How to implement DHCP in W5200

Version 0.9ßeta

`

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

Table of Contents

1 Introduction

DHCP server uses DHCP to assign details like IP address or settings to the client that uses the DHCP. This application note will explain how to implement DHCP client by using W5200.

2 DHCP

DHCP uses UDP (User Datagram Protocol) through Transport Layer, and communicates with the DHCP server using UDP broadcast. Fig.1 describes the communication between the DHCP server and client.

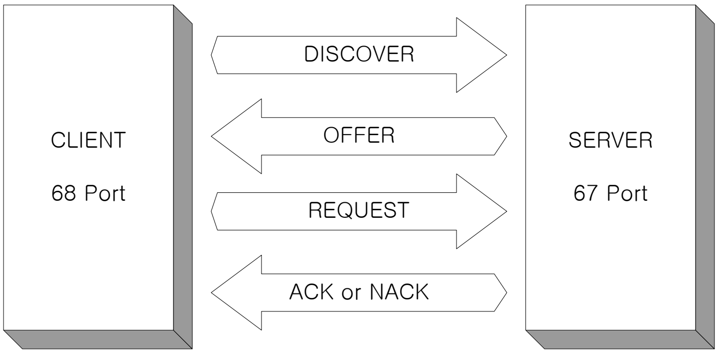


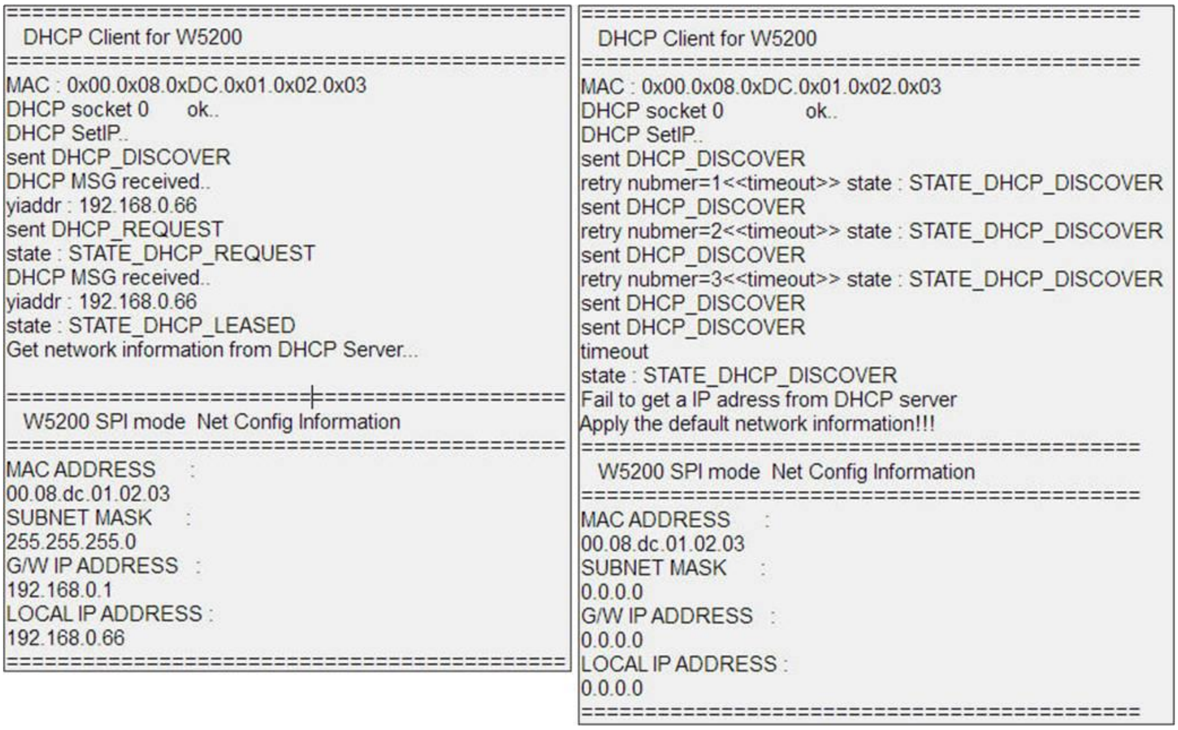
Fig. 1 DHCP Message

DHCP Client broadcasts the 'DISCOVERY message' on the connected network. If a DHCP Server exists on the computer network, the DHCP Server will receive the 'DISCOVERY message' and send the 'OFFER message' to the DHCP client. 'OFFER message' consists of an IP that can be used by DHCP client, Gateway(G/W), network information that is the same as the DNS server IP, and information like lease time. DHCP Client can acknowledge the DHCP Server by receiving the 'OFFER message,' and send the 'REQEUST message' to use the information suggested by the server. Then after the 'REQUEST message' is received, the DHCP Server decides whether the Lease duration (the period of time where the DHCP client IP address can be used) and Network configuration can be used or not. If the above information can be used, the DHCP Server will send an ACK message to the DHCP Client; or a NACK message if the information can't be used.

3 DHCP Client

3.1 Demonstration

Fig.2 shows the result of DHCP Client Test by connecting W5200E01-M3 and network gear. The DISCOVER/OFFER/REQUEST/ACK messages can be checked. The DHCP Client network configuration assigned by the DHCP Server can be checked also.



```
============================================
  DHCP Client for W5200
============================================
MAC : 0x00.0x08.0xDC.0x01.0x02.0x03
DHCP socket 0     ok..
DHCP SetIP..
sent DHCP_DISCOVER
DHCP MSG received..
yiaddr : 192.168.0.66
sent DHCP_REQUEST
state : STATE_DHCP_REQUEST
DHCP MSG received..
