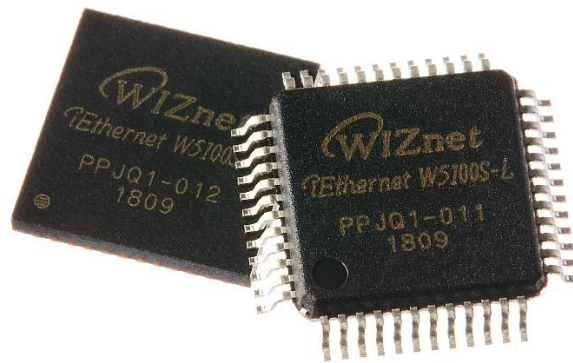


IPRAW Application Note for W5100S



Version 1.1.0



© 2018 WIZnet Co., Ltd. All Rights Reserved.

For more information, visit our website at <http://www.wiznet.io>

Table of Contents

1	IPRAW Introduction	3
2	IPRAW SOCKET	4
2.1	IPRAW Life Cycle	5
2.1.1	OPEN	5
2.1.2	SEND.....	5
2.1.3	RECEIVE.....	5
2.1.4	CLOSE	6
3	IPRAW Application Example	7
3.1	ICMP (Internet Control Message Protocol) Echo.....	7
3.2	PING Implementation.....	8
3.2.1	Calling PING Function	13
3.2.2	PING Request	13
3.2.3	PING Reply	14
4	Document History Information.....	15

1 IPRAW Introduction

W5100S는 IP Layer 상위의 프로토콜을 사용자가 직접 처리할 수 있는 IPRAW Mode를 제공한다. IPRAW Mode로 OPEN된 SOCKET은 TCP/IP Layer 중 IP Layer 상위의 정보들을 HOST가 직접 처리할 수 있도록 SOCKET TX/RX Buffer에 저장한다. Figure 1은 TCP/IP Layer에서 IP Layer 위치를 나타내며, Application Data가 각 하위 Layer로 전달되는 Data Encapsulation과정을 도식화 한 그림이다.

HOST는 IPRAW Mode에서 IP 헤더의 프로토콜 필드 값을 설정할 수 있으며 이를 통해 IP Layer의 다양한 프로토콜을 구현할 수 있다.

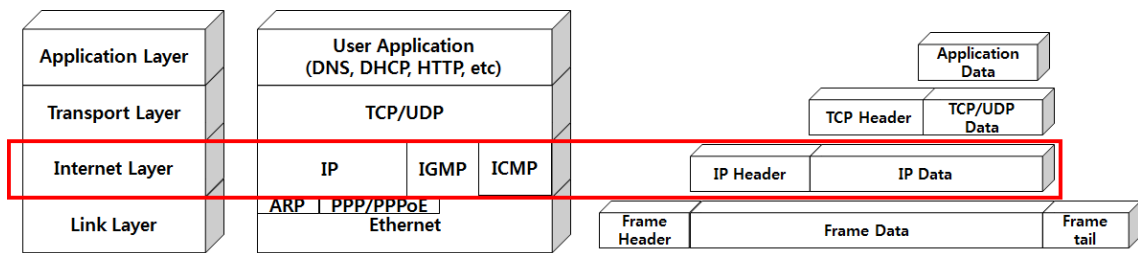


Figure 1 Encapsulation Process in TCP/IP

note

IPRAW를 사용할 경우 protocol 당 SOCKET 하나만을 사용 해야 하며, 여러 소켓을 사용할 경우 SOCKET Number가 작은 순으로 우선순위가 높다. 따라서 같은 protocol로 여러 SOCKET을 사용할 수 없으며 UDP와 TCP, IGMP와 IPv6관련 protocol은 사용 할 수 없다.

2 IPRAW SOCKET

W5100S은 4개의 SOCKET을 지원하며, 모든 SOCKET은 IPRAW Mode로 설정 가능하다. IPRAW Mode로 SOCKET n(n번째 SOCKET)을 사용할 경우, SOCKET OPEN이전에 Sn_PROTO(SOCKET n Protocol Register)를 통해 IP Layer 프로토콜을 설정해야 한다.

