ACKSYS DTUS0434 Servercom Firmware For Ethernet and Wi-Fi Port Servers User Guide

Contents

ACKSYS DTUS0434 Servercom Firmware For Ethernet and Wi-Fi Port Servers

USER GUIDE

FOR ETHERNET AND WI-FI PORT SERVERS

COPYRIGHT (©) ACKSYS 2009

REGISTERED TRADEMARKS ®

NOTICE

I. INTRODUCTION

The SERVERCOM software allows any port server made by ACKSYS to be used as a distant serial communication port for a TCP/IP compliant computer. SERVERCOM may operate in three distinct modes, depending on needs of the remote application software :

- In RFC2217-compliant mode, SERVERCOM allows the remote application software to receive and send data, monitor input control signals, set output control signals, change the data format and baud rate, through the remote computer native serial port interface.This mode is relevant when the remote computer has a RFC2217-compliant client driver which emulates a serial port for the application software. It is especially useful when the application software must be made to use a remote port, but it cannot be changed to support TCP/IP communication (i.e. when the application source code is not available).

  For more information about RFC2217, see : http://www.ietf.org/rfc/rfc2217.txt

- In RAW mode, SERVERCOM has a much simpler TCP/IP interface which only allows to receive and send data. All the serial communication parameters can be set up locally in the port server through the administration interface.This mode is relevant in all the other cases:
  - when the remote computer is another port server in TCP-CLIENT mode;
  - when the remote computer has no RFC2217-compliant client driver;
  - when the remote application software can be written from scratch to use the TCP/IP SOCKET interface;
- In TELNET mode, SERVERCOM allows a standard TELNET client to to receive and send data. This mode is similar to RFC2217, but it does not handle any COM-related operation, only data exchange.

This mode is relevant for testing purposes, and when the device connected to the port server needs only a serial

console interface with the user.

In all modes you can use the port server administration interface to set the serial control signals to behave locally; this allows to have a much faster response time for flow control between the port server and the serial device.

The SERVERCOM firmware functions as a **network server**. This means that it provides serial port service to the network: when the SERVERCOM firmware is in use, the port server just sits there waiting for some remote network device (usually a computer or another port server in network client mode) to call in and use its serial port. In this mode the port server will never attempt on its own to connect to a remote network device.

II. WHEN TO USE THE SERVERCOM FIRMWARE ?

In order to identify the cases where SERVERCOM can be used, it is important to know that the SERVERCOM firmware has the following properties:
• It uses TCP communications on the network side, forbidding undetected data loss at the expense of slower communications.
• It conveys no protocol information in the data exchanged between the remote application software and the device connected to the port server 1.
• It can handle serial communications up to 230400 bauds.
• It can drive and monitor serial control signals, locally or remotely.

The SERVERCOM firmware can be used to solve the following needs:

• Application software using full remote COM port emulation.
• Application software using partial (data only) remote COM port emulation.
• Application software using a TCP SOCKET to exchange data with a serial comm server.
• Tunelling two-ways data between SERVERCOM and TCP-CLIENT.
• Tunnelling MODBUS frames (or other asynchronous protocols) in pointto- point configurations.
• TELNET client used as a serial console to the device connected to the port server.

III. USING SERVERCOM IN RFC2217 MODE

III.1 Configuration

The network configuration, including IP address, netmask, gateway (router) address, DHCP, and so on, is described in the port server user manual.

The SERVERCOM firmware comes with defaults settings for the RFC2217 mode. These settings can be reinstated with the “**set default**” command. Important settings are:

- set serial mode: by default this is set to “set serial mode rcf2217”
- set sendtrigger: by default the SERVERCOM firmware sends incoming serial data onto the network after waiting at most 2 milliseconds. Often you will want to change this. See the detailed documentation of this command.
- flow control: by default the SERVERCOM firmware uses no local flow control. Often you will want to change this. See the detailed documentation of the “set serial” commands.
- set keepalive…: allows the SERVERCOM firmware to detect when the client crashes and thus allows later reconnection from the same or another client.
- set reconnect…: allows the (same) network client to establish a new connection, forcing the SERVERCOM

firmware to close the previous one. This allows faster recovery of client failure than keepalives.

