

PowerSCADA Expert Design Reference Guide

Version 7.40

63220-100-19D4

04/2014

Safety information

Important information

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 bulletin or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.



The addition of either 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.s



DANGER

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



WARNING

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



CAUTION

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

NOTICE

NOTICE is used to address practices not related to physical injury. The safety alert symbol shall not be used with this signal word.

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, installation, and operation of electrical equipment and has received safety training to recognize and avoid the hazards involved.

Safety precautions

During installation or use of this software, pay attention to all safety messages that occur in the software and that are included in the documentation. The following safety messages apply to this software in its entirety.



WARNING

UNINTENDED EQUIPMENT OPERATION

- Do not use the software for critical control or protection applications where human or equipment safety relies on the operation of the control action.
- Do not use the software to control time-critical functions because communication delays can occur between the time a control is initiated and when that action is applied.
- Do not use the software to control remote equipment without securing it with an authorized access level, and without including a status object to provide feedback about the status of the control operation.

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



WARNING

INACCURATE DATA RESULTS

- Do not incorrectly configure the software, as this can lead to inaccurate reports and/or data results.
- Do not base your maintenance or service actions solely on messages and information displayed by the software.
- Do not rely solely on software messages and reports to determine if the system is functioning correctly or meeting all applicable standards and requirements.
- Consider the implications of unanticipated transmission delays or failures of communications links.

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

Contents

Safety information	3
Important information	3
Safety precautions	4
Contents	i
Introduction	1
Ethernet Network Design	2
Physical Planning and Design of an Ethernet Network	2
Understanding Basic Network Structure	2
Star Topology	2
Ring Topology	3
Dual Ring Topology	3
Mesh Topology	4
Other LAN Considerations	4
Full-Duplex vs. Half-Duplex	5
When to Use a Switch	5
Bandwidth	5
Physical Planning and Layout	5
Industrial Ethernet Cable Planning	5
Factors that Affect System Performance	5
Additional Recommendations for High-Performance Systems	6
Recommended Devices for Industrial Ethernet	6
Combating EMI in Ethernet Networks	6
Installation Measures to Combat EMI in Ethernet Networks	7
Communication Selection	7
Summary	7
MODBUS Messaging	7
SNMP	7
Note about time synchronization:	7

Physical Planning and Layout	9
Industrial Ethernet Cable Planning	9
Factors that Affect System Performance	9
Additional Recommendations for High-Performance Systems	9
Recommended Devices for Industrial Ethernet	9
PLSCADA Expert Architecture	11
I/O Server roles:	11
Alarm Server roles:	11
Trend Server roles:	11
Report Server roles:	11
Display Client roles:	11
Web Display Client roles:	11
Network architecture considered in PLSCADA Expert	11
Small Site: Up to 100 Devices (16,900 Tags): Standalone and Redundancy Architecture	12
System Performance Goals	12
Parameter recommendations	13
Medium site: up to 1,000 devices (169,000 Tags): Standalone and Redundancy Clusters	13
System Performance Goals	14
Parameter recommendations	14
Redundancy	14
Parameter recommendations	15
Windows Server Recommended	16
Network configuration	17
CPU load balancing recommendation	18
Data Center with Branch Circuit Monitors, Circuit Monitors and MicroLogic Trip Units	19
Redundancy	19
Data Flow Limit to Achieve the Performance Requirement	19
Parameters Recommended	19
CPU load-balancing recommendation	20
PC and core re-partition:	20

System Performance Evaluation	22
Introduction	22
System response time	22
Introduction	22
What is the response time?	22
How do we estimate response time?	22
MODBUS Messaging Server Response Times	23
Devices Connected with the Ethernet Network through a Gateway	23
Device response time estimation	23
Calculation of Serial Line Transmission Time	23
Frame MODBUS example; read N bits functions (1 and 2)	24
Frame MODBUS example; read N word functions (3 and 4)	24
Gateway Response Times estimation	25
Calculation of the Number of Supported Devices per Gateway	26
Devices connected directly with the Ethernet network.	27
MODBUS TCP/IP Server Response Times	27
CM4 response time	27
Calculation of the Ethernet Timeout	27
MODBUS Messaging Client Response Times	27
SCADA system response time	28
System Performance Tuning	29
Dataflow prioritization	29
General Information	29
Bandwidth allocation	29
Bandwidth allocation for type of data	29
Sepam Example:	29
Bandwidth allocated for the device	30
Tag scan rates	30
Driver Optimization	32
Introduction	32
General Information	32

Packet Blocking Optimization	33
Device communication failure optimization	35
InitUnitCheckTime:	35
Appendix A: Device Response Times	38
DNP3 Protocol	39
ION 7650 (DNP3) Response Times	39
System Configuration	39
ION Device Information	39
Test Procedure	39
Test Results	39
IEC 61850 Protocol	40
Circuit Monitor 4000 (G3200/IEC 61850) Response Times	40
System Configuration	40
Test Procedure	40
Test Results	40
ION 7650 (IEC 61850) Response Times	40
System Configuration	40
Test Procedure	41
Test Results	41
MicroLogic Type P (G3200/IEC 61850) Response Times	42
System Configuration	42
Test Procedure	42
Test Results	42
Power Meter 850 (G3200/IEC 61850) Response Times	43
System Configuration	43
Test Procedure	43
Test Results	43
Sepam T87 (ACE850/IEC 61850) Response Times	44
System Configuration	44
Test Procedure	44
Test Results	44

Sepam T87 (ECI850/IEC 61850) Response Times	45
System Configuration	45
Test Procedure	45
Test Results	45
Modbus Protocol	46
Circuit Monitor 4000 (ECC/Modbus) Response Times	46
System Configuration	46
Test Procedure	46
Test Results	46
Circuit Monitor 4000 (EGX/Modbus) Response Times	47
System Configuration	47
Test Procedure	47
Test Results	47
MicroLogic Type P (EGX/Modbus) Response Times	48
System Configuration	48
Test Procedure	48
Test Results	48
MicroLogic Type P (G3200/Modbus) Response Times	49
System Configuration	49
Test Procedure	49
Test Results	49
Power Meter 800 (ECC/Modbus) Response Times	50
System Configuration	50
Test Procedure	50
Test Results	50
Power Meter 800 (EGX/Modbus) Response Times	51
System Configuration	51
Test Procedure	51
Test Results	52
Power Meter 1200 (CM4ECC/Modbus) Response Times	53
System Configuration	53

Test Procedure	53
Test Results	53
Sepam S42 (EGX Modbus) Response Times	54
System Configuration	54
Test Procedure	54
Test Results	54
Sepam 80 (EGX Modbus) Response Times	56
System Configuration	56
Test Procedure	56
Sepam 2000 S36 (EGX/Modbus) Response Times	58
System Configuration	58
Test Procedure	58
Test Results	58
TeSys T (EGX/Modbus) Response Times	59
System Configuration	59
Test Procedure	59
Test Results	59
Multiple Protocols	60
Multiple Device/Multiple Protocol Response Times	60
System Configuration	60
Test Procedure	60
Test Results	61
Appendix B: Configuring PLSCADA Expert as an OPC-DA Server	62
Configuration Process	62
Appendix C: Configuring PLSCADA Expert as an OPC-DA Client	65
Configuration Process	65

Introduction

Follow the guidelines in this document to ensure the best possible performance for your customer's StruxureWare PowerSCADA Expert (SPE) system.

As you plan the customer's system, follow the guidelines outlined below:

Ethernet network design

- Design network topology

- LAN considerations

- Industrial Ethernet cable planning

Design the SCADA architecture.

System performance evaluation

Performance tuning

Ethernet Network Design

Physical Planning and Design of an Ethernet Network

Understanding Basic Network Structure

The physical layout, or topology, of a network consisting of cables, components, and devices can be structured in any of these architectures. Each architecture has its advantages and disadvantages. They are outlined in the following sections. Click the links below for descriptions of each of these architectures:

- [Star Topology](#)
- [Ring Topology](#)
- [Dual Ring Topology](#)
- [Mesh Topology](#)

In addition to choosing the correct topology, there are additional LAN considerations to take into account when planning a robust industrial application network.

Star Topology

In a star topology, all the devices are connected through a central device. A star topology is a common network layout for office environments and also for newer automation environments.

In a star topology, devices can use dedicated sections of the network for various services.

Advantages	Disadvantages
Network throughput is much higher than on a shared-media bus topology.	Star topologies are more costly because a dedicated cable must be run to each device. To offset this disadvantage, network infrastructure components (switches, hubs, etc.) are used in cabinets on the factory floor so that a group of local devices can be connected together. A single long cable can be run back to a central point to support the group, rather than using separate cables for each device.
Network reconfiguration is much easier.	
Centralizing network components makes administration easier; centralized management and monitoring of network traffic enhances network performance.	
Diagnostics are simple; if a network segment fails, it affects only the devices directly connected to that segment.	
Infrastructure components use monitoring software and device-based LEDs to indicate failures; most single points of failures can be diagnosed and repaired quickly.	
Resilience; a cable failure takes only that device out of service.	
You can have more devices on a single network than on a bus topology.	

In an Ethernet star, the intermediate device may be a hub or a switch. A star is the most commonly used topology in office networks and has been adopted in most automation applications. For industrial Ethernet applications, the use of a full duplex switch as the central device, rather than a hub, is strongly recommended.

Ring Topology

In a ring topology, all devices or network infrastructure components are connected in a loop with no beginning or end. Packets travel in a single direction on the ring as they are passed from one device to the next. Each device checks a packet for its destination and passes it on to the next device.

Ring topologies provide redundancy. The failure of a single link is handled by routing traffic in the opposite direction. A ring may be based on token rotation or random/shared access. Alternatively, it may be a switched network where all the devices access the network at the same time at different speeds.

Advantages	Disadvantages
Redundancy; the failure of a single link or infrastructure component does not affect the entire network.	High cost; more cabling is needed to complete the ring.
A ring topology uses software to monitor the network links.	Network infrastructure components need intelligence to respond to device failures; they are more costly than simple bus or star components.

Ethernet rings usually form the backbone for high-availability applications. Two paths are available to reach the same device. If ring topology is required, you must use switches that support either a proprietary ring topology or spanning tree protocol (either spanning tree or rapid spanning tree).

Spanning tree protocol (STP; IEEE 802.1D) or rapid spanning tree protocol (RSTP; IEEE 802.1w) are protocols that avoid communication loops and find a new communication path when the initial one is no longer available. The recovery time (time to find a new path) is about 30 s with STP. With RSTP and proper network design, recovery time could be as low as 100 ms.

Dual Ring Topology

When industrial automation systems are used in critical applications where downtime is unacceptable, a dual ring topology may be deployed.

A dual ring has all the features of a single ring with more fault tolerance. It comprises infrastructure components connected together with multiple rings. Each device is connected to two infrastructure components. Each infrastructure component is connected to a separate ring. When a single link or infrastructure device fails, all other devices can still communicate.

Dual ring topologies used in automation environments have additional features not always found in typical data communications environments. For example, hot standby links are used between rings. When a link fails, the standby becomes active and prevents any interruption in network communications. Watchdog packets are sent out to inactive connections and they create logs if the connection remains inactive. The watchdog packets create log entries that are monitored by the network administrator.

Advantages	Disadvantages
Redundancy; the failure of multiple devices or cables does not cause the network to fail.	Cost, compared to a single ring, since the amount of equipment is doubled
Separate power supplies can be used for each ring.	The need to regularly monitor unused links so that they are known to be healthy in the event that they are needed.
Multiple interfaces within a device can connect the device to different rings so that the flooding of one ring with collisions or broadcast traffic does not cause the system to fail.	

Ethernet rings usually form the backbone for high-availability applications. Two paths are available to reach the same device. If ring topology is required, you need to use switches that support either a proprietary ring topology or spanning tree protocol (either spanning tree or rapid spanning tree).

Spanning tree protocol (STP; IEEE 802.1D) or rapid spanning tree protocol (RSTP; IEEE 802.1w) are protocols that avoid communication loops and find a new communication path when the initial one is no longer available. The recovery time (time to find a new path) is about 30 s with STP. With RSTP and proper network design, recovery time could be as low as 100 ms.

