Basic Terminologies in Networking: -

UDP (User Datagram Protocol): A connectionless, lightweight transport layer protocol that prioritizes speed over reliability, making it suitable for time-sensitive applications like streaming media and online gaming where some packet loss is acceptable.

TCP (Transmission Control Protocol): A connection-oriented, reliable transport layer protocol that ensures the ordered and error-checked delivery of data by establishing and maintaining a connection between a sender and receiver.

ICMP (Internet Control Message Protocol): A network layer protocol used by devices to send error messages and operational information, most famously used by the `ping` and `traceroute` utilities to diagnose network issues.

IGMP (Internet Group Management Protocol): A network protocol used by hosts and routers to establish and manage multicast group memberships on an IPv4 network, allowing data to be efficiently sent to multiple recipients simultaneously.

ARP (Address Resolution Protocol): A protocol used in local area networks to map logical IP address to physical MAC address, enabling communication between devices.

PPPoE (Point-to-Point Protocol over Ethernet): A network protocol that encapsulates PPP frames inside Ethernet frames, most often used by internet service providers for authentication, bandwidth control, and accounting fibre networks. LCP refers to **Link Control Protocol**. It is a component of the Point-to-Point Protocol (PPP) used to manage the link connection, including authentication, configuration, and termination.

Ping: A network administration utility that sends an ICMP Echo Request to a destination IP address to test connectivity and measure the round-trip time (latency) of the connection.

MII (Media-Independent Interface): A parallel interface standard that connects a MAC (Media Access Controller) to a PHY (Physical Layer) chip for 10/100 Mbps Ethernet data transfer using a 4-bit data bus and separate transmit and receive clocks.

RMII (Reduced Media-Independent Interface): A simplified version of the MII interface that cuts the pin count by using a 2-bit data bus and a single, shared 50 MHz clock for both transmit and receive paths.

CSMA/CD (Carrier Sense Multiple Access with Collision Detection): A media access control protocol, formerly used by half-duplex Ethernet, that ensures only one device transmits at a time and provides a method for detecting and recovering from data collisions.

Magic packet: A special type of broadcast frame sent over a network to wake up a computer that is in a low-power "sleep" mode, a feature known as *Wake-on-LAN (WoL).*

PHY (Physical Layer Transceiver): An electronic circuit, typically an integrated circuit, that implements the physical layer of the OSI model, converting digital data from a MAC into analog signals for transmission over the physical medium and vice-versa.

Ethernet transformer: A component within an Ethernet port that provides electrical isolation between the digital PHY chip and the analog network cable, protecting against electrical surges and noise.

TTL (Time-to-Live): An 8-bit value in an IP packet header that defines the maximum number of networks hops a packet can take before it is discarded by a router, preventing indefinite routing loops.

ToS: Type of Service (ToS) field is a byte in the IPv4 packet header that was originally intended to indicate the desired handling of a packet to routers. Its purpose was to allow for different Quality of Service (QoS) priorities, such as for minimal delay, high throughput, or high reliability.

DHCP (Dynamic Host Configuration Protocol): A network protocol that automatically and dynamically assigns an IP address, subnet mask, and default gateway to your W5500 from a DHCP server on the network.

DNS (Domain Name System): The system that translates human-readable domain names (like `example.com`) into numerical IP addresses. The W5500 can be used to make DNS requests to connect to web servers by name.

Gateway: A network node or router that acts as an access point to another network. Your W5500 needs to know the gateway's IP address to send data to the internet or another network.

Subnet Mask: A numeric mask that determines which part of an IP address refers to the network and which part refers to the specific host (your W5500).

IP Fragmentation: The process of breaking large IP packets into smaller packets. The W5500's hardware TCP/IP stack does not support IP fragmentation, so your application must send packets within the Maximum Transmission Unit (MTU) to avoid errors.

MACRAW: (MAC Raw) is a special operating mode on the W5500 that allows your microcontroller to bypass the chip's built-in TCP/IP stack and handle raw Ethernet frames directly. In MACRAW mode, the W5500 functions more like a standard Ethernet MAC/PHY chip, providing your application with low-level access to network packets

Keep Alive: A keep-alive setting in networking is a mechanism used to maintain a connection between two devices even when there is no data being actively sent. It is crucial for preventing idle TCP connections from being terminated by network devices, such as firewalls or NAT routers, that have connection-tracking timeouts. The W5500's hardwired TCP stack includes a `Sn_KPALVTR` (Keep-Alive Time Register) that you can configure to enable this functionality.

Client: Your device initiates a connection to an external server.

Server: Your device listens for and accepts incoming connection requests from other clients.

Unicast: A type of network communication where data is sent from a single source to a single destination. Most client/server applications use unicast.

Multicast: A type of network communication where data is sent from a single source to a specific group of multiple destinations. IGMP is used to manage these group memberships.

Broadcast: A communication method where data is sent to every device on the local network simultaneously. The magic packet for Wake-on-LAN is a broadcast message.

Port: A virtual end point for communication on a network. The W5500 uses port numbers to distinguish between different services running on the same IP address (e.g., port 80 for HTTP, port 25 for SMTP).