- Let say that you connect the port server to a device that sends frames of 3 to 100 chars at 1200 bauds, followed by a silence of at least 3 char times. The default sendtrigger will work, but it is not suited for this kind of data since each received char would be sent on the Ethernet in its own frame, loosing a lot of network bandwidth. A better sendtrigger in this case is:

  set sendtrigger idledelay 3c

  Do not forget also in this case:

  set serial baudrate 1200

- Let say that you connect the port server to a device honors XON/XOFF protocol. Then you can set it in the port server:

  set serial xonxoff use

- Let say that you connect the port server to a device honors RTS/CTS protocol. Then you can set it in the port server:

  set serial rts flow set serial cts flow

  Note that this is not required if the remote (network client) computer has a fully compliant RFC2217 driver, since the remote computer can use the RFC2217 protocol to set the flow control remotely.

- Let say that you access the port server from a remote computer that is on the other side of a firewall. Say that this firewall forbids the use of TCP port 2300 but allows TCP port 4000. Then you can set it in the port server:

set serial port 4000

(this is considered a “serial” parameter since it would be different for each serial port, had the port server had more than one serial port).

III.2 Using VIP

VIP is a RFC2217 compliant COM port emulator that allows Windows applications written to use native PC COM ports, to access the port server serial port transparently.
To use it, or other RFC2217 compliant port redirector, the “mode” parameter of the “set serial mode mode” command must be set to “rfc2217“.
The VIP software, more information, and a link to get the latest version are available on the CD-ROM.

VIP Windows COM port redirector software installation
Before installing the redirector software, read the release notes.
Run the executable file on the provided disk. This installs the VIP software, allowing COM port redirection from MSWindows to the port server.
Run the VIP config program from the desktop icon or the start menu.
If needed, stop the VIP service in the “Setup” tab, then click the “scan for devices” button. Fill in the IP range to scan, click “scan” to find the available ACKSYS port servers. Choose one and click “add”.
Note: if your port server does not appear in the scan list, your network may be improperly set or overloaded. You can still close the scanner, select the “virtual port” tab and use the “New” button to manually add a virtual port.
You may enter a custom description for the chosen port server. Then select a COM port name. Other options

should be left in their default state.
When you have set up all the virtual ports you need, restart the service with the “setup” tab.
You are now ready to use the port server through port redirection. Just run your application and specify the COM port name that you selected in the previous step.

If you need to write program from scratch, the usual Win32 COMM API can be used. Please refer to the Win32 documentation (included in your development environment) for more details.

III.3 Using the SOCKET interface

The application software can use the SOCKET interface to communicate with a port server put in RFC2217 mode. This involves the capacity to handle the TELNET protocol (transparency and option negotiation) as well as the RFC2217 specific features. Since this is not an easy task, using the SOCKET interface is not recommended in RFC2217 mode.

III.4 Troubleshooting

Before trying to troubleshoot the SERVERCOM firmware in RFC2217 mode, you should insure that the port server is normally visible on the network. Please first refer to the relevant troubleshooting section in the port server user’s manual. In the following instructions it is assumed that you are able to connect to the administration system from the same computer from which you access the port server..

In the “VIP config” setup tab, you can enable a trace log that will appear in the window below. The trace log can also be saved in a file if you need (the file is located in the VIP programs directory). The trace will stay on through reboots. Be warned, this trace slows down the VIP service.

In the “VIP config” virtual port tab, you should see the COM port name that you assigned in the installation. When the port is in use, warning lights are displayed on the left of the name. You can check this by opening the port with Hyperterminal.

If the warning lights do not show up, the address or port given for the port is bad. Also, there may be a problem with the computer’s network parameters : in this case, you cannot PING the port server either.

