LAMPIRAN A

LIST PROGRAM SERVER DAN CLIENT

KETERANGAN OBJEK KONTROL SERVER (VB.6)

OBJEK	PROPERTY	SETTING
Command1	Caption	Exit
Command2	Caption	START
Command3	Caption	UP
Command4	Caption	DOWN
Command5	Caption	LEFT
Command6	Caption	RIGHT
Command7	Caption	GRIP
Command8	Caption	RELEASE
Command8	Caption	BACKWARD
Command10	Caption	FORWARD
Command11	Caption	PWM CONTROL
Text1	Name	Text1
	TabIndex	0
	Text	
Text2	Name	Text2
	TabIndex	4
	Text	
Form	Name	CONCENTRATOR
	Caption	Contoller
Frame	Name	Frame1
	Caption	JOYSTIC
	BackColor	Biru
Label1	Name	Label1
	Alignment	0-LeftJustify
	Autosize	False
	Backstyle	1-opaque
	Borderstyle	0-none
	Caption	LIMIT-SWITCH
	Font	MS Sans Serif
	Forecolor	Hitam
	TabIndex	15
Label2	Name	Label2
	Alignment	2-Center
	Autosize	False
	Backstyle	1-opaque
	Borderstyle	0-none
	Caption	PORT
	Font	MS Sans Serif
	Forecolor	Hitam
	TabIndex	1
Label3	Name	Label3
	Alignment	0-LeftJustify
	Autosize	False

	Backstyle	1-opaque
	Borderstyle	0-none
	Caption	DISCONNECT
	Font	MS Sans Serif
	Forecolor	Hitam
	TabIndex	14
Label4	Name	Label4
	Alignment	2-Center
	Autosize	False
	Backstyle	1-opaque
	Borderstyle	0-none
	Caption	WIFI ROBOTIC CONTROLLER
	Font	MS Sans Serif
	Forecolor	Hitam
	TabIndex	17
Timer1	Name	Timer1
	Interval	500
	Enabled	False
Slider1	Name	Slider1
	Max	180
	Min	0
	ThickFrecuency	30
	Enabled	False
Winsock1	Name	Winsock1
	Protocol	0-sckTCPProtocol

- **LIST PROGRAM SERVER PADA VB 6.0 (PENGENDALI ROBOT)**

```
Dim pwm1 as Integer
```

```
Private Sub Command1_Click()
Unload CONCENTRATOR
End Sub
```

```
Private Sub Command2_Click()
Winsock1.Close
Winsock1.Listen
End Sub
```

```
Private Sub Command3_Click()
Winsock1.SendData ("w")
End Sub
```

```
Private Sub Command4_Click()
Winsock1.SendData ("s")
End Sub
```

```
Private Sub Command5_Click()
Winsock1.SendData ("a")
End Sub
```

```
Private Sub Command6_Click()
Winsock1.SendData ("d")
End Sub
```

```
Private Sub Command7_Click()
Winsock1.SendData ("q")
End Sub
```

```
Private Sub Command8_Click()
```

```
Winsock1.SendData ("z")
End Sub
```

```
Private Sub Command9_Click()
Winsock1.SendData ("x")
End Sub
```

```
Private Sub Command10_Click()
Winsock1.SendData ("e")
End Sub
```

```
Private Sub Form_Load()
Command3.Enabled = False
Command4.Enabled = False
Command5.Enabled = False
Command6.Enabled = False
Command7.Enabled = False
Command8.Enabled = False
Command9.Enabled = False
Command10.Enabled = False
Command11.Enabled = False
Slider1.Enabled = False
Timer1.Enabled = False
Winsock1.LocalPort = Text1.Text
End Sub
```

```
Private Sub Command11_Click()
Select Case (pwm1)
Case "0": Winsock1.SendData ("p")
Case "30": Winsock1.SendData ("o")
```

```
Case "60": Winsock1.SendData ("i")
Case "90": Winsock1.SendData ("u")
Case "120": Winsock1.SendData ("y")
Case "150": Winsock1.SendData ("t")
Case "180": Winsock1.SendData ("r")
End Select
End Sub
```

```
Private Sub Text1_Change()
Winsock1.LocalPort = Text1.Text
End Sub
```

```
Private Sub Timer1_Timer()
pwm1 = Slider1.Value
pwm1 = pwm1 \ 30
pwm1 = pwm1 * 30
Slider1.Value = pwm1
End Sub
```

```
Private Sub Winsock1_Connect()
Label3.Caption = "CONNECTED"
Command3.Enabled = True
Command4.Enabled = True
Command5.Enabled = True
Command6.Enabled = True
Command7.Enabled = True
Command8.Enabled = True
Command9.Enabled = True
Command10.Enabled = True
```

```
Command11.Enabled = True
Slider1.Enabled = True
Timer1.Enabled = True
End Sub
```

```
Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)
Label3.Caption = "CONNECTED"
Command3.Enabled = True
Command4.Enabled = True
Command5.Enabled = True
Command6.Enabled = True
Command7.Enabled = True
Command8.Enabled = True
Command9.Enabled = True
Command10.Enabled = True
Command11.Enabled = True
Slider1.Enabled = True
Timer1.Enabled = True
If Winsock1.State <> sckClosed Then Winsock1.Close
Winsock1.Accept requestID
Winsock1.SendData "Connection Succes"
End Sub
```

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
Dim dat As String
Winsock1.GetData dat, vbString
Text2.Text = dat
End Sub
```

```
Private Sub Winsock1_Error(ByVal Number As Integer, Description As String, ByVal
Scode As Long, ByVal Source As String, ByVal HelpFile As String, ByVal HelpContext As
Long, CancelDisplay As Boolean)

MsgBox "Error : " & Err.Description

Winsock1.Close

Winsock1.Listen

End

End Sub
```

- **LIST PROGRAM PADA CODEVISION AVR (CLIENT)**

```
/*****************************************************
This program was produced by the
CodeWizardAVR V1.25.3 Professional
Automatic Program Generator
© Copyright 1998-2007 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project :
Version :
Date    : 11/15/2007
Author  : F4CG
Company : F4CG
Comments:


Chip type           : ATmega16
Program type        : Application
Clock frequency     : 12.000000 MHz
Memory model        : Small
External SRAM size  : 0
Data Stack size     : 256
*****************************************************/

#include <mega16.h>
#include<delay.h>
char temp;
int a;

#define RXB8 1
#define TXB8 0
#define UPE 2
#define OVR 3
#define FE 4
#define UDRE 5
#define RXC 7

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<OVR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

// USART Receiver buffer
#define RX_BUFFER_SIZE 8
char rx_buffer[RX_BUFFER_SIZE];
```

```
#if RX_BUFFER_SIZE<256
unsigned char rx_wr_index,rx_rd_index,rx_counter;
#else
unsigned int rx_wr_index,rx_rd_index,rx_counter;
#endif

// This flag is set on USART Receiver buffer overflow
bit rx_buffer_overflow;

// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)
{
char status,data;
status=UCSRA;
data=UDR;
if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
   {
   rx_buffer[rx_wr_index]=data;
   if (++rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
   if (++rx_counter == RX_BUFFER_SIZE)
      {
      rx_counter=0;
      rx_buffer_overflow=1;
      };
   };
a = 1;                                          //bila ada interrupt maka var a =1
}

#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
char data;
while (rx_counter==0);
data=rx_buffer[rx_rd_index];
if (++rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
#asm("cli")
--rx_counter;
#asm("sei")
return data;
}
#pragma used-
#endif

// USART Transmitter buffer
```

```
#define TX_BUFFER_SIZE 8
char tx_buffer[TX_BUFFER_SIZE];

#if TX_BUFFER_SIZE<256
unsigned char tx_wr_index,tx_rd_index,tx_counter;
#else
unsigned int tx_wr_index,tx_rd_index,tx_counter;
#endif

// USART Transmitter interrupt service routine
interrupt [USART_TXC] void usart_tx_isr(void)
{
if (tx_counter)
   {
   --tx_counter;
   UDR=tx_buffer[tx_rd_index];
   if (++tx_rd_index == TX_BUFFER_SIZE) tx_rd_index=0;
   };
}

