: Disable Destination unreachable Interrupt** **1: Enable Destination unreachable Interrupt**
5	IM_IR5	**PPPoE Close Interrupt Mask** **0: Disable PPPoE Close Interrupt** **1: Enable PPPoE Close Interrupt**
4	IM_IR4	**Magic Packet Interrupt Mask** **0: Disable Magic Packet Interrupt** **1: Enable Magic Packet Interrupt**
3~0	Reserved	Reserved

SIR (Socket Interrupt Register) [R/W] [0x0017] [0x00]

SIR indicates the interrupt status of Socket. Each bit of SIR be still '1' until Sn_IR is cleared by the host. When Sn_IR is not equal to '0x00', the n-th bit of SIR is '1' and INTn PIN is asserted until SIR will be '0x00'.

7	6	5	4	3	2	1	0
S7_INT	S6_INT	S5_INT	S4_INT	S3_INT	S2_INT	S1_INT	S0_INT

Bit	Symbol	Description
7 ~ 0	Sn_INT	When the interrupt of Socket n occurs, the n-th bit of SIR becomes '1'.

SIMR (Socket Interrupt Mask Register) [R/W] [0x0018] [0x00]

Each bit of SIMR corresponds to each bit of SIR. When a bit of SIMR is '1' and the corresponding bit of SIR is '1', Interrupt will be issued. In other words, when a bit of SIMR is '0', an interrupt will be not issued even if the corresponding bit of SIR is '1'.

7	6	5	4	3	2	1	0
S7_IMR	S6_IMR	S5_IMR	S4_IMR	S3_IMR	S2_IMR	S1_IMR	S0_IMR

Bit	Symbol	Description
7 ~ 0	Sn_IMR	**Socket *n*(Sn_INT) Interrupt Mask** **0: Disable Socket *n* Interrupt** **1: Enable Socket *n* Interrupt**

RTR (Retry Time-value Register) [R/W] [0x0019 – 0x001A] [0x07D0]

RTR configures the retransmission timeout period. The unit of timeout period is 100us. The default of RTR is '0x07D0' or '2000'. And so the default timeout period is 200ms(100us X 2000).

During the time configured by RTR, W5000 wait the peer response to the packet that is transmitted by Sn_CR(CONNECT, DISCON, CLOSE, SEND, SEND_MAC, SEND_KEEP command). If the peer does not response within the RTR time, W5500 retransmits the packet or issues timeout.

Ex) When timeout-period is set as 400ms, RTR = (400ms / 1ms) X 10 = 4000(0x0FA0)

0x0019	0x001A
0x0F	0xA0

RCR (Retry Count Register) [R/W] [0x001B] [0x08]

RCR configures the number of retransmission times. When retransmission occurs as many as 'RCR+1' times, Timeout interrupt is issued (Sn_IR[TIMEOUT] = '1').

Ex) RCR = 0x0007

0x001B
0x07

The timeout of W5500 can be configurable with RTR and RCR. W5500 has two kind timeout such as Address Resolution Protocol (ARP) and TCP retransmission.

At the ARP (Refer to RFC 826, http://www.ietf.org/rfc.html) retransmission timeout, W5500 automatically sends ARP-request to the peer's IP address in order to acquire MAC address information (used for communication of IP, UDP, or TCP). During waiting for ARP-response from the peer, if there is no response during the configured RTR time, a temporary timeout is occurred and ARP-request is retransmitted. It is repeated as many as 'RCR + 1' times. Even after ARP-request retransmissions are repeated as 'RCR+1' and there is no response to the ARP-request, the final timeout is occurred and Sn_IR(TIMEOUT) becomes '1'. The time of final timeout (ARPTO) of ARP-request is as below.

$$ARP_{TO} = (RTR \times 0.1ms) \times (RCR + 1)$$

At the TCP packet retransmission timeout, W5500 transmits TCP packets (SYN, FIN, RST, DATA packets) and waits for the acknowledgement (ACK) during the configured RTR time and RCR. If there is no ACK from the peer, a temporary timeout occurs and

the TCP packet is retransmitted. The retransmission is repeated as many as 'RCR+1'. Even after TCP retransmission is repeated as 'RCR+1' and there is no response to the TCP retransmission, the final timeout is occurred and Sn_IR(TIMEOUT) becomes '1'. The time of final timeout (TCPTO) of TCP retransmission is as below.

$$TCP_{TO} = \left(\sum_{N=0}^{M} (RTR \times 2^N) \quad + \quad \left((RCR - M) \times RTR_{MAX} \right) \right) \times 0.1ms$$

N : Retransmission count, $0 \le N \le M$

M : Minimum value when RTR x $2^{(M+1)}$ > 65535 and $0 \le M \le RCR$

RTRMAX : RTR x 2^M

Ex) When RTR = 2000(0x07D0), RCR = 8(0x0008),

ARP_{TO} = 2000 X 0.1ms X 9 = 1800ms = 1.8s

TCP_{TO} = (0x07D0+0x0FA0+0x1F40+0x3E80+0x7D00+0xFA00+0xFA00+0xFA00+0xFA00) X 0.1ms

= (2000 + 4000 + 8000 + 16000 + 32000 + ((8 - 4) X 64000)) X 0.1ms

= 318000 X 0.1ms = 31.8s

PTIMER (PPP Link Control Protocol Request Timer Register) [R/W] [0x001C] [0x0028]

PTIMER configures the time for sending LCP echo request. The unit of time is 25ms.

Ex) in case that PTIMER is 200,

200 * 25(ms) = 5000(ms) = 5 seconds

PMAGIC (PPP Link Control Protocol Magic number Register) [R/W] [0x001D] [0x00]

PMAGIC configures the 4bytes magic number to be used in LCP negotiation.

Ex) PMAGIC = 0x01

0x001D
0x01

LCP Magic number = 0x01010101

PHAR (Destination Hardware Address Register in PPPoE mode)

[R/W] [0x001E-0x0023] [0x0000]

PMAC configures the PPPoE server hardware address that is acquired during PPPoE connection process.

Ex) in case that destination hardware address is 00:08:dc:12:34:56

Address	Description
0x001E	0x00
0x001F	0x08
0x0020	0xDC
0x0021	0x12
0x0022	0x34
0x0023	0x56

PSID (Session ID Register in PPPoE mode) [R] [0x0024-0x0025] [0x0000]

PSID configures the PPPoE sever session ID acquired during PPPoE connection process.