Mesh Topology

A mesh topology is used in very large networks or network backbones where every end device or infrastructure device has a connection to one or more components of the network. Ideally, each device is directly connected to every other device in the mesh.

Another mesh implementation is as a network backbone that connects separate star structures. This combined topology provides fault tolerance to the backbone without the high cost of a mesh topology throughout the entire network.

Mesh topologies are used less frequently because of cost and complexity.

Advantages	Disadvantages
Fault tolerance; if a break occurs anywhere in the network cable segment, traffic can be rerouted.	Complexity; difficult to manage and administer.
	High cost; more cabling and interfaces are needed to support the redundant connections. Star Mesh

An Ethernet mesh network offers more redundancy than an Ethernet ring architecture. In a ring, two paths are typically available to the same device. In a mesh network, more than two paths are typically available.

To develop an Ethernet mesh topology, switches that support spanning tree or rapid spanning tree protocol are required.

Other LAN Considerations

Switch and hub configurations work in conjunction with network architecture to help ensure performance. Schneider Electric recommendations for network layout are described below.

Full-Duplex vs. Half-Duplex

Schneider Electric recommends the use of full-duplex switches wherever possible.

Full-duplex switches:

- give greater bandwidth (100 MB in both directions on certain networks)
- allow a device to send responses while receiving additional requests or other traffic
- result in less delays and errors with a device

When to Use a Switch

Switches should always be used in the design of your new network. They offer more intelligence than hubs at an equal or lesser cost.

The industrial switches available today work reliably under extreme conditions such as with electromagnetic interference, high operating temperatures, and heavy mechanical loads. Protect industrial switches by using field-attachable connectors up to IP67 and redundant ring cabling.

Bandwidth

10 MB of bandwidth can be used for smaller end devices, but not for links to PLC/SCADA or to main network links.

100 MB is adequate for most automation systems.

1 GB is useful for the main network link. This capacity is not required, but ensures that more bandwidth is available if needed. 1 GB is necessary if other services share the network with the automation system.

Physical Planning and Layout

Industrial Ethernet Cable Planning

Because there are as yet no defined standards for the physical layout of an industrial Ethernet network, Schneider Electric has chosen to conform to the recommendations submitted by standards organizations such as MODBUS-IDA, IAONA, PNO, and the work in progress by the IEC.

An industrial site is a physical facility in which manufacturing or process control activities take place. In most cases, the site consists of multiple buildings or plants that manage interconnected, but separate, processes. The physical layout and environmental variables inherent in each of these facilities may result in different requirements for the cabling system at each site.

Refer to the Schneider Electric or cable manufacturer recommendation.

Factors that Affect System Performance

Each of these items can affect system performance:

- inherent limitations of each communications protocol
- robustness of the network (e.g., number of retries, timeouts, lost packets)
- response times of the devices in the system
- type of connection for each device (serially or direct to Ethernet)
- number of masters requesting information (PLSCADA Expert, ION E, PLCs, third party)
- routing path for each packet (e.g., hubs, switches, and Gateway)

Additional Recommendations for High-Performance Systems

- Keep serial daisy chains as short as possible: no more than six devices per daisy chain.
- Have a minimum baud rate of 19.2k.
- If you require high-speed performance from your devices, connect them directly to Ethernet.

Recommended Devices for Industrial Ethernet

Generally, you should use switches as much as possible to eliminate collisions, increase performance, and simplify network design. Avoid using hubs whenever possible. Understand network traffic and segment network properly. Follow environmental recommendations provided in this manual.

Schneider Electric recommends the following for use with industrial Ethernet infrastructure devices:

When high bandwidth availability is required:	Use full-duplex switches (10Base-T/100Base-TX). Understand network traffic and segment network properly.
For applications where minimum application downtime is required:	Use self-healing ring or redundant self-healing ring.
For networks that require basic level diagnostics (e.g. no link or failure of one P/S):	Use unmanaged switches with alarm relay.
For networks that require high-level services and traffic administration:	Use managed switches.
For applications that require network discovery and monitoring	Use managed switches.
For applications that require interconnecting devices separated by long distances (>100 m):	Use fiber optic products. Multimode fiber: Up to 2 km between nodes. Monomode fiber: Up to 15 km between nodes. Note: Depending on the fiber and the optical budget, could reach 4 km on multimode and 30 km on monomode.
For networks that require immunity to electromagnetic noise:	Use products with fiber optic ports.
For applications that require physical medium change:	Use transceivers or use switches with a combination of copper and fiber optic ports.
For applications that require external (IP67) mounting of the switch:	Use IP67 switches and cables.

Combating EMI in Ethernet Networks

When properly incorporated into the planning of your network, the following methods can help you avoid electromagnetic disturbances and create an EMC-compliant environment.

Protecting the Ethernet network from electromagnetic interference (EMI) is an issue that involves your complete installation. Although it is important to be concerned about EMI immunity throughout your entire system, this section describes only methods that apply to your Ethernet network. By equipotentially bonding, earthing, properly wiring,

and shielding your site and equipment, you can significantly reduce a large percentage of EMI issues.

Installation Measures to Combat EMI in Ethernet Networks

The following list describes key measures you need to consider in your installation in order to reduce EMI in an industrial Ethernet network:

- earthing and equipotential bonding
- EMC-compatible wiring and cable runs
- balancing circuits
- cable selection
- shielding
- filtering
- placement of devices
- placement of wires
- transposition of outgoing and return lines
- electrical isolation

For more information on EMC, refer to the environmental requirements provided by Schneider Electric. For more information on EMI, refer to EMI requirements provided by Schneider Electric.

Communication Selection

Summary

The following descriptions of methods, provided or tested in the PLSCADA Expert offer, can help you decide which services are best for your application.

MODBUS Messaging

The MODBUS messaging service comprises client and server services. The client initiates a request to the server using the MODBUS protocol; the server responds to the client's request, resulting in information exchange. MODBUS messaging supports both reading and writing of data, as well as a set of programming commands.

MODBUS messaging should be used when data needs to be exchanged between two devices at irregular intervals or infrequent periods. An example is a command to start a process or report on the completion of a process. MODBUS messaging lets you initiate communications only when they are required, making more efficient use of your network and device resources.

SNMP

The SNMP service is for managing networks. It is a network management system that uses SNMP-compliant devices that are queried for information about themselves and the network. SNMP is in almost every Ethernet device and should be used as the basis for most network management systems. It can be used to discover, monitor, and configure devices on a network. SNMP is normally used to transfer device and network status, not device status.

Note about time synchronization:

The time synchronization service provides distribution of a central time source to multiple devices on the network. Accurate time in all devices allows you to properly synchronize events and manage the order of operations across a plant.

The time synchronization service should be used in any environment where timing plays an important role in operations. It eliminates the need to manually set the time on

each network device. Also, the accuracy can be as close as 1 ms in all devices, a level of precision that cannot be achieved when you set the time manually.

Physical Planning and Layout

Industrial Ethernet Cable Planning

Because there are as yet no defined standards for the physical layout of an industrial Ethernet network, Schneider Electric has chosen to conform to the recommendations submitted by standards organizations such as MODBUS-IDA, IAONA, PNO, and the work in progress by the IEC.

An industrial site is a physical facility in which manufacturing or process control activities take place. In most cases, the site consists of multiple buildings or plants that manage interconnected, but separate, processes. The physical layout and environmental variables inherent in each of these facilities may result in different requirements for the cabling system at each site.

Refer to the Schneider Electric or cable manufacturer recommendation.

Factors that Affect System Performance

Each of these items can affect system performance:

- inherent limitations of each communications protocol
- robustness of the network (e.g., number of retries, timeouts, lost packets)
- response times of the devices in the system
- type of connection for each device (serially or direct to Ethernet)
- number of masters requesting information (PLSCADA Expert, ION E, PLCs, third party)
- routing path for each packet (e.g., hubs, switches, and Gateway)

Additional Recommendations for High-Performance Systems

- Keep serial daisy chains as short as possible: no more than six devices per daisy chain.
- Have a minimum baud rate of 19.2k.
- If you require high-speed performance from your devices, connect them directly to Ethernet.

Recommended Devices for Industrial Ethernet

Generally, you should use switches as much as possible to eliminate collisions, increase performance, and simplify network design. Avoid using hubs whenever possible. Understand network traffic and segment network properly. Follow environmental recommendations provided in this manual.

Schneider Electric recommends the following for use with industrial Ethernet infrastructure devices:

When high bandwidth availability is required:	Use full-duplex switches (10Base-T/100Base-TX). Understand network traffic and segment network properly.
For applications where minimum application downtime is required:	Use self-healing ring or redundant self-healing ring.
For networks that require basic level diagnostics (e.g. no link or failure of one P/S):	Use unmanaged switches with alarm relay.

For networks that require high-level services and traffic administration:	Use managed switches.
For applications that require network discovery and monitoring	Use managed switches.
For applications that require interconnecting devices separated by long distances (>100 m):	Use fiber optic products. Multimode fiber: Up to 2 km between nodes. Monomode fiber: Up to 15 km between nodes. Note: Depending on the fiber and the optical budget, could reach 4 km on multimode and 30 km on monomode.
For networks that require immunity to electromagnetic noise:	Use products with fiber optic ports.
For applications that require physical medium change:	Use transceivers or use switches with a combination of copper and fiber optic ports.
For applications that require external (IP67) mounting of the switch:	Use IP67 switches and cables.

PLSCADA Expert Architecture

The PLSCADA Expert is based on client-server architecture. It must be utilized at the task level. Each task works as a distinct client and/or server module, performing its own role, and interfacing with the other tasks through the client-server relationship.

PLSCADA Expert has these fundamental servers/clients:

- I/O server, which handles communications with I/O devices
- Alarm server which handles monitoring of alarm conditions
- Trend server which handles monitoring of trend values
- Report server type output
- Display client which handles displaying of HMI
- Web client

I/O Server roles:

- Manage communication with all I/O devices
- Manage the retrieval of waveform and date/time device update
- Support network monitoring by SNMP
- Support hot standby configurations, providing complete I/O device redundancy
- Support clustering and multi-processor distribution

Alarm Server roles:

- Manage monitoring of alarm conditions
- Support redundancy

Trend Server roles:

- Manage monitoring of trend values
- Archive data in the proprietary file format
- Support redundancy

Report Server roles:

- Support report management

Display Client roles:

- Manage HMI displays (graphic pages and waveforms)
- Handle genie animations
- Handle dynamic coloring of busbars

Web Display Client roles:

- Support HMI display within a Web browser

Network architecture considered in PLSCADA Expert

The following table illustrates some basic architecture types.

Abbreviations:

S: server

EGX: Gateway MODBUS TCP/IP MODBUS serial line (RTU)

CL: cluster

Devices: MODBUS communication devices

C: client

Architecture	Type	Machine	I/O Srvr	Alarm Srvr	Trend Srvr	Report Srvr	Display Client	Comments
	Standalone	One	Yes	Yes	Yes	Yes	Yes	
	Standalone Cluster	One or several clients (C) One or several servers (S)	No Yes	Yes Yes	Yes Yes	Yes Yes	Yes No	
	Basic redundant architecture (without communication redundancy)	One or several clients (C) One or several servers (S)	No Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes	If the load is large, alarm servers, trend servers, and repo servers can be separate from the I/O servers.
	Redundant architecture (with communication redundancy). The self-healing ring is the default solution. The dual link should be implemented, but the user must modify the setting.	One or several clients (C) One or several servers (S)	No Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes	The alarm and log viewers are limited in the display client. The viewer cannot display alarms on the same page for all clusters. Design one alarm page per cluster. (See the PLSCADA Expert System Integrators Guide for more information.)

Small Site: Up to 100 Devices (16,900 Tags): Standalone and Redundancy Architecture

These recommendations and requirements are true for a standalone, as well as a redundant architecture that includes a standby server. All requirements are the same for both PCs, except the client operating system is Windows Vista Business.