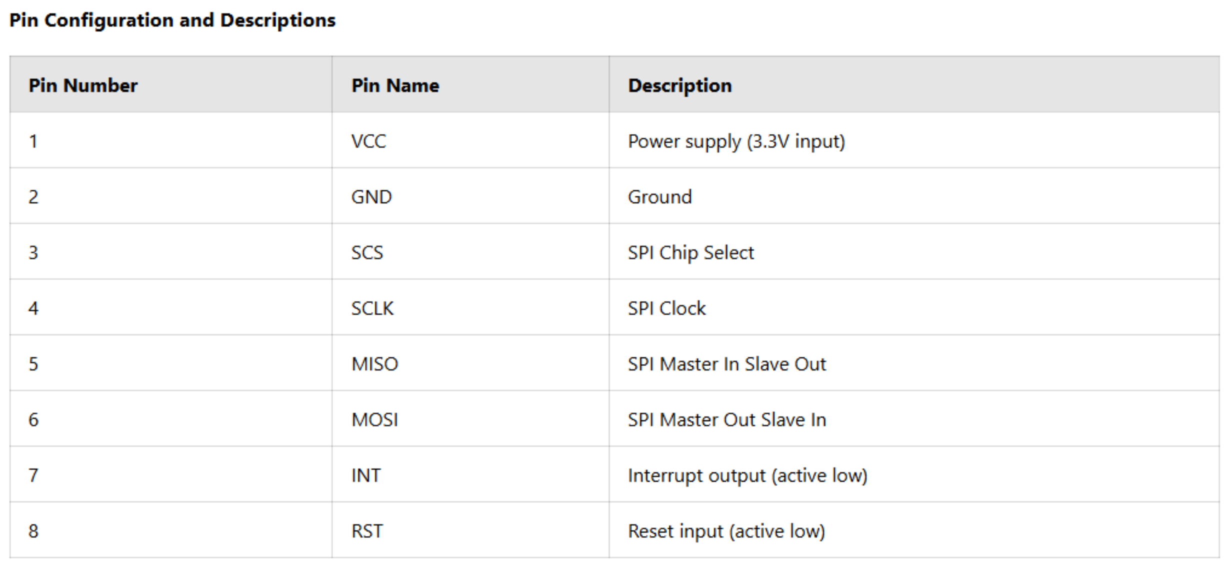
Socket: The W5500 abstracts communication using up to eight independent hardware sockets. Each socket is a software construct that represents a single network connection, handling all the underlying protocol details.

W5500 (W5500 TCP / IP *SPI to LAN* Ethernet Interface module)

Features

- Supports Hardwired TCP/IP Protocols: TCP, UDP, ICMP, IPv4, ARP etc.,
- Supports 8 independent sockets simultaneously - Supports Power down mode - Supports Wake on LAN over UDP.
- Supports High Speed Serial Peripheral Interface (SPI MODE 0, 3) of maximum SPI speed being 80Mhz.
- Internal 32Kbytes Memory for TX/RX Buffers - 10BaseT/100BaseTX Ethernet PHY embedded.
- 3.3V operation with 5V I/O signal tolerance.
- LED outputs (Full/Half duplex, Link, Speed, Active).

Pinouts (SPI to Ethernet Chip)



Pin Configuration and Descriptions

Pin Number	Pin Name	Description
1	VCC	Power supply (3.3V input)
2	GND	Ground
3	SCS	SPI Chip Select
4	SCLK	SPI Clock
5	MISO	SPI Master In Slave Out
6	MOSI	SPI Master Out Slave In
7	INT	Interrupt output (active low)
8	RST	Reset input (active low)

How to Use the Component in a Circuit

1. Connect the VCC pin to a 3.3V power source and GND to ground.
2. Interface the SPI pins (SCS, SCLK, MISO, MOSI) with your microcontroller's corresponding SPI pins.
3. Connect the RST pin to a digital output pin on your microcontroller for hardware reset control.
4. Use the INT pin if you require interrupt-driven programming.

