The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

© 2017 Schneider Electric. All Rights Reserved.
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safety Information</td>
<td>5</td>
</tr>
<tr>
<td>About the Book</td>
<td>9</td>
</tr>
<tr>
<td><strong>Chapter 1</strong> Example Description</td>
<td>13</td>
</tr>
<tr>
<td>Overview</td>
<td>13</td>
</tr>
<tr>
<td><strong>Chapter 2</strong> Project Information</td>
<td>15</td>
</tr>
<tr>
<td>2.1 Hardware Configuration</td>
<td>16</td>
</tr>
<tr>
<td>Overview of the Hardware Configuration</td>
<td>16</td>
</tr>
<tr>
<td>2.2 Application</td>
<td>17</td>
</tr>
<tr>
<td>Overview of the Application</td>
<td>18</td>
</tr>
<tr>
<td>Libraries</td>
<td>18</td>
</tr>
<tr>
<td>Data Format - Application Protocol</td>
<td>19</td>
</tr>
<tr>
<td>TCP Client</td>
<td>21</td>
</tr>
<tr>
<td>TCP Server</td>
<td>22</td>
</tr>
<tr>
<td>UDP Peer</td>
<td>24</td>
</tr>
<tr>
<td>Simulation</td>
<td>25</td>
</tr>
<tr>
<td>Task Configuration</td>
<td>25</td>
</tr>
<tr>
<td>2.3 Visualization</td>
<td>26</td>
</tr>
<tr>
<td>Visualization Screens</td>
<td>26</td>
</tr>
<tr>
<td>2.4 Running the Application Example</td>
<td>30</td>
</tr>
<tr>
<td>Preconditions for Running the Application Example</td>
<td>31</td>
</tr>
<tr>
<td>Downloading the Application</td>
<td>31</td>
</tr>
<tr>
<td>Operating the Application</td>
<td>32</td>
</tr>
<tr>
<td><strong>Glossary</strong></td>
<td>33</td>
</tr>
<tr>
<td><strong>Index</strong></td>
<td>37</td>
</tr>
</tbody>
</table>
Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠️ DANGER

DANGER indicates a hazardous situation which, if not avoided, will result in death or serious injury.

⚠️ WARNING

WARNING indicates a hazardous situation which, if not avoided, could result in death or serious injury.

⚠️ CAUTION

CAUTION indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.
PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

BEFORE YOU BEGIN

Do not use this product on machinery lacking effective point-of-operation guarding. Lack of effective point-of-operation guarding on a machine can result in serious injury to the operator of that machine.

⚠️ WARNING

UNGUARDED EQUIPMENT
- Do not use this software and related automation equipment on equipment which does not have point-of-operation protection.
- Do not reach into machinery during operation.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

This automation equipment and related software is used to control a variety of industrial processes. The type or model of automation equipment suitable for each application will vary depending on factors such as the control function required, degree of protection required, production methods, unusual conditions, government regulations, etc. In some applications, more than one processor may be required, as when backup redundancy is needed.

Only you, the user, machine builder or system integrator can be aware of all the conditions and factors present during setup, operation, and maintenance of the machine and, therefore, can determine the automation equipment and the related safeties and interlocks which can be properly used. When selecting automation and control equipment and related software for a particular application, you should refer to the applicable local and national standards and regulations. The National Safety Council's Accident Prevention Manual (nationally recognized in the United States of America) also provides much useful information.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the operator's hands and other parts of the body are free to enter the pinch points or other hazardous areas and serious injury can occur. Software products alone cannot protect an operator from injury. For this reason the software cannot be substituted for or take the place of point-of-operation protection.

Ensure that appropriate safeties and mechanical/electrical interlocks related to point-of-operation protection have been installed and are operational before placing the equipment into service. All interlocks and safeties related to point-of-operation protection must be coordinated with the related automation equipment and software programming.
NOTE: Coordination of safeties and mechanical/electrical interlocks for point-of-operation protection is outside the scope of the Function Block Library, System User Guide, or other implementation referenced in this documentation.

START-UP AND TEST

Before using electrical control and automation equipment for regular operation after installation, the system should be given a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such a check be made and that enough time is allowed to perform complete and satisfactory testing.

