**W5100S Datasheet**

**Version 1.0.0**



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

**W5100S**

W5100S은 WIZnet의 Hardwired TCP/IP 기술을 이용한 Embedded Internet Controller Chip이다. W5100S는 Internet protocol (TCP/IP) 처리를 위한 Full Hardwired Logic, Ethernet MAC(Media Access Control), 그리고 10 BaseT / 100BaseTX Ethernet PHY를 모두 내장한 Internet connectivity One-chip Solution이다.

W5100S는 Host(사용자 MCU)의 부담을 최소화할 수 있도록 TCP, UDP, ICMP, IPv4, IGMP, ARP, PPPoE등의 다양한 통신 프로토콜을 Hardwired Logic으로 처리할 수 있고, Low-end급 Host의 메모리를 극소화할 수 있도록 데이터 송신/수신에 필요한 8KB의 전용 버퍼 메모리를 각각 내장하고 있다. 또한 사용자는 W5100S의 독립적인 Hardwired SOCKET 4개를 동시에 사용하여 다양한 Internet응용제품을 개발할 수 있다.

W5100S는 Host와의 통신을 위해 Serial 방식의 고속 SPI Interface 와 Parallel 방식의 System BUS Interface를 지원한다. 또한, 시스템의 소비전력을 낮추기 위해 저전력/저발열 설계, WOL (Wake On LAN)와 Ethernet Power Down mode등를 제공한다.

W5100S는 제품명 그대로 기존 W5100제품을 개선한 저가형 제품이다. W5100S는 W5100에서 사용되는 Firmware를 수정하지 않고 곧바로 사용할 수 있는 장점을 가지고 있으며, W5100보다 작은 Package(LQFP48)를 채택하여 제품을 소형화할 수 있다.

**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 gating 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
* 10BaseT/100BaseTX Ethernet PHY Integrated
* Support Auto Negotiation (Full and half duplex, 10 and 100-based )
* Not support IP Fragmentation
* 3.3V operation with 5V I/O signal tolerance
* LED outputs (Full/Half duplex, Link, 10/100 Speed, Active)
* 48 Pin LQFP Lead-Free Package (7x7mm, 0.5mm pitch)

**Target Applications**

W5100S은 다음과 같은 Embedded application에 적합하다.

* 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**



**Table of Contents**

[1 PIN Description 11](#_Toc483830015)

[1.1 PIN Description 12](#_Toc483830016)

[2 Memory Map 18](#_Toc483830017)

[3 W5100S Registers 20](#_Toc483830018)

[3.1 Common registers 20](#_Toc483830019)

[3.2 SOCKET Registers 3](#_Toc483830020)

[4 Register Descriptions 6](#_Toc483830021)

[4.1 Common Registers 6](#_Toc483830022)

[4.1.1 MR (Mode Register) 6](#_Toc483830023)

[4.1.2 GWR (Gateway IP Address Register) 7](#_Toc483830024)

[4.1.3 SUBR (Subnet Mask Register) 7](#_Toc483830025)

[4.1.4 SHAR (Source Hardware Address Register) 7](#_Toc483830026)

[4.1.5 SIPR (Source IP Address Register) 7](#_Toc483830027)

[4.1.6 IR (Interrupt Register) 7](#_Toc483830028)

[4.1.7 IMR (Interrupt Mask Register) 4](#_Toc483830029)

[4.1.8 RTR (Retry Timer-value Register) 5](#_Toc483830030)

[4.1.9 RCR (Retry Counter-value Register) 5](#_Toc483830031)

[4.1.10 RMSR(RX Memory Size Register) 6](#_Toc483830032)

[4.1.11 TMSR(TX Memory Size Register) 6](#_Toc483830033)

[4.1.12 IR2(Interrupt Register 2) 7](#_Toc483830034)

[4.1.13 IMR2(Interrupt Mask Register 2) 7](#_Toc483830035)

[4.1.14 PTIMER(PPP Link Control Protocol Request Timer Register) 8](#_Toc483830036)

[4.1.15 PMAGIC(PPP Link Control Protocol Magic number Register) 8](#_Toc483830037)

[4.1.16 UIPR(Unreachable IP Address Register) 8](#_Toc483830038)

[4.1.17 UPORTR(Unreachable Port Register) 9](#_Toc483830039)

[4.1.18 MR2(Mode Register 2) 9](#_Toc483830040)

[4.1.19 PHAR(Destination Hardware Address Register in PPPoE Mode) 10](#_Toc483830041)

[4.1.20 PSIDR(Session ID Register in PPPoE Mode) 11](#_Toc483830042)

[4.1.21 PMRUR(PPPoE Maximum Receive Unit Register) 11](#_Toc483830043)

[4.1.22 PHY 11](#_Toc483830044)

[4.1.23 Request Command(RQ CMD) 16](#_Toc483830045)

[4.1.24 DBG\_OUT(Debug Out) 4](#_Toc483830046)

[4.1.25 NICMAXCOLR(NIC Maximum Collision Register) 6](#_Toc483830047)

[4.1.26 CHIPLCKR(CHIP Lcok Register) 6](#_Toc483830048)

[4.1.27 NETLCKR(Network Lock Register) 7](#_Toc483830049)

[4.1.28 PHYLCKR(PHY Lock Register) 7](#_Toc483830050)

[4.1.29 VER(W5100S Version Register) 7](#_Toc483830051)

[4.1.30 TCNTR(Ticker Counter Register) 7](#_Toc483830052)

[4.1.31 TCNTCLKR(Ticker Counter Clear Register) 7](#_Toc483830053)

[4.2 SOCKET Register 9](#_Toc483830054)

[4.2.1 Sn\_MR (SOCKET n Mode Register) 9](#_Toc483830055)

[4.2.2 Sn\_CR (SOCKET n Command Register) 10](#_Toc483830056)

[4.2.3 Sn\_IR (SOCKET n Interrupt Register) 12](#_Toc483830057)

[4.2.4 Sn\_SR (SOCKET n Status Register) 13](#_Toc483830058)

[4.2.5 Sn\_PORTR (SOCKET n Source Port Register) 15](#_Toc483830059)

[4.2.6 Sn\_DHAR (SOCKET n Destination Hardware Address Register) 15](#_Toc483830060)

[4.2.7 Sn\_DIPR (SOCKET n Destination IP Address Register) 16](#_Toc483830061)

[4.2.8 Sn\_DPORTR (SOCKET n Destination Port Register) 16](#_Toc483830062)

[4.2.9 Sn\_MSS ( SOCKET n Maximum Segment Size Register) 17](#_Toc483830063)

[4.2.10 Sn\_PROTOR ( SOCKET n IP Protocol Register) 17](#_Toc483830064)

[4.2.11 Sn\_TOS (SOCKET n IP Type Of Service Register) 17](#_Toc483830065)

[4.2.12 Sn\_TTL (SOCKET n IP Time To Live Register) 18](#_Toc483830066)

[4.2.13 Sn\_RXBUF\_SIZE (SOCKET n RX Buffer Size Register) 18](#_Toc483830067)

[4.2.14 Sn\_TXBUF\_SIZE (SOCKET n TX Buffer Size Register) 18](#_Toc483830068)

[4.2.15 Sn\_TX\_FSR (SOCKET n TX Free Size Register) 19](#_Toc483830069)

[4.2.16 Sn\_TX\_RD (SOCKET n TX Read Pointer Register) 20](#_Toc483830070)

[4.2.17 Sn\_TX\_WR (SOCKET n TX Write Pointer Register) 20](#_Toc483830071)

[4.2.18 Sn\_RX\_RSR (SOCKET n RX Received Size Register) 20](#_Toc483830072)

[4.2.19 Sn\_RX\_RD (SOCKET n RX Read Pointer Register) 21](#_Toc483830073)

[4.2.20 Sn\_RX\_WR (SOCKET n RX Write Pointer Register) 21](#_Toc483830074)

[4.2.21 Sn\_IMR (SOCKET n Interrupt Mask Register) 21](#_Toc483830075)

[4.2.22 Sn\_FRAGR (SOCKET n Fragment Register) 22](#_Toc483830076)

[4.2.23 Sn\_MR2 (SOCKET n Mode register 2) 22](#_Toc483830077)

[4.2.24 Sn\_KPALVTR (SOCKET n Keep Alive Timer Register) 23](#_Toc483830078)

[4.2.25 Sn\_TSR (SOCKET n Timer Status Register) 24](#_Toc483830079)

[4.2.26 Sn\_RTR (SOCKET n Retry Time-value Register) 25](#_Toc483830080)

[4.2.27 Sn\_RCR (SOCKET n Retry Count-value Register) 25](#_Toc483830081)

[5 Functional Description 27](#_Toc483830082)

[5.1 Initialization 27](#_Toc483830083)

[5.1.1 Basic Setting 27](#_Toc483830084)

[5.1.2 Network Information Setting 27](#_Toc483830085)

[5.1.3 SOCKET Memory Setting 28](#_Toc483830086)

[5.2 TCP 28](#_Toc483830087)

[5.2.1 TCP Server 30](#_Toc483830088)

[5.2.2 TCP Client 34](#_Toc483830089)

[5.2.3 Other Functions 35](#_Toc483830090)

[5.3 UDP 38](#_Toc483830091)

[5.3.1 UDP Unicast & Broadcast 38](#_Toc483830092)

[5.3.2 UDP Multicast 41](#_Toc483830093)

[5.3.3 Other Functions 43](#_Toc483830094)

[5.4 IPRAW 45](#_Toc483830095)

[5.5 MACRAW 48](#_Toc483830096)

[5.6 Rquest Command 50](#_Toc483830097)

[5.6.1 ARP command 51](#_Toc483830098)

[5.6.2 PING command 51](#_Toc483830099)

[5.6.3 Rquest Command TIMEOUT 52](#_Toc483830100)

[5.7 PHY Control 53](#_Toc483830101)

[5.7.1 PHY Control Register 53](#_Toc483830102)

[5.7.2 MDC/MDIO 53](#_Toc483830103)

[6 External Interface 55](#_Toc483830104)

[6.1 SPI W5100 Mode 55](#_Toc483830105)

[6.1.1 SPI Frame 57](#_Toc483830106)

[6.1.2 Control Phase 57](#_Toc483830107)

[6.1.3 Address Phase 57](#_Toc483830108)

[6.1.4 Data Phase 57](#_Toc483830109)

[6.1.5 SPI Write Access 58](#_Toc483830110)

[6.1.6 SPI Read Access 59](#_Toc483830111)

[6.2 SPI W5500 Mode 59](#_Toc483830112)

[6.2.1 SPI Frame 60](#_Toc483830113)

[6.2.2 Address Phase 60](#_Toc483830114)

[6.2.3 Control Phase 60](#_Toc483830115)

[6.2.4 Data Phase 60](#_Toc483830116)

[6.2.5 SPI Write Access 61](#_Toc483830117)

[6.2.6 SPI Read Access 62](#_Toc483830118)

[6.3 Indirect Operation Mode 63](#_Toc483830119)

[7 Clock 64](#_Toc483830120)

[7.1 Quartz Crystal 64](#_Toc483830121)

[7.2 Oscillator 65](#_Toc483830122)

[8 Electrical Specification 66](#_Toc483830123)

[8.1 Absolute Maximum ratings 66](#_Toc483830124)

[8.2 Absolute Maximum ratings (Electrical Ssensitivity) 66](#_Toc483830125)

[8.3 DC Characteristics 67](#_Toc483830126)

[8.4 POWER DISSPATION 68](#_Toc483830127)

[8.5 AC Characteristics 68](#_Toc483830128)

[8.5.1 Reset Timing 68](#_Toc483830129)

[8.5.2 Wake up Time 68](#_Toc483830130)

[8.5.3 Crystal Characteristics 69](#_Toc483830131)

[8.5.4 SPI Timing 70](#_Toc483830132)

[8.5.5 Transformer Characteristics 71](#_Toc483830133)

[8.5.6 MDIX 71](#_Toc483830134)

[9 IR Reflow Temperature Progile (Lead-Free) 72](#_Toc483830135)

[10 Package Description 73](#_Toc483830136)

[Document History Information 74](#_Toc483830138)

**Table of Figures**

[Figure 1. W5100S Pin Layout 10](#_Toc483830139)

Figure 2 Memory Map [16](#_Toc483830140)

Figure 3. State Diagram [16](#_Toc483830141)

[Figure 4 . TCP SERVER and TCP CLIENT 17](#_Toc483830142)

[Figure 5 . TCP Server Operation Flow 29](#_Toc483830143)

Figure 6 . TCP Client Operation Flow [30](#_Toc483830144)

Figure 7 . UDP Operation Flow [34](#_Toc483830145)

Figure 8. Received Data in UDP Mode SOCKET RX Buffer Block [38](#_Toc483830146)

Figure 9 IPRAW positioned to Network Stack [46](#_Toc483830147)

Figure 10 IPRAW Operation Flow [48](#_Toc483830148)

Figure 11 IPRAW Data in Socket Sending Packet [55](file:///D:\\Wiznet\\IOP4IoT\\W5100S\\DataSheet\\W5100S_DS_V100K_170526(수정중).docx" \l "_Toc483830149)

Figure 12 Received Data in IPRAW Mode SOCKET RX Buffer Block [56](file:///D:\\Wiznet\\IOP4IoT\\W5100S\\DataSheet\\W5100S_DS_V100K_170526(수정중).docx" \l "_Toc483830150)

Figure 13 Ethernet TCP/IP stack with OSI 7 Layers [56](file:///D:\\Wiznet\\IOP4IoT\\W5100S\\DataSheet\\W5100S_DS_V100K_170526(수정중).docx" \l "_Toc483830151)

Figure 14 Ethernet Frame [57](file:///D:\\Wiznet\\IOP4IoT\\W5100S\\DataSheet\\W5100S_DS_V100K_170526(수정중).docx" \l "_Toc483830152)

Figure 15 Received DATA Format in MACRAW [58](#_Toc483830153)

Figure 16. SOCKET-less Command Operation Flow [59](#_Toc483830154)

Figure 17 SCSn controlled by Host [60](file:///D:\\Wiznet\\IOP4IoT\\W5100S\\DataSheet\\W5100S_DS_V100K_170526(수정중).docx" \l "_Toc483830155)

Figure 18 SCSn is always connected by Ground [61](#_Toc483830156)

Figure 19 SPI Mode 0 & 3 [62](#_Toc483830157)

Figure 20 W5100 Mode SPI Frame [63](#_Toc483830158)

Figure 21 W5100 Mode Write SPI Frame [65](#_Toc483830159)

Figure 22 W5100 Mode Read SPI Frame [68](#_Toc483830160)

Figure 23 Direct & Indirect Mode Control by Host [70](#_Toc483830161)

Figure 24 Direct Mode Write Access [71](#_Toc483830162)

Figure 25 Direct Mode Read Access [72](#_Toc483830163)

[Figure 26 Indirect Mode Write Access 73](#_Toc483830164)

Figure 27 Indirect Mode Continuous Write Access [72](#_Toc483830163)

[Figure 28 Indirect Mode Read Access 73](#_Toc483830164)

Figure 29 Indirect Mode Continuous Read Access [72](#_Toc483830163)

[Figure 30. Quartz Crystal Model 73](#_Toc483830164)

Figure 31. Reset Timing [72](#_Toc483830163)

[Figure 32. SPI Timing 73](#_Toc483830164)

[Figure 33. Transformer Type 73](#_Toc483830164)

Figure 34. IR Reflow Temperature [72](#_Toc483830163)

[Figure 35. Package Dimensions 73](#_Toc483830164)

1. PIN Description



Figure 1. 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. PIN Description

Table 2. PIN Description

|  |  |  |  |
| --- | --- | --- | --- |
| **PIN #** | **Symbol** | **Type** | **Description** |
| 1 | GNDA | AP | **Analog Ground** |
| 2 | TXON | AO | **Differential Transmitted Signal Pair**  차동 데이터는 MDI 모드에서 TXOP/TXON 신호 쌍을 통해 미디어로 전송된다. |
| 3 | TXOP | AO |
| 4 | 1V2A | AP | **Analog 1.2V Power**  1V2O 전압 소스로부터 공급받는다. |
| 5 | RXIN | AI | **Differential Received Signal Pair**  차동 데이터를 MDI 모드에서 RXIP/RXON 신호 쌍을 통해 미디어로부터 수신한다. |
| 6 | RXIP | AI |
| 7 | GNDA | AP | **Analog Ground** |
| 8 | 3V3A | AP | **Analog 3.3V Power** |
| 9 | RSET\_BG | AO | **Off-chip Bias Resistor**  외부 12.3KΩ, 오차 1% 저항을 통해 Analog Ground로 반드시 연결해야 한다. |
| 10 | GND | AP | **Digital Ground** |
| 11 | XSCO | AO | **25MHz Clock**  W5100S는 외부로부터 25MHz clock(CLK\_25M)을 입력을 받고, System Operation Clock(SYS\_CLK)으로 1배의 25MHz(Power save mode)이나 4배의 100MHz(internal PLL clock)으로 동작한다.  25MHz Crystal Oscillator(XTAL)나 Oscillator(OSC)로 연결한다.  XTAL이 아닌 OSC를 사용할 경우 25MHz@1.2V를 사용하며 XSCI만 연결하고 XSCO는 반드시 Floating 시킨다.  자세한 내용은 Clock Selection Guide를 참조한다. |
| 12 | XSCI | AI |
| 13 | 1V2D | P | **Digital 1.2V Power**  1V2O 전압 소스로부터 공급받는다. |
| 14 | 1V2O | PO | **Internal Regulator 1.2V Power Output**  W5100S Core를 위한 내부 Regulator의 1.2V Power Output으로 Max 150mA를 지원하며, 반드시 외부 안정화 Capacitor 3.3uF를 통해 W5100S의 1V2D 와 1V2A로 공급한다.  1V2O는 Ferrite Bead를 사용하여, 1V2D와 1V2A 분리하여 공급한다.  이 Power는 W5100S Core를 위한 것으로 다른 Device의 Power로 사용될 수 없다. |
| 15 | 3V3A | AP | **Analog 3.3V Power** |
| 16 | GNDA | AP | **Analog Ground** |
| 17 | LNKn | OUM | **Link Status LED / I\_RSTn**  Link Status LED when MOD[3:0] != 0010 or MOD[3] = 0  Low : Link up  High : Link down  Probing internal I\_RSTn (RSTn 2 sampled signal by 25MHz clock) signal when MOD[3:0] = 0010 |
| 18 | SPDn | OUM | **Link Speed LED / SYS\_RSTn & CLK\_25M\_DV1K**  Link Speed LED when MOD[3:0] != 0010 or MOD[3] = 0  Low : 100Mbps  High : 10Mbps  Probing SYS\_RSTn & CLK\_25M\_DV1K when MOD[3:0] = 0010  RSTn PIN이 Low에서 High로 될 때 W5100S의 내부 System Reset(SYS\_RSTn)이 60.2ms (25MHz XTAL의 XSCO/XSCI의 unstable time + 내부 PLL 안정화 time)동안 Low로 출력된 후 25MHz clock의 1K divided clock(CLK\_25M\_DV1K)가 출력된다. |
| 19 | DPXn | OUM | **Link Duplex LED / SYS\_CLK\_DV4**  Link Duplex LED when MOD[3:0] != 0010 or MOD[3] = 0  Low : Full-Duplex  High : Half-Duplex  Probing SYS\_CLK\_DV4 when MODE[3:0] == 0010  System Operation Clock(SYS\_CLK) 의 4 Divided clock를 출력한다. |
| 20 | ACTn | OUM | **Link Activity LED / CLK\_SELn**  Link Activity LED when MOD[3:0] != 0010 or MOD[3] = 0  Low : Link up state without TX/RX  Flash : Link up state with TX/RX data  High : Link-down state  Probing CLK\_SELn when MOD[3:0] == 0010  System Operation Clock(SYS\_CLK)이 25MHz인 100MHz인지 출력한다.  Low : 100MHz  High : 25MHz |
| 21 | COLn | IOUM | **Link Collision Detect LED / Ext OSC / SYS\_CLK\_DV4**  Link Collision Detect LED when MOD[3:0] != “0010” or MOD[1] = ‘0’  W5100S의 Data 송신 시 충돌을 감지한 경우 알려준다.  Low : Collision Detected  High : No Collision  Ext. OSC  System Operation Clock(SYS\_CLK)를 Ext. OSC (100MHz@3.3V)를 통해 직접 공급한다.  When MOD[3:0] != 0010 and MOD[3:0] == 0X1X.  Probing SYS\_CLK\_DV4 when MOD[3:0] = 0010  System Operation Clock(SYS\_CLK)의 4 divided clock을 출력한다. |
| 22 | 1V2D | P | **Digital 1.2V Power**  1V2O로부터 공급 받는다. |
| 23 | GND | P | **Digital Ground** |
| 24 | 3V3D | P | **Digital 3.3V power** |
| 25 | MOD[0] | ID | **W5100S Mode Selection**  W5100S를 MOD[3:0]에 따라 아래와 같이 선택한다.  “0000” : SPI Mode  “0001” : W5500 Compatible SPI Mode  “0010” : Test Mode using W5500 Compatible SPI Mode  “0011” : W5500 Compatible SPI mode using external OSC(100MHz@3.3V)  “010X” : Parallel BUS Mode  “011X” : Parallel Bus Mode using external OSC (100MHz@3.3V)  “1XXX” : Factory Test Mode |
| 26 | MOD[1] | ID |
| 27 | MOD[2] | ID |
| 28 | MOD[3] | ID |
| 29 | CSn | IU | **W5100S Chip Select**  SPI(MOD[2]=0)나 Parallel BUS(MOD[3:0] != 0010 and MOD[2]=1) mode일 때, Chip Selection신호로 사용된다.  Low : Select  High : No Select |
| 30 | SCLK | ID | **SPI Clock**  SPI Mode(MOD[2]=0)일 때 Host로부터SPI Clock을 입력 받는다.  사용하지 않을 경우 GND와 연결하거나 Floating 한다. |
| 31 | 1V2D | P | **Digital 1.2V Power**  1V2O로부터 공급받는다. |
| 32 | MOSI /ADDR0 | IDM | **SPI Master Output Slave Input / Address 0**  SPI Master Output Slave Input when MOD[3:2] = 00  SPI mode일 때 Host로부터 SPI Data를 입력 받는다.  ADDR0 when MOD[3:0] != 0010 && MOD[2] = 1  Parallel Bus Mode일 때 Address 0을 입력 받는다. |
| 33 | MISO /ADDR1 | IOPM | **SPI Master Input Slave Output / Address 1**  SPI Master Input Slave Output when MOD[3:2] = 00  SPI mode일 때 Host로 SPI Data를 출력한다.  ADDR0 when MOD[3:0] != 0010 && MOD[2] = 1  Parallel Bus Mode일 때 Address 1을 입력 받는다. |
| 34 | RDn | IU | **Read Bus Control**  **For Parallel Bus Mode, this pin is the active LOW read strobe.**  **HOST reads W5100S Register/Memory selected by ADDR[1:0] through DAT[7:0].** |
| 35 | WRn | IU | **Write Bus Control**  **For Parallel Bus Mode, this pin is the active LOW write strobe.**  **HOST writes W5100S Register/Memory selected by ADDR[1:0] to DAT[7:0]. DAT[7:0] are latched in W5100S according to the configuration of the Write-data-fetch-timing.** |
| 36 | 3V3D | P | **Digital 3.3V Power** |
| 37 | DAT0 | IOU | **8 Bits Data Bus**  Parallel Bus Mode(MOD[3:0] != 0010 and MOD[2] = 1)일 때, Host의 Data를 입력하거나, W5100S의 Data를 출력한다.  W5100S Data Output when CSn = 0 and RDn = 0  Host Data Input when CSn = 0 and WRn = 0  Pulled-up when CS = 1 or Host Data Input |
| 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**  W5100S의 Ethernet packet 통신 처리(IP 충돌감지, WOL magic packet 수신, 각 통신 SOCKET별 데이터 송수신 등)를 위한 Event가 발생할 경우, Host에게 알려준다.  Low : Interrupt Occurred  High : No Interrupt  MR2(Mode Register 2)의 IEN(Interrupt pin Enable), IMR(Interrupt Mask Register), IMR2(Interrupt Mask Register 2), SLIMR(SOCKET-less Interrupt Mask Register) 참조. |
| 48 | RSTn | IP | **Reset**  W5100S을 초기화 시킨다. RSTn 신호는 반드시 500ns 이상 Low를 유지해야 한다. W5100S는 RSTn 신호가 인가 된 후 60.2ms 이후에 완전히 초기화가 된다.  Low : W5100S를 초기화시킨다.  High : W5100S를 정상 동작 시킨다. |

1. Memory Map

W5100S는 W5100에서 사용된 Common Register를 포함한 Common Register Block 과 SOCKET Register Block, TX, RX Memory Block들을 갖고 있다. 각 Block들은 Host Interface를 통해 선택된다.

