

W3150A+/W5100 Errata Sheet

Document History

Ver 1.0.0 (OCT. 27, 2007)	First release (errata 1)
Ver 2.0.0 (SEP. 10, 2008)	Add W5100 solution for errata 1 Remove Recommendation for errata 1
Ver 2.1 (APR. 5, 2010)	Remove the errata solution for W5100 in v2.0 and bring back the recommendation of v1.0
Ver 2.2 (FEB. 17, 2012)	Add errata2, 3
Ver 2.3 (MAR. 5, 2012)	Add a solution for erratum 2, 3
Ver 2.4 (OCT. 28, 2013)	Add a description of solution for erratum 2
Ver 2.5 (JUL. 7, 2014)	Modify a description of solution for erratum 2

© 2007 ~ 2012 WIZnet Co., Ltd. All Rights Reserved.
 For more information, visit our website at <http://www.wiznet.co.kr>

Errata 1	
Phenomenon	UDP/IP-Raw mode에서 SEND, SENDMAC command에 의한 전송이 종료되지 않는다. 전송이 완료되었음을 나타내는 SEND_OK interrupt도 발생하지 않고 해당 소켓의 Socket n TX Read Pointer Register (Sn_TX_RD)와 Socket n TX Write Pointer Register (Sn_TX_WR)의 값이 같아지지 않는다.
Condition	<p>일반적으로 전송이 종료되면 Sn_TX_RD와 Sn_TX_WR의 값은 같아지고 SEND_OK 인터럽트가 발생한다. 하지만 UDP/IP-Raw모드에서 Host MCU가 SEND, SENDMAC명령을 내려 전송을 시도할 때, 전송하려는 해당 소켓의 data packet이 수신처리를 하는 경우 (즉, RECV interrupt가 발생하는 시점)에 위와 같은 현상이 발생한다.</p> <div style="text-align: center;"> </div> <p>Figure 1. General network</p> <p>그림 1과 같은 상황에서,</p> <ol style="list-style-type: none"> 1. AA에서 UDP 1000 port로 소켓을 생성한다. 2. AB에서 AA의 1000 port로 UDP data packet을 전송한다. 3. AA는 AB에서 들어온 UDP data packet을 정상 수신처리 한다. 4. AA는 동시에 보낼 데이터가 있어 SEND 명령을 수행한다. <p>AA에서 3번 과정이 완료되는 시점 (수신 인터럽트가 발생하는 시점)에 4번 과정을 수행하면 전송명령을 완료하지 못하는 버그가 있다.</p>
Solution	<p>In case of W3150A+,</p> <p>MII input signal 중에 RXDV signal을 모니터링 하면 이 문제점을 해결할 수</p>

	<p>있다. Host MCU가 SEND, SENDMAC 명령을 수행하기 전에 RXDV가 '0'인지 확인하고, '0' 인 경우에 전송명령을 수행하면 된다. RXDV가 '0'이라는 의미는 현재 수신중인 packet이 없다는 것을 의미하므로 에러가 발생하는 상황을 피할 수 있다.</p> <pre> /* Change sendto() function */ Function sendto() { ... While (RXDV == '1') ; SEND command; /* Complete Sending */ } </pre>
<p>Recommendation</p>	<p>In case of W5100,</p> <p>W5100의 경우 아래와 같은 recommendation외에 완전한 해결방법은 현재 없다.</p> <p>전송과정이 끝나고 나서, Sn_TX_RD와 Sn_TX_WR의 값이 같은지 확인한다. 만약 전송과정이 끝났음에도 불구하고 이 두 값이 다르다면, 강제로 socket을 close하고 W5100을 reset한다.</p> <pre> /* Change sendto() function */ Function sendto() { ... /* Complete Sending */ /* wait until Sn_TX_WR and Sn_TX_RD are same */ While (Sn_TX_WR != Sn_TX_RD) { wait some time; loop_cnt++; if (loop_cnt > CONST_BLOCK_CNT) goto RESET } } </pre>

Errata 2	
Phenomenon	Source IP 주소가 “0.0.0.0” 인 Node에서 W3150A+/W5100 칩에 ARP Request 패킷을 전송하면 W3150A+/W5100 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;"> <p>Normal Node⁺ IP: 0.0.0.0⁺ SN: 255.255.255.0⁺ GW: 192.168.1.254⁺</p> <p>W5100/W7100A⁺ IP: 192.168.1.2⁺ SN: 255.255.255.0⁺ GW: 192.168.1.254⁺</p> <p>ARP Request⁺</p> <p>ARP Reply with⁺ Wrong Target IP address: 192.168.1.254⁺ Normally must be 0.0.0.0⁺</p> </div> <p>W5100/W7100/W7100A 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> <p>W5100 Initialization</p> <p>... set GW : 192.168.1.254 set IP : 192.168.1.2 set SN : 255.255.255.0 ...</p> </div> <div style="font-size: 2em;">→</div> <div style="border: 1px solid black; padding: 5px;"> <p>After applying (with solution)</p> <p>W5100 Initialization</p> <p>... set GW : 192.168.1.254 set IP : 192.168.1.2 set SN : 255.255.255.0 ...</p> </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> <p>... Get SIPR If SIPR = 0.0.0.0 Clear SN : 0.0.0.0 Else to maintain previous SN value after " Execute connect command "</p> </div> <div style="border: 1px solid black; padding: 5px; width: 45%;"> <p>UDP Send</p> <p>... Get SIPR If SIPR = 0.0.0.0 Clear SN : 0.0.0.0 Else to maintain previous SN value after " Execute connect command "</p> </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 완료</p>

되면 다시 subnet mask register 값을 저장해놓은 값으로 변경한다.
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]);

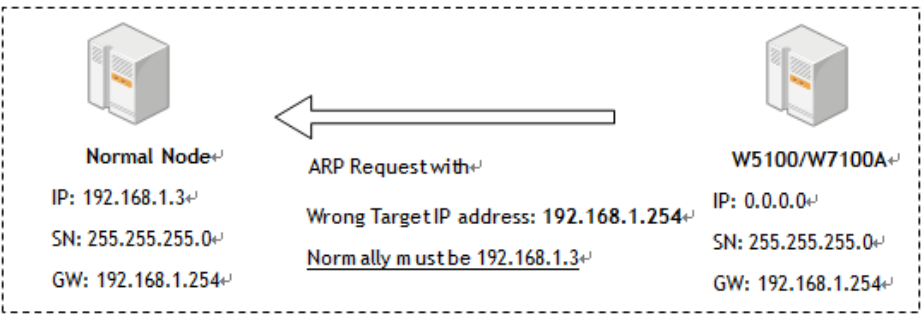
...
}

/* 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]);
...
}
```

Errata 3	
Phenomenon	<p>W5100/W7100/W7100A 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[↙] Normally must be 192.168.1.3[↙]</p> <p>W5100/W7100A[↙] IP: 0.0.0.0[↙] SN: 255.255.255.0[↙] GW: 192.168.1.254[↙]</p> </div> <p>W5100/W7100/W7100A 칩 내부의 ARP 처리과정에서 자신의 source IP주소가 “0.0.0.0”일 때, Subnet계산오류로 인해 설정된 Target IP 주소를 보내지 못하고 Target IP주소를 자신의 저장된Gateway 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>Erratum3의 원인은 Erratum2와 동일하기 때문에 Erratum2의 솔루션과 동일하다.</p>