VAPIX® VERSION 3

Serial Port API

AXIS COMMUNICATIONS

Serial Port API

COPYRIGHT NOTICE

VAPIX® LICENSE AGREEMENT

This VAPIX® License Agreement ("License") is a legal agreement between you (either individual or an entity) and Axis Communications AB ("Axis"). By using the INTERFACE and INTERFACE DESCRIPTION (each defined below), whether in whole or in part, you agree to be bound by the terms of this License.

1. GRANT OF LICENSE

Axis hereby grants to you the right to use the AXIS VAPIX application programming interface ("INTERFACE") and the written specification of the INTERFACE (the "INTERFACE DESCRIPTION") for the sole and limited purpose of creating, manufacturing and developing a solution that integrates any unit or portion included in the product range of Axis network products, as defined by Axis at its discretion (an "Axis Product") and to market, sell and distribute any such solution.

2. COPYRIGHT

The INTERFACE and the INTERFACE DESCRIPTION are owned by Axis and are protected by copyright laws and international treaty provisions. Any use of the INTERFACE and/or the INTERFACE DESCRIPTION outside the limited purpose set forth in Section 1 above is strictly prohibited.

3. RESTRICTIONS ON USE

You have no rights with respect to the INTERFACE, INTERFACE DESCRIPTION or any portions thereof and shall not use the INTERFACE, INTERFACE DESCRIPTION or any portion thereof except as expressly set forth herein. You may not reverse engineer, decompile, or disassemble the INTERFACE except to the extent required to obtain interoperability with other independently created computer programs as permitted by mandatory law.

4. THIRD PARTY RIGHTS

You agree that you are fully responsible for your own conduct while using the INTERFACE and integrating any Axis Products into your solution and the consequences thereof. Axis Products may be combined with a virtually infinite number of potential solutions. Consequently, you recognize that (i) other third parties may claim to own patents or copyrights that could cover certain solutions which integrate Axis products, or which result from the combination of Axis products and additional technology or solutions and (ii) you are responsible for ensuring that any solution which integrates with an Axis Product, or a combination of a solution and an Axis product, does not infringe upon or misappropriate any intellectual property or personal right of any third party.

5. TERMINATION

This License is effective until terminated. Your rights under this License will terminate automatically without notice from Axis if you fail to comply with any term(s) of this License. Upon the termination of this License, you shall cease all use and disposition of the INTERFACE and/or THE INTERFACE DESCRIPTION whether for the purpose set forth in Section 1 above or not.

6. REPRESENTATIONS AND WARRANTIES; DISCLAIMER

6.1. You represent and warrant that (i) any solution created, manufactured and/or developed by you which integrates an Axis Product shall not infringe or otherwise violate any third party rights, including but not limited to third party intellectual property rights; and (ii) your use of the INTERFACE and INTERFACE DESCRIPTION will comply with all applicable foreign and domestic laws, rules and regulations.

6.2. YOUR USE OF THE INTERFACE IS AT YOUR SOLE RISK. THE INTERFACE AND THE INTERFACE DESCRIPTION ARE DELIVERED FREE OF CHARGE AND "AS IS" WITHOUT WARRANTY OF ANY KIND. THE ENTIRE RISK AS TO THE USE, RESULTS AND PERFORMANCE OF THE INTERFACE AND THE INTERFACE DESCRIPTION IS ASSUMED BY THE USER/YOU. AXIS DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, NON-INFRINGEMENT AND PRODUCT LIABILITY, OR ANY WARRANTY ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE WITH RESPECT TO THE INTERFACE AND THE INTERFACE DESCRIPTION. Without limiting the generality of the foregoing, you acknowledge and agree that Axis does not make any representation or warranty that the integration of Axis Products into your solution does not infringe any third party rights. You are solely responsible for any intellectual property infringement claims that are based on or relate to solutions created, manufactured and distributed by you which integrate Axis Products. Axis is unaware of the details regarding your particular solution, has not conducted any investigation relating to potential third party rights issues relating to your solution and does not accept any responsibility or liability with respect thereto.

