



HelloDevice

( HD1320E/1320/1321)

Version 1.1

가

가

2001 Sena Technologies, Inc. All rights reserved.

, HelloDevice / 가

**1**

**2**

**3**

3.1 HD1320/1320E

3.1.1

3.1.2

3.1.3 RS232

3.2 HD1321

3.2.1

3.2.2

3.3

**4**

4.1

4.2

4.2.1 HD1320/1320E

4.2.2 HD1321

4.3

4.3.1 IP

4.3.2 IP

4.4

4.4.1

4.4.2

4.4.3 RS232

**5**

5.1

5.2

5.2.1 TCP

5.2.2 TCP

5.2.3

## **6 RS232-TCP/IP**

6.1 TCP

6.1.1 TCP

6.1.2 TCP

6.1.3 TCP

/

6.2 RS232

## 1.

## ▶ HD132x

		HD1320E	HD1320	HD1321
	PCB		PCB	PCB
	7.5V	○	-	-
	RS232	○	-	-
	CD-ROM	○	-	-
		HelloDevice <a href="http://www.sena.com/korean/hellodevice/download.shtml">http://www.sena.com/korean/hellodevice/download.shtml</a>		
		○	-	-



- = 7.5 V ~ 9 V DC (HD1320/1320E), 5V DC  $\pm$ 10% (HD1321)
- = 150mA
- = 0 ~ 95%
- = 5 ~ 55



HelloDevice

137-130

210

: (02) 573-7772

: (02) 573-7710

: [support@sena.com](mailto:support@sena.com): <http://www.sena.com>

**2.**

HD132x , RS232

TCP/IP

RS232

TCP/IP

, TCP/IP

RS232

HD132x HD1320E/1320/1321 가

2.1

. RS232 TCP/IP

HD1320E

HD1320

, HD1321

DIP

	HD1320/HD1320E/HD1321
<b>CPU</b>	Ubicom Sx52BD (8-Bit , 50 MIPS)
	32 KB SRAM
	10 Base-T (IEEE802.3)
	1 RS-232, 115 Kbps
	DHCP
	TCP
	IP / ICMP / ARP
	(IEEE802.3)
	HelloDevice ( 95/98/NT/2000 ) : IP ,

**2.1 HelloDevice 132X**

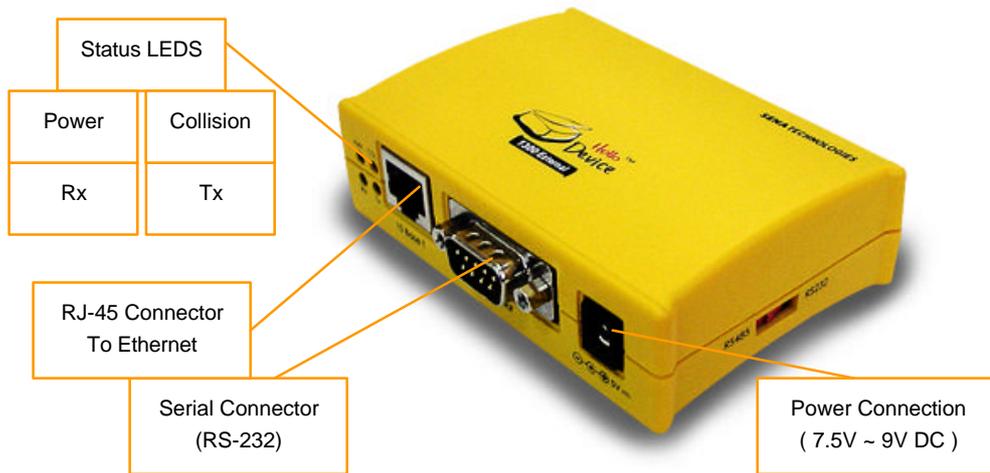
### 3.

#### 3.1 HD1320/HD1320E

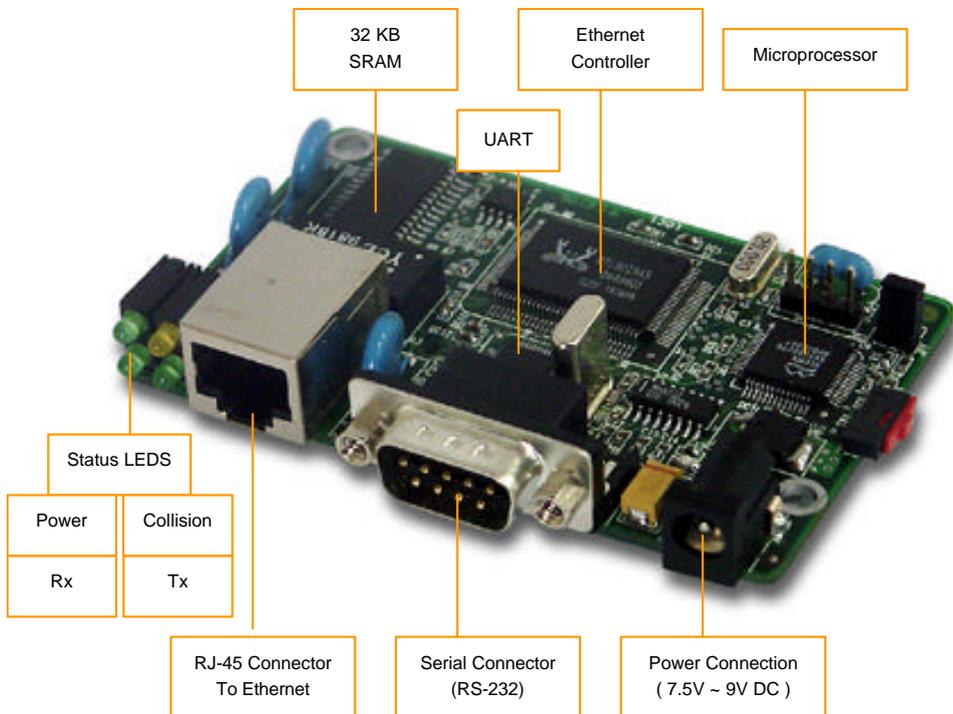
HD1320E

3.1 (a), (b)

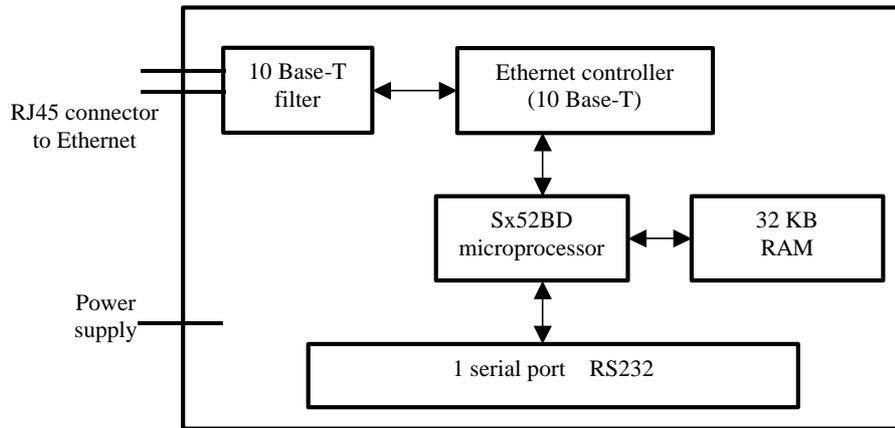
3.2



3.1(a) HD1320E



3.1(b) HD1320/1320E



### 3.2 HD1320

#### 3.1.1

- = 7.5 V ~ 9V DC
- = 150mA

#### 3.1.2

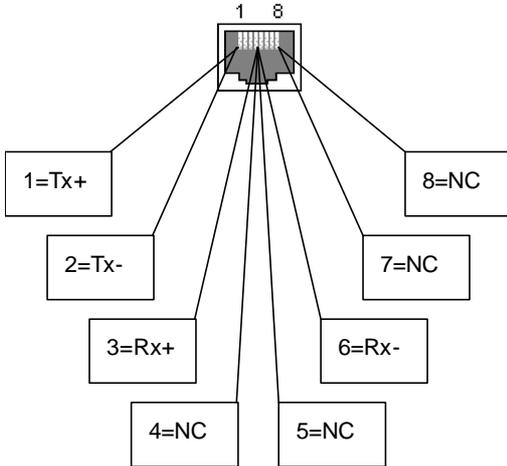
HelloDevice RJ45 , 100m  
가 .

#### (1)

- :
- RealTek Full-duplex : RTL8019AS
  - IEEE802.3 10Base-T
  - NE2000

## (2) RJ45

- AT&T258 Shield



Pin		
1	Tx+	White with orange
2	Tx-	Orange
3	Rx+	White with green
4	Not used	Blue
5	Not used	White with blue
6	Rx-	Green
7	Not used	White with brown
8	Not used	Brown

### 3.3 RJ45

## (3) LED

LED Tx, Rx, Collision, Power LED 4 가 ( 3.1),

- **Power LED**

HelloDevice ON

- **Rx LED**

/

- **Tx LED**

HelloDevice

1

/

PC 가

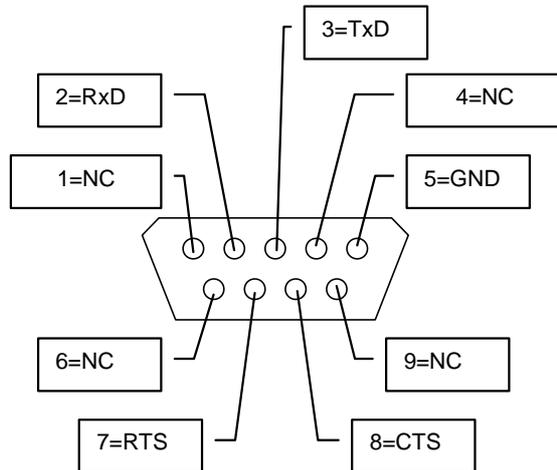
/

, ping

- **Collision LED**

### 3.1.3 RS232

- RS232 DB9
- 150bps ~ 115Kbps



Pin			
1	NC	Not connected	-
2	RxD	Receive Data	Data Input
3	TxD	Send Data	Data Output
4	NC	Not connected	-
5	GND	Signal Ground	-
6	NC	Not connected	-
7	RTS	Ready To Send	Handshake Output (+12V=Ready to Rx, -12V=Not Ready to Rx)
8	CTS	Clear To Send	Data sent
9	NC	Not connected	Data sent only at +3V ~ 12V