Table 1 Key Protocol in IP layer

Protocol	Number	Semantic	W5100S Support
HOPOPT	0	Reserved	X
ICMP	1	Internet Control Message Protocol	O
IGMP	2	Internet Group Management Protocol	X
IPv4	4	IPv4 encapsulation	O
TCP	6	Transmission Control Protocol	X
UDP	17	User Datagram Protocol	X
IPv6	41	IPv6 encapsulation	X
Others	-	Another Protocols(not related to IPv6)	O

W5100S은 4개의 SOCKET을 지원하며, 모든 SOCKET은 IPRAW Mode로 설정 가능하다. IPRAW Mode로 SOCKET n(n번째 SOCKET)을 사용할 경우, SOCKET OPEN이전에 Sn_PROTO(SOCKET n Protocol Register)를 통해 IP Layer 프로토콜을 설정해야 한다.

Table 1은 IPRAW Mode SOCKET이 지원하는 IP Layer 프로토콜을 보여준다. IPRAW Mode SOCKET은 TCP(0x06), UDP(0x17), IPv6(0x41) 통신을 지원하지 않으며, Sn_PROTOR에 설정된 프로토콜 이외의 통신 또한 지원하지 않는다.

Sn_PROTOR = ICMP(0x01)로 설정된 경우, W5100S는 상대방의 PING(Packet Internet Grouper) Request에 대한 Auto PING Reply를 더 이상 지원하지 않는다. 대신 PING Request Packet은 IPRAW Mode SOCKET n RX Buffer Block에 저장된다.

IPRAW Mode에서 수신되는 Data 구조는 Figure 2와 같다. 수신된 Data는 6 Bytes의 PACKET-INFO와 DATA로 이루어지며, PACKET-INFO는 송신자의 정보(IP address)와 DATA Packet의 길이가 포함된다.

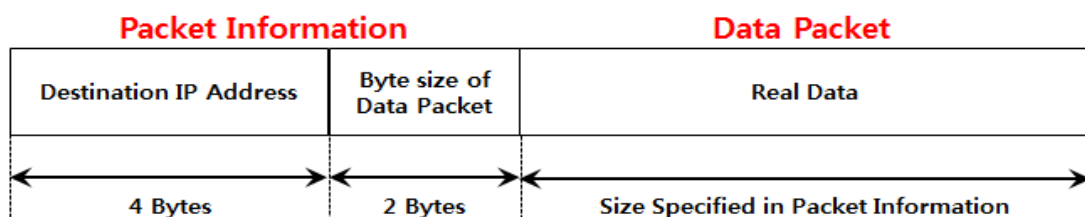


Figure 2 Received Data Format on IPRAW Mode

2.1 IPRAW Life Cycle

IPRAW Mode SOCKET Lifecycle은 OPEN, SEND, RECEIVE, CLOSE로 이루어 진다. 다음은 PING Application 예제를 통한 IPRAW Mode SOCKET의 Lifecycle 및 구현 방법을 나타낸 코드이다.

2.1.1 OPEN

SOCKET Number를 s로 선택한 후, Sn_PROTO에 Protocol Number를 ICMP로 설정하고 socket() 함수를 사용하여 IPRAW Mode로 설정되어 있는 SOCKET n을 OPEN한다. Sn_SR를 체크하여 SOCK_IPRAW(0x32)로 변경되었다면 SOCKET n의 OPEN이 완료된 것이다.

```
/* Create Socket */
IINCHIP_WRITE(Sn_PROTO(s), IPPROTO_ICMP); // set ICMP Protocol
if(socket(s,Sn_MR_IPRAW,port,0)!=s)
{ // open the SOCKET with IPRAW mode, if fail then Error
    printf( "\r\n socket %d fail r\n", (s) );
}
/* Check socket register */
while(getSn_SR(s)!=SOCK_IPRAW);
```

Example 1 SOCKET OPEN

2.1.2 SEND

sendto() 함수를 이용하여 PingRequest에 저장된 정보들을 Destination Address로 전송한다. IPRAW Mode로 설정된 SOCKET을 사용한다.

```
/* sendto ping_request to destination */
// Send Ping-Request to the specified peer.
if(sendto(s,(uint8_t *)&PingRequest,sizeof(PingRequest),addr,port)==0){
    printf( "\r\n Fail to send ping-reply packet r\n" );
}
```

