

W3150A Errata Sheet

Document History

Ver 1.0.0 (OCT 27, 2005)	First release (errata 1,2)
Ver 1.1.0 (NOV 23, 2005)	Second release (errata 3)
Ver 1.1.1 (NOV 24, 2005)	Modify contents

© 2005 WIZnet Co., Inc. All Rights Reserved. For more information, visit our website at <u>http://www.wiznet.co.kr</u>

© Copyright 2005 WIZnet Co., Inc. All rights reserved.





© Copyright 2005 WIZnet Co., Inc. All rights reserved.



HOST_AB and HOST_BA are in the different subnet.

How to find out they are in the different subnet.

- 1. Do a bitwise AND operation with HOST_AB (10.11.12.4) and Subnet mask (255.255.255.0). The result will be 10.11.12.0.
- 2. Do a bitwise AND operation with HOST_BA (10.11.22.3) and Subnet mask (255.255.255.0). The result will be 10.11.22.0.
- 3. The different result of 1 and 2 means HOST_AB and HOST_BA are in the different subnet.

The following process shows under the errata condition.





	Generally, HOST_AA gets the Ethernet MAC address of HOST_AB by automatic
	hardware ARP processing before trying to connect or sending UDP data packet
	to HOST_AB.
	On the other hand, it's unable to get the Ethernet MAC address of HOST_AB
	due to errata HOST AA, which has the different gateway address of the
	different subnet. Therefore, HOST AA can't connect or send LIDP data packet
	to HOST AB in the same subnet
Solution	Change gateway ID address into destination ID address before trying to
301011011	connect or sonding UDD data packet
	Refer to the following pseudo-code.
	Assume that Host_AA tries to connect or sends UDP data packet to Host AB.
	/* Check whether both HOST_AA and HOST_AB are in the same subnet */
	if (is_same_subnet(HostAB) == true)
	{
	/* Check whether both HOST_AA and gateway are in the same subnet */
	<pre>if (is_same_subnet(gateway) == true) no problem;</pre>
	else
	{
	/* Under the Errata condition */
	/* Change gateway address into destination address (HOST_AB) */
	setGatewayAddress(HOST_AB);
	}
	}
	connect or send UDP data packet
	/* Change gateway address back into the original value */
	setGatewayAddress(gateway):
	setGatewayAudiess(gateway),



Condition	The result of doing bitwise OR operation with "Destination IP address" and
oonartion	"Subnet mask" is 255 255 255 in UDP mode
Phenomenon	Unable to send UDP data packet to a host in the different subnet.
Description	If "Destination IP address" is in the same subnet and the result of doing
Description	bitwise OR operation with "Destination IP address" and "Subnet mask" in
	255 255 255 255 in UDP mode UDP data packet is broadcast (i.e. the
	destination Ethernet MAC is FEFEFEFEFEFEFEFEFE
	If it is in the different subnet. UDP data packet is unicast.
	On the other hand, UDP data packet is broadcast under the errata condition
	although it is in the different subnet.
Solution	Use SENDMAC command with gateway Ethernet MAC address instead o
	SEND command.
	Refer to the following pseudo-code.
	Change sendto() function included errata 1.
	/* Check whether both HOST_AA and HOST_AB are in the same subnet */
	if (is_same_subnet(HostAB) == true)
	{
	/* Check whether both HOST_AA and gateway are in the same subnet */
	if (is_same_subnet(gateway) == true) no problem;
	else
	{
	/* Under the Errata condition */
	/* Change gateway address into destination address (HOST_AB) */
	setGatewayAddress(HOST_AB);
	}
	<pre>/* data processing for sending */</pre>
	<pre>/* send UDP data packet by SEND command */</pre>
	Sn_CR = SEND;
	/* Change gateway address back into the original value */
	setGatewayAddress(gateway);
	}
	else
	{



/* include In Errata 2 condition */
/* if Destination IP is in the different subnet, write Gateway Hardware
Address (GW_MAC) to Socket n Destination Hardware Address
Register(Sn_DHAR). For getting GW_MAC, refer to 5.1 Initialization of
W3150A datasheet. */
Sn_DHAR = GW_MAC;
/* Data processing for sending */
/* Send UDP data packet by SENDMAC command */
Sn_CR = SENDMAC;
}







	low speed. In this case, PEER_B informs it to PEER_A (Figure 3. (1) window=0
	packet), and the size of sending data of PEER_A is limited until the RXBUF or
	PEER_B makes room (Figure 3. 2) window=n packet) by TCP flow control as
	Figure 3. (1). When Figure 3. (2) packet is received, all data are sent by
	Transmission Timeout (Figure 3, 3), and then Sn TX RD value is equal to
	Sn TX WR value
	Due to errata
	Case 1. If SEND command is given before Figure 3. (i.e. Sp. TV, PD value
	is different from Sn_TX_WD value). TCD data are unable to be cent
	is different from sn_1x_we value), TCP data are unable to be sent.
	Case 2. Data cannot be sent as the data size to transmit but can be sent as the
	received window size, in case of receiving Figure 3. (2) packet just at that
	moment Transmission Timeout occurs (during 40ns.) as Figure 3. (3). This
	case rarely happens.
Recommen-	1. Be sure to check whether Sn_TX_RD has same value as Sn_TX_WR or no
dation	after a transmission process is complete.
	2. If the two values are still different, close the socket by force and make a
	TCP connection again.
	Refer to following pseudo-code.
	/* Change send() function */
	Function send()
	{
	/* Complete Sending */
	/* wait until Sn TX WR and Sn TX RD are same */
	while (Sn TX WR!= Sn TX RD)
	{
	if socket was closed, then goto CLOSE state
	wait 200ms or 400ms
	if(loop ont , 10) gets $CLOSE$ state
	II (100p_citt > 10) goto CLOSE state.
	}

© Copyright 2005 WIZnet Co., Inc. All rights reserved.