0x0000~0x002F Register 는 W5100과 동일한 Common Register Block이고 0x0030~0x0088 Register는 W5100S에 새롭게 추가된 Common Register Block이다.

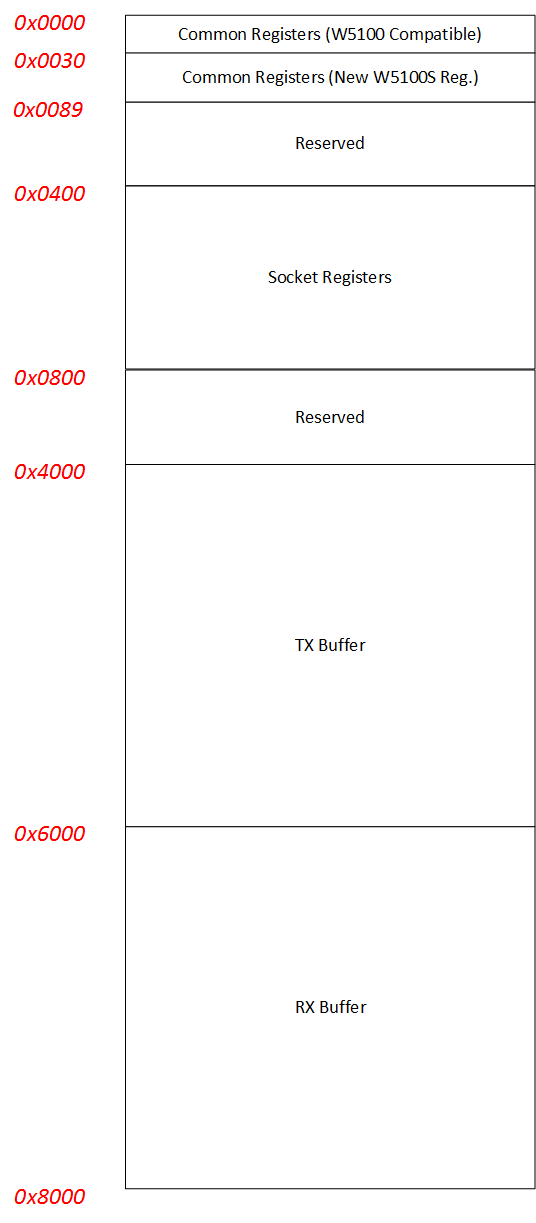


Figure 2 Memory Map

Figure 2는 Common & SOCKET Register Block과 SOCKET TX/RX Memory Block들의 사용 가능한 Offset Address 범위를 나타낸다. 별도의 설정이 없는 경우에 8KBytes의 TX/RX Memory Block은 4개의 SOCKET에 각각 2KBytes씩 할당되며 이는 Ethernet을 통해 송수신 되는 Data가 저장되는 공간이다.

초기에 할당된 2KBytes의 SOCKET TX/RX Memory Size는 SOCKET n TX/RX Buffer Size Register(Sn\_TXBUF\_SIZE/Sn\_RXBUF\_SIZE)를 사용해 변경할 수 있다.

그러나 SOCKET의 TX/RX Memory Size의 각각의 총합이 8KBytes를 초과할 경우 Data송수신에 오류가 발생하므로 주의를 요한다.

1. W5100S Registers
   1. 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 Low Level Timer  (INTLEVEL0)  (INTLEVEL1) |
| 0x0015 | Interrupt (IR) |
| 0x0016 | Interrupt Mask (IMR) |
| 0x0017  0x0018 | Retry Time-value  (RTR0)  (RTR0) |
| 0x0019 | Retry Count-value  (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  in PPPoE  (PHAR0)  (PHAR1)  (PHAR2)  (PHAR3)  (PHAR4)  (PHAR5) |
| 0x0038  0x0039 | Session ID in PPPoE  (PSIDR0)  (PSIDR1) |
| 0x003A  0x003B | Maximum Receive Unit in PPPoE  (PMRUR0)  (PMRUR1) |
| 0x003C  0x003D | PHY Status  (PHYSR0)  (PHYSR1) <- 내부용 |

|  |  |
| --- | --- |
| Address | Register |
| 0x003E | PHY Address Value (PHYAR) |
| 0x003F | PHY Register Address (PHYRR) |
| 0x0040  0x0041 | PHY Data Input  (PHYDIR0)  (PHYDIR1) |
| 0x0042  0x0043 | PHY Data Output  (PHYDOR0)  (PHYDOR1) |
| 0x0044 | PHY Action (PHYACR) |
| 0x0045 | PHY Division (PHYDIVR) |
| 0x0046  0x0047 | PHY Control  (PHYCR0)  (PHYCR1) |
| 0x0048  ~  0x004B | Reserved |
| 0x004C | SOCKET-less Control (SLCR) |
| 0x004D  0x004E | SOCKET-less Retry Time value  (SLRTR0)  (SLRTR1) |
| 0x004F | SOCKET-less Retry Count-value  (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 |
| 0x005E0x005E | SOCKET-less Interrupt Mask (SLIMR) |
| 0x005F | SOCKET-less Interrupt (SLIR) |
| 0x0060  0x0061  0x0062 | Debug Output [7:0]  [7:3] 은 Top에서 끊겨있음  DBG\_OUT0  DBG\_OUT1  DBG\_OUT2 |
| 0x0063 | NIC Collision Max-value  (NICMAXCOLR) |
| 0x0070 | Chip Lock (CHIPLCKR) |
| 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 |
|  |  |
| 0x0090 | FPGA TEST Address  (TEST\_ADDR) |
| 0x0091  0x0092  0x0093  0x0094 | FPGA TEST Value1 Address  (TEST\_VAL1\_ADDR0)  (TEST\_VAL1\_ADDR1)  (TEST\_VAL1\_ADDR2)  (TEST\_VAL1\_ADDR3) |
| 0x0095  0x0096 | FPGA TEST Value2 Address  (TEST\_VAL2\_ADDR0)  (TEST\_VAL2\_ADDR1) |
| 0x009E | FPGA TEST Checksum Error  (TEST\_CHCKSUM\_ERR) |
| 0x009F | FPGA TEST CRC Error  (TEST\_CRC\_ERR) |

* 1. SOCKET Registers

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Symbol | Description | Address | | | | |
| Sn\_ | S0\_ | S1\_ | S2\_ | S3\_ |
| Sn\_MR | SOCKET n  Mode | 0x0400+(0x0100xn) | 0x0400 | 0x0500 | 0x0600 | 0x0700 |
| Sn\_CR | SOCKET n  Command | 0x0401+(0x0100xn) | 0x0401 | 0x0501 | 0x0601 | 0x0701 |
| Sn\_IR | SOCKET n  Interrupt | 0x0402+(0x0100xn) | 0x0402 | 0x0502 | 0x0602 | 0x0702 |
| Sn\_SR | SOCKET n  Status | 0x0403+(0x0100xn) | 0x0403 | 0x0503 | 0x0603 | 0x0703 |
| Sn\_PORTR0  Sn\_PORTR1 | SOCKET n  Source Port | 0x0404+(0x0100xn) | 0x0404 | 0x0504 | 0x0604 | 0x0704 |
| 0x0405+(0x0100xn) | 0x0405 | 0x0505 | 0x0605 | 0x0705 |
| Sn\_DHAR0  Sn\_DHAR1  Sn\_DHAR2  Sn\_DHAR3  Sn\_DHAR4  Sn\_DHAR5 | SOCKET n  Destination Hardware Address | 0x0406+(0x0100xn) | 0x0406 | 0x0506 | 0x0606 | 0x0706 |
| 0x0407+(0x0100xn) | 0x0407 | 0x0507 | 0x0607 | 0x0707 |
| 0x0408+(0x0100xn) | 0x0408 | 0x0508 | 0x0608 | 0x0708 |
| 0x0409+(0x0100xn) | 0x0409 | 0x0509 | 0x0609 | 0x0709 |
| 0x040A+(0x0100xn) | 0x040A | 0x050A | 0x060A | 0x070A |
| 0x040B+(0x0100xn) | 0x040B | 0x050B | 0x060B | 0x070B |
| Sn\_DIPR0  Sn\_DIPR1  Sn\_DIPR2  Sn\_DIPR3 | SOCKET n  Destination IP  Address | 0x040C+(0x0100xn) | 0x040C | 0x050C | 0x060C | 0x070C |
| 0x040D+(0x0100xn) | 0x040D | 0x050D | 0x060D | 0x070D |
| 0x040E+(0x0100xn) | 0x040E | 0x050E | 0x060E | 0x070E |
| 0x040F+(0x0100xn) | 0x040F | 0x050F | 0x060F | 0x070F |
| Sn\_DPORTR0  Sn\_DPORTR0 | SOCKET n  Destination Port | 0x0410+(0x0100xn) | 0x0410 |  |  |  |
| 0x0411+(0x0100xn) | 0x0411 |  |  |  |
| Sn\_MSSR0  Sn\_MSSR1 | SOCKET n  TCP Maximum  Segment Size | 0x0412+(0x0100xn) | 0x0412 |  |  |  |
| 0x0413+(0x0100xn) | 0x0413 |  |  |  |
| Sn\_PROTOR | SOCKET n  Protocol  in IPRAW  mode | 0x0414+(0x0100xn) | 0x0414 |  |  |  |
| Sn\_TOS | SOCKET n TOS  in IP Header | 0x0415+(0x0100xn) | 0x0415 |  |  |  |
| Sn\_TTL | SOCKET n TTL  in IP Header | 0x0416+(0x0100xn) | 0x0416 |  |  |  |
| Reserved | Reserved | 0x0417+(0x0100xn) | 0x0417 |  |  |  |
| Reserved | Reserved | 0x041D+(0x0100xn) | 0x041D |  |  |  |
| Sn\_RX\_SIZE | SOCKET n  Receive  Buffer Size | 0x041E+(0x0100xn) | 0x041E |  |  |  |
| Sn\_TX\_SIZE | SOCKET n  Transmit  Buffer Size | 0x041F+(0x0100xn) | 0x041F |  |  |  |
| Sn\_TX\_FSR0  Sn\_TX\_FSR1 | SOCKET n  TX Free Size | 0x0420+(0x0100xn) | 0x0420 |  |  |  |
| 0x0421+(0x0100xn) | 0x0421 |  |  |  |
| Sn\_TX\_RD0  Sn\_TX\_RD1 | SOCKET n  TX Read  Pinter | 0x0422+(0x0100xn) | 0x0422 |  |  |  |
| 0x0423+(0x0100xn) | 0x0423 |  |  |  |
| Sn\_TX\_WR0  Sn\_TX\_WR1 | SOCKET n  TX Write  Pointer | 0x0424+(0x0100xn) | 0x0424 |  |  |  |
| 0x0425+(0x0100xn) | 0x0425 |  |  |  |
| Sn\_RX\_RSR0  Sn\_RX\_RSR1 | SOCKET n  RX Receive  Size | 0x0426+(0x0100xn) | 0x0426 |  |  |  |
| 0x0427+(0x0100xn) | 0x0427 |  |  |  |
| Sn\_RX\_RD0  Sn\_RX\_RD1 | SOCKET n  RX Read  Pinter | 0x0428+(0x0100xn) | 0x0428 |  |  |  |
| 0x0429+(0x0100xn) | 0x0429 |  |  |  |
| Sn\_RX\_WR0  Sn\_RX\_WR1 | SOCKET n  RX Write  Pointer | 0x042A+(0x0100xn) | 0x042A |  |  |  |
| 0x042B+(0x0100xn) | 0x042B |  |  |  |
| Sn\_IMR | SOCKET n  Interrupt Mask | 0x042C+(0x0100xn) | 0x042C |  |  |  |
| Sn\_FRAGR0  Sn\_FRAGR1 | SOCKET n  Fragment  Offset  in IP Header | 0x042D+(0x0100xn) | 0x042D |  |  |  |
| 0x042E+(0x0100xn) | 0x042E |  |  |  |
| Sn\_MR2 | SOCKET n  Mode 2 | 0x042F+(0x0100xn) | 0x042F |  |  |  |
| Sn\_KPALVTR | SOCKET n  Keep-alive Timer | 0x0430+(0x0100xn) | 0x0430 |  |  |  |
| Sn\_TSR | SOCKET n  Timer Status | 0x0431+(0x0100xn) | 0x0431 |  |  |  |
| Sn\_RTR0  Sn\_RTR1 | SOCKET n  Retry Time  value | 0x0432+(0x0100xn) | 0x0432 |  |  |  |
| 0x0433+(0x0100xn) | 0x0433 |  |  |  |
| Sn\_RCR | SOCKET n  Retry Count  value | 0x0434+(0x0100xn) | 0x0434 |  |  |  |

1. 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]은 Register Symbol[Upper Bit:Lower Bit]을 나타낸다.  Sn\_IR[3:0] = ‘0001’은 Sn\_IR[3]=’0’, Sn\_IR[2]=’0’, Sn\_IR[1]=’0’, Sn\_IR[0]=’1’을 나타낸다. |

**[Register Type]: Register의 Type을 나타낸다.**

* [RW] : Read와 Write 모두 가능한 Register
* [R=W] : Write한 값과 Read한 값이 같은 Register
* [RO] : Read Only Register
* [WO] : Write Only Register

**[Address Offset]: 레지스터의 Address Offset을 나타낸다.**

**[Reset Value]: 레지스터에 Reset 신호가 들어왔을 때 설정되는 값을 의미한다.**

**[Bit Type]: Bit의 Type을 나타낸다.**

* [RW] : Read와 Write 모두 가능한 Bit
* [R=W] : Write하면 그 값이 그대로 Read되는 Bit
* [R] : Read만 가능한 Bit
* [W] : Write만 가능한 Bit
* [WC] : Write ‘1’을 해야 Clear되는 Bit
* [AC] : Auto Clear되는 Bit

Ex)

4.1.1 MR ( Mode Register)

- [RW][0x0000][0x03]

MR의 Full Name은 Mode Register을 의미하며, Read와 Write가 모두 가능하고, Address Offset은 0x0000이며, Reset 신호가 들어왔을 때 설정되는 값은 0x03이다.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RST | - | - | PB | PPPoE | - | - | - |
| W | - | - | R=W | R=W | - | - | - |

* 1. Common Registers

* + 1. MR (Mode Register)
* [RW][0x0000] [0x03]

MR은 Reset, PING Block, PPPoE Mode Enable을 설정한다.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RST | - | - | PB | PPPoE | - | - | - |
| W | - | - | R=W | R=W | - | - | - |

|  |  |  |
| --- | --- | --- |
| Bit | Symbol | Description |
| 7 | RST | Reset  W5100S를 초기화한다.  Reset 후에 Host는 60.2ms이 후 W5100S를 Control 할 수 있다.  Write 1 : Reset  Write 0 : - |
| [6:5] | - | Reserved |
| 4 | PB | Ping Block  Ping Request Packet을 Block한다.  1 : Enable Ping request Block  0 : Disable Ping request Block |
| 3 | PPPoE | PPPoE Mode Enable  PPPoE Mode를 Enable한다.  1 : Enable PPPoE Mode  0 : Disable PPPoE Mode |
| [2:0] | - | Reserved |

* + 1. GWR (Gateway IP Address Register)
* [R=W] [0x0001-0x0004] [0x00]

GWR은 Gateway의 IP Address를 설정한다.

Ex) GWR = “192.168.0.1”,

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

* + 1. SUBR (Subnet Mask Register)
* [R=W] [0x0005–0x0008] [0x00]

SUBR은 Subnet Mask 범위를 설정한다.

Ex) SUBR = “255.255.255.255”,

|  |  |  |  |
| --- | --- | --- | --- |
| SUBR0(0x0005) | SUBR0(0x0006) | SUBR0(0x0007) | SUBR0(0x0008) |
| 255 (0xFF) | 255 (0xFF) | 255 (0xFF) | 255 (0xFF) |

* + 1. SHAR (Source Hardware Address Register)
* [R=W] [0x0009-0x000E] [0x00]

SHAR은 MAC Address를 설정한다.

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 |

* + 1. SIPR (Source IP Address Register)
* [R=W] [0x000F-0x0012] [0x00]

SIPR은IP Address를 설정한다.

Ex) SIPR = “192.168.0.100”,

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

* + 1. IR (Interrupt Register)
* [RW] [0x0015] [0x00]

IR은 W5100S의 Event나 SOCKET n의 Event가 발생했을 때 해당 Event bit가 1로 설정된다.

IR의 Event가 발생하고 IMR의 1:1 대응되는 Bit가 설정되어 있을 경우, INTn 핀이 Low상태가 된다.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 Interrupt  Read 1 : IP Conflict 감지  Read 0 : - |
| 6 | UNREACH | Destination Port Unreachable Interrupt  수신한 Unreachable Packer의 Unreachable IP Address와 Port Number는 각각 UIPR (Unreachable IP Register) 과 UPORTR (Unreachable Port Register)에 저장된다.  Read 1 : Unreachable Packet 수신  Read 0 : - |
| 5 | PPPTERM | PADT/LCPT Interrupt  Read 1 : PPPT 또는 LCPT Packet 수신  Read 0 : - |
| 4 | - | Reserved |
| [3:0] | Sn\_INT | SOCKET n Interrupt (n : 3 ~ 0)  Read 1 : Sn\_IR(SOCKET n Interrupt Register)가 0이 아닌 경우  Read 0 : - |

* + 1. IMR (Interrupt Mask Register)
* [R=W] [0x0016] [0x00]

IMR은 **IR**에 1:1 대응되는 Event 발생 시 INTn 핀이 Low상태가 되도록 설정한다.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 | IM\_IR7 | IP Conflict Interrupt Mask  1 : Enable IP Conflict Interrupt  0 : Disable IP Conflict Interrupt |
| 6 | IM\_IR6 | Destination Port Unreachable Interrupt Mask  1 : Enable Destination Port Unreachable Interrupt  0 : Disable Destination Port Unreachable Interrupt |
| 5 | IM\_IR5 | PADT/LCPT Interrupt Mask  1 : Enable PADT/LCPT Interrupt  0 : Disable PADT/LCPT Interrupt |
| 4 | - | Reserved |
| 3 | IM\_IR3 | SOCKET 3 Interrupt Mask  1 : Enable SOCKET 3 Interrupt  0 : Disable SOCKET 3 Interrupt |
| 2 | IM\_IR2 | SOCKET 2 Interrupt Mask  1 : Enable SOCKET 2 Interrupt  0 : Disable SOCKET 2 Interrupt |
| 1 | IM\_IR1 | SOCKET 1 Interrupt Mask  1 : Enable SOCKET 1 Interrupt  0 : Disable SOCKET 1 Interrupt |
| 0 | IM\_IR0 | SOCKET 0 Interrupt Mask  1 : Enable SOCKET 0 Interrupt  0 : Disable SOCKET 0 Interrupt |

* + 1. RTR (Retransmission Timer value Register)
* [R=W] [0x0017-0x0018] [0x07D0]

RTR은 Sn\_RTR(SOCKET n Retransmission Timer value Register)의 초기값을 설정한다.

단위는 100us이다.

SOCKET 재전송 관련내용은 여기를 참조하면 된다.

Ex) RTR = 5000 (0x1388),

5000\*100us = 0.5s

|  |  |
| --- | --- |
| RTR0(0x0017) | RTR1(0x0018) |
| 0x013 | 0x88 |

* + 1. RCR (Retransmission Counter-value Register)
* [R=W] [0x0019] [0x08]

RCR은 Sn\_RCR(SOCKET n Retransmission Counter value Register)의 초기값을 설정한다.

SOCKET 재전송 관련내용은 여기를 참조하면 된다.

ARP와 PING Request Packet에 대한 재전송의 경우는 RTR(Retry Timer-value Register)에 의해 설정된 시간마다 재전송 Packet을 전송하고 RCR(Retry Count-value Register) 값을 초과하여 재전송을 시도한 경우에도 상대방의 응답이 없는 경우 Sn\_IR[TIMEOUT] = ‘1’이 된다. ARP와 PING의 경우 Sn\_IR[TIMEOUT] = ‘1’이 될 때까지의 재전송의 총 시간()은 다음과 같이 계산할 수 있다.

|  |
| --- |
| = (RTR x 0.1ms) x (RCR + 1) |

TCP의 경우 SYN, FIN, DATA Packet을 전송했을 때 상대방의 ACKPacket을 받아야지만 Sn\_IR[SENDOK]=’1’이 되고 전송이 완료가 된다. 때문에 상대방으로 ACKPacket을 받지 못한 경우 전송을 완료하기 위해서 재전송 과정을 수행하는데 이때 RTR(Retry Timer-value Register), RCR(Retry Counter-value Register)의 설정에 의해 재전송 시간과 전송 횟수가 결정된다. TCP의 경우 재전송 횟수가 증가할 수록 설정된 RTR값은 지수배 증가하게 된다. 그리고 설정된 재전송 횟수를 초과하게 되면 Sn\_IR[TIMEOUT] = ‘1’이 된다.

TCP에서 Sn\_IR[TIMEOUT] = ‘1’이 될 때의 최종 재전송 시간은 다음과 같이 계산할 수 있다.

|  |
| --- |
| N : 재전송 횟수, 0  M : RTR 65535 and 0 에서의 최소값  : RTR |

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

= 2000 X 0.1ms X 9 = 1.8s

= (0x07D0+0x0FA0+0x1F40+0x3E80+0x7D00+0xFA00+0xFA00+0xFA00+0xFA00) X 0.1ms

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

= 318000 X 0.1ms = 31.8s

* + 1. RMSR (RX Memory Size Register)
* [R=W] [0x001A] [0x55]

RMSR은 각 SOCKET의 RX Buffer Size를 설정한다.

SOCKE RX Buffer Size의 총합은 8KB를 넘을 수 없다.

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

각각의 SOCKET은 S0와 S1의 값에 의해서 아래와 같이 Memory가 할당된다.

|  |  |  |
| --- | --- | --- |
| Memory Size | S1 | S0 |
| 1 KB | 0 | 0 |
| 2 KB | 0 | 1 |
| 4 KB | 1 | 0 |
| 8 KB | 1 | 1 |

* + 1. TMSR (TX Memory Size Register)
* [R=W] [0x001B] [0x55]

TMSR은 각 SOCKET의 TX Buffer Size를 설정한다.

SOCKET TX Buffer Size의 총합은 8KB를 넘을 수 없다.

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

각각의 SOCKET은 S0와 S1의 값에 의해서 아래와 같이 Memory가 할당된다.

|  |  |  |
| --- | --- | --- |
| Memory Size | S1 | S0 |
| 1 KB | 0 | 0 |
| 2 KB | 0 | 1 |
| 4 KB | 1 | 0 |
| 8 KB | 1 | 1 |

* + 1. IR2 (Interrupt Register 2)
* [RW] [0x0020] [0x00]

IR2는 WOL이 발생했을 때 설정된다.

WOL이 발생하고, IMR2의 WOL이 설정되어 있을 경우, INTn 핀이 Low 상태가 된다.

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

|  |  |  |
| --- | --- | --- |
| Bit | Symbol | Description |
| [7:1] | - | Reserved |
| 0 | WOL | WOL MAGIC PACK Interrupt  Read 1 : UDP 기반 WOL MAGIC Packet 수신  Read 0 : - |

* + 1. IMR2 (Interrupt Mask Register 2)
* [R=W] [0x0021] [0x00]

IMR2는 IR2에 Event 발생 시 INTn 핀이 Low 상태가 되도록 설정한다.

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

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

* + 1. PTIMER (PPP Link Control Protocol Request Timer Register)
* [R=W] [0x0028] [0x28]

PTIMER은 LCP echo request를 보내는 주기를 설정한다.

단위는 25ms이다.

Ex) PTIMER = 200 (0xC8),

200 \* 25ms = 5s

* + 1. PMAGIC (PPP Link Control Protocol Magic number Register)
* [R=W] [0x0029] [0x00]

PMAGIC은 LCP negotiation Packet에 사용될 4byte Magic number의 값을 설정한다.

**Ex)** PMAGIC = 0x01,

|  |
| --- |
| PMAGIC(0x0029) |
| 0x01 |

LCP Magic number = 0x01010101

* + 1. UIPR (Unreachable IP Address Register)
* [RO] [0x002A-0x002D] [0x0000]

UIPR은 Unreachable Packet을 수신 했을 때(IR[UNR] = ‘1’), 수신한 Packet의 IP Address가 저장된다.

Ex) UIPR = “192.169.0.21”,

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

* + 1. UPORTR (Unreachable Port Register)
* [RO] [0x002E-0x002F] [0x0000]

UPORTR은 Unreachable Packet을 수신 했을 때(IR[UNR] = ‘1’), 수신한 Packet의 Port가 저장된다.