Example 2 Send Data

2.1.3 RECEIVE

recvfrom() 함수를 이용하여 Destination Address로부터 수신 받은 Data를 data_buf에 저장

한다. IPRAW Mode로 설정된 SOCKET을 사용한다.

```
if ( (rlen = getSn_RX_RSR(s) ) > 0)
{
    /* receive data from a destination */
    len = recvfrom(s, (uint8_t *)data_buf, rlen,addr, (uint16_t*)&destport);
}
```

Example 3 Receive Data

2.1.4 CLOSE

SOCKET을 Close한다. IPRAW Mode SOCKET이 더 이상 필요하지 않을 경우 사용한다.

```
close(s);
```

Example 4 Close SOCKET

3 IPRAW Application Example

IPRAW Application Example로 ICMP(Internet Control Message Protocol)의 Echo Request, Echo Reply를 구현할 수 있다.

3.1 ICMP (Internet Control Message Protocol) Echo

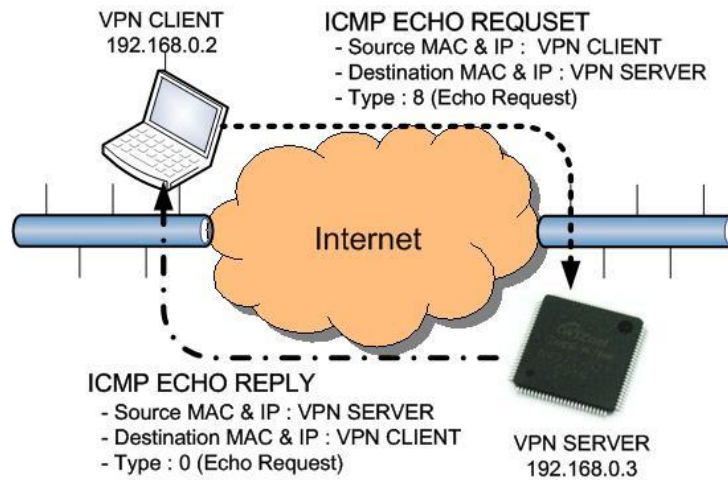


Figure 3 ICMP Echo Request/Reply

ICMP는 네트워크의 IP 상태 및 오류 정보를 공유하게 하며, PING(Packet Internet Groper)에서 사용되는 프로토콜이다. ICMP Message는 여러 Type이 존재하며, 그 중 ICMP Echo Message인 Echo Request와 Echo Reply는 PING Request Packet과 PING Reply Packet에서 이용되고, Device간 IP Datagram의 도달 여부를 확인하는데 사용된다.

Figure 3은 ICMP Echo Message의 Request/Reply 과정을 보여준다. Echo Request 일 경우 Type Filed 값은 8 이고, Echo Reply 일 경우 0의 값을 가진다. Table 2과 Table 3은 각각 Message Format과 Message Type을 나타낸다.

Table 2 ICMP Message Format

Type	Semantic
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect
8	Echo Request
11	Time Exceeded

12	Parameter Problem
13	Timestamp
14	Timestamp Reply

Table 3 ICMP Message Type

1 Byte	1 Byte
Type	Code
Check Sum	
Type dependent	
Data	

PING Command를 실행하면 Figure 3과 같이 Source(VPN client)에서 Destination(VPN server)에 대한 PING Request Packet 송신된다. PING Request Packet을 수신한 Destination은 Source에 대해 PING Reply Packet을 송신한다. PING Reply Packet은 PING Request Packet과 동일한 ID, Sequence Number, Data로 구성된다. 따라서, Source는 Destination으로부터 수신한 PING Reply Packet과 PING Request Packet의 ID, Sequence Number, Data를 비교하여 특정 Destination과의 연결을 확인 할 수 있다.

3.2 PING Implementation

PING Message의 ICMP Type Field는 '0'(PING Reply) or '8'(PING Request)을 가지며, Code Field는 '0'만을 가진다. 그리고 Check Sum, ID, Sequence Number Field는 각각 2Byte씩 값을 가진다. PING Data는 가변길이를 가진다. PING Message Format은 Table 4와 같다.