#ifndef _DEBUG_TERMINAL_IO_
// Write a character to the USART Transmitter buffer
#define _ALTERNATE_PUTCHAR_
#pragma used+
void putchar(char c)
{
while (tx_counter == TX_BUFFER_SIZE);
#asm("cli")
if (tx_counter || ((UCSRA & DATA_REGISTER_EMPTY)==0))
   {
   tx_buffer[tx_wr_index]=c;
   if (++tx_wr_index == TX_BUFFER_SIZE) tx_wr_index=0;
   ++tx_counter;
   }
else
   UDR=c;
#asm("sei")
}
#pragma used-
#endif

// Standard Input/Output functions
#include <stdio.h>

// Declare your global variables here

void main(void)
{
```

```
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTA=0x00;
DDRA=0xFF;

// Port B initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTB=0x00;
DDRB=0xFF;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTD=0x00;
DDRD=0xFF;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 1500.000 kHz
// Mode: Ph. & fr. cor. PWM top=ICR1
```

```
// OC1A output: Non-Inv.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x80;
TCCR1B=0x12;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x3A;
ICR1L=0x98;
OCR1AH=0x04;
OCR1AL=0x65;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 9600
UCSRA=0x00;
UCSRB=0xD8;
UCSRC=0x86;
```

```
UBRRH=0x00;
UBRRL=0x4D;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// Global enable interrupts
#asm("sei")
DDRD.1=1;
DDRA = 0x00;
PORTA = 0xFF;

while (1)
      {
      // Place your code here
if(a ==1)                                   //jika variabel a = 1 ada interupt
{
 temp = getchar();                          //ambil karakter dari komputer
 a=0;                                       //nilai variabel a kembali = 0
};

 if (PINA.0 == 0)                                   //cek sensor limit-switch
 {
 putchar('1');                              //mengirim karakter ke komputer
};
 if (PINA.1 == 0)
 {
 putchar('2');
};
 if (PINA.2 == 0)
 {
 putchar('3');
};
 if (PINA.3 == 0)
 {
 putchar('4');
};


     switch (temp)                          //seleksi kondisi variabel temp
     {
     case 'w': PORTB = 0x01;                        //menjalankan motor dc
     delay_ms(500);
     break;
     case 's': PORTB = 0x02;
```