Ex) UPORTR = 3000 (0x0BB8),

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

* + 1. MR2 (Mode Register 2)
* [R=W] [0x0030] [0x40]

MR2는 W5100S 동작 Clock 선택, INTn 핀 활성화, TCP Reset Packet Send, UDP Unreachable port Send, WOL Packet Recv, PSH Flag 활성화, UDP ARP Packet Send를 설정 한다.

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

|  |  |  |
| --- | --- | --- |
| Bit | Symbol | Description |
| 7 | CLKSEL | System Clock Select  TCP/IP Core 동작 Clock을 선택할 수 있다.  CLKLCKR(Clock Lock Register)를 Unlock 해야 해당 bit를 설정할 수 있다. (참조 4.1.26 CHIPCFGR(CHIP Configuration Register) [WO] [0x0070] [0x00])  1 : 25MHz  0 : 100MHz |
| 6 | IEN | Interrupt pin(INTn) Enable  INTn을 Enable또는 Disable할 수 있다.  1 : INTn Enable  0 : INTn Disable (INTn is Always High) |
| 5 | NOTCPRST | No TCP Reset Packet Send  TCP통신에서 상대방이 잘못된 Port로 Packet을 전송했을 때 TCP SOCKET이 Reset Packet전송을 하지 않도록 설정할 수 있다. Port scan 공격을 막을 수 있다.  1 : TCP Reset Packet Send Block  0 : TCP Reset Packet Send |
| 4 | UDPURB | UDP Unreachable Packet Send Block  UDP통신에서 상대방이 잘못된 Port로 Packet을 전송했을 때 UDP SOCKET이 Destination Port Unreachable Packet전송을 하지 않도록 설정할 수 있다. Port scan 공격을 막을 수 있다.  1 : Destination Port Unreachable Packet Send Block  0 : Destination Port Unreachable Packet Send |
| 3 | WOL | Wake On LAN  WOL Packet을 수신할지 결정하는 bit  1 : Receive WOL Packet  0 : No Receive WOL Packet |
| 2 | NOCHK | MACRAW No Size Check  SOCKET이 MACRAW Mode일때, 상대방이 보내온 Packet의 Size가 MAX Packet Size보다 커도 Size를 확인하지 않고 수신할 수 있도록 한다.  1 : No Check Packet Size  0 : Check Packet Size |
| 1 | FARP | UDP Force ARP  UDP와 IPRAW Mode에서 Send Command 마다 ARP를 수행한다.  01 : UDP Force ARP Enable  01 : UDP Force ARP Disable |
| 0 | SSHA | Skip SRC Hardware Address  상대방의 Hardware Address를 확인하지 않고 수신한다. 따라서 상대방이 W5100S와 동일한 Hardware Address를 가지고 있어도 송수신이 가능하다.  1 : 상대방의 Hardware Address를 확인하지 않는다.  0 : 상대방의 Hardware Address를 확인한다. |

* + 1. PHAR (Destination Hardware Address Register in PPPoE Mode)
* [R=W] [0x0032-0x0037] [0x0000]

PHAR은 PPPoE Mode일 때만 유효하다.

PPPoE Destination Hardware Address를 설정한다.

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 |

* + 1. PSIDR (Session ID Register in PPPoE Mode)
* [R=W] [0x0038-0x0039] [0x0000]

PSIDR은 PPPoE Mode일 때만 유효하다.

PPPoE Session ID를 설정한다.

Ex) PSIDR = 0x1234,

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

* + 1. PMRUR (PPPoE Maximum Receive Unit Register)
* [R=W] [0x003A-0x003B] [0xFFFF]

PMRUR은 PPPoE Mode일 때만 유효하다.

PMRUR은 PPPoE Mode에서 MRU(Maximum Receive Unit)를 설정하며, 1472 보다 큰 값을 설정하면 1472로 설정된다.

PMRUR은 SOCKET 생성(Sn\_CR[OPEN] = ‘1’) 전에 설정 해야 한다.

Ex) PMUR = 1000 (0x03E8),

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

* + 1. PHY Access Registers

Internal PHY의 SPEED, DPX, LINK 등의 상태를 확인하거나 Configuration을 할 수 있다.

MDC/MDIO Interface를 통해 직접 Internal PHY의 Register에 Access하여 Configuration하려면 PHYRAR, PHYDIR, PHYDOR, PHYACR을 사용하면 된다.

PHYSR (PHY Status Register)

* [RO] [0x003C] []

PHYSR은 PHYCR0(PHY Control Register 0)을 통해 설정된 PHY Operation Mode 및 LINK 상태를 확인 할 수 있다.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

PHYRAR (PHY Register Address Register)

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

PHYRAR은 MDC/MDIO Interface의 PHY Register Address를 설정한다.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 | - | Reserved |
| 6 | - | Reserved |
| 5 | - | Reserved |
| [4:0] | A[4:0] | PHY Register Address |

PHYDIR (PHY Data Input Register)

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

PHYDIR은MDC/MDIO Interface의 DATA를 설정한다.

Ex) PHYDIR = 0x1234,

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

PHYDOR (PHY Data Output Register)

* [R=W] [0x0042-0x0043] [0x0000]

PHYDOR은MDC/MDIO Interface의 DATA가 저장된다.

Ex) PHYDOR = 0x1234,

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

PHYACR (PHY Access Control Register)

* [R=W] [0x0044] [0x00]

PHYACR은MDC/MDIO Interface의 Access Type을 설정한다.

Read/Write Access가 성공적으로 끝나게 되면 PHYACR은 자동적으로 0x00으로 Clear된다.

|  |  |
| --- | --- |
| Access Type | Value |
| Write | 0x01 |
| Read | 0x02 |

PHYDIVR (PHY Division Register)

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

PHYDIVR(PHY Division Register)는 PHY의 Clock Division Register이다.

|  |  |
| --- | --- |
| Value | Division |
| 0x00 | 1/32 |
| 0x10 | 1/64 |
| others | 1/128 |

PHYCR0 (PHY Control Register 0)

* [WO] [0x0046] [0x00]

PHYCR0는 PHY의 Operation Mode를 설정한다. PHYCR0를 사용하기 위해서는 먼저 PHYLCKR(PHY Lock Register)를 Unlock으로 설정해야 한다. PHYCR0를 통해 설정된 bit들은 PHYSR [5:3]을 통해 알 수 있다.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | - | - | - | - | AUTO | SPD | DPX |
|  |  |  |  |  | WO | WO | WO |

|  |  |  |
| --- | --- | --- |
| Bit | Symbol | Description |
| [7:3] | - | Reserved |
| 2 | AUTO | Auto Negotiation 을 결정하는 bit  1 : Disable Auto Negotiation  0 : Enable Auto Negotiation  AUTO가 1로 설정되어 있을 경우 SPD와 DPX는 무시된다. |
| 1 | SPD | 10/100 Mbps 를 결정하는 bit  1 : 10 Mbps  0 : 100 Mbps |
| 0 | DPX | Half/Full Duplex 를 결정하는 bit  1 : HDX  0 : FDX |

PHYCR1 (PHY Control Register 1)

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

PHYCR1는 PHY의 Power Down, Reset을 설정한다. PHYCR1를 사용하기 위해서는 먼저 PHYLCKR(PHY Lock Register)를 Unlock으로 설정해야 한다.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | - | PWDN | - | - | - | - | Reset |
| - | - | R=W | - | - | - | - | R=W |

|  |  |  |
| --- | --- | --- |
| Bit | Symbol | Description |
| [7:6] | - | Reserved |
| 5 | PWDN | PHY Power Down Bit  1 : Enable Power Down mode  0 : Disable Power Down mode |
| [4:1] | - | Reserved |
| 0 | RST | PHY Reset Bit  1 : Normal  0 : PHY Reset  최소 PHY Reset Valid time(400us) 이후 1로 설정해야 한다. |

* + 1. SOCKET-less Registers

SOCKET-less Register는 SOCKET 없이 ARP/PING Request Packet을 전송하는 기능을 제공한다.

SLCR (SOCKET-less Command Register)

* [RW] [0x004C] [0x00]

SLCR은 ARP 또는 PING Request 전송 Command를 설정한다. 각 Command는 동시에 수행 될 수 없다.

Command는 수행 완료 후 Auto Clear 되며, Auto Clear전에는 다른 Command를 설정할 수 없다. Command 결과는 SLIR(SOCKET-less Interrupt Register)를 통해 알 수 있다.

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

|  |  |  |
| --- | --- | --- |
| Bit | Symbol | Description |
| [7:2] | - | Reserved |
| 1 | ARP | Write 1: ARP Packet을 전송한다.  Read 1 : Command 수행 중 |
| 0 | PING | Write 1: PING Packet을 전송한다.  Read 1 : Command 수행 중 |

SLRTR (SOCKET-less Retransmission Timer-value Register)

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

SLRTR은 SLCR의 재전송 시간을 설정한다. 단위는 100us이다. SLCR에 의해 전송된 ARP 또는 PING Request Packet에 대한 응답이 없는 경우 자동으로 Request Packet을 재전송한다.

자세한 내용은 5.6 Request Command를 참조

Ex) SLRTR = 5000 (0x1388),

5000 \* 100us = 0.5s

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

SLRCR (SOCKET-less Retransmission Count-value Register)

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

SLRCR은 SLCR의 재전송 횟수를 설정한다. 재전송 횟수가 SLRCR을 초과하면 SLIR[TIMEOUT] = ‘1’이 된다.

자세한 내용은 5.6 Request Command를 참조

SLPIPR (SOCKET-less Peer IP Address Register)

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

SLPIPR은 SLCR에 의한ARP 또는 PING Request Packet을 전송할 Peer IP Address를 설정한다.

Ex) SLPIPR = “192.169.0.21”,

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

SLPHAR (SOCKET-less Peer Hardware Address Register)

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

SLPHAR은 SLCR에 의한 ARP Reply Packet의 Peer Hardware Address가 설정된다.

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 |

PINGSEQR (PING Sequence-number Register)

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

PINGSEQR은 PING Request Packet의 PING Sequence Number를 설정한다.

Ex) PINGSEQR = 1000 (0x03E8),

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

PINGIDR (PING ID Register)

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

PINGIDR은 PING Request Packet의 PING ID를 설정한다.

Ex) PINGIDR = 256 (0x0100),

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

SLIMR (SOCKET-less Interrupt Mask Register)

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

SLIMR은 SLIR(SOCKET-less Interrupt Register)의 1:1 대응되는 Event 발생 시 INTn 핀이 Low 상태가 되도록 설정한다.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

SLIR (SOCKET-less Command Interrupt Register)

* [RW] [0x005F] [0x00]

SLIR은 SLCR로 인한 Event가 발생했을 때 해당 Event bit가 ‘1’로 설정된다.

SLIR의 Event가 발생하고, SLIMR의 1:1 대응되는 Bit가 설정되어 있을 경우, INTn 핀이 Low 상태가 된다.

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

|  |  |  |
| --- | --- | --- |
| Bit | Symbol | Description |
| [7:3] | - | Reserved |
| 2 | TIMEOUT | TIMEOUT 발생 시, ‘1’로 설정된다. |
| 1 | ARP | ARP Reply 수신 시, ‘1’로 설정된다. |
| 0 | PING | PING Reply 수신 시, ‘1’로 설정된다. |

* + 1. CLKLCKR (Clock Lock Register)
* [WO] [0x0070] [0x00]

CLKLCKR은 MR2(CLKSEL)의 설정을 Lock/Unlock한다. 초기 상태는 Lock이다.

MR2(CLKSEL)의 설정 이 후에도 Lock 상태는 변하지 않는다.

|  |  |
| --- | --- |
| Unlock | 0xCE |
| Lock | others |

* + 1. NETLCKR (Network Lock Register)
* [WO] [0x0071] [0x00]

NETLCKR은 W5100S Network Configuration에 관련된 GWR, SUBR, SHAR, SIPR의 설정을 Lock/Unlock한다. 초기 상태는 Unlock이다.

GWR, SUBR, SHAR, SIPR의 설정 이 후에도 Lock 상태는 변하지 않는다.

|  |  |
| --- | --- |
| Unlock | 0xC5 |
| Lock | 0x3A |

* + 1. PHYLCKR (PHY Lock Register)
* [WO] [0x0072] [0x00]

PHYLCKR은 PHYCR0, PHYCR1의 설정을 Lock/Unlock한다. 초기 상태는 Lock이다.

PHYCR0, PHYCR1의 설정 이후에도 Lock 상태는 변하지 않는다.

|  |  |
| --- | --- |
| Unlock | 0x53 |
| Lock | Others |

* + 1. VERR (W5100S Version Register)
* [RO] [0x0080] [0x51]

VERR는 W5100S Version이다.

* + 1. TCNTR (Ticker Counter Register)
* [RO][0x0082-0x0083][0x0000]

TCNTR은 W5100S 내부 Clock에 따라 100us마다 1씩 증가하는 Register이다.

* 1. SOCKET Register
     1. Sn\_MR (SOCKET n Mode Register)
* [R=W] [0x0000+0x0100\*(n+4)] [0x00]

Sn\_MR은 통신 Protocol과 No Delayed ACK, IGMP Report Version, MAC Filter, Multicast등과 같은 SOCKET Option을 설정 한다.

Sn\_MR은 SOCKET 생성(Sn\_CR[OPEN] = ‘1’) 전에 설정 해야 한다.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 | Multicast in UDP  UDP Mode에서 Multicast를 설정한다. 5.2.3 UDP MULTICAST 참조.  1 : Enable UDP Multicast  0 : Disable UDP Multicast |
| 6 | MF | MAC Filter Enable in MACRAW  MACRAW Mode(Sn\_MR[3:0] = ‘MACRAW’)에서 W5100S로 들어오는 Packet을 전부 수신하게 된다. Filter Enable 시 W5100S의 MAC 주소(Multicast, Broadcast 포함)로 보내온 Packet만을 수신 한다.  1 : Enable MAC Filter  0 : Disable MAC Filter |
| 5 | ND / MC | ND(No Delayed Ack)  TCP Mode에서 Packet을 수신하자마자 자동으로 ACK Packet이 전송된다.  1 : Enable No Delayed ACK  0 : Disable No Delayed ACK  MC(Multicast IGMP Version)  UDP Mode이고 Multicast일 때(Sn\_MR[3:0] = ‘UDP’ & Sn\_MR[MULTI] = ‘1’), IGMP Version을 선택한다.(version 3은 지원하지 않는다.)  0 : Using IGMP version 2  1 : Using IGMP version 1 |
| 4 | - | Reserved |
| [3:0] | P[3:0] | Protocol  SOCKET의 Protocol을 설정한다.  MACRAW Mode는 SOCKET 0에서만 사용할 수 있다.     |  |  |  |  |  | | --- | --- | --- | --- | --- | | P3 | P2 | P1 | P0 | Protocol Mode ~~Meaning~~ | | 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 | |

* + 1. Sn\_CR (SOCKET n Command Register)
* [RW] [0x0001+0x0100\*(n+4)] [0x00]

Sn\_CR은 SOCKET Command를 설정한다. Command는 수행 완료 후 Auto Clear 되며, Auto Clear전에는 다른 Command를 설정할 수 없다.

|  |  |  |
| --- | --- | --- |
| Value | Symbol | Description |
| 0x01 | OPEN | SOCKET을 OPEN한다.  Sn\_MR에 따른 Command 수행 결과는 Sn\_SR에서 확인된다.     |  |  | | --- | --- | | 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) |   . |
| 0x02 | LISTEN | TCP OPEN(Sn\_SR = ‘0x13’)상태에서, 상대방의 접속을 기다린다.  5.1.3 TCP SOCKET Listen 참조. |
| 0x04 | CONNECT | TCP OPEN(Sn\_SR = ‘0x13’)상태에서, 상대방에게 접속을 요청한다.  5.1.4 TCP SOCKET Connect 참조. |
| 0x08 | DISCON | SOCK\_ESTABLESHED(Sn\_SR = ‘0x17’) 또는 SOCK\_CLOSE\_WAIT(Sn\_SR = ‘0x1C’)상태에서 상대방에게 접속 해제를 요청한다.  즉, 접속중인 상대방에게 Disconnect-request(FIN Packet)를 전송한다.  5.1.9 TCP FIN Send, 5.1.10 TCP FIN Receive 참조. |
| 0x10 | CLOSE | SOCKET을 CLOSE한다.  Sn\_SR(SOCKET n Status Register)은 SOCK\_CLOSED로 변경된다.  (주의) TCP인 경우, FIN Packet 전송 없이 강제 CLOSE 된다. |
| 0x20 | SEND | 설정된 Size의 Data를 전송한다.  Size는 Sn\_TX\_WR(SOCKET n TX Write Point Register)을 통해 설정 가능하다.  SEND에 대한 처리가 완료되었을 경우 Sn\_IR [SENDOK]=‘1’로 된다. Host는 Sn\_IR[SENDOK]=‘1’를 확인 후 그 다음 Data에 대한 SEND를 수행할 수 있다.  TCP나 UDP mode에서, 설정된 Size의 Data가 MSS(Maximum Segment Size)보다 클 경우 자동으로 MSS 단위로 나누어 전송한다.  IPRAW 나 MACRAW Mode에서, 설정된 Size의 Data가 MSS보다 클 경우 Host는 MSS 단위로 직접 나누어 전송해야 한다.  TCP mode에서, Data를 상대방에게 성공적으로 전송한 경우(상대방으로부터 ACK를 수신한 경우) Sn\_TX\_FSR(SOCKET n TX Free Size Register)는 전송한 Data Size만큼 증가한다.  그렇지 못한 경우(상대방으로부터 ACK를 수신하지 못한 경우)는 Sn\_IR[TIMEOUT] = ‘1’이 되고 Sn\_SR(SOCKET n STATUS Register)은 SOCK\_CLOSED로 변경된다.  UDP, IPRAW, MACRAW mode에서, Sn\_IR[SENDOK] = ‘1’ 이 후에 Sn\_TX\_FSR은 전송한 Data Size만큼 증가한다. |
| 0x21 | SEND\_MAC | SEND\_MAC은 UDP mode/IPRAW mode일 때만 사용된다.  기본동작은 SEND와 같다. SEND는 자동으로 ARP-process를 통해 Destination hardware address를 얻은 후 Data를 전송하는 반면, SEND\_MAC은 Host가 설정한 Sn\_DHAR(SOCKET n Destination Hardware Address Register)을 Destination hardware address로 하여 Data를 전송한다. SEND\_MAC은 Hardware address를 이미 알고 있는 Destination으로 UDP나 IPRAW data를 전송할 때 불필요한 ARP-process를 없애 Network traffic을 감소시킬 수 있다. |
| 0x22 | SEND\_KEEP | SEND\_KEEP은 TCP mode일 때만 사용된다.  Keep alive Packet을 송신하여 Connection이 유효한지 확인한다. 만약 상대방이 응답이 없는 경우 Connection이 끊어지고, Timeout Interrupt가 발생한다.  5.1.8 TCP KEEPALIVE 참조. |
| 0x23 | JOIN | JOIN\_IGMP는 UDP mode에서 Multicasting 기능을 이용할 때만 사용된다.  IGMP Report 메시지를 전송하여 Multicasting 그룹에 가입한다.  자세한 설명은 5.2.3 UDP MULTICAST 를 참고하라. |
| 0x24 | LEAVE | LEAVE\_IGMP는 UDP mode에서 Multicasting 기능을 이용할 때만 사용된다.  IGMP Leave 메시지를 전송하여 Multicasting 그룹에서 탈퇴한다.  자세한 설명은 5.2.3 UDP MULTICAST 를 참고하라. |
| 0x40 | RECV | Host가 SOCKET n이 수신한 DATA를 READ했음을 알린다.  Sn\_RX\_RSR(SOCKET n RX Received Size Register), Sn\_RX\_WR(SOCKET n RX Write Pointer Register), Sn\_RX\_RD (SOCKET n RX Read Pointer Register) 참조. |

* + 1. Sn\_IR (SOCKET n Interrupt Register)
* [RW] [0x0002+0x0100\*(n+4)] [0x00]

Sn\_IR은 SOCKET의 상태변화나 Command 수행 결과를 알려준다.

Sn\_IR의 Event가 발생하고, Sn\_IMR의 1:1 대응되는 Bit가 설정되어 있을 경우, IR[Sn\_INT]=’1’로 설정된다.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 : Sn\_CR[SEND]을 완료했을 경우  0 : when others |
| 3 | TIMEOUT | Timeout occurred  1 : ARP 또는 TCP 통신과정에서 Sn\_RCR(SOCKET Retransmission Count Register)이상의 재전송이 발생한 경우  0 : when others |
| 2 | RECV | Received  1 : SOCKET n RX Memory에 Data를 수신하거나 Sn\_CR[RECV] 수행 후 Data가 남아있는 경우  0 : when others |
| 1 | DISCON | Disconnected  1 : FIN Packet을 수신하거나 전송한 FIN에 대한 ACK Packet을 수신한 경우 마지막으로, 상대방으로부터 RST Packet을 수신한 경우  0 : when others |
| 0 | CON | Connected  1 : TCP mode에서 접속이 완료된 경우  0 : when others |

* + 1. Sn\_SR (SOCKET n Status Register)
* [RO] [0x0003+0x0100\*(n+4)] [0x00]

Sn\_SR은 SOCKET n의 상태를 알려준다. Sn\_SR는 Sn\_CR의 Command수행 또는 Data 송수신중에 변경될 수 있다.

|  |  |  |
| --- | --- | --- |
| Value | Symbol | Description |
| 0x00 | SOCK\_CLOSED | SOCKET이 Close된 상태. |
| 0x13 | SOCK\_INIT | SOCKET이 TCP mode로 Open된 상태. |
| 0x14 | SOCK\_LISTEN | SOCKET이 TCP mode이고 상대방의 접속을 기다리는 상태. |
| 0x17 | SOCK\_ESTABLISHED | SOCKET이 TCP mode이고 상대방과 접속된 상태. |
| 0x1C | SOCK\_CLOSE\_WAIT | SOCKET이 TCP mode이고 접속해제 요청을 수신한 상태. |
| 0x22 | SOCK\_UDP | SOCKET이 UDP mode로 Open된 상태. |
| 0x32 | SOCK\_IPRAW | SOCKET이 IPRAW Mode로 Open된 상태. |
| 0x42 | SOCK\_MACRAW | SOCKET이 MACRAW Mode로 Open된 상태. |

아래 SOCKET status은 Sn\_SR의 전이 과정에서 관찰될 수 있는 temporary Status들이다.

|  |  |  |
| --- | --- | --- |
| Value | Symbol | Description |
| 0x15 | SOCK\_SYNSENT | Connect-request를 전송한 상태. |
| 0x16 | SOCK\_SYNRECV | Connect-request를 수신한 상태. |
| 0x18 | SOCK\_FIN\_WAIT | SOCKET이 Closing되는 상태. |
| 0X1B | SOCK\_TIME\_WAIT |
| 0X1D | SOCK\_LAST\_ACK |

D:\___WORK\2017_08\status diagram_2017_08_29.emf

Figure . State Diagram

* + 1. Sn\_PORTR (SOCKET n Source Port Register)
* [R=W] [0x0004+0x0100\*(n+4), 0x0005+0x0100\*(n+4)] [0x0000]

Sn\_PORTR은 SOCKET n의 Source Port Number를 설정한다.

Ex) S0\_PORTR = 5000 (0x1388),

|  |  |
| --- | --- |
| S0\_PORTR0(0x0404) | S0\_PORTR1(0x0405) |
| 0x013 | 0x88 |

* + 1. Sn\_DHAR (SOCKET n Destination Hardware Address Register)
* [R=W] [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]

Sn\_DHAR는 상대방의 MAC Address이다.

TCP인 경우, 상대방과 접속되었을 때(Sn\_SR = ‘SOCK\_ESTABLISHED’) 상대방의 MAC Address가 설정된다.

UDP, IPRAW인 경우, Sn\_CR[SEND\_MAC] 사용시 상대방의 MAC Address를 설정한다.

UDP Multicast인 경우, Multicast Group의 MAC Address를 설정한다. (Multicast Application Note 참조)

**Ex)** Sn\_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 |

* + 1. Sn\_DIPR (SOCKET n Destination IP Address Register)
* [R=W] [0x000C+0x0100\*(n+4), 0x000D+0x0100\*(n+4), 0x000E+0x0100\*(n+4), 0x000F+0x0100\*(n+4)] [0x00000000]

Sn\_DIPR은 상대방의 IP Address이다.

TCP인 경우, 접속할 상대방의 IP Address를 설정하거나, 접속된 상대방의 IP Address를 확인한다.

UDP, IPRAW인 경우, 전송할 상대방의 IP Address를 설정한다.

UDP Multicast인 경우, Multicast Group의 IP Address를 설정한다.(Multicast Application Note 참조)

UDP, UDP Multicast, IPRAW인 경우, 수신된 상대방의 IP Address는 SOCKET n RX Buffer에서 확인한다.

