



Simply Better Connections

Command Line Interface (CLI) Guide

ATEN Control System

Contents

About This Guide.....	2
Establishing an SSH/Telnet Session	3
SSH/Telnet Commands	5
Help	5
Enabling/Disabling Echo Reply	5
Configuring Telnet Timeout Interval or Login	5
Configuring Serial Port Settings.....	6
Configuring Relay Port Settings	7
Configuring I/O Port Settings	9
Configuring IR Settings	10
Reboot	11

About This Guide

This guide provides information on the available SSH and Telnet commands to remotely manage ATEN controllers and expansion boxes.

IMPORTANT: Configuration made via command-line interface will be overwritten if a project file is uploaded through ATEN Configurator (GUI) afterwards.

Establishing an SSH/Telnet Session

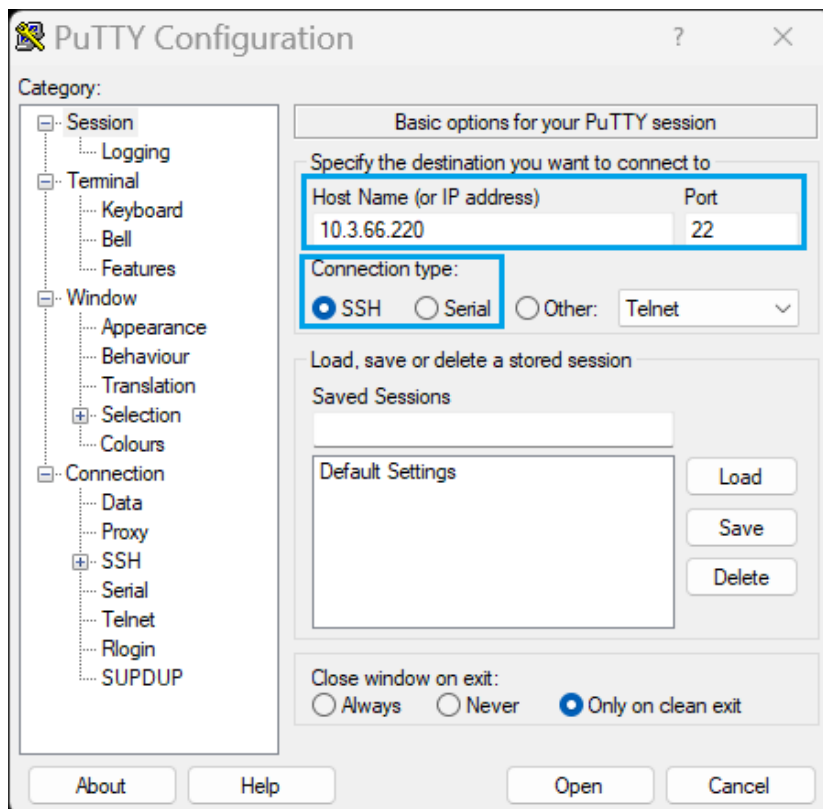
1. On a computer that has access to the target ATEN controller or Expansion Box, download an SSH/Telnet client, e.g. PuTTY.

Note: To obtain the installer, search “download putty” in a web browser.

2. Run the downloaded client.
3. To establish an **SSH session**, use the following settings.

Field	Setting
Host Name / IP address	<i>IP address or host name of the target controller or expansion box</i>
Connection Type	SSH
Port	22

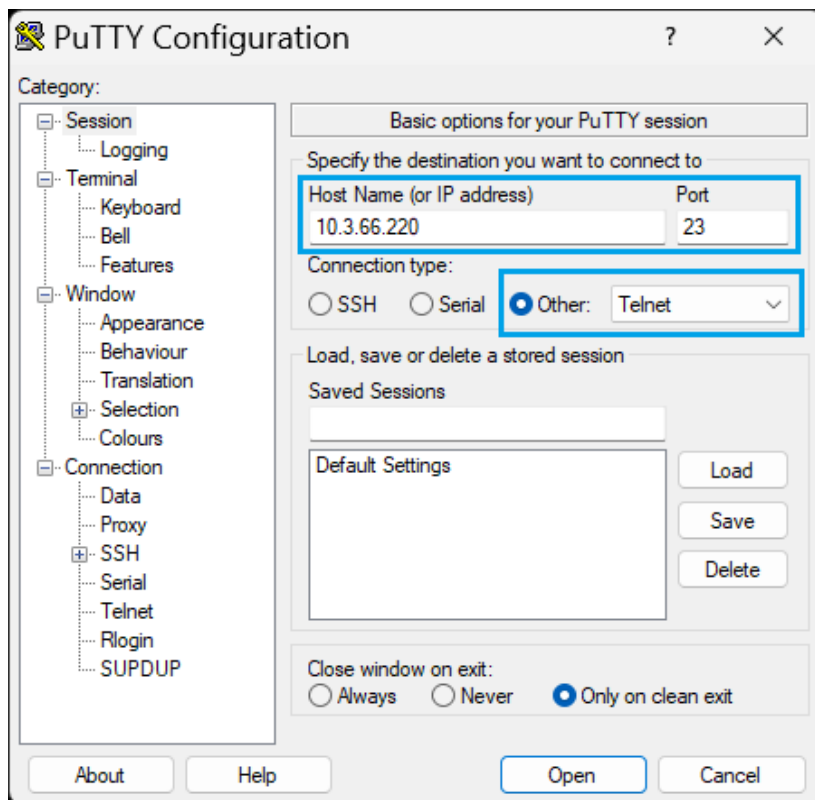
For example:



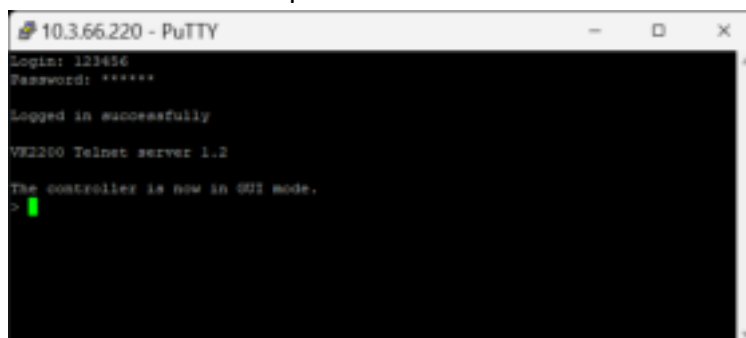
4. To establish a **telnet session**, use the following settings.

Field	Setting
Host Name / IP address	<i>IP address or host name of the target controller or expansion box</i>
Connection Type	Other > Telnet
Port	23

For example:



5. Click **Open** and follow the on-screen instructions to enter the login credentials. A session is established. For example:



Note: If the session cannot be established, log in the device management console to check if the access key needs to be changed.

SSH/Telnet Commands

Help

Usage:

Display the instruction for enabling the CLI mode of the controller.

Syntax:

help↵

Acknowledge:

Command OK↵ : Command is correct and the function is executed.

Command incorrect↵ : Command and/or parameters are incorrect and not executed.

Example:

help↵ : display instruction of CLI mode

Enabling/Disabling Echo Reply

Usage:

Enable or disable the controller to automatically response to received messages.

Syntax:

echo[control]↵

Parameter:

control : on - enable this function

off - disable this function (default is off)

Acknowledge:

Command OK↵ : Command is correct and the function is executed.

Command incorrect↵ : Command and/or parameters are incorrect and not executed.

Example:

eho on↵ : set the controller to automatically respond to received messages.

Configuring Telnet Timeout Interval or Login

Usage:

Configure Telnet CLI mode settings.

Syntax:

telnet[timeout interval][login control]↵

Keyword:

timeout : Telnet session timeout interval setting

login : Telnet login function setting

Parameter:

interval : session timeout in minute. (0 means never timeout)(default is 5)

control : on – enable login function (default is on)

off – disable login function

Acknowledge:

Command OK↵ : Command is correct and the function is executed.

Command incorrect↵ : Command and/or parameters are incorrect and not executed.

Example:

telnet timeout 0↵ : configure the timeout interval to never timeout

telnet timeout 5↵ : configure the timeout interval to 5 minutes

telnet login off↵ : disable the login function

Configuring Serial Port Settings

Usage:

Control and configure serial port settings.

Syntax:

serial [p sequence] [type type] [baud baudrate] [dbit databit] [parity sparity] [sbit stopbit]
[fctrl flowctrl] [dtype datatype] [endchar chars] [checksumtype checksumtype] [acktimeout
duration] [control "data"] [help]↵

Keyword:

p : port

type: serial type

baud : baud rate

dbit : data bit

parity : serial parity

sbit : stop bit

fctrl : flow control

dtype : data type

acktimeout : timeout duration (ms) is used for waiting for the feedback from sendack control.

endchar : specific end character is used to identify a complete message.

checksumtype : if need fw to auto-calculate command checksum, assign a checksum type

help: show instructions

Parameter:

sequence : port, separated by comma for multiple ports

* (all ports)

stype: 232, 422, 485 (default 232)
 baudrate : 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200(default 9600)
 databit : 7, 8 (default 8)
 parity : none, even, odd (default none)
 stopbit : 1, 2 (default 1)
 flowctrl : none, hw (default none)
 datatype : ascii, hex (default ascii)
 checksumtype: none, modbus (default none)
 duration : specific timeout (ms)(default 300)
 chars : end characters following a complete message.
 control : read - display received message of the specific serial port
 write - send data from the serial port. Use " symbol covers data. Use \x symbol to
 send Hex code
 sendack - send data from the serial port and display received message

Acknowledge:

Command OK↵ : Command is correct and the function is executed.

Command incorrect↵ : Command and/or parameters are incorrect and not executed.

Example:

Serial help ↵ : show instructions

serial p01 ↵ : display the setting information of serial port1.

serial p* ↵ : display the setting information of all serial ports.

serial p01 type 232 baud 115200 dbit 8 parity none sbit 1 fctrl none ↵ : configure port1.

serial p01,04,07 baud 19200 ↵ : configure port1, 4, 7 to baud rate 19200.

serial p02 baud 19200 ↵ : configure port2 to baud 19200 and use default settings for other parameters.

serial p02 dtype hex read ↵ : display the received message of port2 in HEX format.

serial p03 write "sw i01 o02\r\n" ↵ : send the ASCII text 'sw i01 o02' from port3.

serial p03 sendack "sw i01 o02\x0D\x0A" ↵ : send the ASCII text 'sw i01 o02' from port3 and display the feedback.

