

Enabled Just-In-Time serial to Ethernet Data Traffic Catcher Diagnostic Interface

- EKI-1500 series Serial Device
Server**

Just-In-Time (JIT) Diagnostic Tool

- Just-In-Time Diagnostic message is designed for debug use. These debug messages can assist customer or fronted-AE easily and quickly trace the system/data status for a specific serial port on a standard firmware without **rebooting** the system.
- Enable JIT Tool that can trace the data communication and status behavior from device server its point of view.
- This diagnostic message support system log with WebGUI interface and advanced remote system log message.
- This JIT Tool support for EKI-152x-BE and future series (-CE)

Configuring the JIT interface (1/3)

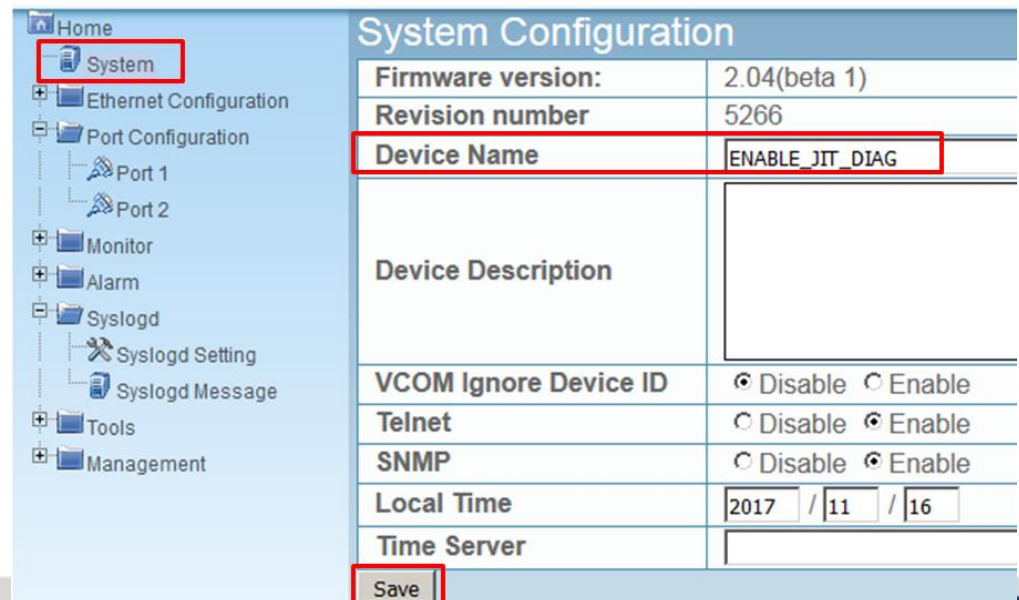
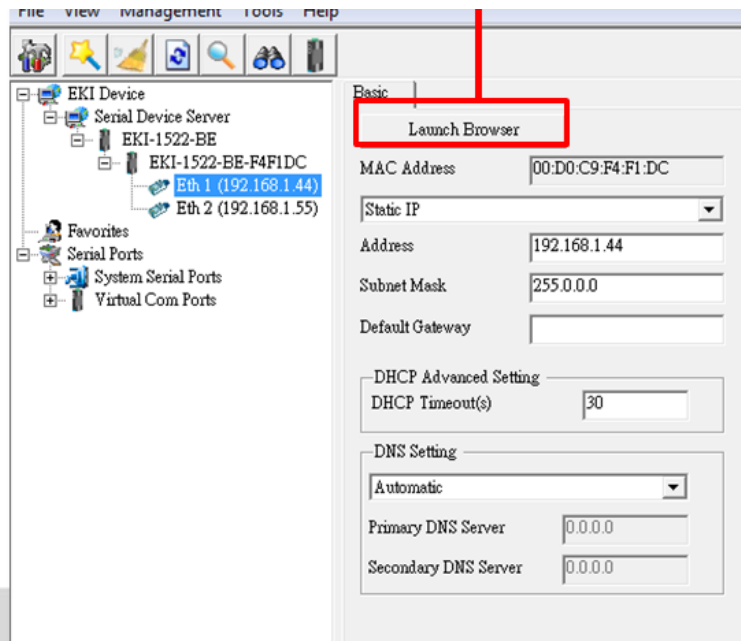
How To Enable “JIT Debug Message Mode”

Step 1: Using “Launch Browser” from Utility or Key in IP Address to open Web GUI interface

Step 2: Support for CE version and BE version (FW “1.99 or upper”)

Step 3: In “System” page, key in Device Name as “ENABLE_JIT_DIAG”

Step 4: “Save” to store the configuration



Configuring the JIT interface (2/3)

Make Sure System log message can be seen, go to “**Syslogd Setting**” page

Step 5: Check Syslogd is “**Enable**”

Step 6: Check VCOM/ USDG is “**Enable**” (p.s. this item is to enable the operation mode which you would like to observe)

Step 7: “**Save**” to store the configuration

ADVANTECH ICom

Home

- System
- Ethernet Configuration
- Port Configuration
 - Port 1
- Monitor
 - Port 1
- Alarm
- Syslogd
 - Syslogd Setting**
 - Syslogd Message

Syslogd Setting

Syslogd	<input checked="" type="radio"/> Enable <input type="radio"/> Disable
Syslogd Remote	<input type="radio"/> Enable <input checked="" type="radio"/> Disable
Syslogd Remote IP address	<input type="text"/>
VCOM	<input checked="" type="radio"/> Enable <input type="radio"/> Disable
USDG	<input checked="" type="radio"/> Enable <input type="radio"/> Disable

Save

Enable JIT Diagnostic Level

Step 8: Click the Port you would like to observe Like Port1 ->“Basic”

Step 9: Click the JIT Diag Level. It separate different level of Trouble shooting log. You can click all for display all level Diagnostic log.