**Ex)** Sn\_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) |

* + 1. Sn\_DPORTR (SOCKET n Destination Port Register)
* [R=W] [0x0010+0x0100\*(n+4), 0x0011+0x0100\*(n+4)] [0x0000]

Sn\_DPORTR은 상대방의 Port이다.

TCP인 경우, 접속할 상대방의 Port를 설정하거나, 접속된 상대방의 Port를 확인한다.

UDP인 경우, 전송할 상대방의 Port를 설정한다.

UDP Multicast인 경우, Multicast Group의 Port를 설정한다.(Multicast Application Note 참조)

UDP, UDP Multicast인 경우, 수신된 상대방의 Port는 SOCKET n RX Buffer에서 확인한다.

**Ex)** S0\_DPORTR = 5000 (0x1388),

|  |  |
| --- | --- |
| S0\_DPORTR0(0x0410) | S0\_DPORTR1(0x0411) |
| 0x13 | 0x88 |

* + 1. Sn\_MSS (SOCKET n Maximum Segment Size Register)
* [R=W] [0x0012+0x0100\*(n+4), 0x0013+0x0100\*(n+4] [0xFFFF]

Sn\_MSS는 SOCKET의 MSS를 설정한다.

SOCKET의 MSS는 Sn\_CR[OPEN] 이전에 설정해야 한다.

설정 가능한 MSS Size를 초과할 경우 최대 MSS Size로 설정된다.

|  |  |  |
| --- | --- | --- |
| 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 |

* + 1. Sn\_PROTOR (SOCKET n IP Protocol Register)
* [R=W] [0x0014+0x0100\*(n+4)] [0x0000]

Sn\_PROTOR은 IPRAW Mode에서 IGMP(0x01), TCP(0x06), UDP(0x11), 를 제외한 Protocol number([IANA 참조](http://www.iana.org/assignments/protocol-numbers))를 설정한다.

IPRAW인 경우, Sn\_PROTR에 설정된 Protocol만 송수신이 가능하다.

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

* + 1. Sn\_TOS (SOCKET n IP Type Of Service Register)
* [R=W] [0x0015+0x0100\*(n+4)] [0x00]

Sn\_TOS는 IP Header의 TOS field([IANA 참조](http://www.iana.org/assignments/ip-parameters))를 설정한다.

* + 1. Sn\_TTL (SOCKET n IP Time To Live Register)
* [R=W] [0x0016+0x0100\*(n+4)] [0x80]

Sn\_TTL은 IP header의 TTL field([IANA 참조](http://www.iana.org/assignments/ip-parameters))를 설정한다.

* + 1. Sn\_RXBUF\_SIZE (SOCKET n RX Buffer Size Register)
* [RW] [0x001E+0x0100\*(n+4)] [0x02]

Sn\_RXBUF\_SIZE는 SOCKET n의 RX Buffer Size를 0, 1, 2, 4, 8 Kbyte 단위로 설정한다.

RX Memory는 SOCKET 0 부터 SOCKET 3 까지 Sn\_RXBUF\_SIZE 값으로 순차적으로 할당된다.

만약 그 외의 값으로 설정되거나 Sn\_RXBUF\_SIZE의 총합이 8Kbyte를 초과하는 경우 오작동할 수 있다.

Sn\_RXBUF\_SIZE는 RMSR을 통해서도 설정할 수 있다.

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

**Ex)** S0\_RXBUF\_SIZE = 8KB

|  |
| --- |
| S0\_RXBUF\_SIZE(0x041E) |
| 0x08 |

* + 1. Sn\_TXBUF\_SIZE (SOCKET n TX Buffer Size Register)
* [RW] [0x001F+0x0100\*(n+4)] [0x02]

Sn\_TXBUF\_SIZE는 SOCKET n이 TX Buffer Size를 0, 1, 2, 4, 8 Kbyte 단위로 설정한다.

TX Memory는 SOCKET 0 부터 SOCKET 3까지 Sn\_TXBUF\_SIZE 값으로 순차적으로 할당된다.

만약 그 외의 값으로 설정되거나 Sn\_TXBUF\_SIZE의 총합이 8 Kbyte를 초과하는 경우 오작동할 수 있다.

Sn\_TXBUF\_SIZE는 TMSR을 통해서도 설정할 수 있다.

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

**Ex)** S0\_TXBUF\_SIZE= 4KB

|  |
| --- |
| S0\_TXBUF\_SIZE(0x041F) |
| 0x04 |

* + 1. Sn\_TX\_FSR (SOCKET n TX Free Size Register)
* [RO] [0x0020+0x0100\*(n+4), 0x0021+0x0100\*(n+4)] [0x0800]

Sn\_TX\_FSR은 SOCKET n TX Buffer Block의 전송 가능한 Buffer Size를 확인한다.

이 값은 W5100S에 의해 자동으로 계산된다.

|  |
| --- |
| 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* |

Sn\_TX\_FSR보다 큰 Data를 SOCKET n TX Buffer Block에 저장하지 않게 주의한다.

**Ex)** S0\_TX\_FSR = 1024 (0x0400),

|  |  |
| --- | --- |
| S0\_TX\_FSR0(0x0420) | S0\_TX\_FSR1(0x0421) |
| 0x04 | 0x00 |

* + 1. Sn\_TX\_RD (SOCKET n TX Read Pointer Register)
* [RO] [0x0022+0x0100\*(n+4), 0x0023+0x0100\*(n+4)] [0x0000]

Sn\_TX\_RD는 SOCKET n TX Buffer block에서 마지막으로 전송된 Data의 주소이다.

Sn\_TX\_RD는 Sn\_CR[OPEN]에 의해 초기화된다. TCP 인 경우, TCP 접속과정에서 재설정 된다.

Sn\_CR[SEND]나 Sn\_CR[SEND\_MAC]는 SOCKET n TX Buffer에 Sn\_TX\_RD부터 Sn\_TX\_WR까지 저장된 Data를 전송하고 Sn\_TX\_RD를 Sn\_TX\_WR과 같은 값으로 자동으로 증가시킨다.

**Ex)** S0\_TX\_RD = 0xd4b3,

|  |  |
| --- | --- |
| S0\_TX\_RD0(0x0422) | S0\_TX\_RD1(0x0423) |
| 0xd4 | 0xb3 |

* + 1. Sn\_TX\_WR (SOCKET n TX Write Pointer Register)
* [RW] [0x0024+0x0100\*(n+4), 0x0025+0x0100\*(n+4)] [0x0000]

Sn\_TX\_WR는 SOCKET n TX Buffer Block 에 마지막으로 저장한 Data의 주소이다.

Sn\_TX\_WR은 Sn\_CR[OPEN]에 의해 초기화된다. TCP인 경우, TCP 접속과정에서 재설정 된다.

전송할 Data를 SOCKET n TX Buffer Block에 저장하고 해당 Data Size만큼 Sn\_TX\_WR을 증가시킨 후, Sn\_CR[SEND]나 Sn\_CR[SEND\_MAC]를 수행한다.

**Ex)** S0\_TX\_WR = 0x0800,

|  |  |
| --- | --- |
| S0\_TX\_WR0(0x0424) | S0\_TX\_WR1(0x0425) |
| 0x08 | 0x00 |

* + 1. Sn\_RX\_RSR (SOCKET n RX Received Size Register)
* [RO] [0x0026+0x0100\*(n+4), 0x0027+0x0100\*(n+4)] [0x0000]

Sn\_RX\_RSR은 SOCKET n RX Buffer에 수신된 Data Size를 알려준다.

이 값은 Sn\_RXBUF\_SIZE를 초과할 수 없으며 W5100S에 의해 자동으로 계산된다.

|  |
| --- |
| 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 |

* + 1. Sn\_RX\_RD (SOCKET n RX Read Pointer Register)
* [RW] [0x0028+0x0100\*(n+4), 0x0029+0x0100\*(n+4)] [0x0000]

Sn\_RX\_RD는 HOST가 마지막으로 읽은 SOCKET n RX Buffer Block의 주소이다.

Sn\_CR[RECV]는 SOCKET n RX Buffer에서 Sn\_RX\_RD부터 Sn\_RX\_WR까지 저장된 Data를 읽고 Sn\_RX\_RD를 Sn\_RX\_WR과 같은 값으로 자동으로 증가시킨다.

**Ex)** S0\_RX\_RD = 0x0600,

|  |  |
| --- | --- |
| S0\_RX\_RD0(0x0428) | S0\_RX\_RD1(0x0429) |
| 0x06 | 0x00 |

* + 1. Sn\_RX\_WR (SOCKET n RX Write Pointer Register)
* [RO] [0x002A+0x0100\*(n+4), 0x002B+0x0100\*(n+4)] [0x0000]

Sn\_RX\_WR은 SOCKET n RX Buffer Block에 마지막으로 수신된 Data의 주소이다.

수신된 Data의 크기가 Sn\_RX\_RSR과 같거나 작은 경우 해당 Data는 SOCKET n RX Buffer Block에 저장되고 Data의 크기만큼 Sn\_RX\_WR는 증가한다.

**Ex)** S0\_RX\_WR = 0x0600,

|  |  |
| --- | --- |
| S0\_RW\_WR0(0x042A) | S0\_RW\_WR1(0x042B) |
| 0x06 | 0x00 |

* + 1. Sn\_IMR (SOCKET n Interrupt Mask Register)
* [R=W] [0x002C+0x0100\*(n+4)] [0xFF]

Sn\_IMR은 Sn\_IR의 1:1 대응되는 Event 발생 시 IR[Sn\_INT]=’1’로 설정한다.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

* + 1. Sn\_FRAGR (SOCKET n Fragment Register)
* [R=W] [0x002D+0x0100\*(n+4), 0x002E+0x0100\*(n+4)] [0x4000]

Sn\_FRAGR은 IP Header의 Fragment field를 설정한다.

**Ex)** S0\_FRAG0 = 0x0000 (Don’t Fragment)

|  |  |
| --- | --- |
| S0\_FRAGR0(0x042D) | S0\_FRAGR1(0x042E) |
| 0x00 | 0x00 |

* + 1. Sn\_MR2 (SOCKET n Mode register 2)
* [R=W] [0x002F+0x0100\*(n+4)] [0x00]

Sn\_MR2는 Multicast Block, IPv6 Block, No check Checksum, Broadcast Block, Unicast Block을 설정한다.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | MBBLK | MMBLK | IPV6BLK | UNOCHK | TNOCHK | UBBLK | UUBLK |
| - | R=W | R=W | R=W | R=W | R=W | R=W | R=W |

|  |  |  |
| --- | --- | --- |
| Bit | Symbol | Description |
| 7 | - | Reserved |
| 6 | MBBLK | Broadcast Blocking in MACRAW mode  MACRAW Mode경우에 Broadcast Packet을 Blocking하기위해 이 Bit는 ‘1’로 설정한다.  0 : disable Broadcast Blocking  1 : enable Broadcast Blocking |
| 5 | MMBLK | Multicast Blocking in MACRAW mode  MACRAW Mode경우에 Multicast MAC Packet을 Blocking 하기 위해 이Bit는 ‘1’로 설정한다.  0 : disable Multicast Blocking  1 : enable Multicast Blocking |
| 4 | IPV6BLK | IPv6 packet Blocking in MACRAW mode  MACRAW Mode경우에 IPv6 Packet을 Blocking 하기 위해 이Bit는 ‘1’로 설정한다.  0 : disable IPv6 Blocking  1 : enable IPv6 Blocking |
| 3 | UNOCHK | No Check Checksum in UDP Mode  UDP Mode 에서 Checksum check를 하지 않기 위해서 이 Bit는 ‘1’로 설정한다.  0: enable check Checksum  1: disable check Checksum |
| 2 | TNOCHK | No Check Checksum in TCP Mode  TCP Mode 에서 Checksum check를 하지 않기 위해 이 Bit는 ‘1’로 설정한다.  0: enable check Checksum  1: disable check Checksum |
| 1 | UBBLK | Broadcast Blocking in UDP mode/ TCP Force PSH  \*Broadcast Blocking in UDP mode  UDP Mode 에서 Broadcast Packet을 Blocking하기위해 이 Bit는 ‘1’로 설정한다.  0 : disable Broadcast Blocking  1 : enable Broadcast Blocking  \*TCP Force PSH  TCP Mode에서 Data전송시, 모든 Packet에 PSH Flag를 설정하여 전송한다.  1: Force PSH Flag  0: No Force PSH Flag |
| 0 | UUBLK | UNICAST Blocking in UDP mode  UDP Mode에서 Multicasting 기능을 사용할 경우에 Unicast Packet을 Blocking하기위해 이 Bit를 ‘1’로 설정한다.  0 : disable Unicast Blocking  1 : enable Unicast Blocking |

* + 1. Sn\_KPALVTR (SOCKET n Keep Alive Timer Register)
* [RO] [0x0030+0x0100\*(n+4)] [0x00]

Sn\_KPALVTR는 SOCKET n의 Keep Alive(KA) Packet의 전송주기를 설정한다.

단위는 5 sec이다. SOCKET n 은 TCP Mode에서 Sn\_SR = ESTABLISHED이고 한 번 이상의 Data를 교환한 경우에 Sn\_KPALVTR의 값이 ‘0’보다 크면 KA Packet을 전송한다.

KA Packet은 Sn\_KPALVTR이 ‘0’인 경우 Sn\_CR[SENDKEEP]을 통해 전송 가능하고, 그렇지 않은 경우, Sn\_CR[SENDKEEP]은 무시된다.

**Ex)** S0\_KPALVTR = 10 (0x0A),

10 \* 5s = 50s

|  |
| --- |
| S0\_KPALVRT(0x0430) |
| 0x0A |

* + 1. Sn\_TSR (SOCKET n Timer Status Register)
* [RO] [0x0031+0x0100 \*(n+4)] [0x01]

Sn\_TSR은 TCP Mode에서 ACK의 송수신 상태를 확인한다.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | - | - | - | - | TX\_NEED | RX\_NEED | DISABLE |

|  |  |  |
| --- | --- | --- |
| Bit | Symbol | Description |
| [7:3] | - | Reserved |
| 2 | TX\_NEED | TX NEEDED ACK  0 : PEER에게 ACK 전송한 상태  1 : PEER에게 ACK 전송하지 않은 상태 |
| 1 | RX\_NEED | RX NEEDED ACK  0 : PEER로부터 ACK를 수신한 상태  1 : PEER로부터 ACK를 수신하지 않은 상태 |
| 0 | DISABLE | Timer disable  ACK를 전송해야 하거나, 수신해야 하는 경우 이 bit는 1로 설정된다.  1로 설정 될 경우 재전송한다. |

* + 1. Sn\_RTR (SOCKET n Retry Time-value Register)
* [R=W] [0x0032+0x0100\*(n+4), 0x0033+0x0100\*(n+4)] [0x0000]

Sn\_RTR은 SOCKET n의 재전송 시간을 설정한다. RTR이 모든 SOCKET의 재전송 시간을 설정할 수 있는데 반해 Sn\_RTR은 특정 SOCKET의 재전송 시간을 설정한다.

Sn\_RTR의 값이 ‘0’인 경우 RTR의 값으로 초기화된다.

Ex) S0\_RTR = 5000 (0x1388),

5000 \* 100us = 0.5s

|  |  |
| --- | --- |
| S0\_RTR0(x0432) | S0\_RTR1(0x0433) |
| 0x013 | 0x88 |

* + 1. Sn\_RCR (SOCKET n Retry Count-value Register)
* [R=W] [0x0034+0x0100\*(n+4)] [0x00]

Sn\_RCR은 SOCKET n의 재전송 횟수를 설정하는 Register이다. RCR이 모든 SOCKET의 재전송 시간을 설정할 수 있는데 반해 Sn\_RCR은 특정 SOCKET 의 재전송 횟수를 설정할 수 있다. Sn\_RCR의 값이 ‘0’인 경우 RCR의 값으로 초기화된다.

SOCKET은 Sn\_CR의 CONNECT, SEND, MACSEND, SENDKEEP, DISCON에 대해 상대방의 응답이 없는 경우 재전송을 수행하고 정해진 재전송 횟수이상으로 재전송하는 경우 Sn\_IR[TIMEOUT] = ‘1’ 이 된다.

ARP Request Packet에 대한 재전송의 경우는 Sn\_RTR에 의해 설정된 시간마다 재전송 Packet을 전송하고 Sn\_RCR만큼 재전송을 시도하고 상대방의 응답이 없는 경우 Sn\_IR[TIMEOUT]= ‘1’ 로 설정된다. ARP와 PING의 경우 Sn\_IR[TIMEOUT] = ‘1’이 될 때까지의 재전송의 총 시간()은 다음과 같이 계산할 수 있다.

|  |
| --- |
| = (RTR x 0.1ms) x (RCR + 1) |

TCP의 경우 SYN, FIN, DATA Packet을 전송했을 때 상대방의 ACK Packet을 수신하지 못한 경우 재전송 과정을 수행한다. 이때 Sn\_RTR, Sn\_RCR의 설정에 의해 재전송 시간과 전송 횟수가 결정된다. TCP의 경우 재전송 횟수가 증가할수록 재전송 시간은 지수 배 증가하게 된다. TCP에서 Sn\_IR[TIMEOUT] = ‘1’이 될 때의 최종 재전송 시간은 다음과 같이 계산할 수 있다.

|  |
| --- |
| N : 재전송 횟수, 0  M : RTR 65535 and 0 에서의 최소값  : RTR |

Ex) RTR = 2000(0x07D0), RXR = 8(0x0008)

= 2000 X 0.1ms X 9 = 1.8s

= (0x07D0+0x0FA0+0x1F40+0x3E80+0x7D00+0xFA00+0xFA00+0xFA00+0xFA00) X 0.1ms

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

= 318000 X 0.1ms = 31.8s

1. Functional Description

W5100S는 간단한 Register 조작만으로 Internet Connectivity를 제공한다. 이 Chapter에서는 W5100S의 초기화와 각 Protocol(TCP, UDP, IPRAW, MACRAW) 및 추가 기능에 따른 Data 통신방법에 대하여 각 단계별로 Pseudo Code를 기반으로 살펴본다.

* 1. W5100S RESET

RESET 타이밍에 결정되는 HOST Interface

* 1. Initialization

~~W5100S의 초기화는 Host Interface Setting -> Network Setting -> SOCKET Memory Setting 순으로 이루어진다.~~

W5100S의 초기화는 Network Setting, SOCKET n TX/RX Buffer Size Setting를 설정한다.

* + 1. Basic Setting

W5100S의 동작을 위하여 아래의 Register들을 사용자의 Application에 맞게 설정한다.

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

위 Register들의 자세한 내용은 Register Description을 통해 알 수 있다.

* + 1. Network Information Setting

W5100S Network Information 설정

|  |
| --- |
| NETWORK SETTING:  {  /\* W5100S MAC address, 11:22:33:AA:BB:CC \*/  SHAR0,1,2 = 0x11, 0x22, 0x33;  SHAR3,4,5 = 0xAA, 0xBB, 0xCC;  /\* W5100S Gateway IP address, 192.168.0.1 \*/  GAR0,1 = 0xC0, 0xA8;  GAR2,3 = 0x00, 0x01;  /\* W5100S Subnet MASK address, 255.255.255.255 \*/  SUBR0,1 = 0xFF, 0xFF;  SUBR2,3 = 0xFF, 0xFF;  /\* W5100S IP address, 192.168.0.100 \*/  SIPR0,1 = 0xC0, 0xA8;  SIPR2,3 = 0x00, 0x64;  } |

* + 1. SOCKET Memory Setting

W5100S의 설정 가능한 TX, RX 메모리의 최대 사이즈는 각각 8Kbytes씩 이다. 8Kbytes의 범위 안에서는 0KB, 1KB, 2KB, 4KB, 8KB의 Size로 4개의 소켓까지 자유롭게 설정이 가능하지만 TX, RX의 사이즈가 각각 8KB를 넘어가서는 안된다.

다음은 SOCKET n의 RX/TX Memory를 설정하는 의사코드를 예로 나타낸 것이다.

|  |
| --- |
| In case of, assign 2KB rx, tx memory per SOCKET  {  gS0\_RX\_BASE = 0x0000(Chip base address) + 0xC000(Internal RX buffer address); // Set base address of RX memory for SOCKET 0  Sn\_RXMEM\_SIZE(ch) = (uint8 \*) 2; // Assign 2K rx memory per SOCKET  gS0\_RX\_MASK = 2K – 1; // 0x07FF, for getting offset address within assigned SOCKET 0 RX memory  gS1\_RX\_BASE = gS0\_RX\_BASE + (gS0\_RX\_MASK + 1);  gS1\_RX\_MASK = 2K – 1;  gS2\_RX\_BASE = gS1\_RX\_BASE + (gS1\_RX\_MASK + 1);  gS2\_RX\_MASK = 2K – 1;  gS3\_RX\_BASE = gS2\_RX\_BASE + (gS2\_RX\_MASK + 1);  gS3\_RX\_MASK = 2K – 1;  gS0\_TX\_BASE = 0x0000(Chip base address) + 0x8000(InternalTX buffer address); // Set base address of TX memory for SOCKET 0  Sn\_TXMEM\_SIZE(ch) = (uint8 \*) 2; // Assign 2K rx memory per SOCKET  gS0\_TX\_MASK = 2K – 1;  /\* Same method, set  gS1\_TX\_BASE, gS1\_TX\_MASK,  gS2\_TX\_BASE, gS2\_TX\_MASK,  gS3\_TX\_BASE, gS3\_TX\_MASK \*/  } |

* 1. TCP

TCP(Transmission Control Protocol)는 IP Layer위의 전송계층에 위치하며 1:1연결 기반의 양방향 데이터 전송 프로토콜이다. 또한 Port Number를 이용해 Application 간 통신을 제공한다.

TCP는 1:1연결 기반이므로 상대방과의 데이터 송수신을 위해서 상대방에게 연결을 요청 하거나 상대방으로부터 연결요청을 받아야 한다. 이 과정에서 먼저 연결을 요청하는 쪽을 ‘TCP CLIENT’, 연결요청을 받은 쪽을 ‘TCP SERVER’로 구분한다. TCP는 또한 신뢰성 있는 전송 프로토콜로써 연결이 완료된 이후의 데이터 송수신 과정에서 확인응답 메커니즘을 통해 전송 데이터의 수신여부를 확인하고 손실되거나 훼손된 데이터는 재전송한다. ‘TCP SERVER’와 ‘TCP CLIENT’는 TCP연결종료가 이루어지기 전까지 연결을 유지하며 데이터를 송수신한다.

C:\Documents and Settings\wiznet\바탕 화면\W5200_TCP_FLOW.emf

Figure 4 . TCP SERVER and TCP CLIENT

* + 1. TCP Server

아래의 Figure 5는 TCP Mode SOCKET이 TCP Server일때의 동작흐름을 나타낸다.

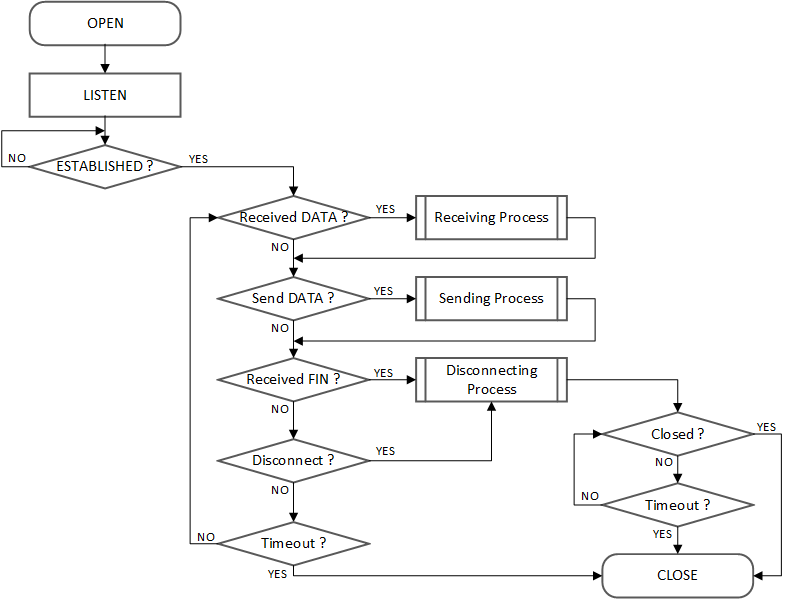


Figure 5 . TCP Server Operation Flow

• OPEN

HOST는 SOCKET n을 TCP Mode로 설정한다.

|  |
| --- |
| {  START :  /\* set TCP Mode \*/  Sn\_MR[3:0] = 4’b0001;  /\* optional flag in TCP Mode \*/  /\* set No Delay ACK \*/  Sn\_MR[ND] = ‘1’;  /\* set PORT number, 5000(0x1388) \*/  Sn\_PORTR0 = 0x13;  Sn\_PORTR1 = 0x88;  /\* set OPEN command \*/  Sn\_CR = OPEN;  /\* check SOCKET for TCP Mode \*/  if(Sn\_SR != SOCK\_INIT) goto START;  } |

• LISTEN

TCP Mode SOCKET은 Sn\_CR[LISTEN]을 통해 ‘TCP SERVER’로 동작한다.

|  |
| --- |
| {  START :  /\* set LISTEN command \*/  Sn\_CR = LISTEN;  /\* check SOCKET status \*/  if(Sn\_SR != SOCK\_LISTEN) goto START;  } |

• ESTABLISHED

‘TCP SERVER’로 동작하는 SOCKET은 SYN Packet수신전까지 Sn\_SR=SOCK\_LISTEN을 유지한다. ‘TCP SERVER’ SOCKET이 ‘TCP CLIENT’로부터 SYN Packet을 수신하면 SYN/ACK Packet을 전송하고 ‘TCP SERVER’와 ‘TCP CLIENT’의 연결이 완료된다.

