W5100S

(W5100S-L & W5100S-Q)

Version 1.2.7

http://www.wiznet.io/

W5100S

W5100S designed with Hardwired TCP/IP, WIZnet Technology, is an embedded Internet Controller Chip. W5100S supporting Full Hardwired, Ethernet MAC (Media Access Control), and 10Base-T/100Base-TX Ethernet PHY is Internet Connectivity One-chip Solution for Internet Protocol (TCP/IP).

With W5100S, Host (User MCU) simply handles variety Internet Protocol such as IPv4, TCP, UDP, ICMP, IGMP, ARP, PPPoE and etc. And W5100S supports each 8KB Memory for Transmit and Receive to minimize using Memory on Low-end level Host. Host also independently uses 4 Hardwired SOCKETs to develop vary Internet Applications in each Hardwired SOCKETs.

W5100S supports SPI and Parallel System BUS Interface for Host Interface. It also provides Low Power / Low Heat design, WOL (Wake On LAN), Ethernet PHY Power Down Mode and etc.

W5100S is Low-cost chip that improves on W5100. Any Firmware using on W5100 can be used on W5100S without any modification. Also, W5100S has 48 Pin LQFP & QFN Lead-Free Package, smaller than W5100 for product miniaturization.

Features

- Support Hardwired Internet protocols
  : TCP, UDP, WOL over UDP, ICMP, IGMPv1/v2, IPv4, ARP, PPPoE
- Support 4 independent SOCKETs simultaneously
- Support SOCKET-less Command
  : ARP-Request, PING-Request
- Support Ethernet Power Down Mode & Main Clock Switching for power save
- Support Wake on LAN over UDP
- Support Serial & Parallel Host Interface
  : High Speed SPI (MODE 0/3), System Bus with 2 Address signal & 8bit Data
- Internal 16Kbytes Memory for TX/ RX Buffers
- 10Base-T/100Base-TX Ethernet PHY Integrated
- Support Auto Negotiation (Full and Half Duplex, 10 and 100-based)
- Support Auto-MDIX only on Auto-Negotiation Mode
- Not support IP Fragmentation
- 3.3V operation with 5V I/O signal tolerance
- Network Indicator LEDs (Full/Half duplex, Link, 10/100 Speed, Active)
- 48 Pin LQFP & QFN Lead-Free Package (7x7mm, 0.5mm pitch)

Target Applications

W5100S is well-suited for many embedded Applications, including:

- User product based on W5100 : No modify Firmware
- 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, Building, Home Automations
- Medical Monitoring Equipment
- Embedded Servers
- Internet of Thing (IoT) Devices
- IoT Cloud Devices

Block Diagram

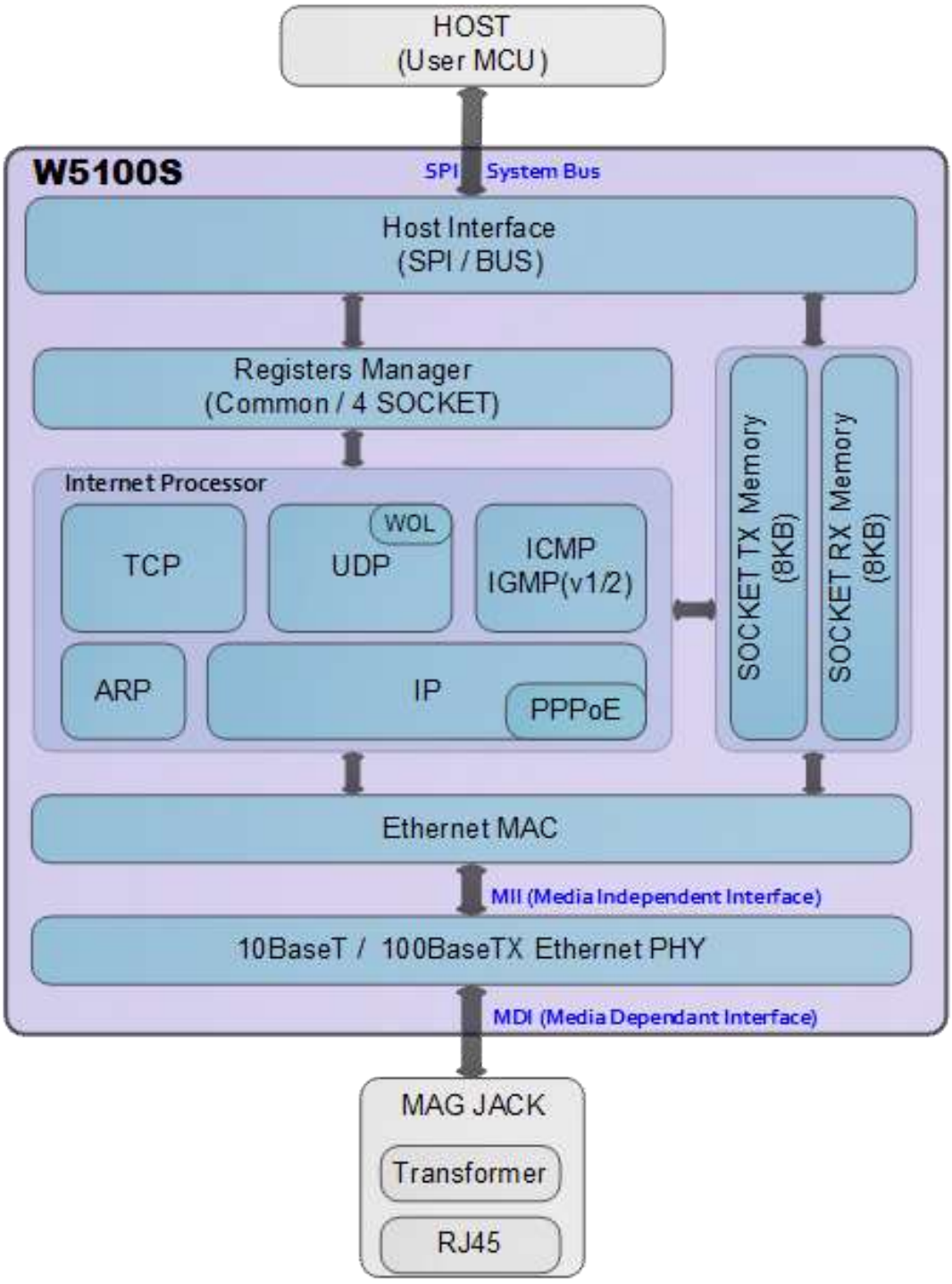


Figure 1 Block Diagram

Contents

List of Figures

List of Tables

1 PIN Description

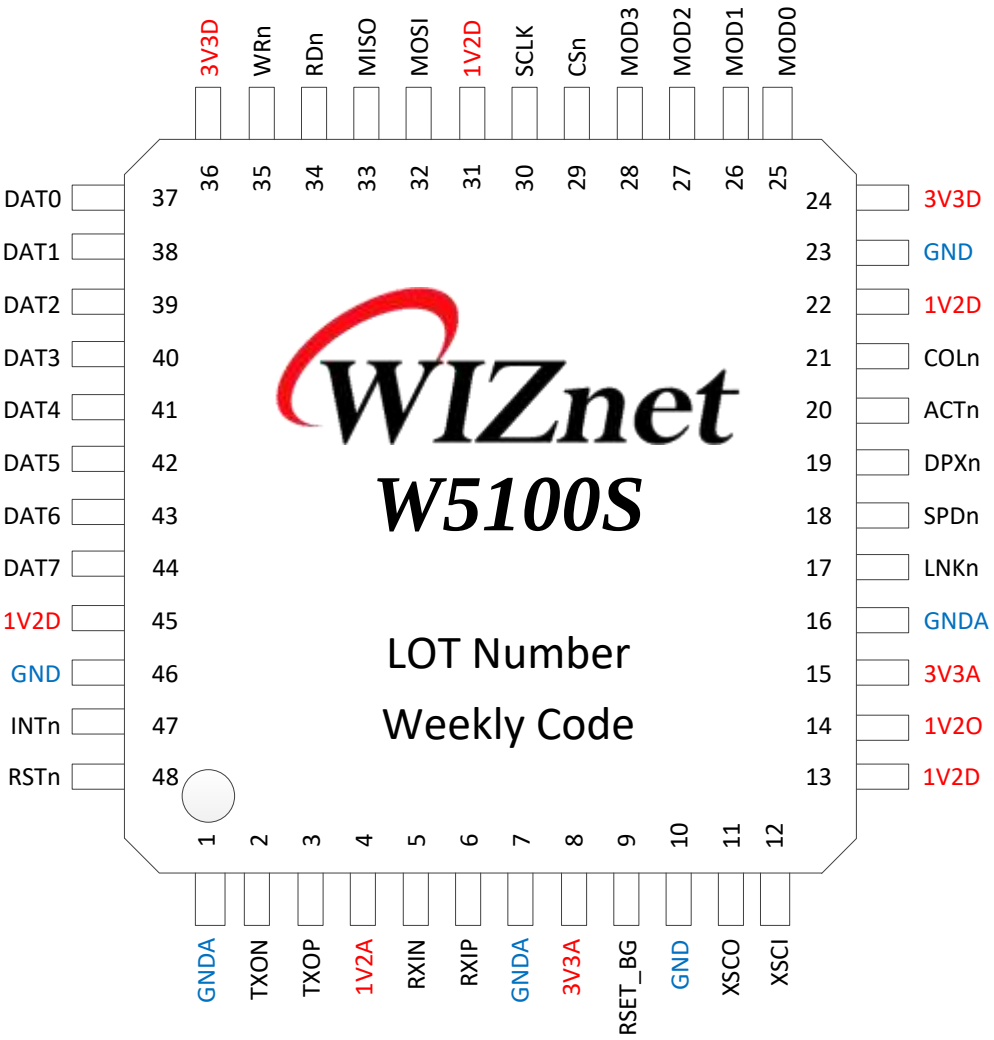


Figure 2 W5100S Pin Layout

Table 1 Pin Type Notation

Type	Description
I	Input
O	Output
M	Alternate (Multi-function) Signal
U	Internal pulled-up 75KΩ resistor
D	Internal pulled-down 75KΩ resistor
A	Analog
P	Power & Ground

1.1 PIN Description

Table 2 PIN Description

PIN #	Symbol	Type	Description
1	GNDA	AP	**Analog Ground**
2	TXON	AO	**Differential Transmitted Signal Pair** Differential Data is transmitted to Media via TXOP/TXON signal pair on MDI Mode.
3	TXOP	AO	
4	1V2A	AP	**Analog 1.2V Power** Supplied from 1V20 voltage source
5	RXIN	AI	**Differential Received Signal Pair** Differential Data is received from Media via RXIP/RXIN signal pair on MDI Mode.
6	RXIP	AI	
7	GNDA	AP	**Analog Ground**
8	3V3A	AP	**Analog 3.3V Power**
9	RSET_BG	AO	**Off-chip Bias Resistor** Must be connected to analog Ground through external 12.3KΩ, error 1% Resistance.
10	GND	AP	**Digital Ground**
11	XSCO	AO	**25MHz Clock** 25MHz Crystal Oscillator (TXAL) or Oscillator (OSC) are used for Internal oscillator stabilization. W5100S uses 25MHz (Low Frequency Mode) or 100MHz (Normal Mode) as Internal Clock from External 25MHz Clock Source. If OSC is used, 25MHz@1.2V must be used and only XSCI must be connected and XSCO must be floated. For more information, refer to *Clock Selection Guide*.
12	XSCI	AI	
13	1V2D	P	**Digital 1.2V Power** Supplied from 1V20 voltage source
14	1V2O	PO	**Internal Regulator 1.2V Power Output** Internal Regulator for W5100S needs Max 150mA for 1.2V Power Output. Make sure to supply 1V2D and 1V2A for External Capacitor 3.3uF stabilization. 1V2O must use Ferrite Bead. 1V2D and 1V2A must be separated and be supplied. This power is only for W5100S. It must not be used for other device.

15	3V3A	AP	**Analog 3.3V Power**
16	GNDA	AP	**Analog Ground**
17	LNKn	OU	**Link Status LED** It is valid on SPI and Parallel Bus Mode. Low : Link up High : Link down
18	SPDn	OU	**Link Speed LED** It is valid on SPI and Parallel Bus Mode. Low : 100Mbps High : 10Mbps
19	DPXn	OU	**Link Duplex LED** It is valid on SPI and Parallel Bus Mode. Low : Full-Duplex High : Half-Duplex
20	ACTn	OU	**Link Activity LED** It is valid on SPI and Parallel Bus Mode. No Flash : Link up state without TX/RX Flash : Link up state with TX/RX data High : Link-down state
21	COLn	OU	**Link Collision Detect LED** It is valid on SPI and Parallel Bus Mode. It indicates a collision during Data transmission. Low : Collision Detected High : No Collision
22	1V2D	P	**Digital 1.2V Power** Supplied from 1V20 voltage source.
23	GND	P	**Digital Ground**
24	3V3D	P	**Digital 3.3V power**
25	MOD[0]	ID	**W5100S Interface Mode Selection**
26	MOD[1]	ID	Interface Mode is selected by MOD [3:0].
27	MOD[2]	ID	
28	MOD[3]	ID	“0000” : SPI Mode “010X” : Parallel Bus Mode
29	CSn	IU	**W5100S Chip Select**

			Low : Select High : No Select
30	SCLK	ID	**SPI Clock** On SPI Mode, it is used to SPI Clock. But on Parallel Bus Mode, it must be connected to GND or be floated.
31	1V2D	P	**Digital 1.2V Power** Supplied from 1V20 voltage source.
32	MOSI /ADDR0	IDM	**SPI Master Output Slave Input / Address 0** MOSI : On SPI Mode, SPI Data is received from HOST ADDR0 : On Parallel Bus Mode, it is used to Address 0
33	MISO /ADDR1	IOPM	**SPI Master Input Slave Output / Address 1** MISO : On SPI Mode, SPI Data is transmitted to HOST ADDR1 : On Parallel Bus Mode, It is used to Address1
34	RDn	IU	**Read Strobe** On Parallel Bus Mode, it indicates Read Operation On SPI Mode, it must be connected to 3V3D or be floated
35	WRn	IU	**Write Strobe** On Parallel Bus Mode, it indicates Write Operation.
36	3V3D	P	**Digital 3.3V Power**
37	DAT0	IOU	**8 Bits Data Bus** On Parallel Bus Mode, DAT [7:0] receives Data from HOST or W5100S. On SPI Mode, DAT [7:0] must be floated.
38	DAT1	IOU	
39	DAT2	IOU	
40	DAT3	IOU	
41	DAT4	IOU	
42	DAT5	IOU	
43	DAT6	IOU	
44	DAT7	IOU	
45	1V2D	P	**Digital 1.2V Power**
46	GND	P	**Digital Ground**
47	INTn	OP	**Interrupt** When the event occurs during W5100S Ethernet Communication, INTn notices to HOST. Low : Interrupt Occurred High : No Interrupt (Ref *IEN (Interrupt pin Enable) in MR2 (Mode Register 2)*, *INTPTMR (Interrupt Pending Time Register)*, *IMR*

			(Interrupt Mask Register), *IMR2 (Interrupt Mask Register 2)*, *SLIMR (SOCKET-less Interrupt Mask Register)*)
48	RSTn	IP	**Reset** RSTn initializes W5100S. RSTn must be asserted to Low longer than 560ns. After asserted RSTn, W5100S spends 60.3ms for initialization. (Ref *7.4.1 Reset Timing*) Low : W5100S initialized. High : Normal Operation.

2 Memory Map

W5100S has the same Memory Map as W5100 for compatibility and additional Common Register for improved functionality. The below Figure 3 shows W5100S Memory Map.

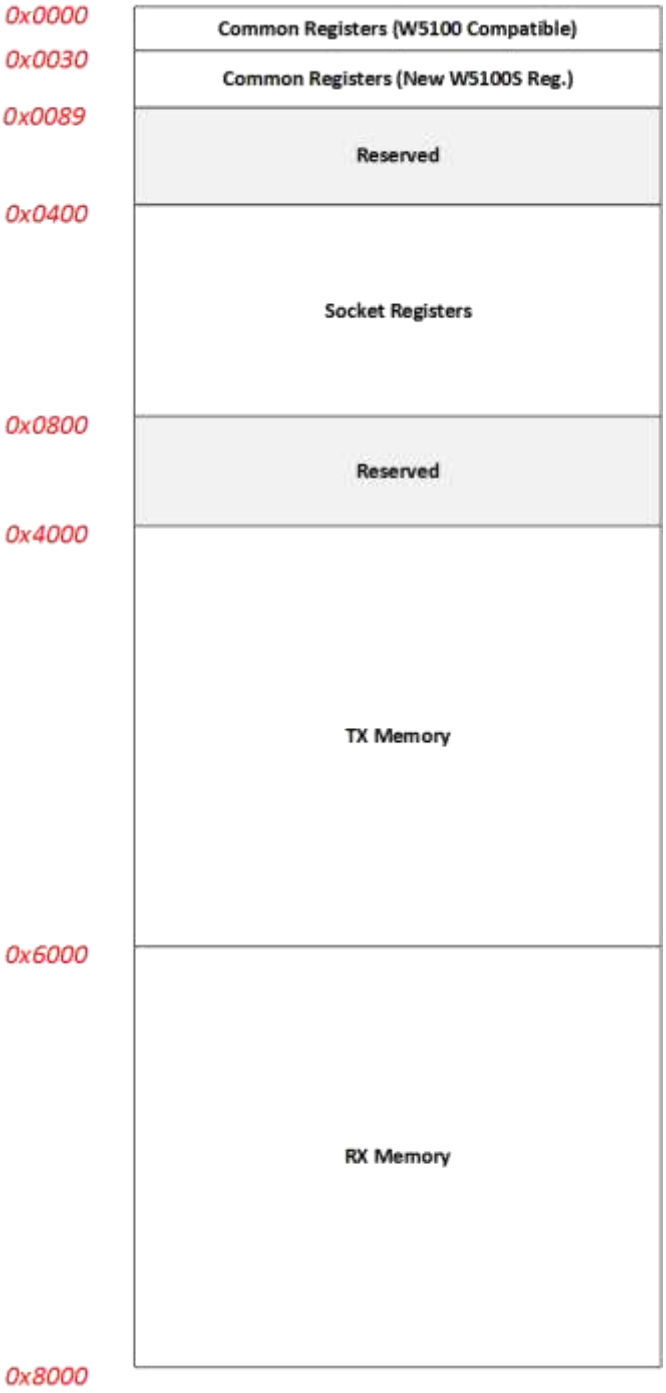


Figure 3 Memory Map

Figure 3 shows the Address Offset of Common & SOCKET Register Block and TX/RX Memory Block. At W5100S Reset, each SOCKET n TX/RX Buffer are assigned with 2KB/2KB from TX/RX Memory Block.

After W5100S Reset, each SOCKET n TX/RX Buffer Size are set by TMSR (TX Memory Size Register) and RMSR (RX Memory Size Register) or by SOCKET n TX/RX Buffer Size Register (Sn_TXBUF_SIZE / Sn_RXBUF_SIZE). The total Buffer Size of SOCKET n TX/RX must not be exceeded by 8 Kbytes.

2.1 W5100S Registers

2.1.1 Common registers

Table 3 Common Registers

Address	Register
0x0000	Mode (MR)
0x0001 0x0002 0x0003 0x0004	Gateway Address (GAR0) (GAR1) (GAR2) (GAR3)
0x0005 0x0006 0x0007 0x0008	Subnet Mask Address (SUBR0) (SUBR1) (SUBR2) (SUBR3)
0x0009 0x000A 0x000B 0x000C 0x000D 0x000E	Source Hardware Address (SHAR0) (SHAR1) (SHAR2) (SHAR3) (SHAR4) (SHAR5)
0x000F 0x0010 0x0011 0x0012	Source IP Address (SIPR0) (SIPR1) (SIPR2) (SIPR3)
0x0013 0x0014	Interrupt Pending Time (INTPTMR0) (INTPTMR1)
0x0015	Interrupt (IR)
0x0016	Interrupt Mask (IMR)
0x0017 0x0018	Retransmission Time (RTR0) (RTR0)
0x0019	Retransmission Time (RCR)
0x001A	RX Memory Size (RMSR)
0x001B	TX Memory Size (TMSR)

Address	Register
0x001C ~ 0x001F	Reserved
0x0020	Interrupt2 (IR2)
0x0021	Interrupt2 Mask (IMR2)
0x0022 ~ 0x0027	Reserved
0x0028	PPP LCP Request Timer (PTIMER)
0x0029	PPP LCP Magic Number (PMAGIC)
0x002A 0x002B 0x002C 0x002D	Unreachable IP Address (UIPR0) (UIPR1) (UIPR2) (UIPR3)
0x002E 0x002F	Unreachable Port (UPORTR0) (UPORTR1)
0x0030	Mode2 (MR2)
0x0031	Reserved
0x0032 0x0033 0x0034 0x0035 0x0036 0x0037	Destination Hardware Address on PPPoE (PHAR0) (PHAR1) (PHAR2) (PHAR3) (PHAR4) (PHAR5)
0x0038 0x0039	Session ID on PPPoE (PSIDR0) (PSIDR1)
0x003A 0x003B	Maximum Receive Unit on PPPoE (PMRUR0) (PMRUR1)
0x003C	PHY Status (PHYSR0)

Address	Register
0x003E	PHY Address Value (PHYAR)
0x003F	PHY Register Address (PHYRAR)
0x0040 0x0041	PHY Data Input (PHYDIR0) (PHYDIR1)
0x0042 0x0043	PHY Data Output (PHYDOR0) (PHYDOR1)
0x0044	PHY Access (PHYACR)
0x0045	PHY Division (PHYDIVR)
0x0046 0x0047	PHY Control (PHYCR0) (PHYCR1)
0x0048 ~ 0x004B	Reserved
0x004C	SOCKET-less Command (SLCR)
0x004D 0x004E	SOCKET-less Retransmission Time (SLRTR0) (SLRTR1)
0x004F	SOCKET-less Retransmission Count (SLRCR)
0x0050 0x0051 0x0052 0x0053	SOCKET-less Peer IP Address (SLPIPR0) (SLPIPR1) (SLPIPR2) (SLPIPR3)
0x0054 0x0055 0x0056 0x0057 0x0058 0x0059	SOCKET-less Peer Hardware Address (SLPHAR0) (SLPHAR1) (SLPHAR2) (SLPHAR3) (SLPHAR4) (SLPHAR5)
0x005A 0x005B	PING Sequence Number (PINGSEQR0) (PINGSEQR1)
0x005C 0x005D	PING ID (PINGIDR0) (PINGIDR1)

Address	Register
0x005E	SOCKET-less Interrupt Mask (SLIMR)
0x005F	SOCKET-less Interrupt (SLIR)
0x0060 ~ 0x006A	Reserved
0x0070	Clock Lock (CLKLCKR)
0x0071	Network Lock (NETLCKR)
0x0072	PHY Lock (PHYLCKR)
0x0073 ~ 0x007F	Reserved
0x0080	Chip Version (VERR)
0x0081	Reserved
0x0082 0x0083	100us Tick Counter (TCNTR0) (TCNTR1)
0x0084 ~ 0x0087	Reserved
0x0088	TCNTCLR