System Performance Goals

Attribute	Requirement
Number of computers	One; includes I/O server, alarm server, trend server, report server, one display client
Device limit	100
Tag limit	15,000
Speed of on-board alarm annunciation	2 seconds (six devices in the daisy chain; additional devices will add time)
Speed of PC-based alarm annunciation	1 second (for the specified system; but this can vary if system is set up differently)
Per-Device Tag Assignments	
Variable tags	169
Digital tags	56
Analog tags	113
Trend tags	31

Parameter recommendations

System Performance	Citect INI Parameters	Value
Polling rate	["DRIVER DEVICE"] CacheRefreshTime [«DRIVER DEVICE”. <ClusterName>. <PortName>] CacheRefreshTime [«DRIVER DEVICE”. <ClusterName>. <PortName>. <IODeviceName>] CacheRefreshTime The polling rate can be global to all devices or specific to a single I/O Device	1000 ms
Alarm scan time	[Alarm]ScanTime	500 ms
Page scan time	[Page Scan Time]	200 ms
Cache mode		Enabled
Driverwatchtime	[<<DRIVER DEVICE"]watchtime	2 s
Timeout	[<<DRIVER DEVUCE"]timeout	5 s
Retry	[<<DRIVER DEVICE"]retry	3

Medium site: up to 1,000 devices (169,000 Tags): Standalone and Redundancy Clusters

Several clusters, including one machine per cluster IO server, Alarm server, Trend Server, Report server, and one or more separate Web/display clients.

System Performance Goals

Attribute	Requirement
Number of computers	One, includes I/O server, alarm server, trend server, report server, one display client
Device limit	1000
Tag limit	169000
Speed of on-board alarm annunciation	2 seconds (six devices in the daisy chain; additional devices will add time)
Speed of PC-based alarm annunciation	1 second (for the specified system; but this can vary if system is set up differently)
Per-Device Tag Assignments	
Variable tags	169
Digital tags	56
Analog tags	113
Trend tags	31

Parameter recommendations

System Performance	Citect INI Parameters	Value
Polling rate	["DRIVER DEVICE"] CacheRefreshTime [«DRIVER DEVICE".<ClusterName>.<PortName>] CacheRefreshTime [«DRIVERDEVICE".<ClusterName>.<PortName>.<IODeviceName>]CacheRefreshTime The polling rate can be global to all devices or specific to a single I/O Device	1000 ms
Alarm scan time	[Alarm]ScanTime	500 ms
Page scan time	[Page Scan Time]	200 ms
Cache mode		Enabled
Driverwatchtime	[<<DRIVER DEVICE"]watchtime	2 s
Timeout	[<<DRIVER DEVUCE"]timeout	5 s
Retry	[<<DRIVER DEVICE"]retry	3

Redundancy

One self-healing ring and four machines including

- four IO servers (250 devices per redundant server)
- alarm server, trend server, report server redundant into two IO servers
- one or more separate machine Web/display client

The hard drive size for trend archive must be sized according to the number and sample rate of trend tags. For the hard drives where the trend data is written to, RAID 0 with at least three hard drives is recommended for maximum performance.

(*) Windows server 2003 SP1 is not reliable in terms of network stability.

Parameter recommendations

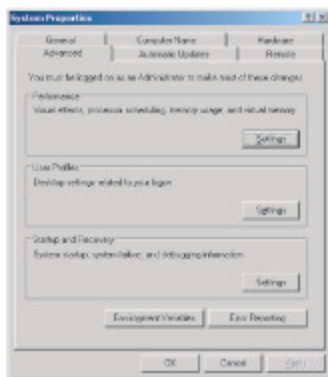
	Citect INI Parameters	Value
Polling rate	["DRIVER DEVICE"] CacheRefreshTime [«DRIVER DEVICE». <ClusterName>. <PortName>] CacheRefreshTime [«DRIVER DEVICE». <ClusterName>. <PortName>. <IODeviceName>] CacheRefreshTime System Performance The back polling rate can be global to all devices or tuned up to a specific I/O Device	1000 ms
Alarm scan time	[Alarm]ScanTime	500 ms
Page scan time	[Page Scan Time]	200 ms
WatchDogPrimary	[IOServer]WatchDogPrimary Check the connection periodically with the I/O devices.	1
ReadPool	[Lan]ReadPool	16000
WritePool	[Lan]WritePool	16000
Cache mode		Enabled
Driverwatchtime	[<<DRIVER DEVICE"]watchtime	2 s
Timeout	[<<DRIVER DEVICE"]timeout	5 s
Retry	[<<DRIVER DEVICE"]retry	3
Server Ports Configuration		
AlarmServPri	Use NetStat -n to find free port and specify AlarmServer Port, and don't use default value; set Publish Alarm Properties to TRUE	2076
Publish Alarm Properties (Pri)		2080
AlarmServStb		2083
Publish Alarm Properties (Stb)		2084
ReportServerPri		2075
ReportServerStb		2275
TrendServerPri		2077
TrendServerStb		2277
IOServerPrimary1	Port	2478

	Citect INI Parameters	Value
	Peer Port	2180
IOServerStandby1	Port	2178
	Peer Port	2181
IOServerPrimary2	Port	2102
	Peer Port	2202
IOServerStandby2	Port	3102
	Peer Port	3202
IOServerStandby3	Port	3103
	Peer Port	3203
IOServerPrimary3	Port	2103
	Peer Port	2203
IOServerStandby4	Port	3104
	Peer Port	3204
IOServerPrimary4	Port	2104
	Peer Port	2204

Windows Server Recommended

The RATIO servers' operating systems must be modified in terms of memory. The user should do the following:

1. Open the operating system's System Properties window, and click the Advanced tab.

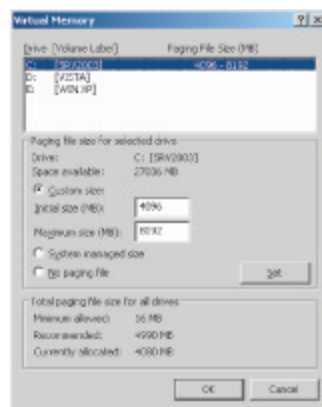


2. In the Performance box, click **Settings**.

The Performance Options window displays:



3. Select the Advanced tab.
4. From both the Processor scheduling box and the Memory usage box, select Programs.
5. Click Change. The Virtual Memory window displays:

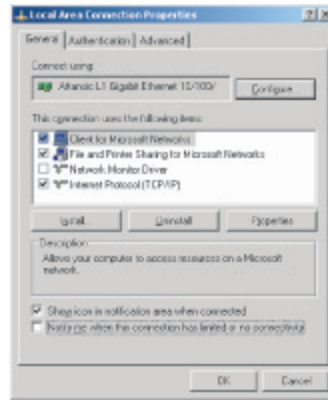


6. In the Paging file size for selected drive box, select Custom size.
7. Enter 4096 for the Initial size and 8192 for Maximum size. (Default values are 2046 and 4092.)
8. Click Set.
9. Click OK to save the changes. The window closes.
10. Click OK twice to close the Performance Options and System Properties windows.
11. Reboot the PC.

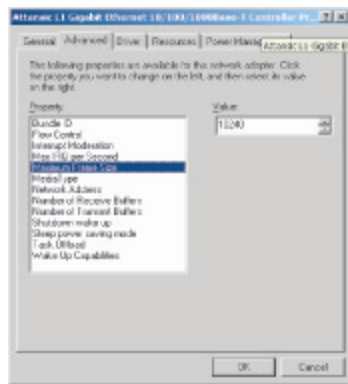
Network configuration

According to the important data flow between both servers, it is necessary to use gigabit range network Ethernet port and configure the maximum range allowed by the hardware configuration.

12. On the Ethernet switch, use the Gigabit Ethernet ports to connect the two servers.
Open the Local Area Connection Properties window:



13. Click **Configure**, then select the Advanced tab:



14. Highlight Maximum Frame Size, then set the value to 10240 (default value is 1514).

15. Click OK to save the changes.

CPU load balancing recommendation

In this architecture, five PCs are used. Each machine has a 4-core CPU processor. Four I/O servers are configured. Thus, each I/O server handles 250 devices.

One display client is added to view data. Report, Alarm, and Trend servers are located on first two servers.

Each of the following servers is configured as described below:

	Primary server	Standby server	Alarm server	Trend server	Report server	Client
S1	IOServerPrimary1	IOServerStandby2	AlarmServerPri	TrendServerPri	ReportServerPri	Client
S2	IOServerPrimary2	IOServerStandby1	AlarmServerStb	TrendServerStb	ReportServerStb	Client
S3	IOServerPrimary3	IOServerStandby4				Client
S4	IOServerPrimary4	IOServerStandby3				Client

PC and core repartition:

Server	Core0	Core1	Core2	Core3
S1	AlarmServerPri	ReportServerPri IOServerStandby2	TrendServerPri	IOServerPrimary1

Server	Core0	Core1	Core2	Core3
S2	AlarmServerStb	ReportServerStb IOServerStandby1	TrendServerStb	IOServerPrimary2
S3	IOServerPrimary3	IOServerStandby4	Client	
S4	IOServerPrimary4	IOServerStandby3	Client	
Server	Core0	Core1	Core2	Core3

Data Center with Branch Circuit Monitors, Circuit Monitors and MicroLogic Trip Units

Redundancy

Two clusters with six pairs of I/O servers (tested)

Configuration: two clusters, each with:

- three primary and three standby I/O servers
- two primary and two standby alarm servers
- two primary and two standby trend servers
- two primary and two standby report servers
- multiple display clients

Data Flow Limit to Achieve the Performance Requirement

Total Tag Assignment

Type of Tag	Number of Tags
Variable Tags	106916
Trend Tags	30034
Alarm Tags	3390
Total	140340

Parameters Recommended

Parameters	Localisation	Value
Back polling rate	In « Citect.ini »: ["DRIVER DEVICE"] CacheRefreshTime ["DRIVER DEVICE".<ClusterName>.<PortName>] CacheRefreshTime ["DRIVER DEVICE".<ClusterName>.<PortName>.<IODeviceName>] CacheRefreshTime The back polling rate can be global to all devices or tuned up to a specific I/O Device. Updating the cacherefresh time of the MicroLogic is recommended.	100 mx

Parameters	Localisation	Value
Alarm scan time	[Alarm[ScanTime iin <<°Citect.ini°>>]	500 ms
Page scan time	[Page Scan Time]Default in «°Citect.ini°»	200 ms
Cache mode		Eenabled
Driver watch time	[<<DRIVER DEVICE"]watchtime in <<°Citect.ini°>>]	2 s
Timeout	[<<DRIVER DEVICE"]timeout in <<°Citect.ini°>>]	20 s
Retry	[<<DRIVER DEVICE"]retry in <<°Citect.ini°>>]	3
Status register	[PWRMODBUS] statusregister	2

CPU load-balancing recommendation

In this architecture, four computers are used. There are three I/O servers, one display client, two report servers, two alarm servers, and two trend servers. Each of the following servers is configured as described below:

	Primary Server	Standby Server	Alarm Server	Trend Server
PC1 (Quad)	IOServerPri1 IOIServerPri2 IOServerPri3		AlarmPri1 AlarmStb2	TrendPri1 TrendStb2
PC2 (Dual)	IOServerPrimary2	IOServerStb1 IOServerStb2	AlarmPri3 AlarmStb1	TrendPri3 TrendStb1
PC3 (Single)	IOServerPrimary3		AlarmPri2 AlarmStb3	TrendPri2 TrendStb3
PC4 (Single)		IOServerStb3		

PC and core re-partition:

Server	Core0	Core1	Core2	Core3
PC1 (Quad)	IOServerPri1 Client	IOServerPri2	IOServerPri3 AlarmStb2 TrendStb2	TrendPri1 AlarmPri1
PC2 (Dual)	AlarmStb1 TrendStb1 Client IOServerStb2	TrendPri3 IOServerStb1 AlarmPri3		
PC3 (Single)	TrendPri2 TrendStb3 AlarmStb3 AlarmPri2 Clients			
PC4 (Single)	IOServerStb3 Client			

System Performance Evaluation

Introduction

This section describes how to obtain system response times for each of the architectures and communication types in your plant. It also describes the checks you should make on the devices and the network. These will ensure that the message load on a device does not exceed its abilities and the network load does not cause communication delays.

System response time

Introduction

The performance of PLS architecture is linked to the hardware and the application services used and to the parameters set for these services.