HOST는 Sn\_IR[CON]이 발생하거나 Sn\_SR이 SOCK\_ESTABLISHED로 바뀌면 연결이 완료 되었음을 알 수 있으며 연결완료 이후 SOCKET을 통해 데이터를 송수신한다.

|  |
| --- |
| First method :  {  START :  /\* check SOCKET interrupt \*/  if (Sn\_IR[CON] != ‘1’) goto START;  /\* clear interrupt \*/  Sn\_IR[CON] = ‘1’;  /\* Sn\_IMR, Sn\_IR 참조 \*/  } |
| Second method :  {  START :  /\* check SOCKET status \*/  if (Sn\_SR != SOCK\_ESTABLISHED) goto START;  } |

• Receive DATA?

TCP Mode SOCKET의 DATA수신여부는 Sn\_IR[RECV] 또는 Sn\_RX\_RSR을 통해 확인한다.

|  |
| --- |
| First method :  {  /\* check SOCKET interrupt \*/  if (Sn\_IR[RECV] == ‘1’) Sn\_IR[RECV] = ‘1’; goto Receiving Process;  /\* Sn\_IMR, Sn\_IR 참조 \*/  } |
| Second method :  {  /\* check SOCKET RX Memory received size \*/  if (Sn\_RX\_RSR != 0x0000) goto Send DATA?;  else goto Receiving Process;  } |

• Receiving Process

TCP Mode SOCKET이 Packet을 수신하면 해당 SOCKET RX Buffer Block에는 Transport Layer의 TCP Header를 제외한 DATA만 저장된다. SOCKET RX Buffer Block으로부터 저장된 DATA를 읽은 후에 반드시 Sn\_CR[RECV]을 통해서 Sn\_RX\_RD를 RX Buffer Block에서 읽은 DATA 크기만큼 증가시킨다 (참조 Sn\_RX\_RD). 만약 Sn\_CR[RECV]시, SOCKET RX Buffer Block에 읽지 않은 DATA가 있는 경우 Sn\_IR[RECV]가 발생한다.

|  |
| --- |
| {  /\* get Received size \*/  get\_size = Sn\_RX\_RSR;  /\* calculate offset address \*/  get\_offset = Sn\_RX\_RD & gSn\_RX\_MASK;  /\* calculate start address \*/  get\_start\_address = gSn\_RX\_BASE + get\_offset;  /\* if overflow SOCKET n RX Buffer Block \*/  If( (get\_offset + get\_size) > gSn\_RX\_MAX )  {  /\* copy upper size bytes of get\_start\_address to destination address \*/  upper\_size = gSn\_RX\_MAX – get\_offset;  memcpy(get\_start\_address, destination\_ptr, upper\_size);    /\* copy left size bytes of gSn\_rx\_BASE to destination address \*/  left\_size = get\_size – upper\_size;  memcpy(gSn\_RX\_BASE, destination address, left\_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 length of get\_size \*/  Sn\_RX\_RD += get\_size;  /\* set RECV command \*/  Sn\_CR = RECV;  } |

• Send DATA? / Sending Process

TCP Mode SOCKET으로 DATA전송 시, DATA의 크기가 Sn\_TXBUF\_SIZE보다 클 경우, HOST는 해당 DATA를 나누어 전송해야 한다. DATA가 MSS(1460 Byte)보다 큰 경우에는 SOCKET에 의해 자동으로 DATA가 나뉘어 전송된다.

|  |
| --- |
| {  /\* check Sending DATA Size \*/  get\_tx\_size = gSn\_TX\_MAX;  if(send\_size > get\_tx\_size) send\_size = get\_tx\_size;  /\* calculate offset address \*/  get\_offset = Sn\_TX\_WR & gSn\_TX\_MASK;  /\* calculate start address \*/  get\_start\_address = gSn\_TX\_BASE + get\_offset;  /\* if overflow SOCKET n TX Buffer Block \*/  If( (get\_offset + send\_size) > gSn\_TX\_MAX )  {  /\* copy upper size byte of source\_address to get\_start\_address \*/  upper\_size = gSn\_TX\_MAX – get\_offset;  memcpy(source\_address, get\_start\_address, upper\_size);  /\* copy left size byte of source\_address to gSn\_TX\_BASE \*/  source\_address += upper\_size;  left\_size = send\_size – upper\_size;  memcpy(source\_address, gSn\_TX\_BASE, left\_size);  }  else  {  /\* copy send size byte of source\_address to get\_start\_address \*/  memcpy(source\_address, get\_start\_address, send\_size);  }  /\* increase Sn\_TX\_WR as length of send size \*/  Sn\_TX\_WR += send\_size;  /\* set SEND command \*/  Sn\_CR = SEND;  } |

• Received FIN?

상대방의 연결종료를 알리는 FIN Packet을 수신한 경우

|  |
| --- |
| {  /\* wait for FIN packet from peer \*/  while(Sn\_SR == SOCK\_ESTABLISHED)  {  /\* receive FIN packet \*/  If(Sn\_IR == DISCON) break; goto Disconnecting Process;  }  } |

• Disconnected

상대방에게 연결종료를 알리는 FIN Packet을 전송한 경우

|  |
| --- |
| {  /\* check SOCKET status \*/  while(Sn\_SR == SOCK\_ESTABLISHED)  {  /\* send FIN packet \*/  Sn\_CR = DISCON; goto Disconnecting Process;  }  } |

• Disconnecting Process

상대방으로부터 FIN Packet을 수신한 경우 또는 상대방에게 FIN Packet을 전송한 경우 TCP Mode SOCKET은 다음과 같은 두 가지 방법으로 정상적인 연결종료를 수행한다.

|  |
| --- |
| First method : /\* Received FIN packet from peer \*/  {  /\* send FIN packet \*/  Sn\_CR = DISCON;    /\* No received ACK packet \*/  if(Sn\_IR[DISCON] != ‘1’) goto Timeout?;  else  {  /\* received ACK packet \*/  /\* clear interrupt \*/  Sn\_IR[DISCON] = ‘1’; goto CLOSED;  /\* Sn\_IMR, Sn\_IR 참조 \*/  }  } |
| Second method : /\* sent FIN packet to peer \*/  {  /\* receive FIN packet \*/  If(Sn\_IR[DISCON] != ‘1’) goto Timeout?;  else  {  /\* received ACK packet \*/  /\* clear interrupt \*/  Sn\_IR[DISCON] = ‘1’; goto CLOSED;  /\* Sn\_IMR, Sn\_IR 참조 \*/  }  } |

• Timeout

TCP Mode SOCKET의 전송 Packet에 대해 아래와 같은 상대방 응답 Packet이 없는 경우 SOCKET은 전송 Packet에 대해 재전송을 수행한다.

1. SYN Packet에 대한 상대방의 SYN/ACK Packet
2. DATA Packet에 대한 상대방의 ACK Packet
3. FIN Packet에 대한 상대방의 ACK Packet

Sn\_RTR, Sn\_RCR에 의한 Sn\_IR[TIMEOUT]발생시, SOCKET은 자동으로 CLOSE된다.

|  |
| --- |
| First method : /\* from ESTABLISHED? state in TCP Client \*/  {  while(Sn\_IR[CON] != ‘1’)  {  /\* check interrupt\*/  if(Sn\_IR[TIMEOUT] == ‘1’) break; goto CLOSE;  }  /\* check interrupt \*/  if(Sn\_IR[CON] == ‘1’) goto ESTABLISHED?;  } |
| Second method : /\* from Sending Process state in TCP Server/ TCP Client \*/  {  while(Sn\_IR[SENDOK] != ‘1’)  {  /\* check SOCKET status \*/  if(Sn\_IR[TIMEOUT] == ‘1’) break; goto CLOSE;  }  if(Sn\_IR[SENDOK] == ‘1’)  {  /\* clear interrupt \*/  Sn\_IR[SENDOK] = ‘1’;  /\* Sn\_IMR, Sn\_IR 참조 \*/  }  } |
| Third method : /\* from Disconnecting Process state in TCP Server/ TCP Client \*/  {  while(Sn\_IR[DISCON] != ‘1’)  {  /\* check SOCKET status \*/  if(Sn\_IR[TIMEOUT] == ‘1’) break; goto CLOSE;  }  if(Sn\_IR[DISCON] == ‘1’) goto Disconnecting Process;  } |

• CLOSE

TCP Mode SOCKET은 Disconnecting Process, Sn\_IR[TIMEOUT], Sn\_CR[CLOSE]에 의해 CLOSE 된다.

|  |
| --- |
| First method : /\* Closed by Disconnecting Process state \*/  {  /\* check SOCKET status \*/  while(Sn\_SR == SOCK\_FIN\_WAIT) || (Sn\_SR == SOCK\_TIME\_WAIT) ||  (Sn\_SR == SOCK\_LAST\_ACK)  {  /\* check interrupt \*/  If(Sn\_IR[SOCK\_DISCON] == ‘1’) break;  }  /\* check SOCKET status \*/  if(Sn\_SR == SOCK\_CLOSED) SOCKET CLOSED;  } |
| Second method : /\* Closed by Sn\_IR[TIMEOUT] \*/  {  /\* check TIMEOUT interrupt \*/  if(Sn\_IR[TIMEOUT] == ‘1’)  {  /\* clear interrupt \*/  Sn\_IR[TIMEOUT] = ‘1’;  /\* Sn\_IMR, Sn\_IR 참조 \*/  /\* check SOCKET status \*/  if(Sn\_SR == SOCK\_CLOSED) SOCKET CLOSED;  }  } |
| Third method : /\* Closed by Sn\_CR[CLOSE] \*/  {  /\* set CLOSE command \*/  Sn\_CR = CLOSE;  /\* check SOCKET status \*/  if(Sn\_SR == SOCK\_CLOSED) SOCKET CLOSED;  } |

* + 1. TCP Client

Figure 6 는 TCP Mode SOCKET이 TCP Client로 동작할 때의 흐름을 나타낸다.

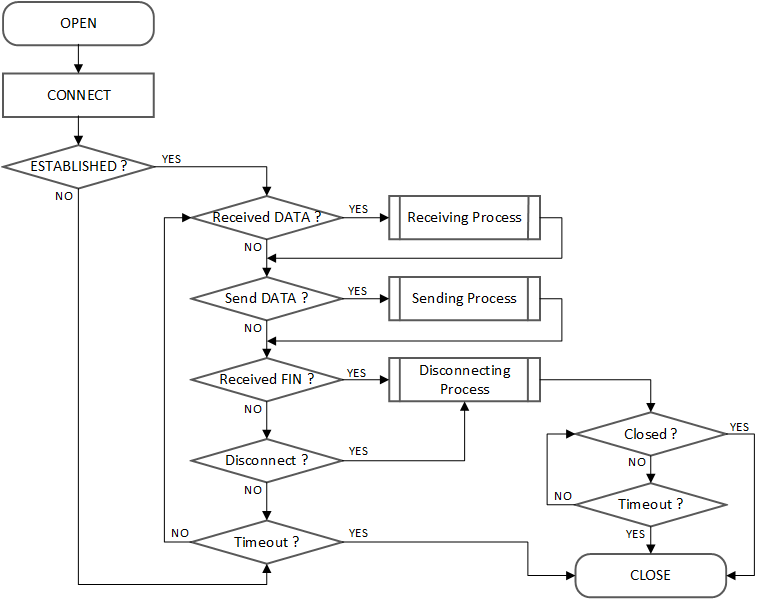


Figure 6 . TCP Client Operation Flow

• CONNECT

TCP Mode SOCKET은 Sn\_CR[CONNECT]을 통해 상대방에게 SYN Packet을 전송하고 ‘TCP CLIENT’로 동작한다.

|  |
| --- |
| {  /\* set destination IP address, 192.168.0.11 \*/  Sn\_DIPR0 = 0xC0;  Sn\_DIPR1 = 0xA8;  Sn\_DIPR2 = 0x00;  Sn\_DIPR3 = 0x0B;  /\* set destination PORT number, 5000(0x1388) \*/  Sn\_DPORTR0 = 0x13;  Sn\_DPORTR1 = 0x88;  /\* set CONNECT command \*/  Sn\_CR = CONNECT; goto ESTABLISHED?; |

• ESTABLISHED?

‘TCP CLIENT’ SOCKET은 ‘TCP SERVER’로 전송한 SYN Packet에 대한 SYN/ACK Packet을 수신하면 ‘TCP CLIENT’와 ‘TCP SERVER’의 연결이 완료된다.

HOST는 Sn\_IR[CON]이 발생하거나 Sn\_SR이 SOCK\_ESTABLISHED로 바뀌면 연결이 완료 되었음을 알 수 있으며 연결완료 이후 SOCKET을 통해 DATA를 송수신할 수 있다.

만약 ‘TCP SERVER’로부터 SYN/ACK Packet을 수신하지 못하면 재전송동작을 수행한다.

|  |
| --- |
| First method :  {  /\* check SOCKET interrupt \*/  if(Sn\_IR[CON] != ‘1’) goto Timeout?;  else  {  /\* clear interrupt \*/  Sn\_IR[CON] = ‘1’; goto Received DATA?  /\* Sn\_IMR, Sn\_IR 참조 \*/  }  } |
| Second method :  {  /\* check SOCKET status \*/  if(Sn\_SR != SOCK\_ESTABLISHED) goto Timeout?;  else goto Received DATA?  } |

* + 1. Other Functions
       1. • TCP SOCKET Option Flag

SOCKET을 OPEN하는 과정에서 Sn\_MR과 Sn\_MR2를 통해서 SOCKET Option Flag을 설정한다.

• No Delayed Ack Flag

No Delayed Ack Flag는 SOCKET이 TCP 통신중에 상대방 Packet에 대하여 자동으로 ACK Packet을 전송하는 기능이다. SOCKET은 Sn\_CR[RECV]를 수행한 이후 ACK Packet을 전송하며, SYN Packet이나 FIN Packet을 수신한 경우에만 자동적으로 ACK Packet을 전송한다. 그러나 SOCKET OPEN 전에 Sn\_MR[NDACK]=’1’로 설정하면, Sn\_CR[RECV]를 수행하지 않아도 Data Packet을 수신하면 ACK Packet을 자동적으로 전송한다.

• TCP Force PSH Flag

TCP Force PSH Flag는 모든 Packet에 PSH를 넣어 전송하는 기능이다. W5100S는 DATA Size가 1460byte보다 클 경우 Packet을 나누어서 전송한다. 이때 PSH는 마지막 Packet에만 적용된다. 그러나 TCP OPEN 전에 Sn\_MR2[UBBLK ]=’1’로 설정하면, 모든 Packet에 PSH를 넣어 전송한다.

* + - 1. • TCP KEEPALIVE

TCP KEEPALIVE는 1byte의 데이터를 전송하여 ACK Packet을 수신함으로써 상대방의 데이터 송수신 가능 여부를 확인하는 기능이다. TCP 통신에서는 Connection 이후에 상대방과의 연결을 계속 유지한다. 이때, W5100S는 상대방의 데이터 송수신 가능여부를 확인하기 위해 Keep Alive packet을 전송한다. SOCKET을 OPEN 하기전에는 Sn\_KPALVTR을 통해 주기를 설정하여 KA Packet을 전송하며, Sn\_CR[SEND\_KEEP]을 통해서도 전송한다. Sn\_CR[SEND\_KEEP]을 통해 전송할 경우에 Sn\_KPALVTR은 0x00으로 설정되어 있어야 한다.

* + - 1. • TCP FIN Send

Sn\_SR=SOCK\_ESTABLISHED일 때, Sn\_CR[DISCON]을 통해 FIN/ACK Packet을 전송하면 Sn\_SR=FINWAIT이 된다. 이때 상대방으로부터 FIN Packet을 수신하면 Sn\_SR=CLOSED가 된다.

Sn\_SR=FINWAIT에서는 Sn\_CR[RECV] 외에 다른 Command를 설정할 수 없다. 데이터 수신이나, 재전송의 경우는 수행된다..

* + - 1. • TCP FIN Receive

Sn\_SR=SOCK\_ESTABLISHED에서 상대방의 FIN Packet을 수신하면, ACK를 전송하고 Sn\_SR=CLOSE\_WAIT이 된다. Sn\_SR=CLOSE\_WAIT에서는 Sn\_SR=SOCK\_ESTABLISHED와 동일하게 DATA 송수신을 한다. 이때 Sn\_CR[DISCON]을 통해 FIN Packet을 전송한다. FIN Packet을 전송한 후에 ACK Packet을 수신하면 Sn\_SR=CLOSED가 된다.

* 1. UDP

UDP(User Datagram Protocol)는 IP Layer위의 전송계층에 위치하며 신뢰성을 보장하지 않는 Datagram 통신을 하는 프로토콜이다. 또한 Port Number를 이용해 Application 간 통신을 제공한다. UDP는 연결과정이 필요 없으며 하나 이상의 상대방과 통신을 할 수 있는 이점이 있는 반면, 데이터 전송에 대한 신뢰성을 보장하지 않으므로 데이터 송수신 과정에서 데이터 손실이나 원하지 않는 상대로부터의 데이터 수신이 발생할 수 있다. UDP 전송방식은 데이터 송수신 범위에 따라 크게 Unicast, Broadcast, Multicast로 구분한다.

아래의 Figure 7 는 UDP Mode SOCKET의 동작흐름을 나타낸다.

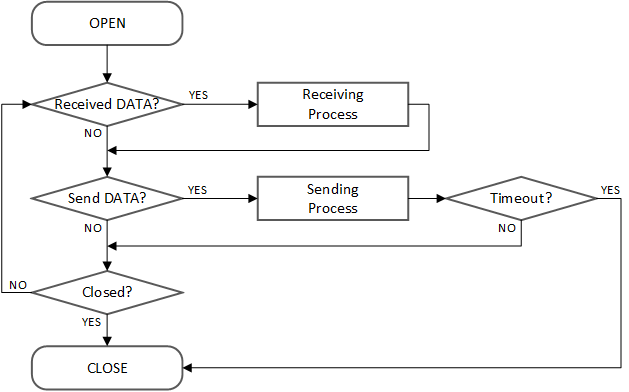


Figure 7 . UDP Operation Flow

* + 1. UDP Unicast

UDP Unicast는 하나의 송신자가 하나의 목적지에 데이터를 전송하는 통신방식이다. 데이터 전송 시SOCKET은 먼저 상대방과 ARP 과정을 수행한다.

• OPEN

HOST는 SOCKET n을 UDP Mode로 설정한다.

|  |
| --- |
| {  START :  /\* set UDP Mode \*/  Sn\_MR[3:0] = 4’b0010;  /\* set source PORT number, 5000(0x1388) \*/  Sn\_PORTR0 = 0x13;  Sn\_PORTR1 = 0x88;  /\* set OPEN command \*/  Sn\_CR = OPEN;  /\* check SOCKET for UDP Mode \*/  if(Sn\_SR != SOCK\_UDP) goto START;  } |

• Received DATA ?

UDP Mode SOCKET의 DATA수신여부는 Sn\_IR[RECV] 또는 Sn\_RX\_RSR을 통해 알 수 있다.

|  |
| --- |
| {  First method :  /\* check SOCKET RX Buffer \*/  If( Sn\_RX\_RSR != 0x0000 ) goto Receiving Process;  } |
| {  Second method :  /\* check SOCKET RECV interrupt \*/  If( Sn\_IR[RECV] == ‘1’) goto Receiving Process;  /\* IR, Sn\_IR, Sn\_IMR을 참조 \*/  } |

• Receiving Process

UDP Mode SOCKET이 Packet을 수신하면 해당 SOCKET RX Buffer Block에 아래 Figure 8과 같이 저장된다. SOCKET RX Buffer Block으로부터 저장된 DATA를 읽은 후에 반드시 Sn\_CR[RECV]을 통해서 Sn\_RX\_RD를 RX Buffer Block에서 읽은 DATA 크기만큼 증가시킨다 (참조 Sn\_RX\_RD). 만약 Sn\_CR[RECV]시, SOCKET RX Buffer Block에 읽지 않은 DATA가 있는 경우 Sn\_IR[RECV]가 발생한다.

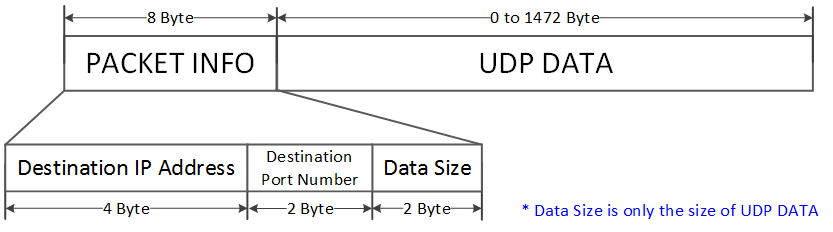


Figure . Received Data in UDP Mode SOCKET RX Buffer Block

|  |
| --- |
| {  START :  /\* first, get received size \*/  get\_size = Sn\_RX\_RSR;  /\* calculate offset address \*/  get\_offset = Sn\_RX\_RD & gSn\_RX\_MASK;  /\* calculate start address(SOCKET RX Buffer Block Address) \*/  get\_start\_address = gSn\_RX\_BASE + get\_offset;  /\* if overflow SOCKET RX Buffer Block \*/  If( (get\_offset + get\_size) > gSn\_RX\_MAX )  {  /\* copy upper size bytes of get\_start\_address to destination\_address \*/  upper\_size = gSn\_RX\_MAX – get\_offset;  memcpy(get\_start\_address, destination\_ptr, upper\_size);    /\* copy left size bytes of gSn\_rx\_BASE to destination\_address \*/  left\_size = get\_size – upper\_size;  memcpy(gSn\_RX\_BASE, destination\_address, left\_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 length of get\_size \*/  Sn\_RX\_RD += get\_size;  /\* set RECV command \*/  Sn\_CR = RECV;  } |

• Send DATA? / Sending Process

UDP Mode SOCKET으로 DATA전송 시, DATA의 크기는 Sn\_TXBUF\_SIZE보다 클 경우, HOST는 해당 DATA를 나누어 전송해야 한다. DATA가 MSS(1460 Byte)보다 큰 경우에는 SOCKET에 의해 자동으로 DATA가 나뉘어 전송된다. 또한 Sn\_CR[SEND\_MAC]을 통해 ARP과정없이 DATA를 전송할 수 있다. 이때, Sn\_DHAR에 상대방 MAC Address를 설정해야 한다.

|  |
| --- |
| {  /\* set destination IP address, 192.168.0.11 \*/  Sn\_DIPR0 = 0xC0;  Sn\_DIPR0 = 0xA8;  Sn\_DIPR0 = 0x00;  Sn\_DIPR0 = 0x0B;  /\* set destination PORT number, 5000(0x1388) \*/  Sn\_PORTR0 = 0x13;  Sn\_PORTR1 = 0x88;  /\* for using SEND\_MAC command \*/  /\* set destination MAC address, 11:22:33:AA:BB:CC \*/  Sn\_DHAR0,1,2 = 0x11, 0x22, 0x33;  Sn\_DHAR3,4,5 = 0xAA, 0xBB, 0xCC;  /\* check Sending DATA Size \*/  get\_tx\_size = gSn\_TX\_MAX;  if(send\_size > get\_tx\_size) send\_size = get\_tx\_size;  /\* calculate offset address \*/  get\_offset = Sn\_TX\_WR & gSn\_TX\_MASK;  /\* calculate start address \*/  get\_start\_address = gSn\_TX\_BASE + get\_offset;  /\* if overflow SOCKET n TX Buffer Block \*/  If( (get\_offset + send\_size) > gSn\_TX\_MAX )  {  /\* copy upper size byte of source\_address to get\_start\_address \*/  upper\_size = gSn\_TX\_MAX – get\_offset;  memcpy(source\_address, get\_start\_address, upper\_size);  /\* copy left size byte of source\_address to gSn\_TX\_BASE \*/  source\_address += upper\_size;  left\_size = send\_size – upper\_size;  memcpy(source\_address, gSn\_TX\_BASE, left\_size);  }  else  {  /\* copy send size byte of source\_address to get\_start\_address \*/  memcpy(source\_address, get\_start\_address, send\_size);  }  /\* increase Sn\_TX\_WR as length of send size \*/  Sn\_TX\_WR += send\_size;  /\* set SEND (or SEND\_MAC) command \*/  Sn\_CR = SEND; goto Timeout?;  } |

• Timeout?