<table>
<thead>
<tr>
<th>WARNING</th>
</tr>
</thead>
<tbody>
<tr>
<td>EQUIPMENT OPERATION HAZARD</td>
</tr>
</tbody>
</table>

- Verify that all installation and set up procedures have been completed.
- Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices.
- Remove tools, meters, and debris from equipment.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Follow all start-up tests recommended in the equipment documentation. Store all equipment documentation for future references.

**Software testing must be done in both simulated and real environments.**

Verify that the completed system is free from all short circuits and temporary grounds that are not installed according to local regulations (according to the National Electrical Code in the U.S.A, for instance). If high-potential voltage testing is necessary, follow recommendations in equipment documentation to prevent accidental equipment damage.

Before energizing equipment:
- Remove tools, meters, and debris from equipment.
- Close the equipment enclosure door.
- Remove all temporary grounds from incoming power lines.
- Perform all start-up tests recommended by the manufacturer.
OPERATION AND ADJUSTMENTS

The following precautions are from the NEMA Standards Publication ICS 7.1-1995 (English version prevails):

- Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly operated.
- It is sometimes possible to misadjust the equipment and thus produce unsatisfactory or unsafe operation. Always use the manufacturer’s instructions as a guide for functional adjustments. Personnel who have access to these adjustments should be familiar with the equipment manufacturer’s instructions and the machinery used with the electrical equipment.
- Only those operational adjustments actually required by the operator should be accessible to the operator. Access to other controls should be restricted to prevent unauthorized changes in operating characteristics.
About the Book

At a Glance

Document Scope

This document describes an example application implementing a data exchange between two controllers over the Ethernet network using the TcpUdpCommunication library.

Validity Note

This document has been updated for the release of SoMachine V4.3.

The technical characteristics of the devices described in this document also appear online. To access this information online:

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Go to the Schneider Electric home page <a href="http://www.schneider-electric.com">www.schneider-electric.com</a>.</td>
</tr>
</tbody>
</table>
| 2    | In the Search box type the reference of a product or the name of a product range.  
|      | ● Do not include blank spaces in the reference or product range.  
|      | ● To get information on grouping similar modules, use asterisks (*). |
| 3    | If you entered a reference, go to the Product Datasheets search results and click on the reference that interests you.  
|      | If you entered the name of a product range, go to the Product Ranges search results and click on the product range that interests you. |
| 4    | If more than one reference appears in the Products search results, click on the reference that interests you. |
| 5    | Depending on the size of your screen, you may need to scroll down to see the data sheet. |
| 6    | To save or print a data sheet as a .pdf file, click Download XXX product datasheet. |

The characteristics that are presented in this manual should be the same as those characteristics that appear online. In line with our policy of constant improvement, we may revise content over time to improve clarity and accuracy. If you see a difference between the manual and online information, use the online information as your reference.
Product Related Information

⚠️ WARNING

LOSS OF CONTROL

- The designer of any control scheme must consider the potential failure modes of control paths and, for certain critical control functions, provide a means to achieve a safe state during and after a path failure. Examples of critical control functions are emergency stop and overtravel stop, power outage and restart.
- Separate or redundant control paths must be provided for critical control functions.
- System control paths may include communication links. Consideration must be given to the implications of unanticipated transmission delays or failures of the link.
- Observe all accident prevention regulations and local safety guidelines.¹
- Each implementation of this equipment must be individually and thoroughly tested for proper operation before being placed into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.


Before you attempt to provide a solution (machine or process) for a specific application using the POUs found in the library, you must consider, conduct and complete best practices. These practices include, but are not limited to, risk analysis, functional safety, component compatibility, testing and system validation as they relate to this library.

⚠️ WARNING

IMPROPER USE OF POUS

- Perform a safety-related analysis for the application and the devices installed.
- Ensure that the POUs are compatible with the devices in the system and have no unintended effects on the proper functioning of the system.
- Use appropriate parameters, especially limit values, and observe machine wear and stop behavior.
- Verify that the sensors and actuators are compatible with the selected POUs.
- Thoroughly test all functions during verification and commissioning in all operation modes.
- Provide independent methods for critical control functions (emergency stop, conditions for limit values being exceeded, etc.) according to a safety-related analysis, respective rules, and regulations.