2.1.2 SOCKET Registers

Table 4 Socket Registers

Symbol	Description	Address				
		Sn_	S0_	S1_	S2_	S3_
Sn_MR	SOCKET n Mode	0x0400+(0x0100 x n)	0x0400	0x0500	0x0600	0x0700
Sn_CR	SOCKET n Command	0x0401+(0x0100 x n)	0x0401	0x0501	0x0601	0x0701
Sn_IR	SOCKET n Interrupt	0x0402+(0x0100 x n)	0x0402	0x0502	0x0602	0x0702
Sn_SR	SOCKET n Status	0x0403+(0x0100 x n)	0x0403	0x0503	0x0603	0x0703
Sn_PORTR0	SOCKET n Source Port	0x0404+(0x0100 x n)	0x0404	0x0504	0x0604	0x0704
Sn_PORTR1		0x0405+(0x0100 x n)	0x0405	0x0505	0x0605	0x0705
Sn_DHAR0	SOCKET n Destination Hardware Address	0x0406+(0x0100 x n)	0x0406	0x0506	0x0606	0x0706
Sn_DHAR1		0x0407+(0x0100 x n)	0x0407	0x0507	0x0607	0x0707
Sn_DHAR2		0x0408+(0x0100 x n)	0x0408	0x0508	0x0608	0x0708
Sn_DHAR3		0x0409+(0x0100 x n)	0x0409	0x0509	0x0609	0x0709
Sn_DHAR4		0x040A+(0x0100 x n)	0x040A	0x050A	0x060A	0x070A
Sn_DHAR5		0x040B+(0x0100 x n)	0x040B	0x050B	0x060B	0x070B
Sn_DIPR0	SOCKET n Destination IP Address	0x040C+(0x0100 x n)	0x040C	0x050C	0x060C	0x070C
Sn_DIPR1		0x040D+(0x0100 x n)	0x040D	0x050D	0x060D	0x070D
Sn_DIPR2		0x040E+(0x0100 x n)	0x040E	0x050E	0x060E	0x070E
Sn_DIPR3		0x040F+(0x0100 x n)	0x040F	0x050F	0x060F	0x070F
Sn_DPORTR0	SOCKET n Destination Port	0x0410+(0x0100 x n)	0x0410	0x0510	0x0610	0x0710
Sn_DPORTR0		0x0411+(0x0100 x n)	0x0411	0x0511	0x0611	0x0711
Sn_MSS0	SOCKET n Maximum Segment Size	0x0412+(0x0100 x n)	0x0412	0x0512	0x0612	0x0712
Sn_MSS1		0x0413+(0x0100 x n)	0x0413	0x0513	0x0613	0x0713
Sn_PROTOR	SOCKET n IP Protocol	0x0414+(0x0100 x n)	0x0414	0x0514	0x0614	0x0714
Sn_TOS	SOCKET n IP Type Of Service	0x0415+(0x0100 x n)	0x0415	0x0515	0x0615	0x0715
Sn_TTL	SOCKET n IP Time To Live	0x0416+(0x0100 x n)	0x0416	0x0516	0x0616	0x0716
Reserved	Reserved	0x0417+(0x0100 x n)	0x0417	0x0517	0x0617	0x0717
Reserved	Reserved	0x041D+(0x0100 x n)	0x041D	0x051D	0x061D	0x071D
Sn_RXBUF_SIZE	SOCKET n RX Buffer Size	0x041E+(0x0100 x n)	0x041E	0x051E	0x061E	0x071E
Sn_TXBUF_SIZE	SOCKET n TX Buffer Size	0x041F+(0x0100 x n)	0x041F	0x051F	0x061F	0x071F
Sn_TX_FSR0	SOCKET n	0x0420+(0x0100 x n)	0x0420	0x0520	0x0620	0x0720

Sn_TX_FSR1	TX Free Size	0x0421+(0x0100 x n)	0x0421	0x0521	0x0621	0x0721
Sn_TX_RD0	SOCKET n TX Read Pointer	0x0422+(0x0100 x n)	0x0422	0x0522	0x0622	0x0722
Sn_TX_RD1		0x0423+(0x0100 x n)	0x0423	0x0523	0x0623	0x0723
Sn_TX_WR0	SOCKET n TX Write Pointer	0x0424+(0x0100 x n)	0x0424	0x0524	0x0624	0x0724
Sn_TX_WR1		0x0425+(0x0100 x n)	0x0425	0x0525	0x0625	0x0725
Sn_RX_RSR0	SOCKET n RX Received Size	0x0426+(0x0100 x n)	0x0426	0x0526	0x0626	0x0726
Sn_RX_RSR1		0x0427+(0x0100 x n)	0x0427	0x0527	0x0627	0x0727
Sn_RX_RD0	SOCKET n RX Read Pointer	0x0428+(0x0100 x n)	0x0428	0x0528	0x0628	0x0728
Sn_RX_RD1		0x0429+(0x0100 x n)	0x0429	0x0529	0x0629	0x0729
Sn_RX_WR0	SOCKET n RX Write Pointer	0x042A+(0x0100 x n)	0x042A	0x052A	0x062A	0x072A
Sn_RX_WR1		0x042B+(0x0100 x n)	0x042B	0x052B	0x062B	0x072B
Sn_IMR	SOCKET n Interrupt Mask	0x042C+(0x0100 x n)	0x042C	0x052C	0x062C	0x072C
Sn_FRAGR0	SOCKET n Fragment Offset in IP Header	0x042D+(0x0100 x n)	0x042D	0x052D	0x062D	0x072D
Sn_FRAGR1		0x042E+(0x0100 x n)	0x042E	0x052E	0x062E	0x072E
Sn_MR2	SOCKET n Mode 2	0x042F+(0x0100 x n)	0x042F	0x052F	0x062F	0x072F
Sn_KPALVTR	SOCKET n Keep-alive Timer	0x0430+(0x0100 x n)	0x0430	0x0530	0x0630	0x0730
Sn_RTR0	SOCKET n Retransmission Time	0x0432+(0x0100 x n)	0x0432	0x0532	0x0632	0x0732
Sn_RTR1		0x0433+(0x0100 x n)	0x0433	0x0533	0x0633	0x0733
Sn_RCR	SOCKET n Retransmission Count	0x0434+(0x0100 x n)	0x0434	0x0534	0x0634	0x0734

3 Register Descriptions

Register Notation

* Register Symbol (Register full Name)

- [Register Type][Address Offset][Reset Value]

Register Description….

7	6	5	4	3	2	1	0
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Bit Type	Bit Type	Bit Type	Bit Type	Bit Type	Bit Type	Bit Type	Bit Type

Sn_IR [3: 0] indicates a Register Symbol [Upper Bit: Lower Bit].

Sn_IR [3: 0] = '0001' indicates Sn_IR [3] = '0', Sn_IR [2] = '0', Sn_IR [1] = '0', Sn_IR [0] = '1'.

[Register/Bit Type]: Type of Register and Bit.

- [RW] : Both reading and writing are possible.
- [R=W] : The value read and written are the same.
- [RO] : Read Only
- [WO] : Write Only
- [W] : Write Only
- [WC] : Cleared by written ‘1’.
- [W0] : Must be written only ‘0’.
- [W1] : Must be written only ‘1’.
- [AC] : Auto Clear
- [1] : Always read ‘1’
- [0] : Always read ‘0’
- [-] : Not available

[Address Offset]: Register Address Offset

[Reset Value]: Default Value.

Ex1) 3.1.1 MR (Mode Register)

[RW][0x0000][0x03]

MR is abbreviation for Mode Register. This Register is possible to be read and written.

The Register Address Offset is ‘0x0000’ and it is set to ‘0x03’ after Reset.

7	6	5	4	3	2	1	0
RST	-	-	PB	PPPoE	-	AI	IND
AC	W0	W0	R=W	R=W	-	1	1

Ex2) MR [RST]

MR [RST] means RST Bit in MR.

Ex3) MR [7:0]

MR [7:0] means the Bits from 7th to 0th bit in MR.

3.1 Common Registers

3.1.1 MR (Mode Register)

[RW][0x0000] [0x03]

MR is used for Reset, PING Block and PPPoE Enable

7	6	5	4	3	2	1	0
RST	-	-	PB	PPPoE	-	-	-

Bit	Symbol	Description
7	RST	**Reset** If this Bit is '1', All W5100S Registers will be initialized. It will be automatically cleared as '0' after 3 SYS_CLK
[6:5]	-	Reserved
4	PB	**PING Response Block** If this Bit is '1', it blocks the Response to a ping request. 1 : Disable PING Response 0 : Enable PING Response
3	PPPoE	**PPPoE Enable** 1 : Enable PPPoE 0 : Disable PPPoE
[2:0]	-	Reserved

3.1.2 GWR (Gateway IP Address Register)

[R=W] [0x0001-0x0004] [0x00]

GWR configures the Gateway Address when NETLCKR (Network Lock Register) is on Unlock Mode.

Ex) GWR = "192.168.0.1"

GWR0(0x0001)	GWR1(0x0002)	GWR2(0x0003)	GWR3(0x0004)
192 (0xC0)	168 (0xA8)	0 (0x00)	1 (0x01)

3.1.3 SUBR (Subnet Mask Register)

[R=W] [0x0005-0x0008] [0x00]

SUBR configures the Subnet Mask Address when NETLCKR (Network Lock Register) is on Unlock Mode.

Ex) SUBR = "255.255.255.255"

SUBR0(0x0005)	SUBR0(0x0006)	SUBR0(0x0007)	SUBR0(0x0008)

255 (0xFF)	255 (0xFF)	255 (0xFF)	255 (0xFF)

3.1.4 SHAR (Source Hardware Address Register)

[R=W] [0x0009-0x000E] [0x00]

SHAR configures the Source MAC Address when NETLCKR (Network Lock Register) is on Unlock Mode.

Ex) SHAR = "11:22:33:AA:BB:CC"

SHAR0(0x0009)	SHAR1(0x000A)	SHAR2(0x000B)
0x11	0x22	0x33
SHAR3(0x000C)	**SHAR4(0x000D)**	**SHAR5(0x000E)**
0xAA	0xBB	0xCC

3.1.5 SIPR (Source IP Address Register)

[R=W] [0x000F-0x0012] [0x00]

SIPR configures the Source IP Address when NETLCKR (Network Lock Register) is on Unlock Mode.

Ex) SIPR = "192.168.0.100"

SIPR0(x000F)	SIPR1(0x0010)	SIPR2(0x0011)	SIPR3(0x0012)
192 (0xC0)	168 (0xA8)	0 (0x00)	100(0x64)

3.1.6 INTPTMR (Interrupt Pending Time Register)

[R=W][0x0013-0x0014][0x0000]

INTPTMR sets internal Interrupt Pending Timer Count. When INTn is de-asserted to High, Timer Count is initialized to INTPTMR and decreased by 1 from initial value to '0' every SYS_CLK x 4. When Interrupt occurs and the corresponding Interrupt Mask is set and INTPTMR is '0', INTn is asserted to Low.

Ex) INTPTMR = 1000(0x03EB)

INTPTMR0(0x0013)	INTPTMR1(0x0014)
0x03	0xEB

3.1.7 IR (Interrupt Register)

[RW] [0x0015] [0x00]

When W5100S or SOCKET n Event occurs, the corresponding Bit in IR is set to '1'. If the Event occurs and the corresponded Interrupt Mask Bit in IMR is set to '1' and internal Interrupt Pending Timer Counter is '0', INTn is asserted to Low. When the Event is cleared or the corresponding Mask Bit is set to '0', INTn is de-asserted to High.

7	6	5	4	3	2	1	0
CONFLICT	UNREACH	PPPTERM	-	S3_INT	S2_INT	S1_INT	S0_INT
WC	WC	WC	-	AC	AC	AC	AC

Bit	Symbol	Description
7	CONFLICT	**IP Conflict** Read 1 : IP Conflict Read 0 : -
6	UNREACH	**Destination Port Unreachable** When receiving the ICMP (Destination port unreachable) packet, this Bit is set as '1'. And Destination Information, IP Address and Port Number, is written on UIPR & UPORTR. Read 1 : Unreachable Packet receive Read 0 : -
5	PPPTERM	**PPPoE Terminated** Read 1 : Received PPPT or LCPT Packet only on PPPoE Read 0 : -
4	-	Reserved
[3:0]	Sn_INT	**SOCKET n Interrupt** Read 1 : Each n-th Bit describes SOCKET n-th Interrupt. Read 0 : When Sn_IR is 0x00, Sn_INT bit is Auto-Clear as '0'

3.1.8 IMR (Interrupt Mask Register)

[R=W] [0x0016] [0x00]

IMR is used for the corresponding IR Bit Mask.

7	6	5	4	3	2	1	0
CNFT	UNREACH	PPPTERM	-	S3_INT	S2_INT	S1_INT	S0_INT
R=W	R=W	R=W		R=W	R=W	R=W	R=W

Bit	Symbol	Description
7	CNFT	**IP Conflict Interrupt Mask** 1 : Enable IP Conflict Interrupt 0 : Disable IP Conflict Interrupt
6	UNREACH	**Destination Port Unreachable Interrupt Mask** 1 : Enable Destination Port Unreachable Interrupt 0 : Disable Destination Port Unreachable Interrupt
5	PPPTERM	**PADT/LCPT Interrupt Mask**

		1 : Enable PADT/LCPT Interrupt 0 : Disable PADT/LCPT Interrupt
4	-	Reserved
3	S3_INT	**SOCKET 3 Interrupt Mask** 1 : Enable SOCKET 3 Interrupt 0 : Disable SOCKET 3 Interrupt
2	S2_INT	**SOCKET 2 Interrupt Mask** 1 : Enable SOCKET 2 Interrupt 0 : Disable SOCKET 2 Interrupt
1	S1_INT	**SOCKET 1 Interrupt Mask** 1 : Enable SOCKET 1 Interrupt 0 : Disable SOCKET 1 Interrupt
0	S0_INT	**SOCKET 0 Interrupt Mask** 1 : Enable SOCKET 0 Interrupt 0 : Disable SOCKET 0 Interrupt

3.1.9 RTR (Retransmission Time Register)

[R=W] [0x0017-0x0018] [0x07D0]

RTR sets initial value of Sn_RTR (SOCKET n Retransmission Time Register). The unit is 100us. RTR and RCR (Retransmission Counter Register) set ARP & TCP Retransmission.
(Ref *4.8 Retransmission*)

Ex) RTR = 5000 (0x1388)
5000*100us = 0.5s

RTR0(0x0017)	RTR1(0x0018)
0x13	0x88

3.1.10 RCR (Retransmission Count Register)

[R=W] [0x0019] [0x07]

If Sn_RCR(SOCKET n Retransmission Count Register) is '0', Sn_RCR is initialized by RCR. RTR and RCR set ARP & TCP Retransmission. (Ref *4.8 Retransmission*)

3.1.11 RMSR (RX Memory Size Register)

[R=W] [0x001A] [0x55]

RMSR configures each SOCKET n RX Buffer Size. And the Sum of SOCKET n RX Buffer Size must not exceeded 8 Kbytes. (Ref *Sn_RXBUF_SIZE (SOCKET n RX Buffer Size Register)*)

7	6	5	4	3	2	1	0
SOCKET 3		SOCKET 2		SOCKET 1		SOCKET 0	

S1	S0	S1	S0	S1	S0	S1	S0

Each SOCKET n RX Buffer Size is set by S0 and S1.

Buffer Size	S1	S0
1 KB	0	0
2 KB	0	1
4 KB	1	0
8 KB	1	1

3.1.12 TMSR (TX Memory Size Register)

[R=W] [0x001B] [0x55]

TMSR configures each SOCKET n TX Buffer Block Size. And the sum of SOCKET n TX Buffer Block Size must not be exceeded 8 Kbytes. (Ref *Sn_TXBUF_SIZE (SOCKET n TX Buffer Size Register)*)

7	6	5	4	3	2	1	0
SOCKET 3		SOCKET 2		SOCKET 1		SOCKET 0	
S1	S0	S1	S0	S1	S0	S1	S0

Each SOCKET n TX Buffer Block Size is set by S0 and S1.

Buffer Size	S1	S0
1 KB	0	0
2 KB	0	1
4 KB	1	0
8 KB	1	1

3.1.13 IR2 (Interrupt Register 2)

[RW] [0x0020] [0x00]

When WOL Event occurs, IR2 [WOL] is set to '1'. If the Event occurs and IMR2 [WOL] is set to '1' and internal Interrupt Pending Timer Counter is '0', INTn is asserted to Low. When the Event is cleared or IMR2 [WOL] is set to '0', INTn is de-asserted to High.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	WOL
-	-	-	-	-	-	-	WC

Bit	Symbol	Description
[7:1]	-	Reserved
0	WOL	**WOL MAGIC Packet Interrupt** 1 : Received UDP based WOL Magic Packet. 0 : -

3.1.14 IMR2 (Interrupt Mask Register 2)

[R=W] [0x0021] [0x00]

IMR2 is used for the corresponding IR2 Bit Mask.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	WOL
-	-	-	-	-	-	-	R=W

Bit	Symbol	Description
[7:1]	-	Reserved
0	WOL	**WOL MAGIC Packet Interrupt Mask** 1 : Enable WOL MAGIC Packet Interrupt 0 : Disable WOL MAGIC Packet Interrupt

3.1.15 PTIMER (PPP Link Control Protocol Request Timer Register)

[R=W] [0x0028] [0x28]

PTIMER configures the sending period for LCP Echo Request. The unit is 25ms.

Ex) PTIMER = 200 (0xC8),

200 * 25ms = 5s

3.1.16 PMAGIC (PPP Link Control Protocol Magic number Register)

[R=W] [0x0029] [0x00]

PMAGIC configures 4 Bytes Magic Number for LCP Echo Request.

Ex) PMAGIC = 0x01

PMAGIC(0x0029)
0x01

LCP Magic Number = 0x01010101

3.1.17 UIPR (Unreachable IP Address Register)

[RO] [0x002A-0x002D] [0x0000]

When W5100S received Unreachable Packet (IR [UNR] = '1'), Peer IP Address in the Packet is written on UIPR.

Ex) UIPR = "192.169.0.21"

UIPR0(0x002A)	UIPR1(0x002B)	UIPR2(0x002C)	UIPR3(0x002D)
192(0xC0)	168(0xA8)	0(0x00)	21(0x15)

3.1.18 UPORTR (Unreachable Port Register)

[RO] [0x002E-0x002F] [0x0000]

When W5100S received Unreachable Packet (IR [UNR] = '1'), Peer PORT Number in the Packet is written on UPORTR.

Ex) UPORTR = 3000 (0x0BB8)

UPORTR0(0x002E)	UPORTR1(0x002F)
0x0B	0xB8

3.1.19 MR2 (Mode Register 2)

[R=W] [0x0030] [0x40]

MR2 configures System Operation Clock (SYS_CLK), Interrupt Activation, TCP & UDP Scan Prevention, WOL (Wake On LAN) and Force ARP.

7	6	5	4	3	2	1	0
CLKSEL	IEN	NOTCPRST	UDPURB	WOL	-	FARP	-
R=W	R=W	R=W	R=W	R=W	W0	R=W	W0

Bit	Symbol	Description
7	CLKSEL	**System Operation Clock(SYS_CLK) Select** When CLKLCKR (Clock Lock Register) is Unlock, this bit can only set. (Ref *3.1.41 CLKLCKR (Clock Lock Register*) ***[WO] [0x0070] [0x00]***) 1 : 25MHz 0 : Depends on PHYCR1[PWDN] SYS_CLK selected by PHCR1 configuration.

PHYCR1[PWDN]	SYS_CLK
0	100 MHz

		1	25 MHz
6	IEN	**INTn Enable / Disable** 1 : INTn Enable 0 : INTn Disable (INTn is Always High)	
5	NOTCPRST	**TCP RST Packet Block** If Peer transmits TCP Packet to a Port that does not exist on W5100S. W5100S automatically transmits RST Packet. But it could be the target for Port Scan Attack. But if this Bit is set as '1', W5100S does not transmit RST Packet against Peer TCP Packet having wrong port. 1 : Block sending RST Packet 0 : Normal	
4	UDPURB	**UDP Port Unreachable Packet Block** If Peer transmits UDP Packet to a Port that does not exist on W5100S. W5100S automatically transmits ICMP Packet (Destination Port Unreachable) to Peer. But it could be the target for Port Scan Attack. But if this Bit is set as '1', W5100S does not transmit ICMP Packet (Destination Port Unreachable). 1 : Block sending ICMP Packet (Destination Port Unreachable Message) 0 : Normal	
3	WOL	**Wake On LAN** This Bit decides receiving WOL Packet. 1 : Receive WOL 0 : No Receive WOL	
2	-	Reserved	
1	FARP	**Force ARP** UDP Mode SOCKET transmits ARP Request once before the first UDP Data Packet when it sends Data continuously to the same Peer. But if this Bit is set to '1', it sends ARP Request for every UDP Data Packet. 1 : Transmit ARP Request for every UDP Data Packet 0 : Normal	
0	-	Reserved	

3.1.20 PHAR (Destination Hardware Address Register on PPPoE)

[R=W] [0x0032-0x0037] [0x0000]

PHAR configures PPPoE Server Hardware Address only on PPPoE.

Ex) PHAR = “11:22:33:AA:BB:CC”

PHAR0(0x0032)	PHAR1(0x0033)	PHAR2(0x0034)
0x11	0x22	0x33
PHAR3(0x0035)	PHAR4(0x0036)	PHAR5(0x0037)
0xAA	0xBB	0xCC

3.1.21 PSIDR (Session ID Register on PPPoE)

[R=W] [0x0038-0x0039] [0x0000]

PSID configures PPPoE Sever Session ID on PPPoE.

Ex) PSIDR = 0x1234

PSIDR0(0x0038)	PSIDR1(0x0039)
0x12	0x34

3.1.22 PMRUR (PPPoE Maximum Receive Unit Register)

[R=W] [0x003A-0x003B] [0xFFFF]

PMRUR configures MRU (Maximum Receive Unit) on PPPoE and the value must not be exceeded 1472. If it is set bigger than 1472, it is configured 1472. And also PMRUR must be configured before SOCKET OPEN (Sn_CR [OPEN] = ’1’).

Ex) PMUR = 1000 (0x03E8)

PMUR0(0x0038)	PMUR1(0x0039)
0x03	0xE8

3.1.23 PHYSR0 (PHY Status Register0)

[RO] [0x003C] [0x00]

PHYSR0 indicates PHY Operation Mode and LINK status configured by PHYCR0 (PHY Control Register 0)

7	6	5	4	3	2	1	0
CABOFF	-	AUTO	SPD	DPX	FDPX	FSPD	LINK
RO		RO	RO	RO	RO	RO	RO

Bit	Symbol	Description
7	CABOFF	**Cable OFF Bit** 1 : Cable Unplugged

		0 : Cable Plugged
6	-	Reserved
5	AUTO	**Auto Negotiation Bit configured by PHYCR0[2]** 1 : Disable Auto Negotiation 0 : Enable Auto Negotiation
4	SPD	**Speed Bit configured by PHYCR0[1]** 1 : 10Mbps 0 : 100Mbps
3	DPX	**Duplex Bit configured by PHYCR0[0]** 1 : Half Duplex 0 : Full Duplex
2	FDPX	**Flag Duplex Bit (When Link up)** 1 : Half Duplex 0 : Full Duplex
1	FSPD	**Flag Speed Bit (When Link up)** 1 : 10Mbps 0 : 100Mbps
0	LNK	**Flag Link Bit** 1 : Link Up 0 : Link Down

3.1.24 PHYSR1 (PHY Status Register 1)

[RO][0x003D][0x81]

PHYSR1 indicates PHY Active state, received RX Page, LPI Mode and Calibrate.

7	6	5	4	3	2	1	0
ACT					RXP	LPI	CAL
RO					RO	RO	RO

Bit	Symbol	Description
7	ACT	**PHY Act** 1 : PHY Act 0 : PHY No Act
[6:3]	-	Reserved
2	RXP	**RX PAGE** 1 : A New Page Received 0 : A New Page is not Received
1	LPI	**LPI(Low Power Idle) Mode** 1 : Enable LPI Mode

		0 : Disable LPI Mode
0	CAL	**Calibrate Done** 1 : Calibration is Removed 0 : Normal

3.1.25 PHYRAR (PHY Register Address Register)

[R=W] [0x003F] [0x00]

PHYRAR configures PHY Register Address of Internal Ethernet PHY.

7	6	5	4	3	2	1	0
-	-	-	A4	A3	A2	A1	A0
			R=W	R=W	R=W	R=W	R=W

Bit	Symbol	Description
[7:5]	-	Reserved
[4:0]	A[4:0]	**PHY Register Address**

3.1.26 PHYDIR (PHY Data Input Register)

[R=W] [0x0040-0x0041] [0x0000]

PHTDIR writes PHY Register specified by PHYAR.

Ex) PHYDIR = 0x1234

PHYDIR0(0x0040)	PHYDIR1(0x0041)
0x34	0x12

3.1.27 PHYDOR (PHY Data Output Register)

[RO] [0x0042-0x0043] [0x0000]

PHYDOR reads the PHY Register specified by PHYAR

Ex) PHYDOR = 0x1234

PHYDOR0(0x0042)	PHYDPR1(0x0043)
0x34	0x12

3.1.28 PHYACR (PHY Access Control Register)

[AC] [0x0044] [0x00]

PHYACR configures Access Type of PHY Register specified by PHYAR

Access Type	Value

Write	0x01
Read	0x02

3.1.29 PHYDIVR (PHY Division Register)

[R=W] [0x0045] [0x01]

Internal Ethernet PHY uses the divided Clock of System Operation Clock (SYS_CLK). And this divided Clock must not be exceeded 2.5MHz.

Value	Divider	SYS_CLK=100MHz	SYS_CLK=25MH
0x00	1/32	3.125MHz (N/A)	781.25KHz
0x01	1/64	1.5625MHz	390.625KHz
others	1/128	781.25KHz	195.3125KHz

3.1.30 PHYCR0 (PHY Control Register 0)

[WO] [0x0046] [0x00]

PHYCR configures Ethernet PHY Operation Mode such as Auto Negotiation, Speed and Duplex. Before set PHYCR, PHYLCKR (PHY Lock Register) must be on Unlock Mode.