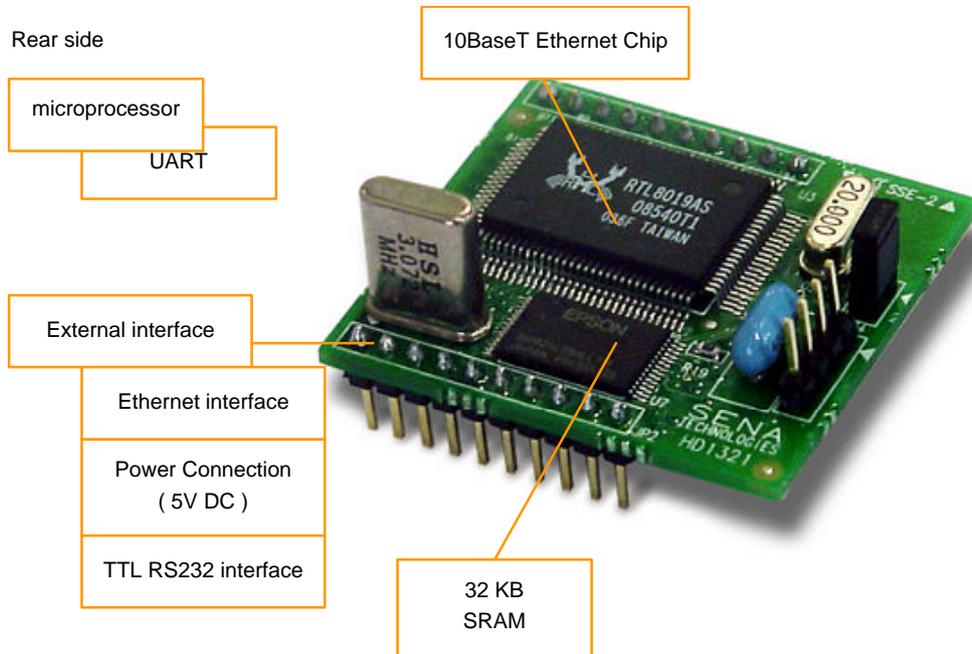
### 3.4 HelloDevice DSUB

### 3.2. HD1321

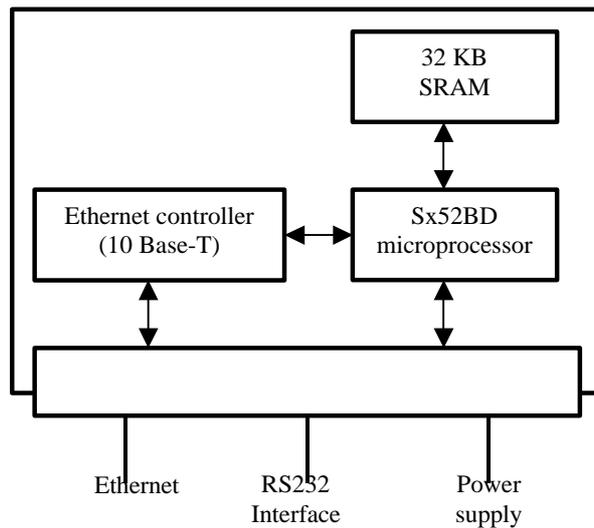
HD1321

3.5

3.6



### 3.5 HD1321



### 3.6 HD1321

### 3.2.1

- = 5 V DC  $\pm 10\%$
- = 150mA

### 3.2.2

HD1321 , JP1, JP2 20 , UART 가 .  
 3.7 , 3.1 HD1321

	#			
JP1	1	GND		
	2	Reset	System Reset	Low Active
	3 ~ 5	LED2 ~ LED0	LED LED0: Tx, LED1: Rx, LED2: Collision	
	6	TPOut-	TPOut-	
	7	TPOut+	TPOut+	
	8	TPIn+	TPIn+	
	9	TPIn-	TPIn-	
	10	Vcc	5V	
JP2	1	Vcc	5V	
	2	NC		TTL Level
	3	DSR	RS232 Data Set Ready	
	4	CTS	RS232 Clear To Send	
	5	DTR	RS232 Data Terminal Ready	
	6	RTS	RS232 Ready To Send	
	7	RxD	RS232 Data Input	
	8	TxD	RS232 Data Output	
	9	NC		
	10	GND		

### 3.1 HD1321

GND	Reset	LED2	LED1	LED0	TPOut-	TPOut+	TPIn+	TPIn-	Vcc
-----	-------	------	------	------	--------	--------	-------	-------	-----

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

**JP1**



1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

**JP2**

Vcc	NC	DSR	CTR	DTR	RTS	RxD	TxD	NC	GND
-----	----	-----	-----	-----	-----	-----	-----	----	-----

### 3.7 HD1321

3.3

4 WORD 가 ,  
 (OSI : Open System Interconnection) TCP/IP  
 RS232 가 . 3.2 , OSI 7

		HelloDevice			
7	Application	RS232		DHCP	
6	Presentation				
5	Session				
4	Transport				
3	Network	TCP		UDP	
2	Data link	IP / ICMP			ARP
1	Physical layer	(IEEE802.3)			

3.2 OSI 7 HelloDevice 132X

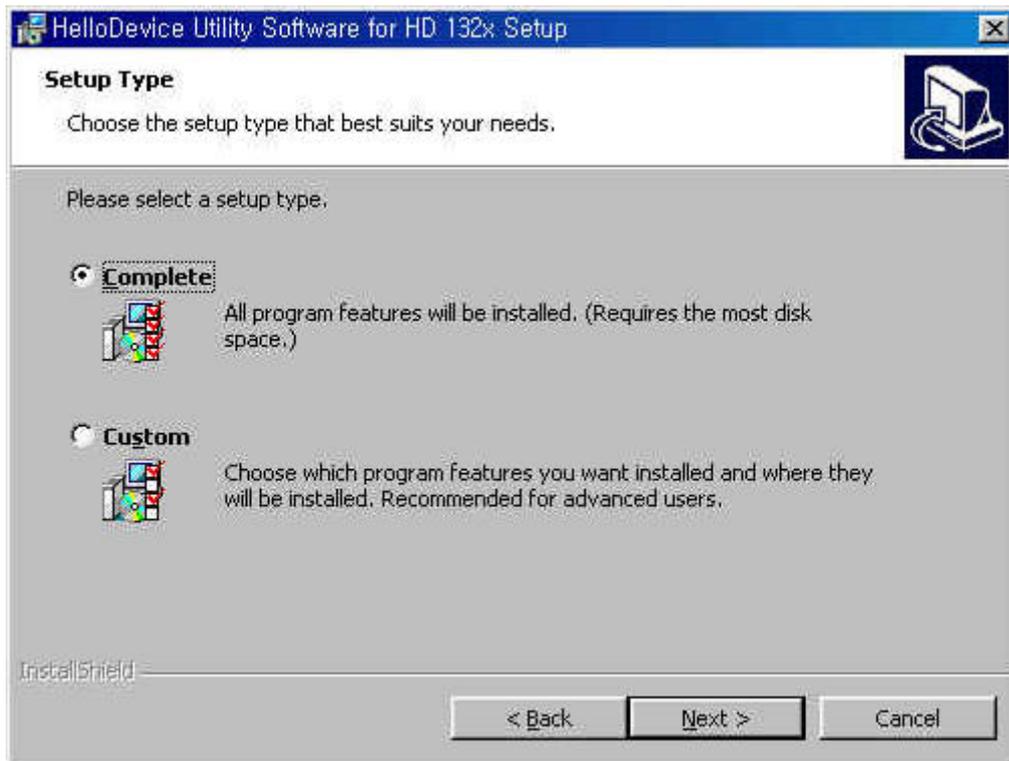
4.

HelloDevice

- (1) HelloDevice
- (2) HelloDevice
- (3) HelloDevice IP
- (4) HelloDevice

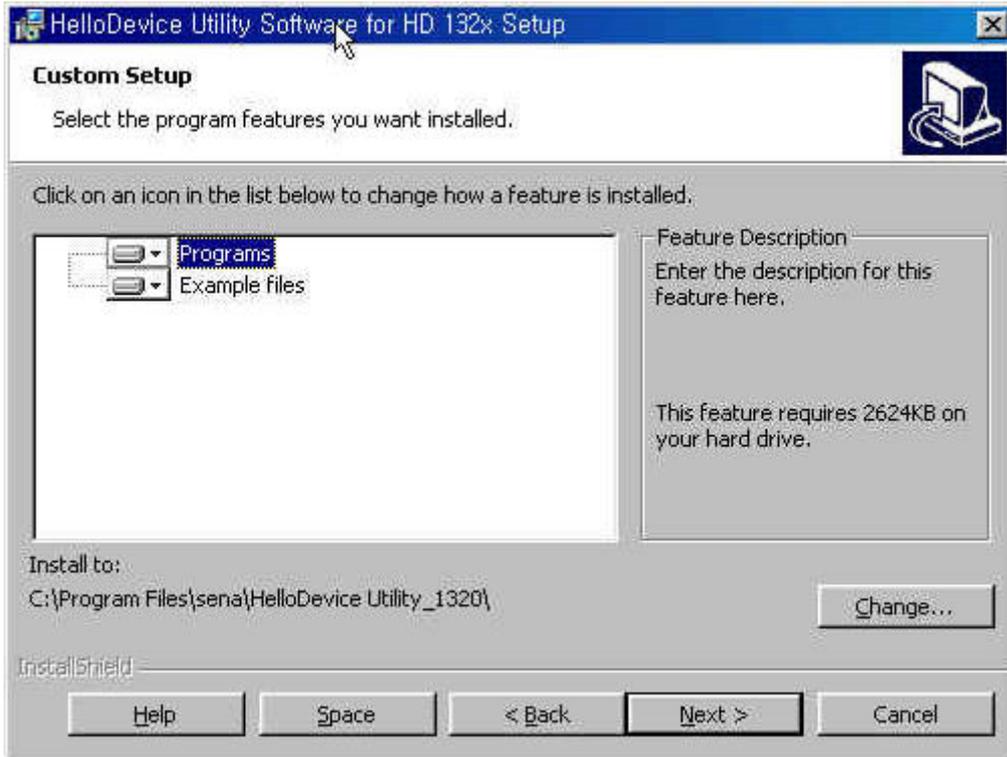
4.1

PC HelloDevice CD-ROM \HD\_UTILITY\HD1x20  
setup132x.exe . Setup132x.exe 95/98/ME/  
NT/2000 . [setup type] [Complete] [Custom]  
[Next ]

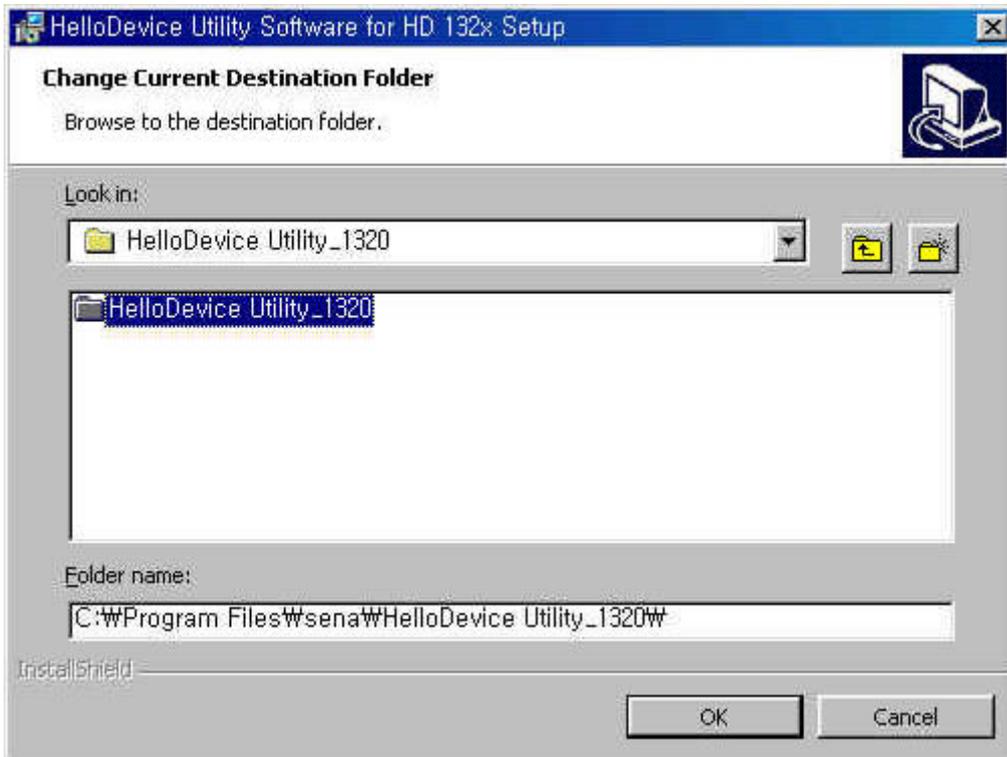


4.1 HelloDevice

[Complete] ,  
"Program Files\sena\HelloDevice Utility\_HD1320\" . [Custom]



4.2(a) [Custom setup]



4.2(b)

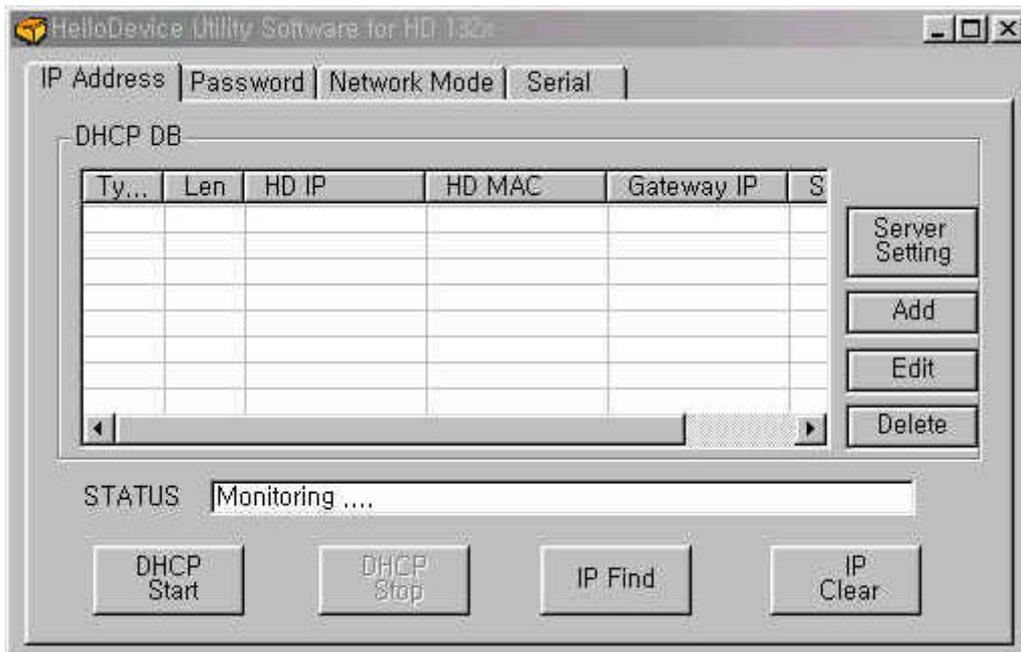
[Change]

