How to use Character LCD for W7100

Version 1.1

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

Table of Contents

1 Character LCD

iMCU7100EVB supports the provided Character LCD connector and can use the MCU, which is embedded in W7100, to control and test the Character LCD. As shown in Fig.1, iMCU7100EVB can use the embedded Hardwired TCP/IP to control Character LCD on Network which is impossible in a normal MCU. This document will help you embody the Network Character LCD by using Internet Embedded MCU, W7100, and show you how convenient and simple W7100 is.

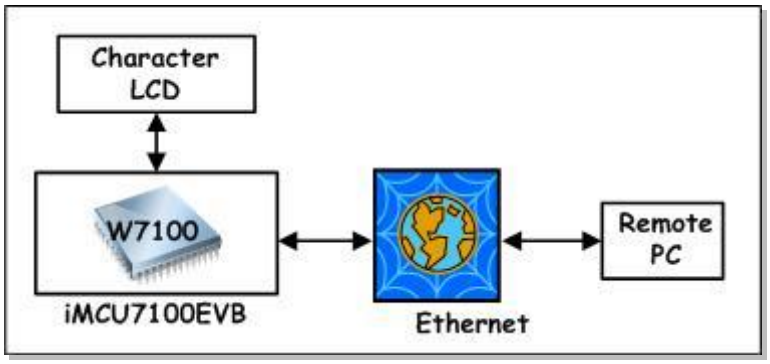
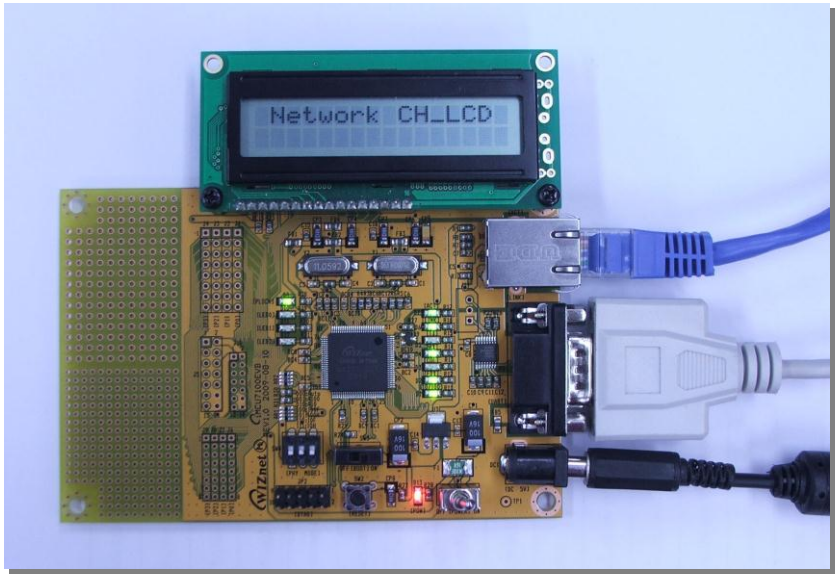


Fig. 1 Block Diagram of Network Character LCD

2 Network Character LCD

Pic.1 is a picture of iMCu7100EVB being used for Network Character LCD. The following are connected to iMCU7100EVB: Character LCD, power cable, UPT cable for Ethernet communication, Serial Cable for Debugging and Programming.



Pic. 1 Network Character LCD

2.1 Connect to Network Character LCD

User can connect to the Network Character by using the Hyper Terminal in Windows. When the "Connect To" window appears as Fig.2, input the Host address (ex. 192.168.1.165) and Port address (ex.5000). Then, set the properties of Hyper Terminal as shown as Fig.3.

Fig. 2 Connect To Network Character LCD

Fig. 3 Setting Hyper terminal Properties

2.2 Send message to Network Character LCD

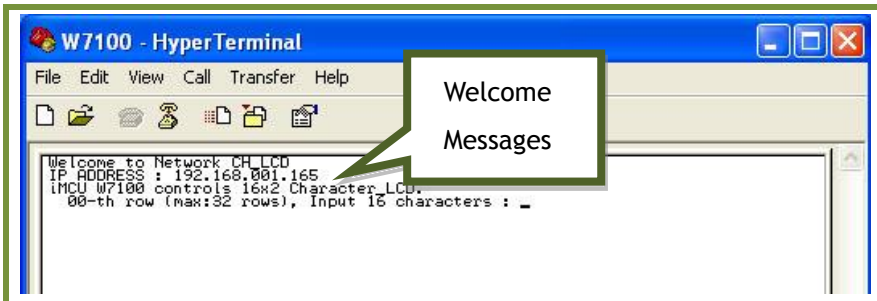


Fig. 4 **Receive welcome message**

Once the connection of Network Character LCD is complete, the Welcome message and Input Message will appear, as shown as Fig.4.