Hardware considerations are:

- network bandwidth
- resources of module or CPU with Ethernet
- embedded processor resources (PC, PLC or other CPUs)

Application services are:

- MODBUS industrial messaging handling service
- global data service, data scanning between PLCs
- I/O scanning service, data scanning of distributed I/O
- others (Web access, TCP open communication)

Because most of these parameters are linked, it may be difficult to determine the correct size of the architecture. However, we can provide the basic rules to achieve the performance.

What is the response time?

Response time must be defined in several cases:

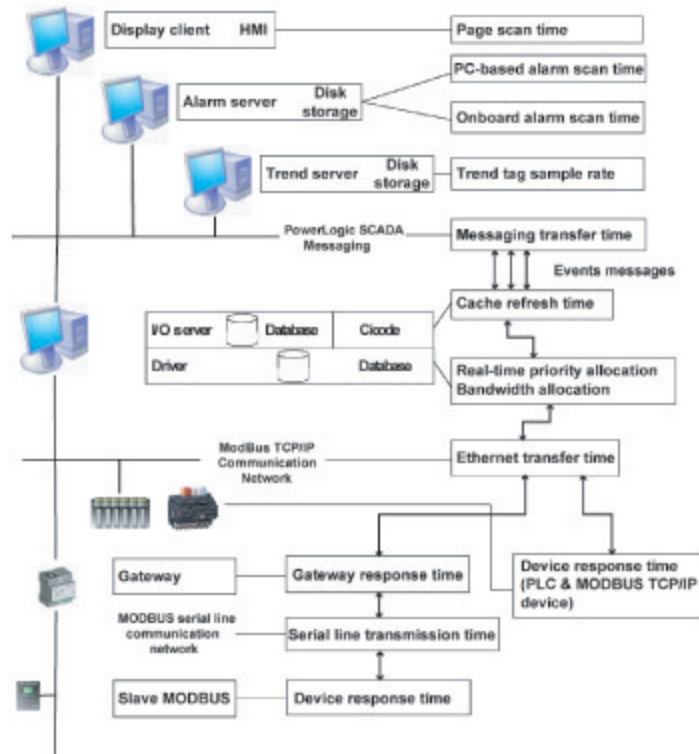
- project without control: the data provided by the devices are only monitored.
The response time is the time between retrieving data from a device and updating the database of the SCADA. The time depends on the data type and where in the SCADA the data will be displayed or stored. Digital values can be displayed in the HMI (Display client); alarm data is always stored in the disk drive. The time can be also different between the onboard alarms and PC-based alarms.
Analog values are similar: one of them is only displayed in the HMI, the other one is stored in disk drive (Trend).
- project with control: the data can be controlled by the operator and monitor.
Usually, the response time is the time between any orders given by an operator at the Supervision level and the display of corresponding status change. The time depends on the data type as described above.

How do we estimate response time?

The method to measure/calculate the response time is different, depending on the architecture and the device. In PLS, two communication architectures are provided:

- devices connected with the Ethernet network through a gateway
- devices connected directly with the Ethernet network

The following graphic illustrates the parameters that affect performance of the indicated Power SCADA component.

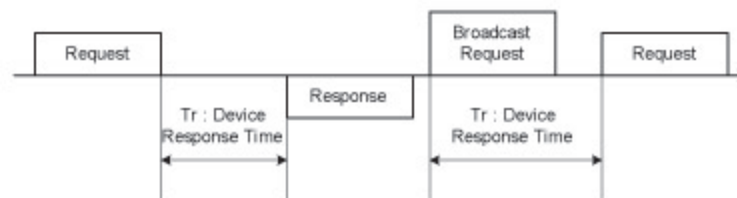


MODBUS Messaging Server Response Times

Devices Connected with the Ethernet Network through a Gateway

Device response time estimation

Each device type has a specific response time, depending on the hardware and firmware design. This time should not be ignored; it may be important for some devices. The best solution to estimate the device response time is to refer to the technical specification of manufacturer.

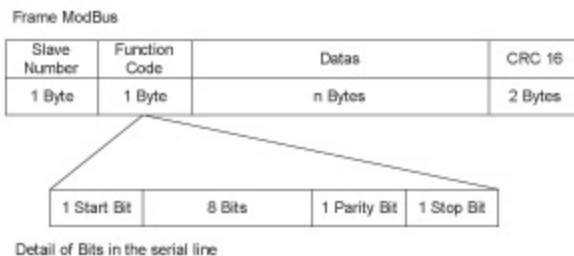


See Appendix A for a list of the round-trip response times for devices included in the PLSCADA Expert offer.

Calculation of Serial Line Transmission Time

The serial line response time is determined by the number of bits sent and the serial line speed. In the MODBUS protocol, the exchanges include two messages: a request by the master and a reply by the device (except the broadcast).

The requests and responses are structured in frame as described in the following graphic:



The maximum size of a frame is 255 bytes.

Frame MODBUS example; read N bits functions (1 and 2)

Request

Slave Number	Function Code	Data		CRC 16
01	01 or 02	Address of first bit to be read	Number of bits N to be read	2 bytes
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Response

Slave Number	Function Code	Data		CRC 16
01	01 or 02	Number of bytes read data	Data	CRC result
1 byte	1 byte	1 byte	(N+7) / 8 bytes	2 bytes

Frame MODBUS example; read N word functions (3 and 4)

Request

Slave Number	Function Code	Data		CRC 16
01	03 or 04	Address of first word to be read	Number of words N to be read	2 bytes
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Response

Slave Number	Function Code	Data		CRC 16
01	03 or 04	Number of bytes read data	Data	CRC result
1 byte	1 byte	1 byte	2N bytes	2 bytes

Refer to the MODBUS protocol specification for the exact number of bits per MODBUS message.

For the actual network transmission time, use:

$$(\text{the number of bits in the message}) \times (1/\text{baud rate})$$

For instance, the calculation of the number of bits in the message described above is:

Request for reading n bits at 9600 baud and transmission set up 8bits, Parity ODD, 1 stop

$$[8 \text{ bytes} * (8 \text{ Bits data} + 3 \text{ Bits})] \times (1/9600) \approx 10\text{ms}$$

Response with N = 2000 bits

$$[(5 \text{ bytes} + [(2000+7)/8]) * (8 \text{ Bits data} + 3 \text{ Bits})] \times (1/9600) \approx 300\text{ms}$$

Request for reading n words at 19200 baud and transmission set up 8bits,

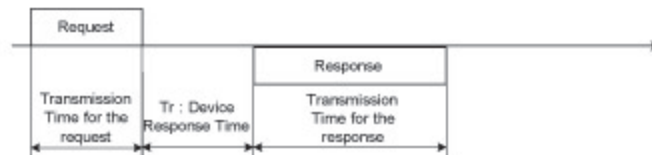
Parity ODD, 1 stop

$$[8 \text{ bytes} * (8 \text{ Bits data} + 3 \text{ Bits})] \times (1/19200) \approx 5\text{ms}$$

Response with N = 125 words

$$[(5 \text{ bytes} + (2 * 125)) * (8 \text{ Bits data} + 3 \text{ Bits})] \times (1/19200) \approx 150\text{ms}$$

Calculation of response time for one request and its response:



For example:

For one Sepam 40, to read 125 words, it takes:

$$\text{Request Time: } 5\text{ms} + \text{Tr : } 15\text{ms} + \text{Response time : } 150\text{ms} = 170\text{ms}$$

Gateway Response Times estimation

The response time for a gateway system can be calculated in one of two ways:

- Gateway with or without protocol conversion: actual calculation including response time of devices on the destination network and queues inside the gateway.
- Gateway using shared memory: For simple response time, the time to read the internal memory can be used. For a full system response for data in a destination device through the gateway and to a device on the source network, the time taken to read data into the gateway (often based on a timer) must be included.

In the PLSCADA Expert offer, the gateway with protocol conversion and queuing are only taken into account.

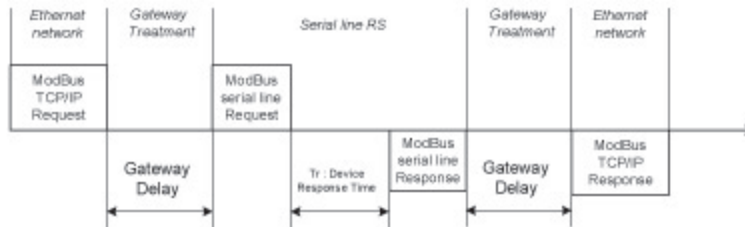
The following actions must occur to retrieve the device data through one gateway:

1. The gateway receives the request; the time that elapses from when the requesting device sends the request to when the gateway receives the request is dependent on the source network. For an Ethernet network, the delay is normally 0.05 ms. This delay is not significant in the response time calculation.
2. The gateway passes the request to the destination network; this is the gateway delay. If there is a queue, this time can be significant. Gateway delay is common if the two networks connected by the gateway have very different response times.
3. The request is received by the destination device; the delay is based on the ability of the destination network to transfer the message. For serial networks, it depends on the speed of the network (Refer to chapter: Calculation of Serial Line Transmission Time).
4. The request is processed by the destination device; this is dependent on the actual device (Refer to chapter: Devices response time estimation).
5. An answer is sent back to the gateway; the delay is based on the ability of the destination network to transfer the message. For serial networks it depends on the

speed of the network (Refer to chapter: Calculation of Serial Line Transmission Time).

6. The gateway passes the response back to the source network; this is the gateway delay. If there is a queue, this time can be significant.
7. The response is received by the requesting device; the delay from the time the requesting device sends the request to the time the gateway receives the request is dependent on the source network. For an Ethernet network, the delay is normally 0.05 ms. This delay is not significant in the response time calculation.

In steps that have a delay, the system response time is the total of all the delays.



See Appendix A for a list of round-trip response times for devices in PLSCADA Expert.

Two items complicate the calculation of the system response time:

- a queue of messages in the gateway due to time-outs or multiple queries
- the time-out of a message on the destination network, which is applicable in a network that must hold all future messages until the current message has timed out (e.g., MODBUS serial line)

Because the EGX gateways do not support retries, you must implement retries and timeouts (if you require such behaviours) in the PLSCADA Expert software. The default behaviour is "no retries." See the System Integrators Manual for details about parameters.

Calculation of the Number of Supported Devices per Gateway

The system response time is determined by the number of requests sent through the gateway; the more requests that are sent, the slower the overall response time for all devices.

To determine the number of devices on a system, first determine the total number of MODBUS requests to gather all the data. The best response time the system can give is:

Number of requests x (time to transmit the request on the serial line + response time of the serial device + time to transmit the response on the serial line + ~20 ms)

The average response time for a serial device is 200 ms, but may vary from 50 to 500 ms. The time needed to transmit the request/response depends on the speed of the network and the MODBUS RTU/ASCII setting.

- RTU is much faster because fewer bytes are transferred.
- An average MODBUS read request, at 9600 baud, is ~ 5 ms.
- A maximum response is ~ 100 ms.

The total best-case system response would therefore be:

5 ms (request) + 200 ms (serial device response) + 100 ms (response) + 20 ms = ~300 ms/request

For eight MODBUS devices with two requests each, the best-case response time to get data from the system is 16 x 300 ms = 4.8 s.

This is too long for most system users to wait for a response, so the number of devices per gateway needs to be reduced. However, with a faster serial device response time, calculating the total best-case gateway response would use the formula:

$$-5 \text{ ms (request)} + 50 \text{ ms (serial device response)} + 20 \text{ ms (response)} + 20 \text{ ms} = \sim 100 \text{ ms/request}$$

For eight MODBUS devices with two requests each, the best-case response time would then be an acceptable $16 \times 100 \text{ ms} = 1.6 \text{ s}$.

To improve the system response time, limit the number of requests being sent through the gateway by limiting the number of devices connected to each gateway. The devices having a good response time should not be in the same serial line with the slowest devices.

For round-trip times tested for CM, PM, S40 and MicroLogic devices, see Appendix A.

Devices connected directly with the Ethernet network.

MODBUS TCP/IP Server Response Times

Two methods must be used to determine the MODBUS TCP/IP server response time:

- measured response times for simple devices (e.g., CM4 series with the ECC)
- calculation based on system operation; for more complex devices like Quantum, Premium, or M340 PLC systems

The measured response times for CM 4 MODBUS server devices are described in the next chapter.

