

B. Braun BCCProtocol Driver

File Name	BCCProtocol.dll
Manufacturer	B. Braun
Devices	SpaceCom
Protocol	BCC Protocol
Version	1.0.3
Last Update	06/12/2025
Platform	Win32
Dependencies	IOKit version 2.0
Superblock Readings	No
Level	0

Introduction

This Driver implements the BCC Protocol, allowing an application developed by **Elipse Software** to communicate with SpaceCom devices by B. Braun.

Driver Operation

To use B. Braun BCCProtocol Driver properly in an application, it is important to understand how it works, because this Driver does not have features most common to other I/O Drivers.

The configuration of SpaceCom devices by B. Braun determines which parameters are sent at each data request of this Driver. With each data request, the station sends all parameters at once. This Driver requests data by trying to respect the lowest scan value found in the group of Tags **In Advise**. The **In Advise** status of a Tag is defined by a constant update, at time intervals defined by its **Scan** property, that is, the scan periods.

The way to change a Tag status to **In Advise** is by configuring its **AllowRead** property to True and by one of these conditions:

- Tag's **AdviseType** property is equal to **0**: AlwaysInAdvise
- Tag's **AdviseType** property is equal to **1**: AdviseWhenLinked and it is linked to an active object on an open Screen

Only one Tag with an expired scan time is enough for this Driver to execute, in background, the whole data communication of all parameters. Thus, the smaller the variety of scan period values, the better, that is, the most optimized way is to try to keep all scan times the same, so that all Tags are read at the same time. This detail must be taken into account when using serial interfaces and low communication speeds. If the communication interface is Ethernet and the minimum scan period is 1000 ms or more, performance impact tends to none.

If any nonexistent parameter Tags in the whole data is **In Advise** and with an expired time, it may cause unnecessary data communication. Therefore, users must be aware of which parameters are requested, so that everything is configured on the remote device and available on data request's response.

The SpaceCom device by B. Braun must be configured in **Request** operation mode, that is, the device waits for this Driver's request to send data.

Driver Configuration

This Driver's **[P]** parameters are not used. All configurations are performed on this Driver's configuration dialog box. For more information about the configuration tabs, please check topic **Documentation of I/O Interfaces**.

Configuring Properties

All configuration parameters to perform communication with a device are available on this Driver's Properties Window.

In addition to this Properties Window, these settings can also be defined at run time in **Elipse E3**, **Elipse Power**, or **Elipse Water** applications. To do so, initialize this Driver in **Offline** mode, that is, execute an application with the **Start driver OFFLINE** option enabled, which is configured on the **Setup** tab of the Properties Window. The configuration options of this Driver are described on the next table.

Configuration options for B. Braun BCCProtocol Driver

TAB	PARAMETER	OFFLINE STRING	DATA TYPE	DESCRIPTION
Setup	Physical Layer	IO.Type	Text	Use the Serial or Ethernet option
	Timeout	IO.TimeoutMs	Number	A time limit, in milliseconds, to receive data from a device's response. For example, the value 1000 defines a 1 (one) second limit
	Logging options / Log to File	IO.Log.Enable	Number	Enable log generation and a log file name to generate for debugging purposes on communication. A value of 0 (zero) disables log generation and any value different from 0 (zero) enables it
Serial	IO.Log.Filename	Text	Number	Physical or virtual serial port. Possible values are 1 : COM1, 2 : COM2, 3 : COM3, ..., n : COMn
	Baud rate	IO.Serial.Baudrate	Number	Use the same value configured on the device. Default value is 9600
	Data bits	IO.Serial.DataBits	Number	Use the default value 8 (eight)
	Parity	IO.Serial.Parity	Number	Use the default value 0 : No parity
	Stop bits	IO.Serial.StopBits	Number	Use the default value 0 : 1 Stop Bit
Ethernet	Transport	IO.Ethernet.Transport	Text	Use the default value TCP

TAB	PARAMETER	OFFLINE STRING	DATA TYPE	DESCRIPTION
	Main IP	IO.Ethernet.MainIP	Text	Device's IP address, in the format [0-255].[0-255].[0-255].[0-255]
	Port	IO.Ethernet.MainPort	Number	Use the same value configured on the device. The default value is 4001
BCCProtocol	Turn on BCC protocol's character Stuffing	BCCProtocol.Character Stuffing	Number	Enables or disables the protocol's replace table application. The default value is enabled. Use the same device configuration. Valid values are 0 : Disabled or 1 : Enabled

All offline properties must be configured via PLC Tags in **String** format, by using parameters *N1* equal to -1 (minus one), *N2* equal to 0 (zero), *N3* equal to 0 (zero), and *N4* equal to 3 (three). For more details and examples, please check topic **Documentation of I/O Interfaces**.

Tag Reference

This section contains information about the configuration of Tags of B. Braun BCCProtocol Driver. All Tags are read-only.

List All Available Parameters

Read

Use a Block Tag to list all parameters available on a device, by using the configuration described on the next table.

Parameters for the List All Available Parameters Tag

B1	0 (zero)
B2	0 (zero)
B3	0 (zero)
B4	0 (zero)
Size	Depends on the number of parameters configured on a SpaceCom device . Any value greater than 0 (zero) is valid

Parameter Reading

Read

Use PLC Tags or Block Tags to read parameters from a given infusion pump. Use as many Tags as needed to read parameters, and there must be one Tag per parameter. This Tag returns a reading error if the selected parameter does not have the same name used by the device when communicating.

Syntactic and size parameters

Device	Address of an infusion pump
Item	Parameter's name
Size	2 (two) if it is a Block Tag

If users want to omit the **Device** field, the expected syntax for the **Item** field is the following:

```
<Infusion pump address>:<Parameter's name>
```

Using a Block Tag

BLOCK ELEMENT	VALUE
Element 1	Time since a SpaceCom device was turned on, in seconds
Element 2	Parameter's value. The data type of this value depends on which parameter it refers to

Using a PLC Tag

The PLC Tag's **Value** property contains the value of the parameter, and its data type depends on which parameter it refers to. Compared to a Block Tag, it is the same content as **Element 2**.

Data Types of Values

Data types of values, depending on which parameters they refer to, can be **DWord**, **Double**, **Date**, or **String**.

NOTES

- The *TIME* parameter contain a value in **DWord** format corresponding to the number of seconds elapsed in a day.
- The *DATE* and *DLCRETIME* parameters contain their respective values in **Date** format.
- A device can provide a parameter without a value. In this case, the **Value** property of a PLC Tag or the Element 2 of a Block Tag display an empty value.

Tables for Interpreting Values

The next tables contain the interpretation corresponding to the values of parameters. If a parameter is not described, please check the technical manual provided by B. Braun.

