How to implement UDP communication for W7100A

version 1.1

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

Table of Contents

1 Introduction

UDP provides unreliable and connectionless datagram transmission structure. UDP processes data without a connection establishment. Therefore, UDP message can be lost, overlapped or reversed. When packets arrive in a fast phase, the recipient cannot process all of the packets. In this case, the user must consider whether UDP is suitable for its application or not. UDP is usually used in a Multicast system since it can broadcast datagram.

2 UDP SOCKET

User can use all eight sockets of W7100 as UDP protocol for UDP communication. In order for the user to use UDP protocol of W7100, the SOCKET should be created. The following are things that are needed when creating a SOCKET; socket number, protocol that will be used, number of the port that will be used, and flag that will be set. Since this document explains about UDP protocol, the protocol is set by Sn_MR_UDP(0x02) register. Socket number means each and all of eight sockets of W7100, and can be numbered from 0 to 7 by user. User can also allocate the port number that will be used from the UDP protocol. When the opening the UDP socket, use the SOCKET() function, provided by WIZnet, and allocate the above information.

2.1 Network Initialization

The network configuration for W7100 requires IP address, Gateway, subnetmask, and MAC address.; network parameters are processed and shown in Fig 2.2.1. Also set the TX/RX buffer size for each sockets and set the other registers.

```
 /* main.c */
// W7100 CPU core initialize interrupts, memory access time, serial and etc..
// W7100 network initialization as follow
uint8 xdata ip[4] = {192,168,1,2};          // Setting IP address
uint8 xdata gw[4] = {192,168,1,1};          // Setting Gateway address
uint8 xdata sn[4] = {255,255,255,0};        // Setting Subnet Mask
uint8 xdata mac[6] = {0x00,0x08,0xDC,0x33,0x33,0x65};       //Setting MAC

// TX and RX buffer size for socket 0 to 7 (2 bytes for each socket)
uint8 xdata txsize[MAX_SOCK_NUM] = {2,2,2,2,2,2,2,2};
uint8 xdata rxsize[MAX_SOCK_NUM] = {2,2,2,2,2,2,2,2};

// Write IP address to SIPR register
IINCHIP_WRITE (SIPR0+0,ip[0]); ... IINCHIP_WRITE (SIPR0+3,ip[3]);
```

```
// Write Gateway address to GAR register
IINCHIP_WRITE (GAR0+0, gw[0]); ... IINCHIP_WRITE (GAR0+3, gw[3]);
// Write Subnet mask to SUB register
IINCHIP_WRITE (SUBR0+0, sn[0]); ... IINCHIP_WRITE (SUBR0+3, sn[3]);
// Wrtie MAC address to SHAR register
IINCHIP_WRITE (SHAR0+0, mac[0]); ... IINCHIP_WRITE (SHAR0+5, mac[5]);
// Set memory size for TX and RX buffer
set_MEMsize(txsize, rxsize);
```

<Fig.2.1> Setting Network Configuration

2.2 OPEN

Since UDP doesn't require a connection establishment, UDP socket can be easily created by calling the socket() function. Fig.2.2.2 shows how to create UDP socket.

```
if(socket(0, Sn_MR_UDP, 3000, 0) == 1) //create UDP socket0 with port 3000
// socket() function returns 1 for success and 0 for failure
```

<Fig.2.2> UDP socket creation

2.3 RECEIVE

Fig.2.2.3 shows the data receiving process. Before receiving data, user can check the Sn_RX_RSR which indicates the size of the received data. If any data is received, user can execute the receiving process using the recvfrom() function.

```
switch(IINCHIP_READ(Sn_SR(s)))
  case SOCK_UDP:                          // check the UDP socket is established
     if((len=getSn_RX_RSR(s)) > 0)              // check if there is any received data
      //Receive the data to data_buf if there is any received data
      len = recvfrom(0, data_buf, len, (uint8*)destip, &destport)
```

<Fig.2.3> Receiving data

2.4 SEND

Before sending out UDP datagram, user must set the destination IP address and port; once these are set, user can use the sendto() function to send out UDP datagram.

```
//Set destination IP address and port
pDestaddr[0]= 192;
pDestaddr[1]= 168;
pDestaddr[2]= 1;
pDestaddr[3]= 69;      // The IP address is depended on user(destination)
```

```
pPort = 5000;
// Send the received data (data_buf) to destination
sentLen = sendto(0, data_buf, len, pDestaddr, pPort)
// sendto() returns the length of sent data
close(0);     //close the socket0
```

<Fig.2.4> Sending UDP data

3 UDP LOOPBACK

This section will explain how to use UDP Loopback with the codes that were explained earlier. UDP Loopback is a code that can send back the exact data that was received from a peer.