Enable the trace log. Each time the virtual port is opened by the application, you should see a bunch of messages beginning with these three: “Connecting to…” then “Connection to … successful” then “Purge buffers”. If only the two first connection messages appear, the port server is in RAW mode. You can change this with the « set serial mode » command in the port server administration system. Check that the protocol is set accordingly in the “VIP config” virtual port parameters.
Enter the port server administration mode, check the IP address and network port with the following commands:
show net ip
show serial port
show serial mode

The mode should be “rfc2217”. The display the VIP virtual port parameters window on the remote computer. Check that the “IP address of server” and “port number” are the same as set in the port server. Check that the “protocol” is set to “Telnet”.

If the mode is “rfc2217”, then DTR and RTS should be set to “driven” or “flow”, incoming signals should be set to “ignore” or “flow”, the sendtrigger parameter must be tailored to your needs (the factory default is a good starting point), other serial parameters are irrelevant since they are reset by VIP.

IV. USING SERVERCOM IN RAW MODE

IV.1 Use cases

The “RAW” mode means that the SERVERCOM firmware makes no interpretation of any kind on the data flow in

either direction.

You will use the port server in “RAW” mode when either:

- You cannot use a COM port redirector (because none is available on your operating system).
- You do not need the COM port redirector facilities because your application does not need information about control signals, data errors and so on.
- You do not need the COM port redirector facilities because your application is already written and uses a SOCKET interface.In “RAW” mode, the asynchronous serial port of the port server must be fully set up locally, since the client application has no way to advertise the intended use of the character format, baud rate, control signals, etc. You must set all this through the administration commands.

  The baud rates are supported by the port server as follows:

- The ‘set serial baudrate’ command has a limit of 429,000 bauds.
- Any baud rate between 229 bauds and 429,000 bauds can be approximated with an baud skew less than 2.3%
- The formula which gives the relative baud skew given the baud rate wantedbaud is:div = E[C/ *wantedbaud* + 0.5]

  realbaud = C/div

  relative baud skew = (*wantedbaud* – *realbaud*) / *wantedbaud*

  with

  C = 15,000,000 (15 MHz)

  E[] = integral part function (round-down function)

- The fastest achievable transfer rate without character lost is 429,000 bauds when using character format 8×1 (x = e, o, m, s but not n). This speed cannot be sustained for long periods of time.

IV.2 Configuration

In many respects the configuration in raw mode is similar to the configuration in rfc2217 mode. Please refer to the RFC2217 mode configuration.

Howeve the important setting in RAW mode is:
• set serial mode raw: must be set, since the default mode is rcf2217.

IV.3 Using VIP

In many respects the configuration in raw mode is similar to the configuration in rfc2217 mode. Please refer to the RFC2217 mode configuration.

Howeve the important setting in RAW mode is:

- set serial mode raw: must be set, since the default mode is rcf2217.

IV.4 Using a redirector for Linux

Third-party open-source redirector software is available for Linux but is not supported by ACKSYS. Search the web for “sredir” or go to
http://packages.debian.org/unstable/source/sredird.

IV.5 Troubleshooting

Troubleshooting with VIP is explained in the RFC2217 mode section. No particular problem is expected in this mode when programing via the SOCKET interface. Should any communication problem arise, the first step of debugging should be:

Try to do the same thing with a standard TELNET client.

IV.6 SOCKET interface example for Linux

The application software can use the SOCKET interface to communicate with a port server put in RAW mode.

