

## W5300 Errata Sheet

### Document History

Ver 1.1.0 (AUG. 19, 2008)	First release (erratum 1, 2, 3)
Ver 1.2.0 (FEB. 23, 2012)	Add Erratum 4, 5 Change the Errata sheet form (Match with W3150A+ / W5100 Errata sheet.)
Ver 1.2.1 (MAR. 23, 2012)	Add a solution for erratum 4,5
Ver 1.2.2 (FEB. 7, 2014)	Add a description of solution for erratum 4
Ver 1.2.3 (MAR. 19, 2015)	Modify a description of solution for erratum 4

© 2012 WIZnet Co., Ltd. All Rights Reserved.

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

Erratum 1	
Phenomenon	소켓이 TCP로 동작하는 경우, TCP 연결 설정 과정에서 Socket status register(Sn_SSR)이 “0x10” 이나 “0x11”에서 변경되지 않는 현상
Condition	TCP 데이터 전송 과정이 완료되지 않은 상태, 즉 “SEND” 명령 후, SEND_OK interrupt 가 발생하지 않은 상태 <sup>1</sup> 에서, 사용자가 “CLOSE” 나 “DISCONNECT” 명령으로 강제 종료를 하면, Socket status register(Sn_SSR)이 “0x10” 이나 “0x11”에서 변경되지 않아 TCP 연결 설정이 되지 않는다.
Solution & Recommendation	<p>데이터 전송 완료를 기다리지 않고 강제 종료를 한 경우에는 반드시 다음의 코드를 추가해서 삽입한다. (또는 close() 함수에 무조건 이 동작을 수행하게 한다.)</p> <pre style="background-color: black; color: yellow; padding: 5px;"> socket(ch, Sn_MR_UDP, 5000, 0x00); // UDP로 강제로 연다. 포트넘버는 임의 값을 사용해도 무방하다. sendto(ch, data_buf, 1,(uchar*)&amp;destip,destport); // 전송명령을 내린다. destip와 destport는 임의 값을 사용해도 무방하다. // 테스트는 destip를 0.0.0.1 destport 5000으로 하여 수행하였다. close(ch); // close 한다..                     </pre> <p>위와 같이 UDP 데이터 전송과정을 강제로 수행하면 데이터 전송 완료 대기 상태를 강제로 해제해 주어, 정상적인 동작이 가능하다.</p>

<sup>1</sup> 일반적으로 수 us 안에 완료된다. 그러나 TCP는 데이터 송수신 중에 상대의 수신 윈도우 사이즈가 송신 데이터의 사이즈보다 작으면 데이터 전송 과정이 완료되지 않고 상대의 수신 윈도우 값이 커지기를 기다리는 대기상태에 있게 되어 가끔 수 초 이상의 시간이 걸리기도 한다.

Erratum 2	
Phenomenon	<p>소켓이 TCP로 동작하는 경우,            “Window Update ACK” packet이 자동으로 나가지 않아 수신 속도가 저하되는 현상</p>
Condition	<p>TCP는 데이터 버퍼 크기(Window)를 서로 교환하여 데이터의 전송 속도를 조절한다. 그래서, 상대방의 데이터 버퍼 크기가 보낼 데이터 크기보다 작은 경우, 전송을 하지 않고 대기하게 된다. 그리고, 상대방은 데이터 버퍼 크기가 일정 수준 이상 되면 즉시 버퍼 크기의 변화를 알려주어(“Window Update ACK” packet) 대기중인 전송을 처리하게 한다.</p> <p>그러나 W5300에는 위에서 설명한 “Window Update ACK” packet이 자동으로 나가지 않아 데이터 수신 속도가 저하되는 현상이 발생한다.</p> <p><i>*) 참고로 W5300이 TCP로 동작하는 경우, “SEND” command와 타임아웃에 의해ACK packet이 전송된다. 그리고, “No delayed option”을 사용하면 데이터 packet을 수신하는 경우에도 ACK packet이 전송된다.</i></p>
Solution & Recommendation	<p>수신 처리를 최대한 빨리 해주어 수신 버퍼 크기를 MSS 이상의 값으로 유지하는 것이 가장 좋다. 이런 경우는 아예 “Windows Update ACK” packet 기능이 필요 없기 때문이다.</p> <p>만약 이런 조건이 안 된다면, 사용자는 강제로 “SEND” command를 내려 “Window Update ACK” packet을 전송 하는 효과를 낼 수 밖에 없다. 즉, 수신 버퍼 크기가 MSS 값 이하가 되었다가, 커지는 경우<sup>2</sup> 반드시 강제로 “SEND” command를 내려 ACK packet이 전송되게 한다<sup>3</sup>.</p>