```
            delay_ms(500);
            break;
            case 'a': PORTB = 0x04;
            delay_ms(500);
            break;
            case 'd': PORTB = 0x08;
            delay_ms(500);
            break;
            case 'q': PORTB = 0x10;
            delay_ms(500);
            break;
            case 'z':PORTB = 0x20;
            delay_ms(500);
            break;
            case 'e': PORTB = 0x40;
            delay_ms(500);
            break;
            case 'x': PORTB = 0x80;
            delay_ms(500);
            break;
            case 'p': OCR1A = 0x1C2;                    // menjankan motor servo
            break;
            case 'o': OCR1A = 0x2A3;
            break;
            case 'i': OCR1A = 0x384;
            break;
            case 'u': OCR1A = 0x465;
            break;
            case 'y': OCR1A = 0x546;
            break;
            case 't': OCR1A = 0x627;
            break;
            case 'r': OCR1A = 0x708;
            break;
            }
            PORTB = 0x00;                       //port b kembali bernilai logic 0
             temp = 0;                         // variabel temp kembali bernilai 0
    };
```

LAMPIRAN B

SPESIFIKASI ACCESS POINT D-LINK DI-524

Spesifikasi Access Point D-Link DI-524

Specifications	
Standards	• IEEE 802.11g • IEEE 802.11b • IEEE 802.3 • IEEE 802.3u
Wireless Signal Rates* with Automatic Fallback	• 54Mbps • 48Mbps • 36Mbps • 24Mbps • 18Mbps • 12Mbps • 11Mbps • 9Mbps • 6Mbps • 5.5Mbps • 2Mbps • 1Mbps
Security	• 802.1X • 64-, 128-bit WEP • WPA — Wi-Fi Protected Access (WEP with TKIP, MIC, IV Expansion, Shared Key Authentication)
Modulation Technology	• Orthogonal Frequency Division Multiplexing (OFDM)
Receiver Sensitivity*	• 54Mbps OFDM, 10% PER,-68dBm) • 48Mbps OFDM, 10% PER,-68dBm) • 36Mbps OFDM, 10% PER,-75dBm) • 24Mbps OFDM, 10% PER,-79dBm)

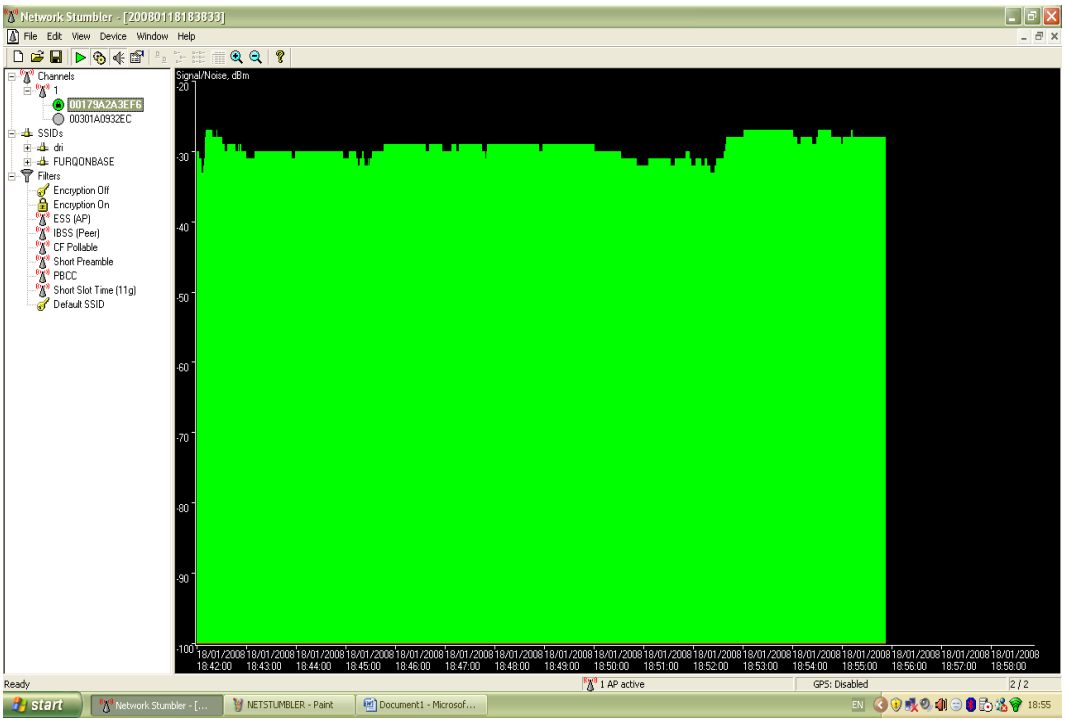
	• 18Mbps OFDM, 10% PER,-82dBm) • 12Mbps OFDM, 10% PER,-84dBm) • 11Mbps CCK, 8% PER,-82dBm) • 9Mbps OFDM, 10% PER,-87dBm) • 6Mbps OFDM, 10% PER,-88dBm) • 5.5Mbps CCK, 8% PER,-85dBm) • 2Mbps QPSK, 8% PER,-86dBm) • 1Mbps BPSK, 8% PER,-89dBm)
VPN Pass Through/Multi-Sessions	• PPTP • L2TP • IPSec
Device Management	• Web-Based – Internet Explorer v6 or later; Netscape Navigator v6 or later; or other Java- enabled browsers. • DHCP Server and Client
Advanced Firewall Features	• NAT with VPN Pass-through (Network Address Translation) • MAC Filtering • IP Filtering • URL Filtering • Domain Blocking • Scheduling
Wireless Signal Range*	• Indoors: Up to 328 ft (100 meters) • Outdoors: Up to 1312 ft (400 meters)
Wireless Frequency Range	• 2.4GHz to 2.462GHz
Wireless Transmit Power	• 15dBm ± 2dBm
External Antenna Type	• Single detachable reverse SMA
Operating Temperature	• 32°F to 131°F (0°C to 55°C)

Humidity	• 95% maximum (non-condensing)
Safety & Emissions	• FCC