[Change]

가

HelloDevice  
HelloDevice

가



### 4.3 HelloDevice

## 4.2

HD1320/1320E , RS232 , 1320E

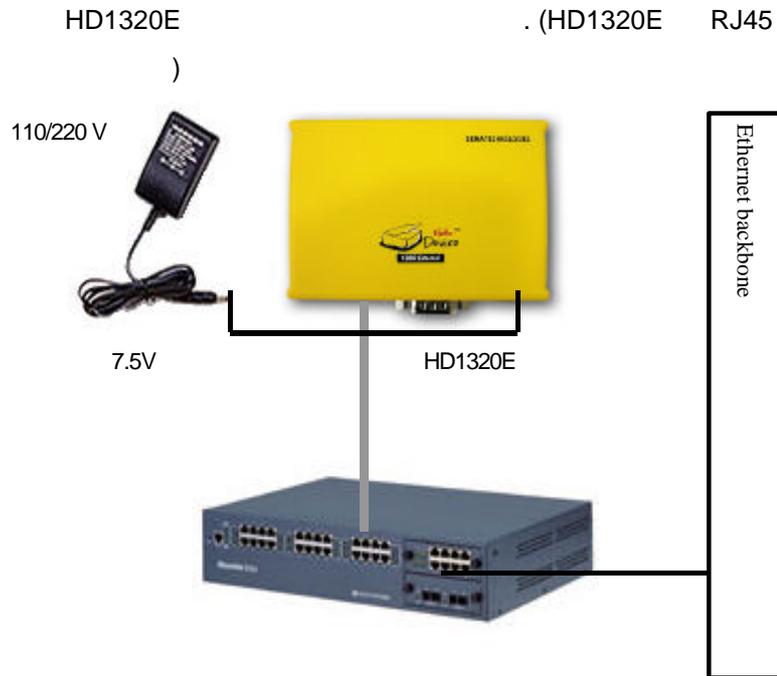
### 4.2.1 HD1320/1320E

(1) 7.5V ~ 9V DC



### 4.4 HD1320E 7.5V

(2) RJ45  
3



### 4.5. HD1320E

(3) HD1320E

LED Tx LED 가 1 /  
IP 가 . ( 3.1 [ LED] )



### 4.3

#### 4.3.1 IP

HD132x 10BaseT ,  
가 가 . 4.7 HelloDevice

#### - IP

IP .

1. DHCP	IP		
.	Server Setting Button	HelloDevice IP	
.IP 가	IP	IP	.

2.			
. Password	HelloDevice	Mac	.
. Password	Send	.	

3.			
. Network Mode	HelloDevice	Mac	.
. Network	Send	.	

4. RS232			
. Serial	HelloDevice	Mac	.
. Serial Baud Rate	Send	.	

#### - IP

HD132x DHCP , IP  
 DHCP(Dynamic Host Configuration Protocol) HelloDevice IP  
 DHCP(RFC2131) .

HelloDevice IP 0.0.0.0 ,  
 HelloDevice DHCP IP ,  
 HelloDevice Tx LED 가 . ( 3.1 [ LED] )

**(1) DHCP**

DHCP HelloDevice IP ,  
 DHCP IP ,  
 Tx LED , HelloDevice IP .

**(2) HelloDevice**

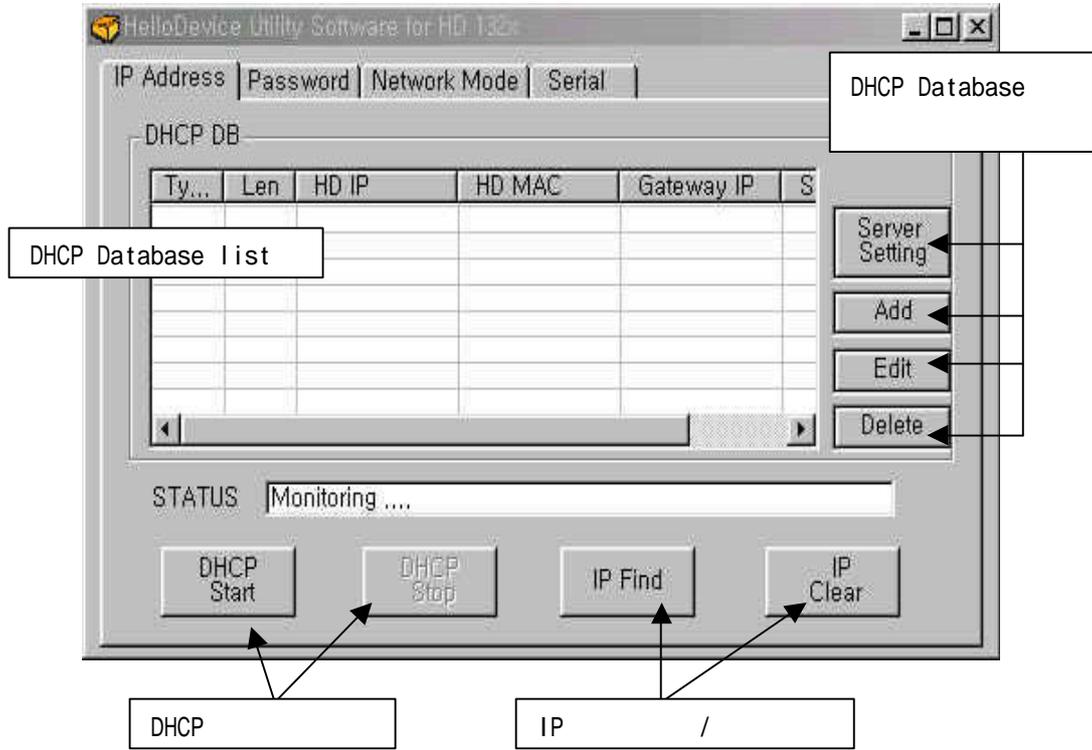
DHCP , DHCP HelloDevice DHCP  
 . HelloDevice IP IP  
 , IP HelloDevice MAC<sup>1</sup>-IP  
 IP . HelloDevice IP  
 가 IP ,  
 가 IP .

---

<sup>1</sup> MAC , 6 byte . HelloDevice MAC MAC  
 00-01-95 xx-xx-xx .  
 ) 00-01-95-01-aa-08, 00-01-95-01-02-01

HelloDevice

IP



4.7 HelloDevice

IP

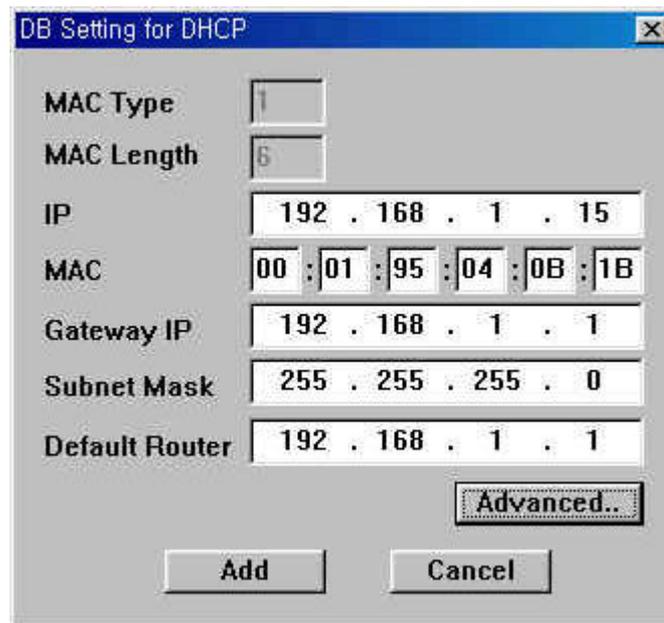
(1) PC HelloDevice 가 , [IP Address]  
 [IP Address]  
 DHCP IP (([DHCP Start], [DHCP Stop]), IP  
 ([IP Find]) ([IP Clear]) .

(2) [Server Setting] , DHCP  
 HelloDevice 가 DHCP [Router] IP  
 . IP  
 4.8 DHCP DHCP Server IP  
 가 PC IP .



#### 4.8 DHCP

(3) [Add] , DHCP .



#### 4.9 DHCP

HelloDevice MAC IP . MAC type MAC length  
 HelloDevice 가 , 1 6 . MAC Address  
 HelloDevice IC . 4.9 ,  
 MAC 가 00:01:95:04:0B:1B , HelloDevice IP 가 192.168.1.15  
 . [Default Router] , 4.8 DHCP

Router IP 가 .

**(4) [Add]**

, (2) 가 IP 가 .

**(5) [DHCP start]**

, DHCP .

```
[DHCP start] HelloDevice Utility Program DHCP Server
[Status] 가 " Monitoring " " Listening DHCP request "
DB List HelloDevice IP [Status]
가 " Listening DHCP request " " DHCP ACK sent... [ 192.168. 1.15 ]"
HelloDevice 가 DB List HelloDevice Utility Program "DB
Setting for DHCP" IP IP HelloDevice
IP 가 .
```

**(6) HelloDevice TX LED**

```
HelloDevice IP , HelloDevice IP
, [Status] "DHCP ACK sent... [ 192.168. 1.15 ]"
. HelloDevice TX LED 가 / , IP
, [Status] "DHCP ACK sent... [ 192.168.
1.15 ]" 가 , HelloDevice IP
DHCP 가 IP
, "It was Request to other DHCP server! Packet Discarded... " 가 .
```

**(7) ping , HelloDevice IP**

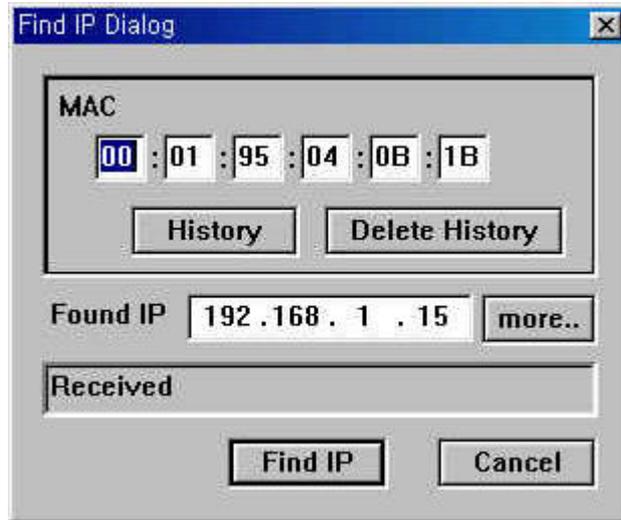
ping Command prompt . , ping

```
>> ping 192.168.1.15
>> Pinging 192.168.1.15 with 32 bytes of data:
Reply from 192.168.1.15: bytes=32 time=10ms TTL=251
Reply from 192.168.1.15: bytes=32 time<10ms TTL=251
Reply from 192.168.1.15: bytes=32 time=10ms TTL=251
```

, IP 가 , (5), (6), (7)

**(8) [IP Find] , HelloDevice IP**

[ IP Find] , 4.10 . HelloDevice MAC  
 , [Find] , "Found IP" IP 가 .