7	6	5	4	3	2	1	0
-	-	-	-	-	MODE2	MODE1	MODE0
					WO	WO	WO

Bit	Symbol	Description
[7:3]	-	Reserved
[2:0]	MODE	

MODE2	MODE1	MODE0	Description
0	x	x	Auto Negotiation
1	0	0	100BASE-TX FDX
1	0	1	100BASE-TX HDX
1	1	0	10BASE-TX FDX
1	1	1	10BASE-TX HDX

3.1.31 PHYCR1 (PHY Control Register 1)

[R=W] [0x0047] [0x41]

PHYCR1 configures Ethernet PHY Operation Mode such as PHY Power Down Mode, PHY Reset. Before set PHYCR1, PHYLCKR (PHY Lock Register) must be on Unlock Mode.

7	6	5	4	3	2	1	0
WOL	-	PWDN	-	-	-	-	RST

W0	W0	R=W	W0	W0	W0	W0	AC

Bit	Symbol	Description
7	WOL	**Wake On LAN** To receive WOL Packet, PHYCR1[WOL] and MR2[WOL] must be set by '1'. 1 : Receive WOL Packet 0 : No Receive WOL Packet
6	-	Reserved
5	PWDN	**PHY Power Down** 1 : Enable Power Down Mode and SYS_CLK switching 25MHz 0 : Disable Power Down Mode and SYS_CLK selected by MR2[CLKSEL] MR2[CLKSEL] / SYS_CLK 0 / 100 MHz 1 / 25 MHz (Ref *7.4.1 Reset Timing*)
[4:1]	-	Reserved
0	RST	**PHY Reset** When PHY Reset Bit is set to '0', SYS_CLK is switched to 25MHz. After PHY Reset completed(PHYSR1[ACT]='1'), this Bit is automatically cleared and SYS_CLK is turn back to the previous clock. (Ref *7.4.1 Reset Timing*) 1 : PHY H/W Reset 0 : Normal

3.1.32 SLCR (SOCKET-less Command Register)

[RW] [0x004C] [0x00]

SLCR configures ARP and PING Request Transmission Command. Each Command must not be executed at the same time and not configured before SLCR cleared. The results of each Command is shown via SLIR (SOCKET-less Interrupt Register)

7	6	5	4	3	2	1	0
-	-	-	-	-	-	ARP	PING
-	-	-	-	-	-	AC	AC

Bit	Symbol	Description
[7:2]	-	Reserved
1	ARP	**ARP Request Transmission Command** 1 :Transmit ARP packet 0 : Ready
0	PING	**PING Request Transmission Command** 1: Transmit PING Request 0 : Ready

3.1.33 SLRTR (SOCKET-less Retransmission Time Register)

[R=W] [0x004D-0x004E] [0x07D0]

SLRTR sets SLCR Retransmission Time. The unit is 100us. If there is no Response for ARP or PING Request Packet transmitted by SLCR, W5100S automatically retransmits Request Packet every SLRTR. (Ref *4.8 Retransmission*)

Ex) SLRTR = 5000 (0x1388),

5000 * 100us = 0.5s

SLRTR0(0x004D)	SLRTR1(0x004E)
0x013	0x88

3.1.34 SLRCR (SOCKET-less Retransmission Count Register)

[R=W] [0x004F] [0x00]

SLRCR sets SLCR Retransmission Number. If the number of Retransmission exceeds SLRCR, SOCKET-less Timeout (SLIR [TIMEOUT] = '1') occurs. (Ref *4.8 Retransmission*)

3.1.35 SLPIPR (SOCKET-less Peer IP Address Register)

[R=W] [0x0050-0x0053] [0x00000000]

SLPIPR configures Peer IP Address for ARP or Ping Request Packet transmitted by SLCR.

Ex) SLPIPR = "192.169.0.21"

SLPIPR0(0x0050)	SLPIPR1(0x0051)	SLPIPR2(0x0052)	SLPIPR3(0x0053)
192(0xC0)	168(0xA8)	0(0x00)	21(0x15)

3.1.36 SLPHAR (SOCKET-less Peer Hardware Address Register)

[RO] [0x0054-0x0059] [0x000000000000]

If W5100S received ARP Reply against SLCR [ARP] and SLIPR [ARP] is set to '1', Peer Hardware Address from ARP Reply is written on SLPHAR.

Ex) SLPHAR = "11:22:33:AA:BB:CC"

SLPHAR0(0x0054)	SLPHAR1(0x0055)	SLPHAR2(0x0056)
0x11	0x22	0x33
SLPHAR3(0x0057)	SLPHAR4(0x0058)	SLPHAR5(0x0059)
0xAA	0xBB	0xCC

3.1.37 PINGSEQR (PING Sequence-number Register)

[R=W] [0x005A-0x005B] [0x0000]

PINGSEQR configures Sequence Number for PING Request Packet and it is not automatically increased.

Ex) PINGSEQR = 1000 (0x03E8)

PINGSEQR0(0x005A)	PINGSEQR1(0x005B)
0x03	0xE8

3.1.38 PINGIDR (PING ID Register)

[R=W] [0x005C-0x005D] [0x0000]

PINGIDR configures ID for PING Request Packet.

Ex) PINGIDR = 256 (0x0100)

PINGIDR0(0x005C)	PINGIDR1(0x005D)
0x01	0x00

3.1.39 SLIMR (SOCKET-less Interrupt Mask Register)

[R=W] [0x005E] [0x00]

SLIMR is used for corresponding SLIR (SOCKET-less Interrupt Register) Bit Mask.

7	6	5	4	3	2	1	0
-	-	-	-	-	TIMEOUT	ARP	PING
-	-	-	-	-	R=W	R=W	R=W

Bit	Symbol	Description
[7:3]	-	Reserved
2	TIMEOUT	**TIMEOUT Interrupt Mask** 1 : Enable TIMEOUT Interrupt 0 : Disable TIMEOUT Interrupt
1	ARP	**ARP Interrupt Mask** 1 : Enable ARP Interrupt 0 : Disable ARP Interrupt

0	PING	**PING Interrupt Mask** 1 : Enable PING Interrupt 0 : Disable PING Interrupt

3.1.40 SLIR (SOCKET-less Interrupt Register)

[RW] [0x005F] [0x00]

When SOCKET-less Event occurs, the corresponding Bit in SLIR is set to '1'. If the Event occurs and the corresponded Interrupt Mask Bit in SLIMR is set to '1' and internal Interrupt Pending Timer Counter is '0', INTn is asserted to Low. When the Event is cleared or the corresponding Mask Bit is set to '0', INTn is de-asserted to High.

7	6	5	4	3	2	1	0
-	-	-	-	-	TIMEOUT	ARP	PING
-	-	-	-	-	WC	WC	WC

Bit	Symbol	Description
[7:3]	-	Reserved
2	TIMEOUT	**TIMEOUT Interrupt** When TIMEOUT occurs, this Bit is set to '1'.
1	ARP	**ARP Interrupt** When ARP Reply received, this Bit is set to '1'.
0	PING	**PING Interrupt** When PING Reply received, this Bit is set to '1'.

3.1.41 CLKLCKR (Clock Lock Register)

[WO] [0x0070] [0x00]

CLKLCKR initial status is Lock. To set MR2[CLKSEL], it must be set as Unlock(0xCE). Before HOST changes CLKLCKR status, CLKLCKR has the previous status.

Unlock	Lock
0xCE	Others

3.1.42 NETLCKR (Network Lock Register)

[WO] [0x0071] [0x00]

NETLCKR initial status is Unlock. In Unlock status, GWR, SUBR, SHAR and SIPR are set by HOST. Before HOST changes NETLCKR as Lock(0xC5), LETLCKR is in Unlock status.

Unlock	Lock

0x3A	0xC5

3.1.43 PHYLCKR (PHY Lock Register)

[WO] [0x0072] [0x00]

PHYLCKR initial status is Lock. To set PHYCR0 and PHYCR1, it must be set as Unlock(0x53). Before HOST changes PHYLCKR status, PHYLCKR has the previous status.

Unlock	Lock
0x53	Others

3.1.44 VERR (Version Register)

- [RO] [0x0080] [0x51]

VERR shows W5100S Version.

3.1.45 TCNTR (Ticker Counter Register)

[RO][0x0082-0x0083][0x0000]

TCNTR is W5100S Internal Counter, it has automatically increased since SYS_CLK operating. The unit is 100us.

3.1.46 TCNTCLR (Ticker Counter Clear Register)

[WO][0x0088][0x00]

With TCNTCLR Write Access, TCNTR Counter value is initialized.

3.2 SOCKET Register

3.2.1 Sn_MR (SOCKET n Mode Register)

[R=W] [0x0000+0x0100*(n+4)] [0x00]

Sn_MR configures SOCKET Mode and Option. Sn_MR must be set before SOCKET OPEN (Sn_CR [OPEN] = '1').

7	6	5	4	3	2	1	0
MULTI	MF	ND / MC	-	P3	P2	P1	P0
R=W	R=W	R=W	-	R=W	R=W	R=W	R=W

Bit	Symbol	Description
7	MULTI	**UDP Multicast** This Bit is valid only on UDP Mode. (Ref *4.4.3 UDP Multicast*) 1 : Enable UDP Multicast 0 : Disable UDP Multicast
6	MF	**MAC Filter Enable** This Bit is valid only on MACRAW Mode. If This Bit is set to '1', W5100S block all Packets without Multicast, Broadcast and the Packets no having Source MAC (SHAR). 1 : Enable MAC Filter. 0 : Disable MAC Filter
5	ND / MC	**No Delayed ACK (ND)** This Bit is valid only on TCP Mode. If this Bit is set to '1', TCP Mode SOCKET transmits ACK Packet without waiting RTR after receiving Data Packet from Peer. 1 : Enable No Delayed ACK 0 : Disable No Delayed ACK Ref) After Sn_CR[RECV] Command operating, If TCP Mode SOCKET Window Size is smaller than MSS, it sends ACK packet immediately. (no concerned with ND Bit) **Multicast IGMP Version (MC)** This Bit is valid only on UDP Multicast Mode (Sn_MR[3:0] = 'UDP' & Sn_MR[MULTI] = '1')

		0 : Using IGMP version 2 1 : Using IGMP version 1
4	-	Reserved
[3:0]	P[3:0]	**Protocol Mode** This Bits set SOCKET Protocol Mode. MACRAW Mode is used only with SOCKET 0.

P3	P2	P1	P0	Protocol Mode
0	0	0	0	SOCKET Closed
0	0	0	1	TCP
0	0	1	0	UDP
0	0	1	1	IPRAW
0	1	0	0	MACRAW

3.2.2 Sn_CR (SOCKET n Command Register)

[RW][AC] [0x0001+0x0100*(n+4)] [0x00]

Sn_CR configures SOCKET n Command. After W5100S executes SOCKET Command, the corresponding Sn_CR Bit is automatically cleared. Next SOCKET Command must be configured after previous SOCKET Command Bit cleared.

Value	Symbol	Description
0x01	OPEN	**SOCKET OPEN Command** Before OPEN Command, HOST must be set SOCKET Mode with Sn_MR. After OPEN Command done, Sn_SR shows SOCKET Status.
0x02	LISTEN	**TCP LISTEN Command** After LISTEN Command, TCP Mode SOCKET waits for SYN Packet from Peer for TCP Connection in SOCK_INIT (Sn_SR = '0x13'). (Ref *4.3.1 TCP Server*)
0x04	CONNECT	**TCP CONNECT Command** After CONNECT Command, TCP Mode SOCKET transmits SYN Packet to Peer for TCP Connection in SOCK_INIT (Sn_SR = '0x13').

Sn_MR (P[3:0])	Sn_SR
Sn_MR_CLOSE ('0000')	SOCK_CLOSED (0x00)
Sn_MR_TCP ('0001')	SOCK_INIT (0x13)
Sn_MR_UDP ('0010')	SOCK_UDP (0x22)
Sn_MR_IPRAW ('0011')	SOCK_IPRAW (0x32)
S0_MR_MACRAW ('0100')	SOCK_MACRAW (0x42)

		(Ref *4.3.2 TCP Client*)
0x08	DISCON	**TCP DISCON Command** After DISCON Command, TCP Mode SOCKET transmits FIN Packet to Peer for TCP Disconnection in SOCK_ESTABLESHED (Sn_SR = '0x17') or SOCK_CLOSE_WAIT (Sn_SR = '0x1C').
0x10	CLOSE	**SOCKET CLOSE Command** After CLOSE Command, SOCKET is closed (Sn_SR = '0x00'). **Caution : Sn_SR is changed to SOCK_CLOSE without FIN Packet sending in TCP Mode*
0x20	SEND	**SOCKET SEND Command** After SEND Command, SOCKET sends Data as much as the calculated size between Sn_TX_WR (SOCKET n TX Write Point Register) and Sn_TX_RD (SOCKET n RX Read Pointer Register). The Sent Data must not be exceeded Sn_RX_FSR (SOCKET n TX Free Buffer Size Register). HOST must execute next SEND Command after Sn_IR [SENDOK] is set to '1'. On TCP and UDP Mode, if the calculated size is over MSS (Maximum Segment Size), Data is separated by MSS and sent. On the other hands, on IPRAW and MACRAW Mode, if Data is over MSS, HOST must separate it by less than MSS. On TCP Mode, if Peer receives Data (It means TCP Mode SOCKET receives ACK Packet from Peer), Sn_RX_FSR is automatically increased by sent Data Size. But if not, Sn_IR [TIMEOUT] is set to '1' and Sn_SR is changed to SOCK_CLOSED. On UDP, IPRAW and MACRAW Mode, Sn_TX_FSR is increased by sent Data Size after Sn_IR [SENDOK] is set to '1'.
0x21	SEND_MAC	**SOCKET SEND_MAC Command** SEND_MAC Command is used only on UDP and IPRAW Mode. Only the different with SEND Command is that it skips ARP Process. Before using SEND_MAC Command, HOST must set Sn_DHAR (SOCKET n Destination Hardware Address Register) with Peer MAC Address.
0x22	SEND_KEEP	**SOCKET SEND_KEEP Command** SEND_KEEP Command is used only on TCP Mode. Before using SEND_KEEP Command, TCP Mode SOCKET must send more than 1 byte Data to Peer. After SEND_KEEP Command, TCP Mode SOCKET continues to send Keep alive (KA) Packet and check the TCP connection. If there is no Peer

		response, KA Packet is retransmitted. After Retransmission Time, Sn_IR [TIMEOUT] is set to '1' and Sn_SR is changed to SOCK_CLOSED. (Ref 4.3.3.2 Keep AliveKeep)
0x40	RECV	**SOCKET RECV Command** After reading received Data from SOCKET n RX Buffer Block, HOST updates Sn_RX_RD (SOCKET n Read Pointer Register) with RECV Command. (Ref *3.2.18 Sn_RX_RSR (SOCKET n RX Received Size Register), 3.2.20 Sn_RX_WR (SOCKET n RX Write Pointer Register), 3.2.16 Sn_RX_RD (SOCKET n RX Read Pointer Register)*)

3.2.3 Sn_IR (SOCKET n Interrupt Register)

[RW] [0x0002+0x0100*(n+4)] [0x00]

When Sn_IR Event occurs, the corresponding Interrupt Bit is set to '1'.

7	6	5	4	3	2	1	0
-	-	-	SENDOK	TIMEOUT	RECV	DISCON	CON
			WC	WC	WC	WC	WC

Bit	Symbol	Description
[7:5]	-	Reserved
4	SENDOK	**SEND OK Interrupt** 1 : When SEND Command is completed 0 : When others
3	TIMEOUT	**TIMEOUT Interrupt** 1 : When the number of retransmission is exceeded Sn_RCR (SOCKET Retransmission Count Register) in ARP or TCP communication. 0 : When others
2	RECV	**RECEIVED Interrupt** 1 : When SOCKET received Data or Data still remained in SOCKET n RX Buffer Block after Sn_CR [RECV] 0 : When others
1	DISCON	**DISCONNECTED Interrupt** 1: When FIN Packet sent and disconnection completed, or When FIN Packet received or RST Packet received. 0 : When others
0	CON	**CONNECTED Interrupt** 1 : When TCP Connection is successfully done. (Sn_MR [3:0]='TCP)

		0 : When others

3.2.4 Sn_SR (SOCKET n Status Register)

[RO] [0x0003+0x0100*(n+4)] [0x00]

Sn_SR describes the Status of SOCKET. Sn_SR is set by Sn_CR or Data transmit/receive.

Value	Symbol	Description
0x00	SOCK_CLOSED	SOCKET Closed
0x13	SOCK_INIT	SOCKET Opened as TCP Mode
0x14	SOCK_LISTEN	SOCKET is TCP Mode and wait for Peer Connection
0x17	SOCK_ESTABLISHED	SOCKET is TCP Mode and TCP Connection is done
0x1C	SOCK_CLOSE_WAIT	SOCKET is TCP Mode and received disconnection request
0x22	SOCK_UDP	SOCKET Opened as UDP Mode
0x32	SOCK_IPRAW	SOCKET Opened as IPRAW Mode
0x42	SOCK_MACRAW	SOCKET Opened as MACRAW Mode

The below table shows a temporary status indicated during changing the status of SOCKET n.

Value	Symbol	Description
0x15	SOCK_SYNSENT	SOCKET n sent the Connect request
0x16	SOCK_SYNRECV	SOCKET n received the Connect request
0x18	SOCK_FIN_WAIT	SOCKET n is in SOCKET Closed
0X1B	SOCK_TIME_WAIT	
0X1D	SOCK_LAST_ACK	

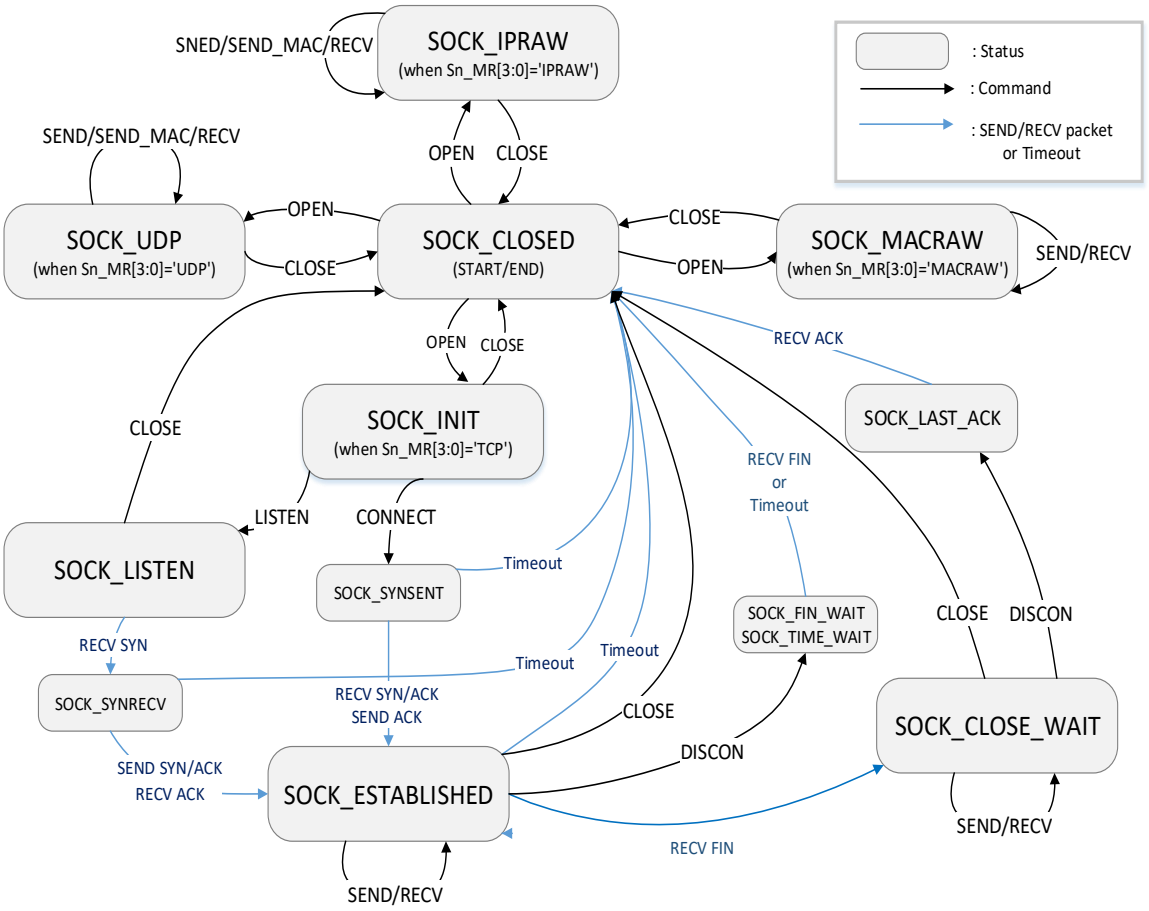


Figure 4 State Diagram

3.2.5 Sn_PORTR (SOCKET n Source Port Register)

[R=W] [0x0004+0x0100*(n+4), 0x0005+0x0100*(n+4)] [0x0000]

Sn_PORTR configures Source Port Number of SOCKET n.

Ex) S0_PORTR = 5000 (0x1388)

S0_PORTR0(0x0404)	S0_PORTR1(0x0405)
0x013	0x88

3.2.6 Sn_DHAR (SOCKET n Destination Hardware Address Register)

[RW] [0x0006+0x0100*(n+4), 0x0007+0x0100*(n+4), 0x0008+0x0100*(n+4), 0x0009+0x0100*(n+4), 0x000A+0x0100*(n+4), 0x000B+0x0100*(n+4)] [0x000000000000]

Peer MAC Address in Peer Packet is written on Sn_DHAR. On TCP Mode, Sn_DHAR is valid after CONNECT Command. On UDP and IPRAW Mode, Sn_DHAR is valid after SEND Command.

If HOST uses SEND_MAC Command, Sn_DHAR must be set with Peer MAC Address before Command. On UDP Multicast Mode, Sn_DHAR must be configured with Multicast Group MAC

Address. (Ref *4.4.3 UDP Multicast*)

Ex) S0_DHAR = "11:22:33:AA:BB:CC"

S0_DHAR0(0x0406)	S0_DHAR1(0x0407)	S0_DHAR2(0x0408)
0x11	0x22	0x33
S0_DHAR3(0x0409)	S0_DHAR4(0x040A)	S0_DHAR5(0x040B)
0xAA	0xBB	0xCC

3.2.7 Sn_DIPR (SOCKET n Destination IP Address Register)

[RW] [0x000C+0x0100*(n+4), 0x000D+0x0100*(n+4), 0x000E+0x0100*(n+4), 0x000F+0x0100*(n+4)] [0x00000000]

Sn_DIPR is set with Peer IP Address. On TCP Mode, Sn_DIPR must be set with Peer IP Address before Connection, or indicates IP Address of connected Peer. On UDP and IPRAW Mode, Sn_DIPR must be set with Peer IP Address before sending Data.

On UDP Multicast Mode, Sn_DIPR must be configured with Multicast Group IP Address. (Ref *4.4.3 UDP Multicast*)

On UDP, UDP Multicast and IPRAW Mode, Peer IP Address is stored with Data in SOCKET n RX Buffer.

Ex) S0_DIPR = "192.168.0.11"

S0_DIPR0(0x040C)	S0_DIPR1(0x040D)	S0_DIPR2(0x040E)	S0_DIPR3(0x040F)
192 (0xC0)	168 (0xA8)	0 (0x00)	11 (0x0B)

3.2.8 Sn_DPORTR (SOCKET n Destination Port Register)

[R=W] [0x0010+0x0100*(n+4), 0x0011+0x0100*(n+4)] [0x0000]

Sn_DPORTR is set with Peer Port Number. On TCP Mode, Sn_DPORTR must be set with Peer Port Number before Connection, or indicates Port Number of connected Peer. On UDP and IPRAW Mode, Sn_DPORTR must be set with Peer Port Number before sending Data.

On UDP Multicast Mode, Sn_DPORTR must be configured with Multicast Group Port Number. (Ref *4.4.3 UDP Multicast*)

On UDP, UDP Multicast and IPRAW Mode, Peer Port Number is stored with Data in SOCKET n RX Buffer.

Ex) S0_DPORTR = 5000 (0x1388),

S0_DPORTR0(0x0410)	S0_DPORTR1(0x0411)
0x13	0x88

3.2.9 Sn_MSS (SOCKET n Maximum Segment Size Register)

[RW] [0x0012+0x0100*(n+4), 0x0013+0x0100*(n+4] [0x0000]

Sn_MSS sets SOCKET n MSS and it must be set before Sn_CR [OPEN]. Sn_MSS must not be exceeded specified size as below table.

Mode	Normal (MR [PPPoE]='0') Range	PPPoE (MR [PPPoE]='1') Range
TCP	1~1460	1~1452
UDP	1~1472	1~1464
IPRAW	1480	1472
MACRAW	1514	

Ex) S0_MSS = 1460 (0x05B4),

S0_MSS0(0x0412)	S0_MSS1(0x0413)
0x05	0xB4

3.2.10 Sn_PROTOR (SOCKET n IP Protocol Register)

[R=W] [0x0014+0x0100*(n+4)] [0x0000]

Sn_PROTOR configures Protocol Number (Ref *IANA_Protocol Number*) of IP Header except for TCP (0x06), UDP (0x11) and IGMP (0x01).

