

# **Reliance**

**SAUTER EY2400 communication driver**

**Version 2.4.3**

# CONTENT

- 1. Introduction ..... 4
- 1.1. Fundamental Terms ..... 4
- 2. Sauter EY2400 Communication driver ..... 5
- 2.1 Basic characteristics ..... 5
  - Starting the driver ..... 5
  - Connection type ..... 5
  - Connection establishment - Life Test ..... 5
  - Device initialization ..... 6
  - Reading the data ..... 6
  - Spontaneous messages ..... 6
  - Device connection control ..... 7
  - Dialup connection - accepting incoming call ..... 7
  - The device status and the connection status ..... 7
  - Communication log view ..... 8
  - Diagnostics and statistics ..... 9
  - Changing response category ..... 10
  - Sending a user defined message ..... 10
  - Interrupted device connection ..... 11
  - Security ..... 11
- 2.2 menu ..... 12
  - Menu File ..... 12
  - Menu Communication ..... 13
  - Menu User ..... 14
  - Menu Settings ..... 14
  - Menu About ..... 14
- 2.3 Description of objects ..... 15
  - Device 15
  - Variable ..... 15
  - Communication port ..... 16
  - Modem ..... 16
  - Folder 17
- 2.4 Settings ..... 18
  - Common ..... 18
  - Communication ..... 19
  - Diagnostics ..... 21

3. Appendix.....	22
3.1 Variable value quality.....	22
3.2 Device connection control.....	23
3.3 Device connection status .....	24

## 1. INTRODUCTION

### 1.1. FUNDAMENTAL TERMS

*runtime module* 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.

*design module* a program designed to build a visualization project. It can be Reliance Design Desktop or Reliance Design Enterprise.

*device* a PLC or another similar industrial automation/control device.

*communication driver* 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.

*communication packet* 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. SAUTER EY2400 COMMUNICATION DRIVER

The program is designed to provide connection to Sauter EY2400 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, except the devices with a variable controlled connection.

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 provides two types of hardware connection. The first is serial link via a RS232 interface and the second one is dial-up via modem. The connection type is set up in the design module.

#### Connection establishment – Life Test

The first communication packet, which is sent to the device, is called “Life Test”. This packet is used for testing the connection alive too. If no spontaneous response is obtained during the preset timeout then the Life Test packet is dispatched to prove the connection.

## **Device initialization**

The initialization is set of communication packets which set up general device parameters. These parameters take effect to the rules of sending spontaneous messages and set the device real time clock. The initialization is always proceeding after the first Life Test. This feature can be disabled by setting up the “Warm start” parameter in the Settings dialog. Then could be the Initialization processed either periodically or following the command from the runtime module.

In case of more then one variable with the same fine address, but different response category, the driver reflects the only one with the lowest response category (with the highest priority).

In case of more then one variable with the same software address and different card code the driver performs logical add of all codes and the resulting value is written down the device.

The spontaneous messaging is activated via bit mask only if the variable’s response category is lower then 3 and the value of the variable is allowed for reading.

## **Reading the data**

After the Life test or after the Initialization the driver sends read packets to obtain values of all defined variables. According to spontaneous messages these packets are not used for common data acquisition except the value is not gained in preset limit. This parameter is set up in the design module. If the value of the parameter is equal to zero then the value is not read periodically. In that case the value can be refreshed by the Update Value function from the local menu of the component Display in runtime module.

## **Spontaneous messages**

In case there are no specific requests then the driver sends common request (broadcast). If any device has a new data then it answers with the spontaneous message. There is an algorithm of automatic decrease or increase of the response category in the communication driver, to ensure the data with higher priority will be delivered in case of high amount of new values.

## **Device connection control**

In case of need it is possible to control the connection with the device. You can start or stop the communication between the driver and the device via the a variable value. The specific bit should be set to log.1 or to log.0.