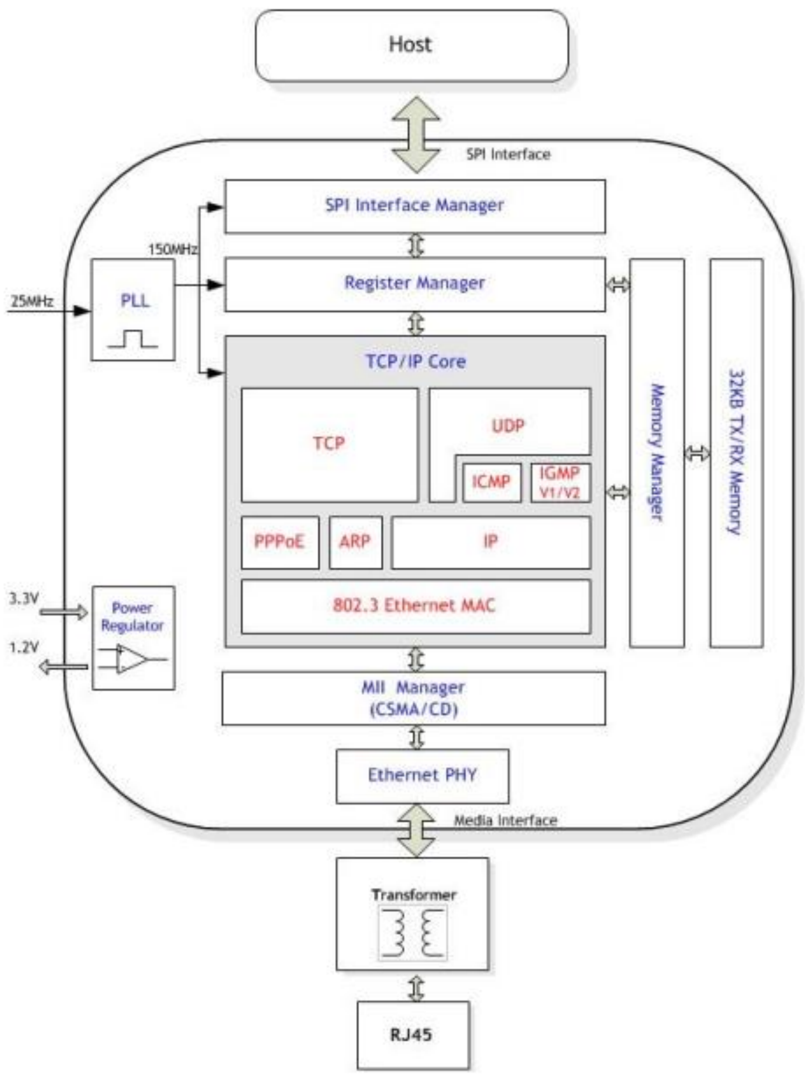
Note:

1. For 5V microcontrollers like the Arduino Uno, ensure your W5500 module has a built-in level shifter or use an external one. The W5500 chip itself has 5V-tolerant SPI pins, but check the module's datasheet to be certain.

2. The CS pin is a standard GPIO pin on your microcontroller and is not controlled by the hardware SPI peripheral. You must manually control it in your code.

3. The RST pin allows for a software-controlled hardware reset, which can resolve lockups.

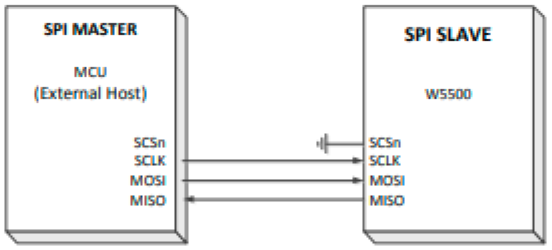
Block Diagram



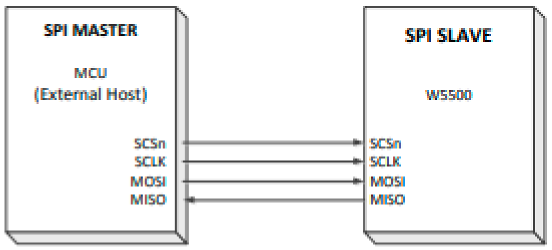
Interfacing SPI Protocol with the Ethernet

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

To use only the 3 pins in MCU without utilizing the SPI bus: -



To use the SPI bus and to use multiple SPI slaves simultaneously: -



SPI modes of operation

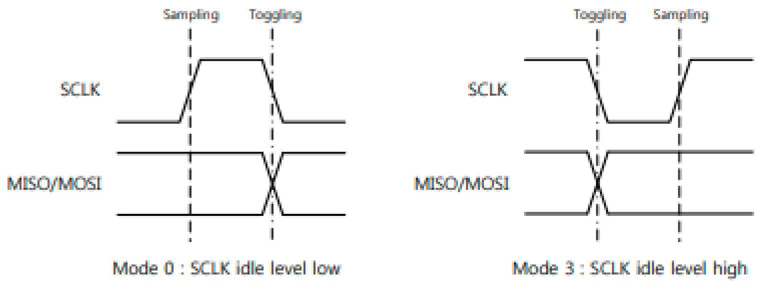


Figure 6. SPI Mode 0 & 3

Note:

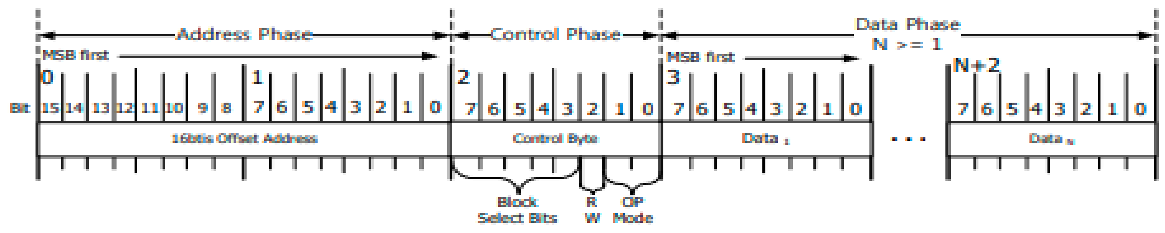
The data transfer direction is from MSB to LSB.

Data format for communication with W5500 using SPI Protocol

W5500 is controlled by SPI Frame which communicates with the External Host.

W5500 SPI Frame consists of 3 phases, Address Phase, Control Phase and Data Phase.