Ex) in case that Session ID is 0x1234

Value	Authentication Type
0x0024	0x12
0x0025	0x34

PMRU (Maximum Receive Unit in PPPoE mode) [R/W] [0x0026-0x0027] [0xFFFF]

PMRU configures the maximum receive unit of PPPoE.

Ex) in case that maximum receive unit in PPPoE is 0x1234

Value	Authentication Type
0x0026	0x12
0x0027	0x34

UIPR (Unreachable IP Address Register) [R] [0x0028-0x002B] [0x00000000]

UPORTR (Unreachable Port Register) [R] [0x002C-0x002D] [0x0000]

When W5500 recevies the ICMP packet(Destination port unreachable) caused to try to send data to a peer who doesn't have the port number, UNREACH bit of IR becomes '1' and UIPR & UPORT indicates the destination IP address & port number repectively.

Ex) In case of "192.168.0.11"

0x0028	0x0029	0x002A	0x002B
192 (0xC0)	168 (0xA8)	0 (0x00)	11 (0x0E)

Ex) In case of "0x1234"

0x002C	002D
18 (0x12)	52(0x34)

PHYCFGR (W5500 PHY Configuration Register) [R/W] [0x002E] [0b10111XXX]

PHYCFGR configures PHY operation mode and resets PHY. In addition, PHYCFGR indicates the status of PHY such as duplex, Speed, Link.

<table>
<tr><th>Bit</th><th>Symbol</th><th>Description</th></tr>
<tr><td>7</td><td>RST</td><td>Reset [R/W]
이 Bit 가 '0'로 설정되면 W5500 내부 PHY(Internal PHY)의 Reset 을 수행한다.
When this bit is '0', internal PHY is reset.
After PHY reset, it should be set as '1'.</td></tr>
<tr><td>6</td><td>OPMD</td><td>Configure PHY Operation Mode
1: Configure with OPMDC[2:0] in PHYCFGR
0: Configure with the H/W PINs(PMODE[2:0])
PHY Operation Mode 를 OPMDC[2:0] Bits 나 PMODE[2:0] PINs 을 이용하여 설정한다. System Reset 시, PHY 는 H/W PINs PMODE[2:0]에 의해 Operation Mode 가 설정되나, 이 bit 와 OPMDC[2:0] bits 를 이용하여 재설정될 수 있다. OPMDC[2:0]를 이용하고자 할 경우 반드시 이 Bit 를 1 로 설정한 후 PHYCFGR 의 RST bit 를 '0'으로 설정하여 PHY 를 Reset 한다.
This bit configures PHY operation mode with OPMDC[2:0] bits or PMODE[2:0] PINs. When W5500 is reset by POR or RSTn PIN, PHY operation mode is configured with PMODE[2:0] PINs by default. After POR or RSTn reset, You can re-configure PHY operation mode with OPMDC[2:0]. If you want to re-configure with PMDC[2:0], should be reset PHY by setting the RST bit to '0' after you configures this bit as '1' and OPMDC[2:0] .</td></tr>
<tr><td>5~3</td><td>OPMDC</td><td>Operation Mode Configuration Bit[R/W]
These bits select the operation mode of PHY such as following table.
<table>
<tr><th>5</th><th>4</th><th>3</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>10BT Half-duplex, Auto-negotiation disabled</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>10BT Full-duplex, Auto-negotiation disabled</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>100BT Half-duplex, Auto-negotiation disabled</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>100BT Full-duplex, Auto-negotiation disabled</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>100BT Half-duplex, Auto-negotiation enabled</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Not used</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Power Down mode</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>All capable, Auto-negotiation enabled</td></tr>
</table></td></tr>
<tr><td>2</td><td>DPX</td><td>Duplex Status [Read Only]
1: Full duplex</td></tr>
</table>

		0: Half duplex
1	SPD	**Speed Status [Read Only]** 1: 100Mpbs based 0: 10Mpbs based
0	LNK	**Link Status [Read Only]** 1: Link up 0: Link down

VERSIONR (W5500 Chip Version Register) [R] [0x0039] [0x04]

VERSIONR always indicates the W5500 version as 0x04.

4.2 Socket Registers

S*n*[3]_MR (Socket n Mode Register) [R/W] [0x0000] [0x00]

Sn_MR configures the option or protocol type of Socket n.

7	6	5	4	3	2	1	0
MULTI/ MFEN	BCASTB	ND / MC /MMB	UCASTB MIP6B	P3	P2	P1	P0

Bit	Symbol	Description
7	MULTI/ MFEN	**Multicasting in UDP mode** 0 : disable Multicasting 1 : enable Multicasting This bit is applied only when UDP mode(P[3:0] = '0010'). For using multicasting, Sn_DIPR & Sn_DPORT should be respectively configured with the multicast group IP address & port number, before Socket n is opened by OPEN command of Sn_CR **MAC Filter Enable in MACRAW mode** 0 : disable MAC Filtering 1 : enable MAC Filtering This bit is applied only when MACRAW mode(P[3:0] = '0100'). When it is '1', W5500 can receive only broadcasting and the sending packet to own. When this bit is '0', W5500 can receive all packets on Ethernet. If you want to implement Hybrid TCP/IP stack, it is recommended that this bit is set as '1' for reducing host overhead to process the all received packets.
6	BCASTB	**Broadcast Blocking in MACRAW and UDP mode** 0 : disable Broadcast Blocking 1 : enable Broadcast Blocking This bit blocks to receive broadcasting packet when UDP mode(P[3:0] = '0010') and using multicasting(Sn_MR[7] = '1'). In addition, This bit does when MACRAW mode(P[3:0] = '0100')
5	ND/MC/ MMB	**Use No Delayed ACK** 0 : Disable No Delayed ACK option

[3] *n* is Socket number (0, 1, 2, 3, 4, 5, 6, 7). *n* is set 'SNUM[2:0]' in Control Bits sets.

<table>
<tr><td></td><td></td><td>1 : Enable No Delayed ACK option,
This bit is applied only when TCP mode (P[3:0] = ‘0001’).
When this bit is ‘1’, It sends the ACK packet without delay as soon as a Data packet is received from a peer. When this bit is ‘0’, It sends the ACK packet after waiting for the timeout time configured by RTR.

Multicast
0 : using IGMP version 2
1 : using IGMP version 1
This bit is applied only when UDP mode(P[3:0] = ‘0010’) and MULTI = ‘1’.
It configures the version for IGMP messages (Join/Leave/Report).