LEDs	• Power • Status • WAN • WLAN (Wireless Connection) • LAN (10/100)
Dimensions	• L = 5.6 inches (142mm) • W = 4.3 inches (109mm) • H = 1.2 inches (31mm)
Weight	• 0.49lbs (22g)
Warranty	• 3 Year

LAMPIRAN C

SOFTWARE NETSTUMBLER

Software Netstumbler



Untuk melakukan site survey, cukup mudah dengan NetStumbler. NetStumbler akan jalan dan otomatis mencari / *scanning* frekuensi untuk melihat Access Point yang sedang beroperasi pada frekuensi *WiFi*. NetStumbler akan melaporkan :

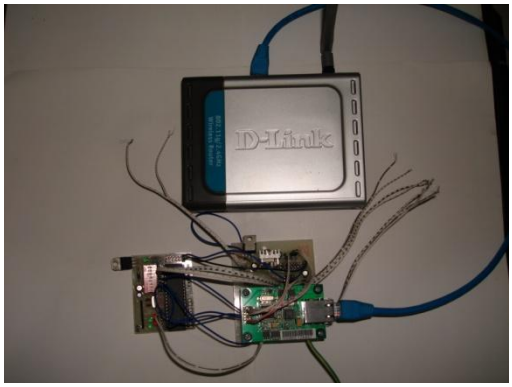
- MAC address Access Point dan frekuensinya.
- Channel yang digunakan Access Point.
- ESSID Access Point.
- Nama Access Point.
- Kekuatan Sinyal yang diterima Access Point. Terdiri dua warna yaitu hijau yang mengindikasikan baik dan kuning yang mengindikasikan kurang baik.
- Signal to Noise Ratio (SNR).

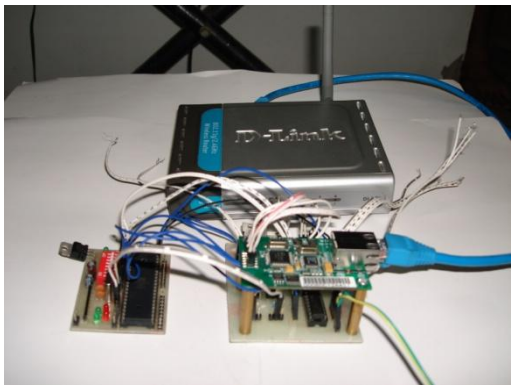
NetStumbler akan memperlihatkan rekaman grafik dari Signal to Noise Ratio (SNR) sebagai fungsi waktu. Kita juga dapat menyimpan hasil laporan sinyal untuk perioda tertentu.

LAMPIRAN D

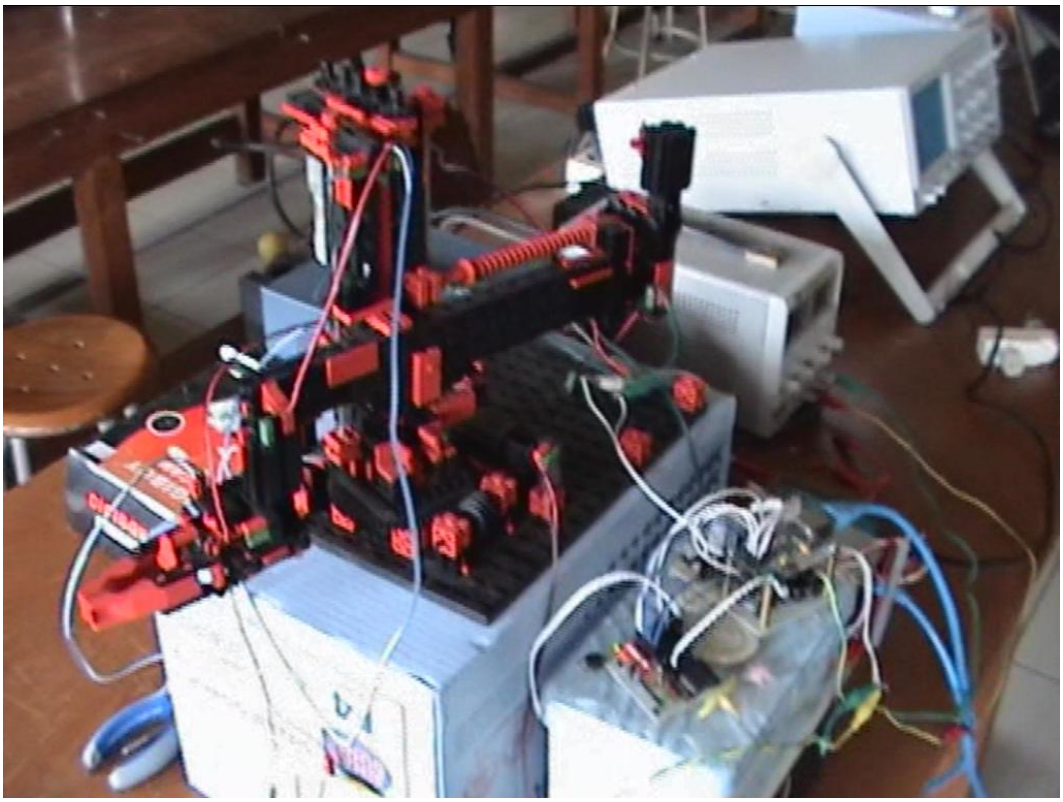
FOTO ALAT DAN ROBOT

- **FOTO ALAT**





- **ROBOT**



LAMPIRAN E

DATASHEET ATMEGA 16

LAMPIRAN F

SPESIFIKASI ETHERNET TO SERIAL GATEWAY

(EGSR-7150MJ)

EG-SR-7150MJ

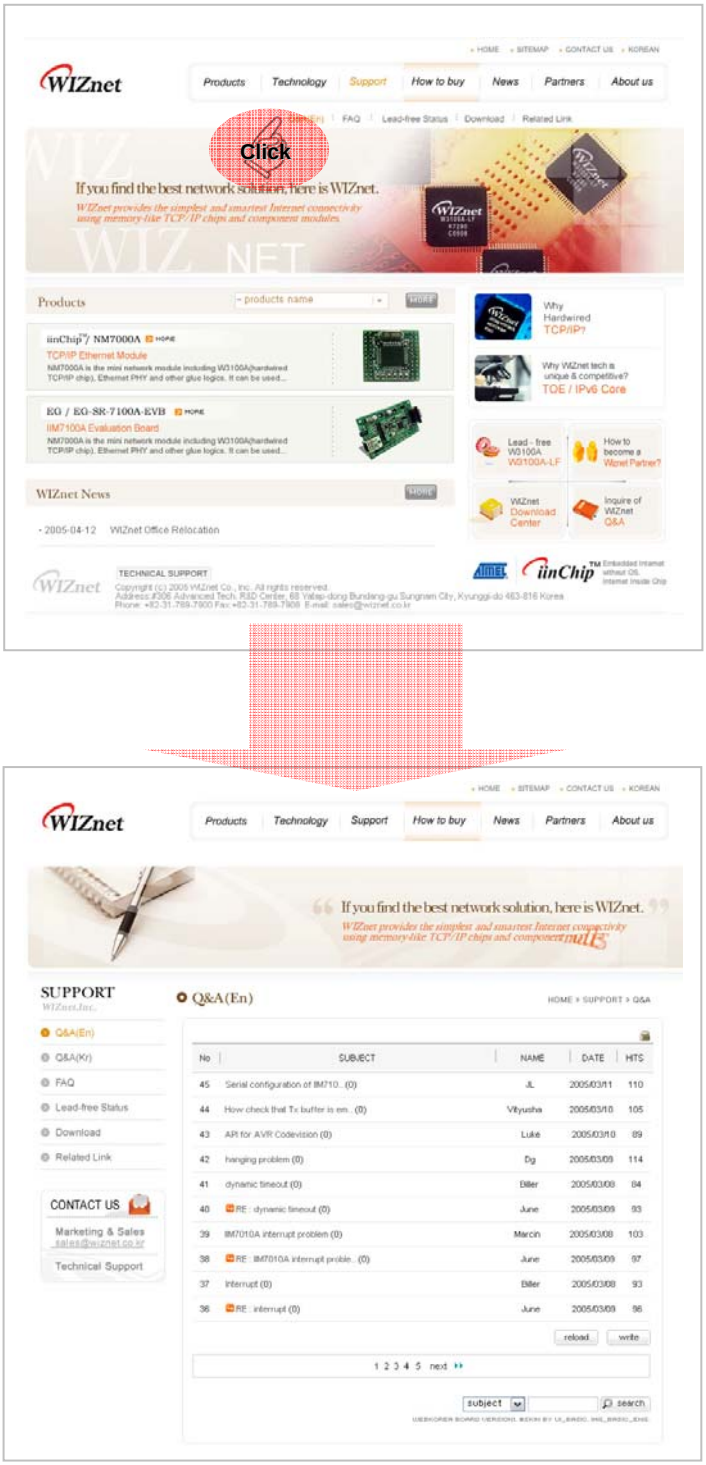
User's Manual

(V 1.1)

WIZnet

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

If you have any question about WIZnet Products, write them down onto our Q&A Board on our website at www.wiznet.co.kr. A WIZnet engineer will promptly provide you with an answer.



EG-SR-7150MJ User's Manual

1. Introduction

The EG-SR-7150MJ is a gateway module that converts serial data into TCP/IP data type. It transmits the data sent by a serial equipment to the Internet and TCP/IP data to the equipment.

With the EG-SR-7150MJ mounted with an RJ-45 connector, users can have an easier and quicker interface with the Ethernet.

The EG-SR-7150MJ provides serial commands, with which the developers of any serial device can add local configuration capability to their products. For example, a card reader developer can program the keypad on a card reader to configure serial or network on-site without the use of a laptop or PC.

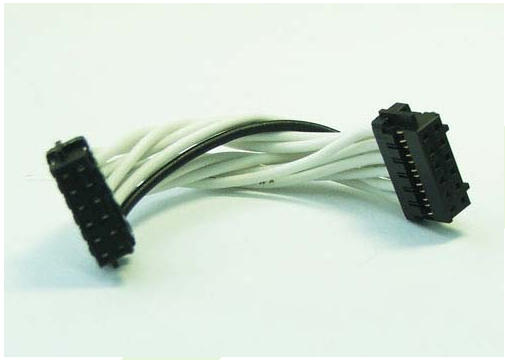
1.1. Key Features

- Ready-to-go serial to Ethernet gateway module mounted with an RJ-45 connector
- Serial Command Support
  - Simple command frame format
  - Comprehensive & readable command set for network and serial settings
  - On-site configuration without PC
- High stability & reliability by using a W3150A WIZnet Chip, a fully-hardwired TCP/IP stack
- Easy and powerful configuration program
- 10/100Mbps Ethernet interface, Max. 230Kbps Serial interface
- RoHS compliant

1.2. Products Contents (EG-SR-7150MJ-EVB)

The EG-SR-7150MJ-EVB, the evaluation kit for the EG-SR-7150MJ contains the following items;