Table 4 PING Message Format

1 Byte	1 Byte
8 (0)	0
Check Sum	
ID	
Sequence Number	
PING Data	

PING Message를 쉽게 구현하기 위해 구조체를 사용했으며, 이는 Example 5 에 정의하였다.

```

#define BUF_LEN 32
#define PING_REQUEST 8
#define PING_REPLY 0
#define CODE_ZERO 0

typedef struct pingmsg
{
    uint8_t Type;           // 0 - Ping Reply, 8 - Ping Request
    uint8_t Code;          // Always 0
    int16_t CheckSum;      // Check sum
    int16_t ID;            // Identification
    int16_t SeqNum;        // Sequence Number
    int8_t Data[BUF_LEN]; // Ping Data : 1452 = IP RAW MTU - sizeof(Type+Code+CheckSum+ID+SeqNum)
} PINGMSG;
    
```

Example 5 PING Message Structure

PING Application은 ioLibrary의 SOCKET API중 Table 5에 언급된 API를 이용하여 구현할 수 있다.

Table 5 SOCKET API Functions

API Function Name	Meaning
socket	OPEN SOCKET with IPRAW Mode
sendto	SEND PING Request to Peer
recvfrom	RECEIVE PING Reply from Peer
close	CLOSE SOCKET

예제로 제공되는 PING Application은 IPRAW Mode에서 사용할 SOCKET과 Destination Address를 Parameter로 설정한다. 제공되는 PING Application은 W5100S가 PING Request Packet을 보내는 Mode와 PING Request Packet을 받는 Mode로 이루어져있으며, request_flag Parameter를 이용하여 Mode를 설정한다.

request_flag가 1이면, W5100S가 PING Request Packet을 보내는 Mode로 설정되며, W5100S는 특정 개수의 PING Request Packet을 Peer에게 보내고, 각각의 PING Request Packet에 대응하는 PING Reply Packet을 받는다. 이후 W5100S가 Peer로부터 받은 PING Reply Packet에 대해 CheckSum과 SeqNum(Sequence Number)을 확인하여, 올바른 PING Reply Packet인지 확인한다.

request_flag가 0이면, W5100S가 PING Request Packet을 받는 Mode로 설정되며, 각각의Peer로부터 받은 PING Request Packet에 대응하는 PING Reply Packet을 보낸다. 또한 Peer로부터

받은 PING Request Packet의 CheckSum과 SeqNum을 확인하여, 올바른 PING Request Packet 인지 확인한다.

다음은 제공되는 예제에 주로 사용되는 함수를 간략하게 설명한 것이다.

uint8 ping_auto(SOCKET s, uint8 *addr, uint8_t request_flag)

Table 6 ping_auto function

Function Name	ping_auto
Arguments	s - SOCKET number addr - Peer IP Address request_flag - PING Mode

IPRAW PING Application의 모든 동작을 수행하는 주요 함수이며, ping_request(), ping_reply(), 등으로 이루어져있다.

uint8 ping_request(SOCKET s, uint8 *addr)

Table 7 ping_request function

Function Name	ping_request
Arguments	s - SOCKET number addr - Peer IP Address

request_flag가 '1'인 경우에 사용되는 함수이며, W5100S에서 Peer로 PING Request Packet을 보낼 때 사용한다.

uint8 ping_reply (SOCKET s, uint8_t *addr, uint16_t len, uint8_t request_flag)

Table 8 ping_reply function

Function Name	ping_reply
Arguments	s - SOCKET number addr - Peer IP Address len - Packet length request_flag - PING Mode

Peer로부터 PING Request Packet 혹은 PING Reply Packet을 받는 경우 사용되는 함수이다. Request_flag에 따라 동작이 달라진다.

uint16 checksum(uint8 * data_buf, uint16 len)

Table 9 checksum function

Function Name	checksum
Arguments	data_buf - PING Message len - PING Message length

ping_request() 함수나 ping_reply() 함수에서 호출되는 함수로, CheckSum을 계산한다.

Figure 4 는 W5100S가 Peer에게 PING Request Packet을 보내는 Mode의 간략한 flow를 보여 준다.