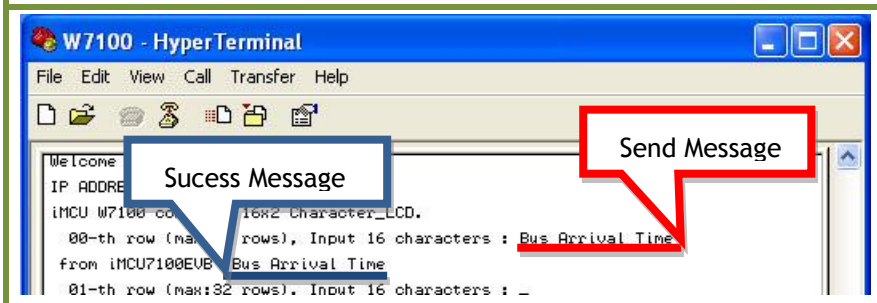


Fig. 5 **Send Data/Success message**

Input the string and if it's sent, receive the exact same string from the Character LCD.

2.3 Display messages

Messages like bus arrival times can be displayed in the Network Character LCD. The figure and pictures below are demonstrations using the Network Character LCD showing the bus arrival time. Fig.6 shows data from Remote PC being received to the Network Character LCD. Pic.2 shows the Network character LCD scrolling up received strings from the Remote PC.

