

Reliance

Johnson Controls communication driver

version 1.5.4

CONTENT

- 1. Introduction 3
- 1.1. Fundamental Terms 3
- 2. Johnson Controls Communication driver 4
- 2.1 Basic characteristics 4
 - Starting the driver 4
 - Connection type 4
 - Reading the data 4
 - Device RTC synchronization 4
 - Communication log view 5
 - Diagnostics and statistics 7
 - Value simulation 7
 - Interrupted device connection 8
 - Security 8
- 2.2 menu 9
 - Menu File 9
 - Menu Communication 10
 - Menu User 10
 - Menu Settings 11
 - Menu About 11
- 2.3 Description of objects 12
 - Device 12
 - Variable 12
 - Communication port 13
- 2.4 Settings 14
 - Common 14
 - Communications 15
- 3. Appendix 17
 - 3.1 Variable value quality 17

1. INTRODUCTION

1.1. FUNDAMENTAL TERMS

<i>runtime module</i>	a program designed to run a visualization project on the end user's computer. It can be Reliance Runtime, Reliance Runtime Server or Reliance Server.
<i>design module</i>	a program designed to build a visualization project. It can be Reliance Design Desktop or Reliance Design Enterprise.
<i>device</i>	a PLC or another similar industrial automation/control device.
<i>communication driver</i>	a program designed to provide transmission of process data from the device to the runtime software and transmission of commands from the runtime software to the device.
<i>communication packet</i>	a communication message that represents a request sent from the driver to the device or an answer from the device sent to the communication driver. The structure of such a message depends on the communication protocol used by the device.

2. JOHNSON CONTROLS COMMUNICATION DRIVER

The program is designed to provide connection to Johnson Controls DX9100, SC9100 and FX15 devices.

2.1 BASIC CHARACTERISTICS

Starting the driver

The driver is a part of Reliance and it is automatically started by the runtime module. During the startup, the driver loads the visualization project and activates a connection to all the connected devices.

The communication driver can be launched separately. In that case loading the project and starting the communication should be done from the menu.

Connection type

The driver connects to JC devices through the N2 bus via an RS232/RS485 serial interface.

Reading the data

After being started the driver sends read packets to obtain values of all defined variables. Then the values are read periodically, along with the preset period of each variable. The time schedule and the holiday schedule data are read following the command from the runtime module.

Device RTC synchronization

The driver allows setting the real time clock in the device.

Communication log view

For better diagnostic there is a view of actually processed communication packets in the driver. Sent and received packets are displayed with additional information like device name and time stamp. The color of displayed packets stands to the type. Read packets are lime or green; the write packets are fuchsia or purple. The lightness of displayed packets stands to the direction. The requests are light, the responses are dark.

To activate this feature you should set up the “Show communications” parameter in the settings dialog. To show the communication log view you should select COM port or modem object in the project tree.

The screenshot displays the Johnson Controls driver software interface. The main window shows a project tree on the left with the following structure:

- Server
 - COM1
 - Johnson1 (checked)
 - RTC (checked)
 - PM2CT1 (True)
 - PM2CT2 (False)
 - PM2CT3 (False)
 - PM2CT4 (False)
 - LRS1 (False)

Below the tree, a table shows the selected object's details:

Jméno	Adresa	Hodnota
Johnson1	50	✓

The communication log on the right, titled 'Komunikační port', shows the following data:

Paket	Stanice	Délka	Čas
>ReadN	Johnson1	16	9:22:48
<ReadN	Johnson1	2	9:22:48
>ReadN	Johnson1	16	9:22:48
<ReadN	Johnson1	2	9:22:48
>ReadN	Johnson1	16	9:22:48
<ReadN	Johnson1	40	9:22:49
>ReadN	Johnson1	16	9:22:49
<ReadN	Johnson1	1	9:22:49
>ReadN	Johnson1	16	9:22:49
<ReadN	Johnson1	2	9:22:49
>ReadN	Johnson1	16	9:22:49
<ReadN	Johnson1	24	9:22:49
>Write	Johnson1	18	9:22:49
<Write	Johnson1	0	9:22:49
>ReadN	Johnson1	16	9:22:49
<ReadN	Johnson1	1	9:22:49
>ReadN	Johnson1	16	9:22:49
<ReadN	Johnson1	2	9:22:49
>ReadN	Johnson1	16	9:22:49

The status bar at the bottom indicates: Počet stanic: 1, Čtení/Zápis: 1/0, OK/Err/Timeout: 798/0/34.