Multicast Blocking in MACRAW mode
0 : disable Multicast Blocking
1 : enable Multicast Blocking
This bit is applied only when MACRAW mode(P[3:0] = ‘0100’). It blocks to receive the packet with multicast MAC address.</td></tr>
<tr><td>4</td><td>UCASTB
MIP6B</td><td>UNICAST Blocking in UDP mode
0 : disable Unicast Blocking
1 : enable Unicast Blocking
This bit block to receive the uncast packet when UDP mode(P[3:0] = ‘0010’) and MULTI = ‘1’.

IPv6 packet Blocking in MACRAW mode
0 : disable IPv6 Blocking
1 : enable IPv6 Blocking
This bit is applied only when MACRAW mode (P[3:0] = ‘0100’).
It blocks to receive the IPv6 packet.</td></tr>
<tr><td>3</td><td>P3</td><td rowspan="4">Protocol
This configures the protocol mode of Socket n.

<table><tr><th>P3</th><th>P2</th><th>P1</th><th>P0</th><th>Meaning</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Closed</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>TCP</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>UDP</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>MACRAW</td></tr></table>* MACRAW mode should be only used in Socket 0.</td></tr>
<tr><td>2</td><td>P2</td></tr>
<tr><td>1</td><td>P1</td></tr>
<tr><td>0</td><td>P0</td></tr>
</table>

S*n*_CR (Socket n Command Register) [R/W] [0x0001] [0x00]

This is used to set the command for Socket n such as OPEN, CLOSE, CONNECT, LISTEN, SEND, and RECEIVE. After W5500 accepts the command, the Sn_CR register is automatically cleared to 0x00. Even though Sn_CR is cleared to 0x00, the command is still being processed. To check whether the command is completed or not, please check the Sn_IR or Sn_SR.

<table>
<tr><th>Value</th><th>Symbol</th><th>Description</th></tr>
<tr><td>0x01</td><td>OPEN</td><td>Socket n is initialized and opened according to the protocol selected in Sn_MR (P3:P0). The table below shows the value of Sn_SR corresponding to Sn_MR.
<table>
<tr><th>Sn_MR (P[3:0])</th><th>Sn_SR</th></tr>
<tr><td>Sn_MR_CLOSE (‘0000’)</td><td>-</td></tr>
<tr><td>Sn_MR_TCP (‘0001’)</td><td>SOCK_INIT (0x13)</td></tr>
<tr><td>Sn_MR_UDP (‘0010’)</td><td>SOCK_UDP (0x22)</td></tr>
<tr><td>S0_MR_MACRAW (‘0100’)</td><td>SOCK_MACRAW (0x02)</td></tr>
</table></td></tr>
<tr><td>0x02</td><td>LISTEN</td><td>This is valid only in TCP mode (Sn_MR(P3:P0) = Sn_MR_TCP). In this mode, the Socket n operates as a ‘TCP server’ which is waiting for connection-request (SYN packet) from any ‘TCP client’.
The Sn_SR changes the state from SOCK_INIT to SOCKET_LISTEN.
When a ‘TCP client connection request is successfully established, the Sn_SR changes from SOCK_LISTEN to SOCK_ESTABLISHED and the Sn_IR(0) becomes from But, when a ‘TCP client’ connection request is failed, Sn_IR(3) becomes ecomes) Sn_SR changes to SOCK_CLOSED.

cf> If the destination port of the ‘TCP client’ does not exist during a connection request, that is, if there is no the listen port in W5500, W5500 will transmit a RST packet and Sn_SR is not changed.</td></tr>
<tr><td>0x04</td><td>CONNECT</td><td>This is valid only in TCP mode and operates the Socket n as a ‘TCP client’. For connecting, a connect-request (SYN packet) is sent to ‘TCP server’ configured by Sn_DIPR & Sn_DPORT(destination address & port). When a clientddress N packet) is sent to a D and the d, the n thise Sn_SR is changed to SOCK_ESTABLISHED and the Sn_IR(0)</td></tr>
</table>

		becomes ‘10) The connect-request fails as following three cases. 1. When a ARPTO occurs (Sn_IR(s)=‘1s)=R(s)=n_IR(s)destination hardware address is not acquired through the ARP-process. 2. When a SYN/ACK packet is not received and TCPTO (Sn_IR[3] = ‘1) 3. When a RST packet is received instead of a SYN/ACK packet. In these cases, Sn_SR is changed to SOCK_CLOSED.
0x08	DISCON	Valid only in TCP mode. Regardless of ‘TCP server’ or ‘TCP client’, the DISCON command processes the disconnect-process (‘Active close’ or ‘Passive close’). **Active close**: it transmits disconnect-request(FIN packet) to the connected peer **Passive close**: When FIN packet is received from peer, a FIN packet is replied back to the peer. When the disconnect-process is successful (that is, FIN/ACK packet is received successfully), Sn_SR is changed to SOCK_CLOSED. Otherwise, TCPTO occurs (Sn_IR(3)=‘1)= and then Sn_SR is changed to SOCK_CLOSED. cf> If CLOSE is used instead of DISCON, only Sn_SR is changed to SOCK_CLOSED without disconnect-process. If a RST packet is received from a peer during communication, Sn_SR is unconditionally changed to SOCK_CLOSED.
0x10	CLOSE	Close Socket n. Sn_SR is changed to SOCK_CLOSED.
0x20	SEND	SEND transmits all the data in the Socket n TX buffer. For more details, please refer to Socket n TX Free Size Register (Sn_TX_FSR), Socket n, TX Write Pointer Register(Sn_TX_WR), and Socket n TX Read Pointer Register(Sn_TX_RD).
0x21	SEND_MAC	Valid only in UDP mode. The basic operation is same as SEND. Normally SEND transmits data after destination hardware address is acquired by the automatic ARP-process(Address Resolution Protocol). But SEND_MAC transmits data without the automatic ARP-process. In this case, the destination hardware address is acquired from Sn_DHAR configured

		by host, instead of APR-process.
0x22	SEND_KEEP	Valid only in TCP mode. It checks the connection status by sending 1byte keep-alive packet. If the peer can't response to the keep-alive packet any more during timeout time, the connection is terminated and the timeout interrupt will occur.
0x00	RECV	RECV completes the processing the received data in Socket n RX Buffer by using a RX read pointer register (Sn_RX_RD). For the detail, refer to Socket n RX Received Size Register (Sn_RX_RSR), Socket n RX Write Pointer Register (Sn_RX_WR), and Socket n RX Read Pointer Register (Sn_RX_RD).