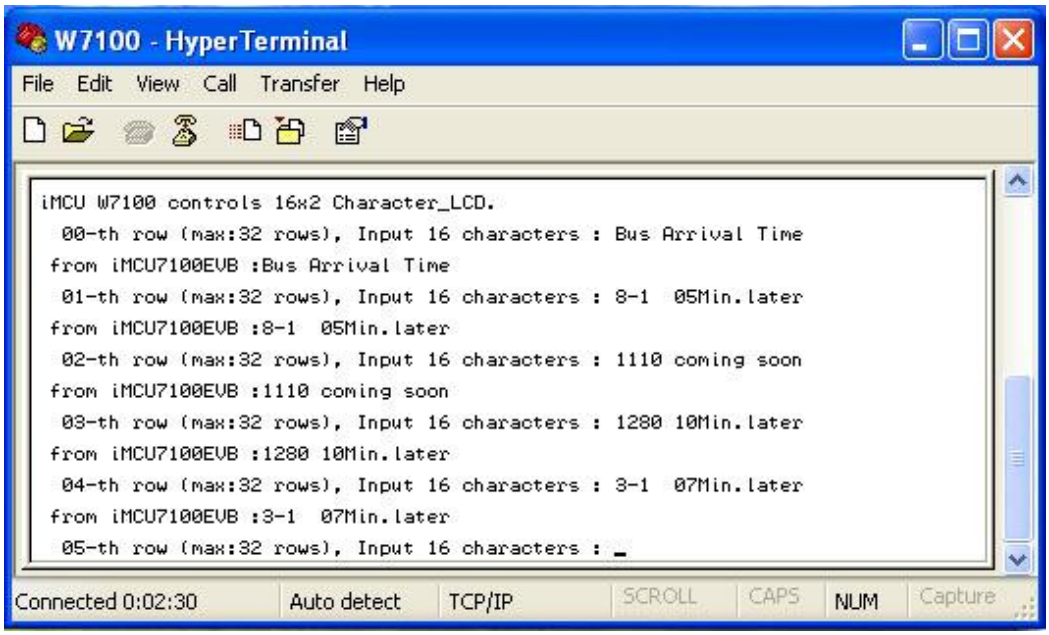
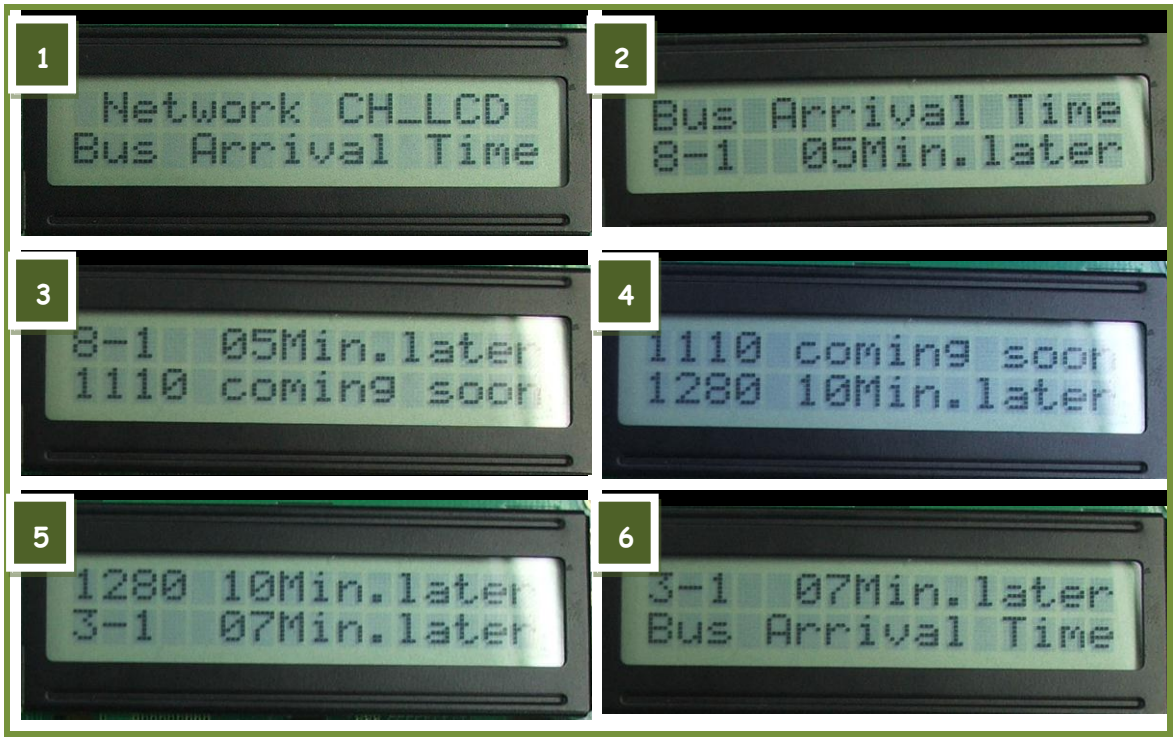


