

iEthernet W5200

Datasheet

Version 1.2.7



<http://www.wiznet.co.kr>

W5200

The W5200 chip is a Hardwired TCP/IP embedded Ethernet controller that enables easier internet connection for embedded systems using SPI (Serial Peripheral Interface). W5200 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.

The W5200 is composed of a fully hardwired market-proven TCP/IP stack and an integrated Ethernet MAC & PHY. Hardwired TCP/IP stack supports TCP, UDP, IPv4, ICMP, ARP, IGMP, and PPPoE, which has been proven in various applications for many years. W5200 uses a 32Kbytes internal buffer as its data communication memory. By using W5200, users can implement the Ethernet application they need by using a simple socket program instead of handling a complex Ethernet Controller.

SPI (Serial Peripheral Interface) is provided for easy integration with the external MCU. The W5200 SPI supports a high speed SPI capable of communicating over SPI at up to 80MHz. In order to reduce power consumption of the system, W5200 provides WOL (Wake on LAN) and power down mode. To wake up during WOL, W5200 should be received magic packet, which is the Raw Ethernet packet.

Features

- Support Hardwired TCP/IP Protocols : TCP, UDP, ICMP, IPv4 ARP, IGMP, PPPoE, Ethernet
- Supports 8 independent sockets simultaneously
- Very small 48 Pin QFN Package
- Support Power down mode
- Support Wake on LAN
- 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)
- Support Auto MDI/MDIX
- Support ADSL connection (with support PPPoE Protocol with PAP/CHAP Authentication mode)
- Not support IP Fragmentation
- 3.3V operation with 5V I/O signal tolerance
- Lead-Free Package
- Multi-function LED outputs (Full/Half duplex, Link, Speed)

Target Applications

The W5200 is well suited for many embedded applications, including:

- 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

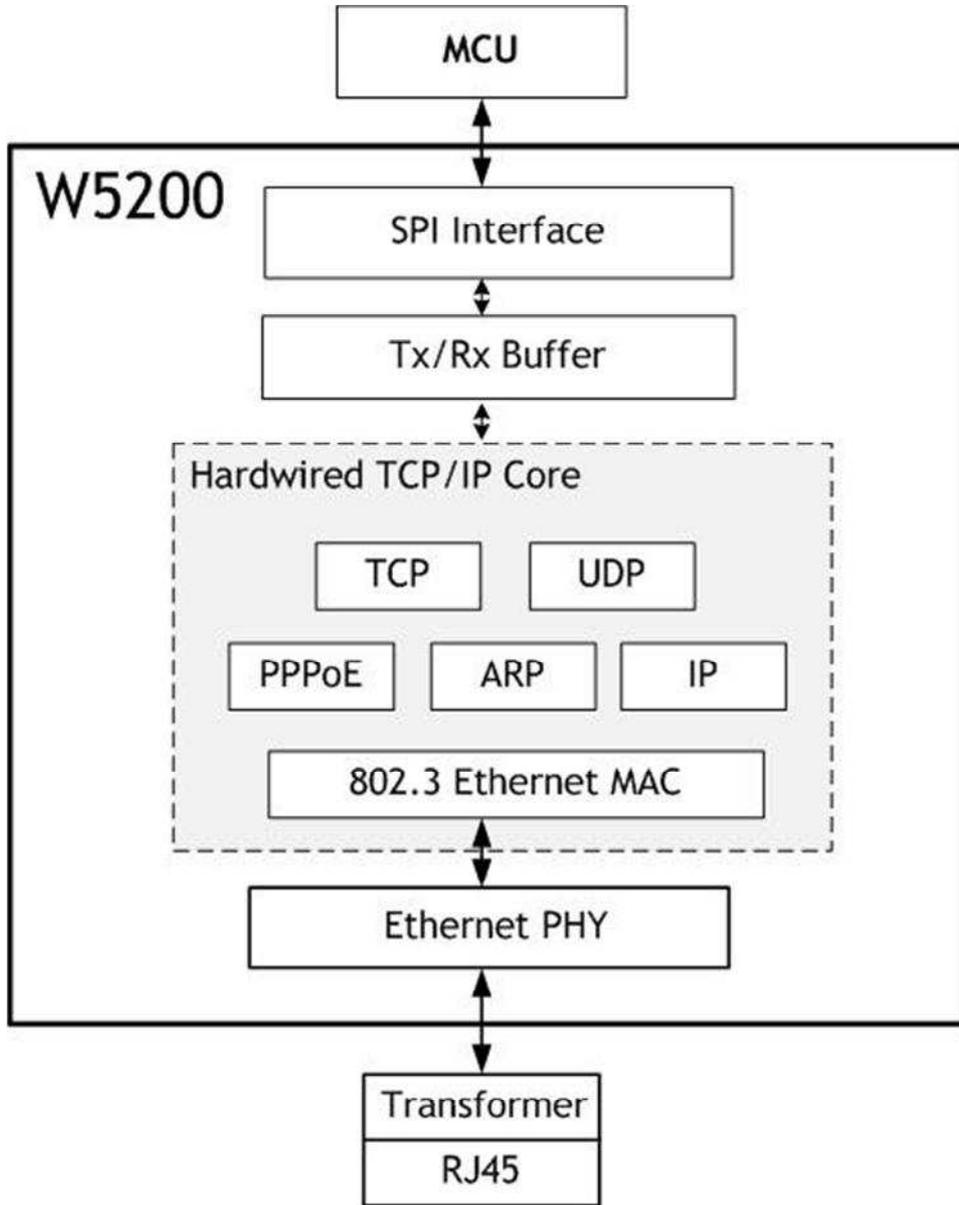


Table of Contents

1	Pin Assignment.....	8
1.1	MCU Interface Signals	8
1.2	1.2 PHY Signals	9
1.3	Miscellaneous Signals	10
1.4	Power Supply Signals	10
1.5	Clock Signals	12
1.6	LED Signals	12
2	Memory Map	13
3	W5200 Registers.....	14
3.1	common registers	14
3.2	Socket registers	15
4	Register Descriptions	16
4.1	Common Registers	16
4.2	Socket Registers	23
5	Functional Descriptions	41
5.1	Initialization	41
5.2	Data Communications	44
5.2.1	TCP	44
5.2.1.1	TCP SERVER.....	45
5.2.1.2	TCP CLIENT	52
5.2.2	UDP	53
5.2.2.1	Unicast and Broadcast.....	53
5.2.2.2	Multicast	60
5.2.3	IPRAW.....	63
5.2.4	MACRAW.....	65
6	External Interface	71
6.1	SPI (Serial Peripheral Interface) mode	71
6.2	Device Operations.....	71
6.3	Process of using general SPI Master device	72
7	Electrical Specifications	77
7.1	Absolute Maximum Ratings	77
7.2	DC Characteristics	77
7.3	POWER DISSIPATION(Vcc 3.3V Temperature 25° C)	77
7.4	AC Characteristics	78
7.4.1	Reset Timing	78
7.4.2	Crystal Characteristics	78
7.4.3	SPI Timing.....	79

7.4.4	Transformer Characteristics	80
8	IR Reflow Temperature Profile (Lead-Free)	81
9	Package Descriptions	82
	Document History Information	84

Table of Figure

Figure 1 Pin Description W5200	8
Figure 2 XTAL_VDD Reference Schematic	11
Figure 3 Power Design	11
Figure 4 Crystal Reference Schematic	12
Figure 5 INTLEVEL Timing	20
Figure 6 Socket Status Transition	31
Figure 7 Physical Address Calculation	37
Figure 8 Allocation Internal TX/RX memory of Socket n-th	43
Figure 9 TCP SERVER and TCP CLIENT	44
Figure 10 TCP SERVER Operation Flow	45
Figure 11 TCP CLIENT Operation Flow	52
Figure 12 UDP Operation Flow.....	53
Figure 13 The Received UDP data Format	55
Figure 14 IPRAW Operation Flow	63
Figure 15 The receive IPRAW data Format	64
Figure 16 MACRAW Operation Flow	65
Figure 17 The received MACRAW data Format.....	66
Figure 18 SPI Interface.....	71
Figure 19 W5200 SPI Frame Format	72
Figure 20 Address and OP/DATA Length Sequence Diagram	72
Figure 21 READ Sequence	73
Figure 22 Write Sequence	75
Figure 23 Reset Timing.....	78
Figure 24 SPI Timing.....	79
Figure 25 Transformer Type	80
Figure 26 IR Reflow Temperature	81
Figure 27 Package Dimensions.....	82

1 Pin Assignment

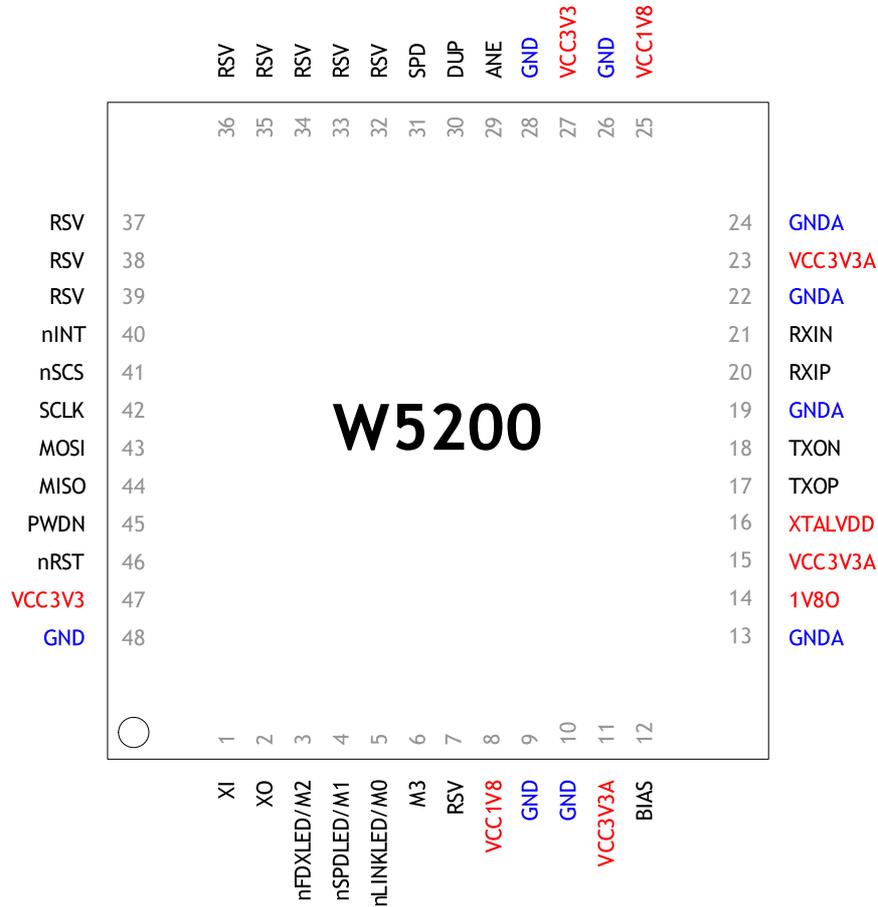


Figure 1 Pin Description W5200

1.1 MCU Interface Signals

Symbol	Type	Pin No	Description
nRST	I	46	RESET (Active LOW) This pin is active Low input to initialize or re-initialize W5200. RESET should be held at least 2us after low assert, and wait for at least 150ms after high de-assert in order for PLL logic to be stable. Refer to RESET timing of “7 Electrical Specification”
nSCS	I	41	SPI SLAVE SELECT (Active LOW) This pin is used to SPI Slave Select signal Pin when using SPI interface.