Failure to follow these instructions can result in death, serious injury, or equipment damage.
Related Documents

<table>
<thead>
<tr>
<th>Document title</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>SoMachine Programming Guide</td>
<td>EIO0000000067 (ENG);</td>
</tr>
<tr>
<td>SoMachine Getting &amp; Setting Real Time Clock SysTimeRtc and SysTimeCore Library Guide</td>
<td>EIO0000000067 (ENG);</td>
</tr>
<tr>
<td>SoMachine TcpUdpCommunication Library Guide</td>
<td>EIO0000002204 (ENG);</td>
</tr>
</tbody>
</table>


Terminology Derived from Standards

The technical terms, terminology, symbols and the corresponding descriptions in this manual, or that appear in or on the products themselves, are generally derived from the terms or definitions of international standards.

In the area of functional safety systems, drives and general automation, this may include, but is not limited to, terms such as safety, safety function, safe state, fault, fault reset, malfunction, failure, error, error message, dangerous, etc.

Among others, these standards include:

<table>
<thead>
<tr>
<th>Standard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN 61131-2:2007</td>
<td>Programmable controllers, part 2: Equipment requirements and tests.</td>
</tr>
<tr>
<td>ISO 12100:2010</td>
<td>Safety of machinery - General principles for design - Risk assessment and risk reduction</td>
</tr>
<tr>
<td>EN 60204-1:2006</td>
<td>Safety of machinery - Electrical equipment of machines - Part 1: General requirements</td>
</tr>
<tr>
<td>ISO 13850:2006</td>
<td>Safety of machinery - Emergency stop - Principles for design</td>
</tr>
</tbody>
</table>
In addition, terms used in the present document may tangentially be used as they are derived from other standards such as:

<table>
<thead>
<tr>
<th>Standard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006/42/EC</td>
<td>Machinery Directive</td>
</tr>
<tr>
<td>2014/30/EU</td>
<td>Electromagnetic Compatibility Directive</td>
</tr>
<tr>
<td>2014/35/EU</td>
<td>Low Voltage Directive</td>
</tr>
</tbody>
</table>

In addition, terms used in the present document may tangentially be used as they are derived from other standards such as:

<table>
<thead>
<tr>
<th>Standard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEC 60034 series</td>
<td>Rotating electrical machines</td>
</tr>
<tr>
<td>IEC 61800 series</td>
<td>Adjustable speed electrical power drive systems</td>
</tr>
<tr>
<td>IEC 61158 series</td>
<td>Digital data communications for measurement and control – Fieldbus for use in industrial control systems</td>
</tr>
</tbody>
</table>

Finally, the term *zone of operation* may be used in conjunction with the description of specific hazards, and is defined as it is for a *hazard zone* or *danger zone* in the *Machinery Directive (2006/42/EC)* and *ISO 12100:2010*.

**NOTE:** The aforementioned standards may or may not apply to the specific products cited in the present documentation. For more information concerning the individual standards applicable to the products described herein, see the characteristics tables for those product references.
Chapter 1
Example Description

Overview

Short Description
The SoMachine project TcpUdpCommunicationExample.project implements an example application which shows how to implement a data exchange between two controllers over the Ethernet network using the TcpUdpCommunication library. The example project contains two controllers with similar applications.

Each application implements the functions:
• TCP client (Transmission Control Protocol)
• TCP server
• UDP peer (User Datagram Protocol)

System Requirements and Limitations
The application example has been created and tested with the components listed in the table:

<table>
<thead>
<tr>
<th>Component</th>
<th>Type and Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software</td>
<td>SoMachine V4.3 or later</td>
</tr>
<tr>
<td>Controller</td>
<td>● TM241CEC24T/U</td>
</tr>
<tr>
<td></td>
<td>● TM251MESE</td>
</tr>
<tr>
<td></td>
<td>The concepts presented in</td>
</tr>
<tr>
<td></td>
<td>the example application are</td>
</tr>
<tr>
<td></td>
<td>transferable to other</td>
</tr>
<tr>
<td></td>
<td>controllers supporting the</td>
</tr>
<tr>
<td></td>
<td>controller firmware released</td>
</tr>
<tr>
<td></td>
<td>in conjunction with SoMachine V4.3 or later.</td>
</tr>
</tbody>
</table>

Additional devices –
Additional requirements –
Limitations –
Chapter 2
Project Information

Overview
This section provides information about the SoMachine project.