UDP Mode SOCKET은 DATA 전송 전 ARP과정에서 상대방의 응답이 없는 경우 ARP Request 를 재전송하며Sn\_RTR, Sn\_RCR에 따라 Sn\_IR[TIMEOUT]=’1’이 된다.

|  |
| --- |
| {  while(Sn\_IR[SENDOK] != ‘1’)  {  /\* check SOCKET status \*/  if(Sn\_IR[TIMEOUT] == ‘1’) break; goto CLOSE;  }  if(Sn\_IR[SENDOK] == ‘1’)  {  /\* clear interrupt \*/  Sn\_IR[SENDOK] = ‘1’;  /\* Sn\_IMR, Sn\_IR 참조 \*/  }  } |

• CLOSE

UDP Mode SOCKET은 Sn\_IR[TIMEOUT] 또는 Sn\_CR[CLOSE]에 의해 CLOSE된다.

|  |
| --- |
| First method :  {  /\* check TIMEOUT interrupt \*/  if(Sn\_IR[TIMEOUT] == ‘1’)  {  /\* clear interrupt \*/  Sn\_IR[TIMEOUT] = ‘1’;  /\* Sn\_IMR, Sn\_IR 참조 \*/  /\* check SOCKET status \*/  if(Sn\_SR == SOCK\_CLOSED) SOCKET CLOSED;  }  } |
| Second method :  {  /\* set CLOSE command \*/  Sn\_CR = CLOSE;  /\* check SOCKET status \*/  if(Sn\_SR == SOCK\_CLOSED) SOCKET CLOSED;  } |

* + 1. UDP Broadcast

UDP Broadcast는 하나의 송신자가 같은 대역의 모든 호스트들에게 데이터를 전송하는 통신방식이다.

• Send DATA? / Sending Process

UDP Mode SOCKET으로 Broadcast를 사용해 DATA 전송 시, Sn\_DIPR은 같은 대역의 Broadcast Address로 설정한다.

|  |
| --- |
| {  /\* set broadcast address, 255.255.255.255 \*/  Sn\_DIPR0 = 0xFF;  Sn\_DIPR0 = 0xFF;  Sn\_DIPR0 = 0xFF;  Sn\_DIPR0 = 0xFF;  /\* set destination PORT number, 5000(0x1388) \*/  Sn\_PORTR0 = 0x13;  Sn\_PORTR1 = 0x88;    /\* UDP Unicast Send DATA?/Sending Process 참조 \*/  } |

* + 1. UDP Multicast

UDP Multicast는 하나의 송신자가 하나의 목적지 그룹에 데이터를 전송하는 통신방식이다.

Multicast-Group Address 범위는 224.0.0.0 ~ 239. 255.255.255 이며 (참조, [https://www.iana.org/.../multicast-addresses.xhtml](https://www.iana.org/assignments/multicast-addresses/multicast-addresses.xhtml)) 또한 이에 대응되는 MAC Address는 01:00:5E:00:00:00 ~ 01:00:5E:FF:FF:FF 이다. Multicast MAC Address 설정 시 하위 23 Bits는 Multicast-Group Address와 동일해야 한다. (참조, <https://tools.ietf.org/html/rfc1112>)

• OPEN

UDP Multicast로 SOCKET을 OPEN하기전 Multicast-Group의 Network Information을 설정한다. 그리고 SOCKET을 OPEN하면 해당 Multicast-Group으로 Internet Layer의 IGMP (Internet Group Management Protocol)을 통해 JOIN 메시지를 보낸다. W5100S SOCKET은 IGMPv1과 IGMPv2를 지원한다.

|  |
| --- |
| {  START :  /\* set Multicast-Group MAC address, 01:00:5E:00:00:64 \*/  Sn\_DHAR0,1 = 0x01, 0x00;  Sn\_DHAR2,3 = 0x5E, 0x00;  Sn\_DHAR4,5 = 0x00, 0x64;  /\* set Multicast-Group IP address, 224.0.0.100 \*/  Sn\_DIPR0,1 = 0xE0, 0x00;  Sn\_DIPR2,3 = 0x00, 0x64;  /\* set Multicast-Group PORT number, 3000(0x0BB8) \*/  Sn\_DPORTR0,1 = 0x0B, 0xB8;  /\* set UDP Mode \*/  Sn\_MR[3:0] = 4’b0010;  /\* set UDP Multicast \*/  Sn\_MR[MULTI] = ‘1’;  /\* set IGMP version \*/  Sn\_MR[MC] = ‘1’; // IGMPv1  Sn\_MR[MC] = ‘0’; // IGMPv2  /\* set source PORT number, 3000(0x0BB8) \*/  Sn\_PORTR0 = 0x0B;  Sn\_PORTR1 = 0xB8;  /\* set OPEN command \*/  Sn\_CR = OPEN;  /\* check SOCKET for UDP Mode \*/  if(Sn\_SR != SOCK\_UDP) goto START;  } |

• Send DATA? / Sending Process

UDP Mode SOCKET을 Multicast로 설정 후 DATA 전송 시 상대방 주소는 Sn\_DHAR, Sn\_DIPR과 Sn\_DPORTR로 자동 설정된다. 또한 Multicast는 ARP과정을 수행하지 않기 때문에 Timeout이 발생하지 않는다.

|  |
| --- |
| {  /\* check Sending DATA Size \*/  get\_tx\_size = gSn\_TX\_MAX;  if(send\_size > get\_tx\_size) send\_size = get\_tx\_size;  /\* calculate offset address \*/  get\_offset = Sn\_TX\_WR & gSn\_TX\_MASK;  /\* calculate start address \*/  get\_start\_address = gSn\_TX\_BASE + get\_offset;  /\* if overflow SOCKET n TX Buffer Block \*/  If( (get\_offset + send\_size) > gSn\_TX\_MAX )  {  /\* copy upper size byte of source\_address to get\_start\_address \*/  upper\_size = gSn\_TX\_MAX – get\_offset;  memcpy(source\_address, get\_start\_address, upper\_size);  /\* copy left size byte of source\_address to gSn\_TX\_BASE \*/  source\_address += upper\_size;  left\_size = send\_size – upper\_size;  memcpy(source\_address, gSn\_TX\_BASE, left\_size);  }  else  {  /\* copy send size byte of source\_address to get\_start\_address \*/  memcpy(source\_address, get\_start\_address, send\_size);  }  /\* increase Sn\_TX\_WR as length of send size \*/  Sn\_TX\_WR += send\_size;  /\* set SEND command \*/  Sn\_CR = SEND;  /\* check send complete \*/  while(Sn\_IR[SENDOK] == ‘1’);  /\* clear interrupt \*/  Sn\_IR[SENDOK] = ‘1’;  /\* Sn\_IMR, Sn\_IR 참조 \*/  } |

* + 1. Other Functions
       1. UDP MAC Send

UDP Unicast의 경우 전송시 상대방의 존재여부 및 MAC주소를 알기 위해서 ARP과정을 수행하는데 같은 목적지로 한번 이상의 전송이 있는 경우에는 이미 상대방의 MAC주소를 저장하고 있기 때문에 최초 한번만 ARP과정을 수행한다. 그러나 목적지가 계속 바뀌는 경우 매번 ARP과정을 수행하게 된다. 이때 상대방의 MAC주소를 알고 있는 경우에 한에서 ARP없이 Sn\_CR(MACSEND) 커맨드를 통해 바로 UDPPacket을 전송할 수 있게 된다. Sn\_CR(MACSEND) 수행전에 Sn\_DHAR을 통해 Destination MAC Address를 설정하고 Sn\_DIPR을 통해 Destination IP Address, Sn\_DPORTR을 통해서 Destination PORT를 설정한 다음에 전송할 Buffer와 Buffer 사이즈를 설정하고 Sn\_CR(MACSEND) 커맨드를 실행하면 ARP없이 UDPPacket이 전송된다.

|  |
| --- |
| {  START :  Sn\_DHAR0 = 0x00; /\* set Destination MAC address \*/  Sn\_DHAR1 = 0x08;  Sn\_DHAR2 = 0xDC;  Sn\_DHAR3 = 0x00;  Sn\_DHAR4 = 0x00;  Sn\_DHAR5 = 0x00;  Sn\_DIPR0 = 192; /\* set Multicast group IP address \*/  Sn\_DIPR1 = 168;  Sn\_DIPR2 = 0;  Sn\_DIPR3 = 100;  Sn\_DPORTR = 5000 /\* set Multicast group PORT number \*/  Sn\_CR = MACSEND;  } |

* + - 1. UDP Data Block

UDP 통신은 비연결성의 특징이 있기 때문에 자신의 IP와 PORT만 맞으면 모든 UDP Packet을 다 수신해야 하기 때문에 원하지 않는 Broadcast를 수신하거나 Multicast로만 SOCKET을 사용하지만 Unicast, Broadcast 들을 다 수신해서 Receive 처리의 지연으로 정작 받아야 할 Packet을 받지 못하는 경우가 생길 수도 있다.

그래서 W5100S는 몇 가지 UDP Block 기능이 있다. SOCKET을 Unicast 전송방식으로 Open했을 경우에는 Sn\_MR2(BCASTB) 을 설정하면 Broadcast 전송의 모든 UDP Packet을 받지 않는다. 또한 SOCKET을 Multicast로 Open한 경우에는 Sn\_MR2(BCASTB) 기능을 통해서 Broadcast 전송 Packet을 Block할 수도 있고 Sn\_MR2(UCASTB) 을 설정해 Unicast 전송으로 오는 UDP까지 Block시켜서 온전히 SOCKET이 가입한 그룹의 Multicast 전송 Packet만 수신하도록 설정할 수 있다.

|  |
| --- |
| {  UNICAST BLOCK :  Sn\_DHAR0 = 0x01; /\* set Multicast group MAC address \*/  Sn\_DHAR1 = 0x00;  Sn\_DHAR2 = 0x5E;  Sn\_DHAR3 = 0x00;  Sn\_DHAR4 = 0x00;  Sn\_DHAR5 = 0x64;  Sn\_DIPR0 = 224; /\* set Multicast group IP address \*/  Sn\_DIPR1 = 0;  Sn\_DIPR2 = 0;  Sn\_DIPR3 = 100;  Sn\_DPORTR = 0x1388 /\* set Multicast group PORT number \*/  Sn\_MR = UDP | Multicast;  Sn\_CR = OPEN;  if(Sn\_SR != SOCK\_UDP) Sn\_CR = CLOSE; goto UNICAST BLOCK;  Sn\_MR2 = UCASTB;  } |
| {  MULTICAST BLOCK :  Sn MR = UDP;  Sn\_CR = OPEN;  if(Sn\_SR != SOCK\_UDP) Sn\_CR = CLOSE; goto MULTICAST BLOCK;  Sn\_MR2 = MCASTB;  } |
| {  BROADCAST BLOCK :  Sn\_MR = UDP;  Sn\_CR = OPEN;  if(Sn\_SR != SOCK\_UDP) Sn\_CR = CLOSE; goto BROADCAST BLOCK;  Sn\_MR2 = BCASTB;  } |

* + - 1. Port Unreachable Block

UDP Packet은 수신할 상대방의 Port 번호를 가지고 있다. 그런데 만약 상대방이 해당 Port가 존재하지 않는다면 UDP Packet을 수신하지 않고 대신 Port Unreachable 이라는 Packet을 전송해 줌으로써 해당 Port가 없음을 알린다. Port Unreachable은 ICMP 프로토콜의 한 메시지 형태로 Type은 3이고 데이터에는 수신되지 못한 UDPPacket의 IP해더와 UDP해더가 포함된다. 하지만 이 port unreachable Packet의 원리를 이용하여 모든 port로 packet을 보내게 되면 해당 Host에서 제공하고 있는 서비스등을 확인하고 식별이 가능하게 된다.

이를 포트 스캔 공격이라고 하는데 W5100S는 이를 막기 위해 unreachable packet send를 Block 하는 기능이 있다. MR2 Register의 UDPURB bit를 1로 설정하면 없는 port에 Packet이 들어오더라도 unreachable packet을 보내지 않는다.

|  |
| --- |
| {  UNREACHABLE BLOCK :  MR2 = UDPURB;  } |

* 1. IPRAW

IPRAW는 Internet Layer를 이용한 DATA통신을 제공하며 SOCKET n은 모두 IPRAW Mode로 사용 가능하다. HOST는 SOCKET을 IPRAW Mode로 사용할 때, Sn\_PROTOR을 통해 IP Header의 Protocol Field 값을 설정함으로써 Internet Layer의 다양한 Protocol을 구현할 수 있다(Internet Protocol 에 관한 자세한 설명, [https://www.iana.org/assignments/protocol-numbers](https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml)). 또한 IPRAW Mode SOCKET은 Sn\_PROTOR로 설정된 Protocol의 Packet만 송수신 가능하다. 다음의 Table 3 은 지원가능 한 Internet Protocol을 나타낸다.

Table 3 Internet Protocol Supported In IPRAW Mode

|  |  |  |  |
| --- | --- | --- | --- |
| Protocol | Number | Semantic | W5100S Support |
| HOPOPT | 0 | IPv6 Hop-by-Hop Option | X |
| ICMP | 1 | Internet Control Message Protocol | O |
| IGMP | 2 | Internet Group Management | X |
| IPv4 | 4 | IPv4 encapsulation | O |
| TCP | 6 | Transmission Control | X |
| UDP | 17 | User Datagram | X |
| IPv6 | 41 | IPv6 encapsulation | X |
| others | - | Other Protocols | O |

아래의 Figure 9은 IPRAW Mode SOCKET의 동작흐름을 나타낸다.

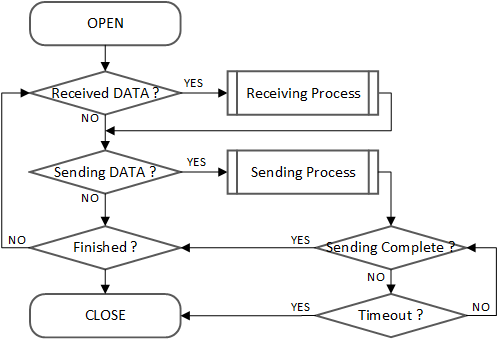


Figure 9 IPRAW Operation Flow

W5100S의 Initialization이 완료되면, PING Request 수신 시 자동으로 PING Reply를 전송한다. 그러나 IPRAW Mode SOCKET의 Sn\_PROTOR이 ICMP Protocol(0x01)로 설정되어 있으면, W5100S는 PING Request 수신 시 자동으로 PING Reply를 전송하지 않고 해당 IPRAW Mode SOCKET의 RX Buffer Block에 Request Packet을 저장한다. 이러한 경우, HOST가 직접 IPRAW Mode SOCKET을 이용해 PING Reply를 전송해야 한다.

* + 1. IPRAW Socket Open

HOST는 SOCKET n을 IPRAW Mode로 설정한다.

|  |
| --- |
| {  START :  /\* set Protocol number \*/  Sn\_PROTOR = protocol\_num;  /\* set IPRAW mode \*/  Sn\_MR = 4’b0011;    /\* set OPEN command \*/  Sn\_CR = OPEN;  /\* check SOCKET for IPRAW mode \*/  if(Sn\_SR != SOCK\_IPRAW) goto START;  } |

* + 1. IPRAW Data Send

IPRAW로 DATA 전송 시, 전송할 DATA가 Sn\_TXBUF\_SIZE나 1472 Byte보다 클 경우, HOST는 해당 DATA를 나누어 전송해야 한다. 또한 Sn\_CR[SEND\_MAC]을 통해 ARP과정없이 DATA를 전송할 수 있다. 이때, Sn\_DHAR에 상대방 MAC Address를 설정해야 한다.

• Sending DATA? / [SEND] Sending Process

|  |
| --- |
| {  START :  /\* set destination IP address, 192.168.0.100 \*/  Sn\_DIPR0,1 = 0xC0, 0xA8;  Sn\_DIPR2,3 = 0x00, 0x64;  /\* for using SEND\_MAC command \*/  /\* set destination MAC address, 74:D4:35:4F:BF:7C \*/  Sn\_DHAR0,1,2 = 0x74, 0xD4, 0x35;  Sn\_DHAR3,4,5 = 0x4F, 0xBF, 0x7C;  /\* check sending Data Size \*/  gSn\_TX\_MAX = Sn\_TXBUF\_SIZE;  If( (send\_size > gSn\_TX\_MAC) or (send\_size > 1472) ) goto START;  /\* calculate offset address \*/  get\_offset = Sn\_TX\_WR & gSn\_TX\_MASK;  /\* calculate start address(SOCKET0 TX Buffer Block address) \*/  get\_start\_address = gSn\_TX\_BASE + get\_offset;  /\* if overflow SOCKET0 TX Buffer Block \*/  If( (get\_offset + send\_size) > gSn\_TX\_MAX )  {  /\* copy upper size byte of source\_address to get\_start\_address \*/  upper\_size = gSn\_TX\_MAX – get\_offset;  memcpy(source\_address, get\_start\_address, upper\_size);  /\* copy left size byte of source\_address to gSn\_TX\_BASE \*/  source\_address += upper\_size;  left\_size = send\_size – upper\_size;  memcpy(source\_address, gSn\_TX\_BASE, left\_size);  }  else  {  /\* copy send size byte of source\_address to get\_start\_address \*/  memcpy(source\_address, get\_start\_address, send\_size);  }  /\* increase Sn\_TX\_WR as length of send size \*/  Sn\_TX\_WR += send\_size;  /\* set SEND (or SEND\_MAC) command \*/  Sn\_CR = SEND; goto Sending Complete?  } |

• Sending Complete?

SOCKET의 DATA 전송이 완료되면 Sn\_IR[SENDOK]가 발생한다.

|  |
| --- |
| {  /\* check SENDOK interrupt \*/  if(Sn\_IR[SENDOK] != ‘1’) goto Timeout;  else  {  /\* clear interrupt \*/  Sn\_IR[SENDOK] = ‘1’; goto Finished?;  /\* Sn\_IMR, Sn\_IR 참조 \*/  }  } |

• Timeout

UDP Mode SOCKET은 DATA 전송 전 ARP과정에서 상대방의 응답이 없는 경우 ARP Request를 재전송하며Sn\_RTR, Sn\_RCR에 따라 Sn\_IR[TIMEOUT]=’1’이 된다.

|  |
| --- |
| {  /\* check TIMEOUT interrupt \*/  if(Sn\_IR[TIMEOUT] != ‘1’) goto Sending Complete?  else goto CLOSE;  } |

• Finished? / CLOSE

IPRAW Mode SOCKET은 Sn\_IR[TIMEOUT] 또는 Sn\_CR[CLOSE]에 의해 CLOSE된다.

|  |
| --- |
| First method :  {  /\* check TIMEOUT interrupt \*/  if(Sn\_IR[TIMEOUT] == ‘1’)  {  /\* clear interrupt \*/  Sn\_IR[TIMEOUT] = ‘1’; goto CLOSE;  /\* Sn\_IMR, Sn\_IR 참조 \*/  /\* check SOCKET status \*/  if(Sn\_SR == SOCK\_CLOSED) SOCKET CLOSED;  }  } |
| Second method :  {  /\* set CLOSE command \*/  Sn\_CR = CLOSE;  /\* check SOCKET status \*/  if(Sn\_SR == SOCK\_CLOSED) SOCKET CLOSED;  } |

* + 1. IPRAW Data Receive

IPRAW Mode SOCKET에 수신된 Packet은 아래의 Figure 10과 같이 SOCKET RX Buffer Block에 저장된다. IPRAW DATA는 Packet에서 Ethernet Header, IP Header와 CRC를 제외한 나머지 DATA를 의미한다.

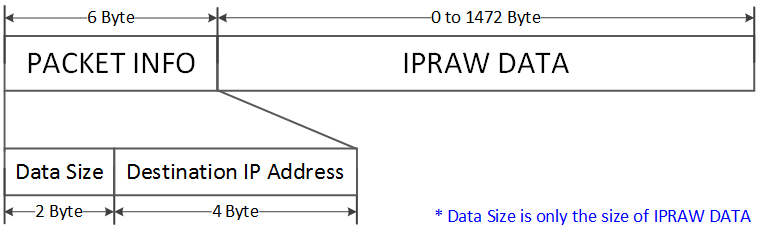


Figure 10 Received Data in IPRAW Mode SOCKET RX Buffer Block

• Received DATA?

|  |
| --- |
| {  First method :  /\* check SOCKET0 RX Buffer \*/  If( S0\_RX\_RSR != 0x0000 ) goto Receiving Process;  } |
| {  Second method :  /\* check SOCKET0 RECV Interrupt \*/  If( S0\_IR[RECV] == ‘1’ ) goto Receiving Process;  /\* IR, Sn\_IR, Sn\_IMR을 참조 \*/  } |

• Receiving Process

SOCKET RX Buffer Block으로부터 저장된 DATA를 읽은 후에 반드시 Sn\_CR[RECV]을 통해서 Sn\_RX\_RD를 RX Buffer Block에서 읽은 DATA 크기만큼 증가시킨다 (참조 Sn\_RX\_RD). 만약 Sn\_CR[RECV]시, SOCKET RX Buffer Block에 읽지 않은 DATA가 있는 경우 Sn\_IR[RECV]가 발생한다.

|  |
| --- |
| {  START :  /\* first, get received size \*/  get\_size = S0\_RX\_RSR;  /\* calculate offset address \*/  get\_offset = S0\_RX\_RD & gS0\_RX\_MASK;  /\* calculate start address(SOCKET0 RX Buffer Block Address) \*/  get\_start\_address = gS0\_RX\_BASE + get\_offset;  /\* if overflow SOCKET0 RX Buffer Block \*/  If( (get\_offset + get\_size) > gS0\_RX\_MAX )  {  /\* copy upper size bytes of get\_start\_address to destination\_address \*/  upper\_size = gS0\_RX\_MAX – get\_offset;  memcpy(get\_start\_address, destination\_ptr, upper\_size);    /\* copy left size bytes of gS0\_rx\_BASE to destination\_address \*/  left\_size = get\_size – upper\_size;  memcpy(gS0\_RX\_BASE, destination\_address, left\_size);  }  else  {  /\* copy get\_size of get\_start\_address to destination address \*/  memcpy(get\_start\_address, destination\_address, get\_size);  }  /\* increase S0\_RX\_RD as length of get\_size \*/  S0\_RX\_RD += get\_size;  /\* set RECV command \*/  S0\_CR = RECV;  } |

* 1. MACRAW

MACRAW는 Ethernet MAC을 이용한 DATA 통신을 제공하며 오직 SOCKET 0에서만 사용 가능하다. Ethernet MAC은 아래 Figure 11에서 보이는 것 처럼 Data Link Layer와 대응되며 하위 Physical layer를 통해서 네트워크 상의 주변 장치들 간 데이터를 전송한다. MACRAW는 Ethernet MAC을 사용해 데이터 전송을 하므로 아래 Figure 12와 같이 Ethernet MAC header인 6 Byte의 ‘Source hardware address’와 ‘Destination hardware address’ 그리고 2 Byte의 ‘Ethernet type’를 반드시 포함한 Ethernet Frame을 구현해야 한다.

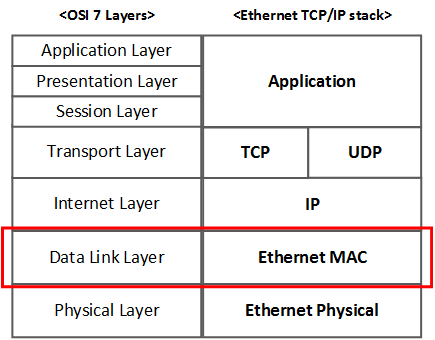


Figure 11 Ethernet TCP/IP stack with OSI 7 Layers

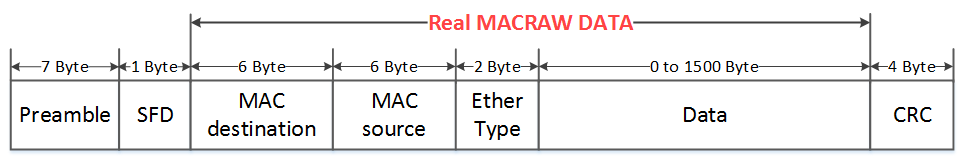


Figure 12 Ethernet Frame

* + 1. MACRAW Socket Open

MACRAW Mode는 SOCKET 0에서만 사용가능 하다. MACRAW Mode SOCKET OPEN시 S0\_MR[MF]를 설정하면 SHAR에 설정된 Hardware Address와 동일한 Destination MAC Address를 갖는 Packet과 Broadcast Packet, Multicast Packet만 수신한다. 또한 BLOCK 기능을 이용해 Broadcast, Multicast 또는 IPv6 Packet을 수신하지 않도록 선택적으로 설정할 수 있다.