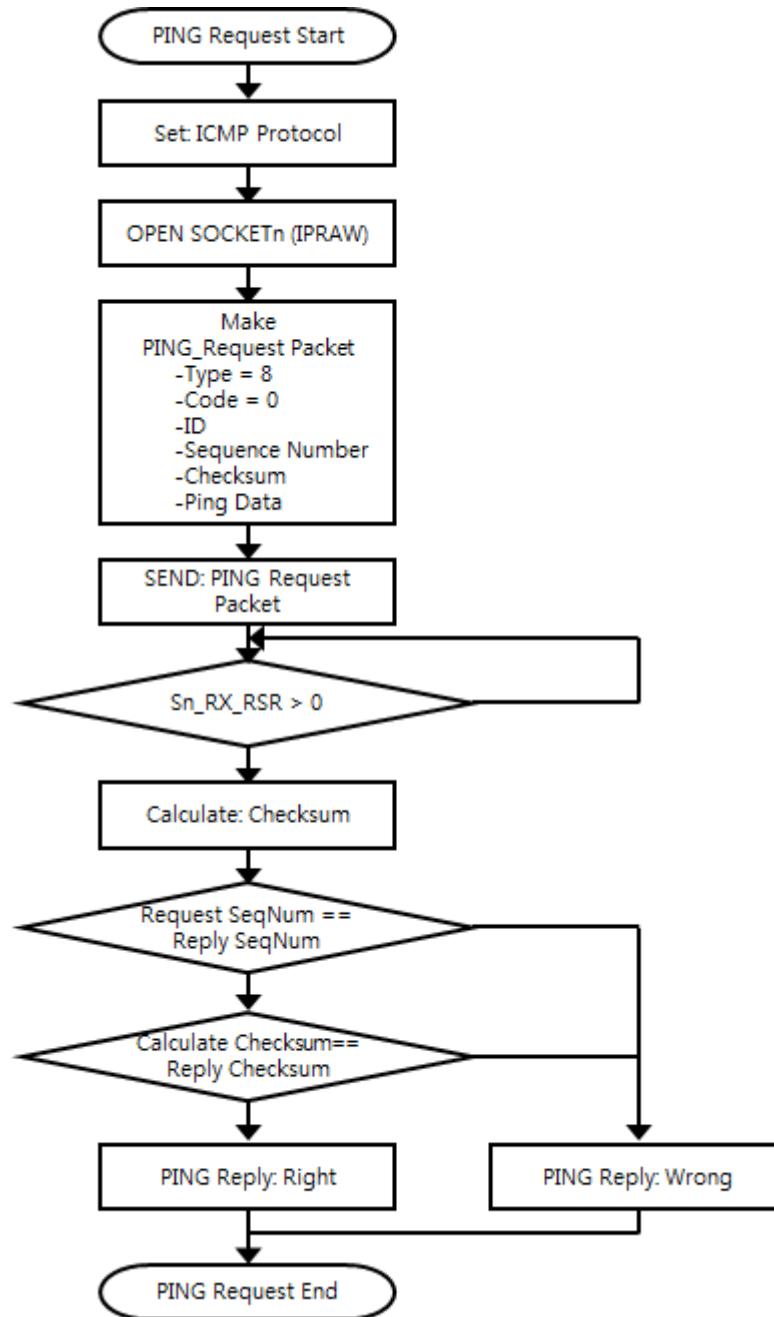


Figure 4 Flow chart of PING Application

3.2.1 Calling PING Function

PING Application Function은 Destination IP 를 설정해야 하며, PING Request Function은 Network Configuration 설정 이후 호출 된다. PING Request Function의 설정 과정을 Example 6 에 정리하였다.

```

/* main.c */
/* setting of Destination IP address */
pDestaddr[4]= {192,168,0,2};
/* Control Ethernet chip(W5100S) mode of request or reply*/
//request_flag = 0; //Send request ping from outside to Ethernet Chip(W5100S)
request_flag = 1; //Send request ping from Ethernet Chip(W5100S) to outside
/* Calling ping_request function */
ping_auto(0,pDestaddr, request_flag);