To be written

IV.7 SOCKET interface example for Windows

The application software can use the SOCKET interface to communicate with a port server put in RAW mode. Following is a Visual C++ sample program that receives and resends data to a port server configured in raw mode

```
/*     This program connects to the port server at IP 192.168.1.253, port 2300.
       Then any character received on the serial port of the port server is
       transfered to this program, which echoes it back to the port server.
       The port server then sends it back to the serial port.

       The program thus implement a remote character echoing of the data provided
       on the port server serial port.

       In order to use this program:
       1) compile it as C++ code with MSVC 4 or greater (tested with MSVC 6.0)
              sample command prompt compilation :
              C:> CL MAIN.CPP /MT
              (creates main.exe. /MT adds required multithreading support.)
       2) connect a PC COM port to
              either a COMETH 232 with a straight RS232 cable
              or a COMETH FIELD with a crossover RS232 cable
              or a WiFi port server with a crossover RS232 cable
       3) configure the port server with the default values (IP, port, baudrate…)
              and set the following option : "set serial mode raw"
       4) connect to the COM local port with Hyperterminal or equivalent software
              and setup communication in the same way that you configured the server
       5) run this program

       Now anything you type in Hyperterminal is echoed back. Also, this program
       displays a 'w' each time it receives and echoes a frame. */

#include <afxsock.h>            // MFC socket extensions
#include <conio.h>

void initsock(void) {     // initializations
       WSADATA Wsadata;
       int rc;
       AfxWinInit(GetModuleHandle(NULL),NULL,"",0);
       if((rc=WSAStartup(0x202,&Wsadata))) {
              fprintf(stderr,"Cannot init WSAStartup, %d\n", rc);
              exit(1);
       }
       if(!AfxSocketInit()) {
              fprintf(stderr,"Cannot init AfxSocket, %d\n", GetLastError());
              exit(1);
       }
}

void main(int argc,char**argv) {
       initsock();
       CSocket *sock = new CSocket;
       if(!sock->Create()) {
              fprintf(stderr,"Err create %d\n",sock->GetLastError());
              exit(1);
       }
       if(!sock->Connect("192.168.1.253",2300)) {
              fprintf(stderr,"Err connect %d\n",sock->GetLastError());
              exit(1);
       }
       BOOL nodelay = TRUE;
       if(!sock->SetSockOpt(TCP_NODELAY, &nodelay, sizeof(nodelay), IPPROTO_TCP)) {
              fprintf(stderr,"Err nodelay %d\n",sock->GetLastError());
              exit(1);
       }
       CSocketFile *ssend = new CSocketFile(sock);
       setbuf(stdout,NULL);
       for(;;) {
              unsigned int nc;
              char sendbuf[80];
              nc = ssend->Read(sendbuf, 80);
              ssend->Write(sendbuf,nc);
              ssend->Flush();// send all now
              putchar('w');
       }
}
```

Command			Default value	Notes	Description
login	*username*				start the administrator identification sequence. Ask password.
set	**default**				restore factory defaults, except the MAC address, the save count, the current firmware and the next firmware to run.
save					save the current configuration to the permanent configuration memory which is used after reboot and remains when the device is powered off.
reset					close the administration session and reboot the device, to ignore parameters changed but not saved, or to reload saved parameters. The following parameters do not need a reset to take effect: location, showperm, netconfigperm, serial interface.
show	**version**				display firmware name and version
quit					close administration session (TELNET only).
set **show**	**login** **login**	*username*	root		change/display administrator username. 8 bytes max. Upper and lower cases.
set **show**	**password** **password**	*password*	root		change/display administrator password. 8 bytes max. Upper and lower cases.
set **show**	**location** **location**	*location*	“Unknown location”		change/display *location* description of the device server. 30 bytes max. Upper and lower cases.
set **show**	**showperm** **showperm**	*perm*	allow		change/display the right to display configuration information without entering the administrator password. *perm* : one of **allow** / **deny**
set **show**	**netconfigperm** **netconfigperm**	*perm*	allow		change/display the right to use the administration system from the network. perm : one of **allow** / **deny**
set **show**	**upgradeperm** **upgradeperm**	*perm*	allow		change/display the right to upgrade the firmware. *perm* : one of **allow** / **deny** If this flag is set to “**allow**”, upgrade is allowed. (through serial port or Wifi interface) else upgrade is not allowed. ***These commands are not available for devices providing several firmwares simultaneously.***
set **show**	**net login** **net login**	*username*	unused (empty)	note 9	change/display exploitation username. (see detailed documentation)
set	**net password**	*password*	empty	note 9	change exploitation password. (see detailed documentation)

set **show**	**prog enable** **prog enable**	*seg*	SERVERCOM firmware located in seg /2	execute after next reset the current firmware located in segment *seg*. Display this firmware.
show	**prog list**			display information about all firmwares.
show	**prog info**	*seg*		display information about firmware located in segment *seg*, in computer readable format.
show	**prog data**	*seg*		display information about firmware located in segment *seg*, in computer readable format.