The MODBUS server response times for the following devices are not fixed and need to be calculated:

- Premium PLC system
- Quantum PLC system
- M340

Refer to PLC manufacturer documentation.

CM4 response time

CM4 response time was measured under controlled conditions and may vary from results obtained in the field. The results are located in Appendix A (Circuit Monitor 4000 Round-Trip Response Times). The results are valid only when the overall limits of device communications are not exceeded on the client or the server.

Calculation of the Ethernet Timeout

If the timeout of a request is included, calculating the worst-case gateway response time gives the required value for the Ethernet timeout field:

Ethernet timeout = timeout of a serial line request x number of serial line retries x number of requests sent to the gateway

If this timeout calculation is not used, and the value in the field is too slow, the failure of one or more serial devices can cause Ethernet requests to other serial devices to timeout due to the delay caused by the incorrect value.

MODBUS Messaging Client Response Times

The MODBUS messaging client response time is part of the total MODBUS messaging system response time. There are two methods for determining the MODBUS client

response times:

- considering the entire MODBUS messaging system (client and server) as one unit
- calculating the system component times separately

In the first method, the total system response time from client request to server response is measured. The second method provides more specific results for a particular system than the total time graphs used in the first method.

Measured response times for several of Schneider Electric's MODBUS client systems are based on various server response times. These response times were measured under controlled conditions and may vary from results obtained in the field. The graphs in this appendix are valid only when the overall limits of device communications are not exceeded on the client or the server.

The following devices may require the calculation of the MODBUS messaging client time as it is not fixed:

- Quantum PLC system
- Premium PLC system
- M340 PLC system
- SCADA system
- OFS server

SCADA system response time

The response time depends essentially on the servers that are set up within architecture, the PLS driver setup, and the tag prioritization on the configuration tools.

The next chapter describes the parameters used in performance tuning the system.

System Performance Tuning

Dataflow prioritization

General Information

Communication to the device is achieved through a Prioritised Data Scheduler that optimizes performance and interleaves requests. The dataflow for all devices or ports (PLSCADA Expert port) is separated into four categories: events, commands, real-time, and waveform. Bandwidth can be allocated for each category.

You can tune bandwidth allocation to perform the dataflow of each category and device.

All the tags included in the real-time category can have a priority assigned which determines the scan rate at which that data is refreshed.

The profile including the tag should be tuned to prioritize the dataflow between:

- tag categories (real-time / alarm / “control / reset tag”)
- waveform dataflow
- real-time tags (e.g. refresh the digital value more quickly than the analog value)

Bandwidth allocation

Bandwidth allocation for type of data

You can allocate bandwidth for the different types of data as desired. The parameters to perform this are as follows:

[Parameter]	[Default Value]	[Parameter Type]
Evenhandedly	25	integer
WaveformsBandwidth	12	integer
RealtimeBandwidth	50	integer

The percentage of bandwidth allocated to each queue is the ratio of an individual queue's value when compared to the total sum of defined bandwidths. The default values have a sum of 100 for ease of reference. Unused bandwidth will be shared among the categories.

This means that, if event data, waveform data, real time data, and commands, are all being sent/received at the same time, in 100 packets we would see 50 packets for real time data, 25 for event data, etc. To increase/decrease the data transfer rate of the different categories of data, you can adjust the allocated bandwidth.

You can configure bandwidth at the port level, but not at the device level.

Sepam Example:

Assuming a device response time of 250ms (the driver can retrieve four requests per second per device), and the EventBandwidth of 25%: the maximum time required to retrieve the first event is one second for Sepam devices (they require one packet to detect a new alarm before it can be retrieved). Sepam devices retrieve subsequent events (where a cascade of events has occurred), four events/second.

Note: Sepam events are read four at a time, thus Sepam events are retrieved at four times the rate of the other devices (MicroLogic, CM4, and PM8) when there are many events.

If the Eventbandwidth is doubled to 50%, the number of events retrieved per second will be increased by a factor of two. In this example, it would be possible to retrieve eight events/second.

Setting a percentage bandwidth for the different types of data makes the system predictable in regards to the device response time. If the response time is doubled to 500ms, the number of events retrieved per second is reduced by a factor of two. The time to retrieve High/Normal/Low priority tags would all double.

Example:

[SEPAM40]	
EventBandwidth	30
WaveformsBandwidth	5
CommandsBandwidth	15
RealtimeBandwidth	50

For the Sepam without waveform and control:

[SEPAM40.MYCLUSTER.PORT_1]	
EventBandwidth	50
WaveformsBandwidth	0
CommandsBandwidth	0
RealtimeBandwidth	50

Bandwidth allocated for the device

This parameter allows you to configure the ratio of bandwidth assigned to each device sharing a port. This parameter can be configured only at the device level. By default, the bandwidth allocated for the device is divided equally. You can adjust the parameter for each device to increase its bandwidth.

[Parameter]	[Default Value]
BandwidthAllocation	<Equal split>

Examples:

[SEPAM40.MYCLUSTER.PORT1_BOARD1.DEVICE_A]	
BandwidthAllocation	70

[SEPAM40.MYCLUSTER.PORT1_BOARD1.DEVICE_B]	
BandwidthAllocation	30

This parameter allows the driver to increase the scan rate for a selected device on a port.

Tag scan rates

You can configure each real-time tag at scan rate priority level as low, normal, or high. Parameters exist to adjust the relative scan rates of the high and low priority tags in comparison to the nominal tag scan rate.

In the Profile Editor, all tags have a predefined priority which can be adjusted.

For example, a variable tag could have the following address:

"m:275:u1;N:-1;E:2 ;L:P:33"

E:X defines the priority, where X is a value from 1 – 3.

There are three priorities:

- E:1, High priority (Priority Queue 10002)
- E:2, Normal priority (Priority Queue 10001)
- E:3, Low priority (Priority Queue 10000)

The address tag above has a normal priority.

The desired tag scan rate refers to the rate at which real-time registers within the device are scanned. To configure this rate, use the "cacheRefreshTime" parameter:

CacheRefreshTime

[Parameter]	[Default Value]	[Parameter Type]
cacheRefreshTime	500	milliseconds

This parameter controls the maximum rate at which the driver will attempt to repopulate its cache.

You can configure RefreshTime at the driver or device level.

Example:

[SEPAM40]
cacheRefreshTime = 1000
[SEPAM40.MYCLUSTER.PORT1_BOARD1.FAST_SEPAM]
cacheRefreshTime = 200
[SEPAM40.MYCLUSTER.PORT1_BOARD1.UNIMPORTANT_DEVICE]
cacheRefreshTime = 5000

However, if the network cannot support this refresh time, the network dynamics will determine this rate. The number of tags configured for each priority type, and the desired relative scan rate of each priority, determine the tag scan rate of "Nominal Tag Scan Rate."

The system allows users to configure the ratio of different tag priority update rates relative to one another:

[Parameter]	[Default Value]	[Parameter Type]
HighScanRate	50	percent relative to nominal
LowScanRate	200	percent relative to nominal

Using the default parameters, the high priority tags are refreshed twice as fast as the normal priority tags, and the low priority tags are refreshed at half the rate of the normal priority tags. You can configure these parameters at the port level and higher.

Using the default settings and a nominal tag refresh rate of one second:

Low Priority Tagh Refresh	2000 ms
Normal Priority Tag Refresh	1000 ms
High Priority Tag Refresh	500 ms

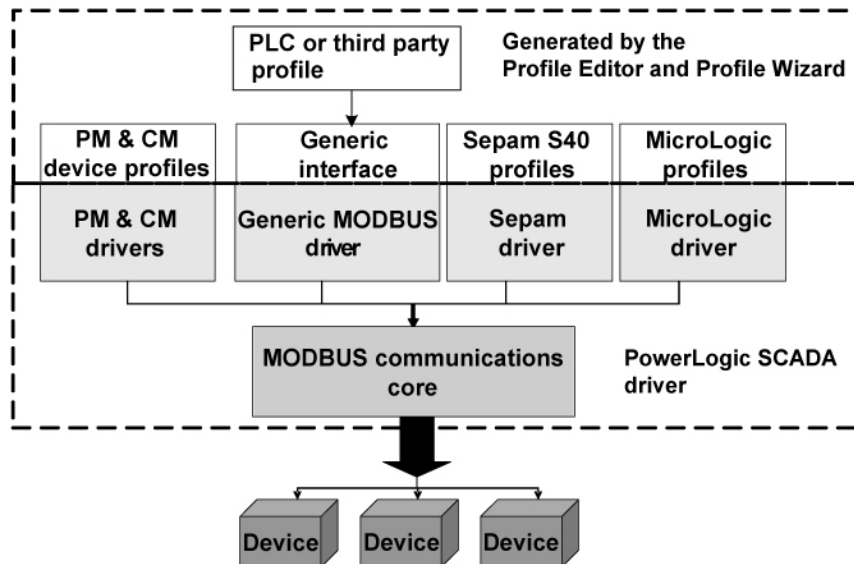
Example:

[PM870.MYCLUSTER.PORT1]	
HighScanRate	25
LowScanRate	500

Driver Optimization

Introduction

To support the range of devices required in a PLSCADA Expert project, there are four drivers. Three of these drivers provide the functionality of the different “standard” device families (power meter/circuit monitor, Sepam, MicroLogic). The final driver is generic MODBUS, which supports third-party devices. This diagram describes the general structure of the drivers:



General Information

To have maximum efficiency when reading blocks of data from a device, the drivers are able to optimize request blocks to efficiently read data from the device. The drivers can also use any additional functionality provided by the devices for this purpose, such as the scattered read function for the PowerLogic devices. Driver optimizations have the following features:

- tailored block reads for specific device profiles, where a device profile exists
- use of scattered reads when supported by the device
- optimized dynamic blocking for non-profile based devices
- the ability to protect ranges of registers from being read as part of a block (necessary for devices when reading certain register ranges will result in an error from the device)
- the ability to set up device parameters to minimize the impact of communication failure

The following sections focus on the scattered read and the management of communication failure.

Packet Blocking Optimization

You can configure parameters to optimize the MODBUS packets that are created to collect data from the device.

NOTE: Note: Sepam devices do not support the functionality and have pre-configured blocks that are already optimized.

The parameters that control the blocking are as follows:

[Parameter]	[Default Value]	[Parameter Type]
gunlocks	20	integer
maxBlockSize	124	integer
percentBlock Fill	50	percentage
enableScatteredReads	1	Boolean flag

EnableScatteredReads:

[Parameter]	[Default Value]	[Parameter Type]
enableScatteredReads	0	Boolean flag

NOTE: The default is 0 for the Generic Power MODBUS driver; the default is 1 for the PowerLogic driver.

This causes the driver to use the 'scattered read' extension that can help improve blocking. Scattered reads can only be configured at the device level.

Example:

[PWRMODBUS.MYCLUSTER.PORT1_BOARD1.DEVICE_A]	
enableScatteredReads	1
[PWRMODBUS.MYCLUSTER.PORT1_BOARD1.DEVICE_B]	
enableScatteredReads	0

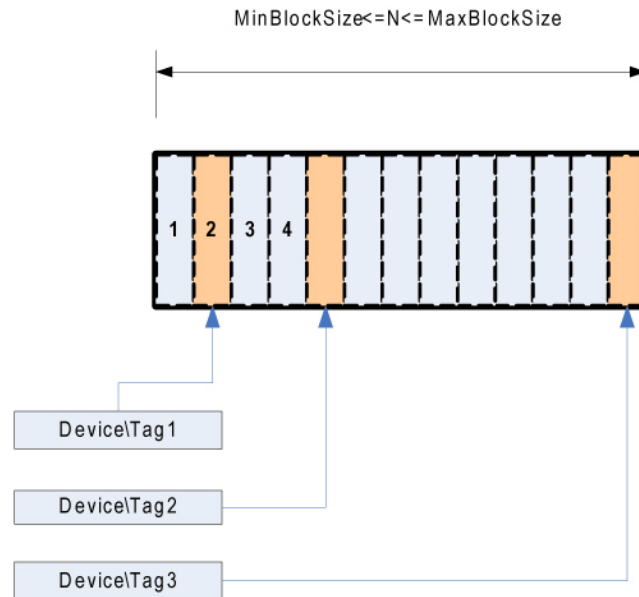
PercentBlockFill:

[Parameter]	[Default Value]	[Parameter Type]
percentBlockFill	50	percentage

This parameter defines the maximum percentage of configured registers contained in a block before the driver creates fixed block(s) instead of scattered blocks.