Step 10: “Save” to save setting

The screenshot shows the Advantech ICom web interface for Port 1 configuration. The left sidebar contains a tree view with categories like Home, System, Ethernet Configuration, Port Configuration, Monitor, Alarm, Syslogd, Tools, and Management. Under Port Configuration, 'Port 1' is selected. The main area is titled 'Port 1 configuration' and has three tabs: 'Basic' (selected), 'Operation', and 'Advanced'. The 'Basic' tab contains several settings: Type (RS232), Baud Rate (9600), Parity (None), Data Bits (8), Stop Bits (1), and Flow Control (None). Below these is the 'JIT Diag Level' section, which is expanded to show five checked options: EXPT, CONN, DATA, PROTO, and TCPWR. At the bottom of the configuration area is a 'Save' button.

Port 1 configuration		
Basic	Operation	Advanced
Type	RS232	
Baud Rate	9600	
Parity	None	
Data Bits	8	
Stop Bits	1	
Flow Control	None	
JIT Diag Level	<input checked="" type="checkbox"/> EXPT <input checked="" type="checkbox"/> CONN <input checked="" type="checkbox"/> DATA <input checked="" type="checkbox"/> PROTO <input checked="" type="checkbox"/> TCPWR	
Save		

JIT Diagnostic Message

- Enable Just-In-Time diagnostic via key in “Enable_JIT_Diag” in Device Name of system page
- Recommend use Chrome browser to scroll wider the screen

The screenshot shows the ADVANTECH iCom web interface. On the left, a navigation tree is visible with the following items: Home, System, Ethernet Configuration, Port Configuration (with sub-items Port 1 and Port 2), Monitor, Alarm, Syslog (highlighted with a red box, containing sub-items Syslog Setting and Syslog Message), Tools (with sub-items Ping and Reboot), and Management. The main content area is titled 'Syslog Message'. It features a 'Filter:' input field, an 'Apply' button, and a checked 'Scroll Down' checkbox. Below this, a list of Syslog messages is displayed, each starting with a timestamp and source: '<13>Jan 21 19:15:24 edgserver:'. The messages include serial data (Serial Rx Len) and connection status updates (CONN:Serial Port Opened, CONN:Serial Cleanup from TCP Session 1, CONN:Close Command From TCP Session 1, CONN:Serial Closed). The messages are truncated with hex codes at the end of each line.

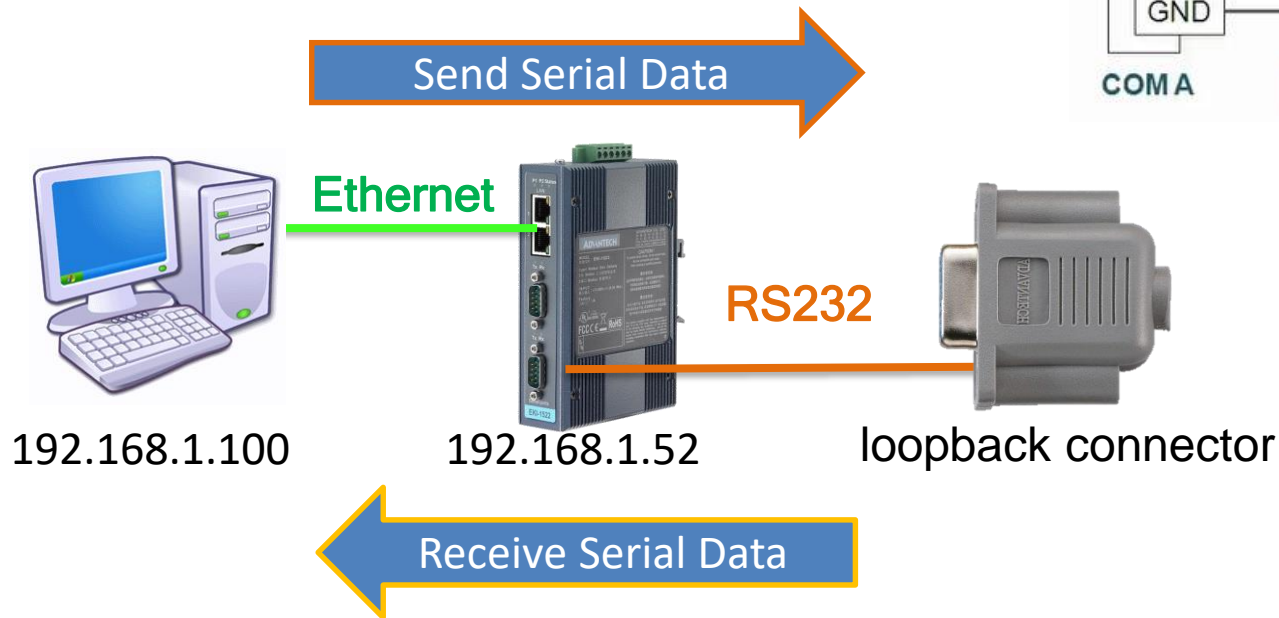


Use Google Chrome can drop wider

Example - Just-In-Time in Virtual COM Mode

-Loopback Test

Loopback (RS-232 only)



This is very important test way to check your device can work or not !!!