Command			Default value	Notes	Description
show	**net ethernet**		Factory defined		display Ethernet address. 6 hex digits separated by columns.
set **show**	**net dhcp** **net dhcp**	*state*	off		turn **on** /**off** or display the DHCP client use. When dhcp is on, the manually specified IP address is not used.
set	**net dhcp clientid**	*ident*	empty (MAC address sent as string)		when sending DHCP option 61 (Client ID), replace the default Client ID (MAC address as a string) by the custom string *ident*. 15 bytes max, upper and lower cases allowed.
set	**net dhcp clientid**				delete the custom client ID and use the default client ID.
show	**net dhcp clientid**				display custom client ID
set **show**	**net dhcp hname** **net dhcp hname**	*hostname*	empty (not sent)		provide the DHCP server with the supplementary Host Name option, with value *hostname*. 19 bytes max, no spaces allowed, upper and lower cases allowed. Value assigned to DHCP option 12, if any.
set **show**	**net ip** **net ip**	*aaa.bbb.ccc.ddd*	192.168.1.253		change/display IP address in dotted decimal notation.
set **show**	**net mask** **net mask**	*aaa.bbb.ccc.ddd*	255.255.255.0		change/display local subnet mask
set **show**	**net gateway** **net gateway**	*aaa.bbb.ccc.ddd*	0.0.0.0		change/display the gateway IP address.
set **show**	**net metric** **net metric**	*mmm*	64 (“WLg”) 10 (others)		change/display the number of gateway hops. mmm is **1** to **255**
set **show**	**net reconnect** **net reconnect**	*state*	On (“WLg”) Off (others)		Turn **on** or **off** / display the “forced reconnection” feature.
set	**net keepalive**	*n t1 t2*	3/3/1 (“WLg”) Off (others)		*n* defines the number of probes to send before closing the connection. *t1* defines the time in seconds before sending the first probe since the connection is inactive (the “activation delay” mentioned earlier). *t2* defines the time in seconds between each probes (the “interval delay mentioned earlier). *n* ranges from **1** to **255**. *t1* and *t2* range from **1** to **65535**. This command also resets the “**segtmo**” parameter to (t1 + t2 x n).
show	**net keepalive**				display keepalive parameters as “n probes, t1/t2 sec”; else “keepalive off”.
set	**net keepalive**	**0 0 0**			disables use of the keep-alive feature.
set	**net segtmo**	*delay*	4 (“WLg”) Off (0) (others)		*delay* defines the number of seconds the firmware will wait for acknowledgement of sent data, after which it will consider that the network has failed and will abort the TCP connection. *delay* ranges from **0** (off) to **65535**. Setting “**keepalive**” changes “**segtmo**”.
show	**net segtmo**				display segtmo parameter value.
show	**net config port**		23		administration port

Command			Default value	Notes	Description
set **show**	**net ssid** **net ssid**	*ssid*	acksys		change/display the SSID of the device. SSID is a case sensitive characters string.
set **show**	**net mode** **net mode**	*mode*	Ad-hoc ("WLg") Infra (others)		configure/display the WIFI mode. One of **ad-hoc** or **infra**. **ad-hoc :** configure the device in AD-HOC mode . **Infra :** configure the device in infrastructure mode.
set **show**	**net channel** **net channel**	*channel*	6		In ad-hoc mode, configures the radio channel used for communication with the other device. *channel* is in the range **0** to **13**. In infrastructure mode this parameter is ignored.
set	**net wepkey**	*keynum key*	no default value		define up to 4 WEP keys. *keynum* is the key number. Range **1** to **4**. *key* is the hexadecimal key value. 10 digits (64 bits key) or 26 digits (128 bits key). ***The last 6 digits are generated by the firmware*** Example : set 64 bits WEP key : set net wepkey 1 1F2564AE12 set 128 bits WEP key : set net wepkey 1 123654875ADFEC236542541A26 ***Note : to enter a 128 bits WEP key, you must before enable 128 bits key mode. See command "set net usekey 1 128" below.***
set **show**	**net wepkey** **net wepkey**	*keynum* **0**			delete wepkey *keynum* display all 4 WEP keys (the last 6 digits are displayed as zeroes).
set **show**	**net usekey** **net usekey**	*[keynum] [128]*			Define the WEP key to use. If the *keynum* parameter is left empty, device won't use any WEP key, else device uses WEP key *keynum*. Example: Activate 64 bits WEP key set net usekey 1 Activate 128 bits WEP key set net usekey 1 128 Disable WEP key using set net usekey
set	**net auth**	*mode*	open		set the authentication mode. *mode* is one of **open**, **share** ***open*** : the device is authenticated by its MAC address. ***share*** : the device is authenticated by its WEP Key. ***This command is not valid for WL-COMETH I.***