Units

Interpretation of values corresponding to units in INDCONU and INDORTU parameters

VALUE	UNIT
1	ml/h
2	ml

VALUE	UNIT
3	hh:mm
4	min:sec
5	Years
6	mm
7	nl
8	degree (angle)
9	N
10	mbar
11	square mm
12	mA
13	ms
14	Unixtime
15	Percent
16	mV
17	degree Celsius
18	mAh
19	m
20	BMI acc to DUBOIS
21	BMI acc. to Boyd
22	kg
23	ng
24	mg
25	micro gram (μg)
26	g
27	mmol
28	mEq
29	IU
30	ng/ml
31	mg/ml
32	ug/ml
33	g/ml
34	mmol/ml
35	mEq/ml
36	IE/ml
37	ng/kg
38	mg/kg
39	micro gram/kg

VALUE	UNIT
40	g/kg
41	mmol/kg
42	mEq/kg
43	IU/kg
44	ng/minute
45	ng/hour
46	ng/day
47	mg/minute
48	mg/ hour
49	mg/day
50	micro gram/minute
51	micro gram/hour
52	micro gram/day
53	gram/minute
54	gram/hour
55	gram/day
56	mmol/minute
57	mmol/hou
58	mmol/day
59	mEq/minute
60	mEq/hour
61	mEq/day
62	IU/minute
63	IU/hour
64	IU/day
65	ng/kg/minute
66	ng/kg/ hour
67	ng/kg/day
68	mg/kg/minute
69	mg/kg/ hour
70	mg/kg/day
71	micro gram/kg/minute
72	micro gram/kg/hour
73	micro gram/kg/day
74	gram/kg/minute
75	gram/kg/hour
76	gram/kg/day

VALUE	UNIT
77	mmol/kg/minute
78	mmol/kg/hour
79	mmol/kg/day
80	mEq/kg/minute
81	mEq/kg/ hour
82	mEq/kg/day
83	IU/kg/minute
84	IU/kg/ hour
85	IU/kg/day
86	mmHg
87	kPa
88	lbs
89	mmol/liter
90	mg/dl
91	g(CH)
92	g(CH)
93	g(CH)
94	hour
95	cm
96	inch
97	ng/m ²
98	ng/m ² /minute
99	ng/m ² /hour
100	ng/m ² /day
101	mcg/m ²
102	mcg/m ² /minute
103	mcg/m ² /hour
104	mcg/m ² /day
105	mg/m ²
106	mg/m ² /minute
107	mg/m ² /hour
108	mg/m ² /day
109	g/m ²
110	g/m ² /minute
111	g/m ² /hour
112	g/m ² /day
113	IU/m ²

VALUE	UNIT
114	IU/m ² /minute
115	IU/m ² /hour
116	IU/m ² /day
117	mIU
118	mIU/ml
119	mIU/kg
120	mIU/m ²
121	mIU/minute
122	mIU/hour
123	mIU/day
124	mIU/kg/min
125	mIU/kg/hour
126	mIU//kg/day
127	mIU/m ² /minute
128	mIU/m ² /hour
129	mIU/m ² /day
130	kIU
131	kIU/ml
132	kIU/kg
133	kIU/hour
134	kIU/day
135	kIU/kg/hour
136	kIU/kg/day
137	MIU
138	MIU/ml
139	MIU/kg
140	MIU/hour
141	MIU/day
142	MIU/kg/hour
143	MIU/kg/day
144	mEq/m ²
145	mEq/m ² /minute
146	mEq/m ² /hour
147	mEq/m ² /day
148	kcal
149	kcal/kg
150	kcal/ml

VALUE	UNIT
151	kcal/day
152	kcal/kg/day
153	ml/kg
154	ml/ml
155	ml/kg/hour
156	Mbps

Alarms

All parameters described on the next tables are **DWord** data types and their bits can be accessed separately.

ALARM1

BIT	DESCRIPTION
0	No alarm
1	Calibration data corrupt
2	Battery alarm
3	Therapy data corrupt
4	Therapy data and pump settings corrupt
5	Battery voltage too low
6	Battery not in pump
7	Battery flat at power on
8	Battery cover open
9	Syringe empty
10	Pressure too high
11	Syringe drive blocked
12	Standby time expired
13	Volume infused
14	Infusion time expired
15	Claw malfunction
16	Pressure sensor defect
17	Syringe holder open
18	Syringe not inserted correct
19	KVO end
20	Pressure alarm upstream sensor
21	Dummy
22	Drop alarm, summary of all drop alarms
23	Drop alarm: no drops
24	Drop alarm: too less drops

BIT	DESCRIPTION
25	Drop alarm: too many drops
26	Drop alarm: Free flow
27	Drop alarm: no drop sensor con. but required
28	Air alarm, summary of all air alarms
29	Air alarm: Air bubble too large > 0,3 ml
30	Air alarm: Air rate exceeded > 4 ml/h
31	Air alarm: sensor test faulty

ALARM2

BIT	DESCRIPTION
0	Dummy
1	Data-lock unauthorized manipulation
2	X hour limit is reached, only for PCA operation
3	SGC-End alarm

STATUS1

BIT	DESCRIPTION
0	1: Pump switched on
1	1: Mains power operation
2	1: Standby active
3	1: Start-up menu at pump active
4	1: Prime active
5	1: Drive in parking position
6	1: Pump is ready to start
7	1: Pump is running or infusing
8	1: Therapy has started
9	1: Ready to start bolus
10	1: Manual bolus is active
11	1: Volume bolus is active
12	1: KVO function released
13	1: KVO active
14	1: Data lock active
15	1: Pump is operating in dose mode
16	1: Dose bolus with body weight
17	1: Patient data stored in pump

BIT	DESCRIPTION
18	1: Manual operation at pump
19	1: act. Alarm is quit
20	1: act. pre alarm is quit
21	1: Space-Control is connec. to pump
22	1: Air sensor is switched off
23	0: Piggyback active or 1: primary active
24	1: Editor at pump is active, that is, rate is edited
25	1: intermittent bolus is active
26	1: minimum one alarm is active

PREALARM

BIT	DESCRIPTION
0	No pre alarm
1	Disposable not inserted
2	Volume to be delivered near end
3	Time near end
4	Battery near flat
5	KVO operation
6	Data lock unauthorized manipulation
7	Incompatible CAN-BUS device detected
8	Piggyback: secondary volume infused
9	SGC pre alarm

