

W5200 Errata Sheet

Document History

| | |
|---------------------------|---|
| Ver 1.0.0 (Feb. 23, 2012) | First release (erratum 1) |
| Ver 1.0.1 (Mar. 28, 2012) | Add a solution for erratum 1, 2 |
| Ver 1.0.2 (Apr. 03, 2012) | Add a solution for erratum 3 |
| Ver 1.0.3 (Feb. 31, 2013) | Modified phenomenon and condition for erratum 1 |
| Ver 1.0.4 (Jun. 5, 2013) | Fixed typos in erratum 1 (W5300 -> W5200) |
| Ver 1.0.5 (Oct. 28, 2013) | Add a description of solution for erratum 1 |
| Ver 1.0.6 (JUL. 8, 2014) | Modify a description of solution for erratum 2 |

© 2012 WIZnet Co., Inc. All Rights Reserved.
 For more information, visit our website at <http://www.wiznet.co.kr>

| Erratum 1 | |
|---------------------------|---|
| Phenomenon | Source IP 주소가 “0.0.0.0” 인 Node에서 W5200 칩에 ARP Request 패킷을 전송하면 W5200 칩은 올바른 Target IP 주소인 “0.0.0.0”가 아닌 저장된 Gateway IP 주소를 Target IP 주소로 ARP Reply 패킷을 전송하는 문제가 있다. |
| Condition | <div style="border: 1px dashed black; padding: 10px; text-align: center;"> </div> <p>W5200 칩 내부의 ARP 처리과정에서 통신할 Target IP주소가 “0.0.0.0”일 때, Subnet계산의 오류로 인해 Target IP 주소인 “0.0.0.0”을 다른 Subnet에 존재하는 IP 주소로 잘못 판단하여 Target IP주소를 저장된 자신의 Gateway IP주소로 변경해서 ARP reply패킷을 전송하는 오류가 발생한다.</p> |
| Solution & Recommendation | <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px;"> <p>Before applying (without solution)</p> <pre> W5100 Initialization ... set GW : 192.168.1.254 set IP : 192.168.1.2 set SN : 255.255.255.0 ... </pre> </div> <div style="font-size: 2em;">→</div> <div style="border: 1px solid black; padding: 5px;"> <p>After applying (with solution)</p> <pre> W5100 Initialization ... set GW : 192.168.1.254 set IP : 192.168.1.2 set SN : 255.255.255.0 ... </pre> </div> </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="border: 1px solid black; padding: 5px; width: 45%;"> <p>TCP Connect</p> <pre> ... Get SIPR If SIPR = 0.0.0.0 Clear SN : 0.0.0.0 Else to maintain previous SN value after " Execute connect command " ... </pre> </div> <div style="border: 1px solid black; padding: 5px; width: 45%;"> <p>UDP Send</p> <pre> ... Get SIPR If SIPR = 0.0.0.0 Clear SN : 0.0.0.0 Else to maintain previous SN value after " Execute connect command " ... </pre> </div> </div> <p>이 Erratum을 피하기 위해 TCP의 “Connect” 명령과 UDP의 “SEND” 명령을 제외하고 항상 설정한 값으로 subnet mask register 값을 유지한다. TCP connect 명령과 UDP send 명령은 ARP 패킷을 송신하기 때문에 위 그림처럼 유효한 subnet mask 값을 전역변수에 저장해두고 TCP connect와 UDP send 명령을 수행할 때만 subnet mask register 값을 0으로 설정하고 명령이 완료되면 다시 subnet mask register 값을 저장해놓은 값으로 변경한다.</p> |

위의 내용을 적용했을 경우에는 Subnet broadcasting이 되지 않는다.