Diagnostics and statistics

There is a status bar at the bottom of the driver which is useful for diagnostics too. Information according to the selected object type in the tree is shown here. In case of selected port there are counters of good and bad communications for example.

If any important or internal error has occurred, then it is written to the event log. For example the communication port is not available. You can see the event log after selecting the server object in the tree.

Value simulation

The driver allows the variable value simulation. It is helpful in cases of testing a visualization project. You can test it without the need to real values in the device. It is possible to stop data refresh from the device and set any testing value. You can choose between to generate random values or triangle waves

The screenshot shows the 'Reliance - Johnson Controls driver' window. The main area displays a tree view of objects under 'Server' and 'COM1'. The selected object is 'PM2CT3', which is a Boolean type. The configuration panel on the right shows the following settings:

Ref:	SubRef:	Typ:
PM 1	CT 3	Bool

Additional settings in the panel include:

- Adresa: 134, 2
- Interval aktualizace [ms]: 1 000
- Povolen zápis
- Povolen očetní
- Simulovat hodnotu
- Typ: Uživatelská hodnota
- Min: 0, Perioda [ms]: 1 000
- Max: 0, Max. přírůstek: 1

The status bar at the bottom shows: Johnson1.PM2CT3, Kvalita: Good, local override, Čas: 1.12.2003 9:24:19

Interrupted device connection

If no communication packet is received from the device for n-times in a sequence the driver marks the established connection to the device as interrupted. In that case “n” is the maximum count of unanswered requests that the driver resent. This parameter can be preset in the settings dialog.

Security

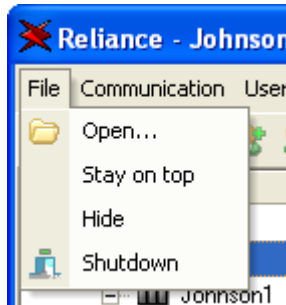
Editing the parameters in the settings dialog can be secured. If the Reliance project is secured against unauthorized shut down the runtime module, then the driver is automatically secured too. Only users with service access rights in the Reliance project are permitted to change the parameters, to stop, to start communications or to shut down the driver.

2.2 MENU

This chapter is intended for describing the menu functions. Some functions could be run from toolbar or by shortcut too. In those cases there is an icon or an appropriate shortcut displayed.

Menu File

Menu ▶ **File** contains commands used to load a visualization project, hide and shutdown the driver.



Open...

Opens the Reliance project file and loads the visualization project (*.prj).

Stay on top

Shows the driver in front of other applications.

Hide

Hides the driver to tray.

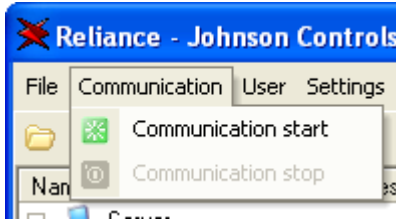


Shutdown

Shuts down the communication driver.

Menu Communication

Menu ▶ **Communication** contains commands used to start and stop communications with devices.



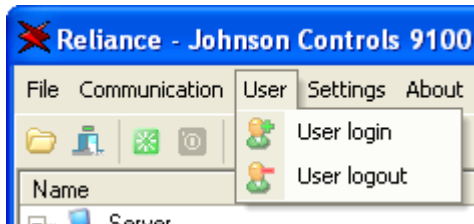
Communications start Runs communication with connected devices defined within the visualization project.



Communications stop Stops communication with connected devices.

Menu User

Menu ▶ **User** contains commands used to login or to logout a user defined in visualization project.



User login Logs in a user defined in visualization project.



User logout Logs off a user defined in visualization project.

Menu Settings

Activates the edit dialog with communication driver parameters. Detailed description of these parameters could be found in chapter 2.4 SETTINGS.

Menu About

Displays information about the communication driver. The most useful information is the version.

2.3 DESCRIPTION OF OBJECTS

Device

Device represents the Johnson Controls hardware device which the driver is connecting to.

<i>Name</i>	Device name unique within the visualization project.
<i>Comment</i>	Optional comment.
<i>Connection type</i>	Type of connection to device (direct via serial port, modem, ethernet ...).
<i>Address</i>	HW device address.
<i>Com port</i>	Serial port number.
<i>Baud</i>	Communication speed.
<i>Timeout [ms]</i>	Time limit in which the device should answer on driver request.

Variable

Variable is the main object carrying the data information.

<i>Name</i>	Variable name unique within the device.
<i>Comment</i>	Optional comment.
<i>Ref</i>	Reference to programmable block in the device.
<i>SubRef</i>	Reference to the subordinated block, within another programmable block in the device.
<i>Type</i>	Value data type.
<i>Address</i>	Variable physical address in the device and bit order of the binary variable.
<i>Update rate</i>	Update interval to refresh the value.