The following figure illustrates how a block of N registers can be constructed. If M<N registers is configured, the block builder can either build a scattered block or multi-register block. If M/N*100% is less than PercentBlockFill, scattered register blocks will

be built. If the percentage of configured registers is equal to or greater than PercentBlockFill, a multi-register block is created instead.



Example:

[PM870.MYCLUSTER.PORT1_BOARD1.PM_DEVICE]	
percentBlockFill	50
[CM4000.MYCLUSTER.PORT1_BOARD1.CM_DEVICE]	
percentBlockFill	80

MaxBlockSize:

[Parameter]	[Default Value]	[Parameter Type]
maxBlockSize	124	integer

This parameter defines the maximum number of registers that can be read in a single request. By default, this is 124, but some third party devices can read more than this. MaxBlockSize can only be configured at the device level.

Example:

[PWRMODBUS.MYCLUSTER.PORT1_BOARD1.DEVICE_A]	
maxBlockSize	1024

MinBlockSize:

[Parameter]	[Default Value]	[Parameter Type]
minBlockSize	20	integer

This parameter defines the minimum number of registers to read as a fixed block before the block builder will add those registers to a scattered block. If latency is low,

and scattered reads are expensive, this value should be lower. If latency is high, or scattered reads are inexpensive, it is better to set this value higher. Only applicable when scattered reads are enabled. MinBlockSize fill can only be configured at the device level.

Example:

[PM870.MYCLUSTER.PORT1_BOARD1.LOW_LATENCY_DE VICE]	
minBlockSize	10

[CM4000.MYCLUSTER.PORT1_BOARD1.HIGH_LATENCY_DE VICE]	
minBlockSize	100

Device communication failure optimization

If a device on a multi-drop connection loses its communication link, the other devices on the drop could be affected due to the time latency. The driver provides many parameters to reduce the impact of one device communication failure.

InitUnitCheckTime:

[Parameter]	[Default Value]	[Parameter Type]
initUnitCheckTime	120	seconds

This parameter controls how long the driver will wait before attempting to bring a device online after it has gone offline. Decrease this value to bring offline devices back into service in a shorter period of time. In a multi-drop scenario, this time should be relatively long, to prevent initUnit requests from stalling communication to the rest of the devices on that port. These parameters can be configured at the device level and higher.

Example:

[SEPAM40]	
initUnitCheckTime	= 5

[SEPAM40.MYCLUSTER.PORT_1]	
initUnitCheckTime	=120

Timeout:

[Parameter]	[Default Value]	[Parameter Type]
Timeout	5000	milliseconds

The 'timeout' parameter controls how long the driver will wait for a response from a device before setting that device offline. This value should be greater than the device/gateway timeout period.

NOTE: A timed out request will not be retried. This is because TCP is a guaranteed transport mechanism, and the lack of a response indicates device failure or communications failure to that device. A device connected via a gateway should use the gateway's retry mechanism. These parameters can be configured at the device level and higher.

Example:

[SEPAM40]	
Timeout	= 1000

[SEPAM40.MYCLUSTER.PORT_1_BOARD1.SLOW_SEPAM]	
Timeout	=5000

Retry:

[Parameter]	[Default Value]	[Parameter Type]
Retry	3	integer

The "retry" parameter defines the number of retry attempts for specific MODBUS requests. Retries only occur in response to these MODBUS errors:

SLAVE_DEVICE_BUSY_EXCEPTION	0x06
MEMORY_PARITY_ERROR_EXCEPTION	0x08
GATEWAY_TARGET_DEVICE_FAILED_TO_RESPOND_EXCEPTION:	0x0B

These parameters can be configured at the device level and higher.

Example:

[SEPAM40]	
retry	= 1

[SEPAM40.MYCLUSTER.PORT1_BOARD1.SEPAM_DEVICE]	
retry	= 5

Appendix A: Device Response Times

This appendix describes the response times for various device types, organized according to the protocols they use.

Click one of the following links to view devices within that protocol.

- [DNP3 Protocol on page 39](#)
- [IEC 61850 Protocol on page 40](#)
- [Modbus Protocol on page 46](#)
- [Multiple Protocols on page 60](#)

DNP3 Protocol

ION 7650 (DNP3) Response Times

This section describes the response times for an ION 7650 using DNP3 protocol.

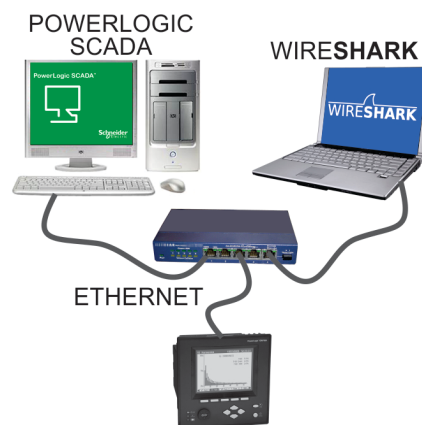
System Configuration

The figure below illustrates the test configuration. We used an isolated network with an Ethernet hub. On one computer, we installed PowerLogic SCADA 7.20 to poll the ION 7650 device. On the other computer, we installed Wireshark to monitor communication times.

ION Device Information

FAC1 Revision: B0042-7650

FAC1 Template: 7650_FAC-PQ-61850_V3.5.0.0.0.



Test Procedure

We used the standard ION 7650 profile. The time represents one full read of system tags from the first variable tag request, to the device response, to the next request for the same tag. Response times were recorded with one ION 7650 in the system.

Test Results

Average Response Time
9.694 seconds

For I/O devices: both minimum update rate and background rate are set to 10 seconds.

IEC 61850 Protocol

Circuit Monitor 4000 (G3200/IEC 61850) Response Times

This section describes the response times for a CM4000 using G3200 gateway and IEC 61850 protocol.

System Configuration

The figure below illustrates the test configuration. We used an isolated network with an Ethernet hub. On one computer, we installed PowerLogic SCADA 7.20 to poll the devices. On the other computer, we installed WireShark to monitor communication times.

Communication Settings.

RS-485, 4 wire
19,200 Baud
Even Parity
Timeout: 4 seconds

Gateway Information

Model Number: G3200 PL
Firmware Version: 1.4
Hardware Version: RD5
BRCB (buffered report control blocks): Used

Manufacture Date

13 November, 2008

Communications Cable Information

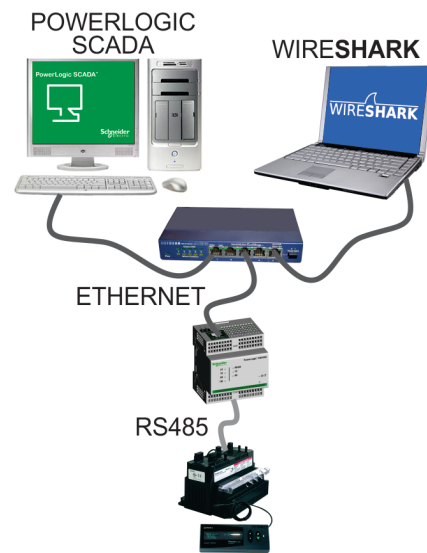
Belden 8723, 2 pair, 22 AWG, Shielded Cable

Communications Cable Length

Less than 10 feet total

Device Information

Model: CM4000T
OS Rev: 12.860
Manufacture Date: 05 June, 2001



Test Procedure

We used the standard CM4000 profile. Times represent one full read of system tags from the first variable tag request, to the device response, to the next request for the same tag. Response times were recorded with one CM4000 in the system.

Test Results

Average Response Time
5.513 seconds

ION 7650 (IEC 61850) Response Times

This section describes the response times for an ION 7650 using IEC 61850 protocol.

System Configuration

The figure below illustrates the test configuration. We used an isolated network with an Ethernet hub. On one computer, we installed PowerLogic SCADA 7.20 to poll the ION 7650 device. On the other computer, we installed WireShark to monitor communication times.

ION Device Information

FAC1 Revision: B0042-7650

FAC1 Template: 7650_FAC-PQ-61850_V3.5.0.0.0

BRCB (buffered report control blocks): Used



Test Procedure

We used the standard ION 7650 profile. The time represents one full read of system tags from the first variable tag request, to the device response, to the next request for the same tag. Response times were recorded with one ION 7650 in the system.

Test Results

Average Response Time
0.515 seconds

MicroLogic Type P (G3200/IEC 61850) Response Times

This section describes the response times for a MicroLogic Type P using a G3200 gateway and IEC 61850 protocol.

System Configuration

The figure below illustrates the test configuration. We used an isolated network with an Ethernet hub. On one computer, we installed PowerLogic SCADA 7.20 to poll the MicroLogic device. On the other computer, we installed WireShark to monitor communication times.

Communication Settings

RS-485, 4 wire.

19,200 Baud

Even Parity

Timeout: 3 seconds

Gateway Information

Model Number: G3200PL

Firmware Version: 1.4

Hardware Version: RD5

BRCB (buffered report control blocks): Used

Manufacture Date

24 November, 2009

Communications Cable information

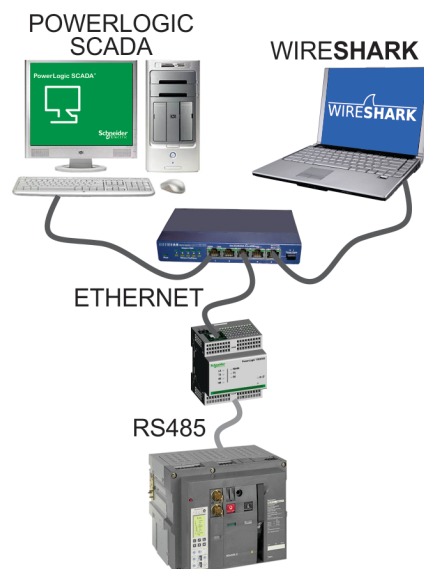
Belden 8723, 2 pair, 22 AWG, Shielded Cable

Communications Cable Length

Less than 10 feet total

Device Information

MicroLogic Type P 6.0



Test Procedure

We used the standard MicroLogic P profile. The time represents one full read of system tags from the first variable tag request, to the device response, to the next request for the same tag. Response times were recorded with one MicroLogic P in the system.

Test Results

Communication Profile (Cache)	Average Response Time
Deactivated	0.393 seconds
Activated	4.997 seconds

Power Meter 850 (G3200/IEC 61850) Response Times

This section describes the response times for the Power Meter 850 using G3200 gateway and IEC 61850 protocol.

System Configuration

The figure below illustrates the test configuration. We used an isolated network with an Ethernet hub. On one computer, we installed PowerLogic SCADA 7.20 to poll the device. On the other computer, we installed WireShark to monitor communication times.

Communication Settings.

RS-485, 2 wire

19,200 Baud

Even Parity

Timeout: 4 seconds

Gateway Information

Model Number: G3200 PL

Firmware Version: 1.4

Hardware Version: RD5

BRCB (Buffered Report Control Blocks): Used

Manufacture Date

13 November, 2008

Communications Cable Information

Belden 8723, 2 pair, 22 AWG, Shielded Cable

Communications Cable Length

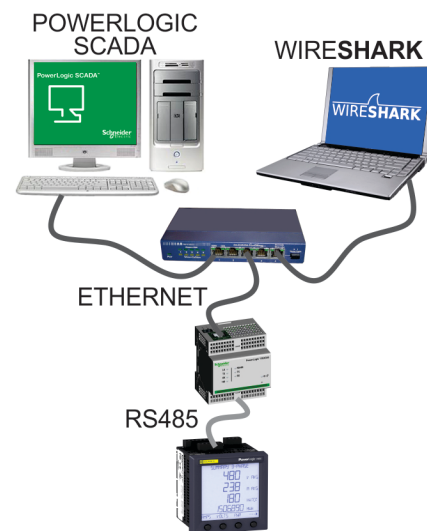
Less than 10 feet total

Device Information

Model: PM850

OS Rev: 10.561

Manufacture Date: 17 October, 2003



Test Procedure

We used the standard Power Meter 800 profile. Times represent one full read of system tags from the first variable tag request, to the device response, to the next request for the same tag. Response times were recorded with one power meter in the system.