Command			Default value	Notes	Description
set	**net unencrypted**	*mode*	Ignore (WLg-range) Accept (other products)		configure if the device accept or ignore the unencrypted WIFI packet. mode is one of **ignore** or **accept** **ignore** : The device ignores all unencrypted WIFI packets **accept** : The device accepts all unencrypted WIFI packets ***This command is not valid for WL-COMETH I.***
set	**net txrate**	*txrate*	automatic		set the WIFI transmit rate. *txrate* is one of **1**, **2**, **5.5**, **11**, **automatic.** **1**, **2**, **5.5** or **11:** device will always use the given transmit rate. **automatic:** device will automatically choose the appropriate transmit rate.
show	**net wlan**				Display WIFI parameters : channel, txrate, authentification mode, RF signal quality. ***authentification mode is not displayed for WL COMETH I.*** ***"WLg" products also display available access points around.***

Command			Default value	Notes	Description
set **show**	**serial mode** **serial mode**	*mode*	rfc2217		Change/display TCP connection mode. *mode* is one of **rfc2217** / **telnet** / **raw** **rcf2217** is used with COM ports redirectors like VIP. **Telnet** allows connection from a TELNET client, this is specially useful for testing. **Raw** mode allows connection from a TCP Socket-based application.
set **show**	**serial port** **serial port**	*nnnn*	2300		Change/display the TCP port used for data transfer. *nnnn* is **1** to **65534** ***except 23***
set **show**	**serial interface** **serial interface**	*mode [option]*	rs232		*mode* : one of **rs232**/**rs422**/**4wires**/**rs485**/**2wires** option : **master** or **slave** for rs422 / 4wires mode, **noecho** or **echo** for rs485 / 2wires mode • On some products, only "rs232" is meaningful. Other choices will result in communication errors. See the serial port specifications of the appropriate port server user manual. • Keywords "**rs422**" and "**4wires**" are synonyms. Their meaning is identical. • Keywords "**rs485**" and "**2wires**" are synonyms. Their meaning is identical. **rs232** : setting for rs232 serial interface equipment **rs422 master** or **4wires master** : setting for master equipment in multidrop, configuration or for both equipments in point to point configuration **rs422 slave** or **4wires slave** : setting for slave in multidrop configuration. **rs485 noecho** or **2wires noecho** : setting for all devices in multidrop or point to point. **rs485 echo** or **2wires echo** : setting for all equipments in multidrop or point to point configuration. In this mode, transmitted characters on RS485 line are echoed on Lan line.
set **show**	**serial dtr** **serial dtr**	*mode*	driven		DTR management**.** *mode* is one of **driven**/**modem**/**high**/**low.** **Driven** is used in « rfc2217 » mode, it means that the signal will be driven by remote control. **Modem** means the signal is used as if a modem was connected to the port (DTR ➔ the device is on line, RTS ➔ the device wants to send data). **High** and **low** mean the signal is permanently set in this state.
set **show**	**serial rts** **serial rts**	*mode*	driven		RTS management**.** *mode* is one of **driven**/**modem**/**high**/**low**/**flow.** **Driven** is used in « rfc2217 » mode, it means that the signal will be driven by remote control. **Modem** means the signal is used as if a modem was connected to the port (DTR ➔ the device is on line, RTS ➔ the device wants to send data). **Flow** means the signal is used for input flow control. **High** and **low** mean the signal is permanently set in this state.
set **show**	**serial dsr** **serial dsr**	*mode*	ignore		DSR management. *mode* is one of **ignore** / **modem** **Modem** means the signal is used to check that the external device is on line, **Ignore** means that the signal is locally ignored and, in « rfc2217 » mode, that the remote control will be notified.
set **show**	**serial cts** **serial cts**	*mode*	ignore		CTS management. *mode* is one of **ignore** / **modem** / **flow** **Modem** means the signal is used to check that the external device allows the port server to send data. **Flow** means the signal is used for output flow control. **Ignore** means that the signal is locally ignored and, in « rfc2217 » mode, that the remote control will be notified.