yiaddr : 192.168.0.66
state : STATE_DHCP_LEASED
Get network information from DHCP Server...

=======================+====================
  W5200 SPI mode  Net Config Information
============================================
MAC ADDRESS     :
00.08.dc.01.02.03
SUBNET MASK     :
255.255.255.0
G/W IP ADDRESS  :
192.168.0.1
LOCAL IP ADDRESS :
192.168.0.66
============================================
```

```
============================================
  DHCP Client for W5200
============================================
MAC : 0x00.0x08.0xDC.0x01.0x02.0x03
DHCP socket 0           ok..
DHCP SetIP..
sent DHCP_DISCOVER
retry nubmer=1<<timeout>> state : STATE_DHCP_DISCOVER
sent DHCP_DISCOVER
retry nubmer=2<<timeout>> state : STATE_DHCP_DISCOVER
sent DHCP_DISCOVER
retry nubmer=3<<timeout>> state : STATE_DHCP_DISCOVER
sent DHCP_DISCOVER
sent DHCP_DISCOVER
timeout
state : STATE_DHCP_DISCOVER
Fail to get a IP adress from DHCP server
Apply the default network information!!!
============================================
  W5200 SPI mode  Net Config Information
============================================
MAC ADDRESS     :
00.08.dc.01.02.03
SUBNET MASK     :
0.0.0.0
G/W IP ADDRESS  :
0.0.0.0
LOCAL IP ADDRESS :
0.0.0.0
============================================
```

Figure 2 DHCP processing (Left:success & Right:timeout)

If there is no DHCP server in you network, DHCP processing will time out (See Figure 2, right). By default, W5200 will retry 3 times until the response of DHCP_DISCOVER (Offer) is received, otherwise W5200 will use zero as network parameters (IP: 0.0.0.0, subnet: 0.0.0.0, G/W: 0.0.0.0)

4. Software

3.2 Socket

3.2.1 OPEN

Since the internet connection is not required between the DHCP Server and DHCP Client, DHCP needs to be opened by UDP. Use Sn_MR_UDP as shown below to OPEN the socket (line886).

```
dhcp.h
/* UDP port numbers for DHCP */
25          #define DHCP_SERVER_PORT          67          /* from server to client */
26          #define DHCP_CLIENT_PORT          68          /* from client to server */


dhcp.c
886:  socket(SOCK_DHCP, Sn_MR_UDP, DHCP_CLIENT_PORT, 0x0)
```