On IPRAW Mode, only the Protocol configured in Sn_PROTR can be transmitted and received.

Ex) ICMP (Internet Control Message Protocol) = 0x01

3.2.11 Sn_TOS (SOCKET n IP Type Of Service Register)

[R=W] [0x0015+0x0100*(n+4)] [0x00]

Sn_TOS configures the TOS (Type Of Service) of IP Header. (Ref *IANA_IP Parameters*)

3.2.12 Sn_TTL (SOCKET n IP Time To Live Register)

[R=W] [0x0016+0x0100*(n+4)] [0x80]

Sn_TTL configures TTL (Time To Live) of IP header. (Ref *IANA_IP Parameters*)

3.2.13 Sn_RXBUF_SIZE (SOCKET n RX Buffer Size Register)

[RW] [0x001E+0x0100*(n+4)] [0x02]

Sn_RXBUF_SIZE configures SOCKET n RX Buffer Size as 0, 1, 2, 4 and 8 Kbytes. RX Memory is sequentially allocated from SOCKET 0 to SOCKET 3. The total sum of Sn_RXBUF_SIZE must not be exceeded 8 Kbytes. Sn_RXBUF_SIZE can also be configured by RMSR.

Value (Dec)	0	1	2	4	8
Buffer size	0KB	1KB	2KB	4KB	8KB

Ex) S0_RXBUF_SIZE = 8KB

S0_RXBUF_SIZE(0x041E)
0x08

3.2.14 Sn_TXBUF_SIZE (SOCKET n TX Buffer Size Register)

[RW] [0x001F+0x0100*(n+4)] [0x02]

Sn_TXBUF_SIZE configures SOCKET n TX Buffer Size as 0, 1, 2, 4 and 8 Kbytes. TX Memory is sequentially allocated from SOCKET 0 to SOCKET 3. The total sum of Sn_TXBUF_SIZE must not be exceeded 8 Kbytes. Sn_TXBUF_SIZE can also be configured by RMSR.

Value (Dec)	0	1	2	4	8
Buffer size	0KB	1KB	2KB	4KB	8KB

Ex) S0_TXBUF_SIZE= 4KB

S0_TXBUF_SIZE(0x041F)
0x04

3.2.15 Sn_TX_FSR (SOCKET n TX Free Size Register)

[RO] [0x0020+0x0100*(n+4), 0x0021+0x0100*(n+4)] [0x0800]

Sn_TX_FSR indicates SOCKET n TX Free Buffer Size.

In UDP, IPRAW and MACRAW mode,

Sn_TX_FSR = | Sn_TX_WR(1) – Sn_TX_RD(2) | + 1

In TCP mode,

Sn_TX_FSR = | Sn_TX_WR – Internal Pointer(3) | + 1

(1) SOCKET n TX Write Pointer Register
(2) SOCKET n TX Read Pointer Register
(3) TCP ACK Pointer managed by W5100S

Data to be stored in SOCKET n TX Buffer must not be bigger than Sn_TX_FSR.

Ex) S0_TX_FSR = 1024 (0x0400)

S0_TX_FSR0(0x0420)	S0_TX_FSR1(0x0421)
0x04	0x00

3.2.16 Sn_TX_RD (SOCKET n TX Read Pointer Register)

[RO] [0x0022+0x0100*(n+4), 0x0023+0x0100*(n+4)] [0x0000]

Sn_TX_RD indicates the last Data Address in SOCKET n TX Buffer Block after transmitting. Sn_TX_RD is initialized by Sn_CR [OPEN]. On TCP Mode, Sn_TX_RD is re-configured in TCP Connection Process. By Sn_CR [SEND] and Sn_CR [SEND_MAC], Sn_TX_WR (SOCKET n TX Write

Pointer Register) is increased to Data Size. After transmitting Data, Sn_TX_RD increases by Sn_TX_WR and Sn_IR [SENDOK] occurs.

Ex) S0_TX_RD = 0xd4b3

S0_TX_RD0(0x0422)	S0_TX_RD1(0x0423)
0xd4	0xb3

3.2.17 Sn_TX_WR (SOCKET n TX Write Pointer Register)

[RW] [0x0024+0x0100*(n+4), 0x0025+0x0100*(n+4)] [0x0000]

Sn_TX_WR indicates the last stored Data Address in SOCKET n TX Buffer Block. Sn_TX_RD is initialized by Sn_CR [OPEN]. On TCP Mode, Sn_TX_RD is re-configured in TCP Connection Process. Before Sn_CR [SEND] and Sn_CR [SEND_MAC], Sn_TX_WR must be set as transmitting Data Size.

Ex) S0_TX_WR = 0x0800

S0_TX_WR0(0x0424)	S0_TX_WR1(0x0425)
0x08	0x00

3.2.18 Sn_RX_RSR (SOCKET n RX Received Size Register)

[RO] [0x0026+0x0100*(n+4), 0x0027+0x0100*(n+4)] [0x0000]

Sn_RX_RSR indicates received Data Size in SOCKET n RX Buffer Block.

In TCP, UDP, IPRAW and MACRAW mode,

$$Sn\_RX\_RSR = | Sn\_RX\_WR^{(1)} - Sn\_RX\_RD^{(2)} |$$

(1) SOCKET n RX Write Pointer Register
(2) SOCKET n RX Read Pointer Register

Ex) S0_RX_RSR = 2048 (0x0800)

S0_RX_RSR0(0x0426)	S0_RX_RSR1(0x0427)
0x08	0x00

3.2.19 Sn_RX_RD (SOCKET n RX Read Pointer Register)

[RW] [0x0028+0x0100*(n+4), 0x0029+0x0100*(n+4)] [0x0000]

Sn_RX_RD is the last Data Address read by HOST in SOCKET n RX Buffer Block. HOST is available to read Data from Sn_RX_RD to Sn_RX_WR in SOCKET n RX Buffer Block. After setting Sn_RX_RD as much as Data read, HOST must set Sn_CR [RECV] to update Sn_RX_RD.

Ex) S0_RX_RD =1536(0x0600)

S0_RX_RD0(0x0428)	S0_RX_RD1(0x0429)
0x06	0x00

3.2.20 Sn_RX_WR (SOCKET n RX Write Pointer Register)

[RO] [0x002A+0x0100*(n+4), 0x002B+0x0100*(n+4)] [0x0000]

Sn_RX_WR is the last received Data Address in SOCKET n RX Buffer Block. If received Data is smaller than Sn_RX_RSR, it is stored in SOCKET n RX Buffer Block and Sn_RX_WR increases by stored Data Size.

Ex) S0_RX_WR = 1536(0x0600)

S0_RW_WR0(0x042A)	S0_RW_WR1(0x042B)
0x06	0x00

3.2.21 Sn_IMR (SOCKET n Interrupt Mask Register)

[R=W] [0x002C+0x0100*(n+4)] [0xFF]

Sn_IMR is used for the corresponding Sn_IR Bit Mask.

7	6	5	4	3	2	1	0
-	-	-	SENDOK	TIMEOUT	RECV	DISCON	CON
-	-	-	R=W	R=W	R=W	R=W	R=W

Bit	Symbol	Description
[7:5]	-	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**

3.2.22 Sn_FRAGR (SOCKET n Fragment Offset in IP Header Register)

[R=W] [0x002D+0x0100*(n+4), 0x002E+0x0100*(n+4)] [0x4000]

Sn_FRAGR configures Fragment Offset in IP Header.

Ex) S0_FRAG0 = 0x0000 (Don't Fragment)

S0_FRAGR0(0x042D)	S0_FRAGR1(0x042E)
0x00	0x00

3.2.23 Sn_MR2 (SOCKET n Mode register 2)

[R=W] [0x002F+0x0100*(n+4)] [0x00]

Sn_MR2 configures SOCKET n Option. Sn_MR2 must be set before Sn_CR[OPEN] = '1'.

7	6	5	4	3	2	1	0
-	MBBLK	MMBLK	IPV6BLK	-	-	BRDB	UNIB
-	R=W	R=W	R=W	-	-	R=W	R=W

Bit	Symbol	Description
7	-	Reserved
6	MBBLK	**Broadcast Blocking on MACRAW Mode** On MACRAW Mode, this Bit is set to '1' to block Broadcast Packet. 0 : Disable Broadcast Blocking 1 : Enable Broadcast Blocking
5	MMBLK	**Multicast Blocking on MACRAW Mode** On MACRAW Mode, this Bit is set to '1' to block Multicast Packet. 0 : Disable Multicast Blocking 1 : Enable Multicast Blocking
4	IPV6BLK	**IPv6 Packet Blocking on MACRAW Mode** On MACRAW Mode, this Bit is set '1' to block IPv6 Packet. 0 : Disable IPv6 Blocking 1 : Enable IPv6 Blocking
[3:2]	-	Reserved
1	BRDB	**Broadcast Blocking on UDP Mode/ Force PSH on TCP Mode** ***Broadcast Blocking on UDP Mode** On UDP Mode, this Bit is set '1' to block UDP Broadcast Packet. 0 : Disable Broadcast Blocking 1 : Enable Broadcast Blocking *** Force PSH on TCP Mode** On TCP Mode, this Bit is set to '1' to set PSH Flag in all transmitting Data Packet. 1: Force PSH Flag 0: No Force PSH Flag

0	UNIB	**Unicast Blocking on UDP Multicast Mode** On UDP Multicast Mode, this Bit is set to '1' to block UDP Unicast Packet. 0 : Disable Unicast Blocking 1 : Enable Unicast Blocking

3.2.24 Sn_KPALVTR (SOCKET n Keep Alive Timer Register)

[RO] [0x0030+0x0100*(n+4)] [0x00]

Sn_KPALVTR sets 'Keep Alive (KA)' Packet transmission time. The unit is 5 sec. TCP Mode SOCKET transmits KA Packet every Sn_KPALVTR. Before transmitting KA Packet, SOCKET must transmit Data (over 1 Byte) Packet at least once. By Sn_CR [SENDKEEP], KA Packet is transmitted without Sn_KPALVTR setting.

Ex) S0_KPALVTR = 10 (0x0A),

10 * 5s = 50s

S0_KPALVRT(0x0430)
0x0A

3.2.25 Sn_RTR (SOCKET n Retransmission Time Register)

[R=W] [0x0032+0x0100*(n+4), 0x0033+0x0100*(n+4)] [0x0000]

Sn_RTR sets SOCKET n Retransmission Time. If Sn_RTR is zero, SOCKET n Retransmission Time is set by RTR. (Ref *4.8 Retransmission*)

Ex) S0_RTR = 5000 (0x1388),

5000 * 100us = 0.5s

S0_RTR0(x0432)	S0_RTR1(0x0433)
0x013	0x88

3.2.26 Sn_RCR (SOCKET n Retransmission Count Register)

[R=W] [0x0034+0x0100*(n+4)] [0x00]

Sn_RCR sets SOCKET n Retransmission Counter. If Sn_RCR is zero, SOCKET n Retransmission Counter is set by RCR. (Ref *4.8 Retransmission*)

4 Functional Description

W5100S provides Internet Connectivity with simple register control. In this chapter, the Initialization of W5100S, Data Communication method according to each TCP, UDP, IPRAW, MACRAW Mode and the additional functions are described step by step based on the pseudo code.

4.1 W5100S RESET

- Set Hardware before Reset. (Ref *7.4.1 Reset Timing*)
- Set HOST Interface Mode with MOD [3:0] pin.
- Supply the Reset signal longer than 500ns on RSTn pin for Hardware Reset.
- Wait for T_{STA} (Ref *7.4.1 Reset Timing*)

4.2 Initialization

The Network Information and SOCKET n TX / RX buffer are set in the W5100S Initialization Process.

4.2.1 Basic Setting

To operate the W5100S properly, set the following Registers according to User's Application.

- Mode Register (MR)
- Interrupt Mask Register (IMR)
- Retransmission Time Register (RTR)
- Retransmission Count Register (RCR)

More Information for the above registers can be found in each Register Descriptions.

4.2.2 Network Information Setting

Set the basic Network Information for Internet Communication.

```
NETWORK SETTING:
{
    /* W5100S MAC Address, 11:22:33:AA:BB:CC */
    SHAR[0:5] = { 0x11, 0x22, 0x33, 0xAA, 0xBB, 0xCC };

    /* W5100S Gateway IP address, 192.168.0.1 */
    GAR[0:3] = { 0xC0, 0xA8, 0x00, 0x01 };
```

```
    /* W5100S Subnet MASK Address, 255.255.255.0 */
    SUBR[0:3] = { 0xFF, 0xFF,, 0xFF, 0x00};

    /* W5100S IP Address, 192.168.0.100 */
    SIPR[0:3] = {0xC0, 0xA8,0x00, 0x64};
}
```

4.2.3 SOCKET TX/RX Buffer Setting

Determine SOCKET n TX/RX Buffer Size using TMSR/RMSR or Sn_TXBUF_SIZE/Sn_RXBUF_SIZE. SOCKET n TX / RX Buffer is a RING-Buffer structure. So, calculation of the base address and MASK value for SOCKET n TX / RX Buffer control must be required.

Please make sure that the Sum of SOCKET n TX / RX Buffer Size must not be exceed 8 KB.

The pseudo-code for setting the SOCKET n TX/RX Buffer is described below.

```
In case of, assign 2KB TX/RX memory per SOCKET
{
  // set Base Address of TX/RX Memory for SOCKET n
  gS0_RX_BASE = 0x8000;    // TX Memory Block Base Address
  gS0_RX_BASE = 0xC000;    // RX Memory Block Base Address
  TxTotalSize = 0;          // For check the total size of SOCKET n TX Buffer
  RxTotalSize = 0;          // For check the total size of SOCKET n RX Buffer

  for (n=0; n<3; n++)  {
    Sn_TXBUF_SIZE = 2; // assign 2 Kbytes TX Memory per SOCKET
    Sn_RXBUF_SIZE = 2; // assign 2 Kbytes RX Memory per SOCKET
    // 0x07FF, for getting offset address within assigned SOCKET n TX/RX Memory
    gSn_TX_MASK = (1024 * Sn_TXBUF_SIZE) - 1;
    gSn_RX_MASK = (1024 * Sn_RXBUF_SIZE) - 1;
    if( n != 0)  {
      gSn_TX_BASE = gSn-1_TX_BASE + (1024 * Sn-1_TXBUF_SIZE);
      gSn_RX_BASE = gSn-1_RX_BASE + (1024 * Sn-1_RXBUF_SIZE);
    } // end if
    TxTotalSize = TxTotalSize + Sn_TXBUF_SIZE;
    RxTotalSize = RxTotalSize + Sn_RXBUF_SIZE;
    If( TxTotalSize > 8 or RxTotalSize > 8 )  goto ERROR;  // invalid Total Size
  } // end for
}
```

4.3 TCP

TCP (Transmission Control Protocol) is a bidirectional Data Transmission Protocol based on a 1:1 communication on Transport Layer. It also provides Communication between Applications by using Port Number.

TCP 1:1 communication needs the Connection Process such as transmitting Connection Request to Peer or receiving Connection Request from Peer. In this Connection Process, the side transmitting Connection Request is 'TCP CLIENT' and the other side receiving Connection Request is 'TCP SERVER'. TCP also provides reliable, ordered and error-checked delivery of a stream Data between applications running on hosts communicating by an IP network.

'TCP SERVER' and 'TCP CLIENT' are maintaining transmit and receive Data until the TCP connection is terminated.

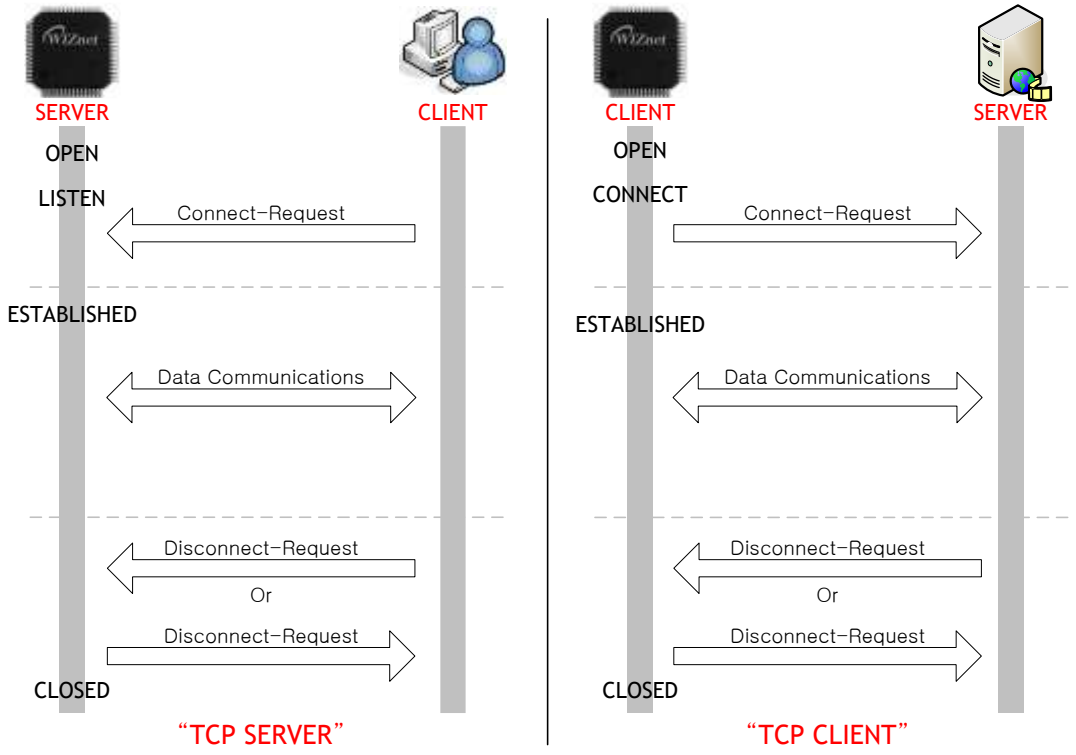


Figure 5 TCP SERVER and TCP CLIENT

4.3.1 TCP Server

Figure 6 show 'TCP SERVER' Operation Flow.

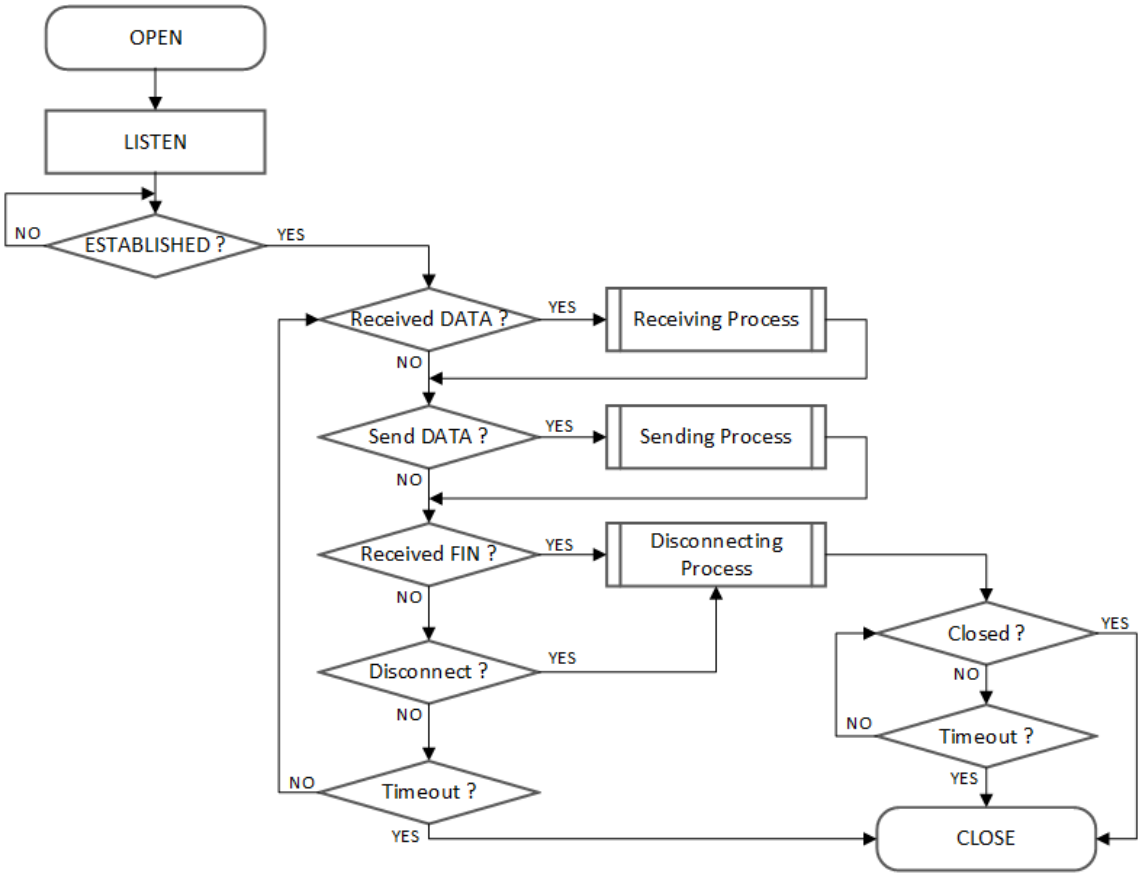


Figure 6 TCP Server Operation Flow

O OPEN

HOST configures SOCKET n to TCP Mode.

```
{
START :
   Sn_MR[3:0] = '0001';    /* set TCP Mode */
   Sn_PORTR[0:1] = {0x13,0x88}; /* set PORT Number, 5000(0x1388) */

   /* configure SOCKET Option when you need it. */
   // Sn_MR[ND] = '1'; /* set No Delay ACK */

   Sn_CR[OPEN] = '1';     /* set OPEN Command */
   while(Sn_CR != 0x00); /* wait until OPEN Command is cleared*/

   if(Sn_SR != SOCK_INIT) goto START; /* check SOCKET Status */
```

```
}
```

○ LISTEN

SOCKET n is operated as 'TCP SERVER' by Sn_CR [LISTEN]. 'TCP SERVER' waits for SYN Packet in Sn_SR (SOCK_LISTEN)

```
{
  Sn_CR = LISTEN; /* set LISTEN Command */
  while(Sn_CR != 0x00); /* wait until LISTEN Command is cleared*/
  if(Sn_SR != SOCK_LISTEN) goto OPEN; /* check SOCKET Status */
}
```

○ ESTABLISHED?

'TCP SERVER' keeps Sn_SR (SOCK_LISTEN) until received SYN Packet. If 'TCP SERVER' receives SYN Packet from 'TCP CLIENT', it transmits SYN/ACK Packet to 'TCP CLIENT' and the Connection Process between 'TCP SERVER' and 'TCP CLIENT' is completed.

If there is no response from Peer against of transmitted SYN Packet or SYN/ACK Packet within the Retransmission Time, Sn_IR [TIMEOUT] is set to '1'.

```
First method :
{
    /* check SOCKET Interrupt */
    if (Sn_IR[CON]   == '1')
    {
        /* clear SOCKET Interrupt */
        Sn_IR[CON] = '1';
        goto Received DATA?;   /* or goto Send DATA?; */
    }
    else if(Sn_IR[TIMEOUT] == '1')   goto Timeout?;
}
```

```
Second method :
{
    if (Sn_SR == SOCK_ESTABLISHED)
    {
        /* clear SOCKET Interrupt */
        Sn_IR[CON] = '1';
        goto Received DATA?     /* or goto Send DATA?; */
    }
```

```
    else if(Sn_IR[TIMEOUT] == '1') goto Timeout?;
}
```

☐ **Receive DATA?**

Whether SOCKET n Data is received is confirmed by Sn_IR [RECV] or Sn_RX_RSR.

```
First method :
{
  /* check SOCKET RX Memory Received Size */
  if (Sn_RX_RSR > 0)   goto Receiving Process;
}
```

```
Second method :
{
  if (Sn_IR[RECV] == '1')
  {
    /* check SOCKET Interrupt */
    Sn_IR[RECV] = '1';      /* clear SOCKET Interrupt */
    goto Receiving Process;
  }
}
```

○ Receiving Process

Received Data is read from SOCKET n RX Buffer Block.

The Read Offset Address of Received Data in RX Memory Block is calculated by gSn_RX_BASE, gSn_RX_MASK and Sn_RX_RD. (Ref *4.2.3 SOCKET TX/RX Buffer Setting*)

After reading received Data, Sn_RX_RD must be increased by Data read Size and Sn_CR [RECV] must be set to '1'. If there is remain Data in SOCKET n RX Buffer Block after Sn_CR [RECV] Command, Sn_IR [RECV] is set to '1'.