Example pseudo code:

```

/* Global variable declaration for subnet mask value */
unsigned char subnet_val[4];
/* W5100 initialization function */
Function Initialize_W5100( )
{
...
/* Clear the subnet mask register */
    IINCHIP_WRITE(SUBR0, 0);
    IINCHIP_WRITE(SUBR1, 0);
    IINCHIP_WRITE(SUBR2, 0);
    IINCHIP_WRITE(SUBR3, 0);
/* Save the right subnet mask value if the subnet is 255.255.255.0 */
    subnet_val[0] = 255;
    subnet_val[1] = 255;
    subnet_val[2] = 255;
    subnet_val[3] = 0;
...
}

/* TCP connect function */
Function TCP_Connect( )
{
...
/* Clear the subnet mask register again and keep it */
    IP_Val[0] = IINCHIP_READ(SIPR0);
    IP_Val[1] = IINCHIP_READ(SIPR0+1);
    IP_Val[2] = IINCHIP_READ(SIPR0+2);
    IP_Val[3] = IINCHIP_READ(SIPR0+3);

    If( IP_Val[0]==0 && IP_Val[1] ==0&& IP_Val[2] ==0&& IP_Val[3] ==0)
    {
        IINCHIP_WRITE(SUBR0, 0);
        IINCHIP_WRITE(SUBR1, 0);
        IINCHIP_WRITE(SUBR2, 0);
    }
}
    
```

```

IINCHIP_WRITE(SUBR3, 0);
}
/* Execute TCP connect command */
IINCHIP_WRITE(Sn_CR(socket), Sn_CR_CONNECT);
/* Wait for command done */
while(Sn_CR(socket));

/* Set the subnet mask register to the right value using the variable */
IINCHIP_WRITE(SUBR0, subnet_val[0]);
IINCHIP_WRITE(SUBR1, subnet_val[1]);
IINCHIP_WRITE(SUBR2, subnet_val[2]);
IINCHIP_WRITE(SUBR3, subnet_val[3]);

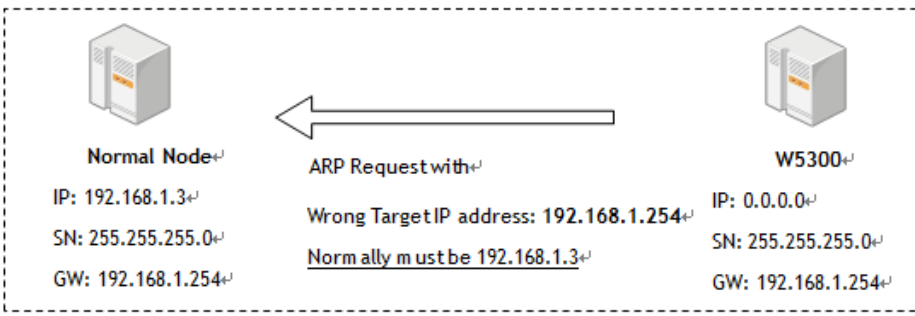
...
}

/* UDP sendto function */
Function UDP_Sendto( )
{
...
/* Clear the subnet mask register again and keep it */
IP_Val[0] = IINCHIP_READ(SIPR0);
IP_Val[1] = IINCHIP_READ(SIPR0+1);
IP_Val[2] = IINCHIP_READ(SIPR0+2);
IP_Val[3] = IINCHIP_READ(SIPR0+3);

If( IP_Val[0]==0 && IP_Val[1] ==0&& IP_Val[2] ==0&& IP_Val[3] ==0)
{
IINCHIP_WRITE(SUBR0, 0);
IINCHIP_WRITE(SUBR1, 0);
IINCHIP_WRITE(SUBR2, 0);
IINCHIP_WRITE(SUBR3, 0);
}
/* Execute UDP send command */
IINCHIP_WRITE(Sn_CR(socket), Sn_CR_SEND);
/* Wait for command done */

```

```
while(Sn_CR(socket));  
/* Set the subnet mask register to the right value using the variable */  
IINCHIP_WRITE(SUBR0, subnet_val[0]);  
IINCHIP_WRITE(SUBR1, subnet_val[1]);  
IINCHIP_WRITE(SUBR2, subnet_val[2]);  
IINCHIP_WRITE(SUBR3, subnet_val[3]);  
...  
}
```