6.3. THIS LICENSE DOES NOT CONVEY ANY LICENSE TO THIRD PARTY INTELLECTUAL PROPERTY. YOU ARE SOLELY RESPONSIBLE FOR (I) EXAMINING WHETHER THE INTERFACE AND THE INTERFACE DESCRIPTION ARE ENCUMBERED BY OR INFRINGES UPON A RIGHT HELD BY A THIRD PARTY AND (II) ANY INTELLECTUAL PROPERTY INFRINGEMENT CLAIMS THAT ARISE OUT OF OR RELATE TO SOLUTIONS CREATED, MANUFACTURED AND DISTRIBUTED BY YOU WHICH INTEGRATE AXIS PRODUCTS.

7. LIMITATION OF LIABILITY

7.1. AXIS SHALL NOT BE LIABLE FOR LOSS OF DATA, LOSS OF PRODUCTION, LOSS OF PROFIT, LOSS OF USE, LOSS OF CONTRACTS OR FOR ANY OTHER CONSEQUENTIAL, ECONOMIC OR INDIRECT LOSS WHATSOEVER IN RESPECT OF USE OR DISPOSITION OF THE INTERFACE AND THE INTERFACE DESCRIPTION.

7.2. AXIS TOTAL LIABILITY FOR ALL CLAIMS IN ACCORDANCE WITH THE USE OF THE INTERFACE AND THE INTERFACE DESCRIPTION SHALL NOT EXCEED THE PRICE PAID FOR THE INTERFACE AND THE INTERFACE DESCRIPTION.

7.3. YOU UNDERTAKE NOT TO PURSUE ANY CLAIMS WHATSOEVER AGAINST AXIS OR ITS AFFILIATES RELATING TO OR EMANATING FROM THE INTERFACE AND THE INTERFACE DESCRIPTION OR YOUR INTEGRATION OF AN AXIS PRODUCT INTO YOUR SOLUTION.

8. INDEMNIFICATION

You will indemnify and hold Axis, its subsidiaries, affiliates, officers, employees, and agents harmless from any and all claims, damages, losses, liabilities, actions, judgments, costs, and expenses brought by a third party, including claims for infringement of intellectual property rights, arising out of or in connection with (i) your use of the INTERFACE or INTERFACE DESCRIPTION other than in accordance with the terms of this agreement, and/or (ii) any solution created, manufactured and/or developed by you which integrates an Axis Product.

9. GOVERNING LAW

This agreement shall be deemed performed in and shall be construed by the laws of Sweden. All disputes in connection with this agreement shall be finally settled by arbitration in accordance with the Rules of the Arbitration Institute of the Stockholm Chamber of Commerce. The place of arbitration shall be Malmö, Sweden. The language of the proceedings, documentation and the award shall be English.

Serial Port API

Table of Contents

Serial Port API

1 Overview

1.1 Description

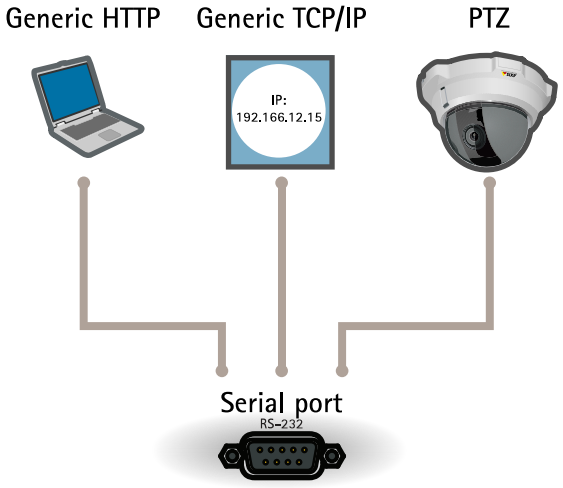
This document describes the VAPIX® Serial Port API on Axis products with serial ports. The information in this document is intended to be used to get knowledge about how to change settings for the serial port. For example when to change serial port settings for an uploaded PTZ driver or get to know what type of settings that could be changed when adding some type of device with a serial port to an Axis product.

The serial port allows connection of equipment such as analog PTZ cameras to Axis products. The equipment is connected to a serial port on the Axis product, which in turn works as a translator between a network client and the equipment.