What Is in This Chapter?
This chapter contains the following sections:

<table>
<thead>
<tr>
<th>Section</th>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Hardware Configuration</td>
<td>16</td>
</tr>
<tr>
<td>2.2</td>
<td>Application</td>
<td>17</td>
</tr>
<tr>
<td>2.3</td>
<td>Visualization</td>
<td>26</td>
</tr>
<tr>
<td>2.4</td>
<td>Running the Application Example</td>
<td>30</td>
</tr>
</tbody>
</table>
Section 2.1
Hardware Configuration

Overview of the Hardware Configuration

Overview

The example project implements two controllers. To execute the example applications, the Ethernet setting of the Ethernet interfaces for both controllers must be configured to comply with the requirements of the network.

The figure shows the layout of the network:

- PC with SoMachine
  - IP Address: 10.128.154.39

- M251 Controller
  - IP Address: 10.128.154.242

- M241 Controller
  - IP Address: 10.128.154.241
Section 2.2
Application

What Is in This Section?
This section contains the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of the Application</td>
<td>18</td>
</tr>
<tr>
<td>Libraries</td>
<td>18</td>
</tr>
<tr>
<td>Data Format - Application Protocol</td>
<td>19</td>
</tr>
<tr>
<td>TCP Client</td>
<td>21</td>
</tr>
<tr>
<td>TCP Server</td>
<td>22</td>
</tr>
<tr>
<td>UDP Peer</td>
<td>24</td>
</tr>
<tr>
<td>Simulation</td>
<td>25</td>
</tr>
<tr>
<td>Task Configuration</td>
<td>25</td>
</tr>
</tbody>
</table>
Overview of the Application

Overview

The applications for both controllers are similar. Each application implements the following functions:

<table>
<thead>
<tr>
<th>Function</th>
<th>Communicates with</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP client</td>
<td>Communicates with the TCP server of the other controller.</td>
</tr>
<tr>
<td>TCP server</td>
<td>Communicates with the TCP client of the other controller.</td>
</tr>
<tr>
<td>UDP peer</td>
<td>Communicates with the UDP peer of the other controller.</td>
</tr>
</tbody>
</table>

To control and to monitor the applications, each application implements a SoMachine visualization. The visualizations are available as web visualizations. They can be accessed with a compatible web browser from a PC which is connected to the same Ethernet network as the controllers.

Libraries

Overview

The following libraries have been added to the Library Manager of each application:

<table>
<thead>
<tr>
<th>Library</th>
<th>Company</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>TcpUdpCommunication</td>
<td>Schneider Electric</td>
<td>Provides the components for communication over the Ethernet network with the use of the transport protocols TCP and/or UDP.</td>
</tr>
<tr>
<td>SysTimeRtc</td>
<td>System</td>
<td>Provides the components to manage the real time clock (RTC) of the controllers.</td>
</tr>
<tr>
<td>SysMem</td>
<td>System</td>
<td>Provides components to access the dynamic memory management of the runtime system.</td>
</tr>
</tbody>
</table>

The TcpUdpCommunication library provides parameters to configure the supported functions. You can access these parameters in the global parameter list (GPL) available in the Library Manager of the application.

The table lists the parameters that have been adjusted in the example application:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Initial value</th>
<th>New value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gc_uiTCPServerMaxConnections</td>
<td>30</td>
<td>5</td>
</tr>
</tbody>
</table>
Data Format - Application Protocol

Overview

The protocols TCP and UDP are used for the socket-based network communication. The data transmitted via these protocols is assembled based on a specific application protocol. Both peers must use the same application protocol for successful communication.

Data Format

For this example application, an application protocol format has been defined. It allows the application to verify the received data and to create the suitable response if a request for information has been received.