Errors

Definitions of the LASTAUTOPROGERR parameter

VALUE	DESCRIPTION
0	OK, no error detected
1	Drug or concentration was not found in drug library
100	One of the passed parameters was not valid
101	<i>Longname</i> mandatory but not found
102	Catalog name mandatory but not found
103	Given <i>DrugName</i> did not match with given <i>DrugId</i>
104	Concentration and a non concentration assigned to drug
105	More than one concentration assigned to a drug and no search criteria informed
106	No concentration found but concentration required

VALUE	DESCRIPTION
107	Given concentration does not match
108	Inconsistent concentration input
109	Given short name does not match
110	Pump type does not match
111	Ward invalid
112	Patient profile invalid
113	Therapy CONT, unit error
114	Therapy DOT, unit error
115	Found drug with Therapy CONT not activated
116	Found drug with Therapy DOT not activated
117	Therapy not supported
118	Therapy filter setting error
119	Given concentration assigned to non concentration
120	Not allowed for Secondary mode
121	Fatal error
1000	Syntax error or CRC fail in received dataset
1001	No valid dataset in proposal
1002	Lookup returned not found, but lookup always mode is active
1003	Drug is not allowed to be a primary infusion
1004	User did not confirm that no lookup was performed (cancel)
1005	The amount of matched Longnames is not 1 (one). Indirect addressing
1006	The amount of datasets is not 1 (one). Direct addressing
1007	The received concentration of a second proposal is not equal the concentration of the primary infusion
1008	Primary infusion is not continuous therapy
1009	Rate Unit of received dataset is not equal to rate unit of primary
1010	Received rate is out of limits
1011	Received VTBI is out of limits
1012	Secondary infusion is indirect addressed
1013	Wrong pump type. Secondary infusion is only valid for infusomat
1014	Drug is not allowed as secondary infusion
1015	Primary therapy is not continuous therapy
1016	Primary does not allow a secondary infusion

VALUE	DESCRIPTION
1017	User did not confirm the secondary infusion to be configured
1018	Received dataset did not contain a rate or a unit
1024	<i>Longname</i> found Catalog ID not found in Online Parameter Change
1025	New Drug Abound or new Container Size mismatch in Online Parameter Change
1026	Invalid new Parameters in Online Parameter Change
1027	Identical values sent for Online Parameter Change

Documentation of I/O Interfaces

This section contains the documentation of I/O Interfaces referring to the **BCCProtocol** Driver.

Configuration of a Driver

I/O Interface configuration is performed on a Driver's configuration dialog box. To access the configuration of this dialog box in **Eclipse E3** in version 1.0, follow these steps:

1. Right-click a Driver object (IODriver).
2. Select the **Properties** item on the contextual menu.
3. Select the **Driver** tab.
4. Click **Other parameters**.

In **Eclipse E3** version 2.0 or later, click **Configure driver**  on a Driver's toolbar. In **Eclipse SCADA**, follow these steps:

1. Open the Organizer.
2. Select a Driver on Organizer's tree.
3. Click **Extras** on the **Driver** tab.

Currently, an I/O Interface allows opening only one connection for each Driver. This means that, if users want to access two serial ports, they must add two Drivers to an application and then configure each one of these Drivers for each serial port.

Configuration Dialog Box

The dialog box of I/O Interfaces allows configuring the I/O connection used by a Driver. This dialog box contains the **Setup**, **Serial**, **Ethernet**, **Modem**, and **RAS** tabs, described on the next topics. If a Driver does not implement a specific I/O connection, its corresponding tab is not available for configuration. Some Drivers may contain additional tabs, specific for that Driver, on the configuration dialog box.

Setup Tab

The **Setup** tab contains general configurations of a Driver. This tab is divided into the following groups:

- **General configurations:** Configurations of a Driver's physical layer, time-out, and initialization mode

- **Connection management:** Configurations on how the I/O Interface keeps a connection and which recovery policy is used on failure
- **Logging options:** Controls the generation of log files

The screenshot shows the 'Setup' tab of a configuration window. At the top, there's a 'Physical Layer' dropdown menu set to 'Ethernet' and a checkbox for 'Start driver OFFLINE' which is unchecked. Below this are two input fields: 'Timeout' set to '1000 ms' and 'Communication check time' set to '5000 ms'. A section titled 'Connection management' contains a 'Mode' dropdown set to 'Automatic (managed by the driver)'. It also has three options: 'Retry failed connection every 20 seconds' (checked), 'Give up after 1 failed retries' (unchecked), and 'Disconnect if non-responsive for 0 seconds' (unchecked). A 'Logging Options' section at the bottom has a 'Log to File' text box containing 'C:\eeLogs\MicrolokII_%DATE%.log' and a 'File size limit (MB)' input set to '0' with a note that '0' is unlimited.

Setup tab

General options on the Setup tab

OPTION	DESCRIPTION
Physical Layer	Select the physical layer on a list. Available options are Serial, Ethernet, Modem, and RAS . The selected interface must be configured on its specific tab
Timeout	Configure a time-out, in milliseconds, for the physical layer. This is the amount of time an I/O interface waits to receive any byte from the reception's buffer
Communication check time	Set the time, in milliseconds, to define the interval at which communication is considered to be in an inactive state. As long as an I/O Driver receives valid data, its communication state is considered active. However, if during operation an I/O Driver does not receive valid data inside this period of time, the state is considered inactive. The communication state is shown in the IO.CommunicationStatus Tag
Start driver OFFLINE	Select this option so that a Driver starts in Offline mode or stopped. This means that the I/O interface is not created until this Driver is configured to Online mode by using a Tag in an application. This mode enables a dynamic configuration of an I/O interface at run time

Options on the Connection management group

OPTION	DESCRIPTION
Mode	Selects a management mode of a connection. Selecting the Automatic option allows a Driver to manage the connection automatically, as specified in the next options. Selecting the Manual option allows an application to fully manage a connection
Retry failed connection every ... seconds	Select this option to enable a Driver's connection retry in a certain interval, in seconds. If the Give up after failed retries option is not selected, this Driver keeps retrying until a connection is performed, or until the application is stopped
Give up after ... failed retries	Enable this option to define a maximum number of connection retries. When the specified number of consecutive connection retries is reached, a Driver goes to the Offline mode, assuming that a hardware problem was detected. If a Driver establishes a successful connection, the number of unsuccessful retries is cleared. If this new connection is lost, then the retry counter starts at zero
Disconnect if non-responsive for ... seconds	Enable this option to force a Driver to disconnect if no byte was received by the I/O interface during the specified time-out, in seconds. This time-out must be greater than the time-out configured in the Timeout option