nINT	O	40	<p>INTERRUPT (Active LOW)</p> <p>This pin indicates that W5200 requires MCU attention after socket connecting, disconnecting, data receiving timeout, and WOL (Wake on LAN). The interrupt is cleared by writing IR(Interrupt Register) or Sn_IR (Socket n-th Interrupt Register). All interrupts are maskable. This pin is active low.</p>
SCLK	I	42	<p>SPI CLOCK</p> <p>This pin is used to SPI Clock signal Pin when using SPI interface.</p>
MOSI	I	43	<p>SPI MASTER OUT SLAVE IN</p> <p>This pin is used to SPI MOSI signal pin when using SPI interface.</p>
MISO	O	44	<p>SPI MASTER IN SLAVE OUT</p> <p>This pin is used to SPI MISO signal pin.</p>
PWDN	I	45	<p>POWER DOWN (Active HIGH)</p> <p>This pin is used to power down pin.</p> <p>Low : Normal Mode Enable</p> <p>High : Power Down Mode Enable</p>

1.2 PHY Signals

Symbol	Type	Pin No	Description
RXIP	I	20	RXIP/RXIN Signal Pair The differential data from the media is received on the RXIP/RXIN signal pair.
RXIN	I	21	
TXOP	O	17	TXOP/TXON Signal Pair The differential data is transmitted to the media on the TXOP/TXIN signal pair.
TXON	O	18	
BIAS	O	12	BIAS Register Connect a resistor of 28.7kΩ±1% to the ground. Refer to the “Reference schematic”.
ANE	I	29	Auto Negotiation Mode Enable This pin selects Enable/Disable of Auto Negotiation Mode. Low :Auto Negotiation Mode Disable High : Auto Negotiation Mode Enable
DUP	I	30	Full Duplex Mode Enable This pin selects Enable/Disable of Full Duplex Mode.

			Low = Half Duplex Mode Enable High = Full Duplex Mode Enable This function activates only during reset period.
SPD	I	31	Speed Mode This pin selects 100M/10M Speed Mode. Low = 10M Speed Mode High = 100M Speed Mode This function activates only during reset period.

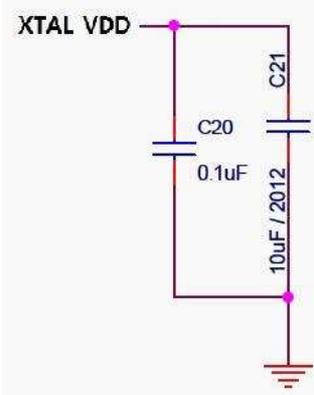
1.3 Miscellaneous Signals

Symbol	Type	Pin No	Description
nFDXLED/M2 nSPDLED/M1 nLINKLED/M0	I	3, 4, 5	W5200 MODE SELECT Normal mode : 111 Other test modes are internal test mode. This function activates only during reset period
M3	I	6	This pin should be pull-up.
RSV	-	7,32,33,34,35,36, 37,38,39	Reserved Pin The pin number 7 should be pull-up. The reserved pins except the pin number 7 should be pull-down or GND.

- Notes: Pull-Up/Down resistor = 40K Ω to 100K Ω . Typical value are 75K Ω .

1.4 Power Supply Signals

Symbol	Type	Pin No	Description
VCC3V3A	Power	11, 15, 23	3.3V power supply for Analog part
VCC3V3	Power	27, 47	3.3V power supply for Digital part
VCC1V8	Power	8, 25	1.8V power supply for Digital part
GND A	Ground	13, 19, 22, 24	Analog ground
GND	Ground	9, 10, 26, 28, 48	Digital ground
1V80	O	14	1.8V regulator output voltage 1.8V/200mA power created by internal power regulator, is used for core operation power (VCC1V8). Be sure to connect tantalum capacitor between 1V80 and GND for output frequency compensation, and selectively connect 0.1 μ F capacitor for high frequency noise decoupling.

			Notice: 1V80 is the power for W5200 core operation. It should not be connected to the power of other devices.
XTALVDD	I	16	 <p>Figure 2 XTAL_VDD Reference Schematic Connect a capacitor of 10.1uF to the ground. ※ Refer to the 'W5200E01-M3 Reference schematic</p>

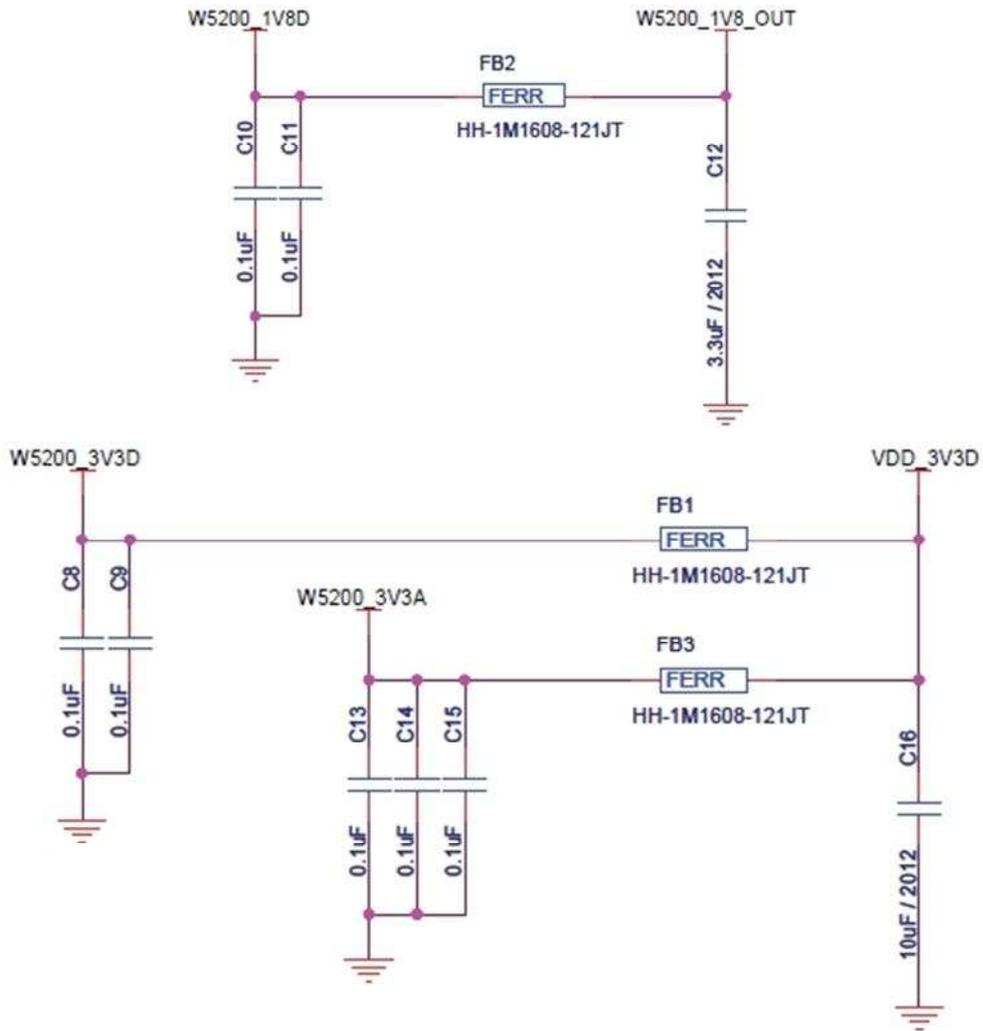


Figure 3 Power Design

Recommend for power design.

1. Locate decoupling capacitor as close as possible to W5200.
2. Use ground plane as wide as possible.
3. If ground plane width is adequate, having a separate analog ground plane and digital ground plane is good practice.
4. If ground plane is not wide, design analog and digital ground planes as a single ground plane, rather than separate them.

1.5 Clock Signals

Symbol	Type	Pin No	Description
XI	I	1	25MHz crystal input/output. A 25MHz crystal and Oscillator is used to connect these pins. <div style="text-align: center;"> </div>
XO	O	2	