This simple protocol contains the following information:
- Transmission ID
- Length of the text message
- Text message which represents the request command or the information

The figure illustrates the data format of the application protocol:

![Data Format Diagram]

The table provides further information on the elements:

<table>
<thead>
<tr>
<th>Item of the protocol</th>
<th>Datatype</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmission ID</td>
<td>UINT</td>
<td>The transmission ID is randomly generated by the TCP client. It is sent back by the server in the response so that the client can verify the received data.</td>
</tr>
<tr>
<td>Length of text message</td>
<td>USINT</td>
<td>Indicates the length of the text message in bytes (number of characters + 1). This allows the receiving site to verify the consistency of the data.</td>
</tr>
<tr>
<td>Text message</td>
<td>STRING[80]</td>
<td>The text message contains the application data.</td>
</tr>
</tbody>
</table>
Client/Server Communication

The communication between client and server via TCP consists of the following steps:

- The client sends a request with the text message GetTimestamp to the server.
- The server responds to that request:
  - If the request was correct, the server responds by sending the same transmission ID and the system time as text STRING in ASCII format.
  - If the server cannot interpret the request message (because it differs from GetTimestamp), the response contains the text STRING UnknownCommand besides the transmission ID and the length of the text message.
TCP Client

Overview
The program SR_TcpClient implements the program code for the TCP client functionality. For control and monitoring of this functionality, the visualization screen VisuTcpClient is provided in the example project.

The variables which link the visualization with the program code are defined in the global variable list GVL_VisuTcpClient.

Connecting to the TCP Server
In the visualization, enter the IP address and the port number of the TCP server you want to connect to. To establish the connection to this TCP server, click the Connect & Send button in the visualization.

Sending Data to the TCP Server
After the TCP client has established a connection to the TCP server, a data packet (see page 19) is sent to the server. You can modify the text message in the visualization. The TCP server implemented in this example application only accepts the message GetTimestamp.

Receiving the Response from the TCP Server
After the TCP client has sent a data packet to the TCP server, a response is expected within a certain time. You can set the allowed time for receiving the response in the visualization.

By sending the text message GetTimestamp, the client expects the system time of the controller running the TCP server as response.

The response is verified using the transmission ID. The transmission ID of the received data packet must be equal to the transmission ID of the sent data packet. If a different transmission ID is received, the received data is rejected and a diagnostic message is generated.

Terminating the Connection to the TCP Server
The connection is terminated by the TCP client if one of the following conditions applies:
- A response has been received from the TCP server (valid or not).
- The timeout has been reached without receiving a response.
TCP Server

Overview
The program SR_TcpServer implements the program code for the TCP server functionality. For controlling and monitoring this functionality, the visualization screen VisuTcpServer is provided in the example project.

The variables which link the visualization with the program code are defined in the global variable list GVL_VisuTcpServer.

Opening/Closing the TCP Port
In the visualization, enter the number of the TCP port you want to open and the IP address of the corresponding Ethernet interface. To open this TCP port, click the Listen button in the visualization. You can close an open port by clicking the Close button.

Accepting Incoming Connections
After the port has been opened, it is monitored by the TCP server for incoming connections and receiving data from connected clients.

Each incoming client connection is accepted. The number of connected clients is shown in the visualization screen VisuTcpServer.

Disconnecting Clients
From the visualization screen VisuTcpServer, you can disconnect a selected client or all clients at once.

The program provides a function for disconnecting clients after a specified time of inactivity. If no data has been received from a client for a certain time, this client is disconnected automatically. You can configure this maximum time of inactivity in the visualization screen VisuTcpServer.

Receiving Data
If the TCP server detects that data has been received from a TCP client, the data is read and evaluated by the application. It is verified whether the data applies to the format of the defined application protocol (see page 19).

Depending on the result of the evaluation, one of the following actions is executed by the SR_TcpServer program:

<table>
<thead>
<tr>
<th>If</th>
<th>Then</th>
</tr>
</thead>
<tbody>
<tr>
<td>If the value received for the length of the text message meets the size of the text received and is not greater than the size of the receive buffer provided by the application (84 bytes) (expected condition)</td>
<td>The data is read and is processed based on the expected data format.</td>
</tr>
</tbody>
</table>
Sending Data

Whenever valid data is received from the TCP client, a response is sent. The response is created depending on the received data. If the text message `GetTimestamp` has been received, the response data contains the time stamp of the system (RTC of the controller) as text message. If a text message other than `GetTimestamp` has been received, the response data contains the text message `UnknownCommand`.