Options on the Logging Options group

OPTION	DESCRIPTION
Log to File	<p>Enable this option and configure the name of a file to write a log. Log files can be large, so use this option for short periods of time, only for testing and debugging purposes. If the %PROCESS% macro is used in the log file name, it is replaced by the identifier of the current process. This option is particularly useful when using several instances of the same Driver in Elipse E3, thus allowing each instance to generate a separate log file. For example, when configuring this option with value "c:\e3logs\drivers\sim_%PROCESS%.log", it generates a file named c:\e3logs\drivers\sim_00000FDA.log for process OFDAh. Users can also use the %DATE% macro in the file name. In this case a log file is generated every day, in the format aaaa_mm_dd. For example, when configuring this option with value "c:\e3logs\drivers\sim_%DATE%.log", it generates a file named c:\e3logs\drivers\sim_2005_12_31.log in 12/31/2005 and a file named c:\e3logs\drivers\sim_2006_01_01.log in 01/01/2006. Similarly, the %DATE_HOUR% macro generates one log file per hour, in the format aaaa_mm_dd_hh</p>
File size limit (MB)	<p>Configure the log file size limit, in megabytes. A value equal to 0 (zero) means that there is no size limit for the log file</p>

Serial Tab

Use this tab to configure parameters for a **Serial** Interface.

Serial

Port:

Baud rate:

Data bits:

Parity:

Stop bits:

Enable 'ECHO' suppression

Handshaking

DTR control:

RTS control:

Wait for CTS before send

CTS timeout: ms

Delay before send: ms

Delay after send: ms

Inter-byte delay (microseconds): μ s

Inter-frame delay (milliseconds): ms

Serial tab

General options on the Serial tab

OPTION	DESCRIPTION
Port	Select a serial port on the list, from COM1 to COM4 , or type the name of a serial port in the format COMn , such as "COM15". When typing the name of a serial port manually, the dialog box only accepts names of serial ports starting with the expression "COM"
Baud rate	Select a baud rate on the list (1200, 2400, 4800, 9600, 19200, 38400, 57600, or 115200) or type a baud rate, such as 600
Data bits	Select 7 (seven) or 8 (eight) data bits on the list
Parity	Select a parity on the list. The available options are None, Even, Odd, Mark, or List
Stop bits	Select the number of stop bits on the list. The available options are 1, 1.5, or 2 stop bits
Enable 'ECHO' suppression	Enable this option to remove the echo received after the I/O Interface sends data via serial port. If this echo is not equal to the bytes just sent, then the I/O Interface aborts communication
Inter-byte delay (microseconds)	Defines a delay between each byte transmitted by the I/O Interface, in millionths of a second, that is, 1000000 is equal to a second. This option must be used with small delays of less than a millisecond
Inter-frame delay (milliseconds)	Defines a delay between packets sent or received by the I/O Interface, in thousandths of a second, that is, 1000 is equal to a second. This delay is applied if the I/O Interface

OPTION	DESCRIPTION
	sends two consecutive packets, or between a received packet and the next sending

The **Handshaking** group configures the usage of **RTS**, **CTS**, and **DTR** signals in the handshaking process, that is, it controls when data can be sent or received via serial line. Most of the time, configuring the **DTR control** option to **ON** and the **RTS control** option to **Toggle** works with **RS232**-type serial lines as well as with **RS485**-type serial lines.

Available options on the Handshaking group

OPTION	DESCRIPTION
DTR control	Select the value ON to keep the DTR signal always on while the serial port is open. Select the value OFF to turn the DTR signal off while the serial port is open. Some devices require the DTR signal always on to allow communication
RTS control	Select the value ON to keep the RTS signal always on while the serial port is open. Select the value OFF to turn the RTS signal off while the serial port is open. Select the value Toggle to turn the RTS signal on while sending bytes via serial port and turn it off when not sending bytes, therefore enabling the reception
Wait for CTS before send	Available only when the RTS control option is configured with the value Toggle . Use this option to force a Driver to check the CTS signal before sending bytes via serial port, after turning the RTS signal on. In this mode, the CTS signal is handled as a permission flag for sending
CTS timeout	Determines a maximum time, in milliseconds, that a Driver waits for the CTS signal after turning the RTS signal on. If the CTS signal is not turned on within this time-out, that Driver then fails the current communication and returns an error
Delay before send	Some serial port devices have a delay when enabling a data sending circuit after the RTS signal is turned on. Configure this option to wait a certain number of milliseconds after turning the RTS signal on and before sending the first byte. IMPORTANT : This delay must be used carefully, because it uses 100% of CPU resources while waiting. System's general performance degrades as this value increases
Delay after send	This is the same effect of the Delay before send option, but in this case the delay is performed after sending the last byte, before turning the RTS signal off

Ethernet Tab

Use this tab to configure parameters of an **Ethernet** Interface. These parameters, except port configurations, must also be configured for use in the **RAS** Interface.

Ethernet

Transport: TCP/IP ▾

PING before connecting
 Timeout: 4000 ms
 Retries: 1

Listen for connections on port: 0

Share listen port with other processes

Interface: (All Interfaces) ▾

Use IPv6 Use SSL SSL Settings

Enable 'ECHO' supression

IP Filter:

Connect to

<input type="checkbox"/> Main IP:	<input style="width: 90%;" type="text"/>	Port:	502	<input type="checkbox"/> Local port:	0
<input type="checkbox"/> Backup IP 1:	<input style="width: 90%;" type="text"/>	Port:	0	<input type="checkbox"/> Local port:	0
<input type="checkbox"/> Backup IP 2:	<input style="width: 90%;" type="text"/>	Port:	0	<input type="checkbox"/> Local port:	0
<input type="checkbox"/> Backup IP 3:	<input style="width: 90%;" type="text"/>	Port:	0	<input type="checkbox"/> Local port:	0

Ethernet tab

Available options on the Ethernet tab

OPTION	DESCRIPTION
Transport	Select the value TCP/IP for a TCP socket (<i>stream</i>) or select the value UDP/IP to use a UDP socket (<i>connectionless datagram</i>)
Listen for connections on port	Use this option to wait for new connections in a specific IP port, common in Slave Drivers. If this option remains unselected, a Driver connects to the address and port specified in the Connect to option
Share listen port with other processes	Select this option to share the listening port with other Drivers and processes
Interface	Select the local network interface, identified by its IP address, that a Driver uses to establish and receive connections, or select the value (All Interfaces) to allow connection in any network interface
Use IPv6	Select this option to force a Driver to use addresses in IPv6 format on all Ethernet connections. Leave this option deselected to use the IPv4 format
Enable 'ECHO' supression	Enable this option to remove the echo from received data. An echo is a copy of sent data, which can be returned before a reply message
IP Filter	List of restricted or allowed IP addresses from where a Driver accepts connections (<i>Firewall</i>). Please check the IO.Ethernet.IPFilter property for more information
PING before connecting	Enable this option to execute a ping command, that is, to check whether a device can be reached on a network, for a device before trying a socket connection. This is a quick way