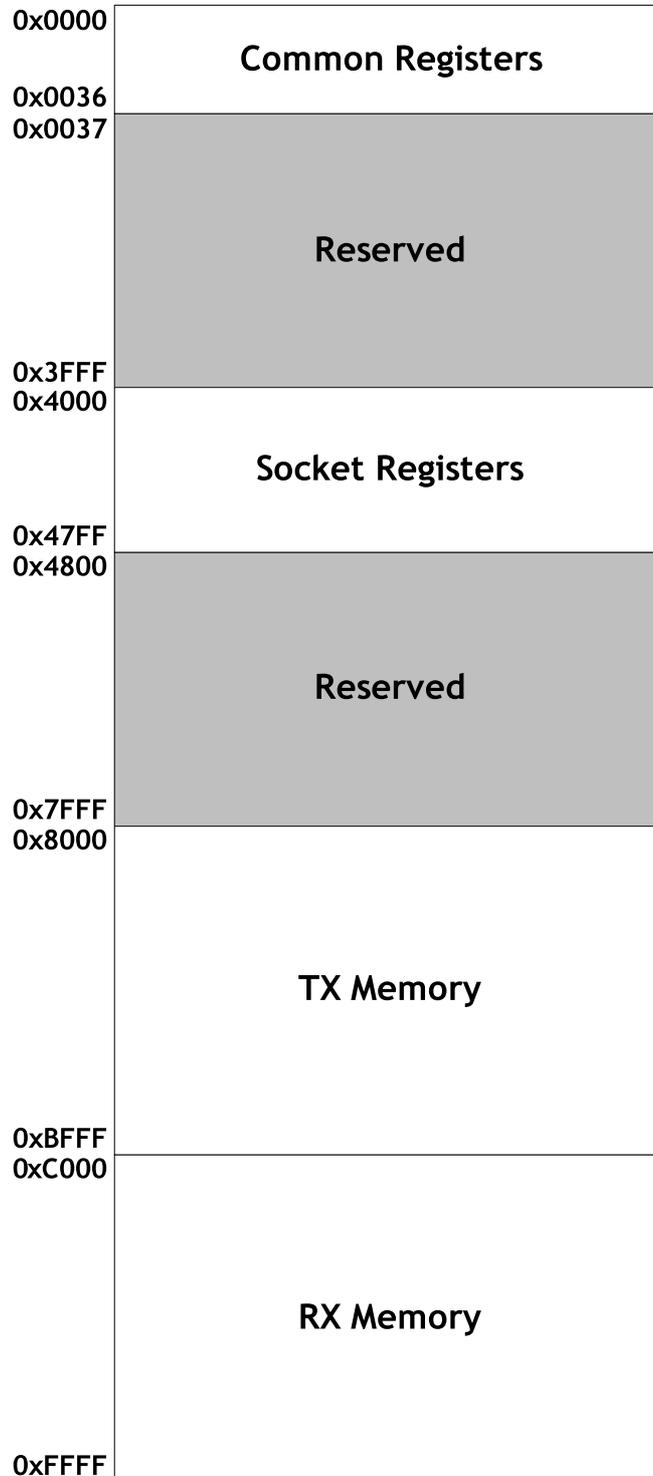
Figure 4 Crystal Reference Schematic

1.6 LED Signals

Symbol	Type	Pin No	Description
nFDXLED/M2	O	3	Full Duplex/Collision LED Low: Full-duplex High: Half-duplex.
nSPDLED/M1	O	4	Link speed LED Low: 100Mbps High: 10Mbps
nLINKLED/M0	O	5	Link LED Low: Link (10/100M) High: Un-Link blink: TX or RX state on Link

2 Memory Map

W5200 is composed of Common Register, Socket Register, TX Memory, and RX Memory as shown below.



W5200 Memory Map

3 W5200 Registers

3.1 common registers

Address	Register	Address	Register
0x0000	Mode (MR)		Authentication Type in PPPoE
0x0001	Gateway Address (GAR0)	0x001C	(PATR0)
0x0002	(GAR1)	0x001D	(PATR1)
0x0003	(GAR2)		Authentication Algorithm in PPPoE
0x0004	(GAR3)	0x001E	(PPPALGO)
	Subnet mask Address	0x001F	Chip version(VERSIONR)
0x0005	(SUBR0)	0x0020	Reserved
0x0006	(SUBR1)	~	
0x0007	(SUBR2)	0x0027	
0x0008	(SUBR3)		PPP LCP RequestTimer (PTIMER)
	Source Hardware Address	0x0028	Reserved
0x0009	(SHAR0)	~	
0x000A	(SHAR1)	0x0029	PPP LCP Magic number (PMAGIC)
0x000B	(SHAR2)		Reserved
0x000C	(SHAR3)	0x002A	
0x000D	(SHAR4)	~	
0x000E	(SHAR5)	0x002F	Interrupt Low Level Timer (INTLEVEL0) (INTLEVEL1)
	Source IP Address		Reserved
0x000F	(SIPR0)	0x0032	
0x0010	(SIPR1)	~	Reserved
0x0011	(SIPR2)	0x0033	
0x0012	(SIPR3)	0x0034	Socket Interrupt (IR2)
0x0013	Reserved	0x0035	PHY Status(PSTATUS)
0x0014		0x0036	Socket Interrupt Mask (IMR2)
0x0015	Interrupt (IR)		
0x0016	Interrupt Mask (IMR)		
	Retry Time		
0x0017	(RTR0)		
0x0018	(RTR1)		
0x0019	Retry Count (RCR)		
0x001A	Reserved		
0x001B			

3.2 Socket registers

Note : n is socket number (0, 1, 2, 3, 4, 5, 6, 7)

Address	Register	Address	Register
0x4n00	Socket n Mode (Sn_MR)		Receive Memory Size
0x4n01	Socket n Command (Sn_CR)	0x4n1E	(Sn_RXMEM_SIZE)
0x4n02	Socket n Interrupt (Sn_IR)		Transmit Memory Size
0x4n03	Socket n Status (Sn_SR)	0x4n1F	(Sn_TXMEM_SIZE)
	Socket n SourcePort		Socket 0 TX Free Size
0x4n04	(SN_PORT0)	0x4n20	(Sn_TX_FSR0)
0x4n05	(SN_PORT1)	0x4n21	(Sn_TX_FSR1)
	Socket n Destination Hardware		Socket 0 TX Read Pointer
	Address	0x4n22	(Sn_TX_RD0)
0x4n06	(Sn_DHAR0)	0x4n23	(Sn_TX_RD1)
0x4n07	(Sn_DHAR1)		Socket 0 TX Write Pointer
0x4n08	(Sn_DHAR2)	0x4n24	(Sn_TX_WR0)
0x4n09	(Sn_DHAR3)	0x4n25	(Sn_TX_WR1)
0x4n0A	(Sn_DHAR4)		Socket 0 RX Received Size
0x4n0B	(Sn_DHAR5)	0x4n26	(Sn_RX_RSR0)
	Socket 0 Destination IP Address	0x4n27	(Sn_RX_RSR1)
0x4n0C	(Sn_DIPR0)		Socket 0 RX Read Pointer
0x4n0D	(Sn_DIPR1)	0x4n28	(Sn_RX_RD0)
0x4n0E	(Sn_DIPR2)	0x4n29	(Sn_RX_RD1)
0x4n0F	(Sn_DIPR3)		Socket 0 RX Write Pointer
	Socket 0 Destination Port	0x4n2A	(Sn_RX_WR0)
0x4n10	(Sn_DPORT0)	0x4n2B	(Sn_RX_WR1)
0x4n11	(Sn_DPORT1)		Socket Interrupt Mask
	Socket 0 Maximum Segment Size	0x4n2C	(Sn_IMR)
0x4n12	(Sn_MSSR0)		Fragment Offset in IP header
0x4n13	(Sn_MSSR1)	0x4n2D	(Sn_FRAG0)
	Socket 0 Protocol in IP Raw mode	0x4n2E	(Sn_FRAG1)
0x4n14	(Sn_PROTO)	0x4n30	Reserved
0x4n15	Socket n IP TOS (Sn_TOS)	~	
0x4n16	Socket n IP TTL (Sn_TTL)	0x4nFF	
0x4n17			
~	Reserved		
0x4n1D			

4 Register Descriptions

4.1 Common Registers

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

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

7	6	5	4	3	2	1	0
RST			PB	PPPoE			

Bit	Symbol	Description
7	RST	S/W Reset If this bit is '1', internal register will be initialized. It will be automatically cleared after reset.
6	Reserved	Reserved
5	Reserved	Reserved
4	PB	Ping Block Mode 0 : Disable Ping block 1 : Enable Ping block If the bit is set as '1', there is no response to the ping request.
3	PPPoE	PPPoE Mode 0 : Disable PPPoE mode 1 : Enable PPPoE mode If you use ADSL without router or etc, you should set the bit as '1' to connect to ADSL Server. For more detail, refer to the application note, " <i>How to connect ADSL</i> ".
2	Reserved	Reserved
1	Reserved	Reserved
0	Reserved	Reserved

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

This Register sets up 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]

This register sets up 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]

This register sets up 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]

This register sets up the Source IP address.

Ex) In case of "192.168.0.2"

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

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

This register is accessed by the host processor to know the cause of interrupt. Any interruption can be masked in the Interrupt Mask Register (IMR). The nINT signal retain low as long as any masked signal is set, and will not go high until all masked bits in this Register have been cleared.

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

Bit	Symbol	Description
7	CONFLICT	IP Conflict It is set as '1' when there is ARP request with same IP address as Source IP address. This bit is cleared to '0' by writing '1' to this bit.
6	Reserved	Reserved
5	PPPoE	PPPoE Connection Close In the Point-to-Point Protocol over Ethernet (PPPoE) Mode, if the PPPoE connection is closed, '1' is set. This bit will be cleared to '0' by writing '1' to this bit.
4	Reserved	Reserved
3	Reserved	Reserved
2	Reserved	Reserved
1	Reserved	Reserved
0	Reserved	Reserved

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

The Interrupt Mask Register is used to mask interrupts. Each interrupt mask bit corresponds to a bit in the Interrupt Register2 (IR2). If an interrupt mask bit is set, an interruption will be issued whenever the corresponding bit in the IR2 is set. If any bit in the IMR is set as '0' an interrupt will not occur though the bit.

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	S7_INT	IR(S7_INT) Interrupt Mask
6	S6_INT	IR(S6_INT) Interrupt Mask
5	S5_INT	IR(S5_INT) Interrupt Mask
4	S4_INT	IR(S4_INT) Interrupt Mask
3	S3_INT	IR(S3_INT) Interrupt Mask
2	S2_INT	IR(S2_INT) Interrupt Mask
1	S1_INT	IR(S1_INT) Interrupt Mask
0	S0_INT	IR(S0_INT) Interrupt Mask

RTR (Retry Time-value Register) [R/W] [0x0017 - 0x0018] [0x07D0]

It configures the retransmission timeout-period. The standard unit of RTR is 100us. RTR is initialized with 2000(0x07D0) and has 200ms timeout-period.

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

0x0017	0x0018
0x0F	0xA0

Re-transmission will occur if there is no response from the remote peer to the commands of CONNECT, DISCON, CLOSE, SEND, SEND_MAC and SEND_KEEP, or the response is delayed.

RCR (Retry Count Register) [R/W] [0x0019] [0x08]

It configures the number of retransmission times. When retransmission occurs as many as 'RCR+1' times, Timeout interrupt is set ('TIMEOUT' bit of Sn_IR is set as '1').

In case of using TCP communication, the value of Sn_SR (Socket n-th Status Register) is changed to 'SOCK_CLOSED' and Sn_IR(Socket n-th Status Register) (TIMEOUT) turns into '1'. In case of not using TCP communication, only Sn_IR(TIMEOUT) turns into '1'.

Ex) RCR = 0x0007

0x0019
0x07

The timeout of W5200 can be configurable with RTR and RCR. W5200's timeout has Address

Resolution Protocol (ARP) and TCP retransmission timeout.

At the ARP (Refer to RFC 826, <http://www.ietf.org/rfc.html>) retransmission timeout, W5200 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). As waiting for ARP-response from the peer, if there is no response during the time set in RTR, Timeout occurs and ARP-request is retransmitted. It is repeated as many as 'RCR + 1' times.

Even after ARP-request retransmissions are repeated 'RCR + 1' times, if there is no ARP-response, the final timeout occurs and Sn_IR(TIMEOUT) becomes '1'.