4.1.2 SEND

The user must set the other user's IP address and port before sending the DHCP data. The DHCP_SERVER_PORT is fixed to 67. When DISCOVER or REQUEST is operated, The IP address uses the BROADCAST address (255.255.255.255) to send data. After receiving the DHCP Server IP from the DHCP message, use the parsed DHCP Server IP (**DHCP_SIP[]**) to send data (line287).

```
dhcp.c
/* DST IP : BroadCasting*/
      270:           if(dhcp_state < STATE_DHCP_LEASED)
                     {
                               ip[0] = 255;
                               ip[1] = 255;
                               ip[2] = 255;
                               ip[3] = 255;
                      }
                      else
                      {
                               ip[0] = DHCP_SIP[0];
                               ip[1] = DHCP_SIP[1];
                               ip[2] = DHCP_SIP[2];
                               ip[3] = DHCP_SIP[3];
                      }
```

```
    // send MSG to Broadcasting
                sendto(s, (u_char*)pRIPMSG, sizeof(RIP_MSG), ip, DHCP_SERVER_PORT)
```

4.1.3 RECEIVE

This section is identical with the RECEIVE section of UDP. Please refer to 'How to implement UDP in W5200.'

```
dhcp.c
               len = recvfrom(s, (u_char *)pRIPMSG, length, svr_addr, &svr_port);
```

4.1.4 DHCP Message Format

The structure below is composed of codes that work with the DHCP message. (Please refer to the RFC1541 document for more details on DHCP Message Format and each Field). The fileds below the ciaddr field are used to send network information; and the options field is used to send message type and information like client identifier.

```
dhcp.h
/**
       * @brief for the DHCP message
   */
   typedef struct _RIP_MSG
   {
        u_char     op;
        u_char     htype;
        u_char     hlen;
        u_char     hops;
        u_long     xid;
        u_int      secs;
        u_int      flags;
        u_char     ciaddr[4];
        u_char     yiaddr[4];
        u_char     siaddr[4];
        u_char     giaddr[4];
        u_char     chaddr[16];
        u_char     sname[64];
        u_char     file[128];
        u_char     OPT[312];
  }RIP_MSG;
```

4.2 DHCP Functions

The functions for implementing the DHCP client are listed under dhcp.h and dhcp.c below.

```
dhcp.h
        void init_dhcp_client(SOCKET s, void (*ip_update)(void),void (*ip_conflict)(void));   // Initialize the
        DHCP client
        u_int init_dhcpc_ch(SOCKET s);          // Initialize the socket for DHCP client
        u_int getIP_DHCPS(void);                //Get the network configuration from the DHCP server
         void check_DHCP_state(SOCKET s);           // Check the DHCP state
dhcp.c
        static void send_DHCP_DISCOVER(SOCKET s);   /* Send the discovery message to the DHCP server */
        static void send_DHCP_REQUEST(SOCKET s);    /* Send the request message to the DHCP server */
        static char parseDHCPMSG(SOCKET s, u_int length);    /*  Receive  the  message  from  DHCP  server
        and parse it. */
        static void reset_DHCP_time(void);          /* Initialize DHCP Timer */
        static void check_DHCP_Timeout(void);       /* Check DHCP Timeout   */
        static void set_DHCP_network(void);         /* Apply the leased IP address to LP-NetCAM II */
        static void proc_ip_conflict(void);         /* called when the leased IP address is conflict */
```