```

Example 6 Setting of PING Request Function

3.2.2 PING Request

PING Request Processing은 Protocol설정, SOCKET n OPEN, PING Request의 Header와 Data의 작성, 전송으로 구성된다. Example 7는 PING Request Processing을 보여준다. Protocol을 ICMP로 설정하여 SOCKET를 IPRAW Mode로 OPEN한다. PING Request의 헤더 및 Data를 작성 후 Checksum을 계산한 뒤, sendto()함수를 이용하여 PING Request를 Peer에게 전송한다.

```

/* ping.c */
/* set ICMP Protocol */
IINCHIP_WRITE(Sn_PROTO(s), IPPROTO_ICMP);
socket(s,Sn_MR_IPRAW,3000,0) ; /* open the SOCKET with IPRAW mode */

=====

/* make header of the ping-request */
PingRequest.Type = PING_REQUEST; // Ping-Request
PingRequest.Code = CODE_ZERO; // Always '0'
PingRequest.ID = htons(RandomID++); // set ping-request's ID to random integer value
PingRequest.SeqNum = htons(RandomSeqNum++); // set ping-request's sequence number to
random integer value

/* Do checksum of Ping Request */
PingRequest.CheckSum = 0;
PingRequest.CheckSum = htons(checksum((uint8*)&PingRequest,sizeof(PingRequest)));
:
/* send ping_request to destination */
sendto(s,(uint8 *)&PingRequest,sizeof(PingRequest),addr,3000);

```

Example 7 PING Request

3.2.3 PING Reply

Example 8 는 W5100S가 PING Request Packet을 받았을 때, PING Reply Processing을 보여준다. PING Reply Processing은 Data의 수신, PING Reply의 타입판별, 타입에 따른 처리로 구성된다. 수신된 Data의 Type을 확인하여 '0' 일 경우 PING Reply정보를 출력한다.

```

/* ping.c */
/* receive data from a destination */
len = recvfrom(s, (uint8 *)data_buf, rlen, addr, (int16_t*)destport);
/* check the Type */
else if(data_buf[0]== PING_REPLY)
{
    printf("PING_REPLY\r\n");
    /* for comp checksum */
    PingReply.Type           = data_buf[0];
    PingReply.Code           = data_buf[1];
    PingReply.CheckSum      = (data_buf[3]<<8) + data_buf[2];
    PingReply.ID            = (data_buf[5]<<8) + data_buf[4];
    PingReply.SeqNum        = (data_buf[7]<<8) + data_buf[6];

    :
    /* check Checksum of Ping Reply */
    tmp_checksum = PingReply.CheckSum;

    /* Calculate CheckSum of Ping Reply */
    PingReply.CheckSum = 0;
    PingReply.CheckSum = htons(checksum((uint8_t*)&PingReply, sizeof(PingReply)));

    if (tmp_checksum != PingReply.CheckSum)
    {
        printf(" \r\n Request CheckSum is incorrect %x should be %x \r\n", (tmp_checksum),
            htons(PingReply.CheckSum));
    }
    else
    {
        printf(" \r\n Request CheckSum is correct \r\n");
    }
    /* End of comp checksum */
    /* For send Ping Reply Packet */
    :
    PingReply.Type = 0;

    for(i=0; i<len-8 ; i++)
    {
        PingReply.Data[i] = data_buf[8+i];
    }

    /* Calculate Checksum of Ping Reply Packet */
    PingReply.CheckSum = 0;
    PingReply.CheckSum = htons(checksum((uint8_t*)&PingReply, sizeof(PingReply)));

    send_rep = sendto(s, (uint8_t *)&PingReply, sizeof(PingReply), addr, 3000);

```

Example 8 PING Reply

4 Document History Information

Version	Date	Descriptions
Ver. 1.0.0	Apr, 2018	Release
Ver. 1.1.0	MAY, 2018	Modified Contents of 3.2 PING Implementation, Figure 1, Figure 4, PING Request Code and PING Reply Code. Edit typo.

Copyright Notice

Copyright 2018 WIZnet Co.,Ltd. All Rights Reserved.

Technical Support: forum.wiznet.io or support@wiznet.io
Sales & Distribution: sales@wiznet.io

For more information, visit our website at www.wiznet.io