Connect the com port which you are mapping to and connected with **loopback connector** with the Device Server

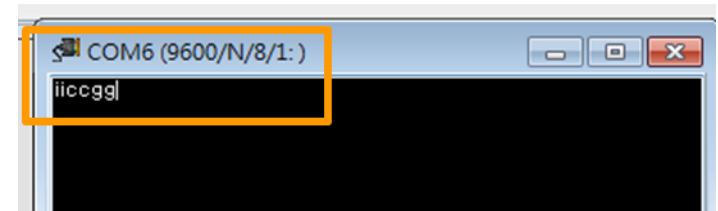
Loopback Test -- with RS-232 head

- Use **RS-232** Loopback test to verify the VCOM function

1. Connect the EKI com port with loopback connector which come with the Device Server



2. USE **Testview** Tool to simulate COM Port data transmission operation



3. USE **Access Port** Tool to monitor COM Port behavior

Request	Port	Result	Data (Hex)	1. Open Port
RP_MJ_CREATE	COM4	SUCCESS	Port Opened	2. Port Setting
OCTL_SERIAL_SET_QUEUE_SIZE	COM4	SUCCESS	InSize: 4096, OutSize: 4096	
OCTL_SERIAL_SET_BAUD_RATE	COM4	SUCCESS	Baud Rate: 9600	2. Port Setting
OCTL_SERIAL_SET_RTS	COM4	SUCCESS		
OCTL_SERIAL_SET_DTR	COM4	SUCCESS		2. Port Setting
OCTL_SERIAL_SET_LINE_CONTROL	COM4	SUCCESS	StopBits: 1, Parity: No, DataBits: 8	
OCTL_SERIAL_SET_CHARS	COM4	SUCCESS	EofChar: 0x0, ErrorChar: 0x0, BreakChar: 0x0, EventChar: 0x0, XonChar: 0x1	2. Port Setting
OCTL_SERIAL_SET_HANDFLOW	COM4	SUCCESS	ControlHandShake: 0x1, FlowReplace: 0x40, XonLimit: 1024, XoffLimit: 1024	
OCTL_SERIAL_SET_TIMEOUTS	COM4	SUCCESS	ReadIntervalTimeout: -1, ReadTotalTimeoutMultiplier: 0, ReadTotalTimeoutCount: 1	2. Port Setting
OCTL_SERIAL_SET_WAIT_MASK	COM4	SUCCESS	Mask: RXCHAR RXFLAG TXEMPTY CTS DSR RLSD BREAK ERR RING RX80FULL	
OCTL_SERIAL_WAIT_ON_MASK	COM4	SUCCESS		2. Port Setting
RP_MJ_WRITE	COM4	SUCCESS	Length: 1, Data: 31	
OCTL_SERIAL_WAIT_ON_MASK	COM4	SUCCESS		3. Data Transmission
RP_MJ_READ	COM4	SUCCESS	Length: 1, Data: 10	
OCTL_SERIAL_WAIT_ON_MASK	COM4	SUCCESS		3. Data Transmission
OCTL_SERIAL_WAIT_ON_MASK	COM4	CANCELLED		
OCTL_SERIAL_PURGE	COM4	SUCCESS	Purge: TXABORT RXABORT	4. Close the Port
RP_MJ_CLOSE	COM4	SUCCESS	Port Closed	

Open Port/Close Port

- Using two Tools to analyze COM Port Open/Close status.
- Access Port Tool is monitor COM Port tool that COM behavior from Host PC.
- JIT Message is show up COM Port behavior from DS point of view.
COM Port open would go after TCP connection success

Monitor COM Port with **Access Port Tool**

IRP_MJ_CREATE	COM8	SUCCESS	Port Opened
IOCTL_SERIAL_PURGE	COM8	SUCCESS	Purge: TXABORT RXABORT TXCLEAR RXCLEAR
IOCTL_SERIAL_SET_BAUD_RATE	COM8	SUCCESS	Baud Rate: 115200
IOCTL_SERIAL_SET_RTS	COM8	SUCCESS	
IOCTL_SERIAL_SET_DTR	COM8	SUCCESS	
IOCTL_SERIAL_SET_LINE_CONTROL	COM8	SUCCESS	StopBits: 1, Parity: No, DataBits: 8
IOCTL_SERIAL_SET_CHARS	COM8	SUCCESS	EofChar: 0x0, ErrorChar: 0x0, BreakChar: 0x0, EventChar: 0x0, XonChar: 0x11, XoffChar: 0x13
IOCTL_SERIAL_SET_HANDFLOW	COM8	SUCCESS	ControlHandShake: 0x1, FlowReplace: 0x80000040, XonLimit: 1024, XoffLimit: 1024
IOCTL_SERIAL_SET_TIMEOUTS	COM8	SUCCESS	ReadIntervalTimeout: 0, ReadTotalTimeoutMultiplier: 0, ReadTotalTimeoutConstant: 500, WriteTotalTimeoutMultiplier: 0, WriteTotalTimeoutConstant: 500
IOCTL_SERIAL_SET_WAIT_MASK	COM8	SUCCESS	Mask: BREAK ERR
IOCTL_SERIAL_WAIT_ON_MASK	COM8	CANCELLED	
IRP_MJ_READ	COM8	CANCELLED	Length: 0, Data:
IOCTL_SERIAL_PURGE	COM8	SUCCESS	Purge: TXABORT RXABORT TXCLEAR RXCLEAR
IRP_MJ_CLOSE	COM8	SUCCESS	Port Closed