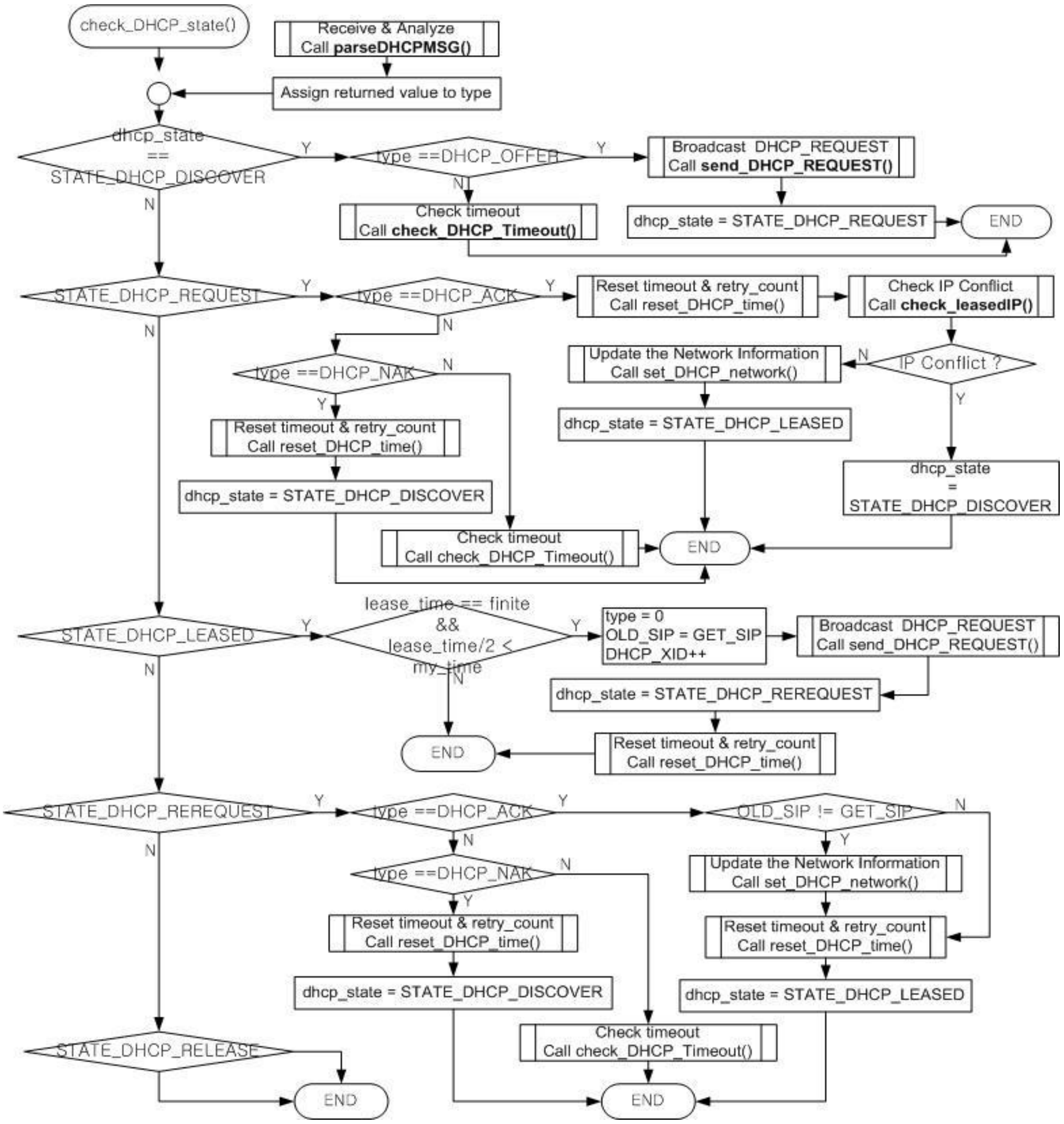


Fig. 2 check_dhcp function

Fig.3 shows the flowchart of check_dhcp(). DHCP_OFFER, DHCP_ACK, and DHCP_REQUEST are operated depending on the change (transition) of dhcp_state. Fig.4 shows the flowchart of parseDHCPMSG(). parseDHCPMSG() is used to phishing (parsing) DHCP messages. Fig.5 shows the flowchart of check_LeaseIP() and send_DHCP_REQUEST().