OPTION	DESCRIPTION
	<p>of determining a successful connection before trying to open a socket with a device. The time-out of a connection with a socket can be very high. The available options are:</p> <ul style="list-style-type: none"> • Timeout: Specify the number of milliseconds to wait for a reply from a ping command. Users must use a ping command to check the normal reply time, configuring this option for a value above that average. Usually this value can be configured between 1000 and 4000 milliseconds, that is, between 1 (one) and 4 (four) seconds • Retries: Number of retries of a ping command, not counting the first attempt. If all attempts fail, then the socket connection is aborted

Available options on the Connect to group

OPTION	DESCRIPTION
Main IP	Type the IP address of a remote device. Users can use an IP address separated by dots, as well as a URL. In case of a URL, a Driver uses the available DNS service to map that URL to an IP address, such as "192.168.0.13" or "Server1"
Port	Type the IP port of a remote device, between 0 (zero) and 65535
Local port	Select this option to use a fixed local IP port when connecting to a remote device
Backup IP 1, 2, and 3	Indicate the IP address, the IP port, and the fixed local IP port of up to 3 (three) backup addresses of a remote device

General Configurations

This section contains information about the configuration of general **I/O Tags** and **Properties** of I/O Interfaces.

I/O Tags

General I/O Interfaces Tags (N2/B2 = 0)

The Tags described next are provided for all supported I/O Interfaces.

IO.CommunicationStatus

Type of Tag	I/O Tag
Type of Access	Reading
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	6 (six)
String Configuration	IO.CommunicationStatus

This Tag informs the communication status of a Driver. It indicates how communication works relative to receiving valid data within a time period arbitrated in the configuration. For more information, please check topic **Setup Tab**. Possible values are **0 - Inactive communication**: The Driver did not receive valid data or stopped receiving data after n milliseconds, as configured in the properties window, or **1 - Active communication**: The Driver is receiving valid data.

IO.IOKitEvent

Type of Tag	Block Tag
Type of Access	Read-Only
B1 Parameter	-1 (minus one)
B2 Parameter	0 (zero)
B3 Parameter	0 (zero)
B4 Parameter	1 (one)
Size Property	4 (four)
ParamItem Property	IO.IOKitEvent

This Block returns Driver events generated by several sources in I/O Interfaces. The **TimeStamp** property of this Block represents the moment this event occurred. The Block Elements are the following:

- **Element 0**: Type of event. Possible values are **0**: Information, **1**: Warning, or **2**: Error
- **Element 1**: Source of an event. Possible values are **0**: Driver (specific of a Driver), **-1**: IOKit (generic events of I/O Interfaces), **-2**: **Serial** Interface, **-3**: **Modem** Interface, **-4**: **Ethernet** Interface, or **-5**: **RAS** Interface
- **Element 2**: Error number, specific for each source of event
- **Element 3**: Message of an event, a **String** specific for each event

NOTE

A Driver keeps a maximum number of 100 events internally. If additional events are reported, older events are discarded.

IO.PhysicalLayerStatus

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	2 (two)
String Configuration	IO.PhysicalLayerStatus

This Tag indicates the status of a physical layer. Possible values are the following:

- **0**: Physical layer stopped, that is, a Driver is in **Offline** mode, the physical layer failed when initializing, or exceeded the maximum number of reconnection attempts
- **1**: Physical layer started but not connected, that is, a Driver is in **Online** mode but the physical layer is not connected. If the **Connection management** option is configured with the value **Automatic**, the physical layer can be connecting, disconnecting, or waiting for a reconnection attempt. If the **Connection management** option is configured with the value **Manual**, then the physical layer remains in this status until forced to connect
- **2**: Physical layer connected, that is, the physical layer is ready for use. This **DOES NOT** mean a device is connected, only that the access layer is working

IO.SetConfigurationParameters

Type of Tag	Block Tag
Type of Access	Read-Only
B1 Parameter	-1 (minus one)
B2 Parameter	0 (zero)
B3 Parameter	0 (zero)
B4 Parameter	3 (three)
Size Property	2 (two)
ParamItem Property	IO.SetConfigurationParameters

Use this Tag to change any property of a Driver's configuration dialog box at run time.

This Tag works only while a Driver is in **Offline** mode. To start a Driver in **Offline** mode, select the **Start driver OFFLINE** option on that Driver's configuration dialog box. Users can write to a PLC Tag or to a Block Tag containing the parameters to change. Writing individual Block Elements is not supported, the whole Block must be written at once.

In **Eclipse SCADA**, users must use a Block Tag. Every parameter to configure uses two Block Elements. For example, if users want to configure 3 (three) parameters, then the size of the Block must be 6 (six, 3×2). The first Element is the property's name, as a **String**, and the second Element is the property's value, according to the next example.

```
// 'Block' must be a Block Tag with automatic reading,
// scan reading, and automatic writings disabled.
// Configure all parameters
Block.element001 = "IO.Type" // Parameter 1
Block.element002 = "Serial"
Block.element003 = "IO.Serial.Port" // Parameter 2
Block.element004 = 1
Block.element005 = "IO.Serial.BaudRate" // Parameter 3
Block.element006 = 19200
// Writes the whole Block
Block.Write()
```

When using **Eclipse E3**, the ability to create arrays at run time allows using an I/O Tag as well as a Block Tag. Users can use the **Write** method of a Driver to send the parameters directly to that Driver, without creating a Tag, according to the next example.

```
Dim arr(6)
' Configure all array elements
arr(1) = "IO.Type"
arr(2) = "Serial"
arr(3) = "IO.Serial.Port"
arr(4) = 1
arr(5) = "IO.Serial.BaudRate"
arr(6) = 19200
' There are two methods to send parameters
' Method 1: Using an I/O Tag
tag.WriteEx arr
' Method 2: Without using a Tag
Driver.Write -1, 0, 0, 3, arr
```

A variation of the previous example uses a bidimensional array.

```
Dim arr(10)
' Configure all array elements. Notice the array was resized
' to 10 elements. Empty array elements are ignored by a Driver
arr(1) = Array("IO.Type", "Serial")
arr(2) = Array("IO.Serial.Port", 1)
arr(3) = Array("IO.Serial.BaudRate", 19200)
Driver.Write -1, 0, 0, 3, arr
```

A Driver does not validate parameter names or passed values, therefore be careful when writing parameters and values. The **Write** method fails if the configuration array is incorrectly created. Users can check the log of a Driver or use the *writeStatus* parameter of the **WriteEx** method to find out the exact cause of an error.

```
Dim arr(10), strError
arr(1) = Array("IO.Type", "Serial")
arr(2) = Array("IO.Serial.Port", 1)
arr(3) = Array("IO.Serial.BaudRate", 19200)
If Not Driver.WriteEx -1, 0, 0, 3, arr, , , strError Then
    MsgBox "Failed configuring Driver parameters: " + strError
End If
```