<table>
<thead>
<tr>
<th>If</th>
<th>Then</th>
</tr>
</thead>
<tbody>
<tr>
<td>If the value received for the length of the text message does not</td>
<td>The data is read and is processed based on the expected data format.</td>
</tr>
<tr>
<td>meet the size of received data and is not greater than the size</td>
<td>A diagnostic message indicating the inconsistency of the given length</td>
</tr>
<tr>
<td>of the receive buffer provided by the application (84 bytes)</td>
<td>is generated.</td>
</tr>
<tr>
<td>If the value received for the length of the text message does not</td>
<td>The data is not read and the connection to this client is terminated.</td>
</tr>
<tr>
<td>meet the size of received data and is greater than the size of</td>
<td></td>
</tr>
<tr>
<td>the receive buffer</td>
<td></td>
</tr>
</tbody>
</table>

Sending Data

Whenever valid data is received from the TCP client, a response is sent. The response is created depending on the received data. If the text message `GetTimestamp` has been received, the response data contains the time stamp of the system (RTC of the controller) as text message. If a text message other than `GetTimestamp` has been received, the response data contains the text message `UnknownCommand`.
UDP Peer

Overview
The program SR_UdpPeer implements the program code for the UDP peer functionality. For control and monitoring of this functionality, the visualization screen VisuUdpPeer is provided in the example project.

The variables which link the visualization with the program code are defined in the global variable list GVL_VisuUdpPeer.

Opening and Binding
In the visualization, enter the number of the port you want to open and the IP address of the corresponding Ethernet interface. To open this UDP port, click the Open & Bind button in the visualization.

You can close an open port by clicking the Close button.

Sending Data
When the UDP port is open and bound, a data packet (see page 19) can be sent to a remote UDP peer. The remote UDP peer is selected by its IP address and the UDP port. You can modify the text message in the visualization. To send data to the remote UDP peer, click the Send button in the visualization.

Receiving Data
If the UDP peer detects that data has been received on the bound port, the data is read from the receive buffer. The received data is converted into the user-defined data format (see page 19). In this example application, the data is not processed, but displayed on the visualization for information purposes.
Simulation

Overview

The application provides a simulation mode. You can activate the simulation mode for each function separately from the respective visualization.

The table lists the processes in simulation mode for each function:

<table>
<thead>
<tr>
<th>Function</th>
<th>Process in simulation mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP client</td>
<td>Cyclically initiates a data exchange with the TCP server of the other controller.</td>
</tr>
<tr>
<td>TCP server</td>
<td>The selected port opens automatically and remains open.</td>
</tr>
<tr>
<td>UDP peer sender</td>
<td>Cyclically sends data to the UDP peer of the other controller.</td>
</tr>
<tr>
<td>UDP peer receiver</td>
<td>The selected port opens automatically and remains open.</td>
</tr>
</tbody>
</table>

You can set the time interval for sending data in the respective visualization screen.

Task Configuration

Overview

In addition to the MAST task, the task configuration of the example application contains the following tasks:

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP_UDP_Communication</td>
<td>Manages the calls of the POU's which are related to the TCP and UDP communication. The task type is <strong>Cyclic</strong> and the <strong>Priority</strong> is lower than that of the MAST task.</td>
</tr>
<tr>
<td>VISU_TASK</td>
<td>This task is added automatically when adding the web visualization. The default settings are left unchanged.</td>
</tr>
</tbody>
</table>
Section 2.3
Visualization

Visualization Screens

Overview

For each implemented communication function, a dedicated visualization screen is available. The visualization VisuStart contains a tab group that allows you to switch between the visualization screens.

The web visualization offers you access to machine control functions over the network. To help prevent unauthorized access to your machine control, implement the following technical and organizational measurements for the system running your application.