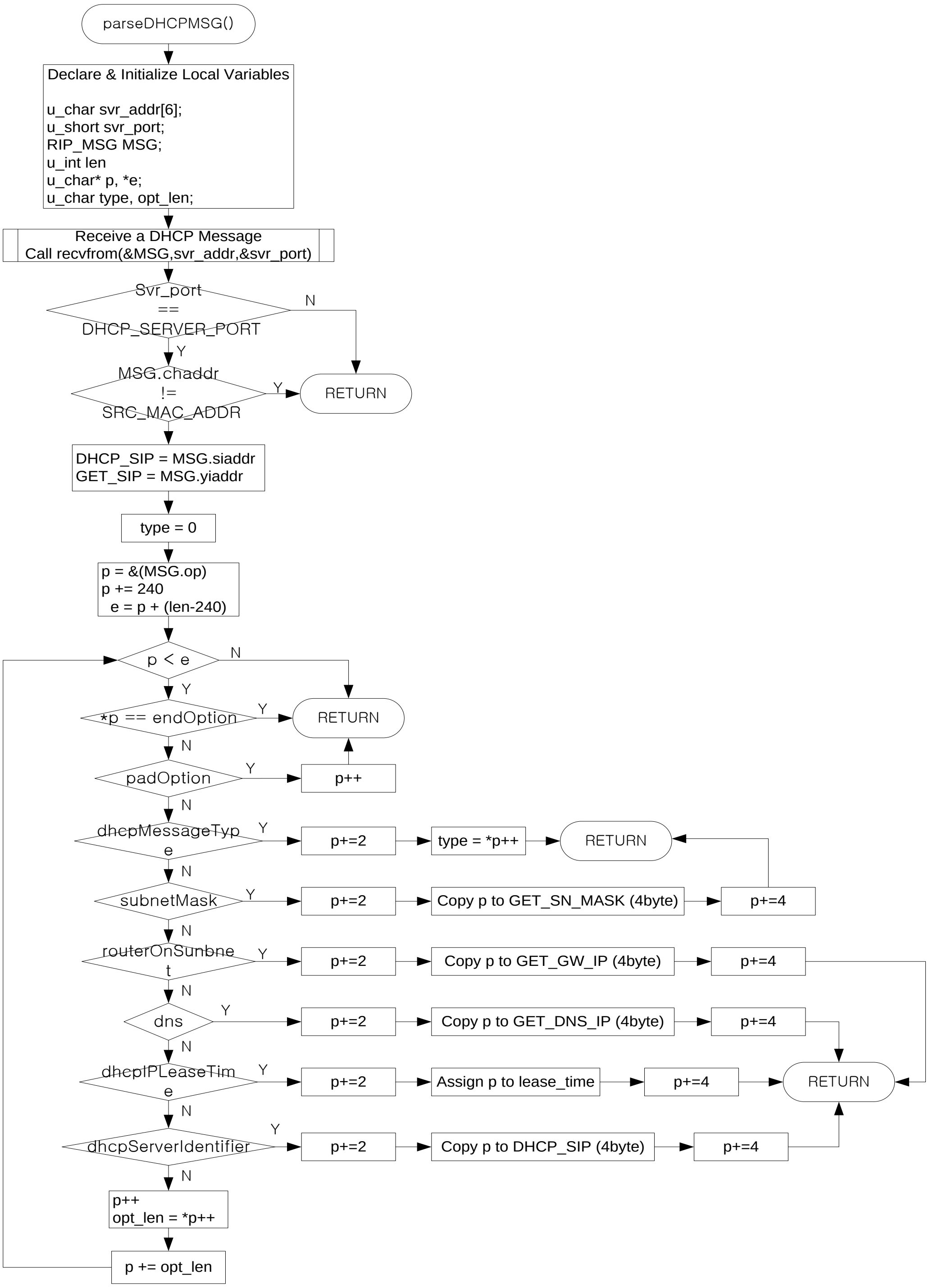


Fig. 3 parseDHCPMSG() function flowchart

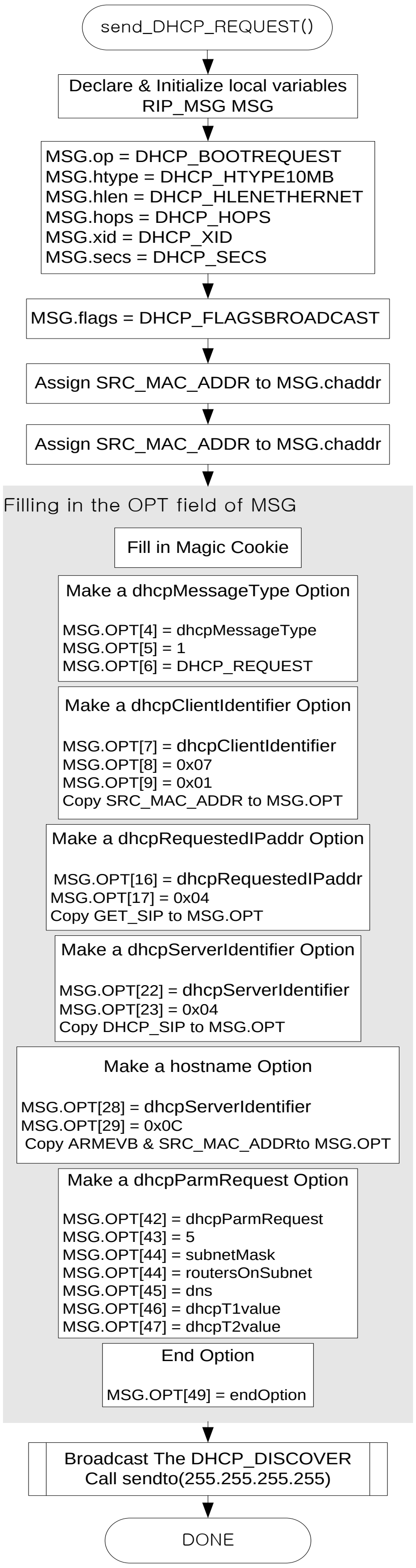

send_DHCP_REQUEST()
Declare & Initialize local variables
RIP_MSG MSG
MSG.op = DHCP_BOOTREQUEST
MSG.htype = DHCP_HTYPE10MB
MSG.hlen = DHCP_HLENETHERNET
MSG.hops = DHCP_HOPS
MSG.xid = DHCP_XID
MSG.secs = DHCP_SECS
MSG.flags = DHCP_FLAGSBROADCAST
Assign SRC_MAC_ADDR to MSG.chaddr
Assign SRC_MAC_ADDR to MSG.chaddr
Filling in the OPT field of MSG
Fill in Magic Cookie
Make a dhcpMessageType Option
MSG.OPT[4] = dhcpMessageType
MSG.OPT[5] = 1
MSG.OPT[6] = DHCP_REQUEST
Make a dhcpClientIdentifier Option
MSG.OPT[7] = dhcpClientIdentifier
MSG.OPT[8] = 0x07
MSG.OPT[9] = 0x01
Copy SRC_MAC_ADDR to MSG.OPT
Make a dhcpRequestedIPaddr Option
MSG.OPT[16] = dhcpRequestedIPaddr
MSG.OPT[17] = 0x04
Copy GET_SIP to MSG.OPT
Make a dhcpServerIdentifier Option
MSG.OPT[22] = dhcpServerIdentifier
MSG.OPT[23] = 0x04
Copy DHCP_SIP to MSG.OPT
Make a hostname Option
MSG.OPT[28] = dhcpServerIdentifier
MSG.OPT[29] = 0x0C
Copy ARMEVB & SRC_MAC_ADDRto MSG.OPT
Make a dhcpParmRequest Option
MSG.OPT[42] = dhcpParmRequest
MSG.OPT[43] = 5
MSG.OPT[44] = subnetMask
MSG.OPT[44] = routersOnSubnet
MSG.OPT[45] = dns
MSG.OPT[46] = dhcpT1value
MSG.OPT[47] = dhcpT2value
End Option
MSG.OPT[49] = endOption
Broadcast The DHCP_DISCOVER
Call sendto(255.255.255.255)
DONE

Fig. 4 send_DHCP_REQUEST() funciton flowchart

Document History Information

Version	Date	Descriptions
Ver. 0.9βeta	Sep, 2011	

Copyright Notice

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

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