S*n*_IR (Socket n Interrupt Register) [R] [0x0002] [0x00]

Sn_IR indicates the status of Socket Interrupt such as establishment, termination, receiving data, timeout). When an interrupt occurs and the corresponding bit of Sn_IMR is '1', the corresponding bit of Sn_IR becomes '1'.

In order to clear the Sn_IR bit, the host should write the bit to '1'.

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	SEND_OK	TIMEOUT	RECV	DISCON	CON

Bit	Symbol	Description
7~5	Reserved	Reserved
4	SEND_OK	**Sn_IR(SENDOK) Interrupt** It is issued when SEND command is completed.
3	TIMEOUT	**Sn_IR(TIMEOUT) Interrupt** It is issued when ARPTO or TCPTO occurs.
2	RECV	**Sn_IR(RECV) Interrupt** It is issued whenever data is received from a peer.
1	DISCON	**Sn_IR(DISCON) Interrupt** It is issued when FIN or FIN/ACK packet is received from a peer.
0	CON	**Sn_IR(CON) Interrupt** It is issued one time when the connection with peer is successful and then Sn_SR is changed to SOCK_ESTABLISHED.

S*n*_SR (Socket n Status Register) [R] [0x0003] [0x00]

Sn_SR indicates the status of Sockt n. The status of Socket n can be changed by Sn_CR or by some packet during data communication.

Value	Symbol	Description
0x00	SOCK_CLOSED	It indicates Socket n is released. When DICON, CLOSE command is ordered, or when a timeout occurs, it is changed to SOCK_CLOSED regardless of previous status.
0x13	SOCK_INIT	It indicates Socket n is opened with TCP mode. It is changed to SOCK_INIT when Sn_MR (P[3:0]) = '0001' and OPEN command is ordered. After SOCK_INIT, You can order LISTEN /CONNECT command.
0x14	SOCK_LISTEN	It indicates Socket n is operating as 'TCP server' mode and waiting connection-request (SYN packet) from a peer ('TCP client'). It will be changed to SOCK_ESTALBLISHED when the connection-request is accepted successfully. Otherwise it will be changed to SOCK_CLOSED after TCPTO occurred (Sn_IR(TIMEOUT) = '1').
0x17	SOCK_ESTABLISHED	It indicates Socket n's connection is established. It is changed to SOCK_ESTABLISHED when SYN packet from conn CLIENTd to SOCK_ESTABLISaccepted at SOCK_LISTEN, or when CONNECT command is successfully performed at SOCK_ INIT. During SOCK_ESTABLISHED, DATA packet can be transferred, that is, SEND or RECV command can be performed.
0x1C	SOCK_CLOSE_WAIT	It indicates Socket n received disconnect-request (FIN packet) from the connected peer. This is half-closing status, So that, data can be transferred. For full-closing, DISCON command is ordered. But For just-closing, CLOSE command is ordered.
0x22	SOCK_UDP	It indicates Socket n is opened in UDP mode(Sn_MR(P[3:0]) = '0010'). It is changed to SOCK_UPD when Sn_MR(P[3:0]) = '0010') and OPEN command is ordered. Unlikely TCP mode, UDP mode Socket can transfer data without connection-process.

0x02	SOCK_MACRAW	It indicates Socket 0 is opened in MACRAW mode (S0_MR(P[3:0]) = '0100'). And It is valid only in Socket 0. It is changed to SOCK_MACRAW when S0_MR(P[3:0] = '0100' and OPEN command is ordered. Likely UDP mode socket, MACRAW mode Socket 0 can transfer a MAC packet (Ethernet frame) without connection-process.

The following table show a temporary status indicated during changing Socket n status.

Value	Symbol	Description
0x15	SOCK_SYNSENT	It indicates Socket n sent connect-request packet (SYN packet) to a peer. It is temporarily shown when Sn_SR is changed from SOCK_INIT to SOCK_ESTABLISHED by CONNECT command. If connect-accept(SYN/ACK packet) is received from the peer at SOCK_SYNSENT, it is changed to SOCK_ESTABLISHED. Otherwise, it is changed to SOCK_CLOSED after TCPTO (Sn_IR[TIMEOUT] = '1') occurred.
0x16	SOCK_SYNRECV	It indicates Socket n did successfully receive the connect-request packet (SYN packet) from a peer. At this time, if socket n send the response (SYN/ACK packet) to the peer successfully, it is changed to SOCK_ESTABLISHED. If not, it is changed to SOCK_CLOSED after timeout occurs (Sn_IR[TIMEOUT] = '1').
0x18	SOCK_FIN_WAIT	These indicate Socket n is closing. These are shown in disconnect-process such as active-close and passive-close. When Disconnect-process is successfully completed, or when timeout occurs, these are changed to SOCK_CLOSED.
0x1A	SOCK_CLOSING	
0X1B	SOCK_TIME_WAIT	
0X1D	SOCK_LAST_ACK	It indicates Socket n is waiting the response (FIN/ACK packet) to disconnect-request (FIN packet) by passive-close. It is changed to SOCK_CLOSED when Socket n received the response successfully, or when timeout occurs (Sn_IR[TIMEOUT] = '1').

S*n*_PORT (Socket n Source Port Register) [R/W] [0x0004-0x0005] [0x0000]

It configures the source port number of Socket n. It is valid when Socket n is used in TCP/UPD mode. It should be set before OPEN command is ordered.

Ex) In case of Socket 0 Port = 5000(0x1388), configure as below,

0x0004	0x0005
0x13	0x88

S*n*_DHAR (Socket n Destination Hardware Address Register)

[R/W] [0x0006-0x000B] [0xFFFFFFFFFFFF]

It configures the destination hardware address of Socket n when using SEND_MAC command in UDP mode or it indicates that is acquired in ARP-process by CONNECT/SEND command.

Ex) In case of Socket 0 Destination Hardware address = 08.DC.00.01.02.10, configuration is as below.

0x0006	0x0007	0x0008	0x0009	0x000A	0x000B
0x08	0xDC	0x00	0x01	0x02	0x0A

S*n*_DIPR (Socket n Destination IP Address Register)

[R/W] [0x000C-0x000F] [0x00000000]

It configures or indicates the destination IP address of Socket n. It is valid when Socket n is used in TCP/UDP mode.