The value of final timeout (ARP_{TO}) of ARP-request is as below.

$$\text{ARP}_{\text{TO}} = (\text{RTR} \times 0.1\text{ms}) \times (\text{RCR} + 1)$$

At the TCP packet retransmission timeout, W5200 transmits TCP packets (SYN, FIN, RST, DATA packets) and waits for the acknowledgement (ACK) during the time set in RTR and RCR. If there is no ACK from the peer, Timeout occurs and TCP packets (sent earlier) are retransmitted. The retransmissions are repeated as many as 'RCR + 1' times. Even after TCP packet retransmissions are repeated 'RCR + 1' times, if there is no ACK from the peer, final timeout occurs and Sn_SR is changed to 'SOCK_CLOSED' at the same time with Sn_IR(TIMEOUT) = '1'

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

N : Retransmission count, 0 ≤ N ≤ M
 M : Minimum value when $\text{RTR} \times 2^{(M+1)} > 65535$ and 0 ≤ M ≤ RCR
 RTR_{MAX}: $\text{RTR} \times 2^M$

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

$$\text{ARP}_{\text{TO}} = 2000 \times 0.1\text{ms} \times 9 = 1800\text{ms} = 1.8\text{s}$$

$$\begin{aligned} \text{TCP}_{\text{TO}} &= (0x07D0 + 0x0FA0 + 0x1F40 + 0x3E80 + 0x7D00 + 0xFA00 + 0xFA00 + 0xFA00 + 0xFA00) \times 0.1\text{ms} \\ &= (2000 + 4000 + 8000 + 16000 + 32000 + ((8 - 4) \times 64000)) \times 0.1\text{ms} \\ &= 318000 \times 0.1\text{ms} = 31.8\text{s} \end{aligned}$$

The value of final timeout (TCP_{TO}) of TCP packet retransmission can be calculated as below,

PATR (Authentication Type in PPPoE mode) [R] [0x001C-0x001D] [0x0000]

This register notifies authentication method that has been agreed at the connection with PPPoE Server. W5200 supports two types of Authentication method - PAP and CHAP.

Value	Authentication Type
0xC023	PAP
0xC223	CHAP

PPPALGO(Authentication Algorithm in PPPoE mode)[R][0x001E][0x00]

This register notifies authentication algorithm in PPPoE mode. For detailed information, please refer to PPPoE application note.

VERSIONR (W5200 Chip Version Register)[R][0x001F][0x03]

This register is the W5200 chip version register.

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

This register indicates the duration for sending LCP Echo Request. Value 1 is about 25ms.

Ex) in case that PTIMER is 200,
 $200 * 25(\text{ms}) = 5000(\text{ms}) = 5 \text{ seconds}$

PMAGIC (PPP Link Control Protocol Magic number Register) [R/W] [0x0029][0x00]

This register is used in Magic number option during LCP negotiation. Refer to the application note, "How to connect ADSL".

INTLEVEL (Interrupt Low Level Timer Register)[R/W][0x0030 - 0x0031][0x0000]

It sets Interrupt Assert wait time (I_{AWT}). It configures nINT Low Assert waiting time until the next interrupt.

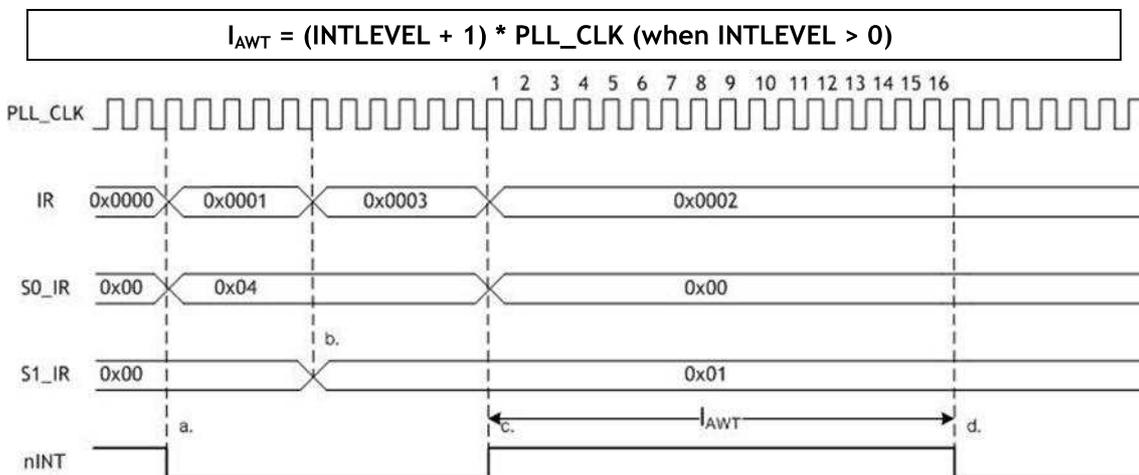


Figure 5 INTLEVEL Timing

- a. At SOCKET 0, Receive Interrupt occurs ($S0_IR(3) = '1'$) and corresponding IR2 bit is set as '1' ($IR(S0_IR) = '1'$). nINT signal is asserted low.
- b. At SOCKET 1, Connected Interrupt occurs ($S1_IR(0) = '1'$) and corresponding IR2 bit set as '1' ($IR(S1_IR) = '1'$).
- c. The Host clears $S0_IR1$ ($S0_IR = 0x00$) and corresponding IR bit is automatically cleared ($IR(S0_IR) = '0'$). nINT signal becomes High.
- d. $S0_IR$ is cleared. As IR2 is not 0x00, nINT should be asserted low right after 1PLL_CLK. However, as INTLEVEL is 0x000F, the interrupt about IR is processed after $I_{AWT}(16 \text{ PLL_CLK})$.

IR2(W5200 SOCKET Interrupt Register)[R/W][0x0034][0x00]

IR2 is the Register to notify W5200 SOCKET interrupt to the Host. If any interrupt occurs, the related bit of IR2 is set as '1'. When related Mask Bit is '1', nINT signal is asserted low. nINT keeps low until all bits of Sn_IR becomes '0'. If all bits of Sn_IR become '0', it becomes high automatically.

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	S7_INT	When an interrupt occurs at SOCKET 7-th, it becomes '1'. This interrupt information is applied to S7_IR. This bit is automatically cleared when S7_IR is cleared to 0x00 by host.
6	S6_INT	When an interrupt occurs at SOCKET 6-th, it becomes '1'. This interrupt information is applied to S6_IR. This bit is automatically cleared when S6_IR is cleared to 0x00 by host.
5	S5_INT	When an interrupt occurs at SOCKET 5-th, it becomes '1'. This interrupt information is applied to S5_IR. This bit is automatically cleared when S5_IR is cleared to 0x00 by host.
4	S4_INT	When an interrupt occurs at SOCKET 4-th, it becomes '1'. This interrupt information is applied to S4_IR. This bit is automatically cleared when S4_IR is cleared to 0x00 by host.
3	S3_INT	When an interrupt occurs at SOCKET 3-th, it becomes '1'. This interrupt information is applied to S3_IR. This bit is automatically cleared when S3_IR is cleared to 0x00 by host.
2	S2_INT	When an interrupt occurs at SOCKET 2-th, it becomes '1'. This interrupt information is applied to S2_IR. This bit is automatically cleared when S2_IR is cleared to 0x00 by host.
1	S1_INT	When an interrupt occurs at SOCKET 1-th, it becomes '1'. This interrupt information is applied to S1_IR. This bit is automatically cleared when S1_IR is cleared to 0x00 by host.
0	S0_INT	When an interrupt occurs at SOCKET 0-th, it becomes '0'. This interrupt information is applied to S0_IR. This bit is automatically cleared when S0_IR is cleared to 0x00 by host.

PHYSTATUS(W5200 PHY status Register)[R/W][0x0035][0x00]

PHYSTATUS is the Register to indicate W5200 status of PHY.

Bit	Symbol	Description
7	Reserved	Reserved
6	Reserved	Reserved
5	LINK	Link Status Register[Read Only] This register indicates Link status. 0 : Link down 1 : Link Up
4	Reserved	Reserved
3	POWERDOWN	Power down mode of PHY[Read Only] This register indicates status of Power down mode 0 : Disable Power down mode(operates normal mode) 1 : Enable Power down mode
2	Reserved	Reserved
1	Reserved	Reserved
0	Reserved	Reserved

IMR2 (Socket Interrupt Mask Register2) [R/W] [0x0036] [0x00]

The IMR2(Socket Interrupt Mask Register) is used to mask interrupts. Each interrupt mask bit corresponds to a bit in the Interrupt Register (IR). If an interrupt mask bit is set, an interrupt will be issued whenever the corresponding bit in the IR is set. If any bit in the IMR2 is set as '0', an interrupt will not occur though the bit in the IR is set.

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

Bit	Symbol	Description
7	IM_IR7	IP Conflict Enable
6	Reserved	Reserved
5	IM_IR5	PPPoE Close Enable
4	Reserved	Reserved
3	Reserved	Reserved
2	Reserved	Reserved
1	Reserved	Reserved
0	Reserved	Reserved

4.2 Socket Registers

Sn^1_MR (Socket n-th-th Mode Register) [R/W] [0x4000+0x0n00] [0x00]²

This register sets up socket option or protocol type for each socket.

7	6	5	4	3	2	1	0
MULTI		ND / MC		P3	P2	P1	P0

Bit	Symbol	Description
7	MULTI	Multicasting 0 : disable Multicasting 1 : enable Multicasting It is applied only in case of UDP. For using multicasting, write multicast group address to Socket n-th Destination IP and multicast group port number to Socket n-th Destination Port Register, before OPEN command.
6	MF	MAC Filter 0 : Disable MAC filter 1 : Enable MAC filter It is used in MACRAW (P3-P0: "0100"). When this bit is set as '1', W5200 can receive packet that is belong in itself or broadcasting. When this bit is set as '0', W5200 can receive all packets on Ethernet. When using the hybrid TCP/IP stack, it is recommended to be set as '1' for reducing the receiving overhead of host.
5	ND/MC	Use No Delayed ACK 0 : Disable No Delayed ACK option 1 : Enable No Delayed ACK option, This only applies to TCP case (P3-P0 : "0001") If this bit is set as '1', ACK packet is immediately transmitted after receiving data packet from a peer. If this bit is cleared, ACK packet is transmitted according to internal timeout mechanism. Multicast 0 : using IGMP version 2 1 : using IGMP version 1 This bit is valid when MULTI bit is enabled and UDP mode is used (P3-P0 : "0010"). In addition, multicast can be used to send out the version number in IGMP messages such as Join/Leave/Report to multicast-group

¹*n* is Socket n-thumber (0, 1, 2, 3, 4, 5, 6, 7).