When Read Offset Address calculated, it is cautious to over the boundary Address (n=0,1,2 : gSn_RX_BASE ~ gSn+1_RX_BASE, n=3 : gS3_RX_BASE ~ 0xFFFF) of SOCKET n RX Buffer Block.

```
{
    /* get Received Size */
    get_size = Sn_RX_RSR;

    /* calculate SOCKET n RX Buffer Size & Offset Address */
    gSn_RX_MAX = Sn_RXBUF_SIZE * 1024;
    get_offset = Sn_RX_RD & gSn_RX_MASK;
```

```
    /* calculate Read Offset Address */
    get_start_address = gSn_RX_BASE + get_offset;

    /* if overflow the upper boundary of SOCKET n RX Buffer */
    If( (get_offset + get_size) > gSn_RX_MAX )
    {
      /* copy upper_size bytes of get_start_address to destination_address
        - destination_address is user Data Memory Address */
      upper_size = gSn_RX_MAX - get_offset;
      memcpy(get_start_address, destination_address, upper_size);
      destination_address += upper_size;
      /* copy the remained size bytes of gSn_RX_BASE to destination_address */
      remained_size = get_size - upper_size;
      memcpy(gSn_RX_BASE, destination_address, remained_size);
    }
    else
    {
      /* copy get_size of get_start_address to destination_address */
      memcpy(get_start_address, destination_address, get_size);
    }

    /* increase Sn_RX_RD as get_size */
    Sn_RX_RD += get_size;

    /* set RECV Command */
    Sn_CR[RECV] = '1';
    while(Sn_CR != 0x00); /* wait until RECV Command is cleared*/
}
```

☐ **Send DATA? / Sending Process**

Written Data in SOCKET n TX Buffer Block is transmitted.

The Write Offset Address in TX Memory Block is calculated by gSn_TX_BASE, gSn_TX_MASK and Sn_TX_WD. (Ref *4.2.3 SOCKET TX/RX Buffer Setting*). And Data to be transmitted form the Write Offset Address is written. After writing Data, Sn_TX_WD must be increased by written Data Size and the Data is transmitted by Sn_CR [SEND].

Before Sn_IR [SENDOK] = '1', next Data Transmission Process is not executed. How long time until Sn_IR [SENDOK] = '1' after transmitting Data is depending on SOCKET Count, Data Size and Network Traffic. Also Sn_IR [TIMEOUT] could be occurred. (Ref *4.8.2 TCP Retransmission*)

When Write Offset Address calculated, it is cautious to over the boundary Address (n=0,1,2 : gSn_TX_BASE ~ gSn+1_TX_BASE, n=3 : gS3_TX_BASE ~ 0xC000) of SOCKET n TX Buffer Block. If there is no response from Peer against of transmitted Data Packet within the Retransmission Time, Sn_IR [TIMEOUT] is set to '1'.

```
{
  /* calculate SOCKETn TX Buffer Size & Offset Address */
  gSn_TX_MAX = Sn_TXBUF_SIZE * 1024;
  get_offset = Sn_TX_WR & gSn_TX_MASK;

  /* check the max size of DATA(send_size) & Free Size of SOCKETn TX
      Buffer(Sn_TX_FSR)*/
  if( send_size >gSn_TX_MAX ) send_size = gSn_TX_MAX;
  while(send <= Sn_TX_FSR);    // Wait until SOCKET n TX Buffer is free */
  /* If you don't want to wait TX Buffer Free
     send_size = Sn_TX_FSR; // write Data as size of Free Buffer
  */

  /* calculate Write Offset Address */
  get_start_address = gSn_TX_BASE + get_offset;

  /* if overflow the upper boundary of SOCKET n TX Buffer */
  If( (get_offset + send_size) > gSn_TX_MAX )
  {
     /* copy upper size bytes of source_address to get_start_address
        - source_address is the start address of user data */
     upper_size = gSn_TX_MAX - get_offset;
     memcpy(source_address, get_start_address, upper_size);

     /* copy the Remained Size Bytes of source_address to gSn_TX_BASE */
     source_address += upper_size;
     remained_size = send_size - upper_size;
     memcpy(source_address, gSn_TX_BASE, remained_size);
  }
  else
  {
     /* copy send_size bytes of source_address to get_start_address
      - source_address is the start address of user data */
```

```
    memcpy(source_address, get_start_address, send_size);
  }

  /* increase Sn_TX_WR as send_size */
  Sn_TX_WR += send_size;

  /* set SEND Command */
  Sn_CR = SEND;
  while(Sn_CR != 0x00); /* wait until SEND Command is cleared*/

  /* wait until SEND Command is completed or TIMEOUT Interrupt is occurred*/
  while(Sn_IR[SENDOK] == '0' and Sn_IR[TIMEOUT] = '0');

  /* clear SOCKET Interrupt*/
  if(Sn_IR[SENDOK] == '1') Sn_IR[SENDOK] = '1';
  else goto Timeout?;
}
```

☐ **Received FIN (Passive Close)**

When FIN Packet received from Peer.

```
First Method:
{
    If(Sn_SR == SOCK_CLOSE_WAIT) goto Disconnecting Process;
}
```

```
Second Method:
{
    If(Sn_IR[DISCON] == '1') goto Disconnecting Process;
}
```

☐ **Disconnected (Active Close)**

When FIN Packet transmitted to Peer.

```
{
    /* send FIN Packet */
    Sn_CR[DISCON] = '1';
    while(Sn_CR != 0x00); /* wait until DISCON Command is cleared*/
    goto Disconnecting Process;
}
```

☐ **Disconnecting Process**

In Passive Close, if FIN Packet is received from Peer and there is no Data to be transmitted, SOCKET transmits FIN Packet and it will be closed. If there is no response from Peer against of transmitted FIN Packet within the Retransmission Time, Sn_IR [TIMEOUT] is set to '1'.

In Active Close, if SOCKET transmits FIN Packet to Peer, SOCKET waits for Peer FIN Packet. SOCKET will be closed after receiving FIN Packet from Peer. If there is no response from Peer against of transmitted FIN Packet within the Retransmission Time, Sn_IR [TIMEOUT] is set to '1'.

```
Passive Close: /* received FIN Packet from Peer */
{
   /* send FIN Packet */
   Sn_CR = DISCON;
   while(Sn_CR != 0x00); /* wait until DISCON Command is cleared*/

   /* wait unit ACK Packet is received*/
   while(Sn_IR[DISCON] == '0' and Sn_IR[TIMEOUT] == '0') ;
   if (Sn_IR[DISCON] == '1')
   {
      /* clear Interrupt */
      Sn_IR[DISCON] = '1';
      goto CLOSED;
   }
   else   goto Timeout?;
}
```

```
Active Close : /* sent FIN Packet to Peer */
{
   /* wait until FIN Packet is received*/
   while(Sn_IR[DISCON] == '0' and Sn_IR[TIMEOUT] == '0') ;
   if (Sn_IR[DISOCN] == '1')
   {
      /* clear Interrupt */
      Sn_IR[DISCON] = '1';
      goto CLOSED;
   }
   else   goto Timeout?;
}
```

☐ **Timeout?**

If there is no response from Peer against of transmitted SYN or SYN/ACK or FIN or Data Packet within the Retransmission Time, Sn_IR [TIMEOUT] is set to '1'. (Ref *4.8.2 TCP Retransmission*)

```
{
  /* check TIMEOUT Interrupt */
  if(Sn_IR[TIMEOUT] == '1')
  {
    /* clear Interrupt */
    Sn_IR[TIMEOUT] = '1';
    goto CLOSE;
  }
}
```

☐ **CLOSE**

SOCKET n is closed by the Disconnect Process, Sn_IR [TIMEOUT] = '1' and Sn_CR [CLOSE] = '1'.

```
{
  /*wait until SOCKET n is closed*/
  while(Sn_SR != SOCK_CLOSED);
}
```

4.3.2 TCP Client

Figure 7 shows 'TCP CLIENT' Operation Flow.

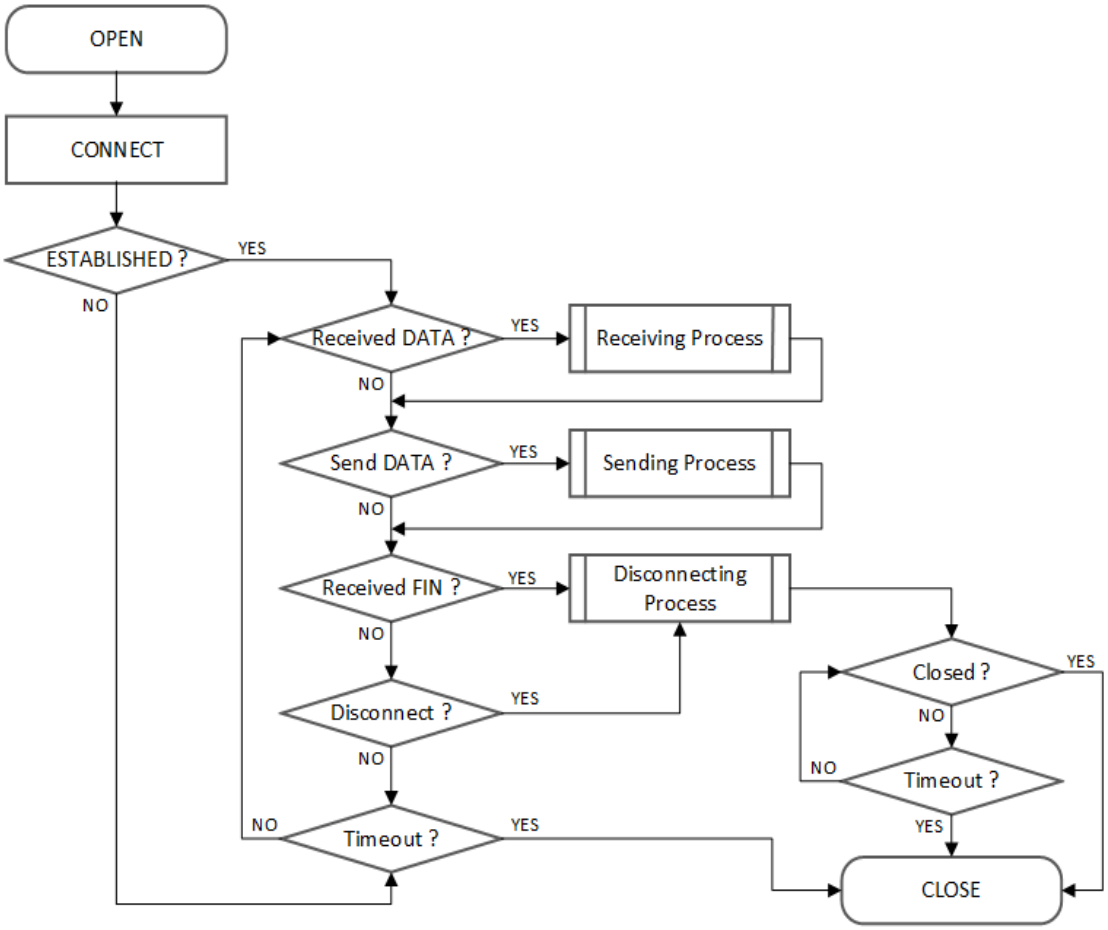


Figure 7 TCP Client Operation Flow

○ OPEN

Refer to *4.3.1 TCP Server : OPEN*

○ CONNECT

SOCKET n is operated as 'TCP CLIENT' by Sn_CR [CONNECT].

SYN Packet is transmitted to 'TCP SERVER' by Sn_CR [CONNECT].

```
{
    /* set Destination IP Address, 192.168.0.11 */
    Sn_DIPR[0:3] ={ 0xC0, 0xA8, 0x00, 0x0B};

    /* set Destination PORT Number, 5000(0x1388) */
    Sn_DPORTR[0:1] = {0x13, 0x88};
```

```
    /* set CONNECT Command */
    Sn_CR = CONNECT;
    while(Sn_CR != 0x00); /* wait until CONNECT Command is cleared*/
    goto ESTABLISHED?;
}
```

○ ESTABLISHED?

'TCP CLIENT' is in Sn_SR (SOCK_SYNSENT) until receiving SYN/ACK Packet from 'TCP SERVER' against of SYN Packet transmitted. If SYN/ACK Packet is received from 'TCP SERVER', the Connection Process between 'TCP SERVER' and 'TCP CLIENT' is completed.

If there is no response from Peer against of transmitted SYN Packet within the Retransmission Time, Sn_IR [TIMEOUT] is set to '1'.
(Ref *4.3.1 TCP Server : Received DATA?*)

○ Others flow

Refer to *4.3.1 TCP Server*.

4.3.3 Other Functions

4.3.3.1 TCP SOCKET Options

Before Sn_CR [OPEN] is set to '1', SOCKET Option can be set by Sn_MR and Sn_MR2.

○ **No Delayed ACK : Sn_MR [NDACK] = '1'**

If No Delayed ACK option is set, SOCKET sends ACK Packet against of Peer Data Packet without Delay.

○ **Delayed ACK : Sn_MR [NDACK] = '0'**

If No Delayed ACK option is not set, SOCKET sends ACK Packet against of Peer Data Packet after RTR or TCP Window Size increased.

○ **Force PSH : Sn_MR2 [BRDB]='1'**

If Force PSH option is set, SOCKET puts PSH flag in every Data Packet to be transmitted.

○ **Auto PSH : Sn_MR2 [BRDB]='0'**

If Force PSH option is not set, SOCKET puts PSH flag in the last Data Packet to be transmitted.

4.3.3.2 Keep Alive

Keep Alive (KA) is a message sent by Peer to check that the link is operating, or to prevent the link from being broken. Keep Alive sends the last 1 Byte from the last transmitted Data. So, Data bigger than 1 Byte must be transmitted before Keep Alive.
If there is no response from Peer against of KA Packet within the Retransmission Time, Sn_IR [TIMEOUT] is set to '1'.

The period of KA Packet transmit is set by Sn_KPALVTR. If Sn_KPALVTR is zero, KA Packet is able to be transmitted by Sn_CR [SEND_KEEP].

4.4 UDP

UDP (User Datagram Protocol) is a Datagram Communication Protocol and doesn't guarantee the stability in Transport Layer above the IP Layer. It also provides Communication between Applications using Port Numbers. UDP can communicate with more than one Peer and doesn't require the Connection Process. On the other hand, UDP has Data Loss and receives Data from any Peers because UDP has no guarantee reliability. UDP Transmission Methods are Unicast, Broadcast and Multicast according to Data transmit/receive range.

Figure 8 shows UDP Operation Flow.

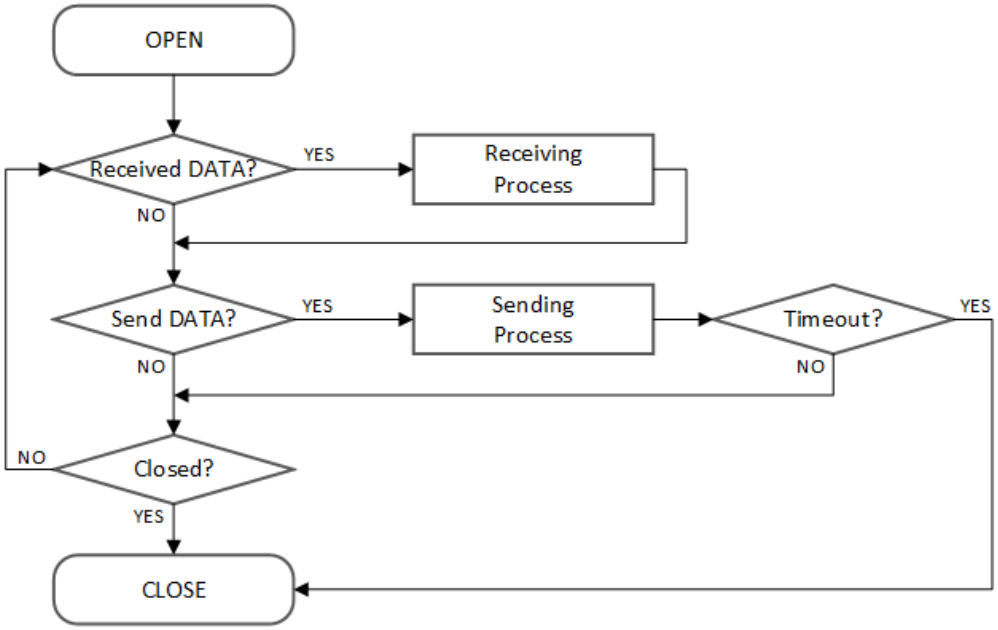


Figure 8 UDP Operation Flow

4.4.1 UDP Unicast

UDP Unicast is the first Communication Method in which sender is one and receiver is one. Before Data Transmit, SOCKET performs the ARP Process. In the ARP Process, if there is no response from Peer against of ARP Request within the Retransmission Time, Sn_IR [TIMEOUT] is set to '1'. (Ref *4.8.1 ARP & PING Retransmission*)

If previous Destination and current Destination are the same, the ARP Process is skipped. Also, the ARP Process is skipped by Sn_CR [SEND_MAD] and Sn_DHAR.

○ OPEN

SOCKET n is configured as UDP Mode by OPEN Command.

```
{
```

```
START :
   /* set UDP Mode */
   Sn_MR[3:0] = ‘0010’;

   /* set Source PORT Number, 5000(0x1388) */
   Sn_PORTR[0:1] = {0x13, 0x88};

   /* set SOCKET Option such as Broadcast Block. */
   // refer to 3.2.23 Sn_MR2 (SOCKET n Mode register 2)
   // Sn_MR2[BRDB] = ‘1’;

   /* set OPEN Command */
   Sn_CR = OPEN;
   while(Sn_CR != 0x00); /* wait until OPEN Command is cleared*/

   /* check SOCKET for UDP Mode */
   if(Sn_SR != SOCK_UDP) goto START;
}
```

○ Received DATA?

Refer to *4.3.1 TCP Server : Received DATA?*

○ Receiving Process

UDP Mode SOCKET can receive Data Packets from more than one Peer. The received Data Packet is stored in SOCKET n RX Buffer Block with “PACKET INFO” as shown in Figure 9. If the received Data is bigger than SOCKET n RX Buffer Free Size, it is discarded.

(Ref *4.3.1 TCP Server : Receiving Process*)

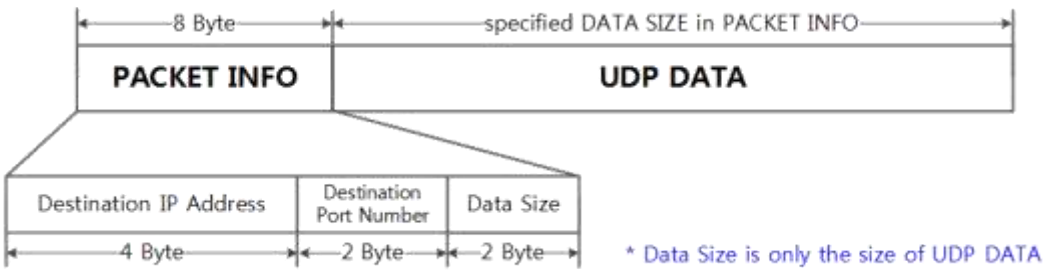


Figure 9 Received UDP DATA in SOCKETn RX Buffer Block

```
{
   /* receive PACKINFO */
```

```
    goto   4.3.1 TCP Server : Receiving Process with get_size = 8;

    /* extract Destination IP, Port, Size in PACKET INFO*/
    dest_ip[0:3] = destination_address[0:3];
    dest_port = (destination_address[4] << 8) + destination_address[5];
    data_size = (destination_address[6] << 8) + destination_address[7];

    /* read UDP Data */
    goto 4.3.1 TCP Server : Receiving Process with get_size = data_size;
}
```

☐ **Send DATA? / Sending Process**

Refer to *4.3.1 TCP Server : Send DATA? / Sending Process.*

```
{
      /* set Destination IP Address, 192.168.0.11 */
      Sn_DIPR[0:3] = {0xC0, 0xA8, 0x00, 0x0B};

      /* set Destination PORT Number, 5000(0x1388) */
      Sn_PORTR[0:1] = {0x13, 0x88};

      /* for using SEND_MAC Command   : */
      // refer to 4.4.4.1 UDP MAC Send
      /* set Destination MAC Address, 11:22:33:AA:BB:CC
          Sn_DHAR[0:5] = {0x11, 0x22, 0x33, 0xAA, 0xBB, 0xCC};
      */
      goto 4.3.1 TCP Server : Sending Process;

      /* for using SEND_MAC Command   : */
      // refer to 4.4.4.1 UDP MAC Send
      /*
        goto 4.3.1 TCP Server : Sending Process replaced Sn_CR[SEND] with
        Sn_CR[SEND_MAC];
      */
}
```

☐ **Timeout?**

If Data packet is transmitted to Peer at the first time or the Peer to be received Data Packet is different from the last Peer, the ARP process is performed before transmitting Data packet.

In the ARP Process, if there is no response from Peer against of ARP Request within the Retransmission Time, Sn_IR [TIMEOUT] is set to '1'.

UDP Mode SOCKET does not closed with Sn_IR [TIMEOUT] because it supports 1:N Communication. (Ref *4.8.1 ARP & PING Retransmission*)

```
{
  /* check SOCKET Status */
  if(Sn_IR[TIMEOUT] == '1')
  {
    /* clear Interrupt */
     Sn_IR[SENDOK] = '1';
     goto Received DATA?     /* or goto Send DATA? or goto CLOSE */
  }
}
```

☐ **CLOSE**

Closed by Sn_CR [CLOSED].

```
{
    /* set CLOSE Command */
    Sn_CR = CLOSE;
    while(Sn_CR != 0x00); /* wait until CLOSE Command is cleared*/

    /* wait until SOCKET n is closed */
    while(Sn_SR == SOCK_CLOSED);
}
```

4.4.2 UDP Broadcast

UDP Broadcast is the second Communication Method in which a sender transmits data to all of the same band. There are two types of Broadcasting. All Node Broadcasting for all Node in Network and Subnet Broadcasting for the Nodes having the same Subnet in Network.

☐ **Send DATA? / Sending Process**

In UDP Mode, when data is transmitted using Broadcast, Sn_DIPR must be set to Broadcast Address of the same band before Sn_CR [SEND] Command.

```
All Node Broadcasting :
{
```

```
   /* set Broadcast Address, 255.255.255.255 */
   Sn_DIPR[0:3] = {0xFF, 0xFF, 0xFF, 0xFF};

   /* set Destination PORT Number, 5000(0x1388) */
   Sn_PORTR[0:1] = {0x13,0x88};

   goto 4.3.1 TCP Server : Sending Process;
}
```

```
Subnet Broadcasting : Assume SIPR = “192.168.0.10” & SUBR = “255.255.255.0”
{
   /* set Broadcast Address, 192.168.0.255 */
   Sn_DIPR[0:3] = {0xC0, 0xA8, 0x00, 0xFF};

   /* set Destination PORT Number, 5000(0x1388) */
   Sn_PORTR[0:1] = {0x13,0x88};

   goto 4.3.1TCP Server : Sending Process;
}
```

4.4.3 UDP Multicast

UDP Multicast is the last Communication Method in which sender is one and receiver is a group. The Multicast-Group Address Range is 224.0.0.0 ~ 239.255.255.255 (Ref *IANA_Multicast Address*) and the corresponding MAC Address is 01:00:5E:00:00:00 ~ 01:00:5E:FF:FF:FF . The Lower 23 bits of MAC Address has the same Address with the Multicast-Group Address. (Ref *rfc1112*)

☐ **OPEN**

Before Sn_CR [OPEN] Command, Multicast-Group Information and Sn_MR [MULTI] must be set. IGMP (Internet Group Management Protocol) JOIN message is transmitted by Sn_CR [OPEN] Command.

IGMP version is set as version 1 or version 2 by Sn_MR [MS].

```
{
START :
      /* set Multicast-Group MAC Address, 01:00:5E:00:00:64 */
      Sn_DHAR[0:5] = {0x01, 0x00, 0x5E, 0x00, 0x00, 0x64};

      /* set Multicast-Group IP Address, 224.0.0.100 */
```

```
    Sn_DIPR[0:3] = {0xE0, 0x00, 0x00, 0x64};

    /* set Multicast-Group PORT Number, 3000(0x0BB8) */
    Sn_DPORTR[0:1] = {0x0B, 0xB8};

    /* set UDP Multicast */
    Sn_MR[MULTI] = '1';

    /* set IGMP Version */
    Sn_MR[MC] = '1';    /* Sn_MR[MC] = '1' : IGMPv1 , Sn_MR[MC] = '0' : IGMPv2 */

    /* set SOCKET Option such as Unicast Block and Broadcast Block.
     refer to 3.2.23 Sn_MR2 (SOCKET n Mode register 2) */
    // Sn_MR2[UNIB] = '1';
    // Sn_MR2[BRDB] = '1';

    /* set UDP Mode */
    Sn_MR[3:0] = 4'b0010;

    /* set Source PORT Number, 3000(0x0BB8) */
    Sn_PORTR[0:1] = {0x0B, 0xB8};

    /* set OPEN Command */
    Sn_CR = OPEN;

    /* check SOCKET for UDP Mode */
    if(Sn_SR != SOCK_UDP) goto START;
}
```

○ Send DATA? / Sending Process

Refer to *4.3.1TCP Server : Sending Process*

4.4.4 Other Functions

4.4.4.1 UDP MAC Send

When Peer MAC Address is known, UDP Mode SOCKET transmits Data Packet without the ARP Process by setting Sn_DHAR.