IO.WorkOnline

Type of Tag	I/O Tag
Type of Access	Reading or Writing
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	4 (four)
String Configuration	IO.WorkOnline

This Tag informs the current status of a Driver and allows starting or stopping the physical layer. Possible values are the following:

- **0 - Driver Offline:** Physical layer is closed or stopped. This mode allows a dynamic configuration of a Driver's parameters using the **IO.SetConfigurationParameters** Tag
- **1 - Driver Online:** Physical layer is open or executing. While in **Online** mode, the physical layer can be connected or disconnected and its current status can be checked using the **IO.PhysicalLayerStatus** Tag

In the next example, using **Eclipse E3**, a Driver is configured to **Offline** mode, its COM port is changed, and then configured to **Online** mode again.

```
'Configure to Offline mode
Driver.Write -1, 0, 0, 4, 0
'Change port to COM2
Driver.Write -1, 0, 0, 3, Array("IO.Serial.Port", 2)
'Configure to Online mode
Driver.Write -1, 0, 0, 4, 1
```

The **Write** method may fail when configuring a Driver to **Online** mode, that is, writing the value 1 (one). In this case, this Driver remains in **Offline** mode. The cause of failure can be:

- Type of physical layer incorrectly configured, probably an invalid value was configured in the **IO.Type** property
- This Driver may have run out of memory
- Physical layer probably did not create its working thread. Search the log file for a message "Failed to create physical layer thread!"
- Physical layer could not start. The cause of this failure depends on the type of physical layer. It can be an invalid serial port number, a failure when starting Windows Sockets, or a failure when starting TAPI (modem), among others. This cause is recorded on the log file

IMPORTANT

Even if the configuration of a Driver to **Online** mode is successful, this does not necessarily mean the physical layer is ready to use, that is, ready to execute input and output operations with an external device. The **IO.PhysicalLayerStatus** Tag must be checked to ensure the physical layer is connected and ready for communication.

Properties

These are general properties of all supported I/O Interfaces.

IO.ConnectionMode

9 Controls the management mode of a Connection. Possible values are **0**: Automatic mode, in which a Driver manages the connection or **1**: Manual mode, in which an application manages the connection.

IO.GiveUpEnable

When configured to True, defines a maximum number of reconnection attempts. If all reconnection attempts fail, a Driver enters the **Offline** mode. When configured to False, a Driver tries until a reconnection is successful.

IO.GiveUpTries

9 Number of reconnection attempts before this one is aborted. For example, if the value of this property is equal to 1 (one), a Driver tries only one reconnection when the connection is lost. If this one fails, this Driver enters the **Offline** mode.

IO.InactivityEnable

Configure to True to enable and to False to disable inactivity detection. The physical layer is disconnected if inactive for a certain period of time. The physical layer is considered inactive only if it is capable of sending data but not capable of receiving it back.

IO.InactivityPeriodSec

9 Number of seconds to check for inactivity. If the physical layer is inactive for this period of time, it is then disconnected.

IO.RecoverEnable

☑ Configure to True to enable a Driver to recover lost connections and to False to leave a Driver in **Offline** mode when a connection is lost.

IO.RecoverPeriodSec

9 Delay time between two connection attempts, in seconds.

NOTE

The first reconnection is executed immediately after a connection is lost.

IO.StartOffline

☑ Configure to True to start a Driver in **Offline** mode and to False to start a Driver in **Online** mode.

NOTE

It is pointless to change this property at run time, as it can only be changed when a Driver is already in **Offline** mode. To configure a Driver in **Online** mode at run time, write the value 1 (one) to the **IO.WorkOnline** Tag.

IO.TimeoutMs

9 Defines a time-out for the physical layer, in milliseconds. One second is equal to 1000 milliseconds.

IO.Type

A Defines the type of physical interface used by a Driver. Possible values are the following:

- **N or None:** Does not use a physical interface, that is, a Driver must provide a customized interface
- **S or Serial:** Uses a local serial port (COMn)
- **M or Modem:** Uses a local modem, internal or external, accessed via TAPI (*Telephony Application Programming Interface*)
- **E or Ethernet:** Uses a TCP/IP or UDP/IP socket
- **R or RAS:** Uses a **RAS** (*Remote Access Server*) Interface. A Driver connects to a RAS device using the **Ethernet** Interface and then sends an **AT** (*dial*) command

Statistical Configuration

This section contains information about the configuration of **I/O Tags** and **Properties** of I/O Interfaces statistics.

I/O Tags

Tags of I/O Interface Statistics (N2/B2 = 0)

The Tags described next display statistics for all I/O Interfaces.

IO.Stats.Partial.BytesRecv

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1101
Configuration by String	IO.Stats.Partial.BytesRecv

This Tag returns the number of bytes received in the current connection.

IO.Stats.Partial.BytesSent

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1100
Configuration by String	IO.Stats.Partial.BytesSent

This Tag returns the number of bytes sent through the current connection.

IO.Stats.Partial.TimeConnectedSeconds

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1102
Configuration by String	IO.Stats.Partial.TimeConnectedSeconds

This Tag returns the number of seconds a Driver is connected in the current connection or 0 (zero) if a Driver is disconnected.

IO.Stats.Partial.TimeDisconnectedSeconds

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1103
Configuration by String	IO.Stats.Partial.TimeDisconnectedSeconds

This Tag returns the number of seconds a Driver is disconnected since the last connection ended or 0 (zero) if a Driver is connected.

IO.Stats.Total.BytesRecv

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1001
Configuration by String	IO.Stats.Total.BytesRecv

This Tag returns the number of bytes received since a Driver was loaded.

IO.Stats.Total.BytesSent

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1000
Configuration by String	IO.Stats.Total.BytesSent

This Tag returns the number of bytes sent since a Driver was loaded.

IO.Stats.Total.ConnectionCount

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1004
Configuration by String	IO.Stats.Total.ConnectionCount

This Tag returns the number of connections a Driver already established, successfully, since it was loaded.

IO.Stats.Total.TimeConnectedSeconds

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1002
Configuration by String	IO.Stats.Total.TimeConnectedSeconds

This Tag returns the number of seconds a Driver remained connected since it was loaded.

IO.Stats.Total.TimeDisconnectedSeconds

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1003
Configuration by String	IO.Stats.Total.TimeDisconnectedSeconds

This Tag returns the number of seconds a Driver remained disconnected since it was loaded.