There is a bit in the control variable which is useful for dial-up connection type. By setting up the bit to the log.1 the function hang-up after the reading data is activated. In that case the driver automatically hangs-up the connection after reading all data from the device. The meaning of single bits in the control variable is described in the table 3.2 in appendix.

## **Dialup connection – accepting incoming call**

The connection to the device could be established after accepting an incoming call. In this case it is better to set aside extra modem for this purposes. The modem should have set-on parameter „Only for incoming call“. This parameter could be set in the design module. The modem should not be switched on automatic answer mode. The driver handled incoming calls after the second ring itself.

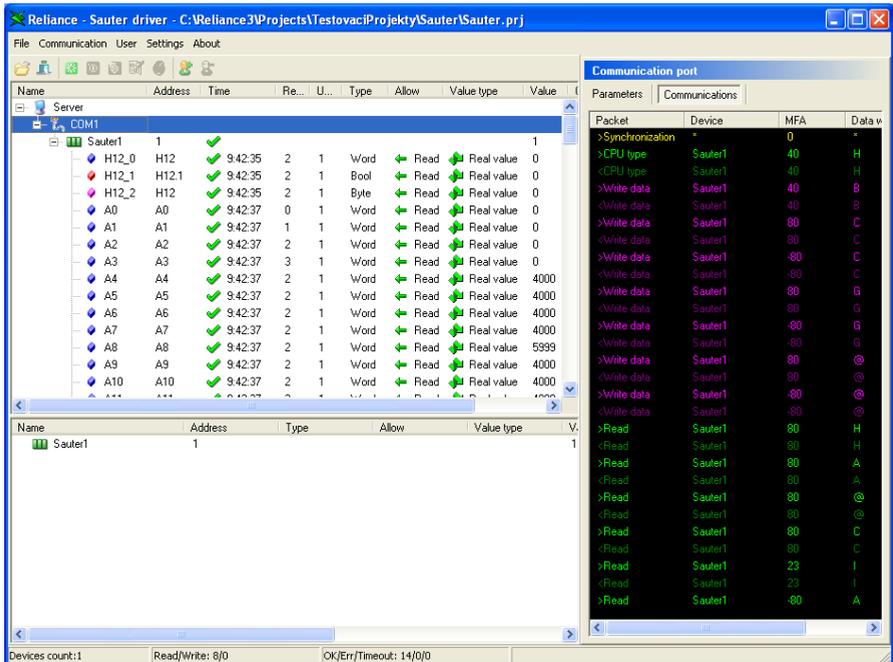
## **The device status and the connection status**

It is possible to use a variable for detecting the device status and the connection status. The variable should be defined in the design module. The meaning of each bit in the variable is described in the table 3.3 in appendix.

### Communication log view

For better diagnostics there is a view of actually processed communication packets in the driver. Sent and received packets are displayed with additional information like device name, machine fine address, data word and time stamp. The color of displayed packets stands to the type. Read packets are lime or green; write packets are fuchsia or purple; the synchronization packet is yellow. The lightness of displayed packets stands to the direction. Requests are light, 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.