Monitor COM Port with **JIT Diagnostic**

```
75 76 77 78 79 7a 31 32 33 34 35 7e 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78 79 7a 31 32 33 34 35 7e
<13>Jan 21 19:15:24 edgserver: Serial 1 DATA:Serial Cleanup from TCP Session 1
<13>Jan 21 19:15:24 edgserver: Serial 1 CONN:Close Command From TCP Session 1
<13>Jan 21 19:15:24 edgserver: Serial 1 CONN:Serial Closed
<13>Jan 21 19:15:34 edgserver: VCOM:Connection Accepted (IP: ::ffff:192.168.1.100)
<13>Jan 21 19:15:34 edgserver: Serial 1 CONN:Serial Port Opened
<13>Jan 21 19:15:34 edgserver: Serial 1 DATA:Serial Cleanup from TCP Session 1
<13>Jan 21 19:15:34 edgserver: Serial 1 CONN:Close Command From TCP Session 1
<13>Jan 21 19:15:34 edgserver: Serial 1 CONN:Serial Closed
```

Host-PC IP address

Data Level – PC Write Data

- Using two Tools to monitor COM Port Write/Read data from PC.
- “IRP_MJ_WRITE” shows data send from Application.
- “Serial 1 DATA: Serial Tx Len=” shows data send from Ethernet side

Monitor COM Port with **Access Port Tool**

TestView.exe	IOCTL_SERIAL_WAIT_ON_MASK	COM8	SUCCESS	
TestView.exe	IRP_MJ_WRITE	COM8	SUCCESS	Length: 1, Data: 31
TestView.exe	IOCTL_SERIAL_WAIT_ON_MASK	COM8	SUCCESS	
TestView.exe	IRP_MJ_READ	COM8	SUCCESS	Length: 1, Data: 31
TestView.exe	IOCTL_SERIAL_WAIT_ON_MASK	COM8	SUCCESS	
TestView.exe	IOCTL_SERIAL_WAIT_ON_MASK	COM8	SUCCESS	
TestView.exe	IRP_MJ_WRITE	COM8	SUCCESS	Length: 1, Data: 32
TestView.exe	IOCTL_SERIAL_WAIT_ON_MASK	COM8	SUCCESS	
TestView.exe	IOCTL_SERIAL_WAIT_ON_MASK	COM8	SUCCESS	
TestView.exe	IRP_MJ_READ	COM8	SUCCESS	Length: 1, Data: 32
TestView.exe	IOCTL_SERIAL_WAIT_ON_MASK	COM8	SUCCESS	
TestView.exe	IRP_MJ_WRITE	COM8	SUCCESS	Length: 1, Data: 33

Monitor COM Port with **JIT Diagnostic**

Accepted (IP: ::ffff:192.168.1.100)
Serial Port Opened

```
<13>Jan 21 19:55:15 edgserver: Serial 1 DATA:Serial Tx Len = 1: 31
<13>Jan 21 19:55:15 edgserver: Serial 1 DATA:Serial Rx Len = 1: 31
<13>Jan 21 19:55:15 edgserver: Serial 1 DATA:Serial Tx Len = 1: 32
<13>Jan 21 19:55:15 edgserver: Serial 1 DATA:Serial Rx Len = 1: 32
<13>Jan 21 19:55:16 edgserver: Serial 1 DATA:Serial Tx Len = 1: 33
<13>Jan 21 19:55:16 edgserver: Serial 1 DATA:Serial Rx Len = 1: 33
<13>Jan 21 19:55:16 edgserver: Serial 1 DATA:Serial Tx Len = 1: 34
<13>Jan 21 19:55:16 edgserver: Serial 1 DATA:Serial Rx Len = 1: 34
<13>Jan 21 19:55:16 edgserver: Serial 1 DATA:Serial Tx Len = 1: 36
<13>Jan 21 19:55:16 edgserver: Serial 1 DATA:Serial Rx Len = 1: 36
```

JIT Diag Level	<input checked="" type="checkbox"/> EXPT <input checked="" type="checkbox"/> CONN <input checked="" type="checkbox"/> DATA <input type="checkbox"/> PROTO <input type="checkbox"/> TCPWR
Save	

Data Level – PC Read Data

- Using two Tools to monitor COM Port Write/Read data from PC.
- “IRP_MJ_READ” shows received data of Application.
- “Serial 1 DATA: Serial Rx Len=” shows received data from physical port

Monitor COM Port with **Access Port Tool**

TestView.exe	IOCTL_SERIAL_WAIT_ON_MASK	COM8	SUCCESS	
TestView.exe	IRP_MJ_WRITE	COM8	SUCCESS	Length: 1, Data: 31
TestView.exe	IOCTL_SERIAL_WAIT_ON_MASK	COM8	SUCCESS	
TestView.exe	IRP_MJ_READ	COM8	SUCCESS	Length: 1, Data: 31
TestView.exe	IOCTL_SERIAL_WAIT_ON_MASK	COM8	SUCCESS	
TestView.exe	IOCTL_SERIAL_WAIT_ON_MASK	COM8	SUCCESS	
TestView.exe	IRP_MJ_WRITE	COM8	SUCCESS	Length: 1, Data: 32
TestView.exe	IOCTL_SERIAL_WAIT_ON_MASK	COM8	SUCCESS	
TestView.exe	IOCTL_SERIAL_WAIT_ON_MASK	COM8	SUCCESS	
TestView.exe	IRP_MJ_READ	COM8	SUCCESS	Length: 1, Data: 32
TestView.exe	IOCTL_SERIAL_WAIT_ON_MASK	COM8	SUCCESS	
TestView.exe	IRP_MJ_WRITE	COM8	SUCCESS	Length: 1, Data: 33