At TCP client mode, it configures an IP address of 'TCP server' before CONNECT command.

At TCP server mode, it indicates an IP address of 'TCP client' after successfully establishing connection.

At UDP mode, it configures an IP address of peer to be received the UDP packet by SEND or SEND_MAC command.

Ex) In case of Socket 0 Destination IP address = 192.168.0.11, configure as below.

0x000C	0x000D	0x000E	0x000F
192 (0xC0)	168 (0xA8)	0 (0x00)	11 (0x0B)

S*n*_DPORT (Socket n Destination Port Register) [R/W] [0x0010-0x0011] [0x00]

It configures or indicates the destination port number of Socket n. It is valid when Socket n is used in TCP/UDP mode.

At 'TCP client' mode, it configures the listen port number of 'TCP server' before CONNECT command.

At 'TCP Server' mode, it indicates the port number of TCP client after successfully establishing connection.

At UDP mode, it configures the port number of peer to be transmitted the UDP packet by SEND/SEND_MAC command.

Ex) In case of Socket 0 Destination Port = 5000(0x1388), configure as below,

0x0010	0x0011
0x13	0x88

Sn_MSSR (Socket n Maximum Segment Size Register) [R/W] [0x0012-0x0013] [0x0000]

It configures or indicates the MTU(Maximum Transfer Unit) of Socket n.

If it is not set, it is set with the default MTU. It is valid when Socket n is used in TCP / UDP mode. However, when you used in PPPoE mode(MR[PPPoE] = '1'), It is determined within PPPoE MTU.

Mode	**Normal** (MR(PPPoE)='0')		**PPPoE** (MR(PPPoE)='1')	
	Default MTU	Range	Default MTU	Range
TCP	1460	1 ~ 1460	1452	1 ~ 1452
UDP	1472	1 ~ 1472	1464	1 ~ 1464
MACRAW	1514			

When Socket n is used in MACRAW mode, it is applied the default MTU because MTU is not processed internally. Therefore, when transmitting the data bigger than default MTU, the host should manually divides the data into the unit of default MTU. When Socket n is used in TCP/UDP mode, if transmitting data is bigger than MTU, the data is automatically divided into the unit of MTU.

At UDP mode, because there is no such connection-process as TCP mode, it is used the configured MTU. When transmitting data to a peer with the different MTU size, the ICMP(Fragment MTU) packet maybe be received. In this case, IR(FMTU) becomes '1' and the peer information such as the MTU size and IP address is indicated from FMTUR and UIPR respectively. If IR[MTU] = '1', you can't transmit the data more to the peer. And so, To resume the communication with the peer, do as follows.

1. Close the Socket n by CLOSE command.
2. Set Sn_MSS to the indicated MTU from FMTUR
3. Open the Socket n by OPEN command
4. Resume the communication with the peer.

Ex) In case of Socket 0 MSS = 1460 (0x05B4), configure as below,

0x0012	0x0013
0x05	0xB4

Sn_TOS (Socket n IP Type of Service Register) [R/W] [0x0015] [0x00]

It configures the TOS(Type Of Service field in IP Header) of Socket n.

It is set before OPEN command.

For more the detail, refer to http://www.iana.org/assignments/ip-parameters.

S*n*_TTL (Socket n TTL Register) [R/W] [0x0016] [0x80]

It configures the TTL(Time To Live field in IP header) of Socket n.

It is set before OPEN command.

For more the detail, refer to http://www.iana.org/assignments/ip-parameters.

S*n*_RXBUF_SIZE (Socket n RX Buffer Size Register) [R/W] [0x001E] [0x02]

It configures the RX buffer block size of Socket n. Socket n RX Buffer Block size can be configured with 1,2,4,8, and 16 Kbytes. If other size is configured, the data can't be normally received from a peer.

Although Socket n RX Buffer Block size is initially configured to 2Kbytes, You can be re-configurable its size using Sn_RXBUF_SIZE. The total sum of Sn_RXBUF_SIZE can't be exceeded to 16Kbytes. When exceeded, the data reception error is occurred.

When all Sn_RXBUF_SIZE have been configured, Socket n RX Buffer is allocated with the configured size in 16KB RX Memory and is assigned sequentially from Socket 0 to Socket 7.

Socket n RX Buffer Block can be accessible with the 16bits Offset Address ranged from 0x0000 to 0xFFFF regardless of the configured size. (Refer to Sn_RX_RD & Sn_RX_WR).

Value (dec)	0	1	2	4	8	16
Buffer size	0KB	1KB	2KB	4KB	8KB	16KB

Ex) Socket 0 RX Buffer Size = 8KB

0x001E
0x08

S*n*_TXBUF_SIZE (Socket n TX Buffer Size Register) [R/W] [0x001F] [0x02]

It configures the TX buffer block size of Socket n. Socket n TX Buffer Block size can be configured with 1,2,4,8, and 16 Kbytes. If other size is configured, the data can't be normally transmitted to a peer.

Although Socket n TX Buffer Block size is initially configured to 2Kbytes, You can be re-configurable its size using Sn_TXBUF_SIZE. The total sum of Sn_TXBUF_SIZE can't be exceeded to 16Kbytes. When exceeded, the data transmission error is occurred.

When all Sn_TXBUF_SIZE have been configured, Socket n TX Buffer is allocated with the configured size in 16KB TX Memory and is assigned sequentially from Socket 0 to Socket 7.

Socket n TX Buffer Block can be accessible with 16bits Offset Address ranged from 0x0000 to 0xFFFF regardless of the configured size. (Refer to Sn_TX_WR & Sn_TX_RD).

Value (dec)	0	1	2	4	8	16
Buffer size	0KB	1KB	2KB	4KB	8KB	16KB

Ex) Socket 0 TX Buffer Size = 4KB

0x001F
0x04

S*n*_TX_FSR (Socket n TX Free Size Register) [R] [0x0020-0x0021] [0x0800]

It indicates the free size of Socket n TX Buffer Block. It is initialized to the configured size by Sn_TXBUF_SIZE. Data is bigger than Sn_TX_FSR that should not be saved in the Socket n TX Buffer because the bigger data overwrites the previous saved data to be sending or send. Therefore, be sure to check before saving the data to the Socket n TX Buffer, and if data is equal to or smaller than its checked size, transmit the data with SEND/SEND_MAC command after saving the data in Socket n TX buffer. But, if data is bigger than its checked size, transmit the data after dividing into the checked size and saving in the Socket n TX buffer.