Test Results

Average Response Time
2.390 seconds

Sepam T87 (ACE850/IEC 61850) Response Times

This section describes the response times for a Sepam T87 using ACE850 gateway and IEC 61850 protocol.

System Configuration

The figure below illustrates the test configuration. We used an isolated network with an Ethernet hub. On one computer, we installed PowerLogic SCADA 7.20 to poll the Sepam devices. On the other computer, we installed WireShark to monitor communication times.

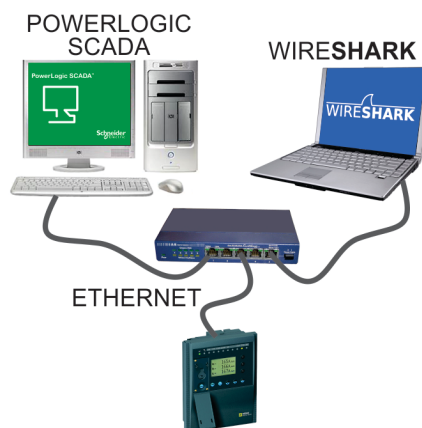
ACE850 Gateway information.

Model Number: ACE850TP
Firmware Version: V 1.00
Hardware Version: V 1.00
BRCB (Buffered Report Control Blocks): Used

Device Model: T87

Software: Version 6.01

Communication: v 1.00



Test Procedure

We used the standard Sepam 80 profile. Times represent one full read of system tags from the first variable tag request, to the device response, to the next request for the same tag. Response times were recorded with one Sepam in the system.

Test Results

Average Response Time
0.256 seconds

Sepam T87 (ECI850/IEC 61850) Response Times

This section describes the response times for a Sepam T87 using ECI850 gateway and IEC 61850 protocol.

System Configuration

The figure below illustrates the test configuration. We used an isolated network with an Ethernet hub. On one computer, we installed PowerLogic SCADA 7.20 to poll the Sepam devices. On the other computer, we installed WireShark to monitor communication times.

Communication Settings

RS-485, 2 wire
19,200 Baud
Even Parity
Timeout: 300 milliseconds

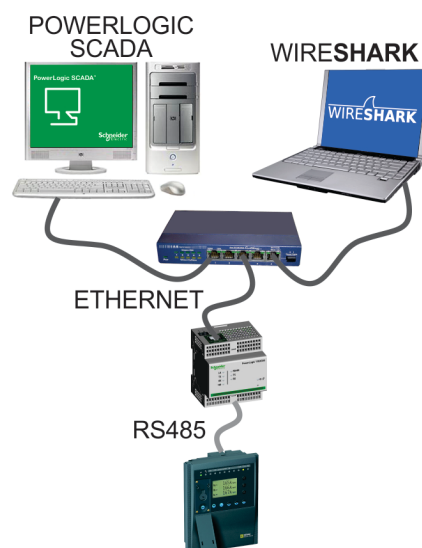
Gateway information

Model Number: ECI850MG
Firmware Version: 1.21
Hardware Version: RD5
BRCB (Buffered Report Control Blocks): Used

Manufacture Date

12 August, 2008

Device Model	Software
T87	Version 6.01



Device Model: T87

Software: Version 6.01

Test Procedure

We used the standard Sepam 80 profile. The time represents one full read of system tags from the first variable tag request, to the device response, to the next request for the same tag. Response times were recorded with one Sepam T87 in the system.

Test Results

Average Response Time
5.005 seconds

Modbus Protocol

Circuit Monitor 4000 (ECC/Modbus) Response Times

This section describes the response times for a CM4000 using an ECC gateway and Modbus protocol.

System Configuration

The figure below illustrates the test configuration. We used an isolated network with an Ethernet hub. On one computer with Windows 2008 R2, we installed PowerLogic SCADA 7.20 to poll the device. On the other computer, we installed Wireshark to monitor communication times.

Communication Settings

RS-485
19,200 Baud
Even Parity
Timeout: 3 seconds

ECC Information

Model Number: 21
Firmware Version: 5.500
Hardware Version: A1

Manufacture Date

August 8, 2000

Communications Cable Information

Belden 8723, 2 pair, 22 AWG, Shielded Cable

Communications Cable Length

Total cable length: less than 10 feet



Test Procedure

We used the standard CM4000 profile with 100 tags. Times represent one full read of system tags from the first variable tag request, to the device response, to the next request for the same tag. See the table below for results.

Test Results

Number of Tags in Profile	100 Tags
Average Time (milliseconds)	4-wire: 99

Circuit Monitor 4000 (EGX/Modbus) Response Times

This section describes the response times for a CM4000 using an EGX gateway and Modbus protocol.

System Configuration

The figure below illustrates the test configuration. We used an isolated network with an Ethernet hub. On one computer with Windows 2008 R2, we installed PowerLogic SCADA 7.20 to poll the device. On the other computer, we installed WireShark to monitor communication times.

Communication Settings

RS-485
19,200 Baud
Even Parity
Timeout: 3 seconds

ECC Information

Model Number: unknown
Firmware Version: unknown
Hardware Version: unknown

Manufacture Date

unknown

Communications Cable Information

Belden 8723, 2 pair, 22 AWG, Shielded Cable

Communications Cable Length

Total cable length: less than 10 feet



Test Procedure

We used the standard CM4000 profile with 100 tags. Times represent one full read of system tags from the first variable tag request, to the device response, to the next request for the same tag. See the table below for results.

Test Results

Number of Tags in Profile	100 Tags
Average Time (milliseconds)	4-wire: 97.3

MicroLogic Type P (EGX/Modbus) Response Times

This section describes the response times for a MicroLogic Type P using an EGX300 gateway and Modbus protocol.

System Configuration

The figure below illustrates the test configuration. We used an isolated network with an Ethernet hub. On one computer, we installed PowerLogic SCADA 7.20 to poll the devices. On the other computer, we installed WireShark to monitor communication times.

Communication Settings

RS-485, 2 wire

19,200 Baud

Even Parity

Timeout: 3 seconds

Gateway Information

Model Number: EGX300SD

Firmware Version: 3.700

Hardware Version: unknown

Manufacture Date

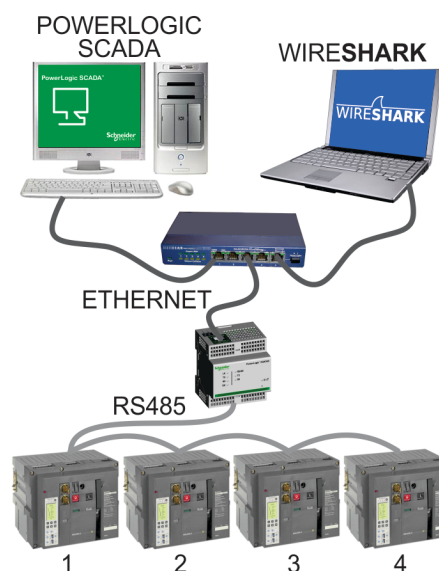
unknown

Communications Cable Information

Belden 8723, 2 pair, 22 AWG, Shielded Cable

Communications Cable Length

Less than 10 feet total



Device Number	Device Model
1	MicroLogic Type P
2	MicroLogic Type P
3	MicroLogic Type P
4	MicroLogic Type P

Test Procedure

We used the standard MicroLogic Type P profile. Times represent one full read of system tags from the first variable tag request, to the device response, to the next request for the same tag. Response times were recorded first with Device 1 in the system, then with Device 1 and Device 2, and so on. See the table below for results.

Test Results

	Response Time in Seconds			
	Device 1	Device 1+2	Device 1+2+3+4	Device 1+2+3+4
Average	1.074	2.145	3.305	4.357

MicroLogic Type P (G3200/Modbus) Response Times

This section describes the response times for a MicroLogic Type P using a G3200 gateway and Modbus protocol.

System Configuration

The figure below illustrates the test configuration. We used an isolated network with an Ethernet hub. On one computer, we installed PowerLogic SCADA 7.20 to poll the MicroLogic device. On the other computer, we installed WireShark to monitor communication times.

Communication settings:

RS-485, 4 wire

19,200 Baud

Even Parity

Timeout: 3 seconds

Communication Profile

Not activated

Gateway Information

Model Number: G3200PL

Firmware Version: 1.4

Hardware Version: RD5

Manufacture Date:

24 November, 2009

Communications Cable Information

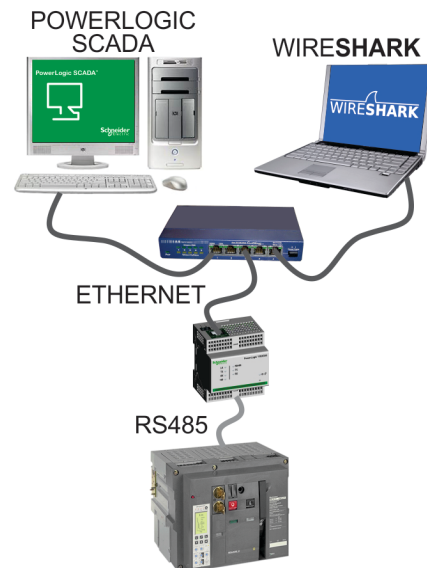
Belden 8723, 2 pair, 22 AWG, Shielded Cable

Communications Cable Length

Less than 10 feet total

Device Information

MicroLogic Type P 6.0



Test Procedure

We used the standard MicroLogic P profile. The time represents one full read of system tags from the first variable tag request, to the device response, to the next request for the same tag. Response times were recorded with one MicroLogic P in the system.

Test Results

Average Response Time: 5.279 seconds

Power Meter 800 (ECC/Modbus) Response Times

This section describes the response times for a Power Meter 800 using an ECC gateway and Modbus protocol.

System Configuration

The figure below illustrates the test configuration. We used an isolated network with an Ethernet hub. On one computer, we installed PowerLogic SCADA 7.20 to poll the device. On the other computer, we installed Wireshark to monitor communication times.

ECC Gateway Information.

Model Number: unknown
Firmware Version: unknown
Hardware Version: unknown

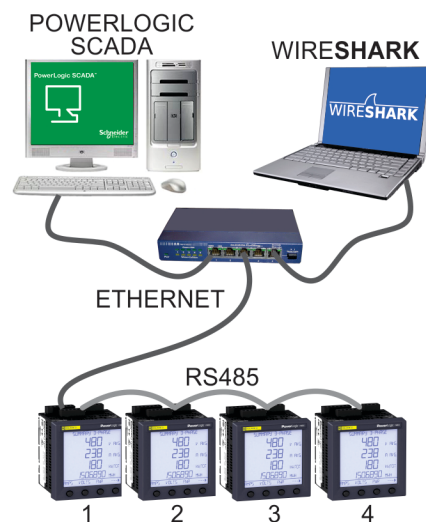
Manufacture Date

unknown

Citect.ini Modification

[PLOGIC870]

CacheRefreshTime=0



Test Procedure

We used the standard Power Meter 800 profile. Times represent one full read of system tags from the first variable tag request, to the device response, to the next request for the same tag. Response times were recorded first with the device connected directly to the PM8-ECC then with all four devices. See the table below for results.

Test Results

(Average Response Times)

Single PM8 Connected to ECC: 0.203 Second

4 Devices Communicating with One PM8-ECC: 1.688 Seconds

Power Meter 800 (EGX/Modbus) Response Times

This section describes the response times for a Power Meter 800 using an EGX300 gateway and Modbus protocol.

System Configuration

The figure below illustrates the test configuration. We used an isolated network with an Ethernet hub. On one computer, we installed PowerLogic SCADA 7.20 to poll the devices. On the other computer, we installed WireShark to monitor communication times.