	EG-SR-7150MJ Module

	Test Board for EG-SR-7150MJ
	12pin Cable (Connecting EG-SR-7150MJ to Test Board)
	Serial Cable (Connecting Serial Device to Test Board)
	LAN Cable (Connecting EG-SR-7150MJ to Host)
	Power (DC 5V Adaptor)

	CD (Containing Manual, H/W & S/W Materials)

☞ Please immediately notify your sales representative if any of the items above is missing or damaged.

1.3. Specifications of the EG-SR-7150MJ

Category	Specifications
Form Factor	2mm Pitch 2x6 pins, 62x40 mm
LAN Interface	10/100 Mbps auto-sensing, RJ-45 connector
Protocol	TCP, UDP, IP, ARP, ICMP, MAC, (IGMP, PPPoE)
CPU	AT89C51RC2 (8bit MCU and 32K Flash)
Serial Interface	RS 232 (LVTTL)
Serial Signals	TXD, RXD, RTS, CTS, GND
Serial Parameters	Parity : None, Even, Odd Data bits : 7, 8 Flow control : RTS/CTS, XON/XOFF Speed : up to 230Kbps
Management	Configuration utility based on Windows
Temperature	0℃~70℃ (Operating), -40℃~85℃ (Storage)
Humidity	10~90%
Power	150mA @ 3.3V (max)
Size	40mm x 62mm x 17mm

2. Getting Started

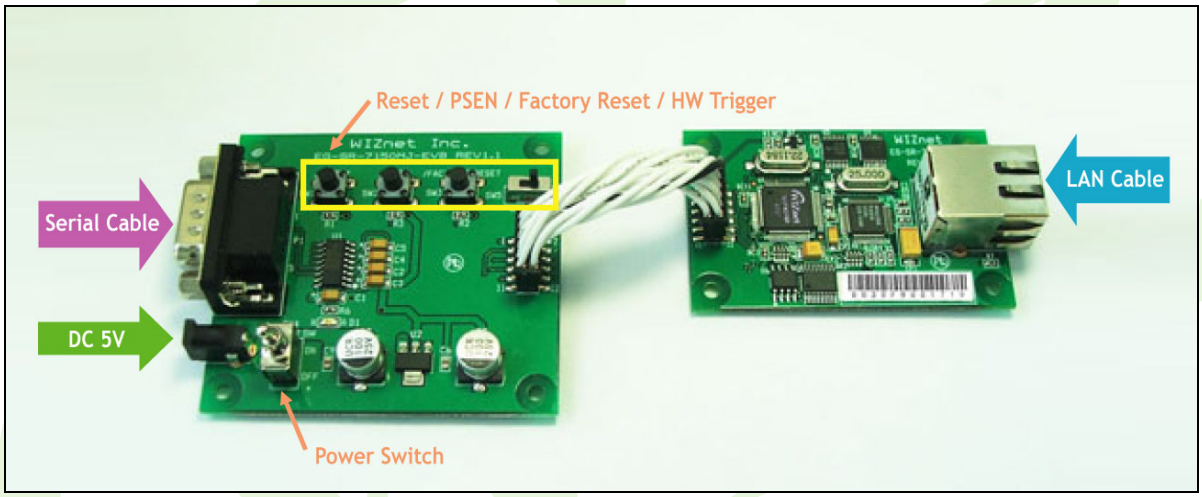
This Chapter describes how to set up and configure the EG-SR-7150MJ.

The following items are required to get started.

- Power Cable (included in the EG-SR-7150MJ-EVB package)
- Serial and Ethernet Cables (included in the of EG-SR-7150MJ-EVB package)
- PC or Laptop with Network Interface Card (hereafter, NIC) and/or one RS232 serial port

2.1. Hardware Installation procedure

Follow steps below to prepare the module and evaluation board for testing.



STEP 1: Connect the EG-SR-7150MJ module to the test board by using the 12pin cable.

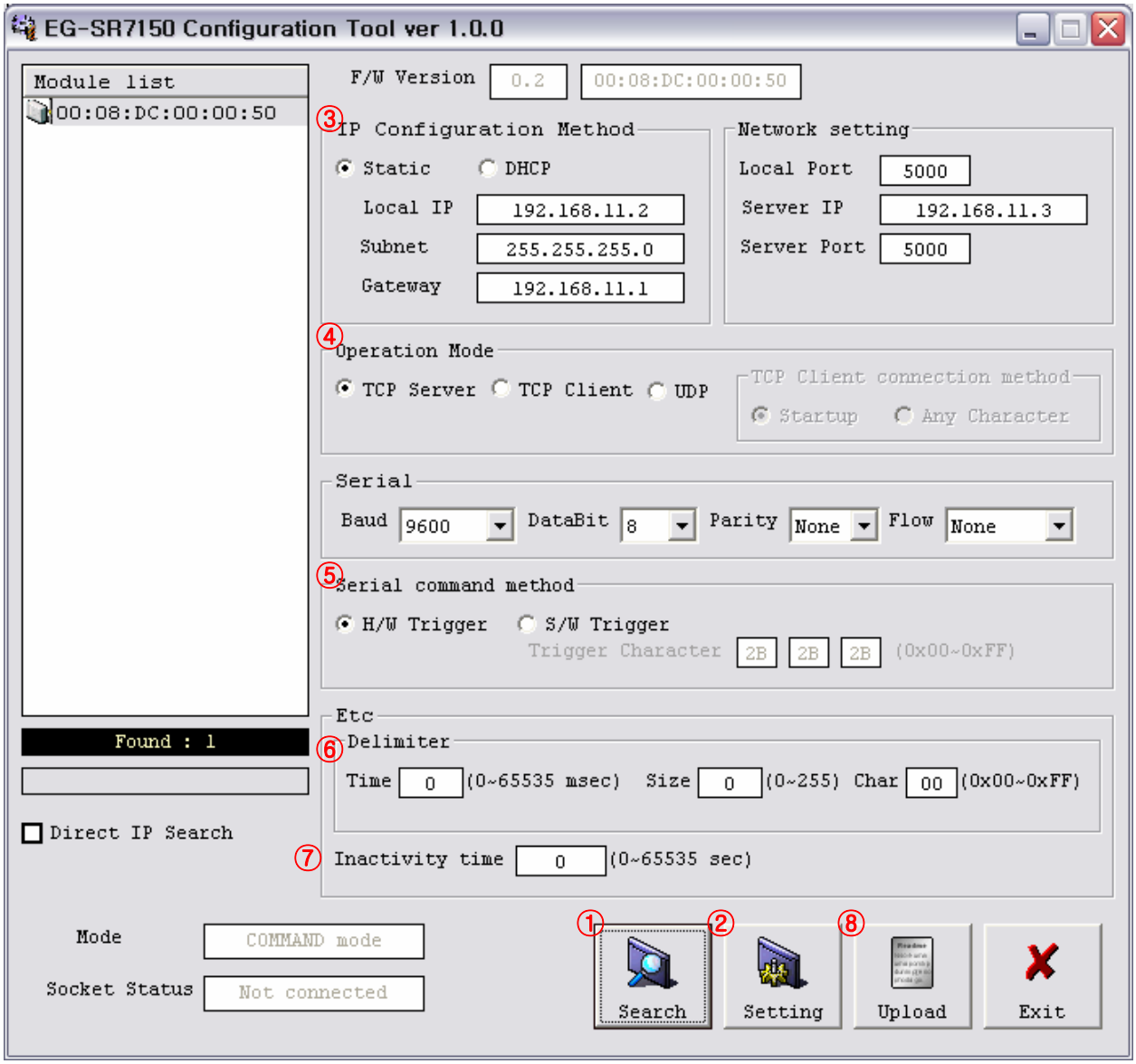
STEP 2: Connect the 5V DC power line to the power jack of the test board.

STEP 3: Use the RJ45 Ethernet cable in order to connect the module to an Ethernet network.

STEP 4: Use the serial data cable to connect the test board to a serial device.

2.2. Configuration Tool

2.2.1. Configuration tool features



① Search

The Search function is used to search all modules existing on the same Subnet. The UDP broadcast is used for searching modules on a LAN.

The MAC address for a searched module will be listed in the "**Module list**".

If **Direct IP Search** is checked, TCP will be used for searching instead of UDP. This mode is used more for searching the EG-SR-7150MJ modules on remote networks than local networks with the same subnet. An IP address assigned to the module will be required.

② Setting

If you select one of the MAC addresses listed in the "**Module list**", the configuration value of the selected module will be displayed. After changing each value in the configuration program, click the "**Setting**" button to complete the configuration.

The module will be initialized with the new configurations.

③ IP Configuration method: Static, DHCP