Monitor COM Port with **JIT Diagnostic**

```
Accepted (IP: ::ffff:192.168.1.100)
Serial Port Opened
<13>Jan 21 19:55:15 edgserver: Serial 1 DATA:Serial Tx Len = 1: 31
<13>Jan 21 19:55:15 edgserver: Serial 1 DATA:Serial Rx Len = 1: 31
<13>Jan 21 19:55:15 edgserver: Serial 1 DATA:Serial Tx Len = 1: 32
<13>Jan 21 19:55:15 edgserver: Serial 1 DATA:Serial Rx Len = 1: 32
<13>Jan 21 19:55:16 edgserver: Serial 1 DATA:Serial Tx Len = 1: 33
<13>Jan 21 19:55:16 edgserver: Serial 1 DATA:Serial Rx Len = 1: 33
<13>Jan 21 19:55:16 edgserver: Serial 1 DATA:Serial Tx Len = 1: 34
<13>Jan 21 19:55:16 edgserver: Serial 1 DATA:Serial Rx Len = 1: 34
<13>Jan 21 19:55:16 edgserver: Serial 1 DATA:Serial Tx Len = 1: 36
<13>Jan 21 19:55:16 edgserver: Serial 1 DATA:Serial Rx Len = 1: 36
```

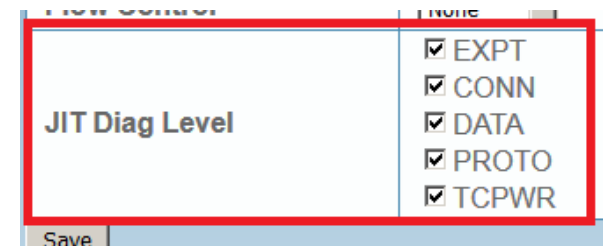
JIT Diag Level	<input checked="" type="checkbox"/> EXPT
	<input checked="" type="checkbox"/> CONN
	<input checked="" type="checkbox"/> DATA
	<input type="checkbox"/> PROTO
	<input type="checkbox"/> TCPWR
Save	

Protocol Level – PC Write/Read Data

- PROTO Level : Shows Tx data sent out to physical port <DS view>
“PROTO: WriteComplete Returned”
- PROTO Level : Shows Rx data received from physical port <DS view>
“PROTO: SendSerialData to TCP Session x success”
* TCP Session x: index of TCP Session that connected to DS

Monitor COM Port with **JIT Diagnostic**

```
<13>Jan 21 20:26:25 last message repeated 1 time(s)
<13>Jan 21 20:26:25 edgserver: Serial 1 DATA:Serial Tx Len = 1: 31
<13>Jan 21 20:26:25 edgserver: Serial 1 PROTO:WriteComplete Returned
<13>Jan 21 20:26:25 edgserver: Serial 1 DATA:Serial Rx Len = 1: 31
<13>Jan 21 20:26:25 edgserver: Serial 1 PROTO:SendSerialData to TCP Session 1 success
<13>Jan 21 20:26:25 edgserver: Serial 1 TCPWR:TCP Write Command from TCP Session1(Return Success)
<13>Jan 21 20:26:25 edgserver: Serial 1 DATA:Serial Tx Len = 1: 32
<13>Jan 21 20:26:25 edgserver: Serial 1 PROTO:WriteComplete Returned
<13>Jan 21 20:26:25 edgserver: Serial 1 DATA:Serial Rx Len = 1: 32
<13>Jan 21 20:26:25 edgserver: Serial 1 PROTO:SendSerialData to TCP Session 1 success
<13>Jan 21 20:26:25 edgserver: Serial 1 TCPWR:TCP Write Command from TCP Session1(Return Success)
<13>Jan 21 20:26:25 edgserver: Serial 1 DATA:Serial Tx Len = 1: 33
<13>Jan 21 20:26:25 edgserver: Serial 1 PROTO:WriteComplete Returned
<13>Jan 21 20:26:25 edgserver: Serial 1 DATA:Serial Rx Len = 1: 33
<13>Jan 21 20:26:25 edgserver: Serial 1 PROTO:SendSerialData to TCP Session 1 success
```



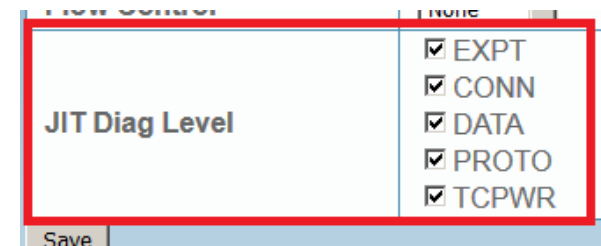
TCPWR Level – PC Write/Read Data

- TCPWR Level : Shows data sent from Ethernet Socket <DS view>
“TCP Write Command from TCP Session%d”

* TCP Session x: index of TCP Session that connected to DS

Monitor COM Port with **JIT Diagnostic**

```
<13>Jan 21 20:26:25 last message repeated 1 time(s)
<13>Jan 21 20:26:25 edgserver: Serial 1 DATA:Serial Tx Len = 1: 31
<13>Jan 21 20:26:25 edgserver: Serial 1 PROTO:WriteComplete Returned
<13>Jan 21 20:26:25 edgserver: Serial 1 DATA:Serial Rx Len = 1: 31
<13>Jan 21 20:26:25 edgserver: Serial 1 PROTO:SendSerialData to TCP Session 1 success
<13>Jan 21 20:26:25 edgserver: Serial 1 TCPWR:TCP Write Command from TCP Session1(Return Success)
<13>Jan 21 20:26:25 edgserver: Serial 1 DATA:Serial Tx Len = 1: 32
<13>Jan 21 20:26:25 edgserver: Serial 1 PROTO:WriteComplete Returned
<13>Jan 21 20:26:25 edgserver: Serial 1 DATA:Serial Rx Len = 1: 32
<13>Jan 21 20:26:25 edgserver: Serial 1 PROTO:SendSerialData to TCP Session 1 success
<13>Jan 21 20:26:25 edgserver: Serial 1 TCPWR:TCP Write Command from TCP Session1(Return Success)
<13>Jan 21 20:26:25 edgserver: Serial 1 DATA:Serial Tx Len = 1: 33
<13>Jan 21 20:26:25 edgserver: Serial 1 PROTO:WriteComplete Returned
<13>Jan 21 20:26:25 edgserver: Serial 1 DATA:Serial Rx Len = 1: 33
<13>Jan 21 20:26:25 edgserver: Serial 1 PROTO:SendSerialData to TCP Session 1 success
```