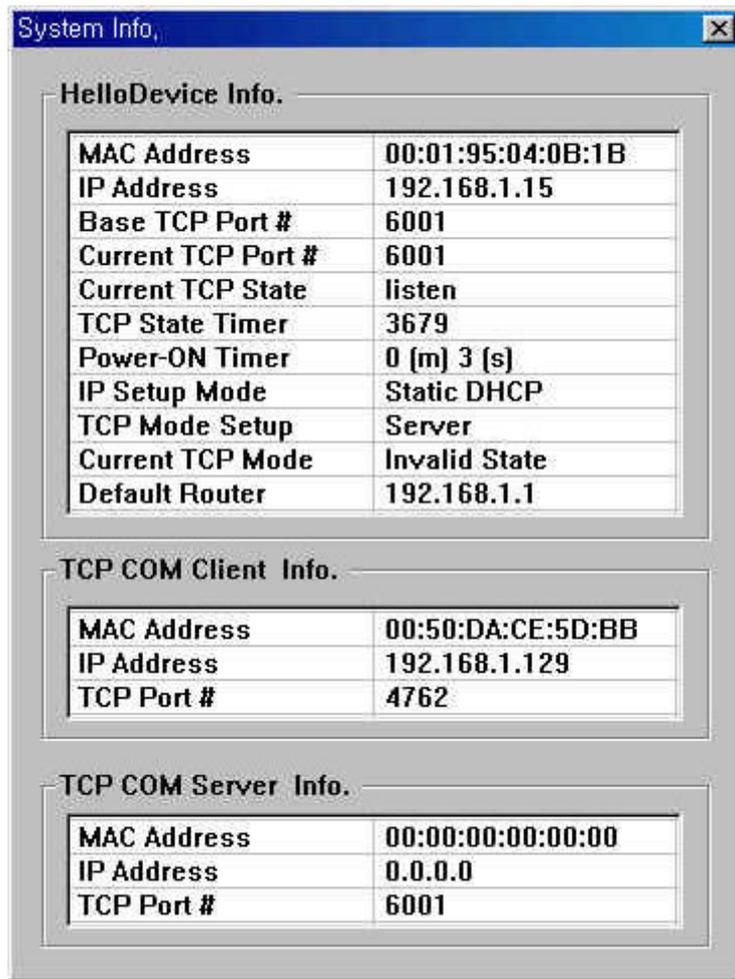


**4.10 [IP Find] IP**

, [more..] , [System info] , HelloDevice

**4.11 [System Info]** ,

4.1 .



4.11 [System info]

Group				
<b>HelloDevice Info.</b>	MAC Address	MAC		
	IP Address	IP		
	Base TCP Port #	TCP		
	Current TCP Port #	TCP		
	Current TCP State	TCP		
	TCP State timer	TCP State		
	Power-ON timer			
	IP setup mode	IP		
	TCP mode setup	TCP		
	Current TCP mode	TCP		
Default Router	IP			
<b>TCP COM Client info</b>	MAC Address	HD 가 Server	Client MAC	HelloDevice 가 Server Server/Client Mode
	IP Address	HD 가 Server	Client IP	
	TCP Port #	HD 가 Server	Client	
<b>TCP COM Server</b>	MAC Address	HD 가 Server	MAC	HelloDevice 가 Client Server/Client
	IP Address	HD 가 Server	IP	

info	TCP Port #	HD 가	Server	Mode
------	------------	------	--------	------

#### 4.1. [System Info]

**Note:**

- 1) [IP Find] , 가 .  
 [IP Find] HelloDevice 4.4.1 .
- 2) [IP Find] LAN(Local Area Network) . ,  
 가 .

#### 4.3.2 IP

IP , IP 0.0.0.0 4.3.1  
 IP , IP  
 , IP 192.168.1.15 192.168.1.18 가 , IP  
 가 .

#### (1) MAC

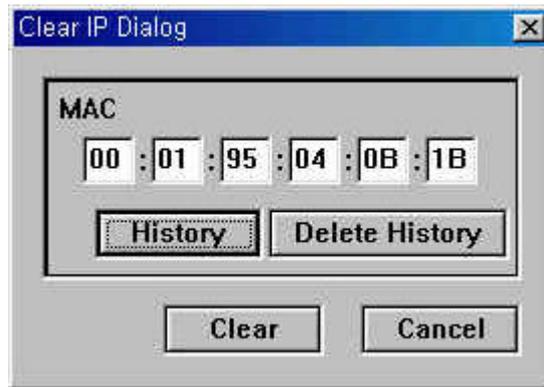
가 IP 192.168.1.15 0.0.0.0 , IP  
 가 HelloDevice MAC . HelloDevice MAC  
 HelloDevice , DHCP

#### (2) IP

[IP Address] [IP Clear] , IP .  
 4.12 HelloDevice MAC , [Clear] .  
 IP 가 192.168.1.15 HelloDevice IP 가 0.0.0.0 .

**Note:**

- IP Mode 가 Dynamic IP [IP Clear] IP 가 .
- Dynamic IP IP 가 IP Mode 4.4.2 .



#### 4.12 IP Clear

### (3) IP

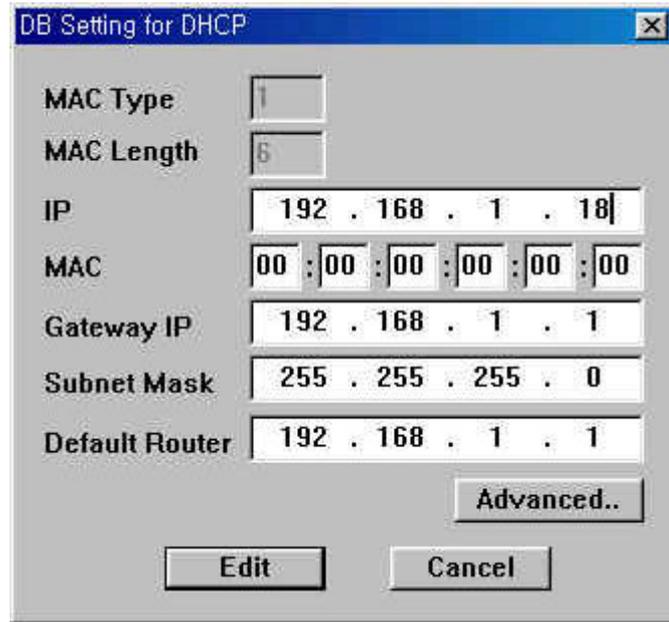
HelloDevice TX LED 가 . IP 가 , IP , HelloDevice TX LED 가 /

### Note:

DHCP 가 , HelloDevice IP [Dynamic-IP] , IP DHCP IP , TX LED 가 IP 4.4

### (4) IP

IP 192.168.1.18 , DHCP [Edit] IP / , 4.3.1 IP



4.13 DHCP

[Edit]

IP

4.4

HD132x 가

RS232

TCP/IP

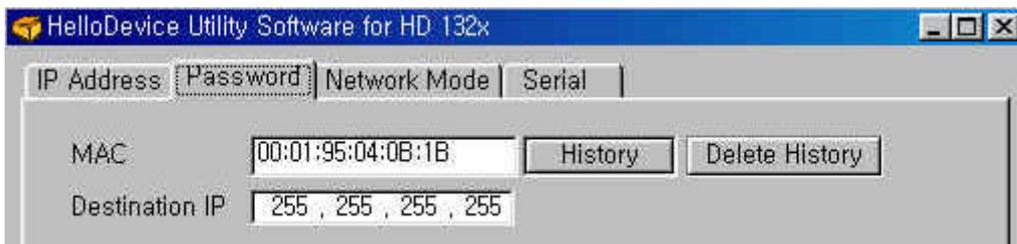
[Network Mode]

Serial Interface

[Serial]

Parameter

[Password]



4.14 Parameter

(1) Password

HelloDevice

MAC

IP

Note:

Parameter

MAC

. LAN

HelloDevie

HelloDevice 1320/1320E/1321

MAC (WAN ) Destination IP HelloDevice Parameter HelloDevice MAC  
255.255.255.255 HelloDevice Parameter HelloDevice MAC  
Destination IP HelloDevice IP

(2) , [Send]

[Send] , HelloDevice

#### 4.4.1.

[Password]

1)

HelloDevice 가 , HelloDevice

, ID : AAAAAAAAA Password : AAAAAAAAA

4.15

, [Send]



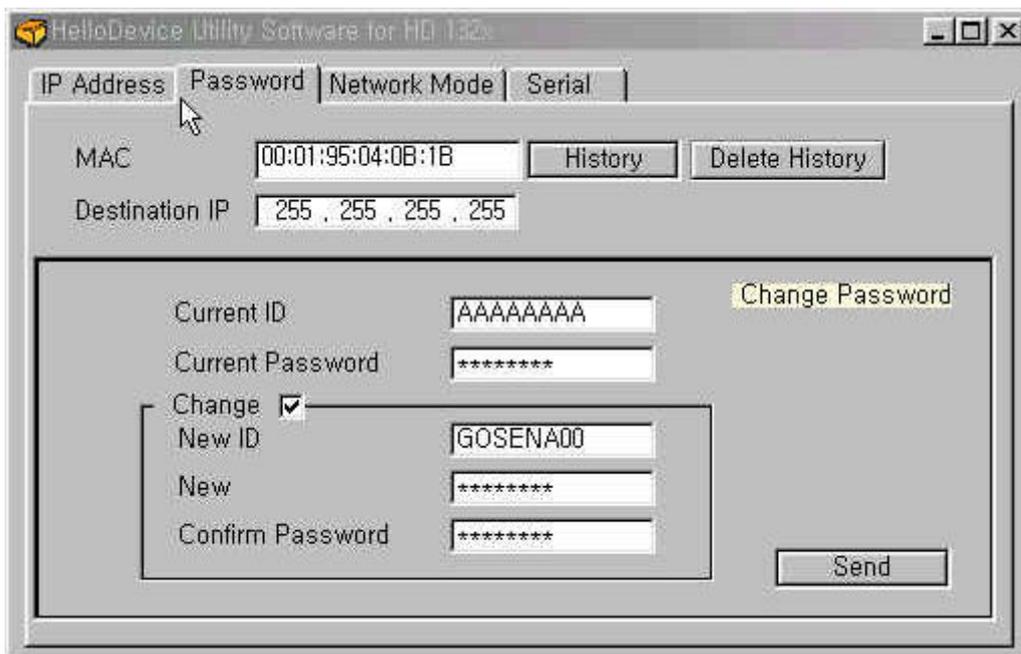
4.15

4.16 , 가 .



4.16

가 , [Change] Combo box , , [Send] . 4.17 4.18 가 ID GOSENA00 HelloDevice .



4.17



4.18

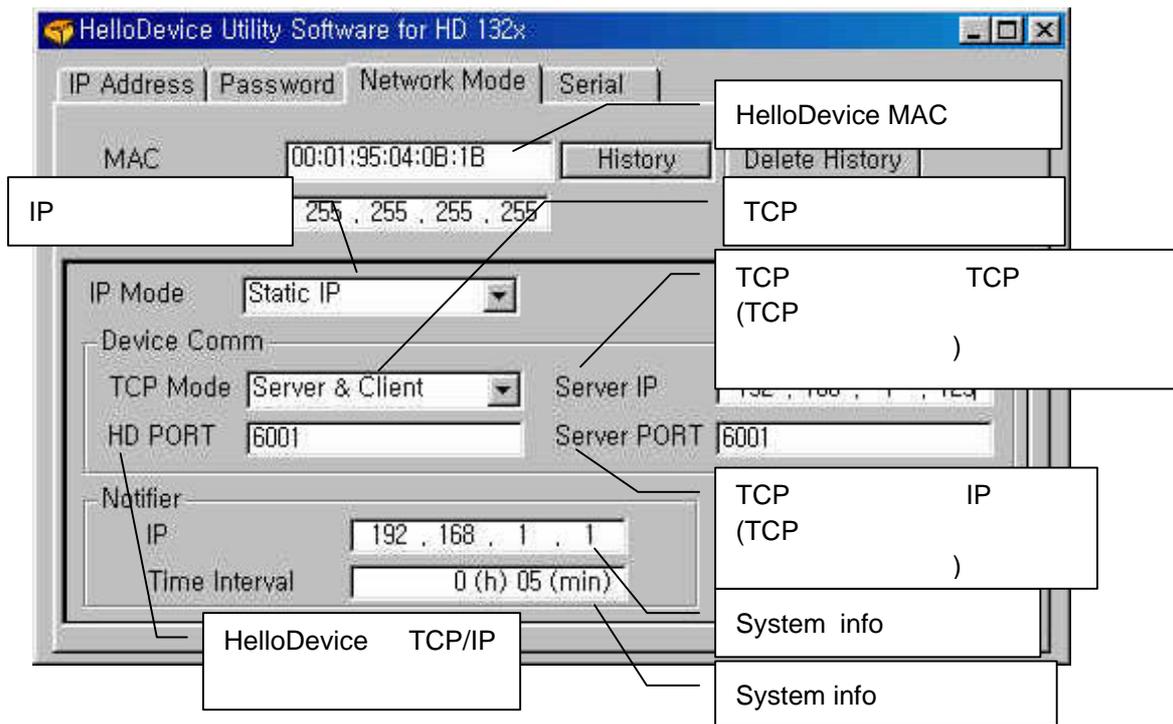
HelloDevice ID / Password                    8                    ,                    /                    /  
 . ID / Password                    가                    PWD.ini                    MAC  
                   ,                    가                    ID / Password                    .                    [IP Clear], [IP Find],  
 [Network mode], [Serial], [ID/Password                    /                    ]

**Note:**

PWD

**4.4.2 Network Mode**

4.19                    Network Mode



4.19

- HelloDevice
- HelloDevice                    TCP
-

**(1) IP Mode**

**Static IP**

IP , IP , IP .  
 , [IP Clear] IP 4.3.2  
IP .