<sup>2</sup> “User Application”에서 수신된 데이터의 처리를 완료하면, “RECV” command를 내려 수신 버퍼 크기를 증가시켜 준다.

<sup>3</sup> “No operation”과 같은 의미의 dummy data를 전송 하면 된다.

Erratum 3	
Phenomenon	소켓이 TCP로 동작하는 경우, Destination Port Number Register (Sn_DPORTR)가 잘못 읽혀진다.
Condition	TCP 연결 설정 후에 Destination Port Number Register(Sn_DPORTR)에 정상적인 Destination Port Number가 저장되지만, 이 값을 읽으면 제대로 읽혀지지 않고 상위바이트 값이 중복적으로 읽혀지는 현상이다. 예를 들어 상대방의 포트숫자가 0x1234인 경우, Sn_DPORTR에는 0x1234가 저장되어 있지만 0x1212로 읽혀진다.
Solution & Recommendation	해결할 수 있는 방법이 없다. 그러나, 상대방 포트 정보가 잘 못 읽혀질 뿐 실제 저장된 값은 정상이므로 데이터 송수신을 포함한 TCP 동작에는 문제가 없다.

Erratum 4	
Phenomenon	Source IP 주소가 “0.0.0.0” 인 Node에서 W5300 칩에 ARP Request 패킷을 전송하면 W5300 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; margin-bottom: 10px;"> </div> <p>W5300 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; width: 30%;"> <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; width: 30%;"> <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 전역 변수에 저장해두고 TCP connect와 UDP send 명령을 수행할 때만 subnet mask register 값을 0으로 설정하고 명령이 완료 되면 다시 subnet mask register 값을 저장해놓은 값으로 변경한다. 위의 내용을 적용했을 경우에는 Subnet broadcasting이 되지 않는다. Example pseudo code:</p>

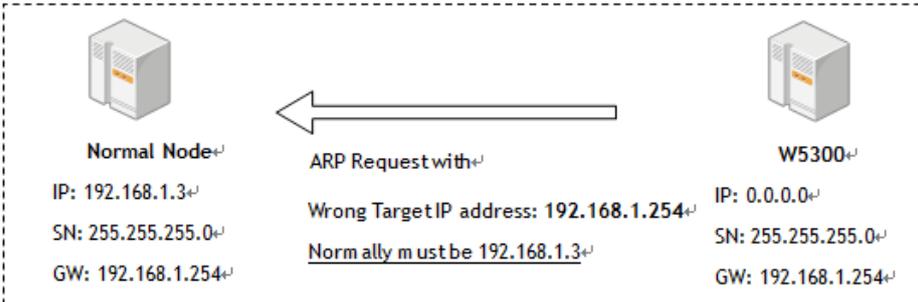
```

/* 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 5	
Phenomenon	<p>W5300 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 gray; padding: 10px; text-align: center;">  <p>Normal Node<sup>↵</sup>            IP: 192.168.1.3<sup>↵</sup>            SN: 255.255.255.0<sup>↵</sup>            GW: 192.168.1.254<sup>↵</sup></p> <p>← ARP Request with<sup>↵</sup>            Wrong Target IP address: 192.168.1.254<sup>↵</sup>  <u>Normally must be 192.168.1.3<sup>↵</sup></u></p> <p>W5300<sup>↵</sup>            IP: 0.0.0.0<sup>↵</sup>            SN: 255.255.255.0<sup>↵</sup>            GW: 192.168.1.254<sup>↵</sup></p> </div> <p>W5300 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>Erratum5의 원인은 Erratum4와 동일하기 때문에 Erratum4의 솔루션과 동일 하다.</p>