(Ref *4.4.1 UDP Unicast : Send DATA? / Sending Process*)

4.4.4.2 UDP SOCKET Options

In UDP Mode, Unicast and Broadcast Packets are received. But if Sn_MR2 [BRDB] is set to '1', Broadcast Packet is blocked.

In UDP Multicast Mode, Unicast, Broadcast and Multicast Packets are received. But if Sn_MR2 [UNIB] or Sn_MR2 [BRDB] are set to '1', Unicast or Broadcast Packets are blocked.

These Block Bits must be set before Sn_CR [OPEN] Command.

Sn_MR[MULTI]	Sn_MR2[BRDB]	Sn_MR2[UNIB]	Unicast	Multicast	Broadcast
0	0	Don't Care	O	X	O
0	1	Don't Care	O	X	X
1	0	0	O	O	O
1	0	1	X	O	O
1	1	0	O	O	X
1	1	1	X	O	X

4.4.4.3 Port Unreachable Block

If Peer transmits UDP Packet to a Port that does not exist on W5100S. W5100S automatically transmits ICMP Packet (Destination Port Unreachable) to Peer. But it could be the target for Port Scan Attack. But if MR2 [UDPURB] is set to '1', W5100S does not transmit ICMP Packet (Destination Port Unreachable).

4.5 IPRAW

IPRAW Mode SOCKET supports Internet Protocol (IPv4) Layer Communication and Internet Protocol is set by Sn_PROTOR. (Ref *IANA_Protocol Number*)

IPRAW Mode SOCKET does not support IPv6, TCP and UDP.

Table 5 Internet Protocol Supported In IPRAW Mode

Protocol	Number	Semantic	W5100S Support
ICMP	1	Internet Control Message Protocol	O
IGMP	2	Internet Group Management	O
IPv4	4	IPv4 encapsulation	O
TCP	6	Transmission Control	X
UDP	17	User Datagram	X
IPv6	-	Protocols over IPv6	X
others	-	Other Protocols	O

If IPRAW Mode SOCKET is opened and Sn_PROTOR is set to ICMP, W5100S does not send PING Reply against of Peer PING Request and IPRAW Mode SOCKET stores Peer PING Request Packet in SOCKET RX Buffer Block.

Figure 10 shows IPRAW Mode SOCKET Operation Flow.

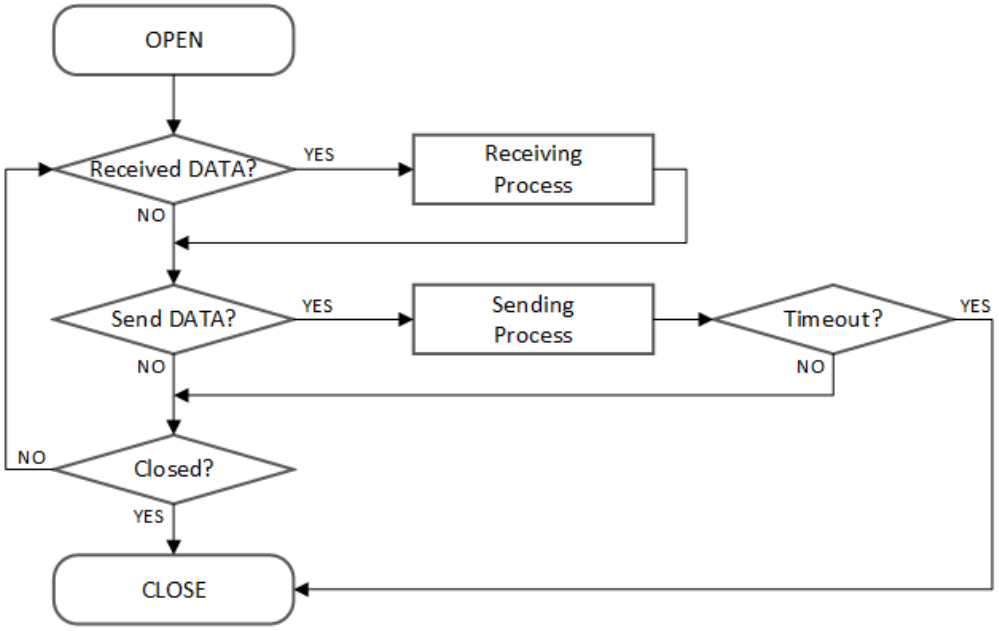


Figure 10 IPRAW Operation Flow

○ OPEN

IPRAW Mode SOCKET is opened.

```
{
START :
   /* set Protocol Number */
   Sn_PROTOR = protocol_num;

   /* set IPRAW Mode */
   Sn_MR[3:0] = “0011”;

   /* set OPEN Command */
   Sn_CR[OPEN] = ‘1’;
   while(Sn_CR != 0x00); /* wait until OPEN Command is cleared*/

   /* check SOCKET for IPRAW Mode */
   if(Sn_SR != SOCK_IPRAW) goto START;
}
```

☐ **Received DATA?**

Refer to *4.3.1 TCP Server : Received DATA?*

☐ **Receiving Process**

IPRAW Mode SOCKET receives Data Packet using the same Internet Protocol from any Peer. IPRAW Mode SOCKET stores Data including “PACKET INFO” in SOCKET RX Buffer Block as the below Figure 11. If the received Data is bigger than SOCKET n RX Buffer Free Size, it is discarded.

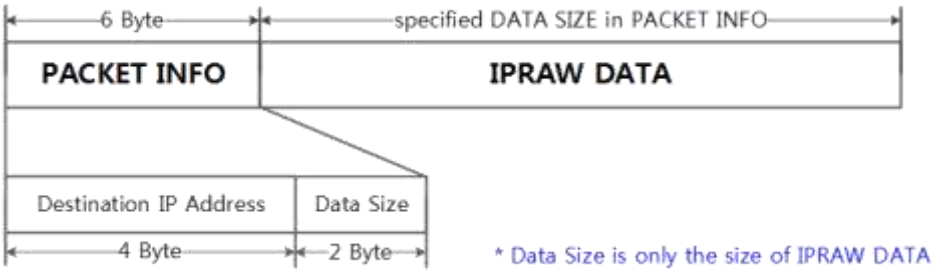


Figure 11 Received Data in IPRAW Mode SOCKET RX Buffer Block

```
{
   /* receive PACKINFO */
   goto 4.3.1TCP Server : Receiving Process with get_size = 6;

   /* extract Destination IP, Size in PACKET INFO*/
   dest_ip[0:3] = destination_address[0:3];
   data_size = (destination_address[4] << 8) + destination_address[5];
```

```
   /* read UDP Data */
   goto 4.3.1 TCP Server : Receiving Process with get_size = data_size;
}
```

○ Sending DATA? / Sending Process

Data to be transmitted must not exceeded SOCKET n TX Buffer Free Size. If the Data is bigger than IPRAW MSS, HOST must separate Data smaller than MSS.

```
{
     /* set Destination IP Address, 192.168.0.11 */
     Sn_DIPR[0:3] = {0xC0, 0xA8, 0x00, 0x0B};

     /* for using SEND_MAC Command   :
      refer to 4.4.4.1 UDP MAC Send */
     /* set Destination MAC Address, 11:22:33:AA:BB:CC
        Sn_DHAR[0:5] = {0x11, 0x22, 0x33, 0xAA, 0xBB, 0xCC};
     */
     goto 4.3.1 TCP Server : Sending Process;

     /* for using SEND_MAC Command   :
      refer to 4.4.4.1 UDP MAC Send */
     /*
        goto 4.3.1 TCP Server : Sending Process replaced Sn_CR[SEND] with
        Sn_CR[SEND_MAC];
     */
}
```

□ Timeout?

Refer to *4.4.1 UDP Unicast : Timeout?*

4.6 MACRAW

MACRAW Mode supports Data Communication in Ethernet MAC and it is only using SOCKET 0.

MACRAW Mode SOCKET 0 receives all Ethernet Packet or receiving Ethernet Packet (as Broadcast, Multicast and a Data Packet having the same Destination MAC Address with SHAR) is restricted by Sn_MR [MR]. And also Sn_MR2 can block receiving Broadcast, Multicast and IPv6 Packet.

MACRAW Mode SOCKET 0 does not receive any Data Packet for other SOCKET but it receives ARP Request and PING Request (If IPRAW Mode SOCKET does not support ICMP). Even though MACRAW Mode SOCKET 0 receives ARP and PING Request, W5100S transmits automatically ARP and PING Reply.

Figure 12 shows MACRAW Mode SOCKET 0 Operation Flow.

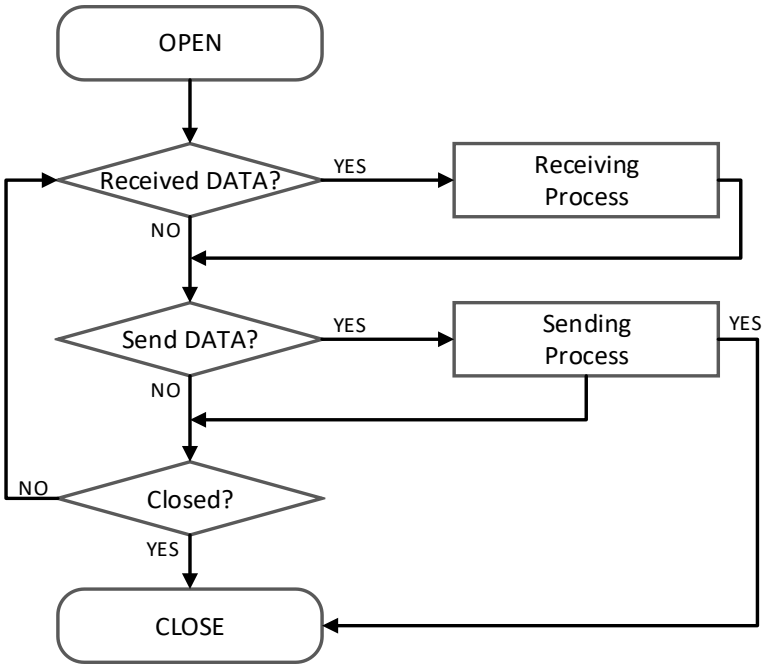


Figure 12 MACRAW Operation Flow

○ OPEN

MACRAW Mode SOCKET 0 is opened.

```
{
START :
   /* set MACRAW Mode */
   S0_MR = “0100”;

   /* MACRAW SOCKET Options */
   /*
```

```
    S0_MR[MR] = '1';          // Enable MAC Filter
    S0_MR2[MBBLK] = '1';      // Broadcast Packet Block
    S0_MR2[MMBLK] = '1';      // Multicast Packet Block
    S0_MR2[IPV6BLK] = '1';    // IPv6 Packet Block
  */

  /* set OPEN Command */
  S0_CR = OPEN;
  while(Sn_CR != 0x00); /* wait until OPEN Command is cleared*/

  /* check SOCKET0 is MACRAW Mode */
  if(S0_SR != SOCK_MACRAW) S0_CR = CLOSE; goto START;
}
```

☐ **Received DATA?**

Refer to *4.3.1 TCP Server : Received DATA?*

☐ **Receiving Process**

MACRAW Mode SOCKET 0 receives Data Packet from more than one Peers. MACRAW Mode SOCKET 0 stores Data including "PACKET INFO" in the SOCKET 0 RX Buffer Block as shown in Figure 13.

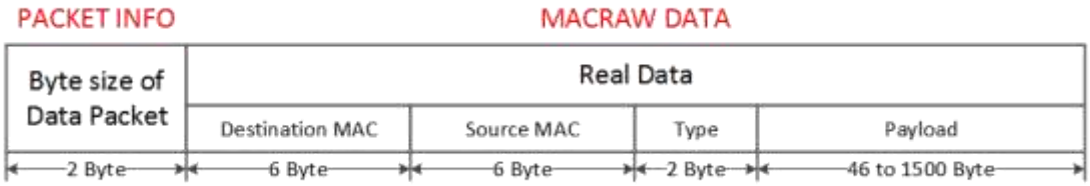


Figure 13 Received DATA Format in MACRAW

```
{
  /* receive PACKINFO */
  goto 4.3.1 TCP Server : Receiving Process with get_size = 2;

  /* extract Size in PACKET INFO*/
  data_size = (destination_address[0] << 8) + destination_address[1];

  /* read UDP Data */
  goto 4.3.1 TCP Server : Receiving Process with get_size = data_size;
}
```

○ Sending DATA? / Sending Process

Data to be transmitted must not be exceeded SOCKET 0 TX Buffer size. If Data is bigger than MSS, it must be divided by MSS and transmitted. If transmitting Data is smaller than 60byte, the Data is Zero Padding to 60 Bytes. (Ref *4.3.1 TCP Server : Send DATA? /Sending Process)*

☐ CLOSE

Refer to *4.4.1 UDP Unicast : CLOSE*

4.7 SOCKET-less Command (SLCR)

SOCKET-less Command (SLCR) transmits ARP Request and PING Request by SLCR [ARP] = '1' and SLCR [PING] = '1' without SOCKET OPEN.

If ARP and PING Reply are received from Peer, SLIR [ARP] and SLIR [PING] are set to '1'.

But if there is no response against of each Requests within SOCKET-less Retransmission Time, TIMEOUT is occurred (SLIR [TIMEOUT] = '1'). (Ref *4.8.1 ARP & PING Retransmission*)

Figure 14 shows SOCKET-less Command Operation Flow.

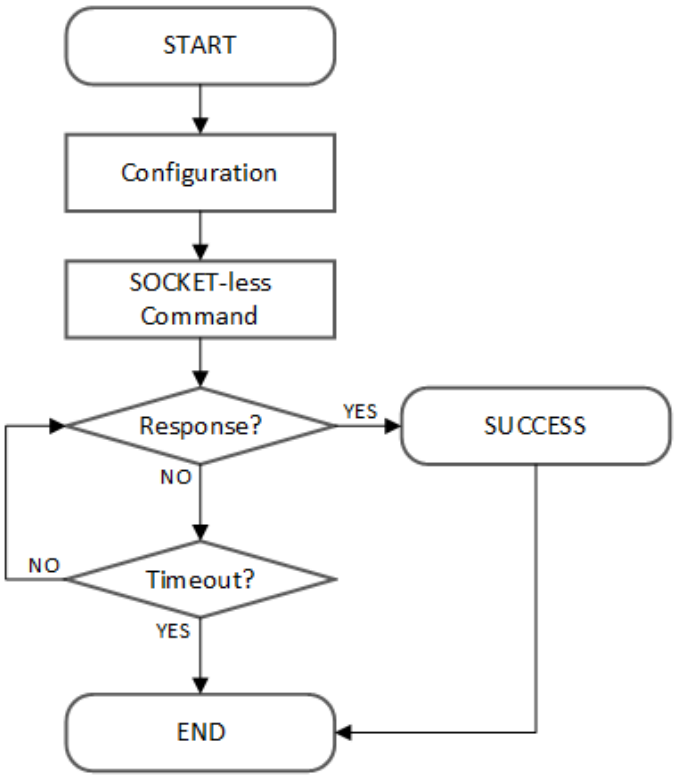


Figure 14 SOCKET-less Command Operation Flow

4.7.1 ARP Request (SLCR [ARP] = '1')

ARP Request Packet is transmitted by SLCR [ARP] Command. Before SLCR [ARP] Command, Peer IP Address must be set in SLPIPR. If ARP Reply is received from Peer, SLIR [ARP] is set to '1' and Peer MAC Address is written in SLPHAR. If there is no response against of ARP Request within SOCKET-less Retransmission Time, SLIR [TIMEOUT] is set to '1'. (Ref *4.8.1 ARP & PING Retransmission*)

☐ **Configuration**

Configure SOCKET-less Retransmission Time and ARP Interrupt Mask Bit and Destination Peer IP Address.

```
{
START :
   /* set SOCKET-less Retransmission Time, 100ms(0x03E8) (The unit is 100us) */
   SLRTR[0:1] = {0x03, 0xE8};

   /* set SOCKET-less Retransmission Counter, 5 */
   SLRCR = 0x05;

   /* set Interrupt Mask Bit */
   SLIMR[ARP] = '1';          // ARP Interrupt Mask Bit
   SLIMR[TIMEOUT] = '1';    // TIMEOUT Interrupt Mask Bit

   /* set Destination IP Address, 192.168.0.100 */
   SLPIPR[0:3] = {0xC0, 0xA8, 0x00, 0x64};
}
```

☐ **SOCKET-less Command**

ARP Request Packet is transmitted by ARP Command.

```
{
   /* set ARP Command */
   SLCR[ARP] = '1';
   while(SLCR != 0x00) ; /* wait until ARP Command is completed*/
}
```

☐ **Response?**

ARP Interrupt indicates whether ARP Reply has been received.

```
{
   /* check ARP Interrupt */
   if(SLIR[ARP] == '1')    /* received ARP Reply Packet */
   {
      /* clear Interrupt */
      SLIR[ARP] = '1';    goto SUCCESS;
   }
   else goto Timeout;
}
```

☐ **Timeout?**

If there is no response from Peer against of transmitted ARP Request Packet within the Retransmission Time, TIMEOUT Interrupt is occurred. (Ref *4.8.1 ARP & PING Retransmission*)

```
{
  /* check TIMEOUT Interrupt */
  if(SLIR[TIMEOUT]   == 1)
  {
    /* clear Interrupt */
    SLIR[TIMEOUT] = '1';
    goto END;
  }
  else goto Response;
}
```

☐ **SUCCESS**

If ARP Reply is received, Peer MAC Address is stored in SLPHAR.

```
{
  /* get Destination MAC Address */
  destination_mac[0:5] = SLPHAR[0:5];
  goto END;
}
```

4.7.2 PING Command (SLCR [PING] = '1')

PING Request Packet is transmitted by SLCR [PING] Command. Before SLCR [PING] Command, SLPIPR, PINGSEQR and PINGIDR must be set. If PING Reply is received from Peer, SLIR [PING] is set to '1' and Peer MAC Address is written in SLPHAR. If there is no response against of PING Request within SOCKET-less Retransmission Time, SLIR [TIMEOUT] is set to '1'. (Ref *4.8.1 ARP & PING Retransmission*)

☐ **Configuration**

Configure SOCKET-less Retransmission Time and PING Interrupt Mask Bit and Destination Peer IP Address.

```
{
START :
  /* set SOCKET-less Retransmission Time, 100ms(0x03E8) (The unit is 100us) */
  SLRTR[0:1] ={ 0x03, 0xE8};
```

```
  /* set SOCKET-less Retransmission Counter, 5 */
  SLRCR = 0x05;

  /* set Interrupt Mask Bit */
  SLIMR[PING] = '1';          // PING Interrupt Mask Bit
  SLIMR[TIMEOUT] = '1';    // TIMEOUT Interrupt Mask Bit

  /* set Destination IP Address, 192.168.0.100 */
  SLPIPR[0:3] = {0xC0, 0xA8, 0x00, 0x64};

  /* set PING Sequence Number, 1000(0x03E8) */
  PINGSEQR[0:1] = {0x03, 0xE8};

  /* set PING ID, 256(0x0100) */
  PINGIDR[0:1] = {0x01,0x00};
}
```

☐ **SOCKET-less Command**

PING Request Packet is transmitted by PING Command.

```
{
  /* set PING Command */
  SLCR[PING] = '1';
  while(SLCR != 0x00) ; /* Wait until PING Command is completed*/
}
```

☐ **Response?**

PING Interrupt indicates whether PING Reply has been received.

```
{
  /* check PING Interrupt */
  if(SLIR[PING] == '1')   /* received PING Reply Packet */
  {
  /* clear Interrupt */
    SLIR[PING] = '1';
    goto SUCCESS;
  }
  else goto Timeout;
}
```

☐ **Timeout? / SUCCESS**

Refer to *4.7.1 ARP Request (SLCR [ARP] = '1') Timeout? / SUCCESS*.

4.8 Retransmission

4.8.1 ARP & PING Retransmission

When there is no Response from Peer against of ARP or PING Request Packet, ARP & PING Retransmission is performed.

In Retransmission Process, Request Packet is retransmitted every RTR until the Response Packet received. And if the number of Retransmission exceeds RCR, TIMEOUT occurs for the Request Packet.

The below table shows Retransmission TIMEOUT (*ARP_{TO}, $PING_{TO}$*).

ARP_{TO}, $PING_{TO}$ = ($TIMEOUT_{VAL}$ x 0.1ms) x ($TIMEOUT_{CNT}$ + 1) $TIMEOUT_{VAL}$ = SLRTR or Sn_RTR $TIMEOUT_{CNT}$ = SLRCR or Sn_RCR

Ex) $TIMEOUT_{VAL}$ = 2000(0x07D0), $TIMEOUT_{CNT}$ = 7(0x0007)

ARP_{TO}= 2000 X 0.1ms X (7+1) = 1.6s

ARP_{TO} is caused by SLCR [ARP], Sn_CR [SEND] and Sn_CR [CONNECT]. And it is checked through SLIR [TIMEOUT] or Sn_IR [TIMEOUT].

$PING_{TO}$ is caused by SLCR [PING] and is checked through SLCR [TIMEOUT].

4.8.2 TCP Retransmission

When TCP Mode SOCKET does not receive ACK Packet from Peer against of SYN, FIN or DATA Packet sent, TCP Retransmission is performed.

In TCP Retransmission process, Request Packet is retransmitted every Sn_RTR until Peer ACK Packet received. And if the number of Retransmission exceeds Sn_RCR, SOCKET n TIMEOUT occurs.

The below table shows TCP Retransmission TIMEOUT (*TCP_{TO}*).

$$\mathrm{TCP_{TO}} = \left(\sum_{N=0}^{M} (\mathrm{TIMEOUT_{VAL}} \times 2^{N}) + ((\mathrm{TIMEOUT_{CNT}} - M) \times \mathrm{TIMEOUT_{MAXVAL}}) \right) \times 0.1\mathrm{ms}$$

N : Number of Retransmission, $0 \leq N \leq M$

M : Minimum value of $TIMEOUT_{VAL} \times 2^{(M+1)}$ >65535 and $0 \leq M \leq \mathrm{TIMEOUT_{CNT}}$

$TIMEOUT_{VAL}$ = SLRTR or Sn_RTR

$TIMEOUT_{CNT}$ = SLRCR or Sn_RCR

$TIMEOUT_{MAXVAL}$: $TIMEOUT_{VAL} \times 2^{M}$

Ex) RTR = 2000(0x07D0), RCR = 7(0x0007)

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

= (2000 + 4000 + 8000 + 16000 + 32000 + ((7 - 5) X 64000)) X 0.1ms

= 190000 X 0.1ms = 19.0s

TCP_{TO} is occurred by CONNECT, SEND and DISCON Command in Sn_CR and it is checked through Sn_IR [TIMEOUT].

4.9 Others Function

4.9.1 System Clock(SYS_CLK) Switching

SYS_CLK is set 25MHz or 100MHz. It is switched by MR2 [CLKSEL], PHYCR1 [RST] or PHYCR1 [PWDN]. At switching Clock speed, Host must wait until stabilized SYS_CLK. (Ref *7.4.1 Reset Timing*)

MR2[CLKSEL]	PHYCR1[RST]	PHYCR1[PWDN]	SYS_CLK(MHz)
0	1	X	25
0	0	0	100 (Default)
0	1	1	25
1	X	X	25

4.9.2 Ethernet PHY Operation Mode Configuration

PHY Operation Mode (Speed, Duplex) set by PHYCR0 is applied after Ethernet PHY H/W Reset. Set PHY Operation Mode is checked through PHYSR [5:3] and the state of Link is described on PHYSR [2:0] after Ethernet PHY Link Up.

Before PHYCR0 is set, PHYLCKR must be unlocked.