WARNING

UNAUTHENTICATED, UNAUTHORIZED ACCESS

- Do not expose controllers and controller networks to public networks and the Internet as much as possible.
- Use additional security layers such as VPN for remote access and install firewall mechanisms.
- Restrict access to authorized personnel by activation and deployment of the user management of the controller and the visualization.
- Change default passwords at start-up and modify them frequently.
- Validate the effectiveness of these measurements regularly and frequently.

Failure to follow these instructions can result in death, serious injury, or equipment damage.
TCP Client

VisuStart → TCP Client

TCP Client Status

- Status: Idle
- Result: Ok
- Diag: No alarm or warning
TCP Server

VisuStart → TCP Server

![TCP Server Interface]

- **Ethernet Status Controller**
  - **IpAddr:** 10.128.154.242
  - **Ethernet Status:** OK

- **TCP Server**
  - **Server IpAddr:** 10.128.154.242
  - **Server Port:** 35274
  - **Auto Listen:** On
  - **Close:**

- **Data Received**
  - **Trans ID:** b818
  - **Length:** 13
  - **Data:** GetTimestamp

- **Data to Send**
  - **Trans ID:** b818
  - **Length:** 23
  - **Data:** DT#2016-10-10-17:59:01

- **Disconnect Clients**
  - **Connected Clients:** 0
  - **Timeout inactivity:** T#30s

- **Client to disconnect**
  - **Ip Addr.:**
  - **Port:** 0

- **TcpServer Status**
  - **Status:** Listening
  - **Result:** Ok
  - **Diag:** No alarm or warning

- **Buttons:**
  - **Clear Diag**
UDP Peer

VisuStart → UDP Peer

TCP Client | TCP Server | UDP Peer

Ethernet Status Controller
IpAddr: 10.128.154.242  Ethernet Status: 

UDP Port
Local IpAddr: 10.128.154.242  Auto Mode: T#1s
Local UDP Port: 35274

Send Data
Remote IpAddr: 10.128.154.241
Remote Port: 35274
Trans ID | Length | Data (Message)
3610 | 13 | Hello World!

Receive Data
Remote IpAddr: 10.128.154.241
Remote Port: 35274
Trans ID | Length | Data (Message)
3fa3 | 13 | Hello World!

UDP Peer Status
Status: Bound
Result: Ok
Diag: No alarm or warning

Close
Section 2.4
Running the Application Example

What Is in This Section?

This section contains the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preconditions for Running the Application Example</td>
<td>31</td>
</tr>
<tr>
<td>Downloading the Application</td>
<td>31</td>
</tr>
<tr>
<td>Operating the Application</td>
<td>32</td>
</tr>
</tbody>
</table>
Preconditions for Running the Application Example

Overview

To run the application example, the following preconditions must be fulfilled:

- Each application was downloaded and is running on the associated controller.
- A PC, laptop, or mobile device with a compatible web browser is connected to the same network as the controllers so that the web visualization can be accessed.

Downloading the Application

Overview

To set the communication parameter of the controllers, and to download the applications, perform these steps:

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Set up the hardware according to the network layout <em>(see page 16)</em>.</td>
<td>Both controllers and the PC must be connected to the same Ethernet network.</td>
</tr>
<tr>
<td>2</td>
<td>Open the TcpUdpCommunicationExample project in SoMachine and open the Logic Builder.</td>
<td>–</td>
</tr>
<tr>
<td>3</td>
<td>Double-click the MyController node in the Devices tree.</td>
<td>A network scan is performed automatically. Available controllers are provided in the Controller selection view of the device editor.</td>
</tr>
<tr>
<td>4</td>
<td>Select the target controller from the list.</td>
<td>–</td>
</tr>
<tr>
<td>5</td>
<td>Right-click the controller entry and execute the command Process communication settings... from the context menu.</td>
<td>–</td>
</tr>
<tr>
<td>6</td>
<td>Enter the Communication parameter.</td>
<td>–</td>
</tr>
<tr>
<td>7</td>
<td>Activate the option Save settings permanently and click the OK button.</td>
<td><strong>Result:</strong> The communication parameters are stored on the controller. They are retained even if it is restarted. These parameters override the application settings.</td>
</tr>
<tr>
<td>8</td>
<td>Repeat steps 3...7 for the other controller.</td>
<td>–</td>
</tr>
<tr>
<td>9</td>
<td>Execute the Multiple Download... command from the Online menu.</td>
<td>–</td>
</tr>
<tr>
<td>10</td>
<td>In the Multiple Download dialog box, select the applications, and click the OK button.</td>
<td><strong>Result:</strong> The selected applications are downloaded to the previously selected controllers.</td>
</tr>
</tbody>
</table>