Properties

Currently, there are no properties defined specifically to display I/O Interface statistics at run time.

Ethernet Interface Configuration

This section contains information about the configuration of **I/O Tags** and **Properties** of an **Ethernet** Interface.

I/O Tags

Tags of an Ethernet Interface (N2/B2 = 4)

The Tags described next allow controlling and identifying an **Ethernet** Interface at run time and they are also valid when the **RAS** Interface is selected.

IMPORTANT

These Tags are available **ONLY** while a Driver is in **Online** mode.

IO.Ethernet.IPSelect

Type of Tag	I/O Tag
Type of Access	Reading or Writing
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	4 (four)
N4 Parameter	0 (zero)
String Configuration	IO.Ethernet.IPSelect

Indicates the active IP address. Possible values are **0**: The main IP address is selected, **1**: The first alternative or backup IP address is selected, **2**: The second alternative or backup IP address is selected, or **3**: The third alternative or backup IP address is selected.

If the **Ethernet** or **RAS** Interface is connected, this Tag indicates which one of the four configured IP addresses is in use. If the Interface is disconnected, this Tag indicates which IP address is used first on the next attempt to connect.

During the connection process, if the active IP address is not available, the I/O Interface tries to connect using the other IP address. If the connection with the alternative IP address works, it is configured as the active IP address (automatic switchover).

To force a manual switchover, write values from 0 (zero) to 3 (three) to this Tag. This forces a reconnection with the specified IP address (**0**: Main address or **1, 2, 3**: Alternative address) if a Driver is currently connected. If a Driver is disconnected, this Tag configures the active IP address for the next attempt to connect.

IO.Ethernet.IPSwitch

Type of Tag	I/O Tag
Type of Access	Write-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	4 (four)
N4 Parameter	1 (one)
String Configuration	IO.Ethernet.IPSwitch

Any value written to this Tag forces a manual switchover. If the main IP address is active, then the first alternative or backup IP address is activated, and so on for all alternative IP addresses and returning to the main address until a connection is established.

If a Driver is disconnected, this Tag configures the active IP address for the next attempt to connect.

IO.Ethernet.SocketState

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	4 (four)
N4 Parameter	2 (two)
String Configuration	IO.Ethernet.SocketState

The Value property of this Tag corresponds to socket states as a map of bits:

- **Bit 0:** 0 (zero, not listening) or 1 (one, listening)
- **Bit 1:** 0 (zero, disconnected) or 1 (one, connected)

Properties

These properties control the configuration of an **Ethernet** Interface.

NOTE

The **Ethernet** Interface is also used by the **RAS** Interface.

IO.Ethernet.AcceptConnection

Configure to False if a Driver must not accept external connections, that is, if a Driver behaves as a master, or configure to True to enable the reception of connections, that is, if a Driver behaves as a slave.

IO.Ethernet.BackupEnable[2,3]

■ Configure to True to enable an alternative or backup IP address. If the reconnection attempt with the main IP address fails, a Driver tries to use an alternative IP address. Configure to False to disable its usage.

IO.Ethernet.BackupIP[2,3]

▲ Alternative or backup IP address of a remote device. Users can use a numerical address, as well as a device's host name, such as "192.168.0.7" or "SERVER2".

IO.Ethernet.BackupLocalPort[2,3]

9 Local port number to be used when connecting to an alternative IP address of a remote device. Used only if **IO.Ethernet.BackupLocalPortEnable** is equal to True.

IO.Ethernet.BackupLocalPortEnable[2,3]

■ Configure to True to force the use of a specific local port when connecting to an alternative or backup IP address or configure to False to use any available local port.

IO.Ethernet.BackupPort[2,3]

9 Port number of an alternative or backup IP address of a remote device, used with the **IO.Ethernet.BackupIP** property.

IO.Ethernet.IPFilter

▲ List with a comma-separated IPv4 or IPv6 addresses, which defines from which addresses a Driver accepts or blocks connections. Users can use asterisks, such as "192.168.*.*", or intervals, such as "192.168.0.41-50", in any part of IP addresses. To block an IP address or a range of IP addresses, use the tilde ("~") character at the beginning of the address, according to the next examples:

- **192.168.0.24**: Accepts only connections from IPv4 address 192.168.0.24
- **192.168.0.41-50**: Accepts connections from IPv4 addresses in the interval between 192.168.0.41 and 192.168.0.50
- **192.168.0.***: Accepts connections from IPv4 addresses in the interval between 192.168.0.0 and 192.168.0.255
- **fe80:3bf:877::*:* (expands to fe80:03bf:0877:0000:0000:0000:0000:0000:*)**: Accepts connections from IPv6 addresses in the interval between fe80:03bf:0877:0000:0000:0000:0000:0000 and fe80:03bf:0877:0000:0000:0000:ffff:ffff
- **192.168.0.10, 192.168.0.15, 192.168.0.20**: Accepts connections from IPv4 addresses 192.168.0.10, 192.168.0.15, and 192.168.0.20
- **~192.168.0.95, 192.168.0.***: Accepts connections from IPv4 addresses in the interval between 192.168.0.0 and 192.168.0.255, except the IPv4 address 192.168.0.95

When a Driver receives a connection attempt, the list of filters is scanned sequentially from left to right, searching for a specific authorization or block for the IP address where the connection comes from. If no element on the list corresponds to the IP address, the authorization or block are dictated by the last element of that list:

- If the last element on the list is an authorization, such as "192.168.0.24", then all IP addresses not found on the list are blocked
- If the last element on the list is a block, such as "~192.168.0.24", then all IP addresses not found on the list are authorized

If an IP address appears on more than one filter on the list, the leftmost filter has precedence. For example, in case of "~192.168.0.95, 192.168.0.*", the IP address 192.168.0.95 fits both rules, but the rule that wins is the leftmost one, "~192.168.0.95", and therefore this IP address is blocked.

When **IOKit** blocks a connection, it logs a message "Blocked incoming socket connection from {IP}!".

In case of UDP connections in broadcast listening mode, in which a Driver can receive packets from different IP addresses, blocks or permissions are performed at each packet received. If a packet is received from a blocked IP address, it logs a message "Blocked incoming packet from {IP} (discarding {N} bytes)!".

IO.Ethernet.ListenIP

A IP address of the local network interface that a Driver uses to establish and accept connections. Leave this property empty to establish and accepts connections using any local network interface.

IO.Ethernet.ListenPort

9 Number of the IP port used by a Driver to listen to connections.

IO.Ethernet.MainIP

A IP address of a remote device. Users can use a numerical address, as well as a device's host name, such as "192.168.0.7" or "SERVER2".