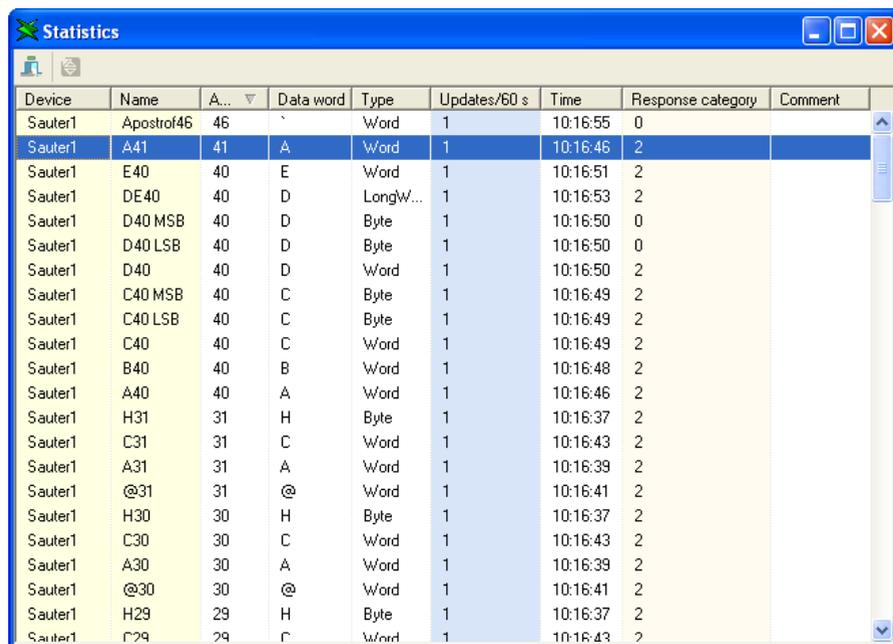


## Diagnostics and statistics

There is a status bar at the bottom of the driver which is useful for diagnostics. Information according to the selected object type in the tree is shown here. In case of selected port or modem 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.

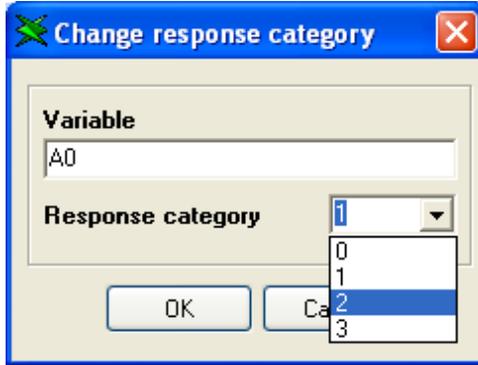
In according to rare algorithms of the Sauter communication protocol there is special functionality in the driver – the spontaneous message statistics. Each variable has own counter of delivered messages in preset interval. The count is displayed in proper column in the tree and it is possible to check it in the separate list. This is useful to determine the device and the variable which is sending messages to much frequently. The interval for the statistic could be set up in the settings dialog.



Device	Name	A...	Data word	Type	Updates/60 s	Time	Response category	Comment
Sauter1	Apostrof46	46	`	Word	1	10:16:55	0	
Sauter1	A41	41	A	Word	1	10:16:46	2	
Sauter1	E40	40	E	Word	1	10:16:51	2	
Sauter1	DE40	40	D	LongW...	1	10:16:53	2	
Sauter1	D40 MSB	40	D	Byte	1	10:16:50	0	
Sauter1	D40 LSB	40	D	Byte	1	10:16:50	0	
Sauter1	D40	40	D	Word	1	10:16:50	2	
Sauter1	C40 MSB	40	C	Byte	1	10:16:49	2	
Sauter1	C40 LSB	40	C	Byte	1	10:16:49	2	
Sauter1	C40	40	C	Word	1	10:16:49	2	
Sauter1	B40	40	B	Word	1	10:16:48	2	
Sauter1	A40	40	A	Word	1	10:16:46	2	
Sauter1	H31	31	H	Byte	1	10:16:37	2	
Sauter1	C31	31	C	Word	1	10:16:43	2	
Sauter1	A31	31	A	Word	1	10:16:39	2	
Sauter1	@31	31	@	Word	1	10:16:41	2	
Sauter1	H30	30	H	Byte	1	10:16:37	2	
Sauter1	C30	30	C	Word	1	10:16:43	2	
Sauter1	A30	30	A	Word	1	10:16:39	2	
Sauter1	@30	30	@	Word	1	10:16:41	2	
Sauter1	H29	29	H	Byte	1	10:16:37	2	
Sauter1	C29	29	C	Word	1	10:16:43	2	

### Changing response category

If there is a need to change the variable response category then it can be done from the driver too. There is a special dialog intended for this purpose. The dialog can be opened from the menu or from the toolbar.