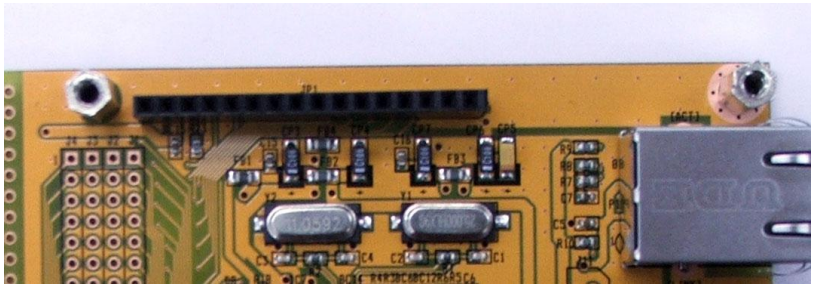
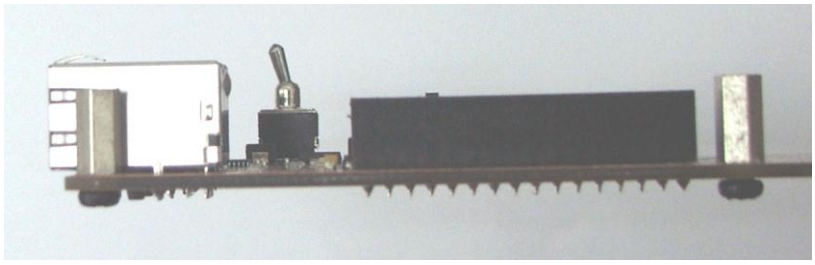
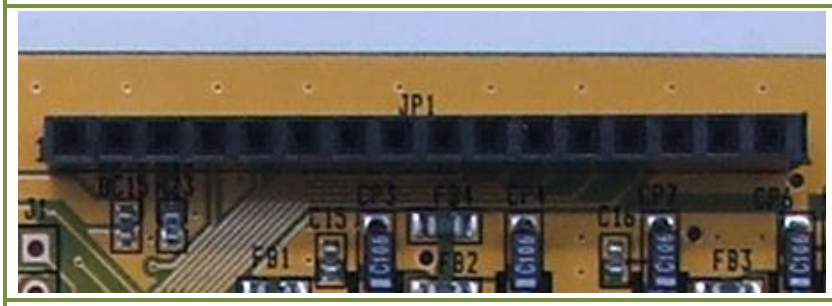
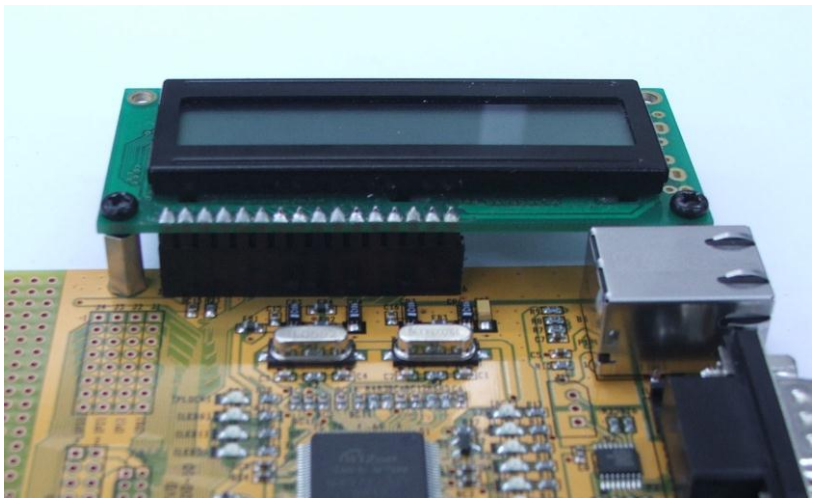
Fig. 6 **Demonstration**



Pic. 2 **Capture screens**

3 Hardware Configuration

3.1 Configuration

<table>
<tr><td colspan="2">Preparations : 4 bolts, 2 supporters, Character LCD (12characters*2lines), iMCU7100EVB</td></tr>
<tr><td>  </td><td rowspan="2">Set and fix the supporter to iMCU7100EVB as shown in the picture.</td></tr>
<tr><td>  </td></tr>
<tr><td> 
</td><td>This picture is the Character LCD Connector (JP1). The first left Port Pin number is 1.</td></tr>
<tr><td>  </td><td>Check the Character LCD and PIN number of JP1. Connect as shown in the picture and fix it with bolts.</td></tr>
</table>

3.2 Character LCD Connector and PIN Description

Fig.6 shows the circuit of Character LCD Connecter(JP1). Tab.1 lists the descriptions of Character LCD PINs.

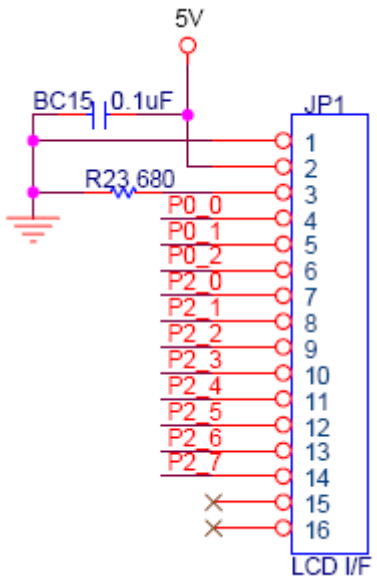


Fig. 6 Character LCD Connecter(JP1)

PIN#	PIN NAME	DIR.	Description
1	GND / VSS		Signal Ground
2	5V / VDD	I	LCD Power Supply
3	V0 / V0	I	Voltage for LCD drive
4	P0_0 / RS	I	Data/Instruction register select
5	P0_1 / RW	I	Read / Write
6	P0_2 / E	I	Enable signal,start data read / write
7 ~ 14	P2_0 ~ P2_7 / DB0 ~ DB7	I/O	Data Bus Line
15	Not Connect / LED A	O	LED Anode, power supply
16	Not Connect / LED K	O	LED Cathode, ground 0V

Tab. 1 Character LCD PIN Description

4 Software

The Network Character LCD operates in TCP server mode. In TCP server mode, W7100 waits for the Client to connect (LISTEN), and when the Client is connected (ESTABLISHED), data transmission with the Client is possible. (For more details, please refer to 'Internet Embedded MCU W7100 Datasheet 9.2.1.1' or 'How to implement TCP in W7100.')