Example code for UDP Loopback is as below.

```
//Set destination IP address and port
void loopback_udp(SOCKET s, uint16 port, uint8 xdata * data_buf, uint16 mode)
{
  uint16 xdata len=32;
  uint8 xdata destip[4];
  uint16 xdata destport;
  uint8 xdata i;
  switch(IINCHIP_READ(Sn_SR(s)))
  {
    case SOCK_UDP:
    if((len=getSn_RX_RSR(s)) > 0)                  // check the size of received data
          {
           // receive data from a destination
            len = recvfrom(s,data_buf,len,(uint8*)destip,&destport);
           // send the data to the destination
            if(len != sendto(s,data_buf,len,(uint8*)destip,destport))
           {
                              printf("Sendto Fail\r\n");
           }
          }
       break;

    case SOCK_CLOSED:
       close(s);                              // close the SOCKET
       socket(s,Sn_MR_UDP,port,mode);         // open the SOCKET with UDP mode
       printf("LOOPBACK_UDPStarted. CH : %d\r\n",(int)s);
       break;

    default:
       break;
  }
}
```

<Fig.3.1> UDP Loopback

Example code for UDP Loopback is as followed. First, set the network information, like IP or port.

And then handle the Sn_SR register by using switch. There are two states of Sn_SR in UDP mode. One state is SOCK_UDP, where USP socket is opened successfully and UDP communication is possible; during this state, when UDP data is received, use recfrom()function, and then sendto function to send it back. The other state is SOCK_CLOSED, where UDP socket is not open and use close() function to close the socket. And then use socket() function to reopen the socket.

4 UDP LOOPBACK Demonstration

In this section, UDP Loopback examples will be executed. After downloading the binary file of UDP loopback application to the iMCUW7100EVB, confirm the package of iMCUW7100EVB for the successful demonstration. Please refer to the 'How to make project in W7100', 'WizISP Program Guide' and 'W7100 Debugger Guide' for more information.

For the UDP loopback, user can follow the steps below.

- Confirm the testing environment. Refer to 'iMCU7100EVB User's Guide'