**Dynamic IP**

Dynamic IP , 가 , DHCP  
IP . DHCP IP DHCP  
 , HelloDevice 24 DHCP IP  
 . , Dynamic IP , 24 IP .

**Note:**

Dynamic IP [IP Clear] IP 가 .

**(2) Device Comm**

**Server Mode**

TCP HelloDevice Port .  
TCP 가 , HelloDevice , TCP/IP  
HelloDevice RS232 . HelloDevice TCP  
 , TCP 가 TCP  
 . TCP , TCP 가 .

**Client Mode**

HelloDevice RS232 가  
TCP HelloDevice 가 . 1  
5 .

**Server & Client Mode**

HelloDevice TCP HelloDevice  
TCP TCP IP  
 , . TCP ,  
TCP 가 , .

**Note:**

RS232 TCP .  
**HD PORT**  
 TCP , HelloDevice 가 TCP TCP  
 . , 가 2000 ~  
 65535 .

**Server IP, Server PORT**

Server IP Server PORT HelloDevice 가 TCP , HelloDevice  
 가 , TCP IP  
 . 가 2000 ~ 65535 . HelloDevice 가 TCP  
 .

**(3) Notifier**

HelloDevice [Time Interval] IP  
 IP HelloDevice  
 , HelloDevice 가 Dynamic IP , TCP  
 (DHCP, TCP ) . , Dynamic IP , IP  
 가 24 , 가 HelloDevice IP  
 가 . , HelloDevice IP  
 가 . PC  
 HelloDevice UDP  
 . 5.3 .  
 가 6 . , 가  
 6 ,  
 . [Time Interval] 1 30 .

**4.4.3 RS232**

RS232 HelloDevice RS232  
 . 4.20 RS232 [Serial] .

**| baud rate**

150, 300, 600, 900, 1200, 2000, 2400, 3600, 4800, 7200, 9600, 19200, 38400

**| parity**

None, Even, Odd

**| Data bits**

5, 6, 7, 8 bits

| **Stop bits**

1, 1½, 2 bit

| **handshake**

None, H/W, X On/Off

ê **Time**

RS232

TCP/IP

4, 100, Unlimited

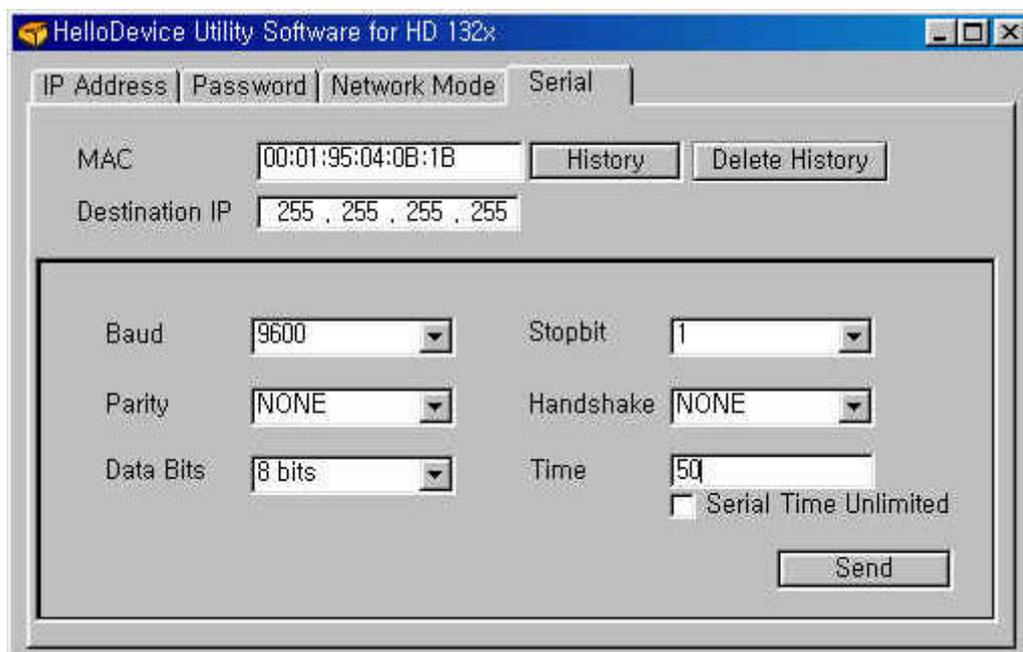
4.20

[Serial]

, MAC

00:01:95:04:0B:1B, 9600 baud rate, Parity None, Data bit 8, Stop bit 1, Handshake None, Time

50



**4.20 RS232**

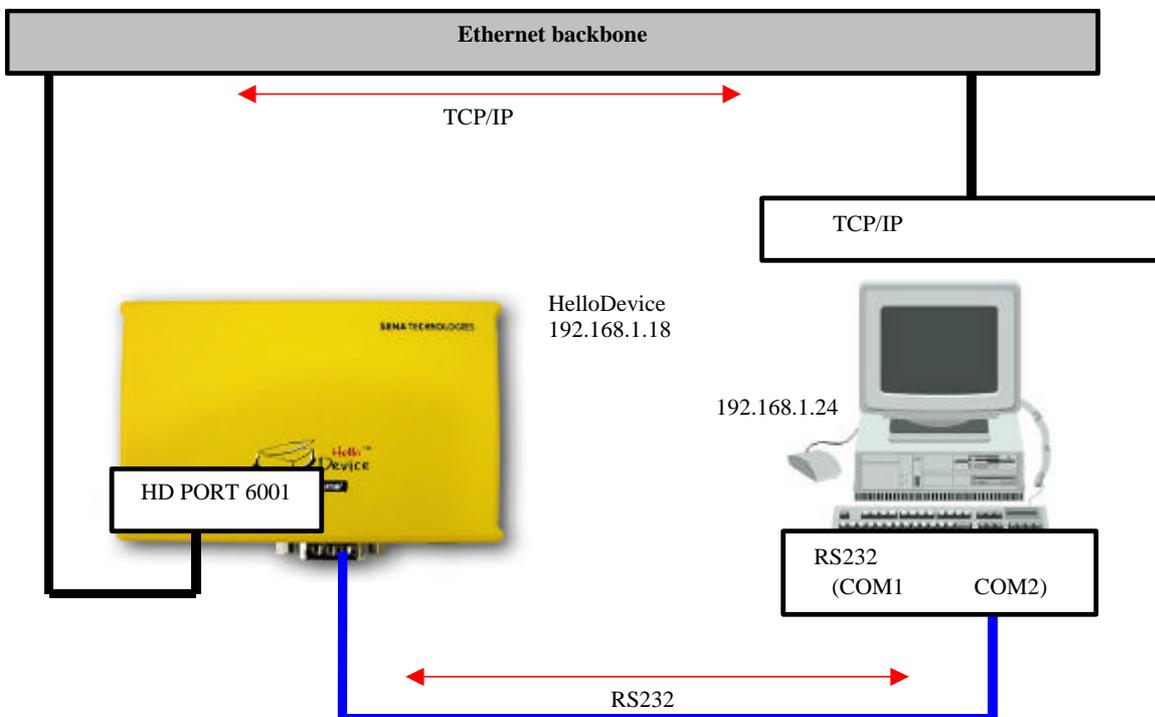
5.

, TeraTerm Pro, HyperTerminal, Telnet  
 RS232-TCP/IP , HelloDevice PC  
 , HelloDevice 가 Dynamic IP, TCP  
 , IP  
 , TeraTerm Pro

5.1

- HD1320E
- RS232
- 가 PC
- TCP/IP , RS232

1) 가 PC RS232 HelloDevice RS232  
 RS232 3



5.1 RS232-TCP/IP

**2) HelloDevice**

HelloDevice TCP , HelloDevice IP  
192.168.1.18 가 . [Network Mode] TCP “ TCP  
” , 6001 (4.4.2 Device Comm )  
, HelloDevice TCP  
. RS232 4.4.3 .

**9600 baud rate**

**Parity None**

**Data bit 8**

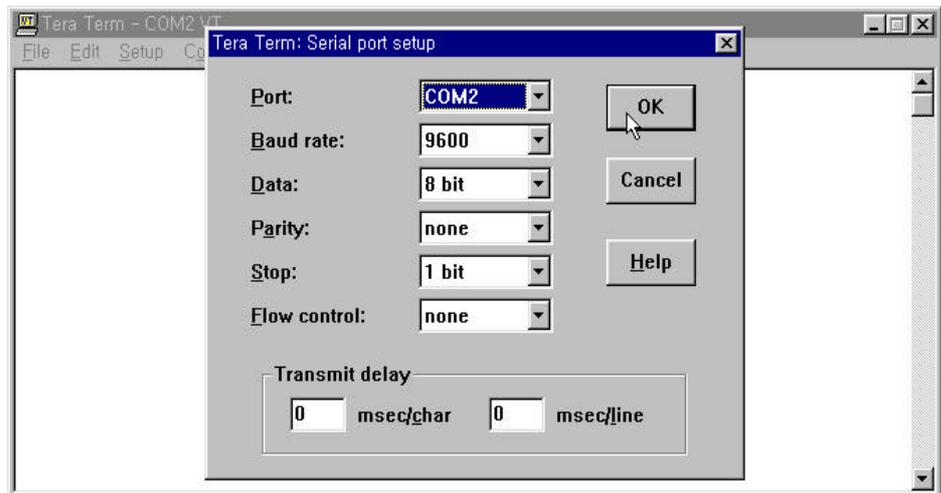
**Stop bit 1**

**Handshake None,**

**Time 50**

**3) PC RS232** .  
, TeraTerm Pro COM2 .

**4) Tera term Pro COM2** .  
, HelloDevice RS232 .

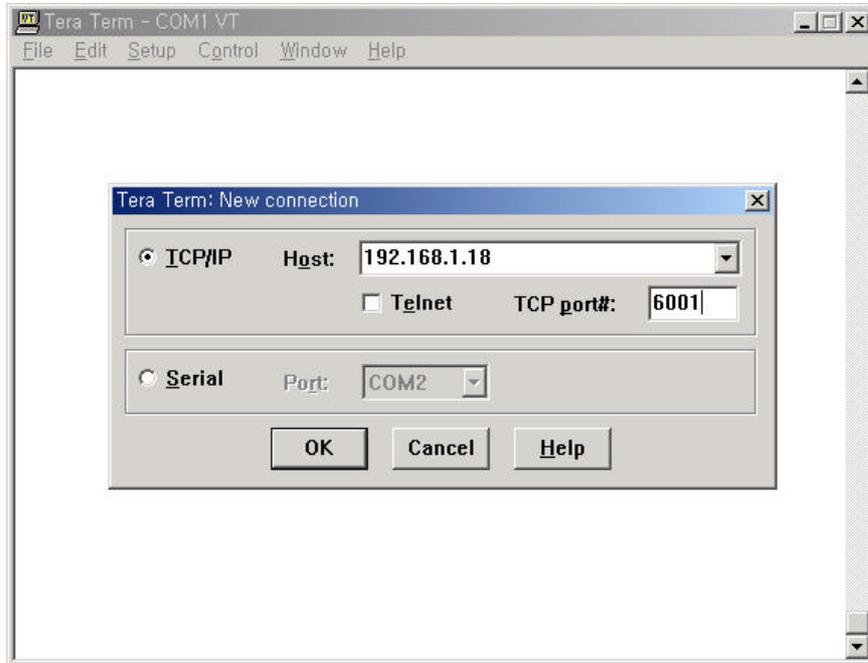


**5.2 TeraTerm RS232**

**5) [OK]** , .

6) PC TCP/IP .

7) HelloDevice IP TCP TCP/IP .



