Technical Note

Writing a Value on Startup

This document describes an example of setting up Kepware® software to solve the following problem:

A. A constant value (e.g., a password) is to be written to a piece of equipment automatically on startup.

- This example uses Modbus Serial as the communications protocol.

B. The software also must be able to write this value on command from an operator.

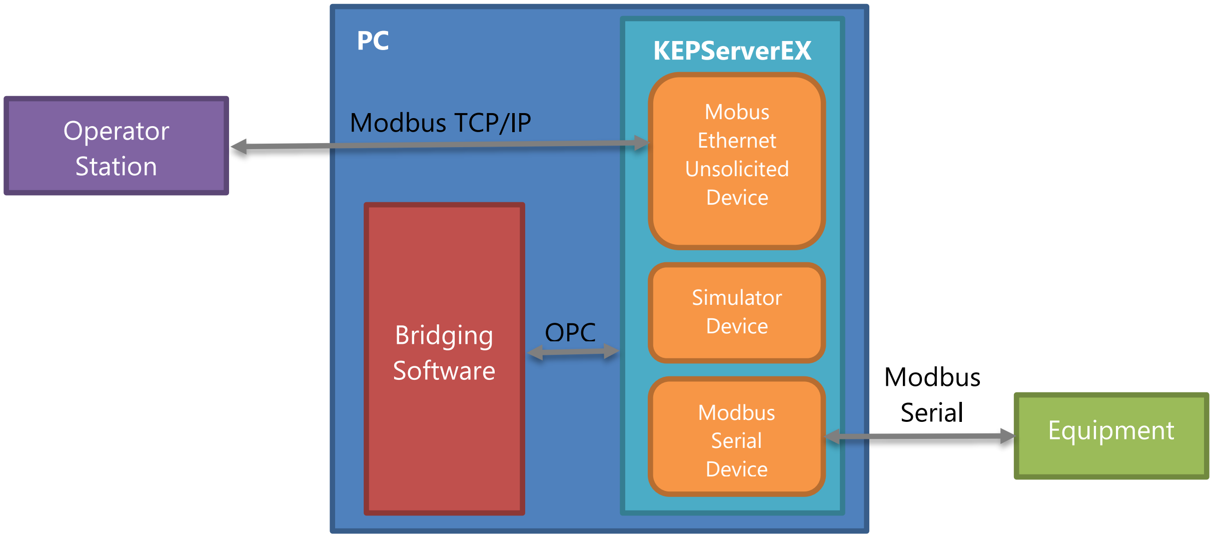
C. The operator command is sent via a device communications protocol, not OPC.

- This example uses Modbus TCP/IP Ethernet as the operator-station communications protocol.
- OPC is meant for reliable plant networks. A WAN between the operator station and the Kepware software may have occasional network outages that can be better tolerated by a device protocol.

D. The operator must be able to monitor the communications status of the equipment.

The reader should be moderately familiar with Kepware software and OPC standards.

The following is a block diagram of the software and hardware:



1. Setup

The block diagram shows two main software components: KEPServerEX® ("the server") and bridging software.

1.1 Server Setup

The devices in the server have the following tags and properties:

Modbus Serial Device

- The tag ("value tag") to which the constant value is to be written:
- <Channel>.<Device>._System._Error tag (automatically provided by the server)
- <Channel>.<Device>._System._NoError tag (automatically provided by server versions 4.190.341 and higher)

Simulator Device

- At the channel level in the Simulator driver, Item Persistence is turned on and a file is specified (*see "Item Persistence File" below*).
- Tag ("value tag," e.g., S0 if a string, K0 if a number, K0.0 if a Boolean) holding the constant value:
- <Channel>.<Device>._System._Enabled tag (automatically provided by the server)

Modbus Ethernet Unsolicited Device

- To make an unsolicited device, set the device ID to 127.0.0.1.<optional 1-byte node ID>
- Boolean write-trigger tag (the operator station writes to this tag to send the value to the equipment)
- Boolean communications-status tag (reflecting the communications status of the Modbus Serial device)

1.2 Bridging Setup

The following links are necessary:

To write the constant value:

- Modbus Ethernet write-trigger tag to Simulator _Enabled tag
- Simulator value tag to Modbus Serial value tag

To write the value if the equipment's power is cycled while the server PC keeps running:

- Modbus Serial _NoError tag to Simulator _Enabled tag

To let the operator station monitor the status of the server's communications to the equipment:

- Modbus Serial _Error tag to Modbus Ethernet communications-status tag

1.3 Item Persistence File

To generate the item persistence file, do the following.

- Use QuickClient (local to the server) to write the value to the Simulator value tag.
- Fully shut down the server once (**File | Exit** and **Shutdown**, if available, otherwise **File | Exit**). This generates the item persistence file.
- The file is then automatically loaded when the server starts up with the project file.

2. Operation

2.1 Writing the Value by Hand

1. The Modbus Ethernet main (at the operator station) writes a 0, then a 1, to the Modbus Ethernet write-trigger tag.
2. The bridging software copies these two writes to the Simulator _Enabled tag.
3. The _Enabled tag changes from 0 to 1, so:
4. The server sends a data-change event for the Simulator value tag to the bridging software.
5. The bridging software writes the data-change value to the Modbus Serial value tag.

2.2 Automatic Operation on Server Startup

Whenever the server starts up, steps 4 and 5 above occur automatically, provided the following note is observed.

Note: The state of the _Enabled tag is saved in the server's project file. The state must be saved as 1 (device enabled) in order for the value to be sent when the server PC powers up. The state of this tag is 1 by default, and can be reviewed by looking, on the Simulator device's Device Properties, General group with data collection enabled. If enabled, the _Enabled tag is 1.

2.3 Automatic Operation on Equipment Restart

If the equipment's power is cycled while the server PC keeps running, the Modbus Serial _NoError tag goes to 0, then 1. This (since the bridging software is linking this _NoError tag to the Simulator _Enabled tag) has the same effect as the procedure listed in "Writing the Value by Hand."