• MACRAW SOCKET OPEN

|  |
| --- |
| {  START :  /\* set MACRAW mode \*/  S0\_MR = MACRAW;  /\* enable MAC Filter \*/  S0\_MR[MR] = ‘1’;  /\* enable MACRAW Block \*/  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;  /\* check SOCKET0 is MACRAW mode \*/  if(S0\_SR != SOCK\_MACRAW) S0\_CR = CLOSE; goto START;  } |

* + 1. MACRAW Data Receive

MACRAW Mode로 SOCKET0을 OPEN하면 W5100S의 Physical Layer를 통해 들어오는 Packet중 1516 Byte보다 작은 모든 Packet을 수신하며 S0\_RX\_RSR또는 S0\_IR[RECV]를 통해 수신여부를 확인할 수 있다. 수신된 Packet의 CRC가 맞으면 해당 SOCKET의 RX Buffer Block에 CRC를 제외한 Packet의 Data를 저장하는데 MACRAW Mode인 경우, Figure 13과 같이 CRC를 제외한 Packet의 길이를 나타내는 2 Byte의 Packet Info를 Data와 함께 RX Buffer Block에 저장한다.

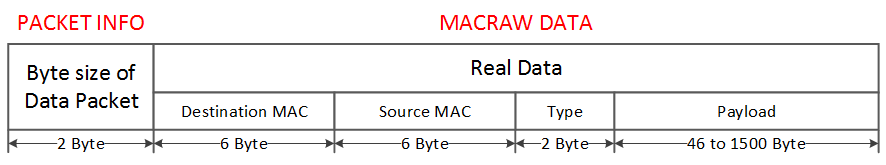


Figure 13 Received DATA Format in MACRAW

SOCKET RX Buffer Block으로부터 저장된 DATA를 읽은 후에 반드시 Sn\_CR[RECV]을 통해서 Sn\_RX\_RD를 RX Buffer Block에서 읽은 DATA 크기만큼 증가시킨다 (참조 Sn\_RX\_RD). 만약 Sn\_CR[RECV]시, SOCKET RX Buffer Block에 읽지 않은 DATA가 있는 경우 Sn\_IR[RECV]가 발생한다.

• Received DATA?

|  |
| --- |
| {  First method :  /\* check SOCKET0 RX Buffer \*/  If( S0\_RX\_RSR != 0x0000 ) goto Receiving Process;  } |
| {  Second method :  /\* check SOCKET0 RECV Interrupt \*/  If( S0\_IR[RECV] == ‘1’ ) goto Receiving Process;  /\* IR, Sn\_IR, Sn\_IMR을 참조 \*/  } |

• Receiving Process

|  |
| --- |
| {  START :  /\* first, get received size \*/  get\_size = S0\_RX\_RSR;  /\* calculate offset address \*/  get\_offset = S0\_RX\_RD & gS0\_RX\_MASK;  /\* calculate start address(SOCKET0 RX Buffer Block Address) \*/  get\_start\_address = gS0\_RX\_BASE + get\_offset;  /\* if overflow SOCKET0 RX Buffer Block \*/  If( (get\_offset + get\_size) > gS0\_RX\_MAX )  {  /\* copy upper size bytes of get\_start\_address to destination\_address \*/  upper\_size = gS0\_RX\_MAX – get\_offset;  memcpy(get\_start\_address, destination\_ptr, upper\_size);    /\* copy left size bytes of gS0\_rx\_BASE to destination\_address \*/  left\_size = get\_size – upper\_size;  memcpy(gS0\_RX\_BASE, destination\_address, left\_size);  }  else  {  /\* copy get\_size of get\_start\_address to destination address \*/  memcpy(get\_start\_address, destination\_address, get\_size);  }  /\* increase S0\_RX\_RD as length of get\_size \*/  S0\_RX\_RD += get\_size;  /\* set RECV command \*/  S0\_CR = RECV;  } |

* + 1. MACRAW Data Send

MACRAW Mode SOCKET0로 DATA 전송 시, 전송할 DATA가 S0\_TXBUF\_SIZE나 1514 Byte보다 큰 경우에 HOST는 DATA를 나누어 전송해야 한다. 그리고 전송 DATA가 60 byte보다 작은 경우 Zero Padding이 이루어진다. 전송이 완료되면 S0\_IR[SENDOK]가 발생한다.

• Sending Process

|  |
| --- |
| {  START :  /\* check Sending DATA Size \*/  get\_tx\_size = gS0\_TX\_MAX;  if(send\_size > get\_tx\_size) || (send\_size > 1516) goto START;  /\* calculate offset address \*/  get\_offset = S0\_TX\_WR & gS0\_TX\_MASK;  /\* calculate start address(SOCKET0 TX Buffer Block address) \*/  get\_start\_address = gS0\_TX\_BASE + get\_offset;  /\* if overflow SOCKET0 TX Buffer Block \*/  If( (get\_offset + send\_size) > gS0\_TX\_MAX )  {  /\* copy upper size byte of source\_address to get\_start\_address \*/  upper\_size = gS0\_TX\_MAX – get\_offset;  memcpy(source\_address, get\_start\_address, upper\_size);  /\* copy left size byte of source\_address to gS0\_TX\_BASE \*/  source\_address += upper\_size;  left\_size = send\_size – upper\_size;  memcpy(source\_address, gS0\_TX\_BASE, left\_size);  }  else  {  /\* copy send size byte of source\_address to get\_start\_address \*/  memcpy(source\_address, get\_start\_address, send\_size);  }  /\* increase Sn\_TX\_WR as length of send size \*/  S0\_TX\_WR += send\_size;  /\* set SEND command \*/  S0\_CR = SEND;  /\* check SENDOK interrupt \*/  while(S0\_IR[SENDOK] == ‘1’);  /\* clear interrupt \*/  S0\_IR[SENDOK] = ‘1’;  /\* Sn\_IMR, Sn\_IR 참조 \*/  } |

* + 1. MACRAW Optional Functions

MACRAW Mode는 MAC Filter외에도 특정 Packet(Broadcast, Multicast, IPv6)을 Block하는 기능을 지원한다. SOCKET0를 MACRAW Mode로 OPEN하기전에 S0\_MR2를 통해 해당 Block을 Enable할 수 있다.

(option function코드빼자)

MACRAW Mode has Packet Block functions for specific Packet, Broadcast, Multicast and IPv6.

This functions must be set by S0\_MR2 before MACRAW Mode SOCKET0 OPEN.

• Enable Broadcast Block

|  |
| --- |
| {  BROADCAST BLOCK :  /\* set MACRAW mode \*/  S0\_MR[3:0] = 4’b0100;  /\* Enable Broadcast Block \*/  S0\_MR2[BCSRBLK] = ‘1’;  /\* set OPEN command \*/  S0\_CR = OPEN;  /\* check SOCKET0 is MACRAW mode \*/  if(S0\_SR != MACRAW) S0\_CR = CLOSE; goto BROADCAST BLOCK;  } |

• Enable Multicast Block

|  |
| --- |
| {  MULTICAST BLOCK :  /\* set MACRAW mode \*/  S0\_MR[3:0] = 4’b0100;  /\* Enable Multicast Block \*/  S0\_MR2[MCSTBLK] = ‘1’;  /\* set OPEN command \*/  S0\_CR = OPEN;  /\* check SOCKET0 is MACRAW mode \*/  if(S0\_SR != MACRAW |

• Enable IPv6 Block

|  |
| --- |
| {  IPV6 BLOCK :  /\* set MACRAW mode \*/  S0\_MR = MACRAW;  /\* Enable IPv6 Block \*/  S0\_MR2 = IPV6BLK;  /\* set OPEN command \*/  S0\_CR = OPEN;  /\* check SOCKET0 is MACRAW mode \*/  if(S0\_SR != MACRAW) S0\_CR = CLOSE; goto MULTICAST BLOCK;  } |

* 1. SOCKET-less Command

W5100S는 SOCKET없이 ARP Request Packet 또는 PING Request Packet을 전송하는 SOCKET-less Commands를 지원한다. 전송한 Packet에 대한 응답이 없는 경우 SLRTR, SLRCR에 의한 TIMEOUT이 발생한다. 아래의 Figure 14는 SOCKET-less Commands의 동작흐름을 나타낸다.

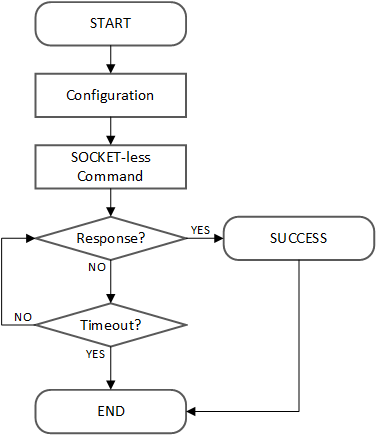


Figure . SOCKET-less Command Operation Flow

* + 1. ARP Command

ARP Command는 SLPIP로 ARP Request Packet을 전송한다. 전송한 ARP Request에 대한 ARP Reply를 수신하면 상대방의 MAC address는 SLPHAR에 저장된다.

• Configuration

|  |
| --- |
| {  START :  /\* set SOCKET-less retransmission timer, 10(0x000A)(단위, 100us) \*/  SLRTR0,1 = 0x00, 0xAA;  /\* set SOCKET-less retransmission counter, 5 \*/  SLRCR = 0x05;  /\* set destination IP address, 192.168.0.100 \*/  SLPIPR0,1 = 0xC0, 0xA8;  SLPIPR2,3 = 0x00, 0x64;  } |

• SOCKET-less Command

|  |
| --- |
| {  /\* set interrupt mask bit \*/  SLIMR[ARP] = ‘1’; // ARP interrupt mask bit  SLIMR[TIMEOUT] = ‘1’; // TIMEOUT interrupt mask bit  /\* set ARP command \*/  SLCR = ARP;  } |

• Response?

|  |
| --- |
| {  /\* check ARP interrupt \*/  if(SLIR[ARP] != ‘1’) goto Timeout  else  {  /\* received ARP reply packet \*/  /\* clear interrupt \*/  SLIR[ARP] = ‘1’; goto SUCCESS;  /\* SLIMR, SLIR 참조 \*/  }  } |

• Timeout?

|  |
| --- |
| {  /\* check TIMEOUT interrupt \*/  if(SLIR[TIMEOUT] != 1) goto Response;  else goto END;  } |

• SUCCESS

|  |
| --- |
| {  /\* get destination MAC address \*/  destination\_mac[0] = SLPHAR0;  destination\_mac[1] = SLPHAR1;  destination\_mac[2] = SLPHAR2;  destination\_mac[3] = SLPHAR3;  destination\_mac[4] = SLPHAR4;  destination\_mac[5] = SLPHAR5;  goto END;  } |

* + 1. PING Command

PING Command는 SLPIPR와 ARP 과정을 수행한 뒤 PING Request Packet을 전송한다.

PING Command를 통해 전송된 PING Request Packet에 대한 PING Reply Packet 수신 시 IPRAW Mode SOCKET의 Sn\_PROTOR이 ICMP Protocol(0x01)로 설정되어 있으면 PING Reply Packet은 해당 IPRAW Mode SOCKET의 RX Buffer Block에 저장된다. 따라서 PING Command 가 완료되지 못하고 SLIR[TIMEOUT]이 발생하므로 주의가 필요하다.

• Configuration

|  |
| --- |
| {  START :  /\* set SOCKET-less retransmission timer, 10(0x000A)(단위, 100us) \*/  SLRTR0,1 = 0x00, 0xAA;  /\* set SOCKET-less retransmission counter, 5 \*/  SLRCR = 0x05;  /\* set destination IP address, 192.168.0.100 \*/  SLPIPR0,1 = 0xC0, 0xA8;  SLPIPR2.3 = 0x00, 0x64;  /\* set PING sequence number, 1000(0x03E8) \*/  PINGSEQR0,1 = 0x03, 0xE8;  /\* set PING id, 256(0x0100) \*/  PINGIDR0,1 = 0x01,0x00;  } |

• SOCKET-less Command

|  |
| --- |
| {  /\* set interrupt mask bit \*/  SLIMR[PING] = ‘1’; // PING interrupt mask bit  SLIMR[TIMEOUT] = ‘1’; // TIMEOUT interrupt mask bit  /\* set PING command \*/  SLCR = PING;  } |

• Response?

|  |
| --- |
| {  /\* check PING interrupt \*/  if(SLIR[PING] != ‘1’) goto Timeout;  else  {  /\* received PING reply packet \*/  /\* clear interrupt \*/  SLIR[PING] = ‘1’; goto SUCCESS;  /\* SLIMR, SLIR 참조 \*/  }  } |

• Timeout?

|  |
| --- |
| {  /\* check TIMEOUT interrupt \*/  if(SLIR[TIMEOUT] != ‘1’) goto Response;  else goto END;  } |

* 1. PHY Control

W5100S는 Ethernet PHY를 내장하고 있으며, MDIO(Management Data Input/Output) Interface를 통해 PHY에 Access한다.

MDIO는 MDC(Management Data Clock)와 MDIO 두 개의 wire로 구성된 Serial Interface이며, W5100S의 MDC는 최대 2.5MHz로 동작하고 MDIO는 양방향 wire이다.

W5100S는 HOST의 편의를 위해 MDIO Interface외에 PHY를 Access할 수 있는 PHYCR0, PHYCR1 및 PHYSR Registers를 제공한다. Power Down Mode를 제외한 PHY의 Mode 설정은 PHY Reset 이 후 적용된다.

* + 1. PHY Access Register

PHYCR0는 PHY의 Speed 및 Duplex Mode를 설정하고 PHYCR1은 Reset, Power Down Mode를 설정한다. 또한 PHYSR로 PHY의 CABOFF, Link 및 Mode 설정을 확인할 수 있다.

PHYCR0, PHYCR1로 PHY에 Access하려면 먼저 PHYLCKR로 해당 Registers를 Unlock해야 한다.

Ex) PHYCR0을 이용한 PHY 10Mbps/Full-Duplex 설정 및 PHYSR을 이용한 PHY 상태확인

|  |
| --- |
| {  PHY\_10FDX :  /\* PHYCR Unlock \*/  PHYLCKR = Unlock;  /\* Set PHYCR0 10Mbps/Full-Duplex \*/  PHYCR0.DPX = FDX;  PHYCR0.SPD = 10 Mbps;  PHYCR0.AUTO = Manual;  /\* PHY Reset Process \*/  PHYCR1.RST = Reset;  Wait 400us; /\* At least time for PHY Reset \*/  PHYCR1.RST = Normal;  /\* Read PHYSR \*/  If( (PHYSR.DPX == FDX)&( PHYSR.SPD== 10Mbps)& (PHYSR.AUTO==Manual) )  PHYCFG DONE; |

Power Down mode가 적용되면 SYSCLK(System Operation Clock)은 25MHz로 설정 되고 해제되면 MR2[CLKSEL]에서 선택한 Clock으로 설정된다.

만약, Power Down mode가 해제되어 SYSCLK이 25MHz에서 100MHz로 설정되는 경우, Clock 안정화까지 210us이상 보장되어야 한다.

Ex) PHYCR1을 이용한 PHY Power Down mode 설정

|  |
| --- |
| {  POWER DOWN :  /\* Enable Power Down Mode \*/  PHYCR1.PWRDOWN = Enable;  /\* Disable Power Down Mode \*/  PHYCR1.PWRDOWN = Disable;  /\* clock 안정화 \*/  Wait 210us;  } |

* + 1. MDC/MDIO Interface

MDIO Interface는 PHYRAR, PHYDOR, PHYDIR과 PHYACR로 설정된다.

PHYRAR은 MDIO Interface에 PHY 내부 Register 주소를 설정하고 PHYDOR은 Read Data, PHYDIR은 Write Data를 저장한다. 그리고, PHYACR은 Read/Write를 설정하고 수행한다.

PHY Access가 완료되면 PHYACR은 ‘0’으로 설정된다.

Ex) MDIO Interface를 통한 PHY Write Access

|  |
| --- |
| {  MDIO\_WRITE :  /\* Set MDIO Interface \*/  PHYRR = PHY register;  PHYDIR = Data;  /\* PHY Access \*/  PHYACR = Write;  If(PHYACR == 0) /\* PHY Access SUCC \*/  {  /\* PHY Reset Process \*/  PHYCR1.RST = Reset;  Wait 400us; /\* At least time for PHY Reset \*/  PHYCR1.RST = Normal;  MDIO DONE;  }  else goto MDIO\_START;  } |

Ex) MDIO Interface를 통한 PHY Read Access

|  |
| --- |
| {  MDIO READ :  /\* Set MDIO Interface \*/  PHYRR = PHY register;  /\* PHY Access \*/  PHYACR = Write;  If(PHYACR = 0) /\* PHY Access SUCC \*/  DATA\_BUF = PHYDOR; /\* Read Data from PHY register \*/  else  goto MDIO\_READ;  } |

1. External Interface

W5100S는 Host를 위해 SPI, Parallel Bus 2가지의 Interface를 지원하고 있다.

MOD[3:0]핀의 상태의 따라 Interface Mode가 설정 된다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | MOD[3] | MOD[2] | MOD[1] | MOD[0] |
| SPI Bus Mode | 0 | 0 | 0 | 0 |
| Parallel Bus Mode | 0 | 1 | 0 | X |

Table 4 Interface Mode Select

또한 PHY control을 위한 Interface로 MDC/MDIO와 MII Interface를 지원한다.

* 1. SPI Bus Mode

W5100S는 Host Interface를 위해 SPI(Serial Peripheral Interface) Bus를 제공하며, 오직 SPI Slave로만 동작한다. W5100S는 SPI Bus를 위해 4개의 Signals(SCSn, SCLK, MOSI, MISO)를 제공하고 있으며 W5100S의 SPI Bus는 Host와 Figure 14과 Figure 15와 같이 2가지 방식으로 연결될 수 있다.

Figure 14방식은 Host가 SPI Bus를 다른 SPI Device들과 공유할 수 있는 반면, Figure 15방식은 Host의 SPI Bus가 W5100S전용으로 사용되어 다른 SPI Device들과 공유할 수 없다.

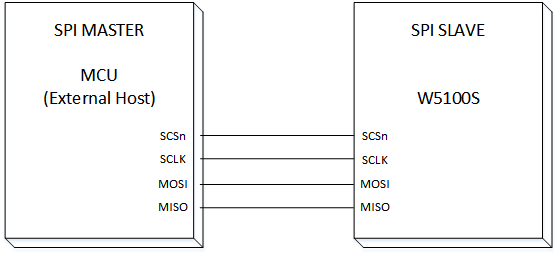


Figure 17 SCSn controlled by Host

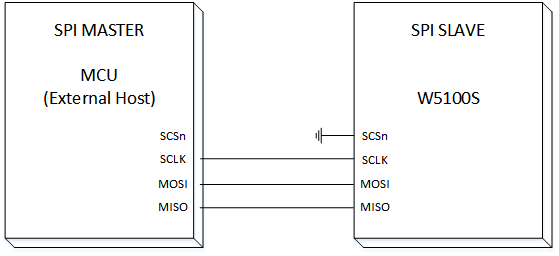


Figure 18 SCSn is always connected by Ground

SPI는 SCLK의 극성(Polarity)과 위상(Phase)에 따라 지원하는 4개의 Mode (0,1,2,3)가 있는데 W5100S의 SPI Bus의 경우 Mode 0과3을 지원한다. Figure 16에서와 같이 SPI Mode 0과 3은 inactive(idle)일 때 SCLK의 극성이 다르며 active시 동일하게 동작하며 Data signals(MPSI & MISO)는 SCLK의 Rising edge때 수신(Sampling)되며, Falling edge일 때 송신(Toggling)된다.



Figure 19 SPI Mode 0 & 3

W5100S는 Host로부터 송신되는 SPI Frame에 의해 제어되는데 SPI Frame은 Control Phase, Address Phase, Data Phase의 순서대로 이루어져있다.

Control Phase에서는 8bits로 Read/Write Access를 지정하고 Address Phase에서는 W5100S의 Register나 TX/RX Memory에 대한 16bits Offset Address를 지정한다. W5100S의 SPI 는 Sequential Data Read/Write를 지원하고 있기 때문에 Data Phase에서는 byte단위의 N개의 Data를 지정한다.

* + 1. SPI Frame

W5100S SPI Frame은 Figure 17와 같이 Control byte, 16bits Offset Address, N bytes의 Data로 구성된다.

W5100S는 Sequential Data Read/Write를 지원하며 N bytes의 Sequential Data 처리를 위해 지정된 Offset Address를 시작으로 Data를 처리(송수신)한 후 자동으로 Offset Address를 1씩 증가(Auto Increment Addressing)시켜 다음 Data를 처리한다.

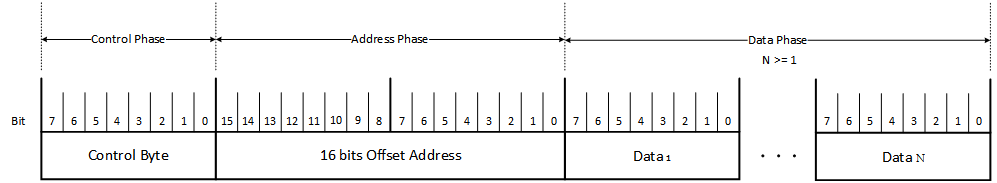


Figure 20 W5100 Mode SPI Frame

* + 1. Control Phase

Control Phase에서는 8bits로 Read/Write를 지정한다. Read Operation을 위한 값은 0x0F 이고, Write Operation을 위한 값은 0xF0 이다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Operation | Address Phase | Control Phase | | Data Phase |
| Write | 2 bytes | 0xF0 | 1111 0000 | n bytes |
| Read | 2 bytes | 0x0F | 0000 1111 | n bytes |

Table 5 W5100 Mode SPI Command

* + 1. Address Phase

W5100S의 Registers, TX/RX Buffer Block에 대한 16bits Offset Address를 지정한다. 이때 16bit Offset Address 값은 MSB부터 LSB 순으로 순차적으로 송신된다.

또한 Sequential Data Read/Write를 지원하기 때문에 SPI Frame의 Offset Address를 시작으로 Data Phase의 매 1byte의 송수신마다 Offset Address가 자동으로 1씩 증가한다.

* + 1. Data Phase

W5100S SPI Bus는 SCSn으로 컨트롤되고 Sequential Data Read/Write가 지원되므로 Data Phase에서 N bytes의 송수신이 가능하다. 1 byte 데이터 송수신마다 Auto Increment Addressing을 통해 Offset Address가 증가하고 데이터는 MSB부터 LSB순으로 MOSI 또는 MISO 를 통해 순차적으로 송수신된다.