Command			Default value	Notes	Description
set **show**	**serial dcd** **serial dcd**	*mode*	ignore		DCD management**.** *mode* is one of **ignore** / **modem** **Modem** means the signal indicates wether received data is valid. **Ignore** means that the signal is locally ignored and, in « rfc2217 » mode, that the remote control will be notified.
set **show**	**serial ring** **serial ring**	*mode*	ignore		RING management **:** *mode* : always **ignore**
set **show**	**serial baudrate** **serial baurate**	*speed*	9600		*speed* : any baud rate from **10** bauds to **230400** bauds (up to 1'000'000 on RS422/RS485 "WLg" products)
set **show**	**serial format** **serial format**	*nbits parity nstops*	8 n 1		*nbits* is **7** or **8** bits, *parity* is one of **e, o, n, m, s** (meaning even, odd, none, mark or space), *nstops* is **1** or **2** stop bits. (*nbits*=**6** bits is also supported on the "WLg" range)
set **show**	**serial xonxoff** **serial xonxoff**	*mode*	ignore		software flow control : *mode* is one of **use** or **ignore**. Mixed (i.e. software and hardware) flow control can be set.
set **show**	**serial tdsr** **serial tdsr**	*delay*	5		*delay* : acceptable delay between DTR rise and corresponding DSR rise at the beginning of a data session, when DSR is in modem mode. DTR➔DSR in tenth of second. **0** to **255**
set **show**	**serial toff** **serial toff**	*duration*	5		*duration* : when **DTR** is in modem mode, and the TCP connection is closed or lost, DTR will stay low for at least *duration*, expressed in tenths of second. **0** to **255**.
set	**sendtrigger charcount**	*count*	Off (0)	notes 5,6,7	number of chars required in the buffer before emission to the client application. Allowed values range from **0** to **255**. When this parameter is not **0**, data received on the asynchronous serial port will not be sent to the client application until there are at least *count* characters in the buffer. Set this parameter to **0** to disable it.
set	**sendtrigger framedelay**	*delay*	Off (0) ("WLg") 2 ms (others)	notes 5, 6, 7, 8	delay between char reception and emission to the client application. Allowed values range from **0** to **255**.The *delay* can be specified in milliseconds by appending a '**m**' to the figure, or in character duration by appending a '**c**' to the figure. '**m**' is the default if no unit is specified. When this parameter is not **0**, data received on the asynchronous serial port will not be resent to the client application until the specified delay has elapsed, after which, all data received in the meantime will be sent. Set this parameter to **0** to disable it.
set	**sendtrigger idledelay**	*delay*	3 ms ("WLg") Off (0) (others)	notes 5, 6, 7, 8	delay between last char reception and emission to the client application. Allowed values range from **0** to **255**.The *delay* can be specified in milliseconds by appending a '**m**' to the figure, or in character duration by appending a '**c**' to the figure. '**m**' is the default if no unit is specified. When this parameter is not **0**, data received on the asynchronous serial port will not be resent to the client application until the specified delay has elapsed since the last character was received, after which, all data received will be sent. Set this parameter to **0** to disable it.
show	**sendtrigger**		send when timeout after 1st char = 2ms or buffer full	notes 5, 6, 7, 8	display the condition used to put the data received on the asynchronous serial port, in the queue for transmission to the client application.