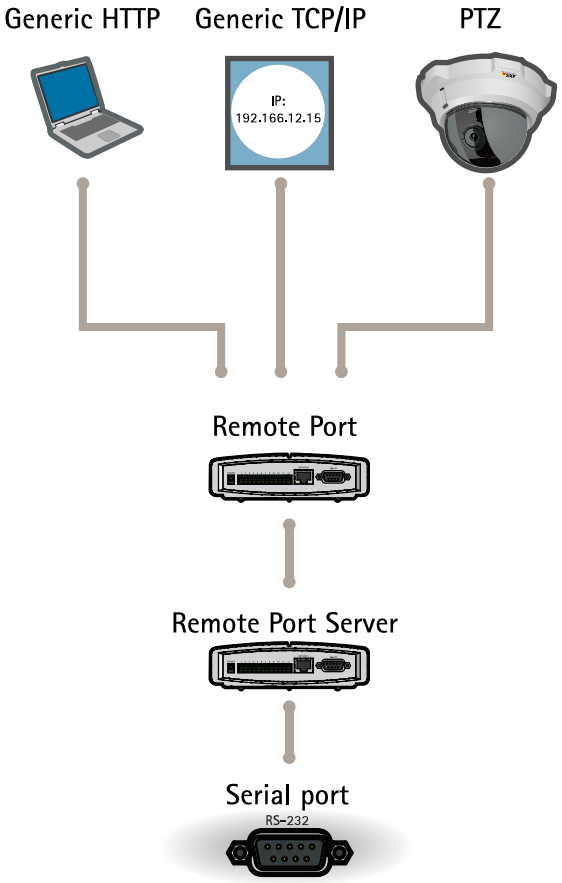
Ports can be a physical serial ports or remote port clients. Remote port clients connect to other CPU:s that have remote port servers open (the other product has a physical serial port that is made available by the server). The remote port connection is used for daisy chain cameras on multi CPU video encoders. A daisy chain connection means virtually connecting several cameras (without physical serial ports) to a CPU with a physical serial port.

For both types of ports, there is a port manager that manages the port. The port manager makes it possible to share the port, for example for PTZ control from multiple video channels.

Settings for port managers are stored in the parameter group `PortManager.P#` (see *PortManager.P#*). Serial ports are described in the group `Serial.Ser#` (see *Serial.Ser#*), and remote port clients in `RemotePort.R#` (see *RemotePort.R#*).



The image above describes the different types of data that can be sent using a physical serial port.



The image above describes an overview of a connection between a remote serial port, a remote port server and a serial port. The remote port forwards data from a virtual port on a CPU to a physical port on another CPU. This is used in multi CPU encoders to share the serial port between the CPU:s. This functionality is only intended for multi CPU video encoders. Remote port is similar to Generic TCP/IP, but in addition to Generic TCP/IP it also adds a protocol header to the data payload.

1.2 Prerequisites

1.2.1 Identification

Property: `Properties.API.HTTP.Version=3`
Property: `Properties.Serial.Serial=yes`

Firmware: 5.20 or later
Product category: Axis products with serial ports.

1.3 References

All VAPIX® references are available at:

http://www.axis.com/vapix

2 Common Examples

Example 1:
Send data to a device connected to the serial port. The argument `write` specifies how much data that should be sent.

```
http://myserver/axis-cgi/com/serial.cgi?write=2153435253500c0d
```

Example 2:
Read `3` bytes from the selected serial port with a time out of 2000 milliseconds.

```
http://myserver/axis-cgi/com/serial.cgi?read=3&timeout=2000
```

Example 3:
Set the baud rate (transmission speed) to `9600` for the serial port and the number of databits included in the set to `8`.

```
http://myserver/axis-cgi/param.cgi?action=update&Serial.Ser1.BaudRate=9600&DataBits=8
```

Example 4:
Enable a generic TCP server on port `3000`.

```
http://myserver/axis-cgi/param.cgi?action=update
&PortManager.P0.GenericTCPServer.Enabled=yes
&PortManager.P0.GenericTCPServer.Listener.Enabled=yes
&PortManager.P0.GenericTCPServer.Listener.Port=3000
&PortManager.P0.PortEnabled=yes
```

Clients that connect to port `3000` will get a TCP socket where raw data can be sent and received. That data will be forwarded to and from the serial port.