SPI Frame Format



Byte 1	Byte 2	Byte 3	Byte 4 and beyond
Address MSB	Address LSB	Control Byte	Data Phase

1) The microcontroller must first assert (pull low) the W5500's Chip Select (SCSn) pin to begin a communication session.
2) Then, send data in the following format from the MCU to the W5500 Chip.

   Byte 1 and 2: Address phase

   The first two bytes transmitted over SPI represent the 16-bit register address you want to access. The address space is split into different regions for common registers, socket registers, and TX/RX buffer memory.

   Byte 3: Control phase

   The operation we perform on the targeted addresses. There are three-bit fields in this byte which is as follows
   | BLOCK SELECT BYTE (Bits 7:3) | READ/WRITE (Bit 2) | OPERATION MODE (Bits 1:0)

   Block select byte defines which socket block I am concerned about.

   SPI Operation Mode supports two modes, the Variable Length Data Mode and the Fixed Length Data Mode. In fixed length, the size of data frame can be 1/2/4 bytes. In VLDM, the size of data frame depends on the CS pin assertion.

Byte 4: Data Phase

For a write operation, the data phase consists of one or more bytes of data that you want to write to the specified address. For a read operation, the data phase consists of "dummy" bytes sent from the microcontroller to receive the data returned by the W5500.

- W5500: After the transaction is complete, the microcontroller de-asserts (pulls high) the `SCSn` pin to end the communication session.

Use SPI Drivers to write and read from W5500. Based on what we write, we will get back the response from the registers concerning the respective operation. W5500 is just a SPI Slave.

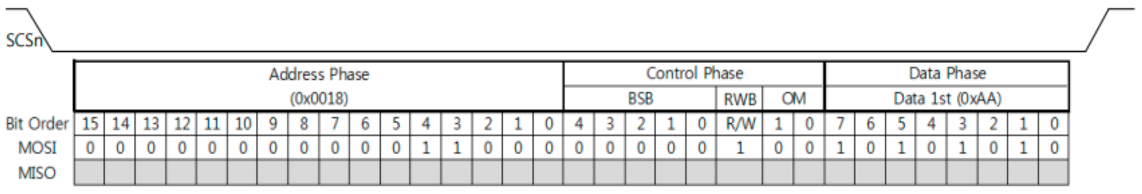
Variable Data Length Mode:

One Byte Write Format:

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

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

The External Host asserts (High-to-Low) SCSn before transmitting SPI Frame, then the Host transmits 1 bit with synchronizing the Toggle SCLK. The External Host de-asserts (Low-to-High) the SCSn at the end of SPI Frame transmit. (Refer to the Figure 9)



	Address Phase (0x0018)																Control Phase								Data Phase							
																	BSB					RWB	OM		Data 1st (0xAA)							
Bit Order	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	4	3	2	1	0	R/W	1	0	7	6	5	4	3	2	1	0
MOSI	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	1	0	1	0
MISO																																

Multiple Bytes Write frame format:

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

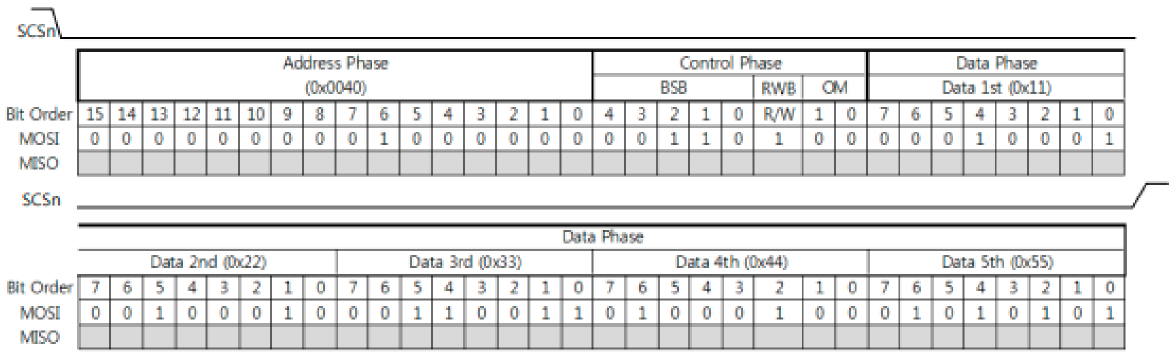
```
Offset Address = 0x0040
BSB[4:0]        = '00110'
RWB            =  '1'
OM[1:0]         = '00'
1st Data        = 0x11
2nd Data        = 0x22
3rd Data        = 0x33
4th Data        = 0x44
5th Data        = 0x55
```

The N-Bytes Write Access is shown in Figure 10.

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

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

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



	Address Phase																Control Phase								Data Phase							
	(0x0040)																BSB					RWB	OM		Data 1st (0x11)							
Bit Order	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	4	3	2	1	0	R/W	1	0	7	6	5	4	3	2	1	0
MOSI	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	1	0	0	0	1
MISO																																

	Data Phase																															
	Data 2nd (0x22)								Data 3rd (0x33)								Data 4th (0x44)								Data 5th (0x55)							
Bit Order	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
MOSI	0	0	1	0	0	0	1	0	0	0	1	1	0	0	1	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	1
MISO																																

Figure 10. 5 Byte Data Write at 1th Socket's TX Buffer Block 0x0040 in VDM mode

Note:
The MISO line is inactive since it is completely a write operation from our MCU to the W5500 SPI Slave.