Ex) How to set PHY Operation Mode

```
PHY_10FDX :
{
  /* PHYCR0 & PHYCR1 Unlock */
  PHYLCKR = 0x53;

  /* set PHYCR0 10Mbps/Full-Duplex */
  PHYCR0[DPX] = '0';      // 0 - FDX, 1 - HDX;
  PHYCR0[SPD] = '1' ;     // 0 - 100Mbps, 1 - 10Mbps;
  PHYCR0[AUTO] = '1';     // 0 - Auto-negotiation, 1 - Manual;

  /* PHY Reset Process */
  PHYCR1[RST] = '1';
  Wait  TPRST;    // refer to 7.4.1 Reset Timing

  /*wait until PHY Link is up*/
  while(PHYSR[LINK] != '0');

  /* read PHYSR */
  If( (PHYSR[DPX] == '0' ) & ( PHYSR[SPD] == '1') & (PHYSR[AUTO] == '1') ) SUCCESS;
  else FAIL;
```

```
  /* PHYCR0 & PHYCR1 Lock */
  PHYLCKR = 0x00;     // for Lock, write any value
}
```

4.9.3 Ethernet PHY Parallel Detection

When Link Partner does not support Auto-negotiation, Internal Ethernet PHY forms Link through Parallel Detection.

If Duplex mode is different (such as 10F/10H), it occurs of Network performance degradation.

PHY \ Link Partner	Auto	10H	10F	100H	100F
Auto	100F / 100F	10H / 10H	10F / 10H	100H / 100H	100F / 100H
Manual 10H	10H / 10H	10H / 10H	10F / 10H		
Manual 10F	10H / 10F	10H / 10F	10F / 10F		
Manual 100H	100H / 100H			100H / 100H	100F / 100F
Manual 100F	100H / 100F			100H / 100F	100F / 100F

4.9.4 Ethernet PHY Auto MDIX

Ethernet PHY supports Auto-negotiation and it is set by PHYCR0 [AUTO]. If Auto-negotiation is set (PHYCR0 [AUTO] = '0'), Auto-MDIX is supported and Symmetric Transformer (Figure 30 Transformer Type) is operated.

If Auto-negotiation is not set (PHYCR0 [AUTO] = '1'), Cross UTP cable must be needed because Auto-MDIX is not supported.

Ref) if either of Linked Nodes supports Auto-MDIX, Both straight and Cross UTP Cable are usable.

4.9.5 Ethernet PHY Power Down Mode

When PHYCR1 [PWDN] is set '1', Ethernet PHY becomes Power Down Mode and SYS_CLK is set 25MHz.

On the other hand, if PHYCR1 [PWDN] is set '0', Ethernet PHY becomes Normal Mode and SYS_CLK is set by MR2 [CLKSEL]. (Ref *3.1.19 MR2 (Mode Register 2)*)

```
Enter Power Down Mode :
```

```
{
   /* PHYCR0 & PHYCR1 Unlock */
   PHYLCKR = 0x53;

   /* Enable Power Down Mode */
   PHYCR1[PWDN]   = '1';

   /* PHYCR0 & PHYCR1 Lock */
   PHYLCKR = 0x00; // for Lock, write any value

   /* wait until clock is stable switched */
   Wait TPRST; // refer to 7.4.1 Reset Timing
}

Exit Power Down Mode :
{
   /* PHYCR0 & PHYCR1 Unlock */
   PHYLCKR = 0x53;

   /* enable Power Down Mode */
   PHYCR1[PWDN]   = '0';

   /* PHYCR0 & PHYCR1 Lock */
   PHYLCKR = 0x00; // for Lock, write any value

   /* wait until clock is stable switched */
   Wait TPRST; // refer to 7.4.1 Reset Timing

   /* wait until clock is switched 25 to 100MHz*/
   Wait TLF; // refer to 7.4.1 Reset Timing
}
```

4.9.6 Ethernet PHY's Registers Control

The Registers in Ethernet PHY are accessed by MDC/MDIO (Management Data Clock / Input Output) Interface. W5100S has MDC/MDIO Controller and it is controlled by PHYDIVR, PHYRAR, PHYDOR, PHYDIR and PHYACR.

Ex) BMCR Write

```
{
```

```
  PHYRAR = 0x00;    // BMCR Address 0x00
  PHYDIR = 0x80;    // BMCR[15] = '1' , PHY SW Reset

  /* write */
  PHYACR = 0x01;
  while(PHYACR != 0); // wait until MDC/MDIO Control is completed
}
```

Ex) BMSR Read

```
{
  PHYRAR = 0x01; // BMSR Address 0x01

  /* read */
  PHYACR = 0x02;
  while(PHYACR != 0); // wait until MDC/MDIO Control is completed

  if( PHYDOR & 0x0004)
  {
    // LINK UP - BMSR[2] = '1'
  }
}
```

5 HOST Interface Mode

5.1 SPI Mode

When MOD [3:0] is “0000”, W5100S connected to HOST via SPI Interface as shown in Figure 15. W5100S supports SPI Mode 0 and Mode 3 as shown in Figure 16. MOSI is sampling at Rising-edge, MISO is toggling at Falling-edge. MOSI and MISO are transmitted and received sequentially from MSB to LSB for each SCLK.

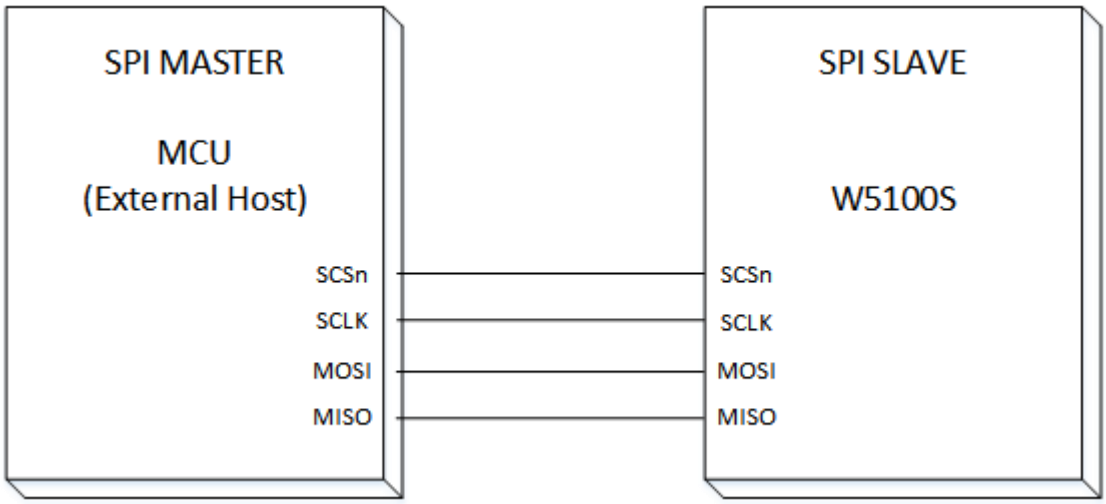


Figure 15 SCSn controlled by Host

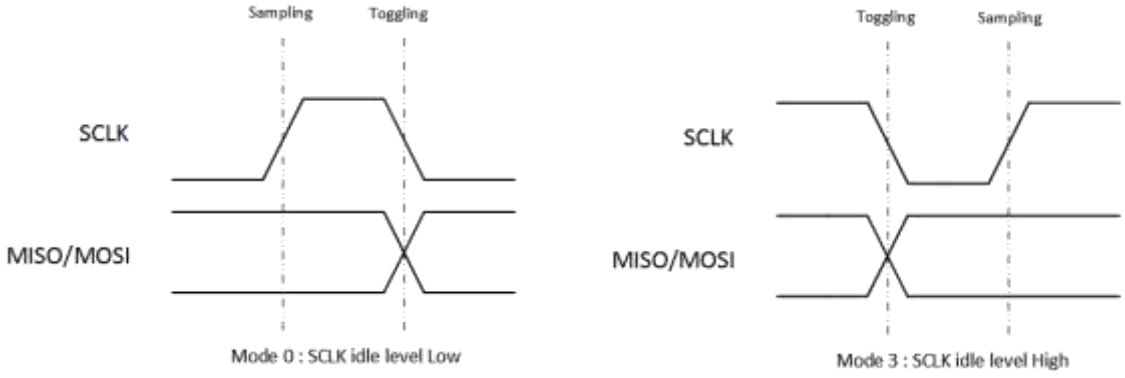


Figure 16 SPI Mode 0 & Mode 3

5.1.1 SPI Frame

W5100S is controlled by SPI Frame (Figure 17 SPI Frame) and is sent by HOST. SPI Frame consists of Control Phase, Address Phase and Data Phase. CSn must be controlled by SPI Frame Unit. W5100 supports only 1-Byte Data read/write but W5100S supports sequential N-Bytes (N = 1, 2, 3, …) Data Read/Write.

Figure 17 SPI Frame

☐ **Control Phase**

Control Phase consists of 8bits to set Read/Write Access Type.

HOST asserts CSn from High to Low and, must after 2 SYS CLK, transmits Control Phase.

Table 6 W5100 Mode SPI Command

Access Type	Value
Write	0xF0
Read	0x0F

☐ **Address Phase**

Address Phase consists of 16bits to set Register Block of the W5100S or 16bits start Offset Address of TX/RX Memory Block. Start Offset Address is automatically incremented by 1 for next Data Access.

☐ **Data Phase**

Data phase consists of N-bytes data. HOST completes Data Access, must after 2 SYS CLK, de-asserts CSn from Low to High.

5.1.2 SPI Write

N-Bytes Date Write SPI Frame, as Shown in Figure 18

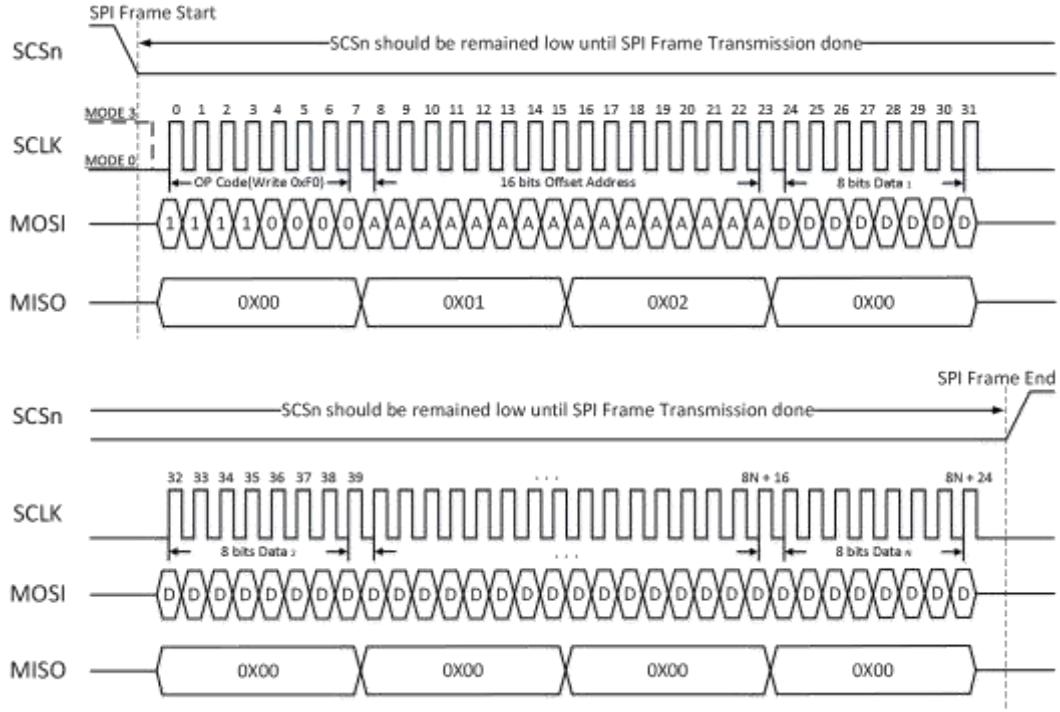


Figure 18 W5100 Mode Write SPI Frame

5.1.3 SPI Read

N-Bytes Date Read SPI Frame, as Shown in Figure 19

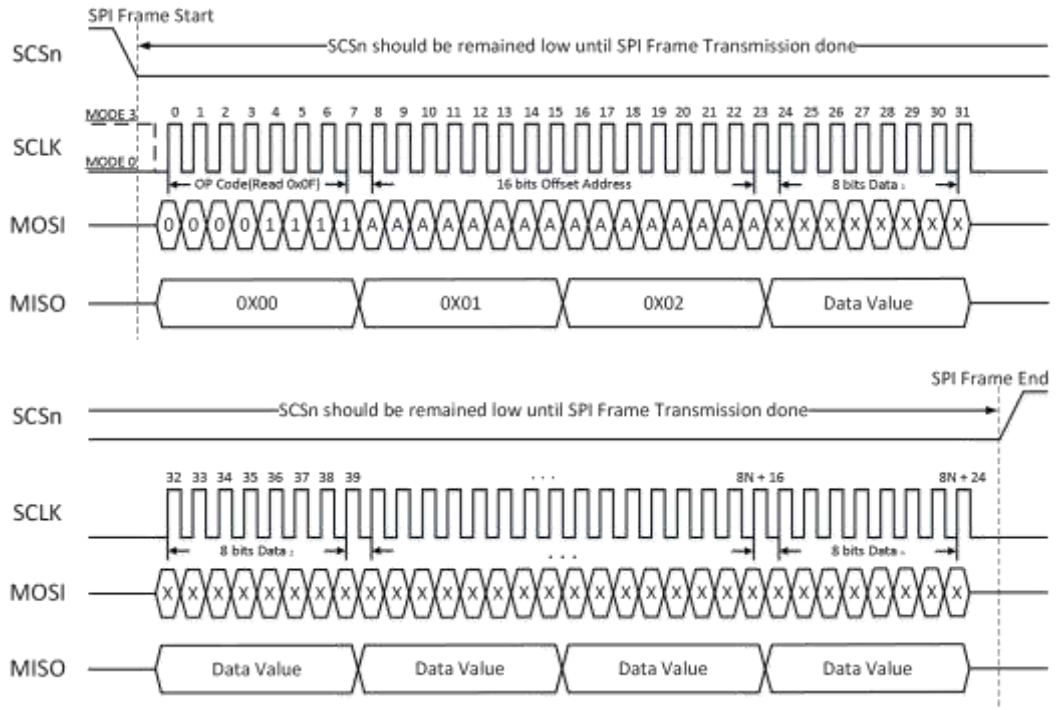


Figure 19 W5100 Mode Read SPI Frame

5.2 Parallel Bus Mode

When MOD [3:0] is “010X”, W5100S connected to HOST via Parallel Bus Interface as shown in Figure 20.

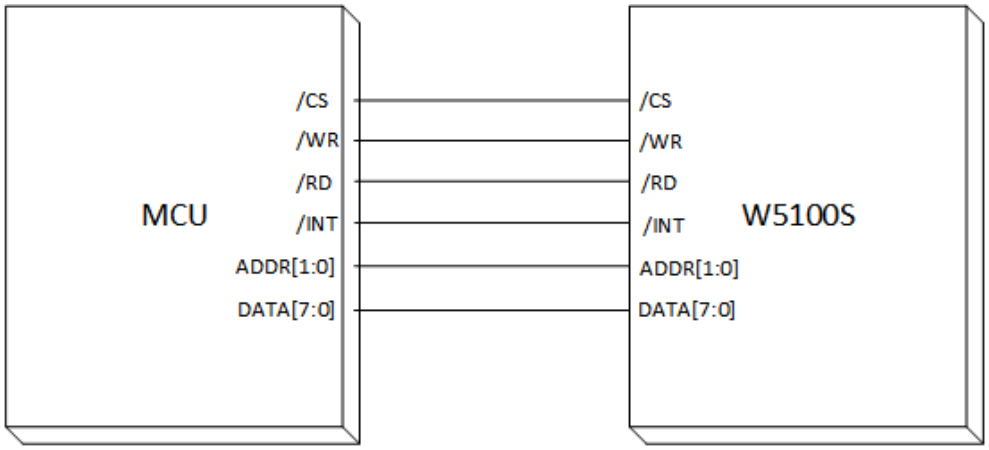


Figure 20 Direct & Indirect Mode Control by Host

Parallel Bus Interface accesses COMMON/SOCKET Register and TX/RX Memory Block of W5100S through Register in Table 7. And also Parallel Bus Interface supports N-byte sequential Data Read/Write.

Table 7 Indirect Mode Address Value

ADDR[1:0]	Symbol	Description
00	MR	Common Register MR
01	IDM_ARH	Upper 8 bits Offset Address Register
10	IDM_ARL	Lower 8 bits Offset Address Register
11	IDM_DR	8 Bits Data Register

5.2.1 Parallel Bus Data Write

N-byte Date write Frame, as Shown in Figure 21.

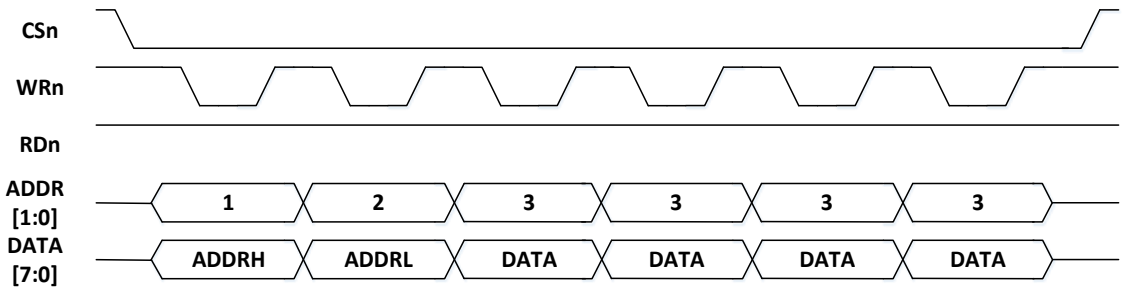


Figure 21 Parallel Bus N-Bytes Data Write

5.2.2 Parallel Bus Data Read

N-byte Date Read Frame, as Shown in Figure 22.

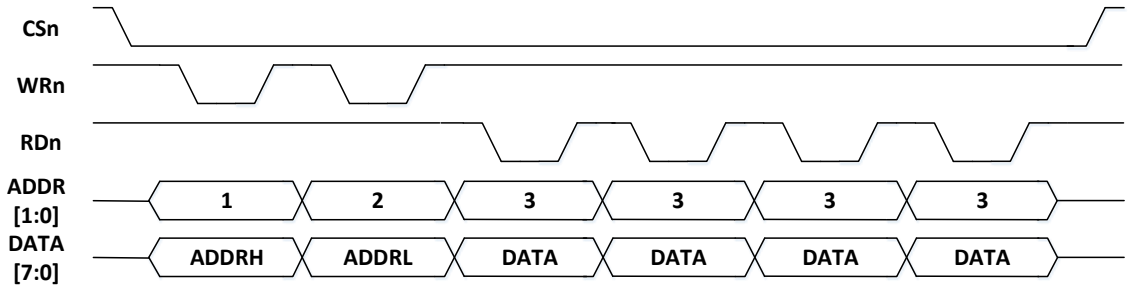


Figure 22 Indirect Mode Continuous Read Access

6 Clock & Transformer Requirements

6.1 Quartz Crystal requirements.

Table 8 Quartz Crystal

Parameter	Condition / Description	Min	Typ	Max	Unit
Frequency(F)			25		MHz
Frequency Tolerance	At 25°C	-50		+50	ppm
Frequency Stability	1 Year aging.	-50		+50	ppm
Load capacitance(C_L)	ESR = 30 Ω		8		pF
Feedback Resistor(R_F)	External resistor		1M		Ω
Startup time	W5100S Reset			60	ms
Trans-conductance(g_m)			16.7		mA/V
gain margin ($gain_{margin}$)	$gain_{margin} = g_m / g_{mcrit}$	6.99			dB

C_0(1) : The Packaging Parasitic Shunt Capacitance.
C_L(1) : Load Capacitance. eq) $C_L = (C_{L1} X C_{L2}) / (C_{L1} X C_{L2}) + C_s$
C_{L1} , C_{L2} : External Capacitances of the circuit connected to the crystal (Typically, $C_{L1} = C_{L2}$)
C_s : Stray Capacitance of printed circuit board and connections.
g_{mcrit} : Oscillator loop critical gain. eq) $g_{mcrit} = 4 \times (ESR + R_{Ext}) \times (2\pi F)^2 \times (C_0 + C_L)^2$
ESR(1) : Maximal equivalent series resistance. eq) $ESR = R_m X (1 + C_0/C_L)^2$
R_{Ext} : Resistor for limiting the drive level(DL) of the crystal.
DL(1) : The power dissipated in the crystal. Excess power can destroy the crystal.
R_F(2) : Feedback resistor.

- *C0, CL, ESR and DL are provided by the crystal manufacturer.*
- *The W5100S has no feedback resistor. Therefore, it must be inserted outside.*

* Crystal circuit is modeled as below Figure 23.

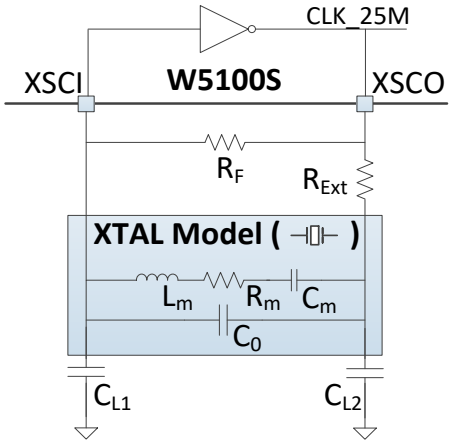


Figure 23 Quartz Crystal Model

Table 9 Crystal Recommendation Characteristics

Parameter	Range

Frequency	25 MHz
Frequency Tolerance (at 25℃)	±30 ppm
Shunt Capacitance	7pF Max
Drive Level	500uW
Load Capacitance	12pF
Aging (at 25℃)	±3ppm / year Max

6.2 Oscillator requirements.

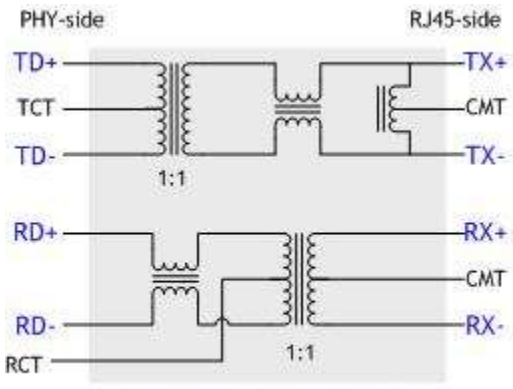
Table 10 Oscillator Characteristics

Parameter	Condition / Description	Min	Typ	Max	Unit
Frequency		25			MHz
Frequency Tolerance	At 25°C	-50		+50	ppm
Frequency Stability	1 Year aging. 25°C	-50		+50	ppm
Clock duty	50% of waveform	45	50	55	%
Input high voltage		-	0.97	-	V
Input low voltage		-	0.13	-	V
Rise/Fall Time	10% to 90% of waveform			8ns	
Start up Time		-	-	10ms	
Operating volatage		1.08V	1.2V	1.32V	
Aging (at 25℃)		±3 / year Max			ppm

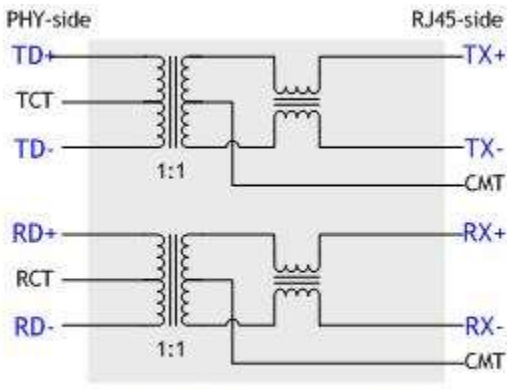
6.3 Transformer Characteristics

Table 11 Transformer Characteristics

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



Asymmetric Transformer



Symmetric Transformer

Figure 24 Transformer Type

7 Electrical Specification

7.1 Absolute Maximum ratings

Table 12 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 4.6	V
V_{OUT}	DC output voltage	-0.5 to 3.63	V
I_{IN}	DC input current	20	mA
T_{OP}	Operating temperature	-40 to +85	°C
T_{JMAX}	Maximum junction temperature	125	°C
T_{STG}	Storage temperature	-65 to +150	°C

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

7.2 Absolute Maximum ratings (Electrical Sensitivity)

Table 13 Electro Static Discharge (ESD)

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

Table 14 Latch up Test

Test Condition	Class	Maximum value	Unit
TA = +25 °C conforming to JESD78	Current	≥ ±100	mA
	Voltage	≥ 1.5*V_{DD}	V

7.3 DC Characteristics

Table 15 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	-	-	V
V_{IL}	Low level input voltage		-		0.8	V
V_{T+}	Schmitt trig Low to High Threshold point	All inputs except Analog PINs	0.8	1.1	-	V
V_{T-}	Schmitt trig High to Low Threshold point	All inputs except Analog PINs	-	1.6	2.0	V
T_J	Junction temperature		-40	25	125	°C
I_L	Input Leakage Current			±1	±10	μA
R_{PU}	Pull-up Resistor		40	75	190	Kohm
R_{PD}	Pull-down Resistor	RSVD(Pin 23, Pin 38 ~ Pin 42)	30	75	190	Kohm
V_{OL}	Low level output voltage	IOL = 2.0mA~ 8.0mA, All outputs except XSCO			0.4	V
V_{OH}	High level output voltage	IOH = 2.0mA~8.0mA, All outputs except XSCO	2.4			V