IO.Ethernet.MainLocalPort

9 Local port number to use when connecting to the main IP address of a remote device. This value is only used if the **IO.Ethernet.MainLocalPortEnable** property is equal to True.

IO.Ethernet.MainLocalPortEnable

☑ Configure to True to force the use of a specific local port when connecting to the main IP address of a remote device or configure to False to use any available local port.

IO.Ethernet.MainPort

9 Number of the IP port of a remote device, used with the **IO.Ethernet.MainIP** property.

IO.Ethernet.PingEnable

☑ Configure to True to enable sending a **ping** command to the IP address of a remote device, before trying to connect to the socket. This socket's connection time-out cannot be controlled, therefore sending a **ping** command before connecting is a fast way to detect if the connection is going to fail. Configure to False to disable a **ping** command.

IO.Ethernet.PingTimeoutMs

9 Delay time to wait for a response from a **ping** command, in milliseconds.

IO.Ethernet.PingTries

9 Maximum number of attempts of a **ping** command. Minimum value is 1 (one), including the first **ping** command.

IO.Ethernet.ShareListenPort

☑ Configure to **True** to share a listening port with other Drivers and processes or **False** to open a listening port in exclusive mode. To successfully share a listening port, all Drivers and processes that use that port must open it in shared mode. When a listening port is shared, each incoming connection is distributed to one of the processes listening. This way, if a Slave Driver only supports one connection at a time, users can use several instances of this Driver listening on the same port, therefore simulating a Driver with support for multiple connections.

IO.Ethernet.SupressEcho

☑ Configure to **True** to eliminate echoes in communication. An echo is the unwanted reception of an exact copy of all data packets a Driver sent to a device.

IO.Ethernet.Transport

⚠ Defines a transport protocol. Possible values are **T or TCP**: Uses the TCP/IP protocol or **U or UDP**: Uses the UDP/IP protocol.

IO.Ethernet.UseIPv6

☑ Configure to **True** to use IPv6 addresses on all Ethernet connections or configure to **False** to use IPv4 addresses (default).

Serial Interface Configuration

This section contains information about the configuration of **I/O Tags** and **Properties** of a **Serial** Interface.

I/O Tags

Tags of a Serial Interface (N2/B2 = 2)

Currently, there are no Tags defined specifically to manage a **Serial** Interface at run time.

Properties

These properties control the configuration of a **Serial** Interface.

IO.Serial.Baudrate

9 Specifies a baud rate of a serial port, such as 9600.

IO.Serial.CTSTimeoutMs

9 Time to wait for a **CTS** signal, in milliseconds. After turning the **RTS** signal on, a timer is started to wait for a **CTS** signal. If this timer expires, a Driver aborts sending bytes through the serial port. Available only when the **IO.Serial.RTS** property is configured with the value **Toggle** and the **IO.Serial.WaitCTS** property is configured to **True**.

IO.Serial.DataBits

9 Specifies the number of data bits to configure a serial port. Possible values are **5**: Five data bits, **6**: Six data bits, **7**: Seven data bits, or **8**: Eight data bits.

IO.Serial.DelayAfterMs

9 Number of milliseconds to delay after the last byte is sent through a serial port, but before turning the **RTS** signal off. Available only when the **IO.Serial.RTS** property is configured with the value **Toggle** and the **IO.Serial.WaitCTS** property is configured to **False**.

IO.Serial.DelayBeforeMs

9 Number of milliseconds to delay after turning the **RTS** signal on, but before data is sent. Available only when the **IO.Serial.RTS** property is configured with the value **Toggle** and the **IO.Serial.WaitCTS** property is configured to **False**.

IO.Serial.DTR

A Indicates how a Driver deals with the **DTR** signal. Possible values are **OFF**: **DTR** signal is always turned off or **ON**: **DTR** signal is always turned on.

IO.Serial.InterbyteDelayUs

9 Delay time, in milliseconds (1/1000000 of a second), for each two bytes sent through a **Serial** Interface.

IO.Serial.InterframeDelayMs

9 Delay time, in milliseconds, before sending a packet after the last packet sent or received.

IO.Serial.Parity

A Specifies a parity for the configuration of a serial port. Possible values are **E or Even**: Even parity, **N or None**: No parity, **O or Odd**: Odd parity, **M or Mark**: Mark parity, or **S or Space**: Space parity.

IO.Serial.Port

9 Number of the local serial port. Possible values are **1**: Uses the COM1 port, **2**: Uses the COM2 port, **3**: Uses the COM3 port, or **n**: Uses the COMn port.

IO.Serial.RTS

A Indicates how a Driver deals with the **RTS** signal. Possible values are **OFF**: **RTS** signal always off, **ON**: **RTS** signal always on, or **Toggle**: Turns the **RTS** signal on when transmitting data and turns the **RTS** signal off when not transmitting data.

IO.Serial.StopBits

9 Specifies the number of stop bits for the configuration of a serial port. Possible values are **1**: One stop bit, **2**: One and a half stop bit, or **3**: Two stop bits.

IO.Serial.SuppressEcho

9 Use a value different from 0 (zero) to enable suppressing the echo or 0 (zero) to disable it.

IO.Serial.WaitCTS

☑ Configure to **True** to force a Driver to wait for the **CTS** signal before sending bytes when the **RTS** signal is turned on. Available only when the **IO.Serial.RTS** property is configured with the value **Toggle**.

Driver Revision History

VERSION	DATE	AUTHOR	COMMENTS
1.0.3	06/12/2025	M. Ludwig	<ul style="list-style-type: none">• Driver updated to IOKit library version 3.0 and Visual Studio 2022 (Case 37943).
1.0.2	06/09/2020	M. Ludwig	<ul style="list-style-type: none">• First version of this Driver.

Headquarters

**Rua Mostardeiro, 322/Cj. 902, 1001 e
1002**

90510-002 — Porto Alegre — RS

Phone: (+55 51) 3346-4699

Fax: (+55 51) 3222-6226

E-mail: elipse-rs@elipse.com.br

Branch in Taiwan

9F., No.12, Beiping 2nd St., Sanmin Dist.

807 — Kaohsiung City — Taiwan

Phone: (+886 7) 323-8468

Fax: (+886 7) 323-9656

E-mail: evan@elipse.com.br

Check our website for information about a representative in your country.

www.elipse.com.br

kb.elipse.com.br

forum.elipse.com.br

www.youtube.com/elipsesoftware

elipse@elipse.com.br



Gartner, Cool Vendors in Brazil 2014, April 2014.

Gartner does not endorse any vendor, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings. Gartner research publications consist of the opinions of Gartner's research organization and should not be construed as statements of fact. Gartner disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability of fitness for a particular purpose.

Microsoft Partner
Gold Independent Software Vendor (ISV)