If Sn_MR(P[3:0]) is not TCP mode('0001'), it is automatically calculated as the difference between 'Socket n TX Write Pointer (Sn_TX_WR)' and 'Socket n TX Read Pointer (Sn_TX_RD)'.

If Sn_MR(P[3:0]) is TCP mode('0001'), it is automatically calculated as the difference between Sn_TX_WR and the internal ACK pointer which indicates the point of data is received already by the connected peer.

Ex) In case of 2048(0x0800) in S0_TX_FSR,

0x0020	0x0021
0x08	0x00

Sn_TX_RD (Socket n TX Read Pointer Register) [R] [0x0022-0x0023] [0x0000]

It is initialized by OPEN command. However, if Sn_MR(P[3:0]) is TCP mode('0001'), it is re-initialized by connection-process.

After its initialization, it is auto-increased by SEND command. SEND command transmits the saved data from the current Sn_TX_RD to the Sn_TX_WR in the Socket n TX Buffer. After transmitting the saved data, the SEND command increases the Sn_TX_RD' as same as the Sn_TX_WR. If its increment value exceeds the maximum

value 0xFFFF, that is, is greater than 0x10000 and the carry bit occurs, then it is automatically updated with the lower 16bits value ignored the carry bit.

S*n*_TX_WR (Socket n TX Write Pointer Register) [R/W] [0x0024-0x0025] [0x0000]

It is initialized by OPEN command. However, if Sn_MR(P[3:0]) is TCP mode('0001'), it is re-initialized by connection-process.

It is sure to be read or to be update like as follows.

1. Read it, for getting the starting save address of the transmitting data.
2. Save the transmitting data from the starting address of Socket n TX buffer.
3. After saving the transmitting data, Update Sn_TX_WR to the increased value as many as transmitting data size. If the increment value exceeds the maximum value 0xFFFF, that is, is greater than 0x10000 and the carry bit occurs, update with the lower 16bits value ignored the carry bit.
4. Transmit the saved data in Socket n TX Buffer by using SEND/SEND command

Sn_RX_RSR (Socket n Received Size Register) [R] [0x0026-0x0027] [0x0000]

It indicates the data size received and saved in Socket n RX Buffer. It is not exceeded to Sn_RXBUF_SIZE and is calculated as the difference between 'Socket n RX Write Pointer (Sn_RX_WR)' and 'Socket n RX Read Pointer (Sn_RX_RD)'.

Ex) In case of 2048(0x0800) in S0_RX_RSR,

0x0026	0x0027
0x08	0x00

S*n*_RX_RD (Socket n RX Read Data Pointer Register) [R/W] [0x0028-0x0029] [0x0000]

It is initialized by OPEN command . It is sure to be read or to be updated like as follows.

1. Read it, for getting the starting save address of the received data
2. Read data from the starting address of Socket n RX Buffer.
3. After reading the received data, Update Sn_RX_RD to the increased value as many as the reading size. If the increment value exceeds the maximum value 0xFFFF, that is, is greater than 0x10000 and the carry bit occurs, update with the lower 16bits value ignored the carry bit.
4. Order RECV command, for notifying the updated Sn_RX_RD to W5500.

Ex) In case of 2048(0x0800) in S0_RX_RD,

0x0028	0x0029
0x08	0x00

Sn_RX_WR (Socket n RX Write Pointer Register) [R/W] [0x002A-0x002B] [0x0000]

It is initialized by OPEN command. And it is auto-increased by the data reception.

If its increased value exceeds the maximum value 0xFFFF, that is, is greater than 0x10000 and the carry bit occurs, it is automatically updated with the lower 16bit value ignored the carry bit.

Ex) In case of 2048(0x0800) in S0_RX_WR,

0x002A	0x002B
0x08	0x00

Sn_IMR (Socket n Interrupt Mask Register) [R/W] [0x002C] [0xFF]

It masks the interrupt of Socket n. Its each bit corresponds to the each bit of Sn_IR. When a Socket n Interrupt is occurred and the corresponding bit of Sn_IMR is '1', the corresponding bit of Sn_IR becomes '1'. When both the corresponding bit of Sn_IMR and Sn_IR are '1' and the n-th bit of IR is '1', Host is interrupted by asserted INTn PIN to low.

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	SEND_OK	TIMEOUT	RECV	DISCON	CON

Bit	Symbol	Description
7~5	Reserved	Reserved
4	SENDOK	Sn_IR(SENDOK) Interrupt Mask
3	TIMEOUT	Sn_IR(TIMEOUT) Interrupt Mask
2	RECV	Sn_IR(RECV) Interrupt Mask
1	DISCON	Sn_IR(DISCON) Interrupt Mask
0	CON	Sn_IR(CON) Interrupt Mask

Sn_FRAG (Socket n Fragment Register) [R/W] [0x002D-0x002E] [0x0400]

It configures the FRAG(Fragment field in IP header).

Ex) Sn_FRAG0 = 0x0000 (Don't Fragment)

0x002D	0x002E

0x00	0x00

Sn_KPALVTR (Socket n Keep Alive Time Register) [R/W] [0x002F] [0x00]

It configures transmitting timer of 'KEEP ALIVE(KA)' packet of SOCKETn. It is valid only in TCP mode, and ignored in other modes. The time unit is 5s.

KA packet is transmittable after Sn_SR is changed to SOCK_ESTABLISHED and data is transmitting or receiving to/from a peer at least once. In case of 'Sn_KPALVTR > 0', W5500 automatically transmits KA packet after time-period for checking the TCP connection (Auto-keepalive-process). In case of 'Sn_KPALVTR = 0', Auto-keep-alive-process does not operate, and KA packet can be transmitted by SEND_KEEP command by the host (Manual-keep-alive-process). Manual-keep-alive-process is ignored in case of 'Sn_KPALVTR > 0'.

Ex) Sn_KPALVTR = 10 (Keep Alive packet will be transmitted every 50 seconds.)