  Connect test PC to iMCU7100EVB by directly using UTP cable

  Connect test PC to iMCU7100EVB by directly using Serial cable

  Enable 5V power adapter to iMCU7100EVB

- Confirm the network information of Test PC as the following

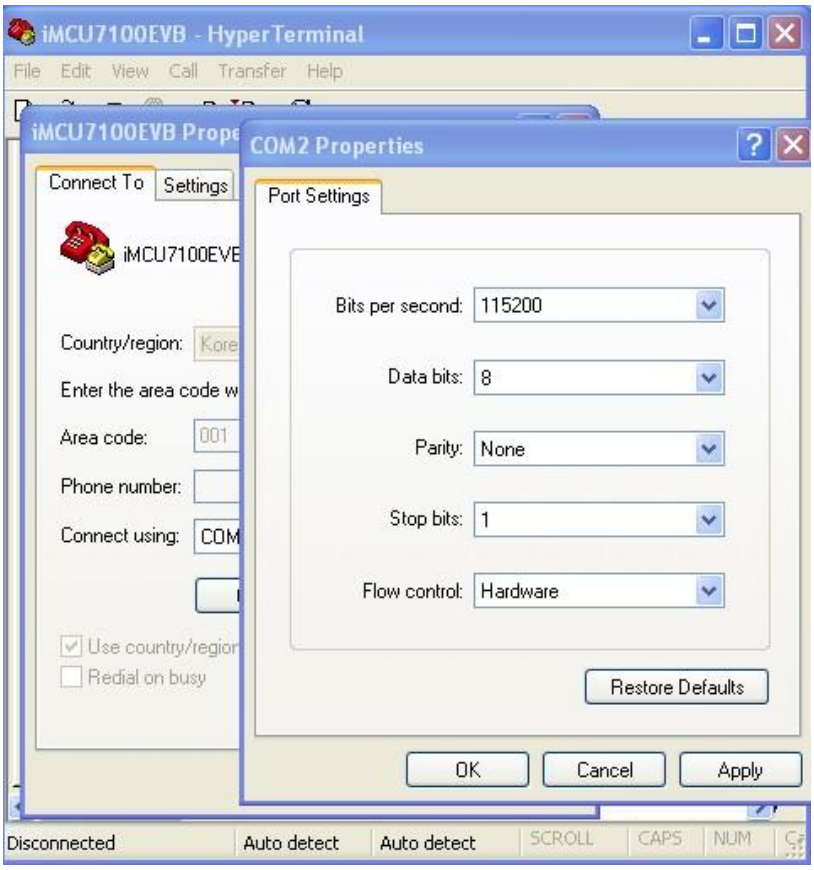
  Source IP Address : 192.168.1.2

  Gatway IP Address : 192.168.1.1

  Subnet Mask : 255.255.255.0

- Run the Hyper Terminal and AX1 program

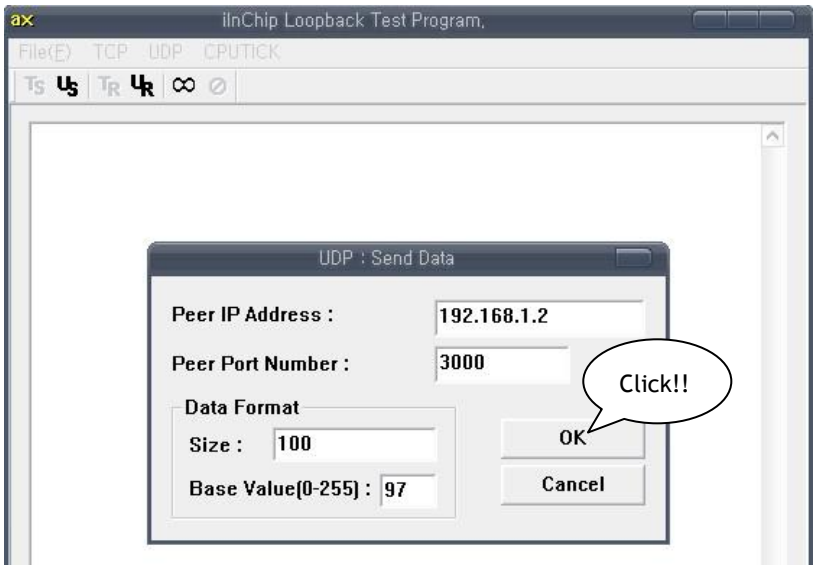
4.1 Hyper Terminal



<Fig.4.1> Hyper terminal options

Run the Hyper Terminal, and as Fig 4.1, set options for Serial communication. Then the hyper Terminal will show iMCU7100EVB's status by using serial communication.