Example - Trouble shooting via Just-In-Time in USDG Data Mode

Disconnected by Application

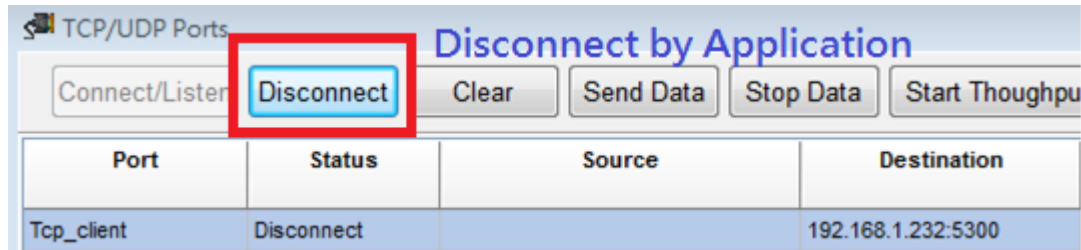
➤ Shows TCP connection: Application actively disconnected TCP Session

Serial x CONN: client x closed TCP connection

Serial x : serial port number ; client x: which initiate client connected

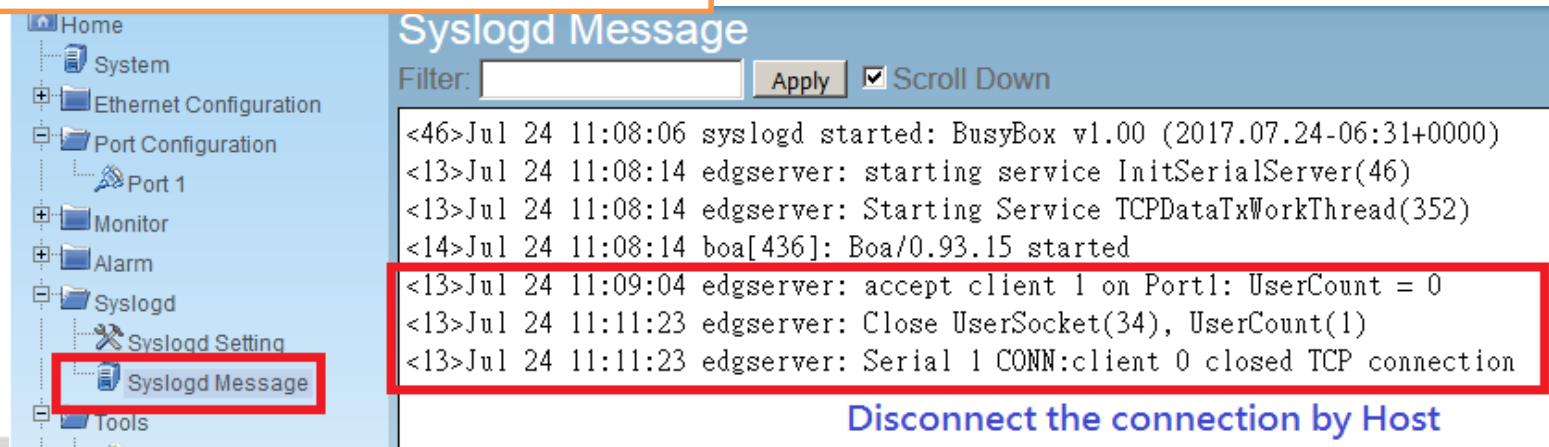
For example: Serial 1 CONN: client 0 closed TCP connection

Application:



Port	Status	Source	Destination
Tcp_client	Disconnect		192.168.1.232:5300

Monitor TCP Session with JIT Diagnostic



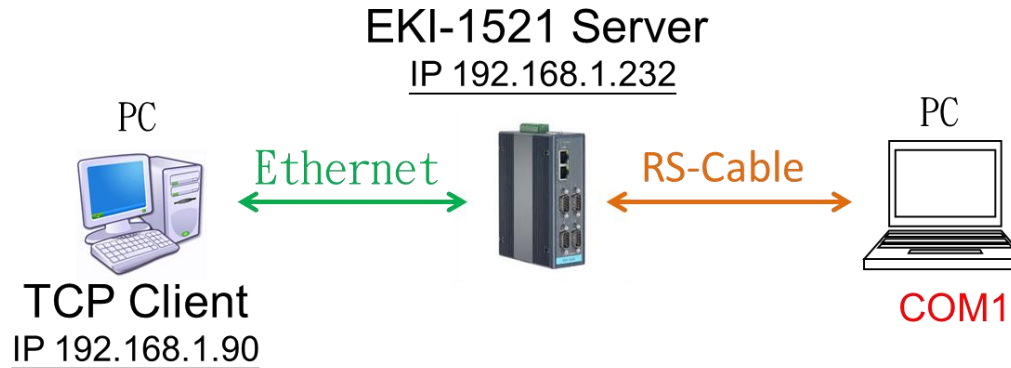
Syslog Message

Filter: ☒ Scroll Down

```
<46>Jul 24 11:08:06 syslogd started: BusyBox v1.00 (2017.07.24-06:31+0000)
<13>Jul 24 11:08:14 edgserver: starting service InitSerialServer(46)
<13>Jul 24 11:08:14 edgserver: Starting Service TCPDataTxWorkThread(352)
<14>Jul 24 11:08:14 boa[436]: Boa/0.93.15 started
<13>Jul 24 11:09:04 edgserver: accept client 1 on Port1: UserCount = 0
<13>Jul 24 11:11:23 edgserver: Close UserSocket(34), UserCount(1)
<13>Jul 24 11:11:23 edgserver: Serial 1 CONN:client 0 closed TCP connection
```