### Sending a user defined message

This function allows you to send your own message independently on driver communications and needs of the running project. The message should be written as a string in special dialog. After the command "Send" is confirmed the message is inserted into packet list with higher priority. If the device answered then the answer occurs in the same dialog.



## **Interrupted device connection**

If no communication packet is received from the device for n-times in a sequence the driver marks the established connection with the device as interrupted. In that case “n” is the maximum count of unanswered requests that the driver resent. This parameter can 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 of 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, to shut down the driver, to change the response category and to send a user defined messages.

## 2.2 MENU

This chapter is intended for describing the menu function. 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.

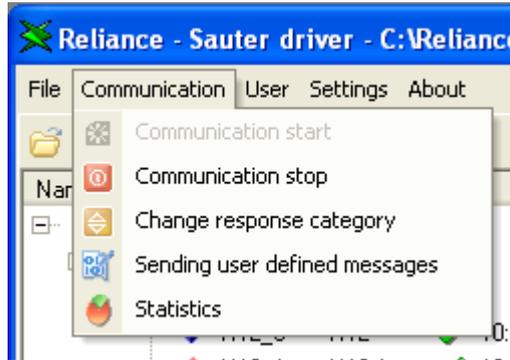


#### ***Shutdown***

Shuts down the communication driver.

## Menu Communication

Menu **Communication** contains commands used to start and stop communication with devices, to change response category, to send a user defined message and to display statistics.



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



**Communications stop** Stops communication with connected devices.



**Change response category** Changes the response category of selected variable. The function is enabled if the selected variable has got this feature.



**Sending user defined messages** Activates the dialog for sending a user defined messages.



**Statistics** Activates the dialog which shows spontaneous response statistics.

## 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

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

## Menu About

This menu displays information about the communication driver. The most useful information is the version.

## 2.3 DESCRIPTION OF OBJECTS

### Device

Device represents the Sauter EY2400 hardware device which the driver is connecting to.

<i>Name</i>	Device name unique within the visualization project.
<i>Comment</i>	Optional comment.
<i>Address</i>	HW device address.
<i>Timeout [ms]</i>	Time limit in which the device should answer on driver request.
<i>Variables count</i>	Overall count of variables defined within the device.

### Variable

Variable is the main object carrying the data information.

<i>Name</i>	Variable name is unique within the device.
<i>Comment</i>	Optional comment.
<i>Data word</i>	Data word is a part of addressing information in the device Sauter.
<i>Machine fine address</i>	Machine fine address is a part of addressing information in device Sauter.
<i>Bit num.</i>	Bit order of the binary variable.
<i>Response category</i>	Kind of priority which takes effect on sending new values from the Sauter device. The lower is response category the higher is the priority.
<i>Card code</i>	Special directive which describes HW or SW cards in the device.

<i>Type</i>	Variable value data type.
<i>Hysteresis</i>	Hysteresis defines the dead bands of the value. Changing of the value inside the dead bands causes no spontaneous response.
<i>Forced update interval</i>	If no spontaneous response was delivered during the preset time, so the value timestamp is too old, the driver reads out the value.
<i>Read</i>	Flag which allows or disallows reading of the value from the device.
<i>Write</i>	Flag which allows or disallows writing a new value to the device.

### **Communication port**

The standard serial communication port is used for connection with devices via the RS232 interface.

<i>Communication port</i>	The COM port number.
<i>Devices count</i>	Overall count of devices connected via this port.
<i>Variables count</i>	Overall count of variable read out via this port.

### **Modem**

Represents the real modem connected to the computer. It is used for dial-up connection.

<i>Communication port</i>	The communication port number via the modem is connected.
<i>Communication speed</i>	Communication speed used to communicate with the modem.
<i>Provider</i>	Represents regional dial-up or mobile provider whose service is used. Thanks this parameter the right device could be assigned to the right modem and the price of the used service could be lower.