* + 1. SPI Write Access

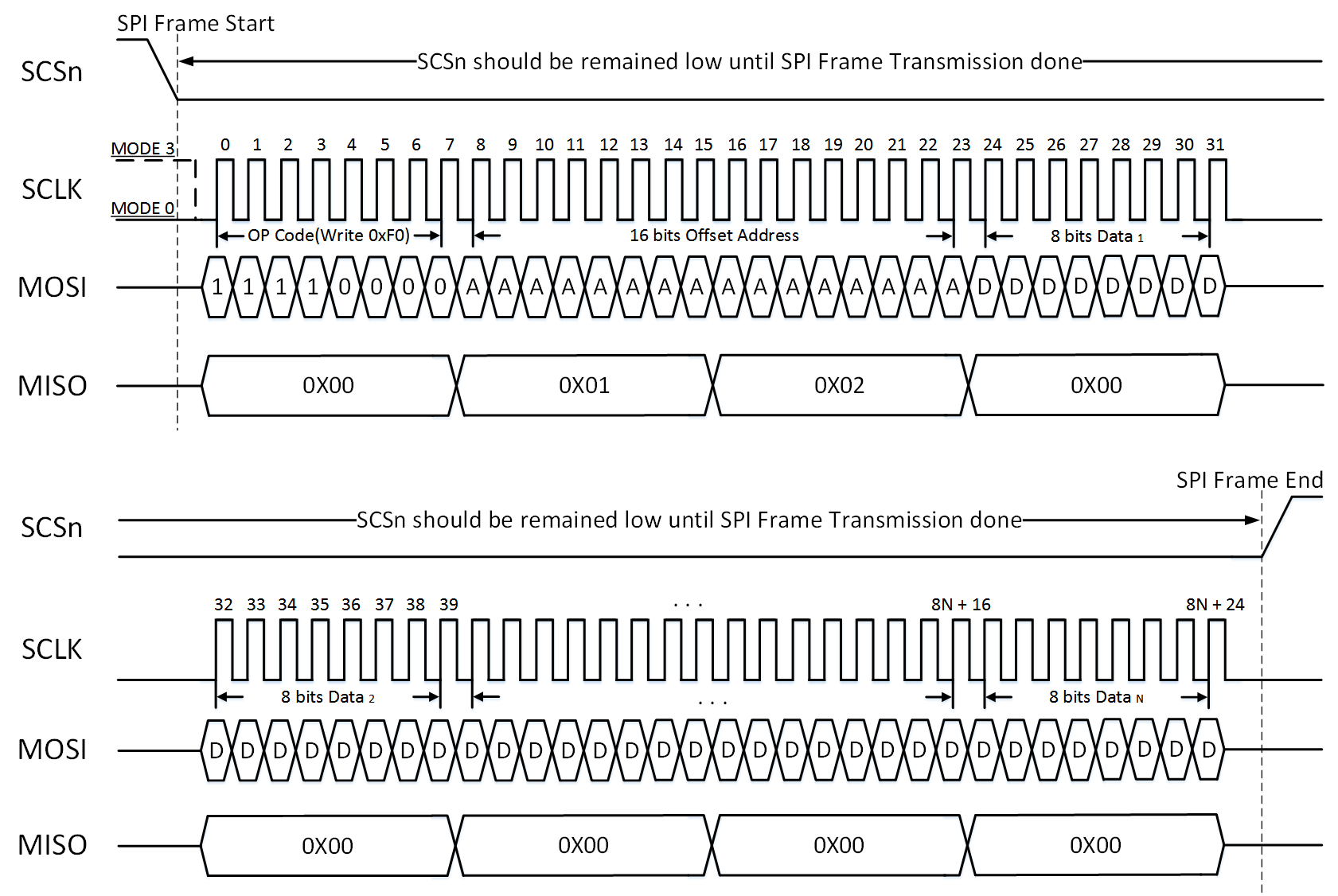


Figure 21 W5100 Mode Write SPI Frame

Figure 18는 Host가 W5100S를 Write Access 할 경우의 SPI Frame을 보여준다. Host의 SCSn signal을 SPI Frame 송신 시작 전에 Assert(High-to-Low)하고, SPI Frame의 모든 bits를 MOSI signal을 통해 Control byte(0xF0) & 16bits Offset Address & Data의 순서대로 Toggling SCLK(Falling-edge)에 동기화하여 1 bit씩 W5100S로 송신한다. 이때 SCSn이 계속 Low이고 송신할 데이터가 있는 경우 Sequential Data Write를 통해 Offset Address가 자동으로 1씩 증가하며 데이터가 전송된다. 송신이 완료된 후에는 SCSn signal을 De-Assert(Low-to-High)하여 Write Access를 끝마친다.

* + 1. SPI Read Access

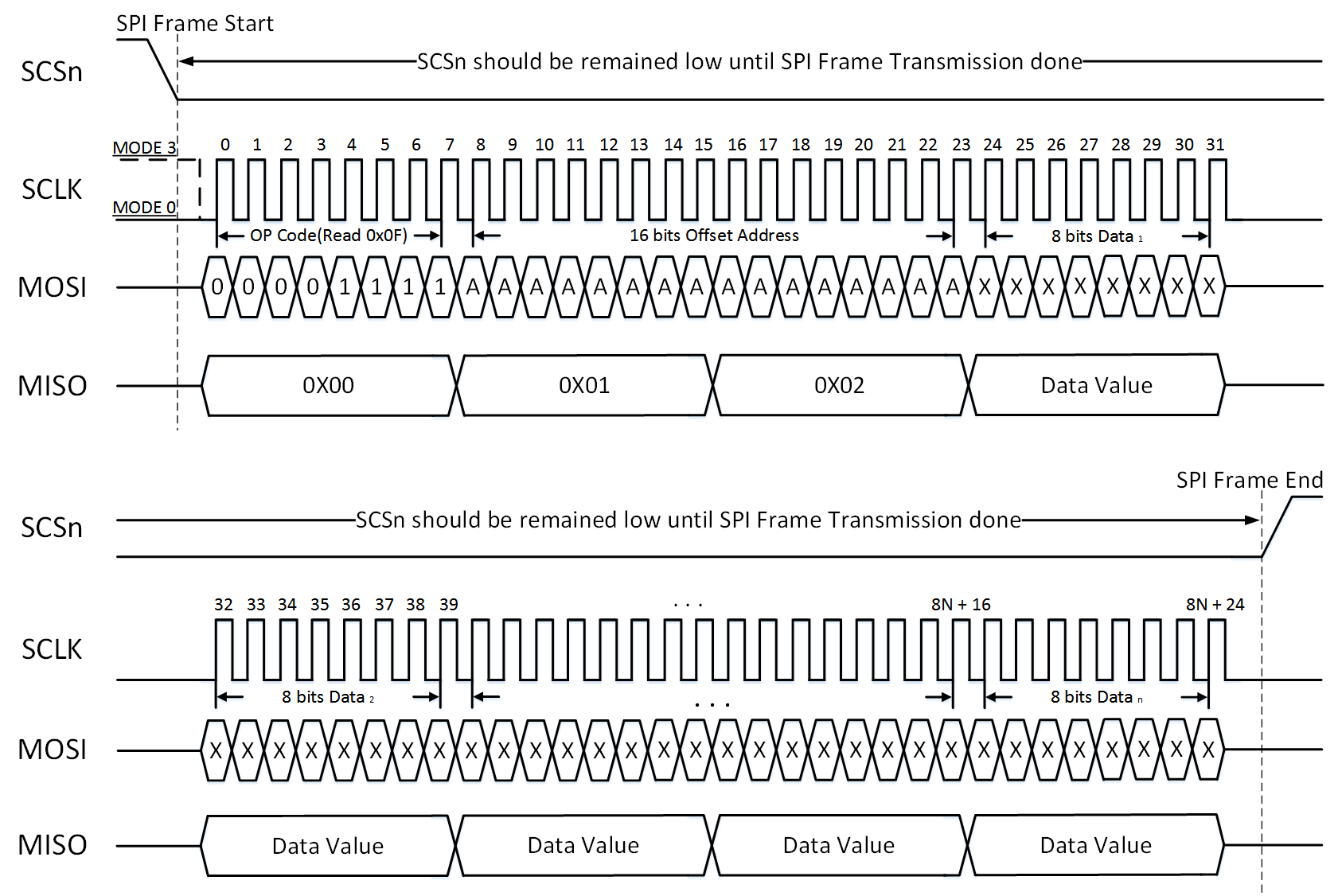


Figure 22 W5100 Mode Read SPI Frame

Figure 19은 Host가 W5100S를 Read Access할 경우의 SPI Frame을 보여준다. Host는 SPI Frame 송신 시작 전에 SCSn signal을 Assert(High-to-Low)하고, MOSI signal을 통해 Control byte(0x0F) & Offset Address & dummy data를 Toggling SCLK(Falling-edge)에 동기화해서 1 bit씩 송신하면, MISO signal을 통해 W5100S로부터 Offset Address Data의 bits를 수신한다. 이때 SCSn이 계속 Low이고 dummy data를 더 보내면 Sequential Data Read를 지원하기 때문에 계속해서 MISO를 통해 Data를 전송한다. Host는 Data Phase 수신이 완료된 후 SCSn signal을 De-Assert(Low-to-High)하여 Read Access를 끝마친다.

* 1. Parallel Bus Mode

Host는 Direct & Indirect Bus 인터페이스를 통해서 W5100S를 컨트롤 할 수 있다. /CS, /RD, /WR, /INT, 2bits Address, 8bits Data 라인을 가지고 송수신한다.

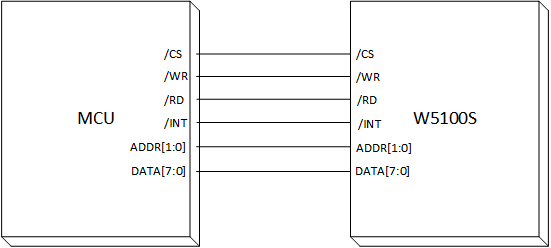


Figure Direct & Indirect Mode Control by Host

Table 7는 Indirect Bus에 관한 ADDR 값에 대한 설명을 나타낸다.

Table Indirect Mode Address Value

|  |  |  |
| --- | --- | --- |
| ADDR[1:0] | Symbol | Description |
| 00 | IDM\_MR | Common Register의 MR값을 Direct로 Read/Write한다.  MR의 자세한 내용은 Register Description을 참조. |
| 01 | IDM\_ARH | 상위 8 bits Address |
| 10 | IDM\_ARL | 하위 8 bits Address |
| 11 | IDM\_DR | Indirect Bus Mode의 8 bits Data Register |

* + 1. Direct Access

Drect Access는 유일하게 Common Register Block에 있는 MR에 Access를 가능하게 한다.

MR의 [0]번째 비트와 [1]번째 비트가 Indirect Access와 관련이 있다. 자세한 내용은 MR register Description을 참조하길 바란다.



Figure Direct Mode Write Access

MR에 0x03을 쓰고 싶다면 ADDR에 0 DATA에 0x03을 쓰면 된다.



Figure Direct Mode Read Access

MR을 읽고 싶다면 ADDR에 0을 쓰고 DATA를 읽으면 된다.

* + 1. Indirect Access

IDM\_ARH(Indirect Mode Address High Register)는 16 bits Offset Address에서 상위 8 bits Address를 지정하고 IDM\_ARL(Indirect Mode Address Low Register)는 16bits Offset Address에서 하위 8 bits Address를 지정한다. IDM\_DR(Indirect Mode Data Register)는 Read/Write Access할 8 bits Data를 지정한다.



Figure Indirect Mode Write Access

만약 IMR에 0x03을 쓰고 싶다면 ADDRH에 0x00 ADDRL에 0x16을 쓰고 DATA에 0x03을 쓰면 된다.

Figure Indirect Mode Continuous Write Access

만약 SIPR에 192.168.0.1을 쓰고 싶다면 MR에 Auto Increment bit가 enable된 상태에서 ADDRH에 0x00 ADDRL에 0x0F DATA에 0xC0, 0xA8, 0x0x, 0x01을 쓰면 주소가 자동으로 증가되면서 SIPR에 DATA가 써진다.



Figure Indirect Mode Read Access

만약 IR을 읽고 싶다면 ADDRH에 0x00 ADDRL에 0x15를 쓰고 DATA를 읽으면 된다.



Figure Indirect Mode Continuous Read Access

만약 SIPR을 읽고 싶다면 MR에 Auto Increment bit가 enable된 상태에서 ADDRH에 0x00 ADDRL에 0x0F를 쓰고 DATA를 4번 읽으면 주소가 자동으로 증가되면서 SIPR에 저장된 DATA가 읽어진다.

1. Clock
   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(CL) | ESR = 30 Ω |  | 12 |  | pF |
| Feedback Resistor(RF) | External resistor |  | 1M |  | Ω |
| Startup time | W5100S Reset |  |  | 60 | ms |
| Trans-conductance(gm) |  |  | 8.43 |  | mA/V |
| gain margin (gainmargin) | gainmargin = gm **/** gmcrit | 6.99 |  |  | dB |
| C0(1) : The Packaging Parasitic Shunt Capacitance.  CL(1) : Load Capacitance. eq) CL = (CL1 X CL2) / (CL1 X CL2) + Cs  CL1 , CL2 : External Capacitances of the circuit connected to the crystal (Typically, CL1 = CL2)  Cs : Stray Capacitance of printed circuit board and connections.  gmcrit : Oscillator loop critical gain. eq) gmcrit = 4 × (ESR + RExt)× (2πF)2 × (C0 + CL)2  ESR(1) : Maximal equivalent series resistance. eq) ESR = Rm X ( 1 + C0/CL)2  RExt : Resistor for limiting the drive level(DL) of the crystal.  *DL*(1) *:* The power dissipated in the crystal. Excess power can destroy the crystal.  RF(2): Feedback resistor. | | | | | |

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

\* Crystal 회로는 아래와 같은 형식으로 모델링된다.



Figure 30. Quartz Crystal Model

* + 1. Crystal recommendation characteristics.

Table 9. Crystal 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 |

* 1. Oscillator requirements.

Table 15. 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 |  |

1.08 1.2 1.32

|  |  |
| --- | --- |
| **Parameter** | **Range** |
| Frequency | 25 MHz |
| Frequency Tolerance (at 25℃) | ±50 ppm |
| Operating volatage | Typ. 1.2V(1.08V ~ 1.32V) |
| Clock duty | ±55 |
| Aging (at 25℃) | ±3ppm / year Max |

1. Electrical Specification
   1. Absolute Maximum ratings

Table 10. Absolute Maximum ratings

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Rating** | **Unit** |
| VDD | DC Supply voltage | -0.5 to 3.63 | V |
| VIN | DC input voltage | -0.5 to 5.5 | V |
| VOUT | DC output voltage | -0.5 to 3.63 | V |
| IIN | DC input current | ±8 | mA |
| TOP | Operating temperature | -40 to +85 | °C |
| TSTG | Storage temperature | -65 to +150 | °C |

**\*COMMENT**: Stressing the device beyond the ‘Absolute Maximum Ratings’ may cause permanent damage.

* 1. Absolute Maximum ratings (Electrical Ssensitivity)

Table 11. Electro Static Discharge (ESD)

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

Table 12. Static Latchup

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

* 1. DC Characteristics

Table 13. DC Characteristics

(Test Condition: Ta = –40 to 85°C)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Test Condition** | **Min** | **Typ** | **Max** | **Unit** |
| VDD | Supply voltage | Apply VDD, AVDD | 2.97 | 3.3 | 3.63 | V |
| VIH | High level input voltage |  | 2.0 | - | - | V |
| VIL | Low level input voltage |  | - |  | 0.8 | V |
| VT+ | Schmitt trig Low to High Threshold point | All inputs except XI | 0.8 | 1.1 | - | V |
| VT- | Schmitt trig High to Low Threshold point | All inputs except XI | - | 1.6 | 2.0 | V |
| TJ | Junction temperature |  | -40 | 25 | 125 | °C |
| IL | Input Leakage Current |  |  | ±1 | ±10 | μA |
| RPU | Pull-up Resistor | SCSn, RSTn, PMODE[2:0] | 40 | 75 | 190 | Kohm |
| RPD | Pull-down Resistor | RSVD(Pin 23, Pin 38 ~ Pin 42) | 40 | 75 | 190 | Kohm |
| VOL | Low level output voltage | IOL = 8mA,  All outputs except XO |  |  | 0.4 | V |
| VOH | High level output voltage | IOH = 8mA,  All outputs except XO | 2.4 |  |  | V |
| IOL | Low level output Current | VOL = 0.4V, All outputs except XO | 8.6 | 13.9 | 18.9 | mA |
| IOH | High level output Current | VOH = 2.4V, All outputs except XO | 12.5 | 26.9 | 47.1 | mA |
| IDD1 | Supply Current  (Normal operation mode) | VDD=3.3V, AVDD=3.3V, Ta = 25°C |  | 132 |  | mA |
| IDD2 | Supply Current  (Power Down mode) | PHY Power Down mode, VDD=3.3V, AVDD=3.3V, Ta = 25°C |  | 13 |  | mA |

* 1. POWER DISSPATION

Table 14. Power Disspation

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Condition | Min | Typ | Max | Unit |
| 100M Link | - | 91 | 110 | mA |
| 실측 | - | 93 |  | mA |
| 10M Link | - | 150 | 170 | mA |
| 실측 | - | 21 |  | mA |
| 100M Unlink(실측) | - | 45 |  | mA |
| 10M Unlink(실측) | - | 17 |  | mA |
| Un-Link (Auto-negotiation mode)(실측) | - | 43 | - | mA |
| Power Down mode | - | 31 | - | mA |
| 실측 | - | 17 | - | mA |

* 1. AC Characteristics
     1. Reset Timing

D:\temp\0708\3_W5500\Datasheet\RESET_TIMING.emf

Figure 31. Reset Timing

Table 15 Reset Timing

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Description** | **Min** | **Max** |
| TRC | Reset Cycle Time | 0.5 us | - |
| TPL | RSTn to System clock stable | - | 60.2 ms |

* + 1. Wake up Time

Voltage Regulator Wake Up Time: 10us

* + 1. SPI Timing

D:\1_project\100_W5500\0_document\W5500용_그림\SPI_MODE_TIME.emf

Figure 32. SPI Timing

Table 16. SPI Timing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Symbol** | **Description** | **Min** | **Max** | **Units** |
| FSCK | SCK Clock Frequency |  | 80/33.3[[1]](#footnote-2) | MHz |
| TWH | SCK High Time | 6 |  | ns |
| TWL | SCK Low Time | 6 |  | ns |
| TCS | SCSn High Time | 30 |  | ns |
| TCSS | SCSn Setup Time | 5 | - | ns |
| TCSH | SCSn Hold Time | 5 |  | ns |
| TDS | Data In Setup Time | 3 |  | ns |
| TDH | Data In Hold Time | 3 |  | ns |
| TOV | Output Valid Time |  | 5 | ns |
| TOH | Output Hold Time | 0 |  | ns |
| TCHZ | SCSn High to Output Hi-Z |  | 2.1[[2]](#footnote-3) | ns |

* + 1. Transformer Characteristics

Table 17. Transformer Characteristics

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Transmit End** | **Receive End** |
| Turn Ratio | 1:1 | 1:1 |
| Inductance | 350 uH | 350 uH |

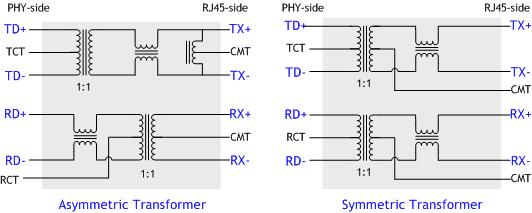


Figure 33. Transformer Type

* + 1. MDIX

W5100S는 Auto-MDIX를 지원한다.

1. IR Reflow Temperature Progile (Lead-Free)

Moisture Sensitivity Level : 3

Dry Pack Required : Yes

Table 18. IR Reflow Temperature Profile

|  |  |
| --- | --- |
| Average Ramp-Up Rate  (Tsmax to Tp) | 3° C/second max. |
| Preheat  – Temperature Min (Tsmin)  – Temperature Max (Tsmax)  – Time (tsmin to tsmax) | 150 °C  200 °C  60-120 seconds |
| Time maintained above:  – Temperature (TL)  – Time (tL) | 217 °C  60-150 seconds |
| Peak/Classification Temperature (Tp) | 265 + 0/-5°C |
| Time within 5 °C of actual Peak Temperature (tp) | 30 seconds |
| Ramp-Down Rate | 6 °C/second max. |
| Time 25 °C to Peak Temperature | 8 minutes max. |

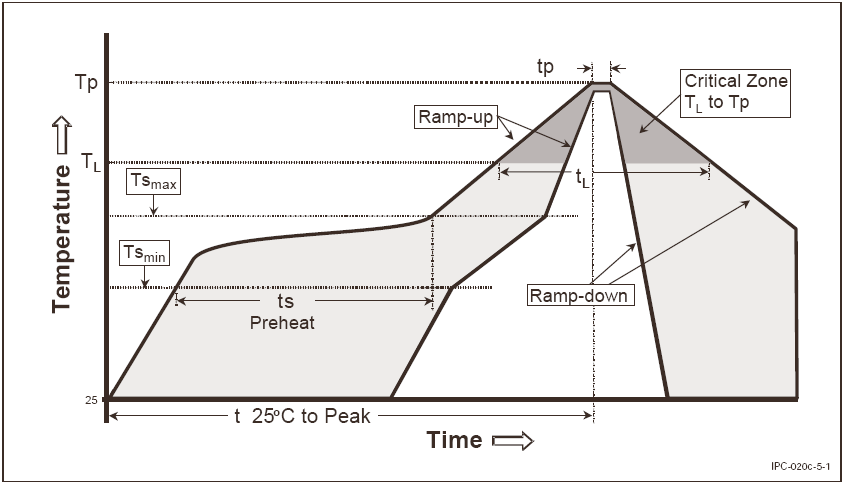


Figure 34. IR Reflow Temperature

1. Package Description

Note

1. These dimensions do not include mold protrusion.

2. ( ) is reference.

3. [ ] is ass’y out quality.

4. UNIT: mm

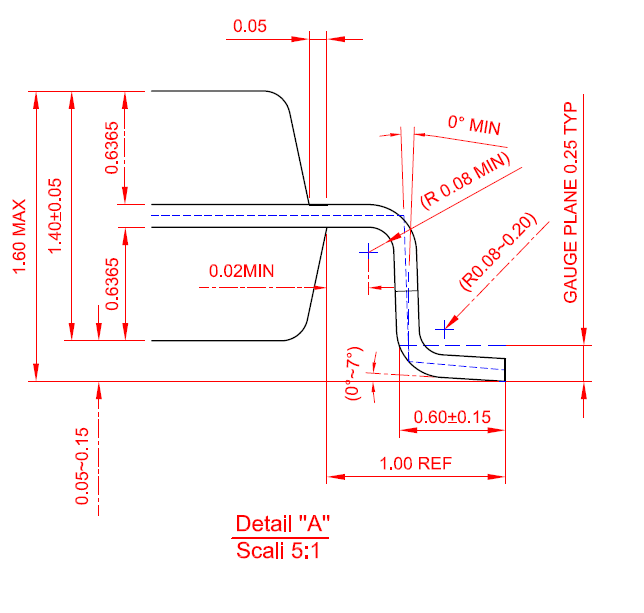
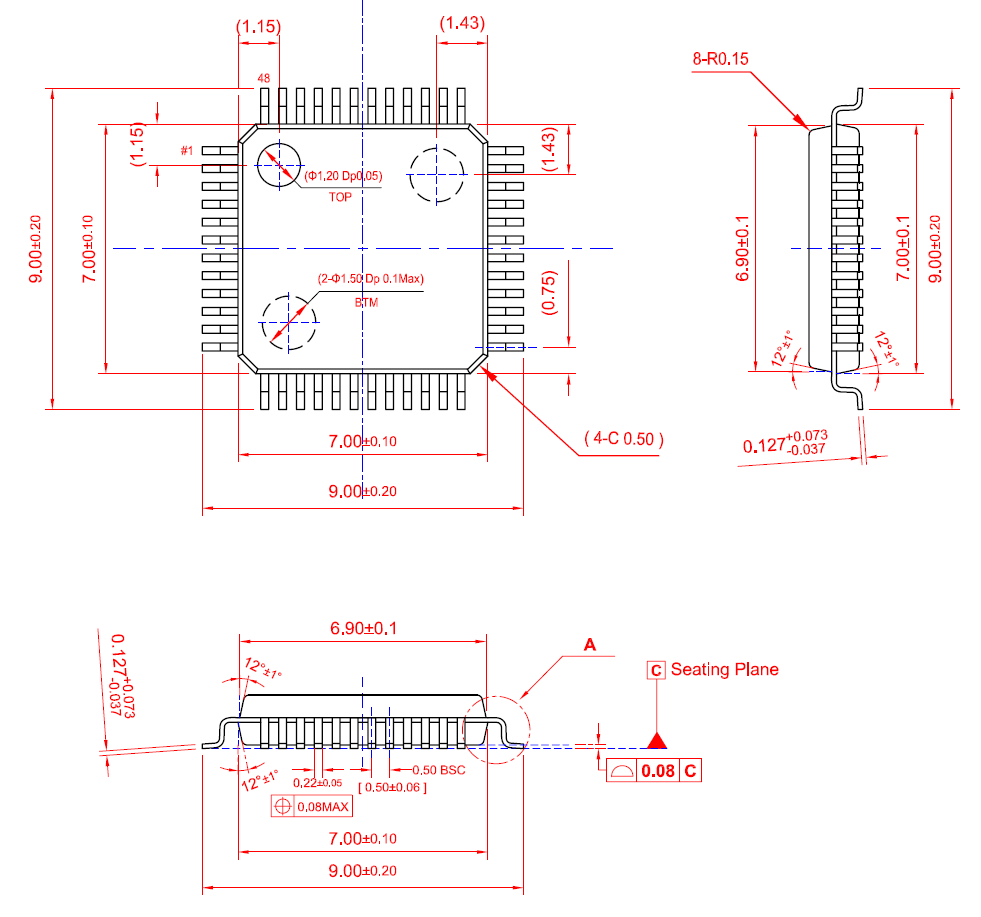


Figure 35. Package Dimensions

Document History Information

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Descriptions** |
| Ver. 1.0 | 30MAY2017 | Initial Release |
|  |  |  |

1. Theoretical Guaranteed Speed

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

   Please refer to the SPI Application Note which shows the WIZnet test environment and results. [↑](#footnote-ref-2)
2. 2.1ns is when pn loaded with 30pF. The time is shorter with lower capacitance. [↑](#footnote-ref-3)