Static: The IP address can be manually assigned by users.

DHCP: The module assigns IP, subnet and gateway addresses by acquiring them from the DHCP server

☞ Other configurations should be set manually except for the IP configuration of DHCP.

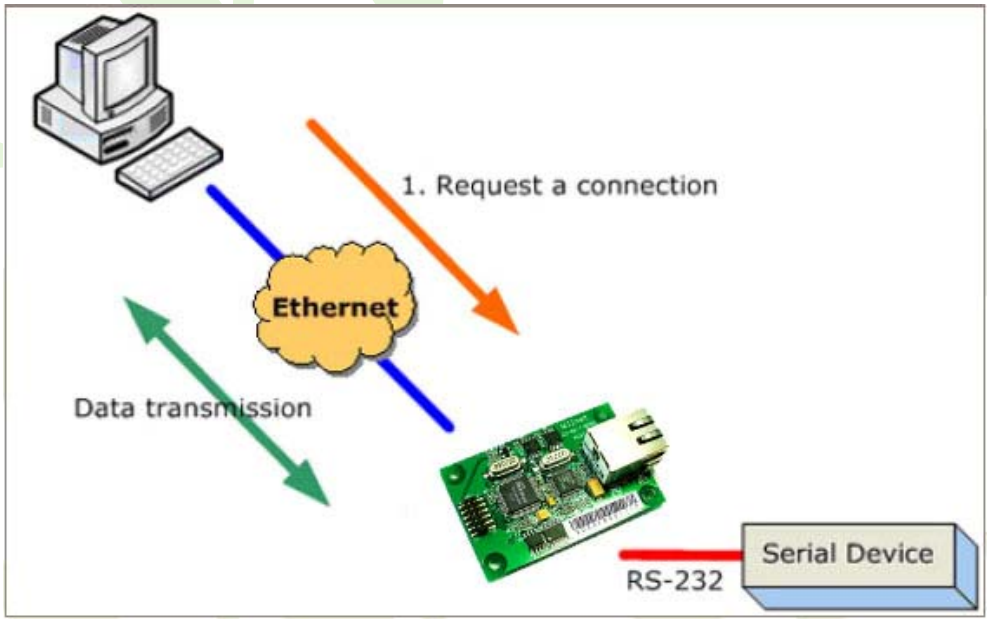
④ Operation mode: TCP server, TCP client, UDP

Three different operation modes are supported — TCP Server, TCP Client, and UDP.

The main difference between the TCP and UDP protocols is that TCP guarantees the delivery of data by requesting the recipient to send an acknowledgement to the sender. On the other hand, UDP does not require this type of verification, so data can be delivered quicker, but its delivery can not be guaranteed.

The TCP Server and TCP Client mode are related to the first step of connection establishment. Once the connection is established, data will be transparently transmitted in both directions (from Server to Client or from Client to Server).

TCP server mode

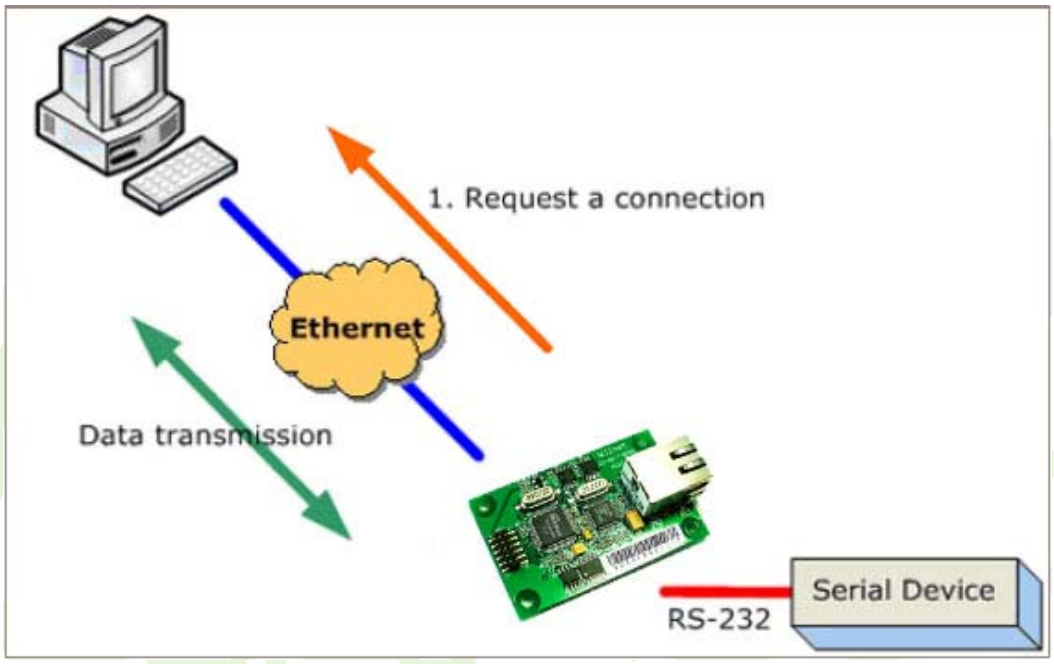


To operate this mode, the **Local IP, Subnet, gateway address and local port number** should be configured. The EG-SR-7150MJ waits to be connected by the host computer, allowing the host computer to establish a connection and get data from the serial device.

As illustrated in the figure above, the data transmission is as follows:

1. The host connects to the EG-SR-7150MJ which is configured as TCP Server Mode.
2. Once the connection is established, data can be transmitted in both directions - from the host to the EG-SR-7150MJ, and from the EG-SR-7150MJ to the host.

TCP client mode

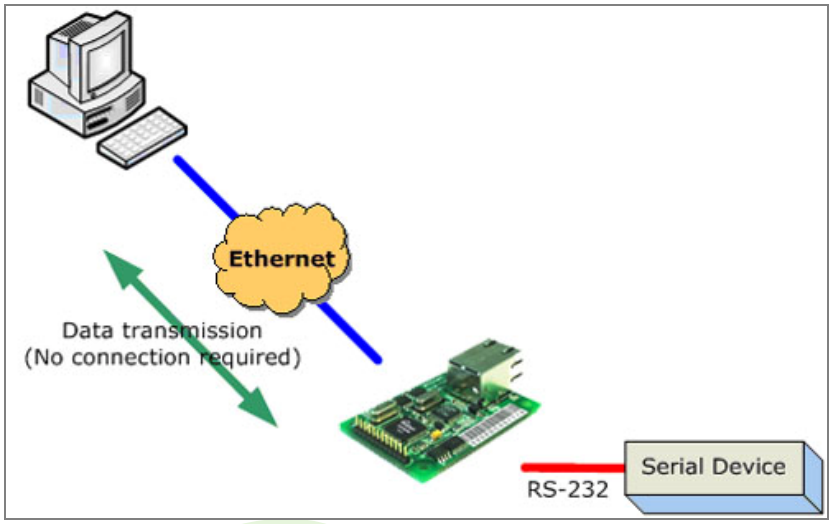


To operate this mode, the **Local IP, Subnet, gateway address, server IP, server port number** should be set. In the **TCP Client mode**, the EG-SR-7150MJ proceeds active open for establishing a TCP connection to a host computer.

As illustrated in the figure above, data transmission is as follows:

1. The EG-SR-7150MJ operating as TCP Client Mode establishes a connection based on the condition set in the **TCP client connection method** (**Startup, Any character**). i.e. the EG-SR-7150MJ can try to connect as soon as one starts up(**Startup**), or later when data from serial device arrives. (**Any character**).
2. After the connection is established, data can be transmitted in both directions - from the host to the EG-SR-7150-MJ, and from the EG-SR-7150-MJ to the host.

UDP mode



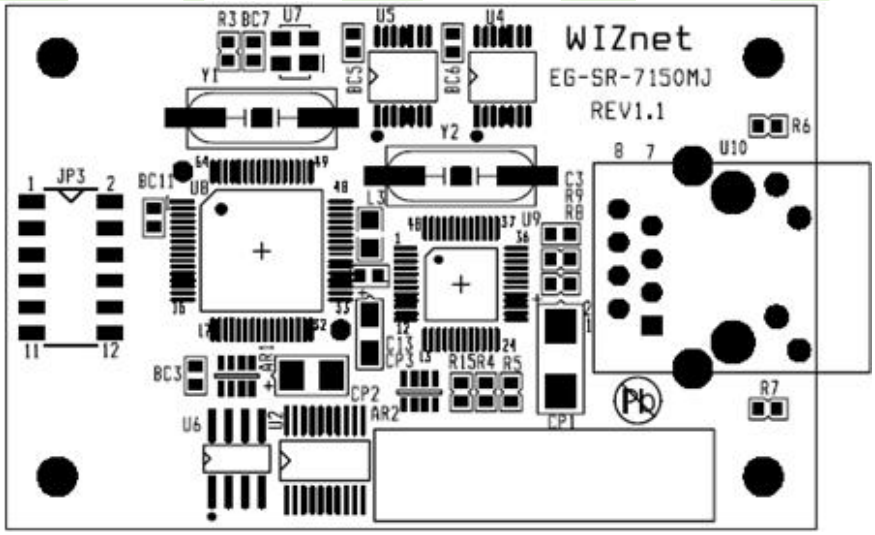
In UDP mode, any TCP/IP connection procedure is not required.

⑤ Serial command method: H/W trigger, S/W trigger

With this menu, you can designate how the Serial command mode can be entered. Two types are supported - H/W Trigger and S/W Trigger.

H/W trigger: Serial command mode can be triggered by pulling H/W trigger pin to low. It can be exited by pulling it to high.