| Erratum 2 | |
|---------------------------|--|
| Phenomenon | <p>W5200 칩에서 자신의 IP주소는 “0.0.0.0”으로 설정되어 있으며, Gateway IP 주소와 Subnet mask 정보가 0이 아닌 값으로 설정되어 있는 상황에서 네트워크상의 다른 Node에게 ARP Request패킷을 전송할 때, Target IP address로 상대방 IP주소를 정확히 설정했음에도 불구하고 ARP 패킷의 Target IP address는 설정한 상대방 IP주소가 아닌 Gateway IP주소로 설정된 ARP Request 패킷을 송신하는 문제가 있다.</p> |
| Condition | <div style="border: 1px dashed black; padding: 10px; text-align: center;">  <p>Normal Node[↵] IP: 192.168.1.3[↵] SN: 255.255.255.0[↵] GW: 192.168.1.254[↵]</p> <p>← ARP Request with[↵] Wrong Target IP address: 192.168.1.254[↵] <u>Normally must be 192.168.1.3[↵]</u></p> <p>W5300[↵] IP: 0.0.0.0[↵] SN: 255.255.255.0[↵] GW: 192.168.1.254[↵]</p> </div> <p>W5200 칩 내부의 ARP 처리과정에서 자신의 source IP주소가 “0.0.0.0”일 때, Subnet계산오류로 인해 설정된 Target IP 주소를 보내지 못하고 Target IP주소를 자신의 저장된 Gateway IP주소로 변경하여 ARP request 패킷을 전송하기 때문에 전달 되어야 할 Normal node로 ARP Request 패킷이 전달되지 않는다. 또한 같은 상황에서 Gateway IP 주소가 “0.0.0.0”일 경우, Target IP 주소가 현재 Gateway 주소인 “0.0.0.0”으로 변경되어 ARP request 패킷을 전송하게 되어 Target IP 주소로 패킷이 전달되지 않는 문제가 발생하게 된다.</p> |
| Solution & Recommendation | <p>Erratum2의 원인은 Erratum1와 동일하기 때문에 Erratum1의 솔루션과 동일 하다.</p> |

| Erratum 3 | |
|---------------------------|--|
| Phenomenon | <p>W5200칩의 macraw mode를 사용하는 경우, 간혹 패킷 수신 사이즈가 실제 패킷 사이즈 보다 크게 들어오는 현상이 있다.</p> <p><u>이 현상으로 수신된 패킷 사이즈가 실제 칩 내부의 최대 수신 사이즈 보다 작은 경우에도 발생하며, 이때 수신된 패킷은 최대 수신 사이즈 보다 크게 되어 버려지는 현상도 존재한다.</u></p> <p>-수신패킷사이즈가 실제패킷사이즈 보다 증가한 경우</p> <p>MACRAW data의 구조는 아래와 같고, PACKET-INFO(data packet 길이)가 실제 패킷 사이즈 보다 크게 들어오는 현상이 있다. 이때, 실제 패킷 사이즈에서 PACKET-INFO길이까지는 Dummy data가 존재한다.</p> <div style="text-align: center;"> </div> <p>Figure The received MACRAW data Format</p> <p>-수신사이즈가 최대 수신 사이즈보다 크게 되어 수신패킷이 버려지는 경우</p> <p>칩 내부에서 1514 바이트가 최대 수신 사이즈로 지정되어 있어 그 이상의 수신 데이터 사이즈의 패킷은 버리게 되는 경우가 존재한다.</p> |
| Solution & Recommendation | <p>-수신사이즈가 크게 들어오는 경우</p> <p>실제 패킷 사이즈 보다 크게 들어오는 오류 상황을 피하기 위한 방법으로 IP Header 의 Total length Filed 를 이용하여 Packet 의 실제 사이즈를 확인하여 패킷의 올바른 수신 사이즈를 판단 할 수 있다. MACRAW data 의 Payload 내부에 IP Header 가 존재하며, 이 IP Header 에는 전체 패킷 사이즈 정보를 가진 Filed 가 존재한다. 따라서, Software stack 을 사용하는 경우 Software stack 이 IP Header 의 Total length 를 이용하여 수신된 Packet 의 실제 사이즈를 확인하도록 설정해야 한다.</p> <p>-수신사이즈가 최대 수신 사이즈보다 크게 되어 수신패킷이 버려지는 경우</p> <p>1514 바이트가 최대 사이즈로 지정되어 있어 최대 사이즈 이상의 패킷을 버리는 경우를 피하기 위해 TCP/UDP 이용시 패킷이 최대사이즈로 송수신되지 않도록 버퍼사용을 제한해야 한다 TCP 의 경우는 간단하게 Windows 사이즈(수신버퍼)를 1459byte 이하로 조절해야 한다. UDP 의 경우 Application</p> |

| | |
|--|--|
| | <p>Layer 에서 한번에 전송할 수 있는 패킷의 사이즈를 1452바이트 미만으로 제한한다.</p> |
|--|--|

단, W5200의 Hardwired TCP/UDP 모드를 사용할 경우 Erratum3의 현상이 발생하지 않는다.