For more information on downloading an application and the communication settings, refer to the SoMachine Programming Guide *(see SoMachine, Programming Guide)*.
Operating the Application

Overview

The example applications implement a web visualization for monitoring and control purposes. To open the web visualization, enter the following address in the address bar of a web browser:
http://<IP Address>:8080\webvisu.htm

The visualization is self-explanatory. However, this section provides a brief description of each function. In addition, refer to the visualization screens (see page 26).

TCP Client

Enter the IP address and the port of the target TCP server, which is usually the second of the two controllers. Enter the text message that shall be sent to the TCP server. Click the Connect & Send button from the TCP client visualization to establish a connection to the server and to send data. The status of the TCP client and the exchanged data are displayed on the visualization.

TCP Server

Open a configurable TCP port to allow incoming client connections to be accepted. Data can be received from the client through the established TCP connection. Depending on the content of a received message, a response is sent to the TCP client. The status of the TCP server and the exchanged data are displayed in the visualization.

UDP Peer

Open and bind a configurable UDP port so that data can be received from another UDP peer. The other UDP peer is usually the second controller. Enter the IP address and the port of the target UDP peer. Enter the text message that shall be sent to the other UDP peer. Click the Send button in the visualization to send the data. The status of the UDP peer and the exchanged data are displayed in the visualization.
Glossary

A

application
A program including configuration data, symbols, and documentation.

C

client
A client is a part of a communications application. The initially active part establishes a connection (TCP) or sends data to the server.

configuration
The arrangement and interconnection of hardware components within a system and the hardware and software parameters that determine the operating characteristics of the system.

controller
Automates industrial processes (also known as programmable logic controller or programmable controller).

E

Ethernet
A physical and data link layer technology for LANs, also known as IEE 802.3.
Ethernet is the most widely spread technology for local networks. Each PacDrive controller has an Ethernet port. The Ethernet standard defines layer 1 and 2 of the communication. Above the Ethernet, there are many different network protocols but only IP is used.

expansion bus
An electronic communication bus between expansion I/O modules and a controller.

I

I/O
(input/output)

IP address
The IP address of IPv4 (Internet Protocol version 4) is a value of 4 bytes which identifies the devices connected to an IP network.
P

packet and datagram
On network level, the term packet is used for the data packets which this level transmits. In the case of UDP, these terms are used synonymously.

peer
Term for the other system participating in the communication. This term is used if it is unimportant whether the other system is a server or a client.

port / port number
A port number, frequently also designated as port, is a number from 1 to 65535. In combination with an IP address, it designates a communication end point. A socket is always connected with a port number. As sockets are used by the communication function blocks of TCP/UDP communication, and these again by a program, a port number identifies a program, a server, or a client running on a controller.

If you communicate with the <IP of a controller>:<Port number>, then you communicate with a program that has connected itself to this port number. (The program has configured its communication function block such that it connects to this port number.)

program
The component of an application that consists of compiled source code capable of being installed in the memory of a logic controller.

R

RTC
(real-time clock) A battery-backed time-of-day and calender clock that operates continuously, even when the controller is not powered for the life of the battery.

S

server
A server is a part of a communications application. At first, the server is passive. It waits until clients initialize communication. A server runs on a defined port number and the clients know its address.

socket
A socket is a resource which is used by TCP/UDP communication internally in order to access the communications functions in the firmware.

T

TCP
TCP (Transmission Control Protocol) is a transmission protocol used in IP networks.
UDP

UDP (User Datagram Protocol) is a transmission protocol used in IP networks.
Index

A
application examples
  data exchange between two controllers, 13

E
examples
  TcpUdpCommunicationExample, 9

T
TcpUdpCommunication library, 18
TcpUdpCommunicationExample
eexample guide, 9