0x002F
0x0A

5 Electrical Specifications

5.1 Absolute Maximum Ratings

Symbol	Parameter	Rating	Unit
V_{DD}	DC Supply voltage	-0.5 to 4.6	V
V_{IN}	DC input voltage	-0.5 to 6	V
V_{OUT}	DC output voltage	-0.5 to 4.6	V
I_{IN}	DC input current	±5	mA
T_{OP}	Operating temperature	-40 to +85	°C
T_{STG}	Storage temperature	-65 to +150	°C

***COMMENT**: Stressing the device beyond the 'Absolute Maximum Ratings' may cause permanent damage.

5.2 Absolute Maximum Ratings (Electrical Sensitivity)

Electrostatic discharge (ESD)

Symbol	Parameter	Test Condition	Class	Maximum value(1)	Unit
VESD(HBM)	Electrostatic discharge voltage (human body model)	TA = +25 °C conforming to MIL-STD 883F Method 3015.7	2	2000	V
VESD(MM)	Electrostatic discharge voltage (man machine model)	TA = +25 °C conforming to JEDEC EIA/JESD22 A115-A	B	200	V
VESD(CDM)	Electrostatic discharge voltage (charge device model)	TA = +25 °C conforming to JEDEC JESD22 C101-C	III	500	V

Static latchup

Symbol	Parameter	Test Condition	Class	Maximum value(1)	Unit
LU	Static latch-up class	TA = +25 °C conforming to JESD78A	I	≥ ±200	mA

5.3 DC Characteristics

(Test Condition: Ta = -40 to 85°C)

Symbol	Parameter	Test Condition	Min	Typ	Max	Unit
V_{DD}	Supply voltage	Apply VDD, AVDD	2.97	3.3	3.63	V
V_{IH}	High level input voltage		2.0		5.5	V
V_{IL}	Low level input voltage		- 0.3		0.8	V
V_T	Threshold point	All inputs except XI	1.30	1.41	1.53	V
V_{T+}	Schmitt trig Low to High Threshold point	All inputs except XI	1.53	1.64	1.73	V
V_{T-}	Schmitt trig High to Low Threshold point	All inputs except XI	0.95	1.02	1.09	V
T_J	Junction temperature		0	25	125	°C
I_L	Input Leakage Current				±1	μA
R_{PU}	Pull-up Resistor	SCSn, RSTn, PMODE[2:0]	62	77	112	Kohm
R_{PD}	Pull-down Resistor	RSVD(Pin 38 ~ Pin 42)	48	85	174	Kohm
V_{OL}	Low level output voltage	IOL = 8mA, All outputs except XO			0.4	V
V_{OH}	High level output voltage	IOH = 8mA, All outputs except XO	2.4			V
I_{OL}	Low level output Current	VOL = 0.4V, All outputs except XO	8.6	13.9	18.9	mA
I_{OH}	High level output Current	VOH = 2.4V, All outputs except XO	12.5	26.9	47.1	mA
I_{DD1}	Supply Current (Normal operation mode)			TBD		mA
I_{DD2}	Supply Current (Power Down mode)			TBD		μA

5.4 POWER DISSIPATION

(Test Condition: VDD=3.3V, Ta = 25°C)

Condition	Min	Typ	Max	Unit
100M Link	-	TBD	TBD	mA
10M Link	-	TBD	TBD	mA
Un-Link	-	TBD	TBD	mA
100M Transmitting	-	TBD	TBD	mA
10M Transmitting	-	TBD	TBD	mA
Power Down mode	-	TBD	TBD	mA

5.5 AC Characteristics

5.5.1 Reset Timing

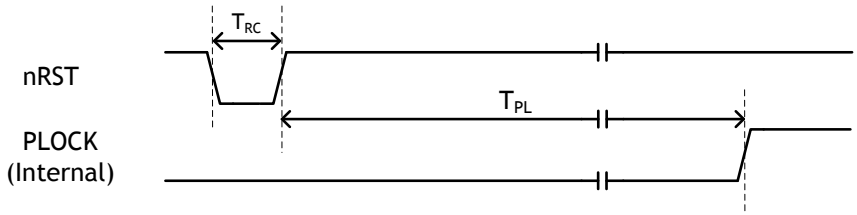


Figure 20. Reset Timing

Symbol	Description	Min	Max
T_{RC}	Reset Cycle Time	500 us	-
T_{PL}	RSTn to internal PLL LOCK	-	1 ms

5.5.2 Wake up Time

Voltage Regulator Wake up Time: **10us**

5.5.3 Crystal Characteristics

Parameter	Range
Frequency	25 MHz
Frequency Tolerance (at 25℃)	±30 ppm
Shunt Capacitance	7pF Max
Drive Level	59.12uW/MHz
Load Capacitance	27pF

Aging (at 25℃)	±3ppm / year Max

5.5.4 SPI Timing

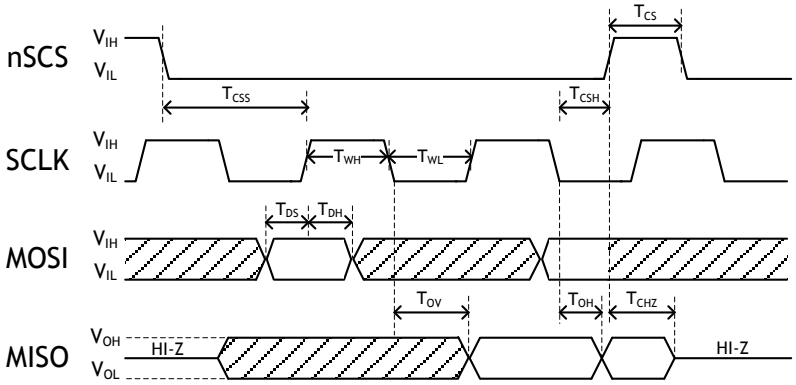


Figure 21. SPI Timing

Symbol	Description	Min	Max	Units
F_{SCK}	SCK Clock Frequency		80/33.3[4]	MHz
T_{WH}	SCK High Time	6		ns
T_{WL}	SCK Low Time	6		ns
T_{CS}	SCSn High Time	30		ns
T_{CSS}	SCSn Setup Time	5	-	ns
T_{CSH}	SCSn Hold Time	5		ns
T_{DS}	Data In Setup Time	3		ns
T_{DH}	Data In Hold Time	3		ns
T_{OV}	Output Valid Time		5	ns
T_{OH}	Output Hold Time	0		ns
T_{CHZ}	SCSn High to Output Hi-Z		2.1[5]	ns

[4] Theoretical Guaranteed Speed

Even though theoretical design speed is 80MHz, the signal in the high speed may be distorted because of the circuit crosstalk and the length of the signal line. The minimum guaranteed speed of the SCLK is 33.3 MHz which was tested and measured with the stable waveform.