Disconnect the connection by Host

Disconnected Due to Data Idle Timeout



- **EKI-1521-BE WEBGUI Setting Data Idle Timeout: 60 secs**

Port 1 configuration	
Basic	Operation
Mode	USDG Data Mode
Protocol	TCP
Data Idle Timeout(s)	60
Data Listen Port	5300
Command Listen Port	5400
Response Timeout(ms)	0
Frame Break(ms)	0
TCP Mode Extra Options	
Auto Connect To Peer IP	<input type="checkbox"/>

Disconnected via Data Idle Timeout

- When idle over 60 seconds without sending data, EKI would actively close the TCP connection.

Serial x CONN: host Idle time out

Serial x : serial port number

For example: Serial 1 CONN: host Idle time out

Application:

Connect/Listen

Disconnect

Clear

Send Data

Stop Data

Start Thoughtput

Stop Thoughtput

Terminal

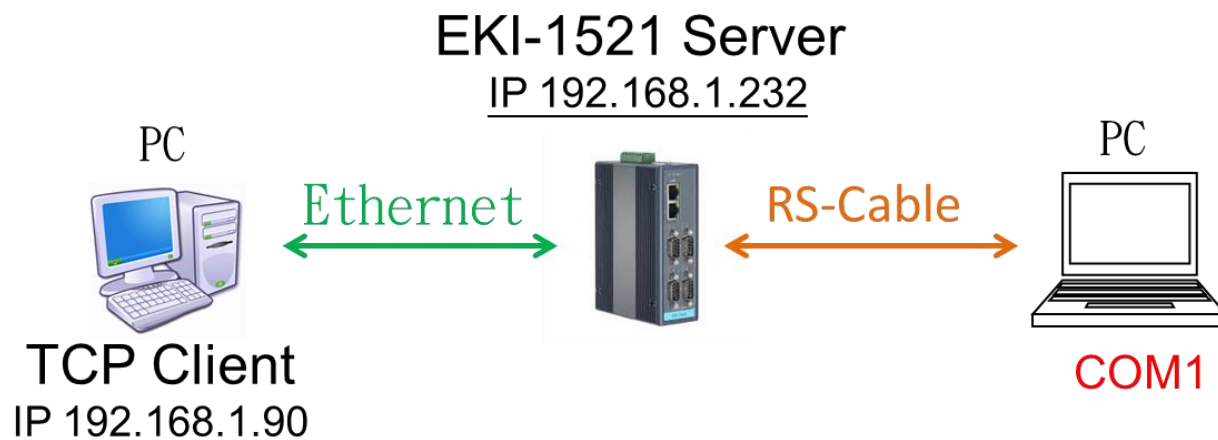
Port	Status	Source	Destination	Send Bytes	Receive Bytes	Transmit throughput	Receive through	Running Time
Tcp_client	Disconnect		192.168.1.232:5300	0	3	0	0	00:01:03

Monitor TCP Session with JIT Diagnostic

```
.00 (2017.07.24-06:31+0000)
<15>Jul 24 11:02:12 edgserver: starting service InitSerialServer(46)
<13>Jul 24 11:02:12 edgserver: Starting Service TCPDataTxWorkThread(352)
<14>Jul 24 11:02:13 boa[435]: Boa/0.93.15 started
<83>Jul 24 11:02:19 boa[435]: Error opening /www/favicon.ico for : No such file or
directory
<13>Jul 24 11:02:54 edgserver: accept client 1 on Port1: UserCount = 0
<13>Jul 24 11:03:11 edgserver: Close UserSocket(34), UserCount(1)
<13>Jul 24 11:03:11 edgserver: Serial 1 CONN: client 0 closed TCP connection
<13>Jul 24 11:03:16 edgserver: accept client 1 on Port1: UserCount = 0
<13>Jul 24 11:03:19 edgserver: Serial 1 DATA:Rx 1 bytes of data
<13>Jul 24 11:04:19 last message repeated 2 time(s)
<13>Jul 24 11:04:19 edgserver: Serial 1 CONN: host Idle time out
<13>Jul 24 11:04:19 edgserver: Close UserSocket(34), UserCount(1)
```

Data Idle timeout default =60secs

USDG Data Server mode Network Topology



TCP/UDP Ports

Connect/Listen Disconnect Clear Send Data Stop Data Start Throughput Stop Throughput Terminal

Port	Status	Source	Destination	Send Bytes	Receive Bytes	Transmit throughput	Receive through	Running Time
Tcp_client	Connect	192.168.1.90:41947	192.168.1.232:5300	3	0	0	0	00:00:24

Host TCP Client Send out TCP Data to EKI-1521 <ASCII "123">

Com Ports

Connect Disconnect Setup Clear Send Data Stop Data Start Throughput Stop Throughput Terminal

Port	Status	Option	RTS	DTR	CTS	DSR	DCD	RI	Send Bytes	Receive Bytes	Parity Error	Overrun Error	Fram Error	Transmit throughput	Rec throu
COM1	Connect	9600/N/8/1: Flow DTRRTS	■	■	■	■	■	●	0	3	0	0	0	0	0

TCP-Client 192.168.1.232:5300 (Connect...)