One Byte Read operation:

1 Byte READ Access Example

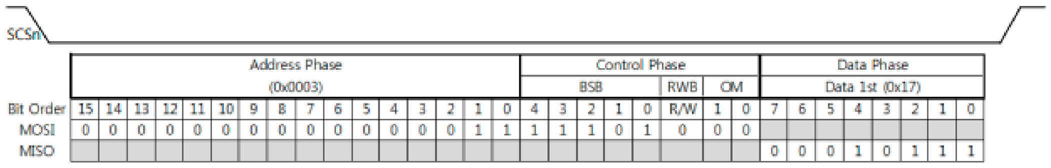
When the Host reads the ‘Socket Status Register(S7_SR) of the Socket 7’s Register Block by using VDM mode, the data is read with the SPI Frame below. Let’s S7_SR to ‘SOCK_ESTABLISHED (0x17)’.

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

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

Then the Host receives Data Phase from the MISO signal.

After finishing the Data Phase receives, the Host deasserts SCSn signal (Low-to-High). (Refer to the Figure 12.)



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

Figure 12. S7_SR Read in VDM Mode

Note:

The reading of SPI data from the SPI Slave is done on the MISO line. In the same format, multiple bytes are read from the SPI Slave.

Multiple Bytes Read Operation:

N-Bytes Read Access Example

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

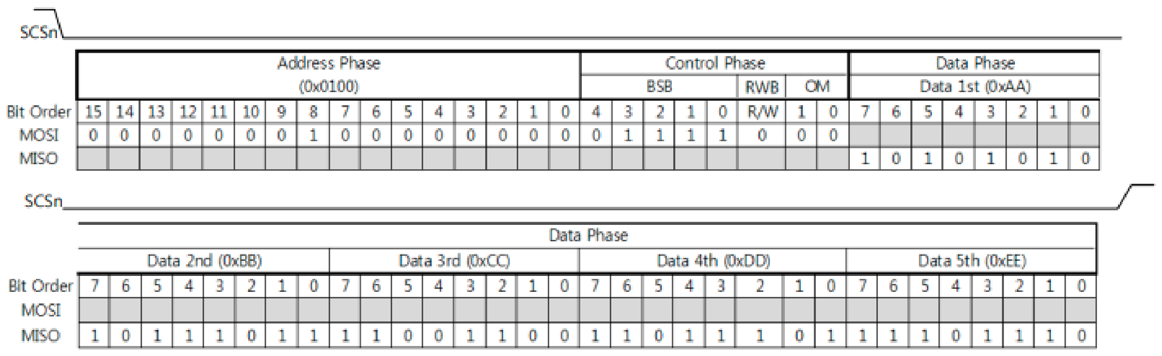
```
Offset Address = 0x0100
BSB[4:0]        = '01111'
RWB            = '0'
OM[1:0]         = '00'
1st Data        = 0xAA
2nd Data        = 0xBB
3rd Data        = 0xCC
4th Data        = 0xDD
5th Data        = 0xEE
```

The N-Bytes Read Access is shown in Figure 13.

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

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

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



	Address Phase (0x0100)																Control Phase								Data Phase							
																	BSB					RWB	OM		Data 1st (0xAA)							
Bit Order	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	4	3	2	1	0	R/W	1	0	7	6	5	4	3	2	1	0
MOSI	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0								
MISO																									1	0	1	0	1	0	1	0

	Data Phase																															
	Data 2nd (0xBB)								Data 3rd (0xCC)								Data 4th (0xDD)								Data 5th (0xEE)							
Bit Order	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
MOSI																																
MISO	1	0	1	1	1	0	1	1	1	1	0	0	1	1	0	0	1	1	0	1	1	1	0	1	1	1	1	0	1	1	1	0

Figure 13. 5 Byte Data Read at Socket 3 RX Buffer Block 0x0100 in VDM mode

Fixed Data Length Mode: (1 / 2 / 4 bytes):

2.4.1 Write Access in FDM

1 Bytes WRITE Access

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

Figure 14. 1 Byte Data Write SPI Frame in FDM mode

2 Bytes WRITE Access

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

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

Figure 15. 2 Bytes Data Write SPI Frame in FDM mode

2.4.2 Read Access in FDM

1 Byte READ Access

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

Figure 17. 1 Byte Data Read SPI Frame in FDM mode

2 Bytes READ Access

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

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

Figure 18. 2 Bytes Data Read SPI Frame in FDM mode

Memory and Register mapping:

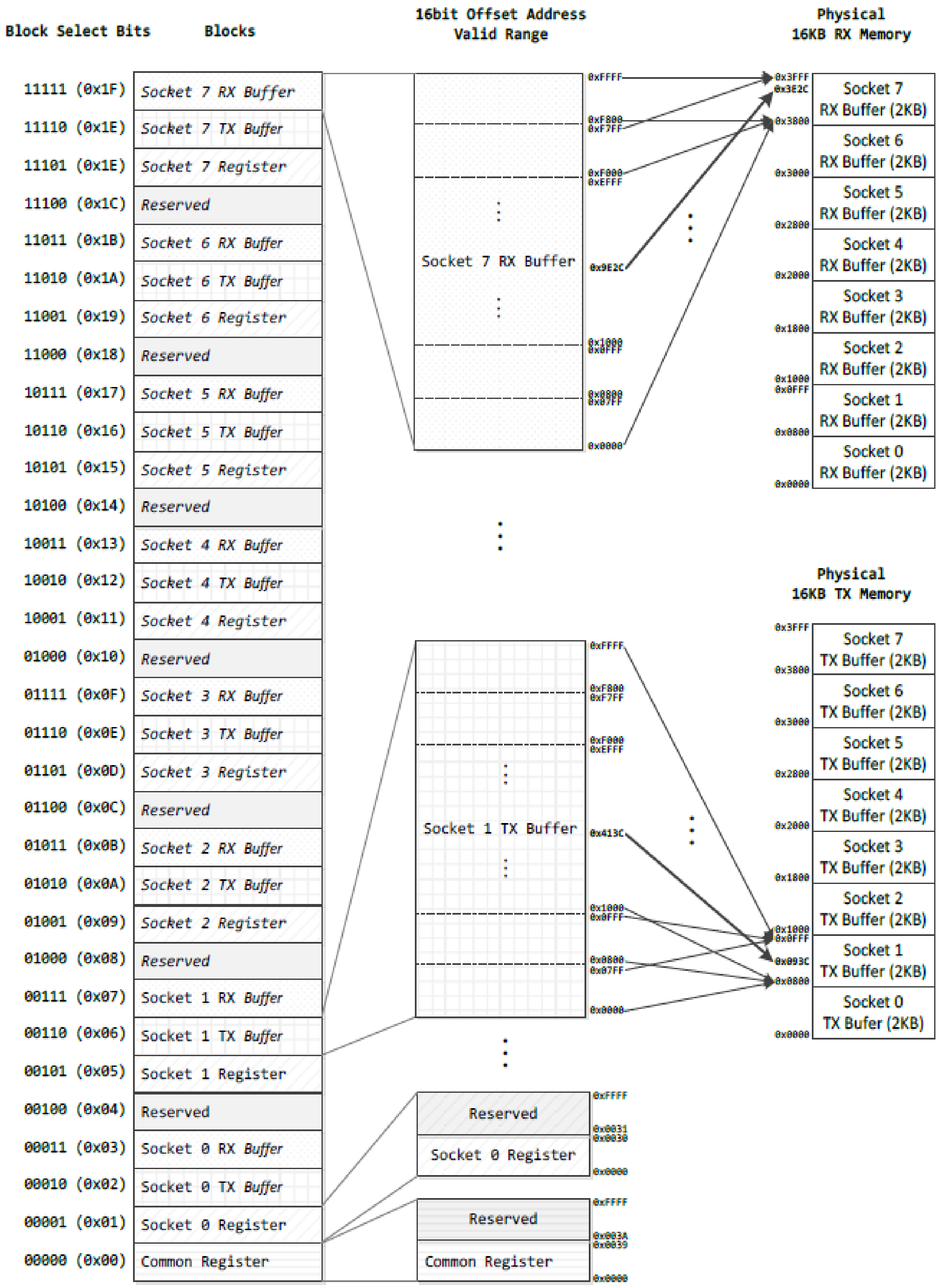


Figure 20. Register & Memory Organization

Socket Registers:

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

Offset	Register
0x0000	Socket n Mode (Sn_MR)
0x0001	Socket n Command (Sn_CR)
0x0002	Socket n Interrupt (Sn_IR)
0x0003	Socket n Status (Sn_SR)
0x0004 0x0005	Socket n Source Port (Sn_PORT0) (Sn_PORT1)
0x0006 0x0007 0x0008 0x0009 0x000A 0x000B	Socket n Destination Hardware Address (Sn_DHAR0) (Sn_DHAR1) (Sn_DHAR2) (Sn_DHAR3) (Sn_DHAR4) (Sn_DHAR5)
0x000C 0x000D 0x000E 0x000F	Socket n Destination IP Address (Sn_DIPR0) (Sn_DIPR1) (Sn_DIPR2) (Sn_DIPR3)
0x0010 0x0011	Socket n Destination Port (Sn_DPORT0) (Sn_DPORT1)
0x0012 0x0013	Socket n Maximum Segment Size (Sn_MSSR0) (Sn_MSSR1)
0x0014	Reserved
0x0015	Socket n IP TOS (Sn_TOS)
0x0016	Socket n IP TTL (Sn_TTL)
0x0017 ~ 0x001D	Reserved
0x001E	Socket n Receive Buffer Size (Sn_RXBUF_SIZE)
0x001F	Socket n Transmit Buffer Size (Sn_TXBUF_SIZE)
0x0020 0x0021	Socket n TX Free Size (Sn_TX_FSR0) (Sn_TX_FSR1)
0x0022 0x0023	Socket n TX Read Pointer (Sn_TX_RD0) (Sn_TX_RD1)
0x0024 0x0025	Socket n TX Write Pointer (Sn_TX_WR0) (Sn_TX_WR1)
0x0026 0x0027	Socket n RX Received Size (Sn_RX_RSR0) (Sn_RX_RSR1)
0x0028 0x0029	Socket n RX Read Pointer (Sn_RX_RD0) (Sn_RX_RD1)
0x002A 0x002B	Socket n RX Write Pointer (Sn_RX_WR0) (Sn_RX_WR1)
0x002C	Socket n Interrupt Mask (Sn_IMR)
0x002D 0x002E	Socket n Fragment Offset in IP header (Sn_FRAG0) (Sn_FRAG1)
0x002F	Keep alive timer (Sn_KPALVTR)
0x0030 ~ 0xFFFF	Reserved