5.3 TCP/IP

TCP/IP

8) TCP/IP

RS232

. RS232

TCP/IP

5.4 TCP/IP

RS232

**Note:**

HelloDevice

RS232

[Time]

RS232

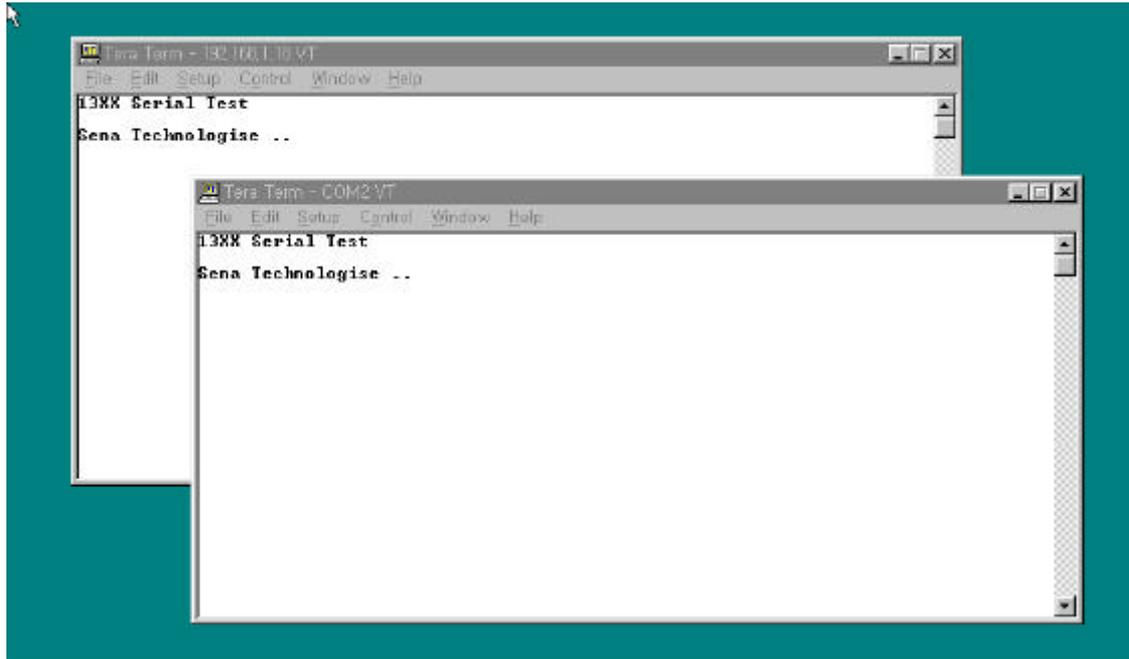
Hello Device

TCP/IP

2)

50 가

TCP/IP



#### 5.4 HelloDevice

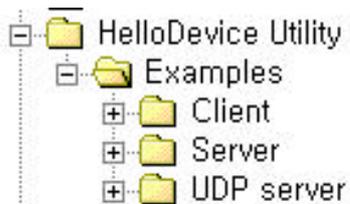
/

## 5.2

HelloDevice TCP/IP  
가

, 5.1

- HelloDevice 가 TCP ( 5.1 "Server" )
- HelloDevice 가 TCP ( 5.1 "Client" )
- HelloDevice ( 5.1 "UDP Server" )



#### 5.5 HelloDevice

, Visual C/C++ 6.0  
(\* .dsw)

, Visual C/C++ 6.0

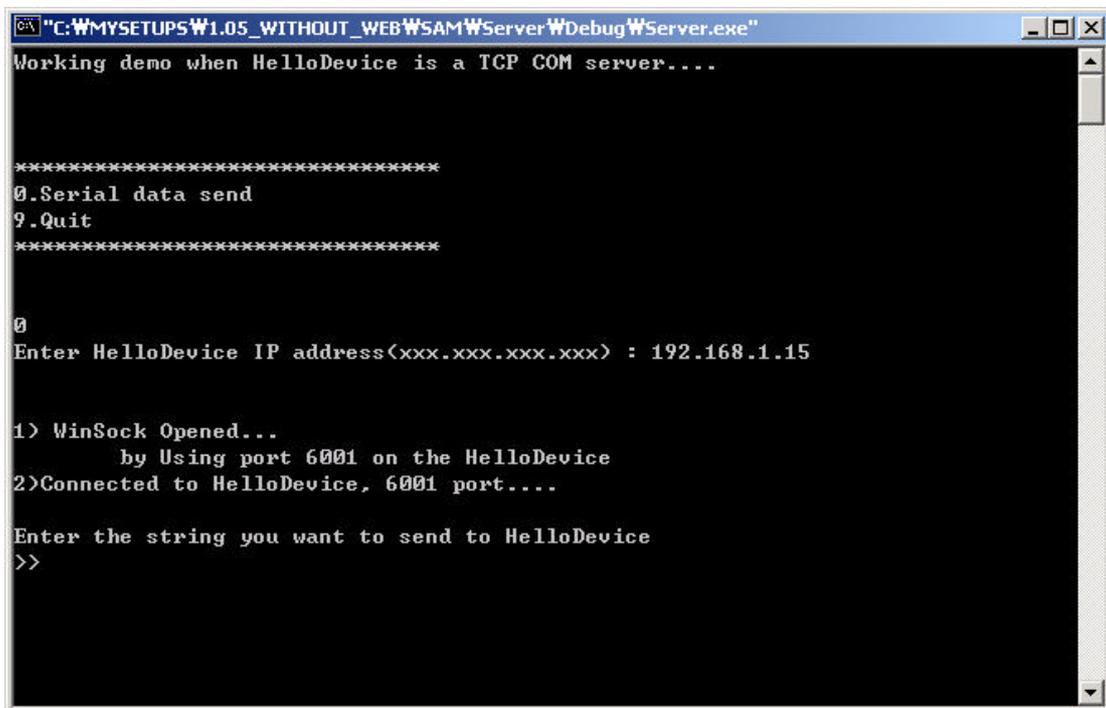
### 5.2.1 TCP

, HelloDevice 가 5.1 TCP  
가 HelloDevice , TCP/IP  
가  
, 5.1 , HelloDevice 가 TCP

1) 5.1  
, HelloDevice RS232 PC RS232 , RS232

2) "Server" "Server.dsw" Open "Build" , "Run"

3) HelloDevice IP  
5.6 HelloDevice IP 가 192.168.1.15



4) 가 , [Enter]

PC RS232

가

가

TCP

TCP , TCP/IP

socket, connect, send

```
//-----
// Process Serial data send
//-----
void SerialSend()
{
    char    commandBuf[512]="" ;
    int     commandLen ;
    int     err ;

    // Re-Initialize TCP socket
    TCPSocketInit() ;

    // Read serial data
    //: just ASCII string excluding control characters...
    //: Max size is limited to 512 bytes in this demo...
    printf("\nEnter the string you want to send to HelloDevice\n>>") ;
    scanf("%s", commandBuf) ;

    // Calc serial data length
    commandLen = strlen(commandBuf) ;

    // Send command to HelloDevice
    err = sendto
        (
            sock,
            &commandBuf,
            commandLen,
            0,
            (struct sockaddr*)&serverAddr,
            sizeof(serverAddr)
        ) ;
    if (err == -1 )
    {
        perror("\nsend error\n");
        exit (1);
    }
}

//-----
// Initialize TCP socket
//-----
void TCPSocketInit()
{
    char    ipAddrStr[32] ;
```

## HelloDevice 1320/1320E/1321

```

int    clientLen ;
int    err ;

// Enter HelloDevice IP address
printf("Enter HelloDevice IP address(xxx.xxx.xxx.xxx) : ") ;
scanf("%s", ipAddrStr) ;

printf("\n\n") ;

// Convert IP address from string to long
ipAddr = decodeAddress(ipAddrStr) ;

// Windows requires that winsock be initialized.
err = WSStartup (0x0101, &lpWSAData);
if ( err != 0 )
{
    printf("\nCannot open WinSock???\n");
    exit (1) ;
}
else
    printf("1) WinSock Opened...\n") ;

// Create TCP socket
clientLen = sizeof(serverAddr);
sock = socket(AF_INET, SOCK_STREAM, 0);
if ( sock < 0 )
{
    perror("\nsocket error???\n");
    exit (1) ;
}

// Clear IP address fields
memset( (char*) &addr, 0, sizeof( addr ) );
memset( (char*) &serverAddr, 0, sizeof( serverAddr ) );

// Set my IP address : TCP port 6001
addr.sin_family = AF_INET;
// You may use any port other than 6001 in host side!
addr.sin_port = htons(6001);
addr.sin_addr.s_addr = INADDR_ANY;

// Set HelloDevice IP address : TCP port 6001
serverAddr.sin_family = AF_INET;
serverAddr.sin_port = htons(6001);
printf("\tbody Using port %d on the HelloDevice\n",
        ntohs(serverAddr.sin_port) );
serverAddr.sin_addr.s_addr = htonl(ipAddr);

// Connecting to HelloDevice
err = connect( sock, (struct sockaddr*) &serverAddr,
               sizeof(serverAddr) );
if ( err == -1 )
{
    perror("\nCannot connect to HelloDevice???\n");
    exit (1) ;
}
printf("2)Connected to HelloDevice, %d port....\n",
        ntohs(serverAddr.sin_port) );
}

```

```
//-----
// Close TCP socket
//-----
void TCPSocketClose()
{
    closesocket(sock) ;
}
```

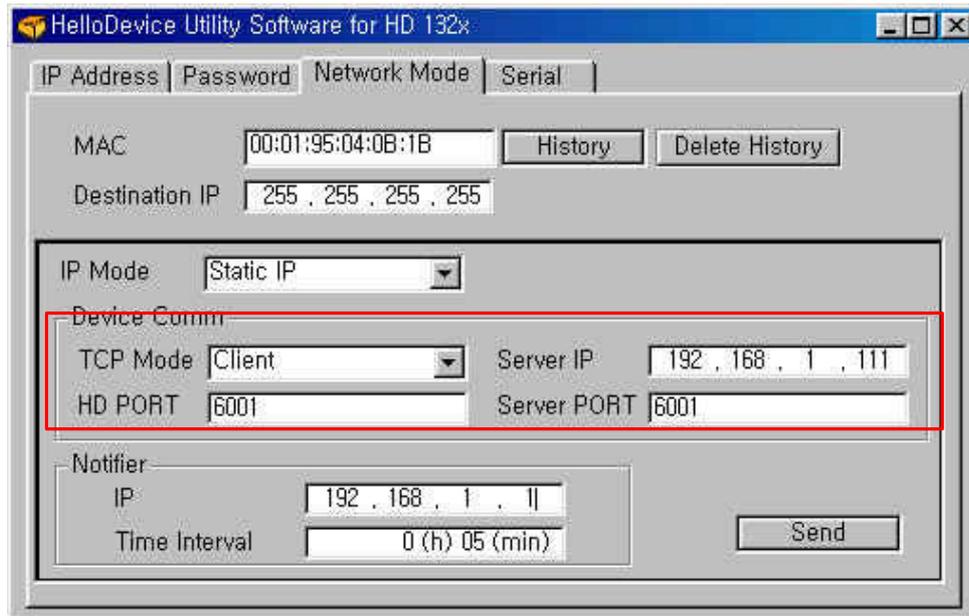
## 5.2.2 TCP

5.1 , HelloDevice TCP  
 TCP  
 , TCP  
 IP , HelloDevice 가 TCP  
 , RS232 HelloDevice , HelloDevice 가  
 IP , ,  
 HelloDevice , HelloDevice  
 RS232 , RS232  
 , RS232 PC , RS232  
 HelloDevice ,

1) 5.1 .  
 , HelloDevice RS232 PC RS232 , RS232

### 2) HelloDevice [Network Mode]

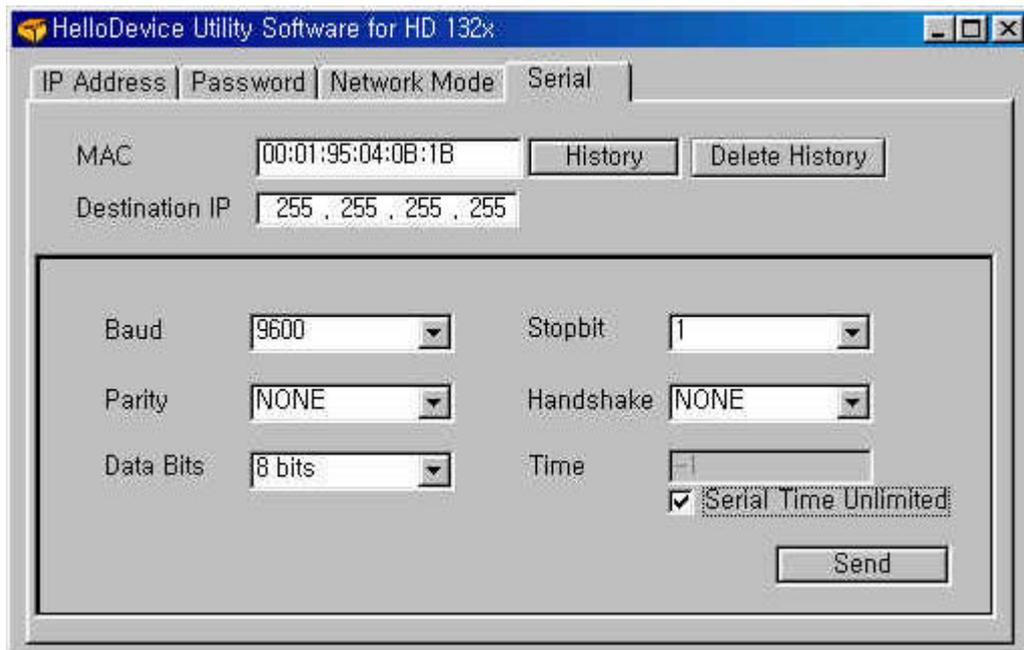
[TCP mode] [Client] , [Server IP, PORT]  
 PC IP , 6001 , 5.2.1  
 . 5.7 PC IP 가 192.168.1.111  
 , [Send] . HelloDevice  
 TCP , HelloDevice RS232 가  
 IP 192.168.1.111 TCP 6001 ,  
 가 HelloDevice RS232