4.2 AX1

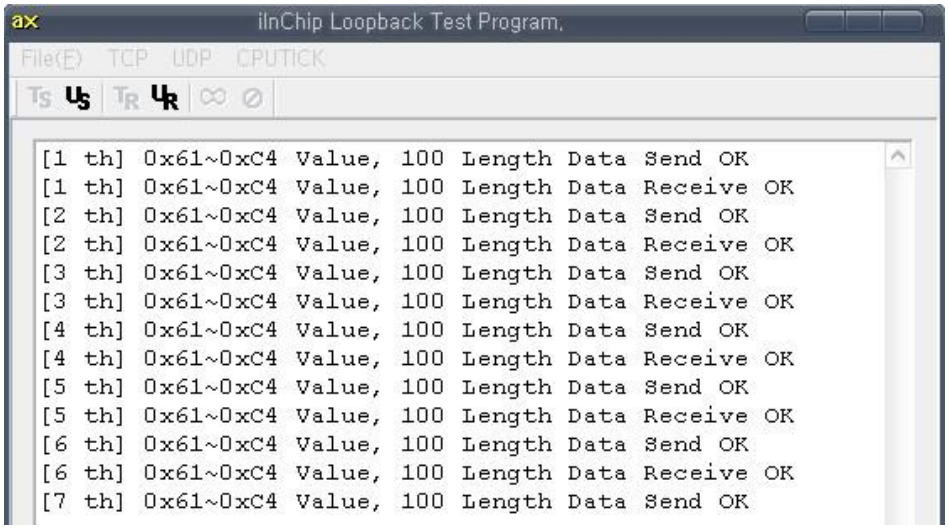


<Fig.4.2> AX1 setting for UDP

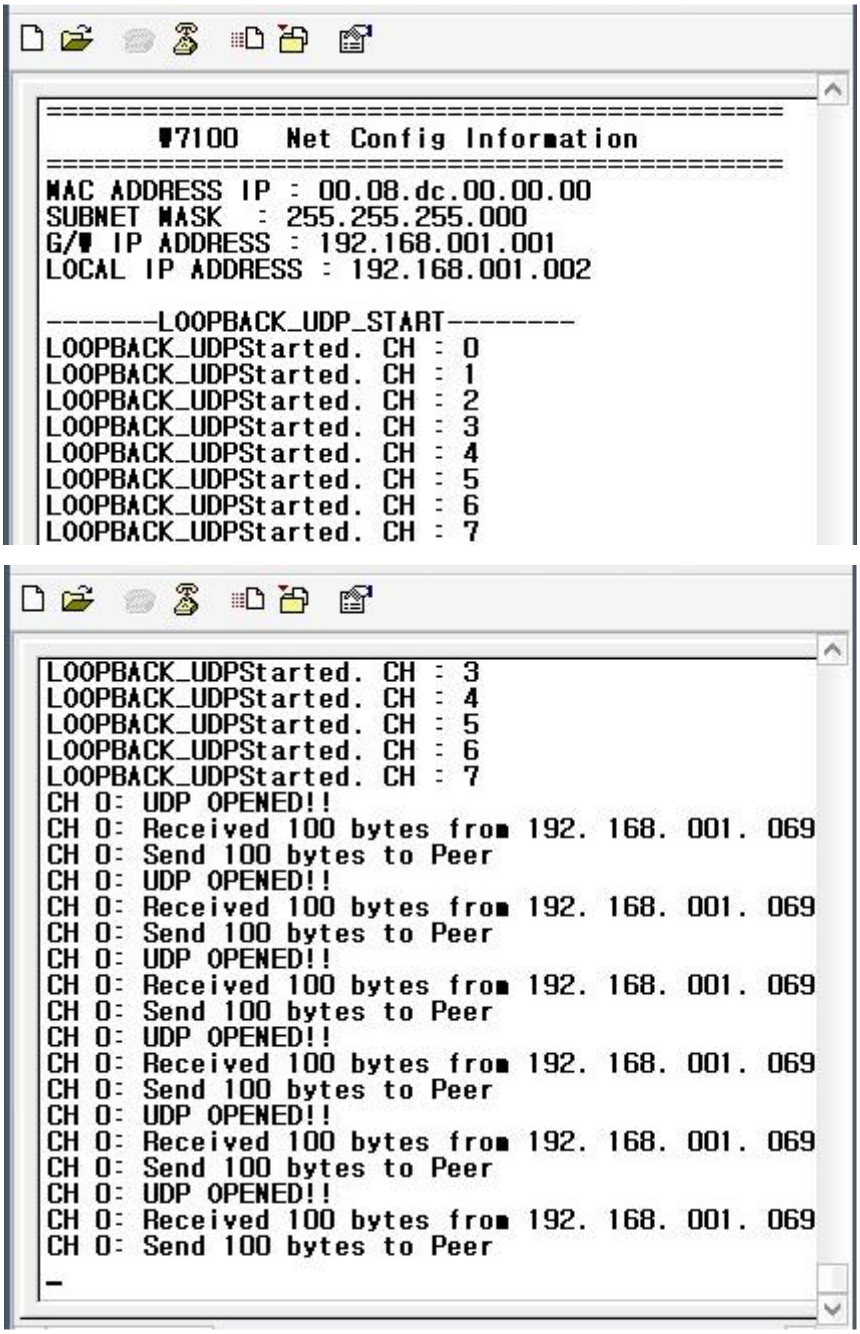
Run the AX1 program, and set the AX1 program for UDP communication, as in Fig.4.2. For more details on the AX1 program setting, please refer to the AX1 manual. For the UDP SEND, select the UDP => SEND menu on the menu bar and click the OK button. The AX1 program window will show the client PC's status.

4.3 UDP loopback Result

After all settings, click UDP send button. Then AX1 shows the process of the PC, as in the Fig.4.4. The Fig.4.4 shows the size of SEND/RECEIVE data. Also, the Hyper Terminal shows the process of server, iMCU7100EVB, as in the Fig.4.5.



<Fig.4.4> Send a temporal data using UDP



<Fig.4.5> Hyper Terminal window

Document History Information

Version	Date	Descriptions
Ver. 0.9Beta	Mar. 2009	Release with W7100 launching
Ver. 1.0	Mar. 2011	Modify for W7100A QFN 64pin package

Copyright Notice

Technical Support: support@wiznet.co.kr
Sales & Distribution: sales@wiznet.co.kr

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