²[Read/Write] [address of socket 0, address of socket 1, address of socket 2, address of socket 3, address of socket 4, address of socket 5, address of socket 6, address of socket 7] [Reset value]

4	Reserved	Reserved																														
3	P3	Protocol Sets up corresponding socket as TCP, UDP, or IP RAW mode <table border="1" style="margin: 10px 0;"> <thead> <tr> <th>P</th> <th>P</th> <th>P</th> <th>P</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td></td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>Closed</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>TCP</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>UDP</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>IPRAW</td> </tr> </tbody> </table>	P	P	P	P	Meaning	3	2	1	0		0	0	0	0	Closed	0	0	0	1	TCP	0	0	1	0	UDP	0	0	1	1	IPRAW
P	P		P	P	Meaning																											
3	2		1	0																												
0	0		0	0	Closed																											
0	0	0	1	TCP																												
0	0	1	0	UDP																												
0	0	1	1	IPRAW																												
2	P2																															
1	P1																															
0	P0	<p>* In case of socket 0, MACRAW and PPPoE mode exist.</p> <table border="1" style="margin: 10px 0;"> <thead> <tr> <th>P</th> <th>P</th> <th>P</th> <th>P</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td></td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>MACRAW</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>PPPoE</td> </tr> </tbody> </table> <p>S0_MR_MACRAW and S0_MR_PPPoE are valid only in SOCKET 0. S0_MR_PPPoE is temporarily used for PPPoE server connection/Termination. After connection is established, it can be utilized as another protocol.</p>	P	P	P	P	Meaning	3	2	1	0		0	1	0	0	MACRAW	0	1	0	1	PPPoE										
P	P	P	P	Meaning																												
3	2	1	0																													
0	1	0	0	MACRAW																												
0	1	0	1	PPPoE																												

Sn_CR (Socket n-th Command Register) [R/W] [0x4001+0x0n00] [0x00]

This is used to set the command for Socket n-th such as OPEN, CLOSE, CONNECT, LISTEN, SEND, and RECEIVE. After W5200 identifies 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 verify whether the command is completed or not, please check the Sn_IR or Sn_SR registers.

Value	Symbol	Description														
0x01	OPEN	<p>Socket n-th 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</p> <table border="1" style="margin-left: 40px;"> <thead> <tr> <th>Sn_MR(P3:P0)</th> <th>Sn_SR</th> </tr> </thead> <tbody> <tr> <td>Sn_MR_CLOSE (0x00)</td> <td>-</td> </tr> <tr> <td>Sn_MR_TCP (0x01)</td> <td>SOCK_INIT (0x13)</td> </tr> <tr> <td>Sn_MR_UDP (0x02)</td> <td>SOCK_UDP (0x22)</td> </tr> <tr> <td>Sn_MR_IPRAW (0x03)</td> <td>SOCK_IPRAW (0x32)</td> </tr> <tr> <td>SO_MR_MACRAW (0x04)</td> <td>SOCK_MACRAW (0x42)</td> </tr> <tr> <td>SO_MR_PPPOE (0x05)</td> <td>SOCK_PPPOE (0x5F)</td> </tr> </tbody> </table>	Sn_MR(P3:P0)	Sn_SR	Sn_MR_CLOSE (0x00)	-	Sn_MR_TCP (0x01)	SOCK_INIT (0x13)	Sn_MR_UDP (0x02)	SOCK_UDP (0x22)	Sn_MR_IPRAW (0x03)	SOCK_IPRAW (0x32)	SO_MR_MACRAW (0x04)	SOCK_MACRAW (0x42)	SO_MR_PPPOE (0x05)	SOCK_PPPOE (0x5F)
Sn_MR(P3:P0)	Sn_SR															
Sn_MR_CLOSE (0x00)	-															
Sn_MR_TCP (0x01)	SOCK_INIT (0x13)															
Sn_MR_UDP (0x02)	SOCK_UDP (0x22)															
Sn_MR_IPRAW (0x03)	SOCK_IPRAW (0x32)															
SO_MR_MACRAW (0x04)	SOCK_MACRAW (0x42)															
SO_MR_PPPOE (0x05)	SOCK_PPPOE (0x5F)															
0x02	LISTEN	<p>This is valid only in TCP mode (Sn_MR(P3:P0) = Sn_MR_TCP). In this mode, the Socket n-th is configured as a TCP server which is waiting for connection-request (SYN packet) from any "TCP CLIENT". The Sn_SR register changes the state from SOCK_INIT to SOCKET_LISTEN.</p> <p>When a client's connection request is successfully established, the Sn_SR changes from SOCK_LISTEN to SOCK_ESTABLISHED and the Sn_IR(0) becomes '1'. On the other hand, Sn_IR(3) is set as '1' and Sn_SR changes to SOCK_CLOSED during a connection failure(SYN/ACK packet failed to transfer)</p> <p>cf> If the destination port of the TCP Client does not exist during a connection request, W5200 will transmit a RST packet and Sn_SR is unchanged.</p>														
0x04	CONNECT	<p>This mode is only valid in TCP mode and operates the Socket n-th as a TCP client. A connect-request (SYN packet) is sent to the TCP server by connecting to the IP address and port stored in destination address and port registers (Sn_DIPR0 and Sn_DPORT0)</p> <p>When a client's connection request is successfully established, the Sn_SR register is changed to SOCK_ESTABLISHED and the Sn_IR(0) becomes '1'. In the following cases, the connect-request fails When a ARPTO occurs (Sn_IR(s)='1') because the Destination Hardware Address is not acquired through the ARP process</p>														

		<p>When a SYN/ACK packet is not received and TCPTO(Sn_IR(3)) is '1'</p> <p>When a RST packet is received instead of a SYN/ACK packet</p> <p>Above three cases, Sn_SR is changed to SOCK_CLOSED.</p>
0x08	DISCON	<p>Only valid in TCP mode</p> <p>Regardless of "TCP SERVER" or "TCP CLIENT", this disconnect command processes the</p> <p>Active close: it transmits disconnect-request(FIN packet) to the connected peer</p> <p>Passive close: When FIN packet is received from peer, a FIN packet is replied back to the peer</p> <p>when FIN/ACK packet is received, Sn_SR is changed to SOCK_CLOSED.</p> <p>When a disconnect request is not received, TCPTO occurs (Sn_IR(3)='1') and Sn_SR is changed to SOCK_CLOSED.</p> <p>cf> If CLOSE is used instead of DISCON, only Sn_SR is changed to SOCK_CLOSED without disconnect-process(disconnect-request). If a RST packet is received from a peer during communication, Sn_SR is unconditionally changed to SOCK_CLOSED.</p>
0x10	CLOSE	<p>Closes Socket n-th.</p> <p>Sn_SR is changed to SOCK_CLOSED.</p>
0x20	SEND	<p>SEND transmits all the data buffered in the TX memory. For more details, please refer to Socket n-th TX Free Size Register (Sn_TX_FSR0), Socket n-th TX Write Pointer Register(Sn_TX_WRO), and Socket n-th TX Read Pointer Register(Sn_TX_RD0).</p>
0x21	SEND_MAC	<p>Used in UDP mode only</p> <p>The basic operation is same as SEND. Normally SEND operation needs Destination Hardware Address which can be retrieved by the ARP (Address Resolution Protocol) process. SEND_MAC uses Socket n-th Destination Hardware Address(Sn_DHAR0) that is chosen by the user without going through the ARP process.</p>
0x22	SEND_KEEP	<p>Used in TCP mode</p> <p>It checks the connection status by sending 1byte data. If the connection has no response from peers or is terminated, the Timeout interrupt will occur.</p>
0x40	RECV	<p>RECV processes the data received by using a RX read pointer register(Sn_RX_RD).</p> <p>For more detail, please refer to 5.2.1.1 SERVER mode Receiving Process with Socket n-th RX Received Size Register (Sn_RX_RSRO), Socket n-th RX Write Pointer Register(Sn_RX_WR), and Socket n-th RX Read Pointer</p>

		Register(Sn_RX_RD).
--	--	---------------------

Below commands are only valid for SOCKET 0 and S0_MR(P3:P0) = S0_MR_PPpOE. For more detail refer to the “How to use ADSL”.

Value	Symbol	Description
0x23	PCON	PPPoE connection begins by transmitting PPPoE discovery packet
0x24	PDISCON	Closes PPPoE connection
0x25	PCR	In each phase, it transmits REQ message.
0x26	PCN	In each phase, it transmits NAK message.
0x27	PCJ	In each phase, it transmits REJECT message.

Sn_IR (Socket *n*-th Interrupt Register) [R] [0x4002+0x0n00] [0x00]

Sn_IR register provides information such as the type of interrupt (establishment, termination, receiving data, timeout) used in Socket *n*-th. When an interrupt occurs and the mask bit of Sn_IMR is '1', the interrupt bit of Sn_IR becomes '1'.

In order to clear the Sn_IR bit, the host should write the bit as '1'. When all the bits of Sn_IR is cleared ('0'), IR(*n*) is automatically cleared.

7	6	5	4	3	2	1	0
PRECV	PFAIL	PNEXT	SEND_OK	TIMEOUT	RECV	DISCON	CON

Bit	Symbol	Description
7	PRECV	Sn_IR(PRECV) Interrupt Mask Valid only in case of 'SOCKET=0' and 'SO_MR(P3:P0)=SO_MR_PPpPoE' PPP Receive Interrupts when the option which is not supported is received
6	PFAIL	Sn_IR(PFAIL) Interrupt Mask Valid only in case of 'SOCKET=0' & 'SO_MR(P3:P0)=SO_MR_PPpPoE' PPP Fail Interrupts when PAP Authentication is failed
5	PNEXT	Sn_IR(PNEXT) Interrupt Mask Valid only in case of 'SOCKET=0' & 'SO_MR(P3:P0)=SO_MR_PPpPoE' PPP Next Phase Interrupts when the phase is changed during ADSL connection process
4	SEND_OK	Sn_IR(SENDOK) Interrupt Mask SEND OK Interrupts when the SEND command is completed
3	TIMEOUT	Sn_IR(TIMEOUT) Interrupt Mask TIMEOUT Interrupts when ARP _{TO} or TCP _{TO} occurs
2	RECV	Sn_IR(RECV) Interrupt Mask Receive Interrupts whenever data packet is received from a peer
1	DISCON	Sn_IR(DISCON) Interrupt Mask Disconnect Interrupts when FIN of FIN/ACK packet is received from a peer
0	CON	Sn_IR(CON) Interrupt Mask Connect Interrupts when a connection is established with a peer

Sn_SR (Socket n-th Status Register) [R] [0x4003+0x0n00] [0x00]

This register provides the status of Socket *n*-th. SOCKET status are changed when using the Sn_CR register or during packet transmission/reception. The table below describes the different states of Socket *n*-th.