Example 5:
Request the remote port client to be managed by port manager `0`.

```
http://myserver/axis-cgi/param.cgi?action=update
&PortManager.P0.PortType=RemotePort
```

Example 6:
Set a remote port server to listen on port `2500` of the host `10.0.0.1`, and connect the remote port client to it.

```
http://myserver/axis-cgi/param.cgi?action=update
&RemotePort.R3.Host=10.0.0.1
&RemotePort.R3.Port=2500
&PortManager.P0.PortEnabled=yes
```

PTZ commands sent out through the remote port client will reach the remote PTZ server.

Example 7:
Open port 2 (`PortManager.P0`), if not already opened, and connect video source 2 to it.

```
http://myserver/axis-cgi/com/serial.cgi?port=2&camera=2
```

Example 8:
Disconnect video source 2 from any port.

```
http://myserver/axis-cgi/com/serial.cgi?port=-1&camera=2
```

3 Parameters

3.1 Port Manager

3.1.1 PortManager.P#

The parameter group `PortManager.P#` describes a port manager. The indices # in `PortManager.P#` go from `0` to (number of port managers – `1`).

[PortManager.P#]

Parameter	Default values	Valid values	Access control	Description
`PortEnabled`	`yes`	`yes` `no`	admin: read, write operator: read	Enable or disable the output port. The output port is defined by the parameter `PortType`. `yes` = Enable the output port. no = Disable the output port.
`PortType`	Product dependent.	`Serial` `RemotePort`	admin: read, write operator: read	Describes what type of local port this port manager manages. `Serial` = The device connects directly via a serial port. `RemotePort` = The port data is forwarded to a physical serial port on another CPU.

3.2 Serial Ports

3.2.1 Serial

The parameter `Serial` controls the functionality/purpose of the serial ports and some configuration that is shared by various applications.

[Serial]

Parameter	Default values	Valid values	Access control	Description
`NbrOfPorts`	Product dependent.	An unsigned integer.	admin: read operator: read viewer: read	The number of physical serial ports on the product.

3.2.2 Serial.Ser#

The sub-group `Serial.Ser#` contains settings for a serial port. The index # in `Serial.Ser#` starts from `1`.

Serial Port API

[Serial.Ser#]

Parameter	Default values	Valid values	Access control	Description
PortMode	Product dependent.	RS232[1] RS485[1] RS485_4[1] RS422[1]	admin: read, write operator: read	Serial port protocol. May be read-only depending on hardware.
BaudRate	Product dependent.	300 600 1200 2400 4800 9600 19200 38400 57600 115200 230400[2] 460800[2]	admin: read, write operator: read	The baud rate (transmission speed) used in the serial communication. The listed values are transmission speed in bites/second.
DataBits	Product dependent.	7 8	admin: read, write operator: read	The number of data bits included in a set of bits.
StopBits	Product dependent.	1 2	admin: read, write operator: read	The number of stop bits included in a set of bits.
Parity	Product dependent.	None Even Odd Mark Space	admin: read, write operator: read	The parity. A parity bit is added to make sure that the number of bits with the value one in a set of bits is even or odd. None = No parity bits are added. Even = The parity bit is set to 1 if the number of ones in a given set of bits (not including the parity bit) is odd. Odd = The parity bit is set to 1 if the number of ones in a given set of bits (not including the parity bit) is even. Mark = The parity bit is always 1. Space = The parity bit is always 0.

[Serial.Ser#] (Continued)

FlowControlXONX-OFF	Product dependent.	yes no	admin: read, write operator: read	Enable and disable XON/XOFF flow control (requires a full duplex port).[3] yes = Enable XON/XOFF flow control. no = Disable XON/XOFF flow control.
FlowControlRTSCTS	Product dependent.	yes no	admin: read, write operator: read	Enable and disable RTS/CTS flow control is enabled (requires a RS232 port).[3] yes = Enable RTS/CTS flow control no = Disable RTS/CTS flow control.
Termination	Product dependent.	yes no	admin: read, write operator: read	Enable and disable RS485/RS422 termination.[3] yes = Enable RS485/RS422 termination. no = Disable RS485/RS422 termination.
Bias	Product dependent.	yes no	admin: read, write operator: read	Enable and disable RS485/RS422 bias.[3] yes = Enable RS485/RS422 bias. no = Disable RS485/RS422 bias.

1. Product-dependent. Check the product specification.
2. This value is only applicable to RS485 and RS485_4.
3. If this parameter is missing then this function is not supported.