Common Registers:

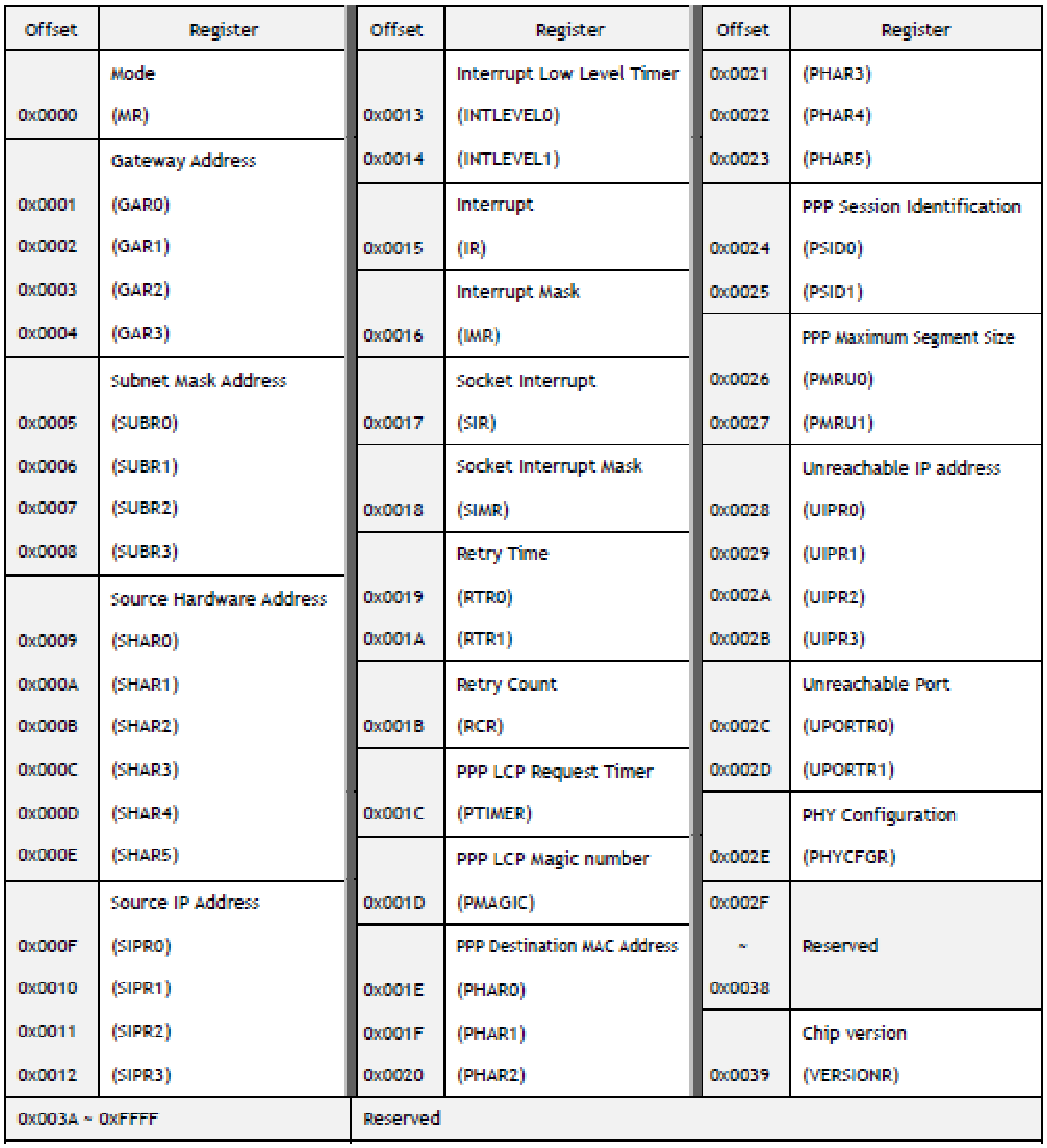
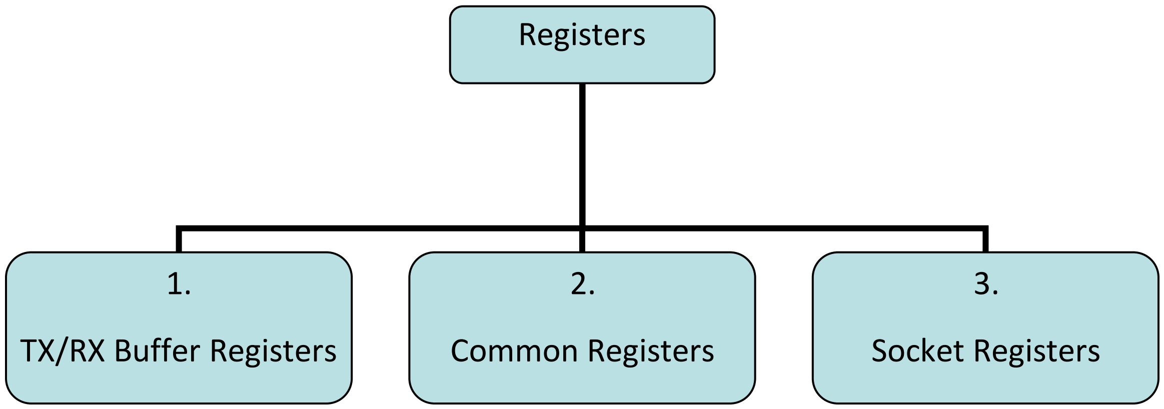