Command			Default value	Notes	Description
set	**wlan**				Run the wizard asking for the WiFi parameters
show	**wlan**				Display the WiFi parameters.
set	**wlan** *{options….}*				Change specific WiFi parameters (you can specify one or more of the following parameters) :
		state	on		*state*=**on** or **off** .Turns radio card on or off
		topology	adhoc		*topology*= one of **infra** or **adhoc**
		ssid *string*	acksys		change the ssid of the device. *string* is a case sensitive characters string.
		band	bg		change the radio protocol: *band*= one of **bonly gonly bg ah** (standard 802.11 protocols)
		superag	sagoff		*superag*= one of **sagoff sagon sagdyn sagstatic** Super AG mode is an atheros card feature.
		region	eu		*region*= one of **il us hk ca au fr eu jp sg kr** (standardized code of the world region).
		chan *channels*	auto		List of channels checked for access points. Available values depend on the region and the band. **auto** allows to scan all the channels allowed in the region.
		antennas	diversity		*antennas*= one of **diversity main aux** If your product has only one antenna, choose diversity or main. If your product has 2 antennas you can choose diversity to use both antennas or specify which antenna you want to use (main or aux).
		tx rate	best		you can enforce a specific standard bit rate. “**best**” selects the best rate available for the given band and reception quality.
		tx power	high		you can change the radio output power *tx power* = one of **high medium low**
		roaming	0 (off)		set the reception level under the bridge will search another access point. The reception level can be specified in units of dBm with negative values, or in percentage with positive values. example: `set wlan infra ssid myssid ah low` This command will be change to infrastructure mode with ssid “myssid” and radio protocol 802.11a/h and a low transmit power.

Command			Default value	Notes	Description
set	**wkey**				Run the wizard asking for the WiFi security parameters
show	**wkey**				Display the WiFi security parameters.
set	**wkey** *{option}*				Change specific WiFi security parameters (you can specify one or more of the following parameters) :
		method	off		*method*= **off** (no security or WEP key), **personal** (uses WPA protocol with a pre-shared key) or **enterprise** (not implemented)
		protocol	wpa		*protocol*= **wpa** or **wpa2**
		cipher	tkip		*cipher*= **tkip** or **aes**. Usually TKIP is used together with WPA and AES is used together with WPA2.
		password *str*	unspecified		change the pre-shared key to *str*.
ping	*ip-adress*				Sends ICMP ECHO-REQUEST four times to the specified destination. The answer (or timeout indication) will be displayed a few seconds after the prompt.
stat					Displays various indications for technical support purpose.
rxfifo	*state*		on		reserved for factory tests. DO NOT CHANGE.

Specifications:

- Product Name: SERVERCOM Firmware
- Compatible with: Ethernet and Wi-Fi Port Servers
- Release Version: A.4, April 22, 2009
- Manufacturer: ACKSYS

FAQ

1. What is the purpose of SERVERCOM firmware?

The SERVERCOM firmware allows port servers to function as distant serial communication ports for TCP/IP compliant computers.

2. Can SERVERCOM operate in multiple modes?

Yes, SERVERCOM can operate in RFC2217 mode, Raw mode, and Telnet mode depending on the requirements of the application software.

3. How can I set up serial control signals using the port server administration interface?

You can configure the serial control signals locally through the administration interface to ensure faster response time for flow control between the port server and the serial device.

Documents / Resources

	ACKSYS DTUS0434 Servercom Firmware For Ethernet and Wi-Fi Port Servers [pdf] User Guide DTUS0434, DTUS0434 Servercom Firmware For Ethernet and Wi-Fi Port Servers, Servercom Firmware For Ethernet and Wi-Fi Port Servers, Firmware For Ethernet and Wi-Fi Port Servers, Ethernet and Wi-Fi Port Servers, Wi-Fi Port Servers, Servers

References

- Debian -- Details of source package sredird in sid
- Solutions de communication WiFi et cellulaires ACKSYS
- Solutions de communication WiFi et cellulaires ACKSYS
- ietf.org/rfc/rfc2217.txt
- User Manual