Expand the TCP Loopback example codes, which were embodied in 'How to implement TCP in W7100,' and embody Network Character LCD. The algorithm of Network Character LCD is shown in Fig.7.

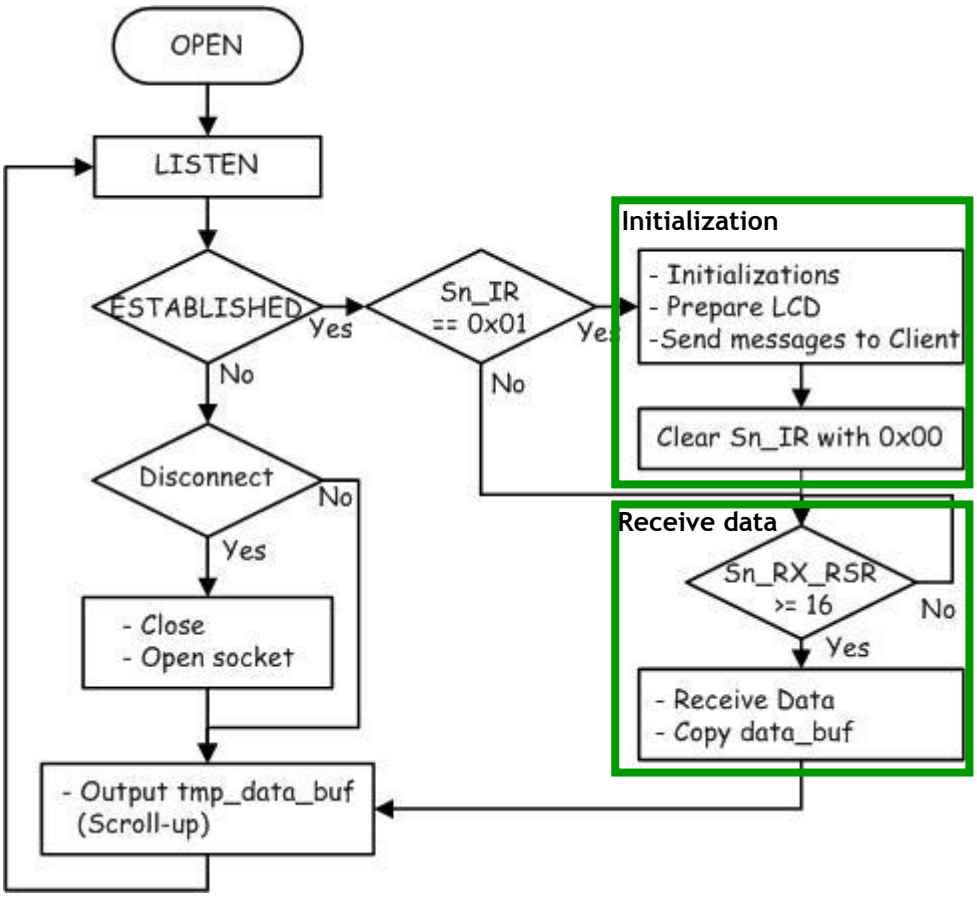


Fig. 7 Flow charts

4.1 Initialization

The value of the Socket Status Register (Sn_SR) is SOCK_ESTABLISHED(0x01) and the value of Socket when the software is initialized. Interrupt Register is Sn_IR_CON (=0x01). This value also means that the Client is successfully connected to the server.

Codes below are used in the software.