Note

The # in Serial.Ser# is replaced with a number starting from 1.

3.2.3 PortManager.P#.SerialCGI

The Enabled parameter in this sub-group is used to enable or disable the internal server that serial.cgi uses for connecting to the port manager.

[PortManager.P#.SerialCGI]

Parameter	Default values	Valid values	Access control	Description
Enabled	no	yes no	admin: read, write operator: read	Enable or disable the server that serial.cgi needs. yes = Enable the server that serial.cgi needs. no = Disable the server that serial.cgi needs.

3.3 Remote Ports

Remote ports can be used to daisychain and forward data from a remote port on one CPU to a physical port on another CPU. This is used in video encoder blades to share the serial port between the CPU:s. This functionality is only intended for multi CPU video encoders. Daisy chain is similar to Generic TCP/IP, but in addition to Generic TCP/IP it also adds a protocol header to the data payload.

3.3.1 RemotePort.R#

The parameter group RemotePort.R# contains settings for a remote port client. The index # in RemotePort.R# starts from 0. Remote port server settings can be found in the parameter group PortManager.P#.RemotePortServer.

Note

If the Axis product does not support the functionality this parameter group does not exist.

[RemotePort.R#]

Parameter	Default values	Valid values	Access control	Description
AutoConf	yes	yes no	admin: read, write operator: read	Automatically determine the address and port of the remote port server using Bonjour®. yes = The client will try to determine the address and port of the remote port server automatically. no = The client will not try to determine the address and port of the remote port server automatically.

[RemotePort.R#] (Continued)

`Authentication-Key`		A string.	admin: read, write operator: read	The key to use for authentication when connecting to a remote port server. The authentication key for the remote port server can be found in the parameter `PortManager.P#.RemotePortServ-er.Authenticat-ionKey`.
`Host`		Host name or IP address.	admin: read, write operator: read	Host name or IP address of the remote port server. Ignored if `AutoConf=yes`.
`Port`	`5555`	`1024 ... 65535`	admin: read, write operator: read	Listen port of the remote port server. Ignored if `AutoConf=yes`.

3.3.2 PortManager.P#.RemotePortServer

The sub-group `RemotePortServer` contains parameters for remote port server, used for tunneling PTZ commands, generic TCP, and `serial.cgi` traffic.

Note

If the Axis product does not support the functionality this parameter group does not exist.

[PortManager.P#.RemotePortServer]

Parameter	Default values	Valid values	Access control	Description
`Enabled`	`no`	`yes` `no`	admin: read, write operator: read	Enable or disable the remote port server. `yes` = Enable the remote port server. `no` = Disable the remote port server.
`AllowedIPAddre-sses`		A comma separated list of IP addresses.	admin: read, write operator: read	Allow only certain IP addresses to connect to the Listener. The default value is an empty string. That means all IP addresses are allowed. Ranges may be specified. For example `1.2.3.4`, `2.3.4.5-9`, `3.4.5.*`

[PortManager.P#.RemotePortServer] (Continued)

Timeout	0	0 ... 3600	admin: read, write operator: read	If there is no activity on the connection for this time (in seconds), it will be closed. 0 = No timeout/wait forever.
Authentication-Key		A string.	admin: read, write operator: read	The key for the remote port clients to use when connecting to this remote port server.

3.3.3 PortManager.P#.RemotePortServer.Listener

The sub-group RemotePortServer.Listener contains settings for listening for connections from remote port clients.

Note

If the Axis product does not support the functionality this parameter group does not exist.

[PortManager.P#.RemotePortServer.Listener]

Parameter	Default values	Valid values	Access control	Description
Enabled	no	yes no	admin: read, write operator: read	Enable or disable to accept connections to this remote port server. yes = Enable connections to this remote port server. no = Disable connections to this remote port server.
Port	1024 + #[1]	1024 ... 65535	admin: read, write operator: read	The port to listen on for connections from remote port clients.

1. # is the indices in PortManager.P#.

3.3.4 PortManager.P#.PTZServer

The Enabled parameter in this sub-group is used to enable or disable the internal server that PTZ drivers use for connecting to the port manager.