123

Observe TCP Data Transmission

Serial x TCPWR: P(x)Lx Write xx ...

TCPWR: TCP Session description

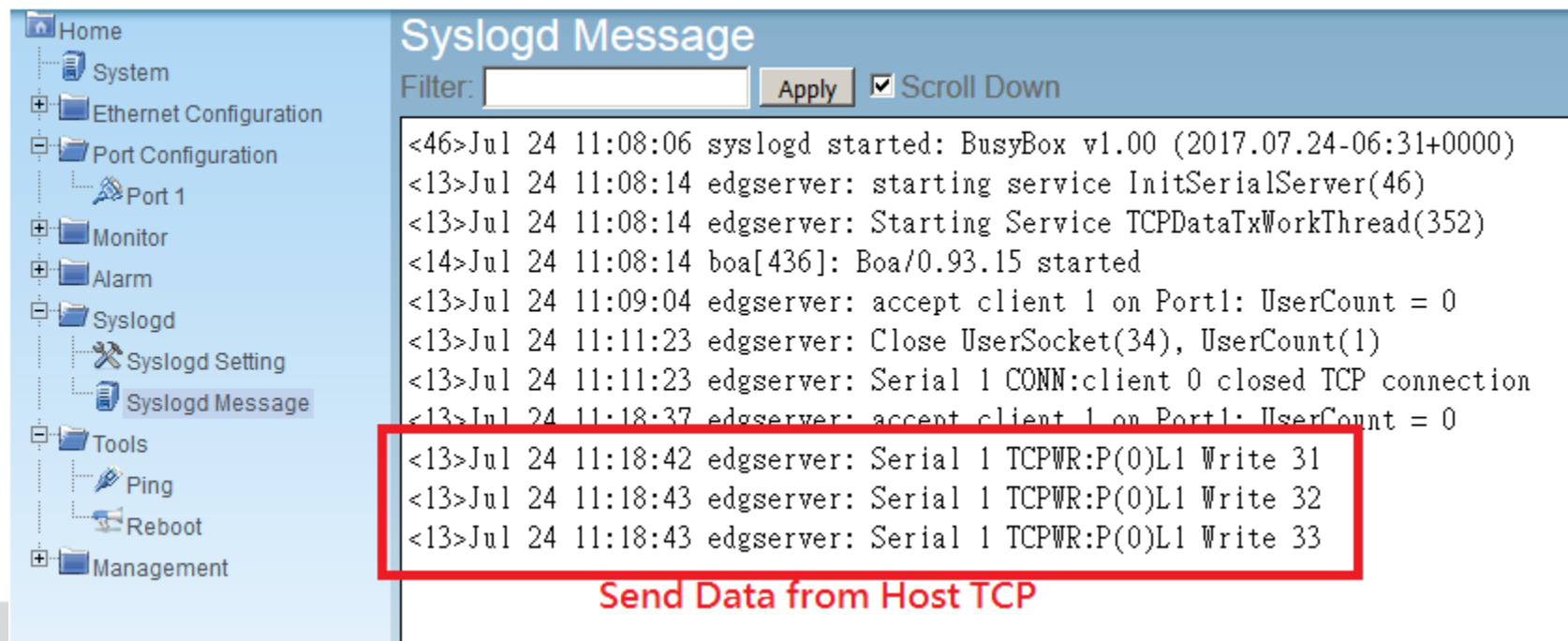
P(x): TCP Session number <start from 0; max support 16 connections>

Lx: Length

Write data content with Hex format

For example: Serial 1 TCPWR: P(0)L1 Write 31

Host send TCP data from first TCP session and data Length 1 the data content "ASCII 1 : Hex 31" to EKI-1521-BE with serial Port 1.



The screenshot shows a web-based configuration interface for a device. On the left is a navigation tree with categories like Home, System, Ethernet Configuration, Port Configuration, Monitor, Alarm, Syslogd, Tools, and Management. Under Syslogd, 'Syslogd Message' is selected. The main area is titled 'Syslogd Message' and contains a list of log entries. A red rectangle highlights three specific log entries:

```
<13>Jul 24 11:18:42 edgserver: Serial 1 TCPWR:P(0)L1 Write 31
<13>Jul 24 11:18:43 edgserver: Serial 1 TCPWR:P(0)L1 Write 32
<13>Jul 24 11:18:43 edgserver: Serial 1 TCPWR:P(0)L1 Write 33
```

Below the highlighted logs, the text **Send Data from Host TCP** is written in red.

Reference Information

EXCEPT	SendSerialData Failed on TCP Session %d(RSPTO)	Client ID
	SendSerialData Failed Socket(%d)	Client ID
	TCP Session %d Write Data Too Long	Client ID
	Serial writeSize = %d	WrittenSize
	Send Write Complete Failed...	
	Serial cannot write	
	CheckWaitOnMaskEvent Failed	
	TCP connetcion %d Terminated	Client ID
	port is busy now, TCP user count exceeded	
	port is busy now, multi-connection DISABLE	
PROTO	SendSerialData to TCP Session %d %s(RSPTO)",	Client ID, success:failed
	"SendSerialData to TCP Session %d %s"	Client ID, success:failed
DATA	Serial Rx Len = %d: %s	Datalen, SerialData(HEX)
CONNECT	Serial Closed	
	Serial Port Opened	
	Close Command From TCP Session %d	Client ID
Write	TCP Write Command from TCP Session%d(Device Busy), Invalid behavior!!	Client ID
	TCP Write Command from TCP Session%d(Return Success)	Client ID
	TCP Write Command from TCP Session%d(Queue Full)	Client ID
DATA	Seral Cleanup from TCP Session %d	Client ID