- to send messages to the client : use code 51-53, 59-61
- to initialize the tmp_data_buf, which saves data_buf : use code 44

- to clear character LCD : use code 47-48
- to avoid the repetition of the initialization process, clear the value of Sn_IR : use code 56

lcd_app.c

```
37:  /* If Socket Interrupt Register valure is Sn_IR_CON(0x01)*/
38:  if(getSn_IR(s) & Sn_IR_CON)
39:  {
40:  /* Initialize coounter, index, and buffers.     */
41:  recv_cnt = 0;
42:  dis_idx = 0;
43:  buf_idx = 0;
44:  memset((void*) tmp_data_buf, '\0', sizeof(tmp_data_buf));

46:  /* Clear LCD & Output Init. messege */
47:  lcd_command(LCD_CLEAR);
48:  evb_set_lcd_text(0," Network CH_LCD ");

50:  /* Send welcome messeges to remote PC      */
51:  send(s, welcome_msg1, sizeof(welcome_msg1));
52:  send(s, str, sizeof(str));
53:  send(s, welcome_msg2, sizeof(welcome_msg2));

55:  /* Set Sn_IP to Sn_IR_CON */
56:  setSn_IR(s, Sn_IR_CON);

58:  /* Send input msg to remote PC */
59:  sprintf(data_buf,"\r\n   %.2d\0",(uint16)buf_idx);
60:  send(s, data_buf, strlen(data_buf));
61:  send(s, drt_msg1, sizeof(drt_msg1));
62:  }
```

Code 1 Initialization

4.2 Receive Data

The 'Receive Read Buffer Size Register' is used to receive strings that must have a data length of 16. If the length of Receive Read buffer is larger than 16, use recv() to receive data (code 65-68). The data that is received in data_buf is copied into tmp_data_buf according to buf_idx (buffer_index). Tmp_data_buf has the size of 17 x32. In other words, 32 strings of 16 characters and one Null character can be saved. If the reception is successful and saved to tmp_data_buf, send a success message to the client (code 84-86).

lcd_app.c

```
64:  /* check RX data */
65:  if ((len = getSn_RX_RSR(s)) >= 16)    // If the received data length is larger than 16
66:  {
67:    /* read the received data */
68:    len = recv(s, data_buf, 16);
69:    /* Input Null ,'0', to divide data_bufs */
70:    data_buf[len]=0;
71:
72:    /* Copy data_buf to tmp_data_buf with buf idx */
73:    memcpy(tmp_data_buf[buf_idx],data_buf,len);

74:    /* send msg to Remote PC*/

79:    /* increase and update buf_idx */
80:    buf_idx++;
81:    buf_idx = buf_idx % LCD_SCROLL_CNT;

83:    /* Send input msg to remote PC */
84:    sprintf(data_buf,"\r\n    %.2d\0",(uint16)buf_idx);
85:    send(s, data_buf, strlen(data_buf));
86:    send(s, drt_msg1, sizeof(drt_msg1));

88:    if(recv_cnt++ > LCD_SCROLL_CNT) recv_cnt=LCD_SCROLL_CNT;
90:  }
```

Code 2 Receive process

4.3 Display

The source codes for scrolling tmp_data_buf are as below; evb_set_lcd_text() function, which is defined in the Character LCD driver source, lcd.c, is used. The factors of evb_set_lcd_text() are a line (0 or 1) and 16 strings. Also, as mentioned above, tmp_data_buf can save at most 32 columns, and lcd_scroll() function is used to scroll all the strings in the 2line Character LCD. Dis_idx (display_index) is used to scroll 32 strings in the 2line Character LCD (code 127-128).

lcd_app.c

```
118: /* Scroll Up*/
119: void lcd_scroll()
120: {
121:    if((recv_cnt)<3){
122:    evb_set_lcd_text(1,tmp_data_buf[dis_idx]);//Output tmp_data_buf to Ch_LCD
123:    }else{
124:    evb_set_lcd_text(0,tmp_data_buf[dis_idx]);//Output tmp_data_buf to Ch_LCD
125:
126:       /*for indexing of display */
127:    dis_idx = (dis_idx + 1) % (recv_cnt);
128:    dis_idx %= LCD_SCROLL_CNT;
129:
130:    evb_set_lcd_text(1,tmp_data_buf[dis_idx]);//Output tmp_data_buf to Ch_LCD
131:
132:    wait_10ms(100); // delay for 1Sec.
133:    }
134: }
```

Code 3 Display

Document History Information

Version	Date	Descriptions
Ver. 0.9Beta	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