**5.7 HelloDevice TCP**

**3) HelloDevice [Serial] [Serial Time Unlimited]**

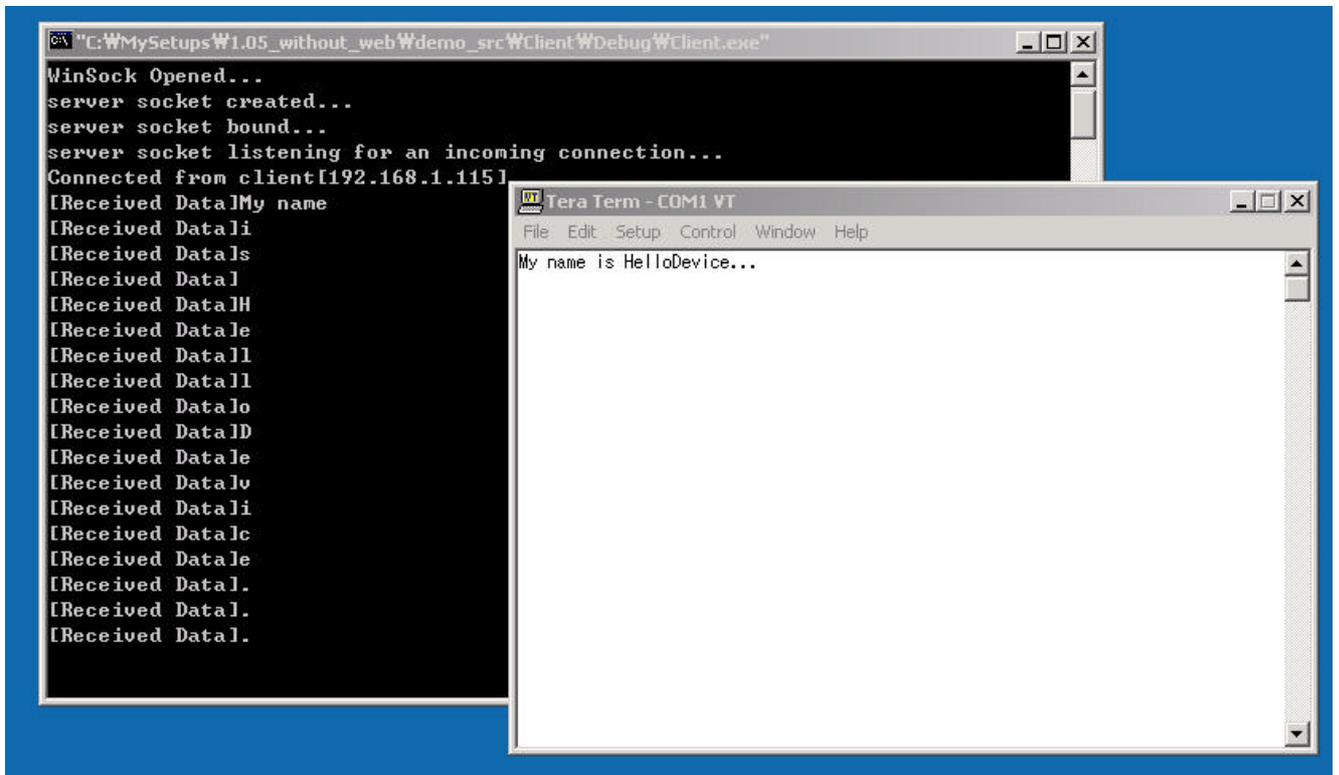
, HelloDevice TCP ,  
 RS232 TCP . 5.8  
 , "Serial Time Unlimited" , [Send] . HelloDevice  
 TCP , TCP



### 5.8 HelloDevice [Serial]

4) “Client” “Client.dsw” Open “Build” , “Run”

5) , RS232  
가 [Received Data] ,  
HelloDevice , RS232 가



### 5.9

listen, bind, accept, socket, recv, sendto

5.2.1

//-----

## HelloDevice 1320/1320E/1321

```

// Process listening : ProcessRecv called
//-----
void ProcessListen()
{
    int clientLen;
    char chDumm;
    clientLen = sizeof(addrClient);
    memset((char*) &addrClient, 0, sizeof(addrClient));

    while (1) {
        //-----
        // accept an incoming connection attempt on the server socket
        //-----
        sockClient = accept(sock, (LPSOCKADDR)&addrClient, &clientLen);
        if (sockClient == INVALID_SOCKET) {
            printf("\naccept error???\n");
            printf("\nPlease, try later(press ENTER)");
            scanf("%c", &chDumm);
            break;
        } else {
            printf("Connected from client[%d.%d.%d.%d]\n"
                ,addrClient.sin_addr.S_un.S_un_b.s_b1
                ,addrClient.sin_addr.S_un.S_un_b.s_b2
                ,addrClient.sin_addr.S_un.S_un_b.s_b3
                ,addrClient.sin_addr.S_un.S_un_b.s_b4);
            //-----
            // receives data from the client socket(HD1320)
            //-----
            if(ProcessRecv()) { // Exit Program
                break;
            } else { // Continue Listen
                // Just close the client socket
                TCPSocketClose(CLOSE_CLIENT_SOCKET);
                memset((char*) &addrClient, 0, sizeof(addrClient));
            }
        }
    }
}

//-----
// Process receive : socket recv function called
// return 9 : exit program
// 0 : continue listening
//-----
int ProcessRecv()
{
    char bufReceived[BUFSIZE];
    int nCountReceived = 0;
    int nCommand = 0;
    int nShowMenu = 0;
    memset(bufReceived, '\0', sizeof(bufReceived));
    while(1) {
        //-----
        // receives data from a connected socket(HD1320)
        //-----
        nCountReceived = recv(sockClient, bufReceived, sizeof(bufReceived), 0);
        if (nCountReceived == 0) { // HelloDevice closed the client socket
            printf("Disconnected from client[%d.%d.%d.%d]\n"
                ,addrClient.sin_addr.S_un.S_un_b.s_b1

```

## HelloDevice 1320/1320E/1321

```

        ,addrClient.sin_addr.S_un.S_un_b.s_b2
        ,addrClient.sin_addr.S_un.S_un_b.s_b3
        ,addrClient.sin_addr.S_un.S_un_b.s_b4);
    DisplayMenu();
    scanf("%d", &nCommand);
    return nCommand;
} else if (nCountReceived < 0) {
    if (++nShowMenu == SHOWMENU) {
        DisplayMenu();
        scanf("%d", &nCommand);
        return nCommand;
    } else {
        continue;
    }
} else { // received some data
    // Send echo to client
    if(!SerialSend(bufReceived, nCountReceived)){ // Sending echo error
        printf("Fail to send echo to client[%d.%d.%d.%d]\n"
            ,addrClient.sin_addr.S_un.S_un_b.s_b1
            ,addrClient.sin_addr.S_un.S_un_b.s_b2
            ,addrClient.sin_addr.S_un.S_un_b.s_b3
            ,addrClient.sin_addr.S_un.S_un_b.s_b4);
        DisplayMenu();
        scanf("%d", &nCommand);
        return nCommand;
    }
    printf("[Received Data]%s\n", bufReceived);
    nCountReceived = 0;
    nShowMenu = 0;
    memset(bufReceived, '\0', sizeof(bufReceived));
}
}
}

//-----
// Process sending serial data to HelloDevice
// return : 1 - success , 0 - failure
//-----
int SerialSend(char* strReceived, int nReceived)
{
    int err;
    char* pCommandBuf = (char*)malloc(nReceived + 2);
    memcpy(&pCommandBuf[0], strReceived, nReceived);
    // Send command to HelloDevice
    err = sendto
        (
            sockClient,
            pCommandBuf,
            nReceived + 2,
            0,
            (struct sockaddr*)&addrClient,
            sizeof(addrClient)
        ) ;
    free(pCommandBuf);
    if (err == -1 )
    {
        return 0;
    }
    return 1;
}

```

```

}

//-----
// Initialize TCP server socket
// return : 1 - success , 0 - failure
//-----
int TCPServerSocketInit()
{
    char    chDummy;
    int     err ;

    //-----
    // Initiate use of WS2_32.DLL by a process
    //-----
    err = WSAStartup (0x0101, &lpWSAData);
    if ( err != 0 )
    {
        printf("\nfail to start up winsock???\n");
        scanf("\nPlease, try later(press ENTER)");
        scanf("%c", &chDummy);
        return 0;
    }
    else
        printf("WinSock Opened...\n") ;

    //-----
    // create a server socket
    //-----
    sock = socket(AF_INET, SOCK_STREAM, 0);
    if ( sock < 0 )
    {
        printf("\nsocket error???\n");
        printf("\nPlease, try later(press ENTER)");
        scanf("%c", &chDummy);
        return 0;
    }
    printf("server socket created...\n") ;

    // Clear server IP address fields
    memset( (char*) &addr, 0, sizeof( addr ) );
    // Set server IP address : TCP port 6001
    addr.sin_family = AF_INET;
    // You may use any port other than 6001 in host side!
    addr.sin_port = htons(6001);
    addr.sin_addr.s_addr = INADDR_ANY;
    //-----
    // associate a local address with a socket
    //-----
    if(bind(sock, (LPSOCKADDR)&addr, sizeof(addr))
        == SOCKET_ERROR) {
        printf("\nserver socket bind error???\n");
        scanf("\nPlease, try later(press ENTER)");
        scanf("%c", &chDummy);
        return 0;
    }
    printf("server socket bound...\n") ;
    //-----
    // places a socket a state where it is listening for an
    // incoming connection.

```

```

//-----
if(listen(sock,5) == SOCKET_ERROR) {
    printf("\nserver socket listen error???\n");
    scanf("\nPlease, try later(press ENTER)");
    scanf("%c", &chDummy);
    return 0;
}
printf("server socket listening for an incoming connection...\n");
return 1;
}

//-----
// Close TCP socket
// parameter
//bCloseOnlyClient : CLOSE_CLIENT_SOCKET(1) - close just client socket
//CLOSE_ALL_SOCKET(0) - close all socket and clean up
//-----
void TCPSocketClose(int bCloseOnlyClient)
{
    //-----
    // close the client socket
    //-----
    if (sockClient != INVALID_SOCKET) {
        closesocket(sockClient);
        sockClient = INVALID_SOCKET;
    }

    if (!bCloseOnlyClient) {
        //-----
        // close the server socket
        //-----
        closesocket(sock);
        //-----
        // terminate use of the WS2_32.DLL
        //-----
        WSACleanup();
    }
}
}

```

### 5.2.3

HelloDevice Dynamic IP , TCP ,  
, 가 HelloDevice IP  
. HelloDevice 가 Dynamic IP , IP 가  
, HelloDevice IP , HelloDevice  
IP , HelloDevice IP  
가 IP  
, UDP (User Datagram Protocol) . ,  
HelloDevice IP , HelloDevice  
Dynamic IP HelloDevice .  
, HelloDevice 가 , HelloDevice 가

## HelloDevice 1320/1320E/1321

가 IP , LAN ,  
 가 IP LAN  
 HelloDevice , IP , UDP 514 MAC  
 , Local Port, IP IP  
 4.4.2 . HelloDevice 가 14

"OK" (2 Byte)+ MAC (6 Byte) + Local Port(2 Byte) + IP (4 Byte)

### HelloDevice

```

/*****
UDP-based data server sample program

HelloDevice notifying message format

: total 14 bytes data
: The number in the parenthesis means the byte size...

'O'(1)+'K'(1)+ MAC address(6) + local port number(2) + IP address(4)

e.g.
4f-4b-00-01-95-04-04-01-17-71-c0-a1-a8-0f
*****/
#include <stdio.h>
#include <time.h>
#include <winsock.h>

// Global Variable definition
WSADATA lpWSAData; // Socket data structure
SOCKADDR_IN addrFrom;
int sock ; // Socket
int byte_received ; // byte received
struct sockaddr_in addr ; // My IP address

// UDP Socket function
void UDPSocketCreate() ;
void UDPSocketRun() ;
void UDPSocketClose() ;

// Main function
void main()
{
    printf("UDP Hello Device Program \n");

    // 1) Socket Creation
    UDPSocketCreate() ;

    while(1)
    {
    
```