<i>Dial timeout</i>	Defines the time in which the modem should response on dial command. In case of expiration the connection fails.
<i>Cmd timeout</i>	Defines the time in which the modem should response on “init” command. In case of expiration the connection fails.
<i>Init Cmd1</i>	1 <sup>st</sup> init command. Is sent to the modem before dialing.
<i>Init Cmd2</i>	2 <sup>nd</sup> init command. Is sent to the modem before dialing, after the first init command.
<i>Dial Cmd</i>	The command used for dialing.
<i>HangUp Cmd</i>	The command used for hanging-up.

### **Folder**

Folder is used only for organization matter.

*Name* Folder name.

## 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 *Sauter\_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. Immediately after the communication start the synchronization packet is sent. The next step could be the initialization of the device, depending on warm or cold start, and consequently values of all readable variables are read out. After that comes on spontaneous message requests. Following parameters takes effect on these communication algorithms.

*Warm start* Determines whether the device initialization is processed on the beginning of communication. Even periodical initialization is not processed when this option is set up.

*Max. reading same resp. category count* This is the highest amount of spontaneous requests with the same response category. If the count reaches this number then the response category is automatically decreased (higher priority). If the driver receives no answer on his spontaneous request the response category is increased again.

*Max. bad communication 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“.

*Periodic initialization [min]* If the value of this parameter is different from “0” then the device initialization is processed in the preset period.

*Life test period [s]* Period in which the communication packet called “Life Test” is used for establishing or verifying the

connection. The packet is sent if there is no response from the device for a long time or after the startup the communication.

*Synchronization period [s]* Period in which is the synchronization packet sent to the devices.

*Delay between the answer and the request [ms]* Delay between accepting the last byte of the response and the next request.

*Communication via ethernet* Devices are not connected directly via RS232 link. Devices are connected via virtual COM port and data are processed through the Ethernet/RS232 converter. In this case the adjusted timing is necessary. The following timeouts depends on setting of this parameter.

*First byte in response timeout [ms]* Maximal timeout between the request and the beginning of the device response. If it has expired the driver is not longer waiting for the answer and repeats the request or continues with another device.

*Last byte in response timeout [ms]* Maximal timeout between the request and the last byte of the device response. If it has expired the driver is no longer waiting for the answer and repeats the request or continues with another device.

*Short response timeout [ms]* Maximal timeout between the request and the last byte of the device response. This timeout is set-on if the length of the request is shorter then 20 bytes. If the timeout has expired the driver is not longer waiting for the answer and repeats the request or continues with another device.

*Long response timeout [ms]*

Maximal timeout between the request and the last byte of answer. This timeout is set-on if the length of the request is longer then 19 bytes. If the timeout has expired the driver is not longer waiting for the answer and repeats the request or continues with another device.

### **Diagnostics**

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

*Num. of comm. lines* Number of lines displayed in the communication log list

*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.

*Update monitoring time* This option determines the interval for the statistics of spontaneous responses.

### 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 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.

### 3.2 Device connection control

Value	Meaning
<i>\$xx01</i>	Activate connection to the device
<i>\$xx02</i>	End (hang-up) connection after reading out all data
<i>\$xx08</i>	Do initialize the device (cold start)
<i>\$xx10</i>	Update device data (warm start)
<i>\$xx20</i>	Initialize device 306 by USI

### 3.3 Device connection status

Value	Meaning
<i>\$xxx1</i>	The connection is active
<i>\$xxx2</i>	Communications run well
<i>\$xxx4</i>	A communication error
<i>\$xx1x</i>	Controlled data reading in progress
<i>\$xx2x</i>	Controlled data writing in progress
<i>\$xx4x</i>	Controlled data reading/writing finished well
<i>\$xx8x</i>	Controlled data reading/writing finished with failure
<i>\$x1xx</i>	Dial-up connection – the device is waiting for a call
<i>\$x2xx</i>	Dial-up connection – calling
<i>\$x3xx</i>	Dial-up connection – connected
<i>\$1xxx</i>	The device is supplied from battery