Value	Symbol	Description
0x00	SOCK_CLOSED	It is the status that resource of SOCKETn is released. When DISCON or CLOSE command is performed, or ARP _{TO} , or TCP _{TO} occurs, it is changed to SOCK_CLOSED regardless of previous value.
0x13	SOCK_INIT	It is shown in case that Sn_MR is set as TCP and OPEN commands are given to Sn_CR. It is changed to SOCK_INIT when Sn_MR (P3:P0) is Sn_MR_TCP and OPEN command is performed. It is the initial step of TCP connection establishment. It is possible to perform LISTEN command at the "TCP SERVER" mode and CONNECT command at the "TCP CLIENT". It is the status that SOCKETn operates as "TCP SERVER" and waits for connect-request (SYN packet) from "TCP CLIENT".
0x14	SOCK_LISTEN	Socket <i>n</i> -th operates in TCP Server Mode and waits for a connection-request (SYN packet) from a "TCP CLIENT". When the LISTEN command is used, the stage changes to SOCK_LISTEN Once the connection is established, the SOCKET state changes from SOCK_LISTEN to SOCK_ESTABLISHED; however, if the connection fails, TCP _{TO} occurs (Sn_IR(TIME_OUT) = '1') and the state changes to SOCK_CLOSED.
0x17	SOCK_ESTABLISHED	It is shown in case that connection is established. It is changed to SOCK_ESTABLISHED when SYN packet from "TCP CLIENT" is successfully processed at the SOCK_LISTEN, or CONNECTS command is successfully performed. At this status, DATA packet can be transferred, that is, SEND or RECV command can be performed.
0x1C	SOCK_CLOSE_WAIT	It is the status that disconnect-request (FIN packet) is received from the peer As TCP connection is half-closed, it is possible to transfer data packet. In order to complete the TCP disconnection, DISCON command should be performed. For SOCKETn close without disconnection-process, CLOSE command should be just performed.

0x22	SOCK_UDP	It is the status that SOCKETn is open as UDP mode. It is changed to SOCK_UDP when Sn_MR(P3:P0) is Sn_MR_UDP and OPEN command is performed. DATA packet can be transferred without connection that is necessary to TCP mode SOCKET.
0x32	SOCK_IPRAW	The socket is opened in IPRAW mode. The SOCKET status is change to SOCK_IPRAW when Sn_MR (P3:P0) is Sn_MR_IPRAW and OPEN command is used. IP Packet can be transferred without a connection similar to the UDP mode.
0x42	SOCK_MACRAW	It is changed to SOCK_MACRAW in case of S0_CR=OPEN and S0_MR (P3:P0)=S0_MR_MACRAW. MACRAW packet (Ethernet frame) can be transferred similar to UDP mode.
0x5F	SOCK_PPPOE	It is the status that SOCKET0 is open as PPPoE mode. It is changed to SOCK_PPPOE in case of S0_CR=OPEN and S0_MR (P3:P0)=S0_MR_PPPOE . It is temporarily used at the PPPoE connection.

Below is shown during changing the status.

Value	Symbol	Description
0x15	SOCK_SYNSENT	This status indicates that a connect-request (SYN packet) is sent to a "TCP SERVER". This status shows changing process from SOCK_INIT to SOCK_ESTABLISHED by CONNECT command. At this status, if connect-accept (SYN/ACK packet) is received from "TCP SERVER", it is automatically changed to SOCK_ ESTBLISHED. If SYN/ACK packet is not received from the "TCP SERVER" before TCPTO occurs (Sn_IR(TIMEOUT)='1'), it is changed to SOCK_CLOSED.
0x16	SOCK_SYNRECV	This status indicate that a connect-request(SYN packet) is received from a "TCP CLIENT". It is automatically changed to SOCK_ESTABLISHED when W5200 successfully transmits connect-accept (SYN/ACK packet) to the "TCP CLIENT". If it is failed, TCPTO occurs (Sn_IR(TIMEOUT)='1'), and it is changed to SOCK_CLOSED
0x18	SOCK_FIN_WAIT	These status shows that Socket n-th is closed. It is observed in the disconnect-process of active close or passive close. It is changed to SOCK_CLOSED, when disconnect-process is successfully finished or TCPTO occurs (Sn_IR (TIMEOUT)='1').
0x1A	SOCK_CLOSING	
0X1B	SOCK_TIME_WAIT	
0X1D	SOCK_LAST_ACK	
0x01	SOCK_ARP	This status indicates that ARP-request is transmitted in order to acquire destination hardware address. This status

is observed when SEND command is performed at the SOCK_UDP or SOCK_IPRAW, or CONNECT command is performed at the SOCK_INIT.

If hardware address is successfully acquired from destination (when ARP-response is received), it is changed to SOCK_UDP, SOCK_IPRAW or SOCK_SYNSENT. If it's failed and ARP_TO occurs (Sn_IR(TIMEOUT)='1'), in case of UDP or IPRAW mode it goes back to the previous status(the SOCK_UDP or SOCK_IPRAW), in case of TCP mode it goes to the SOCK_CLOSED.

cf> ARP-process operates at the SOCK_UDP or SOCK_IPRAW when the previous and current values of Sn_DIPR are different. If the previous and current values of Sn_DIPR are same, ARP-process doesn't operate because the destination hardware address is already acquired.

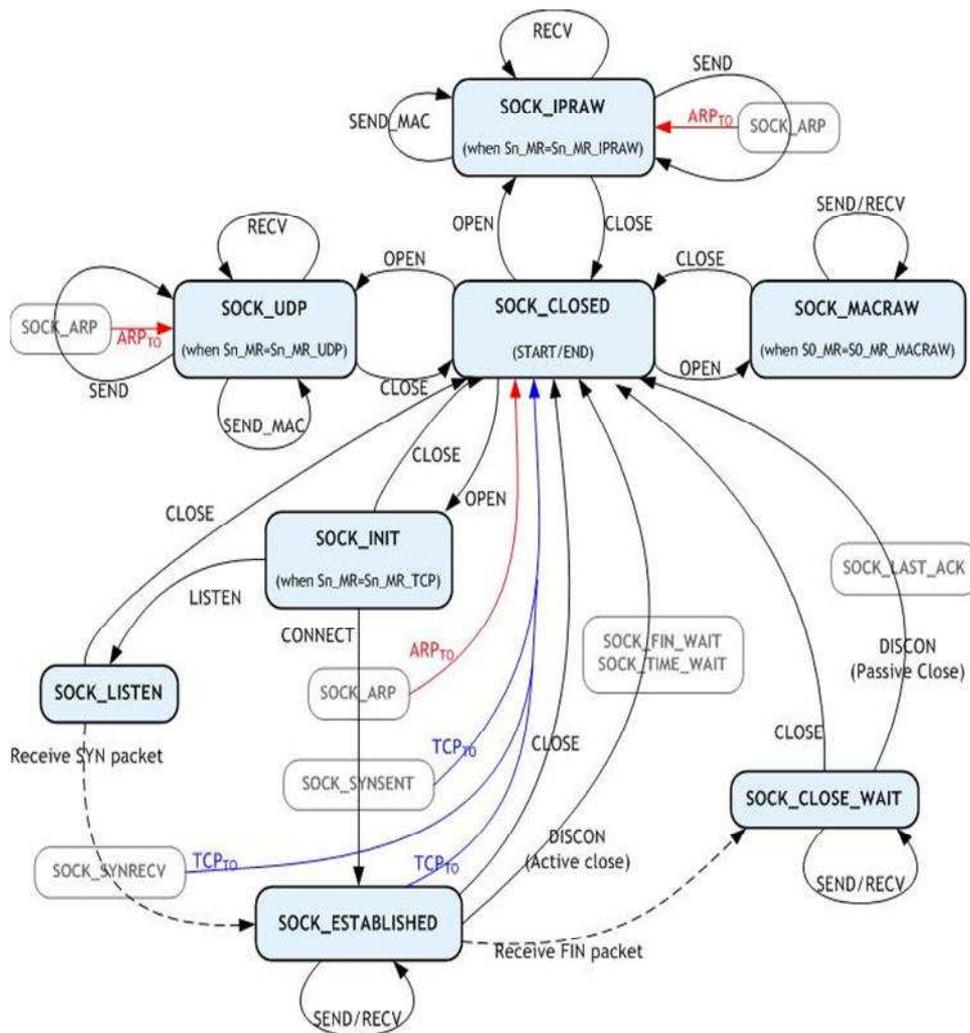


Figure 6 Socket Status Transition

Sn_PORT (Socket n-th Source Port Register) [R/W] [0x4004+0x0n00-0x4005+0x0n00] [0x0000]

This register sets the Source Port number for each Socket when using TCP or UDP mode, and the set-up needs to be made before executing the OPEN command.

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

0x4004	0x4005
0x13	0x88

Sn_DHAR (Socket n-th Destination Hardware Address Register) [R/W] [0x4006+0x0n00-0x400B+0x0n00] [0xFFFFFFFF]

It sets or is set as destination hardware address of Socket *n-th*. Also, if SOCKET 0 is used for PPPoE mode, S0_DHAR sets as PPPoE server hardware an address that is already known.

When using SEND_MAC command at the UDP or IPRAW mode, it sets the destination hardware address of Socket *n-th*. At the TCP, UDP and IPRAW mode, Sn_DHAR is set as destination hardware address that is acquired by ARP-process of CONNECT or SEND command. The host can acquire the destination hardware address through Sn_DHAR after successfully performing CONNET or SEND command.

When using PPPoE-process of W5200, PPPoE server hardware address is not required to be set. However, even if PPPoE-process of W5200 is not used, but implemented by yourself with MACRAW mode, in order to transmit or receive the PPPoE packet, PPPoE server hardware address(acquired by your PPPoE-process), PPPoE server IP address, and PPP session ID should be set, and MR(PPPoE) also should be set as '1'.

S0_DHAR sets the PPPoE server hardware address before the OPEN command. PPPoE server hardware address which is set by S0_DHAR is applied to PDHAR after performing the OPEN command. The configured PPPoE information is internally valid even after the CLOSE command.

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

0x4006	0x4007	0x4008	0x4009	0x400A	0x400B
0x08	0xDC	0x00	0x01	0x02	0x0A

Sn_DIPR (Socket n-th Destination IP Address Register)[R/W][0x400C+0x0n00 0x400F+0x0n00] [0x00000000]

It sets or is set as destination IP address of Socket *n-th*. If SOCKET0 is used as PPPoE mode, S0_DIPR sets PPPoE server IP address that is already known. It is valid only in TCP, UDP, IPRAW or PPPoE mode, but ignored in MACRAW mode. At the TCP mode, when operating as "TCP CLIENT" it sets as IP address of "TCP SERVER" before performing CONNECT command and when operating as "TCP SERVER", it is internally set as IP address of "TCP CLIENT" after successfully establishing connection.