## HelloDevice 1320/1320E/1321

```
        // 2) Listen until any incoming data
        // 3) Receive if any incoming data
        // 4) Print data
        UDPSocketRun();
    }
    // 5) Close UDP Socket
    UDPSocketClose() ;
}

//-----
// UDP Socket Creation
//-----
void UDPSocketCreate()
{
    // Windows requires that winsock be initialized.
    if (WSAStartup (0x0101, &lpWSAData) == INVALID_SOCKET)
    {
        printf("\nCannot open WinSock???\n");
        exit (1) ;
    }
    else
        printf("WinSock Opened...waiting..\n") ;

    // Create Windows socket for UDP
    sock = socket(AF_INET, SOCK_DGRAM,0);
    if (sock < 0)
    {
        perror("\nsocket error???\n");
        exit (1) ;
    }

    // Setupo the port configuration
    // UDP port : 514
    addr.sin_family = AF_INET;
    addr.sin_port = htons(514);
    addr.sin_addr.s_addr = htonl(INADDR_ANY);

    // Launch UDP socket
    if (bind(sock,(LPSOCKADDR)&addr,sizeof(addr)) == SOCKET_ERROR)
    {
        printf("\n Socket error program terminated..\n " );
        exit(1);
    }
}
//-----
// UDP Socket Run
//-----
void UDPSocketRun()
{
    int nAddrFromLen = sizeof(addrFrom);
    IN_ADDR inFrom;
    char test_buff[300];
    char dbuffer [9];
    char tbuffer [9];
    int i ;

    // Wait until it receives data
    byte_received =
```

## HelloDevice 1320/1320E/1321

```
    recvfrom(sock,test_buff,250,0,(LPSOCKADDR)&addrFrom,&nAddrFromLen);
if (byte_received==SOCKET_ERROR)
{
    printf("\n Socket error program terminated..\n");
    exit(1);
}
memcpy(&inFrom, &addrFrom.sin_addr,4);

// Calculate date & time
_strdate( dbuffer );
printf( "\n Info. was notified on %s ", dbuffer );
_strtime( tbuffer );
printf( "%s \n", tbuffer );

// Decode the incoming datagram
printf("\n from %s \n\n MAC address: " , inet_ntoa(inFrom) );

for(i=2; i<8; i++)
    printf("%2x " ,(unsigned char)test_buff[i]);
printf("\n Port # :");

for(i=8;i<10;i++)
    printf("%3x", (unsigned char)test_buff[i]);

printf(" \n IP address :");

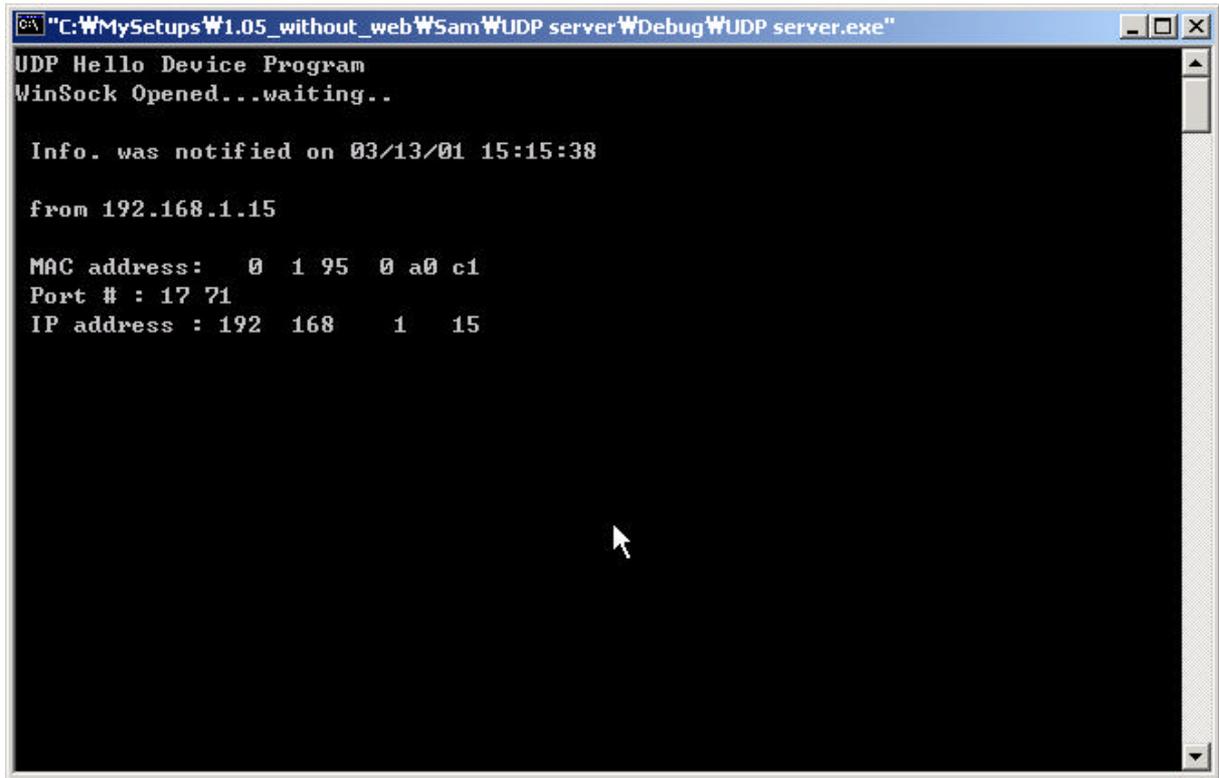
for(i=10;i<14;i++)
    printf(" %3d ", (unsigned char)test_buff[i]);

printf("\n \n \n \n");
}
//-----
// UDP Socket Close
//-----
void UDPSocketClose()
{
    closesocket(sock);
}
```

Visual C/C++ console application

5.10

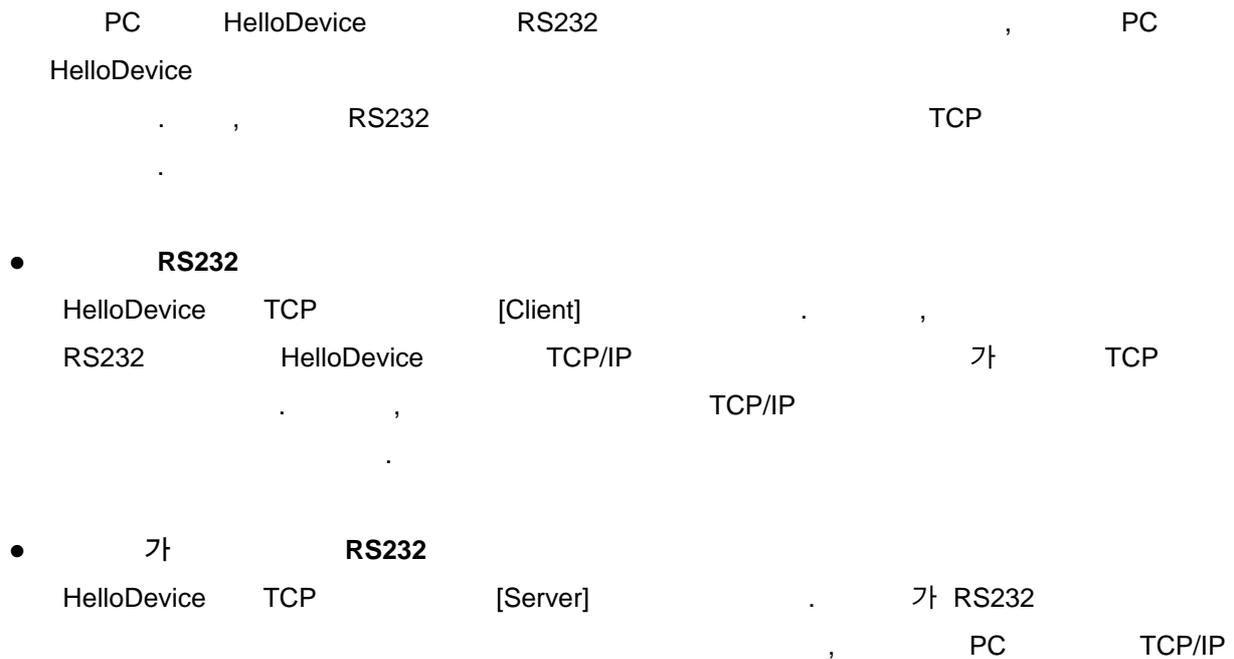
HelloDevice



5.10

## 6 RS232-TCP/IP

### 6.1 TCP



HelloDevice RS232 ,  
HelloDevice TCP/IP PC .

- 가  
HelloDevice TCP [Server/Client] , TCP  
TCP , RS232  
가 ,  
[Server] [Client] ,

### 6.1.1 TCP

- HelloDevice 가 TCP .
- TCP HelloDevice TCP/IP .
- HelloDevice TCP 가 ,  
TCP .
- RS232 가 , 가 TCP 가  
TCP 가  
.( 32KB)
- 가 TCP , RS232 IP
- , [RS232 ] , TCP
- , TCP ,  
TCP (Socket Close) .

### 6.1.2 TCP

- HelloDevice 가 TCP .
- TCP HelloDevice TCP/IP .
- TCP , HelloDevice TCP .
- HelloDevice TCP , RS232 가

- RS232 , TCP 가 , TCP 가 , 5 , 가
- RS232 , TCP . TCP , [RS232 ] , , TCP TCP TCP
- , HelloDevice RS232 가 , TCP (Socket Close)

### 6.1.3 TCP /

- HelloDevice 가 TCP
  - TCP HelloDevice TCP/IP
  - TCP , RS232 가 , TCP TCP , TCP 가 , RS232
  - , , 6.1.1
- 6.1.2

## 6.2 RS232

6.1 RS232 9

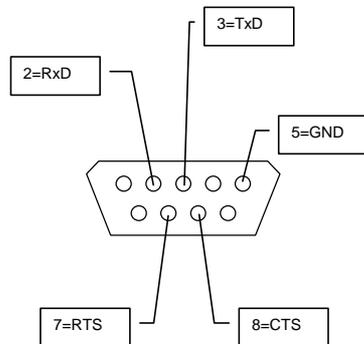
Pin 1	Received Line Signal Detector (Data Carrier Detect)	DCD
Pin 2	Received Data	RxD
Pin 3	Transmit Data	TxD
Pin 4	Data Terminal Ready	DTR
Pin 5	Signal Ground	GND
Pin 6	Data Set Ready	DSR
Pin 7	Request To Send	RTS
Pin 8	Clear To Send	CTS

Pin 9 Ring Indicator	RI
----------------------	----

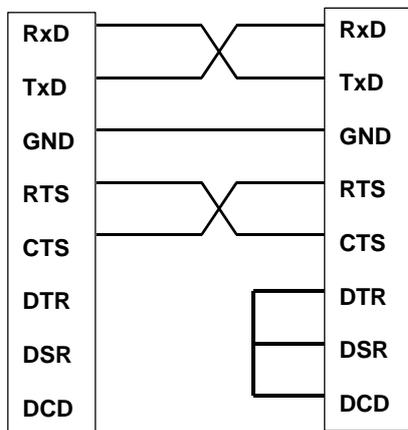
**6.1 RS232 9**

HelloDevice RS232 가 , HD1320/1320E RxD, TxD RS232 RTS, CTS , DTR, DSR , DTR, DSR 가 , 6.1 RS232 1(DCD), 4(DTR), 6(DSR) , RTS, CTS 가 .

**HelloDevice**

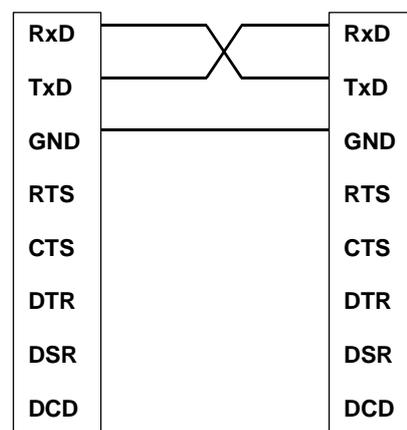


HelloDevice



(a)

HelloDevice



(b)

**6.1 HD1320/1320E RS232**

HD1321 , RS232 5V TTL , , RxD, TxD, RTS, CTS, DTR, DSR