<i>Writeable</i>	Flag which allows or disallows writing a new value to the device.
<i>Readable</i>	Flag which allows or disallows reading of the value from the device.
<i>Simulate value</i>	Flag to simulate the value and the relevant parameters.

Communication port

The standard serial communication port is used for connection to N2 bus via the RS232/RS485 converter.

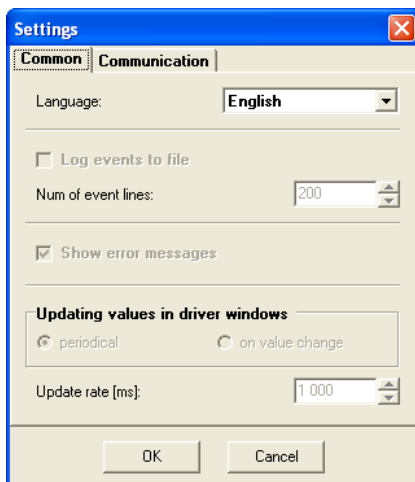
Communication port The COM port number.

2.4 SETTINGS

This dialog could be opened from the ► **Settings** menu. It contains general parameters and some communication parameters. These parameters are saved to the initialization file *Johnson_Serv.ini*.

Common

<i>Language</i>	Language selection (English, Czech).
<i>Log events to file</i>	Activates event logging to text file.
<i>Num of event lines</i>	Highest number of line count displayed in event log.
<i>Show error messages</i>	Activates or suppresses the appearance of error messages. It is recommended to deactivate this flag.
<i>Updating values in driver windows</i>	Mode of refreshing values in the driver forms. <u>It has no effect on reading values from the device and refreshing values in the runtime module.</u> For common use periodical mode is recommended.
<i>Update rate</i>	Defines refresh period in case of periodical mode of refreshing. <u>Preset value has no effect on reading values from device and refreshing values in runtime module.</u>



Communication

Communication with devices is based on the request – answer principle.

Max. communication error count

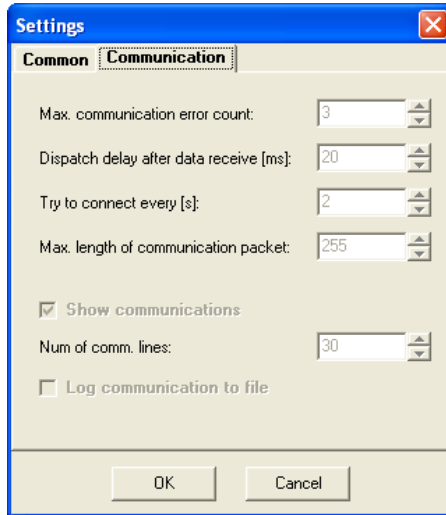
In case no answer or a damaged answer is received the request is repeated. If the amount of the repeated consecutive requests reaches this preset number, the request is no longer repeated. The device communication status is labeled as bad. The quality value of all variables in this device becomes „Bad, comm. failure“.

Dispatch delay after data receive [ms]

Delay between the accepting last byte of the device response and the next request.

Try to connect every[s]

If the connection to the device has failed the driver tries to reconnect to the device in a preset period.



Max. length of communication packet

The driver generates communication packets that do not exceed preset value. It is suitable to decrease the value in case of high level interferences. This value should be changed in some situations where radio or cable TV modems are used as a type of transmission medium.

Show communication

Activates communication log display after selecting the modem or com port object in the tree. This function is switched off by default.

Num. of comm. lines
log list

Number of lines displayed in the communication

Log communication to file

This option activates the function to log communication packets to a text file. This file can be used at a later time for analysis.

3. APPENDIX

3.1 Variable value quality

Quality	Extended	Meaning
<i>Bad</i>	<i>comm failure</i>	Value is not valid; driver could not establish a connection with the device.
	<i>last known value</i>	Time stamp expired; the value is the last known valid value, connection with the device is interrupted.
	<i>configuration error</i>	Value is not valid, a configuration error has occurred; for example a variable with this name is not available.
	<i>out of service</i>	Value is not valid; variable is not read out from the device; for example the variable is not active.
<i>Uncertain</i>	<i>non-specific</i>	Unspecified value, the variable was not yet read out from the device.
<i>Good</i>	<i>non-specific</i>	Value is valid.
	<i>local override</i>	Value is valid; the value is not a real value from the device. The value is simulated.