At the UDP or IPRAW mode, Sn_DIPR sets as destination IP address to be used for transmitting UDP or IPRAW DATA packet before performing SEND or SEND_MAC command.

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

0x400C	0x400D	0x400E	0x400F
192 (0xC0)	168 (0xA8)	0 (0x00)	11 (0x0B)

Sn_DPORT (Socket n-th Destination Port Register)[R/W][0x4010+0x0n00-0x4011+0x0n00] [0x00]

The destination port number is set in the Sn_DPORT of Socket n-th. If SOCKET 0 is used as PPPoE mode, S0_DPORT0 sets PPP session ID that is already known. It is valid only in TCP, UDP or PPPoE mode, and ignored in other modes.

At the TCP mode, when operating as "TCP CLIENT", it listens for the port number of the "TCP SERVER" before performing the CONNECT command.

At the UDP mode, the destination port number is set in the Sn_DPORT to be used for transmitting UDP DATA packets before performing SEND or SEND_MAC command.

At the PPPoE mode, the PPP session ID that is already known is set in the S0_DPORT. PPP session ID (set by S0_DPORT0) is applied to PSIDR after performing the OPEN command.

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

0x4010	0x4011
0x13	0x88

Sn_MSS (Socket n-th Maximum Segment Size Register)[R/W][0x4012+0x0n00-0x4013+0x0n00] [0x0000]

This register is used for MSS (Maximum Segment Size) of TCP, and the register displays MSS set by the other party when TCP is activated in Passive Mode. It just supports TCP or UDP mode. When using PPPoE (MR(PPPoE)='1'), the MTU of TCP or UDP mode is assigned in the range of MTU of PPPoE.

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
IPRAW	1480		1472	
MACRAW	1514			

At the IPRAW or MACRAW, MTU is not processed internally, but default MTU is used. Therefore, when transmitting the data bigger than default MTU, the host should manually divide the data into the unit of default MTU.

At the TCP or UDP mode, if transmitting data is bigger than MTU, W5200 automatically divides

the data into the unit of MTU. MTU is called as MSS at the TCP mode. By selecting from Host-Written-Value and peer's MSS, MSS is set as smaller value through TCP connection process.

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

0x4012	0x4013
0x05	0xB4

Sn_PROTO (Socket n-th IP Protocol Register) [R/W] [0x4014+0x0n00] [0x00]

It is a 1 byte register that sets the protocol number field of the IP header at the IP layer. It is valid only in IPRAW mode, and ignored in other modes. Sn_PROTO is set before OPEN command. When Socket n-th is opened in IPRAW mode, it transmits and receives the data of the protocol number set in Sn_PROTO. Sn_PROTO can be assigned in the range of 0x00 ~ 0xFF, but W5200 does not support TCP(0x06) and UDP(0x11) protocol number

Protocol number is defined in IANA(Internet assigned numbers authority). For the detail, refer to online document (<http://www.iana.org/assignments/protocol-numbers>).

Ex) Internet Control Message Protocol (ICMP) = 0x01, Internet Group Management Protocol = 0x02

Sn_TOS (Socket n-th IP Type Of Service Register) [R/W] [0x4015+0x0n00] [0x00]

It sets the TOS(Type of Service) field of the IP header at the IP layer. It should be set before the OPEN command. Refer to <http://www.iana.org/assignments/ip-parameters>.

Sn_TTL (Socket n-th IP Time To Live Register) [R/W] [0x4016+0x0n00] [0x80]

It sets the TTL(Time To Live) field of the IP header at the IP layer. It should be set before the OPEN command. Refer to <http://www.iana.org/assignments/ip-parameters>.

Sn_RXMEM_SIZE(Socket n-th RX Memory Size Register) [R/W] [0x401E+0x0n00] [0x02]

It configures the internal RX Memory size of each SOCKET. RX Memory size of each SOCKET is configurable in the size of 0, 1, 2, 4, 8, and 16 Kbytes. 2Kbytes is assigned when reset.

Sn_RXMEM_SIZE_{SUM}(sum of Sn_RXMEM_SIZE) of each SOCKET should be 16KB.

Value	0x00	0x01	0x02	0x04	0x08	0x10
Memory size	0KB	1KB	2KB	4KB	8KB	16KB

Ex1) SOCKET 0 : 8KB, SOCKET 1 : 2KB

0x401E	0x411E
0x08	0x02

Ex2) SOCKET 2 : 1KB, SOCKET 3 : 1KB

0x421E	0x431E
0x01	0x01

Ex3) SOCKET 4 : 1KB, SOCKET 5 : 1KB

0x441E	0x451E
0x01	0x01

Ex4) SOCKET 6 : 1KB, SOCKET 7 : 1KB

0x461E	0x471E
0x01	0x01

Sn_TXMEM_SIZE(Socket n-th TX Memory size Register) [R/W][0x401F+0x0n00] [0x02]

It configures the internal TX Memory size of each SOCKET. TX Memory size of each SOCKET is configurable in the size of 0, 1, 2, 4, 8, and 16Kbytes. 2Kbytes is assigned when reset.

Sn_TXMEM_SIZE_{SUM}(summation of Sn_TXMEM_SIZE) of each SOCKET should be 16KB.

Ex1) SOCKET 0 : 4KB, SOCKET 1 : 1KB

0x401F	0x411F
0x04	0x01

Ex2) SOCKET 2 : 2KB, SOCKET 3 : 1KB

0x421F	0x431F
0x02	0x01

Ex3) SOCKET 4 : 2KB, SOCKET 5 : 2KB

0x441F	0x451F
0x02	0x02

Ex4) SOCKET 6 : 2KB, SOCKET 7 : 2KB

0x461F	0x471F
0x02	0x02

Sn_TX_FSR (Socket n-th TX Free Size Register) [R] [0x4020+0x0n00-0x4021+0x0n00] [0x0800]

It notifies the available size of the internal TX memory (the byte size of transmittable data) of Socket n-th. The host can't write data as a size bigger than Sn_TX_FSR. Therefore, be sure to check Sn_TX_FSR before transmitting data, and if your data size is smaller than or the same as Sn_TX_FSR, transmit the data with SEND or SEND_MAC command after copying the data.

At the TCP mode, if the peer checks the transmitted DATA packet (if DATA/ACK packet is received from the peer), Sn_TX_FSR is automatically increased by the size of that transmitted DATA packet. At the other modes, when Sn_IR(SENDOK) is '1', Sn_TX_FSR is automatically increased by the size of the transmitted data. *When checking this register, user should read upper byte(0x4020, 0x4120, 0x4220, 0x4320, 0x4420, 0x4520, 0x4620, 0x4720) first and lower byte(0x4021, 0x4121, 0x4221, 0x4321, 0x4421, 0x4521, 0x4621, 0x4721) later to get the correct value.*

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

0x4020	0x4021
0x08	0x00

Sn_TX_RD (Socket n-th TX Read Pointer Register) [R] [0x4022+0x0n00-0x4023+0x0n00] [0x0000]

This register shows the address of the last transmission finishing in the TX memory. With the SEND command of Socket n-th Command Register, it transmits data from the current Sn_TX_RD to the Sn_TX_WR and automatically updates after transmission is finished. Therefore, after transmission is finished, Sn_TX_RD and Sn_TX_WR will have the same value. When reading this register, user should read upper byte (0x4022, 0x4122, 0x4222, 0x4322, 0x4422, 0x4522, 0x4622, 0x4722) first, and lower byte (0x4023, 0x4123, 0x4223, 0x4323, 0x4423, 0x4523, 0x4623, 0x4723) later to get the correct value.

Sn_TX_WR (Socket n-th TX Write Pointer Register) [R/W] [0x4024+0x0n00-0x4025+0x0n00] [0x0000]

This register offers the location information to write the transmission data. When reading this register, user should read upper byte (0x4024, 0x4124, 0x4224, 0x4324, 0x4424, 0x4524, 0x4624, 0x4724) first, and lower byte (0x4025, 0x4125, 0x4225, 0x4325, 0x4425, 0x4525, 0x4625, 0x4725) later to get the correct value.

Caution: This register value is changed after the send command is successfully executed to Sn_CR.