```
-----------------------------------
            JP3
-----------------------------------
/RESET - 1   2 -  3.3V
   RXD - 3   4 - 3.3V
   CTS - 5   6 - /FACTORY_RESET
   TXD - 7   8 - /HW_TRIGGER
   RTS - 9   10 - /PSEN
 GND - 11  12 - GND
```



S/W trigger: Serial command mode can be triggered when 3 user-defined characters are detected. It can be exited by using the WR command.

⑥ Delimiter: Time, Size, Character

You can designate how the serial data can be packed and sent to the Ethernet. There are 3 delimiters - Time, Size and Character. If all of them are set as '0', whenever the serial data arrives, they will be sent to the Ethernet without any condition. When any of the three delimiters is satisfied, data can be sent to the Ethernet.

Ex) Delimiter: Size=10, Char=0x0D

Serial data: 0123456789abc

Ethernet data: 0123456789

☞ "abc" data remains in the serial buffer of module

⑦ Inactivity time

After the connection is established, if there is not any data transmission within the time defined in "**Inactivity time"**, the connection will be automatically closed.

⑧ Upload

Upload the firmware through the network.

☞ After uploading the firmware, 10~20 seconds are required for initialization.

2.3. Serial Communication Specification

In this chapter, we describe the structure of the data frames used in issuing commands and receiving responses to and from the device.

2.3.1. Frame Format

Command Frame format

Descriptor	STX	Command code	Parameter	ETX
Length(bytes)	1	2	Variable	1

Reply Frame format

Descriptor	STX	Reply code	Parameter	ETX
Length(bytes)	1	1	Variable	1

2.3.2. STX & ETX

Setting	Comments
STX	'<' : Hex = 3Ch
ETX	'>' : Hex = 3Eh

2.3.3. Reply Code

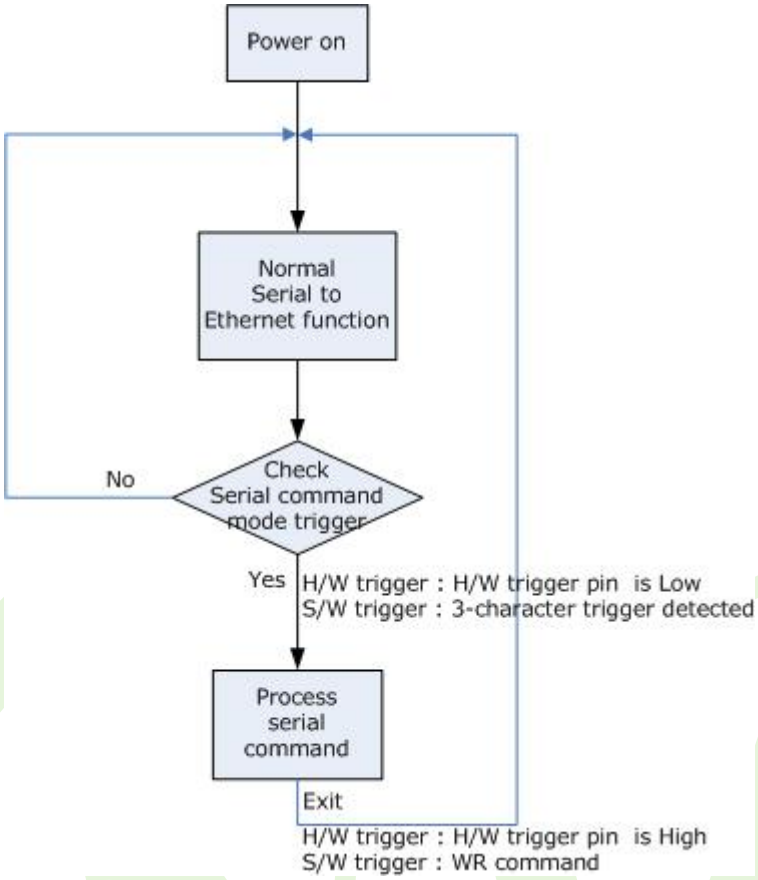
Reply	Comments
S	Command was successful
F	Command failed
1	Invalid command
2	Invalid parameter
E	Enter serial command mode

2.3.4. Command Code

Command	Parameter	Comments
WI	xxx.xxx.xxx.xxx (eg. 192.168.11.133)	Set Local IP
WS	xxx.xxx.xxx.xxx (eg. 255.255.255.0)	Set Subnet mask
WG	xxx.xxx.xxx.xxx (eg. 192.168.11.1)	Set Gateway
WP	0~65535	Set Local IP's port number
WD	0 : Static 1 : DHCP	Set the IP configuration method
WM	0 : TCP server 1 : TCP client 2 : UDP	Set the operation mode
WC	0 : startup 1 : any character	TCP client method
WB	XXXX eg. [Baudrate]1: 115200, 2: 57600, 3: 38400, 4: 19200, 5: 9600, 6: 4800, 7: 2400,8: 1200 [data byte] 7 : 7bit, 8bit [parity] 0 : no parity, 1 : Odd, 2 :Even [Flow] 0 : no, 1 : Xon/Xoff, 2 :RTS/CTS	Set the serial baud rate, data, parity and flow control. 4bytes:[Baud][data byte][parity][flow]
WT	0 : Disable 1 : H/W trigger 2 : S/W trigger	Set the serial command method
WE	xxxxxx (eg. In hex format : 2B 2B 2B)	Set the command mode character

WX	xxx.xxx.xxx.xxx (eg. 192.168.11.144)	Set server IP address
WN	0~65535	Set server port number
WR		Restart
OC	XX	Set delimiter character in hex
OS	0~255	Set delimiter size
OT	0~65535	Set delimiter time
OI	0~65535	Set Inactivity timer value
Command	Parameter	Comments
RI		Get Local IP
RS		Get Subnet mask
RG		Get Gateway
RP		Get Local IP's port number
RD		Get the IP configuration method
RM		Get the operation mode
RC		Get the TCP client method
RB		Get the serial baud rate
RT		Get the serial command method
RE		Get the command mode character
RF		Get the firmware version
RX		Get the server IP address
RN		Get the server port number
QC		Get delimiter character in hex
QS		Get delimiter size
QT		Get delimiter time
QI		Get Inactivity timer value

2.4. Operation Flow



2.5. Factory Default

While the Factory Reset is low and the /Reset is applied, the module is initialized with the factory default value.

IP configuration	Static
Local IP address	192.168.11.2
Subnet mask	255.255.255.0
Gateway address	192.168.11.1
Local port number	5000
Server IP address	192.168.11.3
Server port number	5000
Operation mode	TCP server mode
Serial port	9600 bps 8-N-1
Serial command method	H/W trigger

3. Demonstration and Test

In this chapter, three examples are given to show how functions of the EG-SR-7150MJ can be tested. The testing environment is as follows:

Hardware

- PC having RS-232 serial port.
- EG-SR-7150MJ & Test board

Software

- Windows operating system installed on testing PC.
- EG-SR-7150MJ Configuration tool
- Hyper Terminal Program

Testing Structure

- Ethernet cross cable to connect the LAN ports of PC and EG-SR-7150MJ.
- RS-232 cable to connect the COM port of PC (usually COM1 or COM2) and serial port of EG-SR-7150MJ-EVB.

3.1. Case 1: Getting IP address using H/W trigger

STEP1: Configure the trigger mode as "H/W trigger" in the Configuration Tool.

STEP2: Check the serial port setting such as baud rate of the module.

STEP3: Start HyperTerminal program and set the serial port of PC to the setting of module checked in STEP2.

STEP4: Pull H/W trigger pin to low to enter the serial command mode.

STEP5: Use HyperTerminal program to send "**<RI>**" (command to request IP address)

STEP6: HyperTerminal program displays "**<S192.168.11.2>**"

(It indicates that the command was successfully executed and IP address is 192.168.11.2)

STEP7: Pull H/W trigger pin to high to exit the serial command mode

3.2. Case 2: Changing IP address using H/W trigger

STEP1: Configure the trigger mode as "H/W trigger" in the Configuration Tool.

STEP2: Check the serial port setting such as baud rate of the module.

STEP3: Start HyperTerminal program and set the serial port of PC to the setting of module checked in STEP2.

STEP4: Pull H/W trigger pin to low to enter serial command mode.

STEP5: Use HyperTerminal program to send "**<WI192.168.11.10>**"

(command to change the IP address as 192.168.11.10)

STEP6: HyperTerminal program displays "**<S>**"