Please refer to the SPI Application Note which shows the WIZnet test environment and results.

[5] 2.1ns is when pn loaded with 30pF. The time is shorter with lower capacitance.

5.5.5 Transformer Characteristics

Parameter	Transmit End	Receive End
Turn Ratio	1:1	1:1
Inductance	350 uH	350 uH

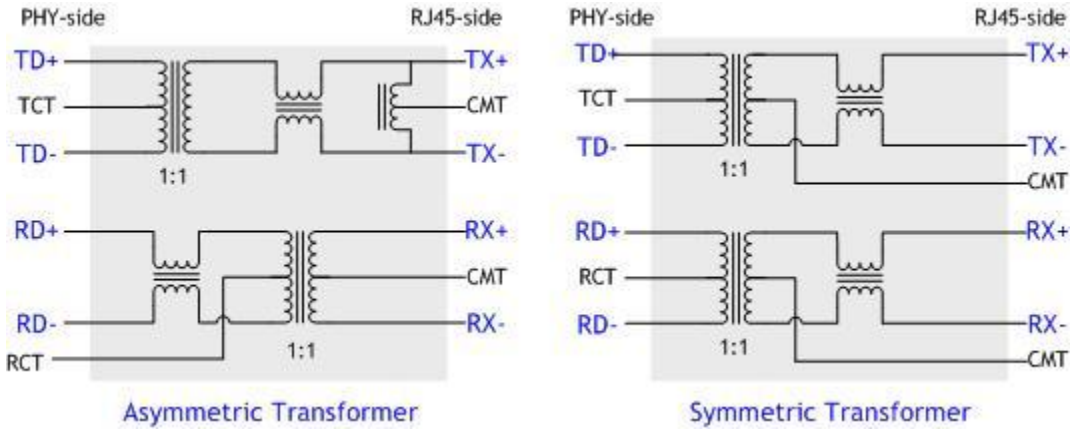


Figure 22. Transformer Type

5.5.6 MDIX

W5500 does not support auto-MDIX feature.

Thus, you should use straight-through cables to connect to other switches or routers and crossover cables to connect to devices such as servers, workstations or another W5500.

And you can use either type of cable to connect to other devices with auto-MDIX enabled, and the interface automatically corrects for any incorrect cabling.

6 Package Descriptions

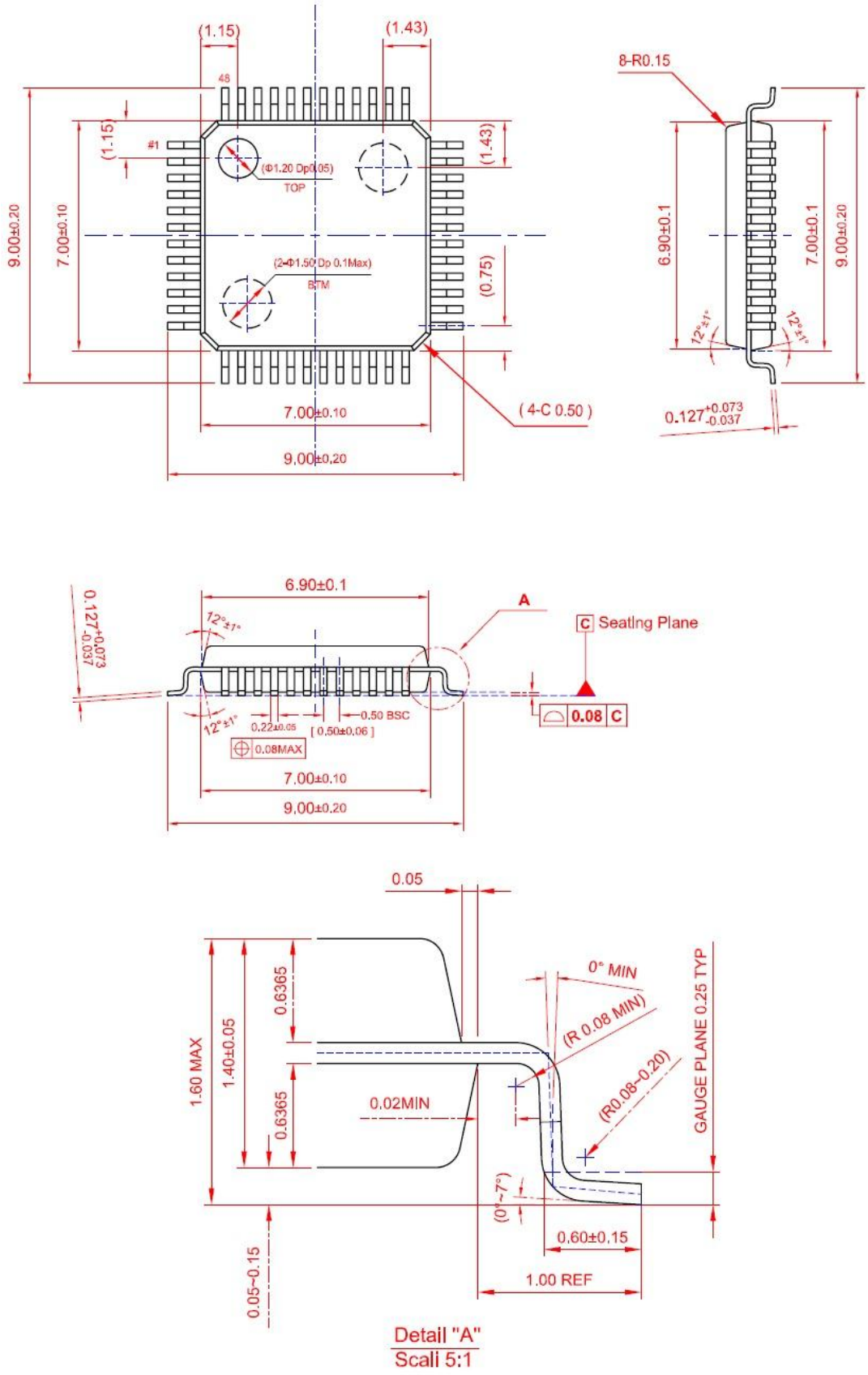


Figure 23. Package Dimensions

Note

1. These dimensions do not include mold protrusion.
2. ( ) is reference.
3. [ ] is ass'y out quality.
4. UNIT : mm