Ex) In case of 2048(0x0800) in S0_TX_WR,

0x4024	0x4025
0x08	0x00

Chip Base Address = 0x0000, 512(0x0200) bytes send

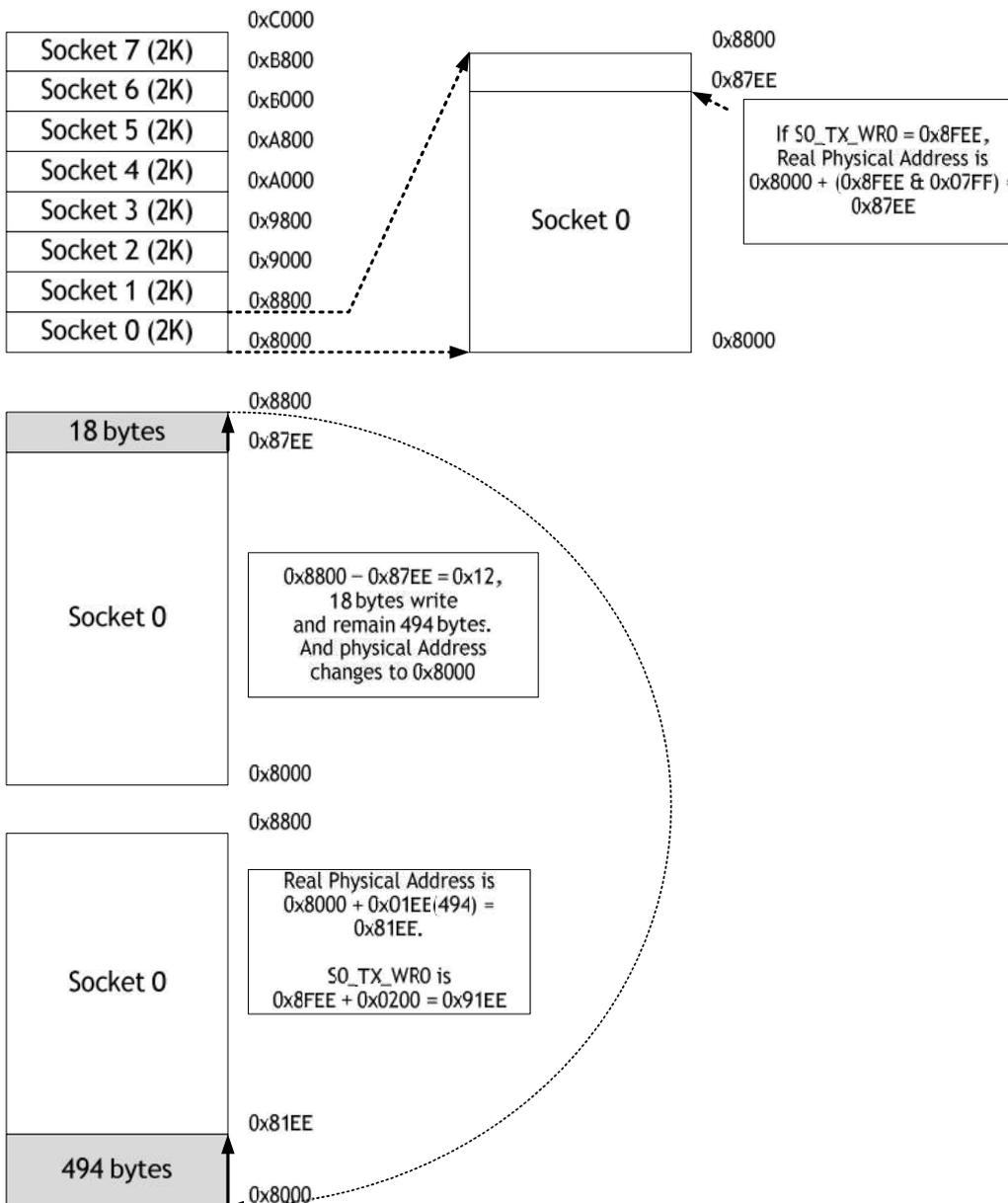


Figure 7 Physical Address Calculation

But this value itself is not the physical address to read. So, the physical address should be calculated as follow.

1. Socket n-th TX Base Address (hereafter we'll call gSn_TX_BASE) and Socket n-th TX Mask Address (hereafter we'll call gSn_TX_MASK) are calculated on TMSR value. *Refer to the pseudo code of the Initialization if detail is needed.*
2. The bitwise-AND operation of two values, Sn_TX_WR and gSn_TX_MASK give result to the offset address(hereafter we'll call get_offset) in TX memory range of the socket.
3. Two values get_offset and gSn_TX_BASE are added together to give result to the physical address(hereafter, we'll call $get_start_address$).

Now, write the transmission data to $get_start_address$ as large as you want. (* There's a case that it exceeds the TX memory upper-bound of the socket while writing. In this case, write

the transmission data to the upper-bound, and change the physical address to the `gSn_TX_BASE`. Next, write the rest of the transmission data.) After that, be sure to increase the `Sn_TX_WR` value as much as the data size that indicates the size of writing data. Finally, give SEND command to `Sn_CR`(Socket n-th Command Register).

Refer to the pseudo code of the transmission part on TCP Server mode if detail is needed.

Sn_RX_RSR (SOCKET n-th Received Size Register) [R] [0x4026+0x0n00-0x4027+0x0n00] [0x0000]

It informs the user of the byte size of the received data in Internal RX Memory of Socket n-th. As this value is internally calculated with the values of `Sn_RX_RD` and `Sn_RX_WR`, it is automatically changed by RECV command of Socket n-th Command Register (`Sn_CR`) and receives data from the remote peer. *When reading this register, user should read upper byte(0x4026, 0x4126, 0x4226, 0x4326, 0x4426, 0x4526, 0x4626, 0x4726) first, and lower byte(0x4027, 0x4127, 0x4227, 0x4327, 0x4427, 0x4527, 0x4627, 0x4727) later to get the correct value.*

Ex) In case of 2048(0x0800) in `SO_RX_RSR`,

0x4026	0x04027
0x08	0x00

The total size of this value can be decided according to the value of RX Memory Size Register.

Sn_RX_RD (Socket n-th RX Read Pointer Register) [R/W] [0x4028+0x0n00-0x4028+0x0n00] [0x0000]

This register offers the location information to read the receiving data. When reading this register, user should read upper byte (0x4028, 0x4128, 0x4228, 0x4328, 0x4428, 0x4528, 0x4628, 0x4728) first, and lower byte (0x4029, 0x4129, 0x4229, 0x4329, 0x4429, 0x4529, 0x4629, 0x4729) later to get the correct value. It has a random value as its initial value.

Caution: This register value is changed after the SEND command is successfully executed to `Sn_CR`.

Ex) In case of 2048(0x0800) in `SO_RX_RD`,

0x4028	0x4029
0x08	0x00

But this value itself is not the physical address to read. So, the physical address should be calculated as follow.

1. Socket n-th RX Base Address (hereafter we'll call `gSn_RX_BASE`) and Socket n-th RX Mask Address (hereafter we'll call `gSn_RX_MASK`) are calculated on RMSR value. *Refer to the pseudo code of the 5.1 Initialization if the detail is needed.*
2. The bitwise-AND operation of two values, `Sn_RX_RD` and `gSn_RX_MASK` give result the offset address(hereafter we'll call `get_offset`), in RX memory range of the socket.
3. Two values `get_offset` and `gSn_RX_BASE` are added together to give result the

physical address(hereafter, we'll call *get_start_address*).

Sn_RX_WR (Socket n-th RX Write Pointer Register)[R/W][(0xFE402A + 0xn00) - (0xFE402B + 0xn00)][0x0000]

This register offers the location information to write the receive data. When reading this register, the user should read upper bytes (0x402A, 0x412A, 0x422A, 0x432A, 0x442A, 0x452A, 0x462A, 0x472A) first and lower bytes (0x402B, 0x412B, 0x422B, 0x432B, 0x442B, 0x452B, 0x462B, 0x472B) later to get the correct value.

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

0x402A	0x402B
0x08	0x00

Sn_IMR (Socket n-th Interrupt Mask Register)[R/W][0x402C+0x0n00][0xFF]

It configures the interrupt of Socket n-th so as to notify to the host. Interrupt mask bit of Sn_IMR corresponds to interrupt bit of Sn_IR. If interrupt occurs in any SOCKET and the bit is set as '1', its corresponding bit of Sn_IR is set as '1'. When the bits of Sn_IMR and Sn_IR are '1', IR(n) becomes '1'. At this time, if IMR(n) is '1', the interrupt is issued to the host. ('nINT' signal is asserted low)

7	6	5	4	3	2	1	0
PRECV	PFAIL	PNEXT	SEND_OK	TIMEOUT	RECV	DISCON	CON

Bit	Symbol	Description
7	PRECV	Sn_IR(PRECV) Interrupt Mask Valid only in case of 'SOCKET = 0' & 'S0_MR(P3:P0) = S0_MR_PPPoE'
6	PFAIL	Sn_IR(PFAIL) Interrupt Mask Valid only in case of 'SOCKET = 0' & 'S0_MR(P3:P0) = S0_MR_PPPoE'
5	PNEXT	Sn_IR(PNEXT) Interrupt Mask Valid only in case of 'SOCKET = 0' & 'S0_MR(P3:P0) = S0_MR_PPPoE'
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-th Fragment Register)[R/W][0x402D+0x0n00-0x402E+0x0n100][0x4000]

It sets the Fragment field of the IP header at the IP layer. W5200 does not support the packet fragment at the IP layer. Even though Sn_FRAG is configured, IP data is not fragmented, and not recommended either. It should be configured before performing OPEN command.

Ex) Sn_FRAG0 = 0x4000 (Don't Fragment)

0x402D	0x402E
0x40	0x00

5 Functional Descriptions

By setting some register and memory operation, W5200 provides internet connectivity. This chapter describes how it can be operated.

5.1 Initialization

Basic Setting

For the W5200 operation, select and utilize appropriate registers shown below.

1. Mode Register (MR)
2. Interrupt Mask Register (IMR)
3. Retry Time-value Register (RTR)
4. Retry Count Register (RCR)

For more information of above registers, refer to the “Register Descriptions.”

Setting network information

Basic network information setting for communication:

It must be set the basic network information.

① SHAR(Source Hardware Address Register)

It is prescribed that the source hardware addresses, which is set by SHAR, use unique hardware addresses (Ethernet MAC address) in the Ethernet MAC layer. The IEEE manages the MAC address allocation. The manufacturer which produces the network device allocates the MAC address to product.

Details on MAC address allocation refer to the website as below.

<http://www.ieee.org/>, <http://standards.ieee.org/regauth/oui/index.shtml>

② GAR(Gateway Address Register)

③ SUBR(Subnet Mask Register)

④ SIPR(Source IP Address Register)

Set socket memory information

This stage sets the socket tx/rx memory information. The base address and mask address of each socket are fixed and saved in this stage.

```

In case of, assign 2KB rx, tx memory per SOCKET
{
gS0_RX_BASE = 0x0000(Chip base address) + 0xC000(Internal RX buffer address); // Set
base address of RX memory for SOCKET 0
Sn_RXMEM_SIZE(ch) = (uint8 *) 2; // Assign 2K rx memory per SOCKET
gS0_RX_MASK = 2K - 1; // 0x07FF, for getting offset address within assigned SOCKET 0 RX
memory
gS1_RX_BASE = gS0_RX_BASE + (gS0_RX_MASK + 1);
gS1_RX_MASK = 2K - 1;
gS2_RX_BASE = gS1_RX_BASE + (gS1_RX_MASK + 1);
gS2_RX_MASK = 2K - 1;
gS3_RX_BASE = gS2_RX_BASE + (gS2_RX_MASK + 1);
gS3_RX_MASK = 2K - 1;
gS4_RX_BASE = gS3_RX_BASE + (gS3_RX_MASK + 1);
gS4_RX_MASK = 2K - 1;
gS5_RX_BASE = gS4_RX_BASE + (gS4_RX_MASK + 1);
gS5_RX_MASK = 2K - 1;
gS6_RX_BASE = gS5_RX_BASE + (gS5_RX_MASK + 1);
gS6_RX_MASK = 2K - 1;
gS7_RX_BASE = gS6_RX_BASE + (gS6_RX_MASK + 1);
gS7_RX_MASK = 2K - 1;
gS0_TX_BASE = 0x0000(Chip base address) + 0x8000(InternalTX buffer address); // Set
base address of TX memory for SOCKET 0
Sn_TXMEM_SIZE(ch) = (uint8 *) 2; // Assign 2K rx memory per SOCKET
gS0_TX_MASK = 2K - 1;
/* Same method, set gS1_TX_BASE, gS1_TX_MASK, gS2_TX_BASE, gS2_TX_MASK,
gS3_TX_BASE, gS3_TX_MASK, gS4_TX_BASE, gS4_TX_MASK, gS5_TX_BASE, gS5_TX_MASK,
gS6_TX_BASE, gS6_tx_MASK, gS7_TX_BASE, gS7_TX_MASK */
}
    
```