(Indicates the command was successfully executed)

STEP7: Use HyperTerminal program to send "**<RI>**" (command to request IP address)

STEP8: HyperTerminal program displays "**<S192.168.11.10>**"
(Indicates the command was executed successfully and IP address is 192.168.11.10)

STEP9: Pull H/W trigger pin to high to exit serial command mode

☞ All changes are applied after exit the serial command mode

3.3. Case 3: Changing IP address using S/W trigger

STEP1: Configure the trigger mode as S/W trigger at the Configuration program, and check the three trigger characters. For example, assume the trigger is "25 25 25"

STEP2: Check the serial port setting such as baud rate of the module.

STEP3: Start HyperTerminal program and set the serial port of the PC to the serial setting of the module checked in STEP2.

STEP4: Use HyperTerminal program to send three trigger characters to enter the serial command mode; **% % %** (in hex :0x25 0x25 0x25) in this case.

STEP5: Use HyperTerminal program to send "**<WI192.168.11.10>**"
(command to change the IP address as 192.168.11.10)

STEP6: HyperTerminal program displays "**<S>**"
(Indicate the command was executed successfully)

STEP7: Use HyperTerminal program to send "**<RI>**" (command to request IP address)

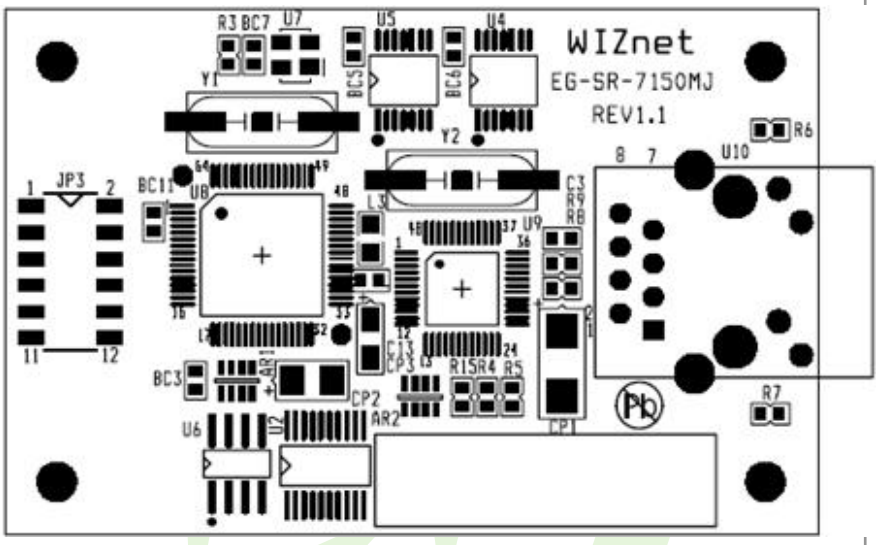
STEP8: HyperTerminal program displays "**<S192.168.11.10>**"
(Indicate the command was executed successfully and IP address is 192.168.11.10)

STEP9: Use HyperTerminal program to send "**<WR>**"
(command to exit serial command mode)

☞ All changes are applied after exiting the serial command mode.

4. PIN Assignment and Dimensions

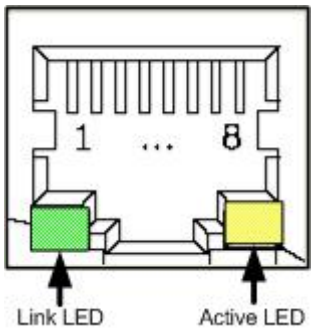
```
-----------------------------------
              JP3
-----------------------------------
/RESET - 1   2 -  3.3V
   RXD - 3   4 - 3.3V
   CTS - 5   6 - /FACTORY_RESET
   TXD - 7   8 - /HW_TRIGGER
   RTS - 9   10 - /PSEN
 GND - 11  12 - GND
```



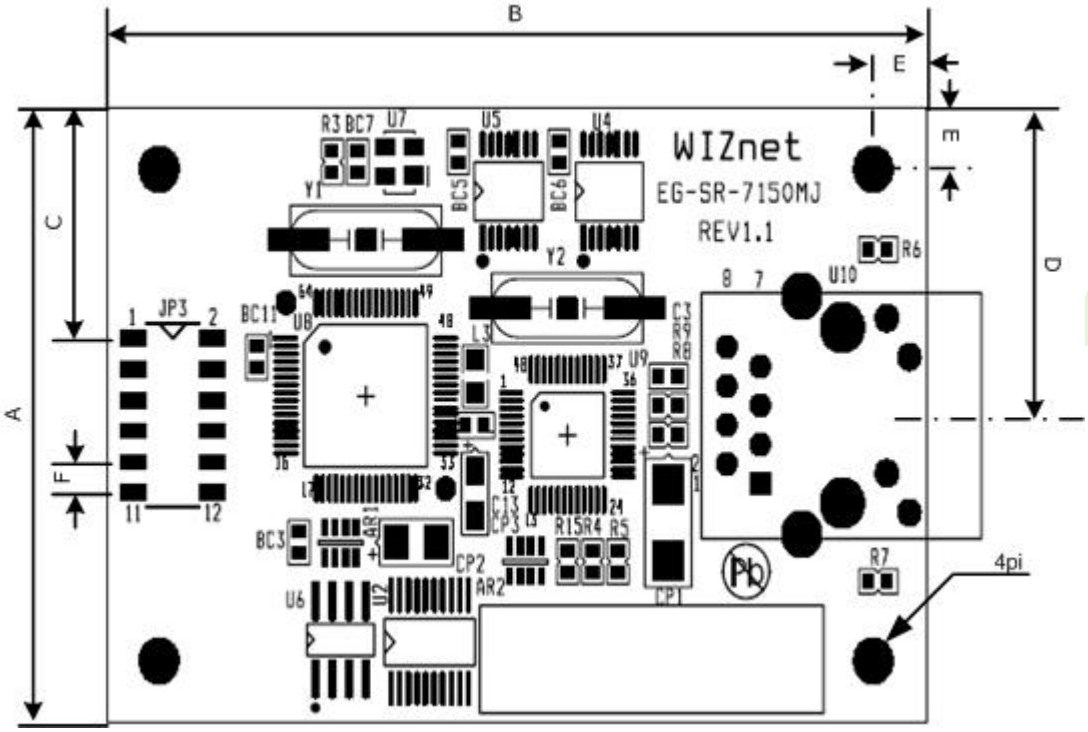
Name	Functions	I/O	
3.3V	Power		
/RESET	Low active reset Minimum 1.2 usec is required.	Input	
RXD	RS-232 Data Input	Input	
CTS	RS-232 Clear To Send	Input	Optional
TXD	RS-232 Data Output	Output	
RTS	RS-232 Request To Send	Output	Optional
Factory Reset	Pull Factory Reset to low and if /RESET is activated, the configuration is changed to factory default.	Input	
H/W Trigger	Pull H/W Trigger to low, enter the serial command mode	Input	
/PSEN	Pull /PSEN to low and if /RESET is activated, the module enter the bootloader for FLIP connection	Input	

☞ All signal levels are 3.3V LVTTL.

Ethernet port Pinouts

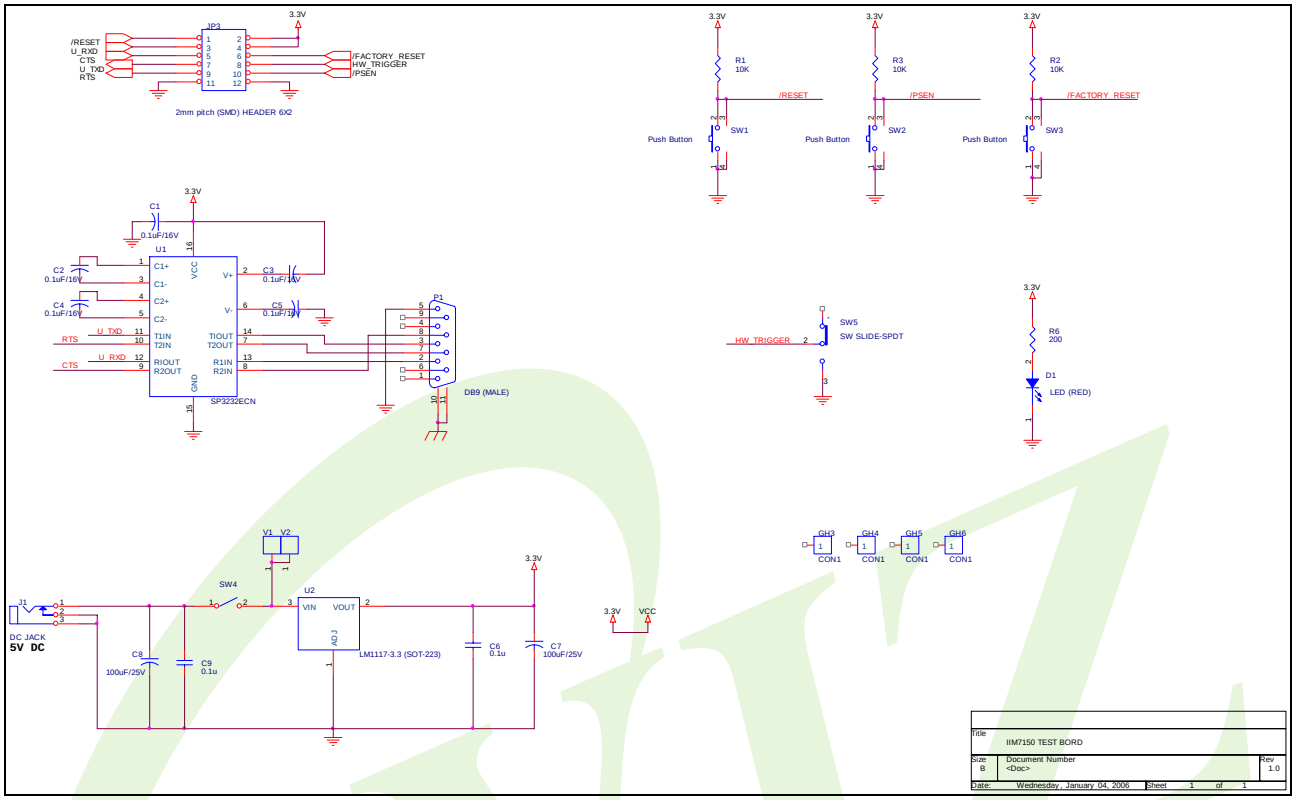


Pin	Signal
1	TX+
2	TX-
3	RX+
6	RX-



Symbol	Dimension(mm)
A	40
B	62
C	15
D	20
E	4
F	2

5. Reference Schematic



6. ETC

6.1. Firmware Uploading through the FLIP software

The following items are required to get started. :

- EG-SR-7150-MJ test board
- UART cross cable
- Program file in HEX file format
- FLIP utility installed on your PC

On a PC, one must have a file in HEX format to program the EG-SR-7150MJ. For example this file could be named “app.hex”.

Step 1

Connect the EG-SR-7150MJ test board to a PC with the UART cable supplied.

Important : If you have any program running on your PC with which COM port is used such as "Hyperterminal", be sure to connect the cable to the COM port not used.

Step 2

Power on the test board.

While pressing the /PSEN button, click the /RESET button. Then release the /PSEN button

Step 3

Run the ISP software named FLIP by ATMEL.

Step 4

Select the device by pushing the F2 button. Here you must choose AT89C51RC2.

Step 5

Set up the communication port by pushing the F3 button. Make sure to select the same port as the one you have plug in the UART cable of the EG-SR-7150MJ test board.

Step 6

Now, you should be connected to the board and able to program.

Now you will have to browse your PC to load your file in hex format.

Step 7

After programming, check if the BSB, SBV and SSB are set as FF, 00 and FF respectively.

6.2. Warranty

WIZnet Co., Ltd offers the following limited warranties applicable only to the original purchaser. This offer is non-transferable.

WIZnet warrants our products and its parts against defects in materials and workmanship under normal use for period of standard ONE(1)YEAR for the EG-SR-7150MJ Module, Evaluation Board and labor warranty after the date of original retail purchase. During this period, WIZnet will repair or replace a defective products or part free of charge.

Warranty Conditions:

1. The warranty applies only to products distributed by WIZnet or our official distributors.
2. The warranty applies only to defects in material or workmanship as mentioned above in 6.2 Warranty. The warranty applies only to defects which occur during normal use

and does not extend to damage to products or parts which results from alternation, repair, modification, faulty installation or service by anyone other than someone authorized by WIZnet Inc. ; damage to products or parts caused by accident, abuse, or misuse, poor maintenance, mishandling, misapplication, or used in violation of instructions furnished by us ; damage occuring in shipment or any damage caused by an act of God, such as lightening or line surge.

Procedure for Obtaining Warranty Service

1. Contact an authorized distributors or dealer of WIZnet Inc. for obtaining an RMA (Return Merchandise Authorization) request form within the applicable warranty period.
2. Send the products to the distributors or dealers together with the completed RMA request form. All products returned for warranty must be carefully repackaged in the original packing materials.
3. Any service issue, please contact to sales@wiznet.co.kr