[PortManager.P#.PTZServer]

Parameter	Default values	Valid values	Access control	Description
Enabled	no	yes no	admin: read, write operator: read	Used to enable or disable the server that PTZ drivers need. yes = Enable the server that PTZ driver need. no = Disable the server that PTZ drivers need.

3.4 Generic TCP

3.4.1 PortManager.P#.GenericTCPServer

The sub-group `GenericTCPServer` contains settings for the generic TCP server. Generic TCP means that the server listens on a TCP port, and when a client connects a socket is opened. Data received over the socket is sent out on the serial/remote port, and data received on the port is sent out over the socket.

[PortManager.P#.GenericTCPServer]

Parameter	Default values	Valid values	Access control	Description
`Enabled`	`no`	`yes` `no`	admin: read, write operator: read	Enable or disable the generic TCP server. `yes` = Enable the generic TCP server. `no` = Disable the generic TCP server.
`AllowedIPAddresses`		A comma separated list of IP addresses.	admin: read, write operator: read	Allow certain IP addresses to connect to the Listener. The default value is an empty string. That means all IP addresses are allowed. Ranges may be specified. For example `1.2.3.4`, `2.3.4.5-9`, `3.4.5.*`
`Timeout`	`0`	`0 ... 3600`	admin: read, write operator: read	If there is no activity on the connection for this time (in seconds), it will be closed. `0` = no timeout/wait forever.

3.4.2 PortManager.P#.GenericTCPServer.ConnectTo

The sub-group `GenericTCPServer.ConnectTo` contains settings used when the port manager acts as a client and connects to another host. A host could for example be another encoder or server. When data is received the host connects to the client via generic TCP. The connection is similar to a remote port connection (see *Remote Ports, on page 11*) but only raw data is sent directly, without a protocol.

[PortManager.P#.GenericTCPServer.ConnectTo]

Parameter	Default values	Valid values	Access control	Description
Enabled	no	yes no	admin: read, write operator: read	Enable or disable the port manager to act as a client and connect to another host. (It may still act as a server as well with Listener.Enabled=yes.) yes = Enable port manager to act as a client. no = Disable port manager to act as a client.
Host		Host name or IP address	admin: read, write operator: read	Set the host name or IP address to use if this port manager should act as a client and connect to another host.
Port	1024 + #[1]	1024 ... 65535	admin: read, write operator: read	Set the TCP port number to use if this port manager should act as a client and connect to another host.

1. # is the indices in PortManager.P#.

3.4.3 PortManager.P#.GenericTCPServer.Listener

This sub-group contains settings for the server functionality of the generic TCP server.

[PortManager.P#.GenericTCPServer.Listener]

Parameter	Default values	Valid values	Access control	Description
Enabled	no	yes no	admin: read, write operator: read	Enable or disable connections to this TCP server. yes = Enable connections to this TCP server. no = Disable connections to this TCP server.
Port	1024 + #[1]	1024 ... 65535	admin: read, write operator: read	The TCP port to listen on, if enabled.

1. # is the indices in PortManager.P#.

4 HTTP API

4.1 Serial Port Control

The CGI `serial.cgi` enables the Axis product to write and read raw data on the serial port through a HTTP request.

Access control: viewer
Method: `GET/POST`

Syntax:

```
http://<servername>/axis-cgi/com/serial.cgi?<argument>=<value>[<argument>=<value>...]
```

With the following arguments and values:

Argument	Valid values	Description
`camera=<int>`	`1 ...`	Select the video channel. If omitted, the default video source is used.
`write=<string>`	*<bytestring>*	Write the specified data string to the selected port. *<bytestring>* = Hexadecimal coded bytes with values {`0`, `1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, `9`, `A`, `B`, `C`, `D`, `E`, `F`, `a`, `b`, `c`, `d`, `e`, `f`}.
`writestring=<string>`	A URL encoded string	Write the URL-encoded string to the selected port.
`read=<int>`	`1` ...	Reads *n* bytes from the selected port. The returned data will be hexadecimal coded and placed between `#`s (for example `#3A#`).
`timeout = <int>`	`0` ... `9000`	Specified in milliseconds. Used together with the `read` argument. A `read` is terminated when the specified number of bytes is `read` or the `timeout` has expired.

VAPIX® version 3
Serial Port API

Ver. M3.1
Date: July 2013
Part No. 50130