Table 3. Offset Address for Common Register

Offset	Register	Offset	Register	Offset	Register
	Mode		Interrupt Low Level Timer	0x0021	(PHAR3)
0x0000	(MR)	0x0013	(INTLEVEL0)	0x0022	(PHAR4)
	Gateway Address	0x0014	(INTLEVEL1)	0x0023	(PHAR5)
0x0001	(GAR0)		Interrupt		PPP Session Identification
0x0002	(GAR1)	0x0015	(IR)	0x0024	(PSID0)
0x0003	(GAR2)		Interrupt Mask	0x0025	(PSID1)
0x0004	(GAR3)	0x0016	(IMR)		PPP Maximum Segment Size
	Subnet Mask Address		Socket Interrupt	0x0026	(PMRU0)
0x0005	(SUBR0)	0x0017	(SIR)	0x0027	(PMRU1)
0x0006	(SUBR1)		Socket Interrupt Mask		Unreachable IP address
0x0007	(SUBR2)	0x0018	(SIMR)	0x0028	(UIPR0)
0x0008	(SUBR3)		Retry Time	0x0029	(UIPR1)
	Source Hardware Address	0x0019	(RTR0)	0x002A	(UIPR2)
0x0009	(SHAR0)	0x001A	(RTR1)	0x002B	(UIPR3)
0x000A	(SHAR1)		Retry Count		Unreachable Port
0x000B	(SHAR2)	0x001B	(RCR)	0x002C	(UPORTR0)
0x000C	(SHAR3)		PPP LCP Request Timer	0x002D	(UPORTR1)
0x000D	(SHAR4)	0x001C	(PTIMER)		PHY Configuration
0x000E	(SHAR5)		PPP LCP Magic number	0x002E	(PHYCFGR)
	Source IP Address	0x001D	(PMAGIC)	0x002F	
0x000F	(SIPR0)		PPP Destination MAC Address	~	Reserved
0x0010	(SIPR1)	0x001E	(PHAR0)	0x0038	
0x0011	(SIPR2)	0x001F	(PHAR1)		Chip version
0x0012	(SIPR3)	0x0020	(PHAR2)	0x0039	(VERSIONR)
0x003A ~ 0xFFFF	Reserved				

What are the settings which we can do using W5500 Registers?



1. The data which needs to be written onto the W5500 is written in this buffer space.
2. We do the following settings in Common Register Block: -

   i. Resetting all the register values to 0.

   ii. WoL option enable or disable based on magic number reception and similar settings.

   iii. Ping option enable and disable.

   iv. PPPoE enable and disable.

   v. ARP option enable and disable.

   vi. Gateway settings.

   vii. Subnet Mask settings.

   viii. Interrupt and interrupt mask settings for each socket.

   ix. Retransmission period setting (RTR setting).

   x. Retrying count setting (RCR setting).

   xi. Settings concerning PPPoE/ WOL based on whether they are disables or enabled.

   xii. PHY Chip related configurations

- Speed setting
- Resetting of PHY chip
- Mode setting (Power Down Mode/Run Mode etc.,)
- PHY link status setting

3. We do the following in the Socket Registers block: -

   I. Protocol selection (TCP/UDP/MAC RAW).

   II. To enable or to disable multicasting or broadcasting in UDP.

   III. To enable the IP version used (IPv4/IPv6).

   IV. Socket control setting which include when top perform the following operations. OPEN, CLOSE, CONNECT, LISTEN, SEND, RECEIVE etc.,

   V. Socket's interrupt related settings. (e.g.: Send complete acknowledgement, send timeout acknowledgement, successful connection established acknowledgement, socket created properly ack etc.,).

   VI. MTP setting (Maximum transmission size of a packet).

   VII. Destination port, IP setting.

   VIII. ToS, TTL setting.

   IX. The following settings need to done in TX/RX buffers

      - Buffer size
      - Read/Write pointer location

   X. Whether to fragment or to not fragment based on the size of the transmitting packet.

   XI. Auto negotiation of speed and duplex nature setting.

   XII. Keep alive time setting.