7.4 AC Characteristics

7.4.1 Reset Timing

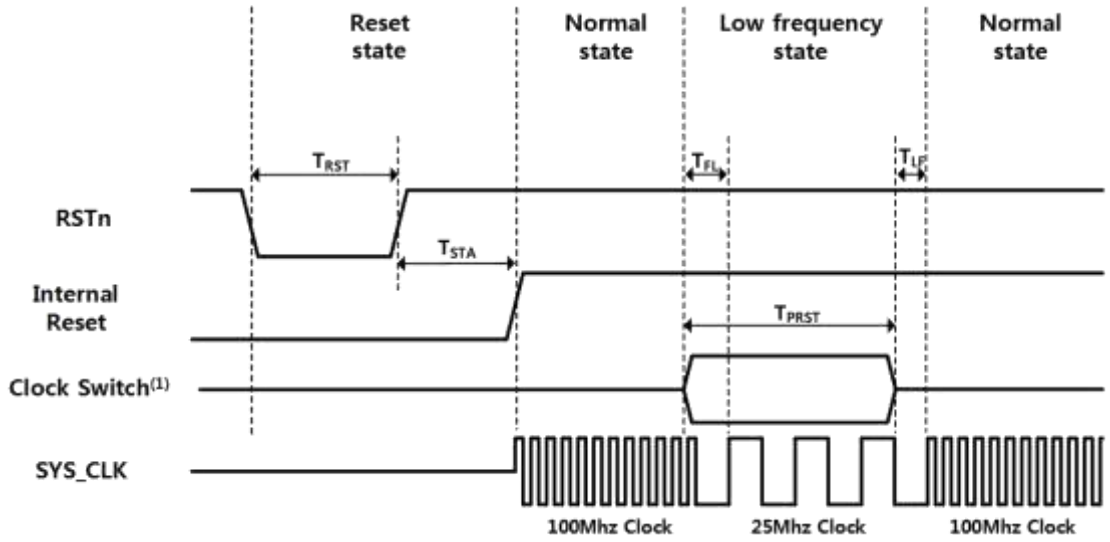


Figure 25 Reset Timing

Table 16 Reset Table

Symbol	Description	Min	Typ	Max
T_{RST}	Reset Time	210 ns	330 ns	560 ns-
T_{STA}	Stable Time	-		60.3 ms
T_{FI}	Fast to Low Time by MR2[CLKSEL]	100 ns		-
T_{FI}	Fast to Low Time by PHYCR1[Reset] or PHYCR1[PWDN]	300 ns		
T_{PRST}	PHY Auto Reset Time	0.6 ms		-
T_{PRST}	PHY Power Down Time	200 us		
T_{PRST}	Clock Switch Time	200 ns		
T_{LF}	Low to Fast Time by MR2[CLKSEL]	100 ns		-
T_{LF}	Low to Fast Time by PHYCR1[Reset] or PHYCR1[PWDN]	100 ns		

***COMMENT**: PHY Power-down Mode has T_{FI} and T_{LF} (In PHY Power-down Mode, SYS_CLK switches to Low Clock. After T_{FI} , User can be disable PHY Power-down Mode.)*

***CAUTION**: User must not set PHY Auto Reset and PHY Power-down Mode at the same time*

7.4.2 BUS ACCESS TIMING

7.4.2.1 READ TIMING

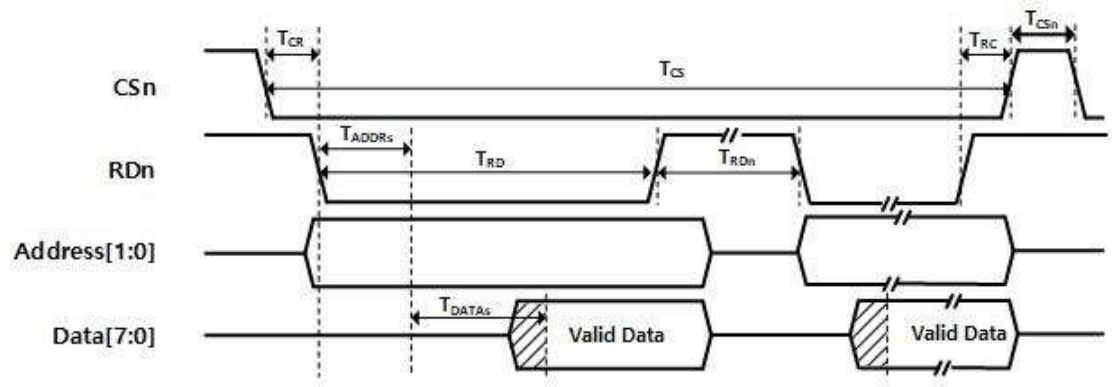


Figure 26 Bus Read Timing

Table 17 BUS Read Timing

Symbol	Description	Min	Max
T_{ADDRs}	Address Setup Time	SYS_CLK	
T_{CR}	CSn Low to RDn Low Time	0 ns	
T_{cs}	CSn Low Time	4 SYS_CLK	
T_{RC}	RDn High to CSn High Time	0 ns	
T_{csn}	CSn Next Assert Time	3 SYS_CLK	
T_{RD}	RDn Low Time	4 SYS_CLK	
T_{RDn}	RDn Next Assert Time	3 SYS_CLK	
T_{DATAs}	DATA Setup Time		3 SYS_CLK+5ns

7.4.2.2 WRITE TIMING

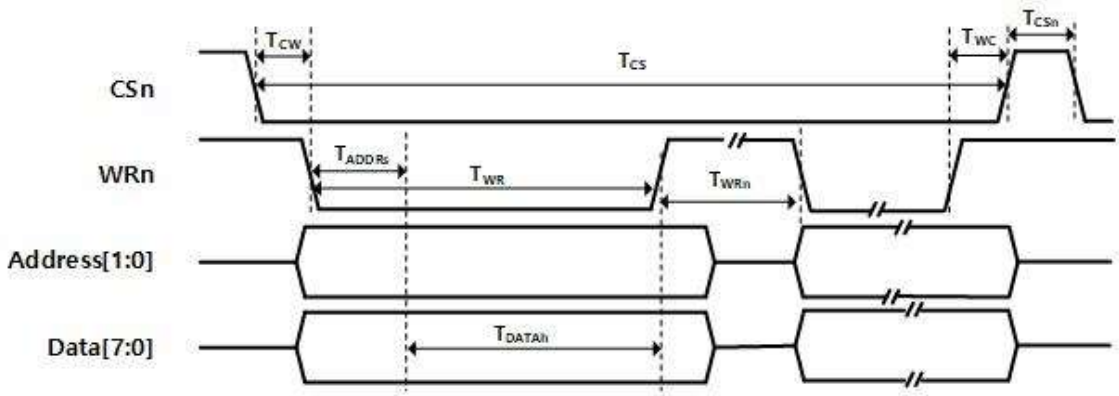


Figure 27 BUS Write Timing

Table 18 BUS Write timing

Symbol	Description	Min	Max
T_{ADDRs}	Address Setup Time	SYS_CLK	
T_{CW}	CSn Low to WRn Low Time	0 ns	
T_{cs}	CSn Low Time	4 SYS_CLK	
T_{WC}	WRn High to CSn High Time	0 ns	
T_{csn}	CSn Next Assert Time	3 SYS_CLK	
T_{WR}	WRn Low Time	4 SYS_CLK	
T_{WRn}	WRn Next Assert Time	3 SYS_CLK	
T_{DATAs}	DATA Setup Time	2 SYS_CLK	

7.4.3 SPI ACCESS TIMING

7.4.3.1 SPI READ TIMING

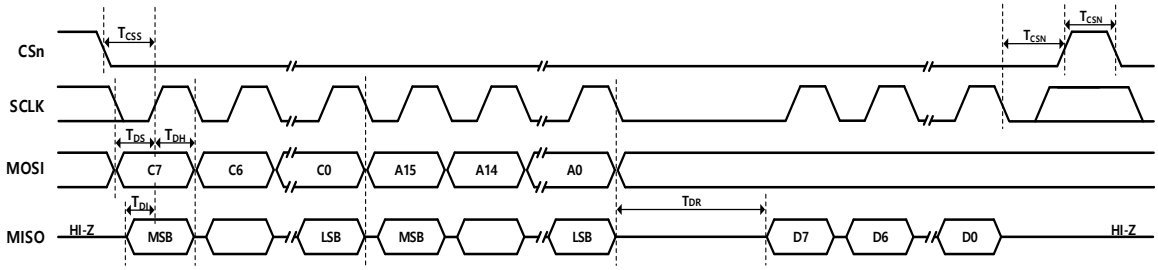


Figure 28 SPI Read Timing

Table 19 SPI Read Timing

Symbol	Description	Min	Max	Units
F_{SCK}	SCK Clock Frequency		70	MHz
T_{CSS}	SCSn Setup Time	3 SYS_CLK		ns
T_{CSN}	SCSn Next Time	2 SYS_CLK		ns
T_{DS}	Data In Setup Time	3		ns
T_{DH}	Data In Hold Time	3		ns
T_{DI}	Data Invalid Time	7		ns
T_{DR}	Data Ready Time	6 SYS_CLK + 30		ns

7.4.3.2 SPI WRITE TIMNIG

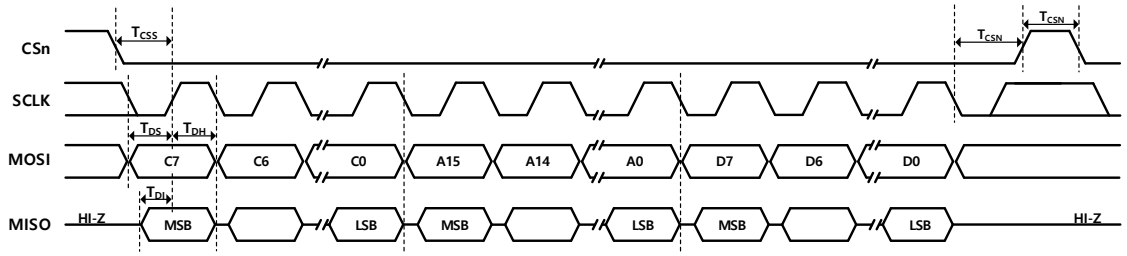


Figure 29 SPI Write Timing

Table 20 SPI Write Timing

Symbol	Description	Min	Max	Units
F_{SCK}	SCK Clock Frequency		70	MHz
T_{CSS}	SCSn Setup Time	3 SYS_CLK		ns
T_{CSN}	SCSn Next Time	2 SYS_CLK		ns
T_{DS}	Data In Setup Time	3		ns
T_{DH}	Data In Hold Time	3		ns
T_{DI}	Data Invalid Time	7		ns

7.4.4 Transformer Characteristics

Table 21 Transformer Characteristics

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

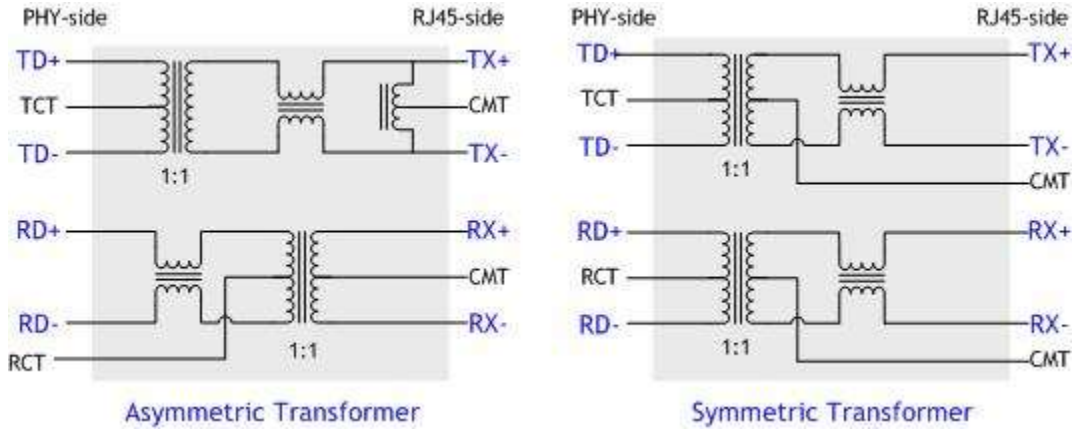


Figure 30 Transformer Type

7.4.5 MDIX

W5100S supports Auto-MDIX on Auto-Negotiation Mode.

7.5 POWER DISSPATION

Table 22 Power Dissipation

(Test Condition: VDD=3.3V, AVDD=3.3V, Ta = 25°C)

Condition	Min	Typ	Max	Unit
100M Link	-	93	110	mA
10M Link	-	93	210	mA
100M Unlink(Actual Measurement)	-	40	170	mA
10M Unlink(Actual Measurement)	-	25	150	mA
Un-Link (Auto-negotiation mode) (Actual Measurement)	-	43	170	mA
Power Down mode	-	10	20	mA

8 Package Information

8.1 LQFP48

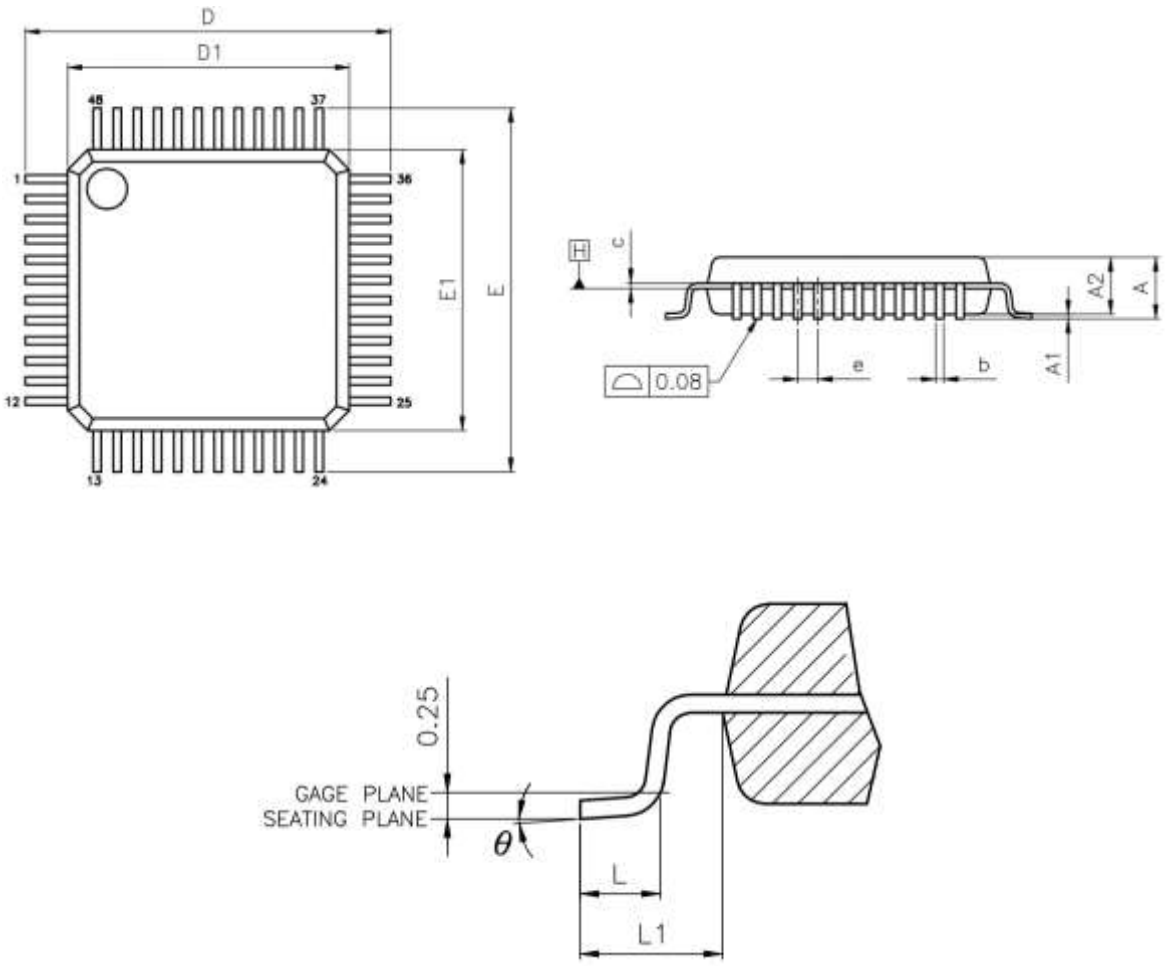


Table 23 LQFP48 VARIATIONS (ALL DEMINSIONS SHOWN IN MM)

SYMBOL	MIN	NOM	MAX
A	--	--	1.60
A1	0.05	--	0.15
A2	1.35	1.40	1.45
b	0.17	0.22	0.27
c	0.09	--	0.20
D	9.00 BSC		
D1	7.00 BSC		
E	9.00 BSC		
E1	7.00 BSC		
e	0.50 BSC		
L	0.45	0.60	0.75

L1	1.00 REF		
θ	0°	3.5°	7°

NOTES:

1. JEDEC OUTLINE:
   MS-026 BBC
   MS-026 BBC-HD (THERMALLY ENHANCED VARIATIONS ONLY)
2. DATUM PLANE H IS LOCATED AT THE BOTTOM OF THE MOLD PARTING LINE COINCIDENT WITH WHERE THE LEAD EXITS THE BODY
3. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 mm PER SIDE. DIMENSIONS D1 AND E1 DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE H.
4. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.

8.2 QFN48

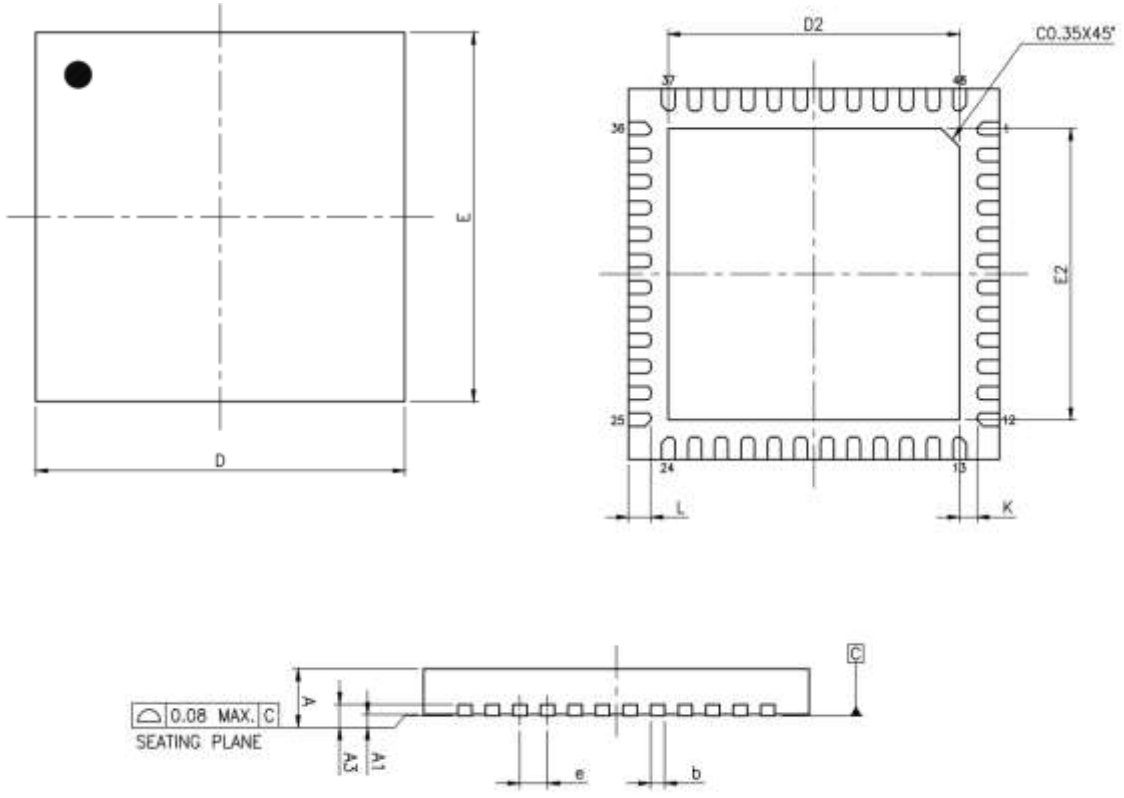


Table 24 QFN48 VARIATIONS (ALL DEMINSIONS SHOWN IN MM)

SYMBOL	MIN	NOM	MAX
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
A3	0.203 REF		
b	0.20	0.25	0.30
D	7.00 BSC		

E	7.00 BSC			
e	0.50 BSC			
D2	5.25	5.30	5.35	
E2	5.25	5.30	5.35	
L	0.35	0.40	0.45	
K	0.20	--	--	
LEAD FINISH	Pure Tin	V	PPF	X
JEDEC CODE	N/A			

NOTES:

1. ALL DIMENSIONS ARE IN MILLIMETERS
2. DEMENSION B APPLIES TO METALLIZED TERMINAL AND IS MEASURED BETWEEN 0.15mm AND 0.30mm FROM THE TERMINAL TIP. IF THE TERMINAL HAS THE OPTIONAL RADIUS ON THE OTHER END OF THE TERMINAL, THE DIMENSION b SHOULD NOT BE MEASURED IN THAT RADIUS AREA.
3. BILATERAL COPLANARITY ZONE APPLIES TO THE EXPOSED HEAT SINK SLUG AS WELL AS THE TERMINALS.

9 Document Revision History

Version	Date	Descriptions
Ver. 1.0.0	1APR2018	Initial Release
Ver. 1.0.1	2ARP2019	Edit typo & incorrect link
Ver. 1.1.0	5SEP2018	1. corrected reset timing in RSTn pin (500ns -> 560ns) (*1.1 PIN Description*) 2. corrected INTPTMR register type (RW -> R=W) (*3.1.6 INTPTMR* (Interrupt Pending Time Register)) 3. modified PHYCR0 Bits symbol (AUTO/SPD/DPX->MODE[3:0]) (*3.1.30 PHYCR0* (PHY Control Register 0)) 4. added register, PHYSR1 (*3.1.24 PHYSR1* (PHY Status Register 1)) 5. modified the complete condition,PHYSR1[ACT]='1', of reset by PHYCR1[RST] (*3.1.31 PHYCR1* (PHY Control Register 1)) 6. corrected NETLCKR value, Lock (0x3A -> 0xC5), Unlock (0xC5 -> 0x3A) (*3.1.42 NETLCKR* (Network Lock Register)) 7. modified UDP received Data Size (*Figure 9*) 8. modified IPRAW received Data Size (*Figure 11*)
Ver. 1.2.0	13NOV2018	1. modified BUS access timing (*7.4.2 BUS ACCESS TIMING*) Added T_{WRn}, T_{RDn}
Ver. 1.2.1	14NOV2018	1. modified BUS access timing (*7.4.2 BUS ACCESS TIMING*) Modified Burst Mode
Ver. 1.2.2	23NOV2018	1. Modified Sn_MSS reset value (*3.2.9 Sn_MSS (SOCKET n Maximum Segment Size Register)*) 2. Modified Load capacitance value (12p -> 8p) (*6.1 Quartz Crystal requirements.*)
Ver. 1.2.3	7JAN2019	1. Modified DC Characteristics (in *7.3 DC Characteristics*)
Ver. 1.2.4	7MAR2019	1. Modified Trans-conductance(g_m)(8.43 -> 16) (*6.1 Quartz Crystal requirements.* 2. Modified pin name of SPI/BUS Timing figure (in *7.4.2 BUS ACCESS TIMING , 7.4.3 SPI ACCESS TIMING*)

		3. Edit typo PHYCR0[RST] -> PHYCR1[0] Modify PHYCR1[RST] value (in *4.9.1 System Clock(SYS_CLK) Switching*) 4. Modified Power Dissipation (in *7.5 POWER DISSPATION*)
Ver. 1.2.5	15MAY2019	1. Added Maximum junction temperature (in *7.1 Absolute Maximum ratings*)
Ver. 1.2.6	21FEB2022	1. Modified register description about initial value (in *3.1.41 CLKLCKR (Clock Lock Register), 3.1.42 NETLCKR (Network Lock Register), 3.1.43 PHYLCKR (PHY Lock Register)*)
Ver. 1.2.7	17MAY2022	1. Corrected RCR reset value '0x08 -> 0x07' (*in 3.1.10 RCR (Retransmission Count Register)*) and related examples (*4.8.1 ARP & PING Retransmission, 4.8.2 TCP Retransmission*)