Configuring Relay Port Settings

Usage:

Control and configure relay ports.

Syntax:

relay [p sequence] [tpulse duration] [control] [help] ↵

Keyword:

p : port

tpulse : closed period for pulse

help: show instructions

Parameter:

sequence : port, separated by comma for multiple ports
* (all ports)

duration : time period in millisecond.

control : open - turn off
close - turn on
pulse - close the relay channel then open
toggle - relay toggle
read - display status

Acknowledge:

Command OK↯ : Command is correct and the function is executed.

Command incorrect↯ : Command and/or parameters are incorrect and not executed.

Example:

relay help ↯ : show instructions

relay p01 close ↯ : close port1 channel.

relay p01 open ↯ : open port1 channel.

relay p01,04,07 close ↯ : close port 1, 4, 7 relay channels.

relay p02 tpulse 500 pulse ↯ : close port2 500ms then open.

relay p02 read ↯ : display port2 status.

Configuring I/O Port Settings

Usage:

Control and configure I/O ports.

Syntax:

io [p sequence] [type iotype] [lthresh threshold] [hthresh threshold] [tpulse duration] [control] [help]↵

Keyword:

p : port
lthresh : low-bound threshold
hthresh : high-bound threshold
tpulse : period which I/O remains in high level in pulse mode
help: show instructions

NOTE: If a parameter is not specified, a previously entered value will be applied.

Parameter:

sequence : port, separated by comma for multiple ports
 * (all ports)
iotype : dry, vdc, dout
threshold : trigger threshold in voltage
duration : time period in millisecond.
control : open
 close
 pulse
 toggle
 read

Acknowledge:

Command OK↵ : Command is correct and the function is executed.

Command incorrect↵ : Command and/or parameters are incorrect and not executed.

Example:

io help ↵ : show instructions
io p01↵ : display the setting information of I/O port1.
io p*↵ : display the setting information of all I/O ports.
io p01 type dry ↵ : configure port 1 to dry contact mode.
io p01 type dout ↵ : configure port 1 to digital output mode.
io p01 type vdc lthresh 1 hthresh 3 ↵ : configure port 1 to vdc mode with threshold settings.
io p01 open ↵ : open port1 channel.
io p01 close ↵ : close port1 channel.
io p01,04,07 close ↵ : close port1,4,7 channels.
io p02 tpulse 500 pulse ↵ : close port2 channel 500ms then open.

io p02 read ↵ : display port2 status.

Configuring IR Settings

Usage:

Control and configure IR ports.

Syntax:

```
ir [p sequence] [type irtype] | [baud baudrate] [dbit databit] [parity sparity]
  [sbit stopbit] [dtype datatype] [checksumtype checksumtype] [control "data"]
  [help]↵
```

Keyword:

p :	port
type :	output type for specific output port
baud** :	baud rate
dbit** :	data bit
parity** :	serial parity
sbit** :	stop bit
dtype** :	data type
checksumtype** :	Controller/extension box automatically calculates the specified checksum type
help:	show instructions

NOTE:

* : This keyword is for IR type

** : These keywords are for 232 type

If a parameter is not specified, a previously entered value will be applied.

Parameter:

sequence :	port, separated by comma for multiple ports * (all ports)
irtype:	ir, 232(default ir)
baudrate :	300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200(default 9600)
databit :	7, 8(default 8)
sparity :	none, even, odd(default none)
stopbit :	1, 2(default 1)
datatype :	ascii, hex(default ascii)
checksumtype:	none, modbus(default none)
control :	write - send IR code or RS232 data

Acknowledge:

Command OK↵ : Command is correct and the function is executed.

Command incorrect↯ : Command and/or parameters are incorrect and not executed.

Example:

ir help ↯ : show instructions

ir p01 ↯ : display the setting information of IR port1.

ir p* ↯ : display the setting information of all IR ports.

ir p01,04,07 type 232 ↯ : configure port1,4,7 to RS232 type

ir p01 type 232 baud 115200 dbit 8 parity none sbit 1 checksum type modbus↯ :
configure port1 to RS232 type and the settings

ir p02 write dtype hex "0a0b0c" ↯ : send hex format data 0a0b0c from port2

ir p02 baud 19200 ↯ : configure port2 to baud 19200 and use the default settings for other parameters.

ir p02 write "sw i01 o02\r\n" ↯ : send the ASCII text 'sw i01 o02' from port2.

ir p02 write "sw i01 o02\x0D\x0A" ↯ : send the ASCII text 'sw i01 o02' from port2.

Reboot

Usage:

Reboot the controller.

Syntax:

reboot↯

Acknowledge:

Command OK↯ : Command is correct and the function is executed.

Command incorrect↯ : Command and/or parameters are incorrect and not executed.

Example:

reboot↯ : reboot device