Communication Settings

RS-485, 2 wire

19,200 Baud

Even Parity

Timeout: 3 seconds

Gateway Information

Model Number: EGX300SD

Firmware Version: 3.700

Hardware Version: A1

Manufacture Date

20 April, 2009

Communications Cable Information

Belden 8723, 2 pair, 22 AWG, Shielded Cable

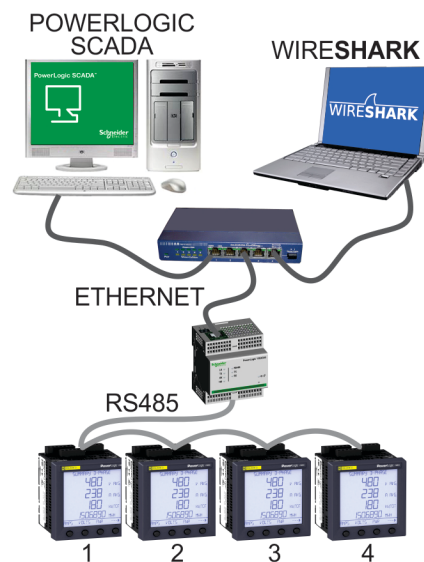
Communications Cable Length

Less than 10 feet total

Citect.ini Modification

[PLOGIC870]

CacheRefreshTime=0



Device Number	Device Model	Software
1	PM850	Version 10.600
2	PM850	Version 10.600
3	PM810	Version 10.410
4	PM810	Version 10.530

Test Procedure

We used the standard Power Meter 800 profile. Times represent one full read of system tags from the first variable tag request, to the device response, to the next request for the same tag. Response times were recorded first with Device 1 in the system, then with Device1 and Device 2, and so on. See the table below for results.

Test Results

	Response Time in Seconds			
	Device 1	Device 1+2	Device 1+2+3+4	Device 1+2+3+4
Average	0.713	1.382	2.118	2.933

Power Meter 1200 (CM4ECC/Modbus) Response Times

This section describes the response times for a Power Meter 1200 using a CM4000/ECC gateway and Modbus protocol.

System Configuration

The figure below illustrates the test configuration. We used an isolated network with an Ethernet hub. On one computer, we installed both PowerLogic SCADA 7.20 and WireShark, to monitor two devices.

Communication Settings

RS-485, 2 wire

19,200 Baud

Even Parity

Timeout: 3 seconds

Gateway Information

Model Number: CM4000 with ECC

Firmware Version: 6.00

Communications Cable Length

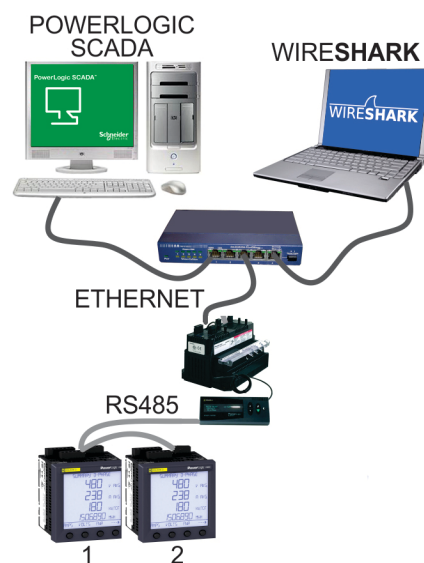
Less than 10 feet total

Device Information

Model: PM1200

Firmware: 3.05

Manufacturer Date: Unknown



Device Number	Device Model
1	PM1200
2	PM1200

Test Procedure

We used the standard Power Meter 1200 profile configured for “Little Endian” data communications. Times represent one full read of system tags from the first variable tag request, to the device response, to the next request for the same tag.

Test Results

	Response Time in Seconds
Min	0.735
Max	1.562
Average	0.964

Sepam S42 (EGX Modbus) Response Times

This section describes the response times for a Sepam S42 using an EGX300 gateway and Modbus protocol.

System Configuration

The figure below illustrates the test configuration. We used an isolated network with an Ethernet hub. On one computer, we installed PowerLogic SCADA 7.20 to poll the Sepam devices. On the other computer, we installed WireShark to monitor communication times.

Communication Settings+

RS-485, 2 wire

19,200 Baud

Even Parity

Timeout: 3 seconds

Gateway Information

Model Number: EGX300SD,

Firmware Version: 3.700,

Hardware Version: A1

Manufacture Date

4/20/2009

Communications Cable Information

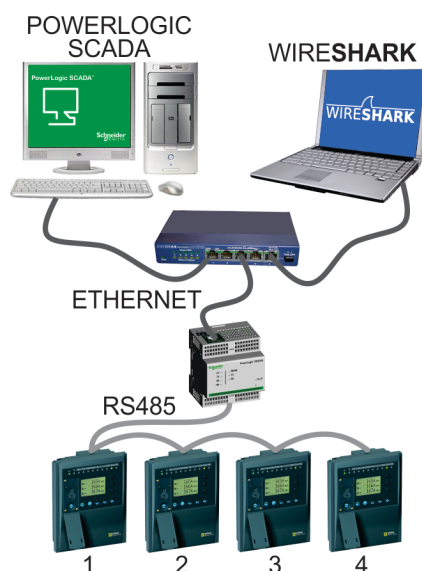
Belden 8723, 2 pair, 22 AWG, Shielded Cable

Communications Cable Length

500 feet from gateway to first device

200 feet from first device to second device

Less than 10 feet between remaining devices



Device Number	Device Model	Software	Communication
1	S42	Version 6.00	Version 0.1
2	S42	Version 5.03	Version 0.1
3	S42	Version 5.03	Version 0.1
4	S42	Version 5.00	Version 0.1

Test Procedure

We used the standard Sepam 40 profile. Times represent one full read of system tags from the first variable tag request, to the device response, to the next request for the same tag. Response times were recorded first with one Device 1 in the system, then with Device 1 and Device 2, and so on. See the table below for results.

Test Results

	Response Time in Seconds			
	Device 1	Device 1+2	Device 1+2+3+4	Device 1+2+3+4
Average	0.826	1.856	3.192	4.110

Sepam 80 (EGX Modbus) Response Times

This section describes the response times for a Sepam S80 using an EGX300 gateway and Modbus protocol. Modifications to default settings were made for performance enhancements.

System Configuration

The figure below illustrates the test configuration. We used an isolated network with an Ethernet hub. On one computer, we installed PowerLogic SCADA 7.20 to poll the Sepam devices. On the other computer, we installed WireShark to monitor communication times.

Communication Settings

RE-485, 2 wire

19,200 Baud

Even Parity

Timeout: 3 seconds

Gateway Information

Model Number: EBX300SD

Firmware Version: 3.700

Hardware Version: A1

Manufacture Date

4/20/2009

Communications Cable Information

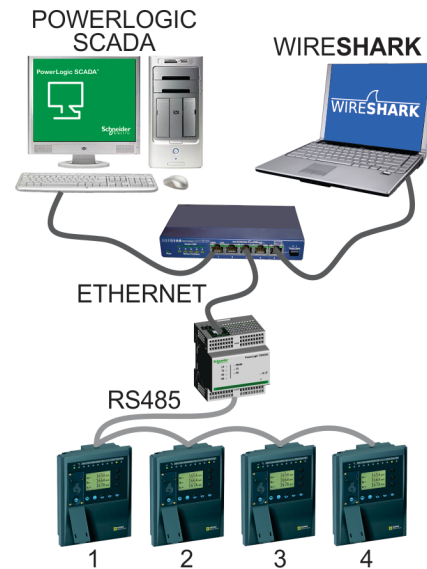
Belden 8723, 2 pair, 22 AWG, Shielded Cable

Communications Cable Length

500 feet from Gateway to first device

200 feet from first device to second device

Less than 10 feet between remaining devices



Device Number	Device Model	Software	Communication
1	G88	V5.00	V1.00
2	S84	V5.04	V1.00
3	G88	V5.00	V1.00
4	C86	V5.00	V1.00

Test Procedure

We used the standard Sepam 80 profile. The time represents one full read of system tags from the first variable tag request, to the device response, to the next request for the same tag. Response times were recorded with four Sepam devices in the system.

Three different settings were used to verify improvements in system performance:

1. The first test was run with no modifications to the Citect.ini file.
2. The second test was run with the following modifications to the Citect.ini file:

[Sepam80]

CacherefreshTime = 0

CommandsBandwidth = 1

EventsBandwidth = 18

Sepam 2000 S36 (EGX/Modbus) Response Times

This section describes the response times for a Sepam 2000 S36 using an EGX200 gateway and Modbus protocol.

System Configuration

The figure below illustrates the test configuration. We used a network with an Ethernet hub. On one computer, we installed PowerLogic SCADA 7.20 to poll the Sepam device. On the other computer, we installed WireShark to monitor communication times.

Communication Settings

RS-485, 4 wire
9600 Baud
Parity - None
Timeout: 3 seconds

Gateway Information

Model Number: EGX200
Firmware Version: 5.500
Hardware Version: A9
Manufacture Date
1 April, 2009

Communications Cable Information

Belden 8723, 2 pair, 22 AWG, Shielded Cable

Communication Cable Length

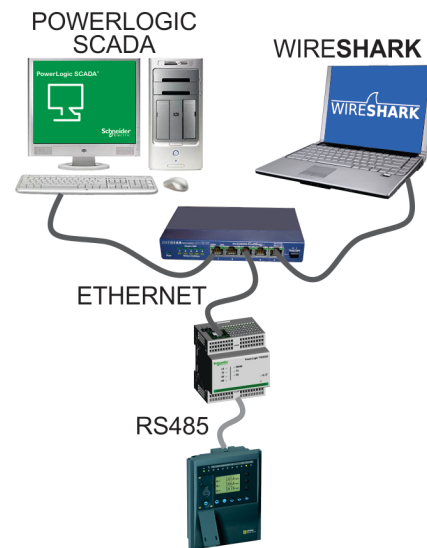
Less than 10 feet total

Device information

Sepam 2000
Model: S36 S02A
F/W: 6XR502AA
Communication JBUS V3.2

Citect.ini Modification:

[Sepam2000]
CacheRefreshTime=0



Test Procedure

We used the standard Sepam 2000 profile. Times represent one full read of system tags from the first variable tag request, to the device response, to the next request for the same tag. Response times were recorded with one Sepam 2000 in the system.

Test Results

Average Response Time: 3.077 seconds

TeSys T (EGX/Modbus) Response Times

This section describes the response times for a TeSys T motor controller using an EGX200 gateway and Modbus protocol.

System Configuration

The figure below illustrates the test configuration. We used a network with an Ethernet hub. On one computer, we installed PowerLogic SCADA 7.20 to poll the Tesys T. On the other computer, we installed Network Protocol Analyzer to monitor communication times.

Communication Settings

RS-485, 4 wire
9600 Baud
Parity - None
Timeout: 3 seconds

Gateway Information

Model Number: EGX200
Firmware Version: 5.500
Hardware Version: A9
Manufacture Date
unknown

Communications Cable Information

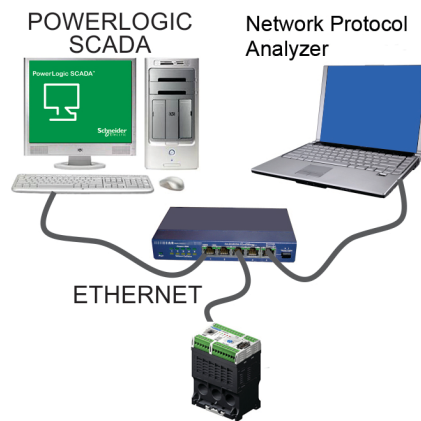
Belden 8723, 2 pair, 22 AWG, Shielded Cable

Communication Cable Length

Less than 10 feet total

Device information

Tesys T
Model (Modbus/TCP): LTMR08EBD
Model (extension module): LTMEV40FM
F/W (controller): V2.40
F/W (extension model): V1.4.0
Communication JBUS V3.2



Test Procedure

We used the standard Tesys T profile. Times represent one full read of system tags from the first variable tag request, to the device response, to the next request for the same tag. Response times were recorded with one Tesys T in the system.

Test Results

Average Response Time: 180.9 milliseconds

Multiple Protocols

Multiple Device/Multiple Protocol Response Times

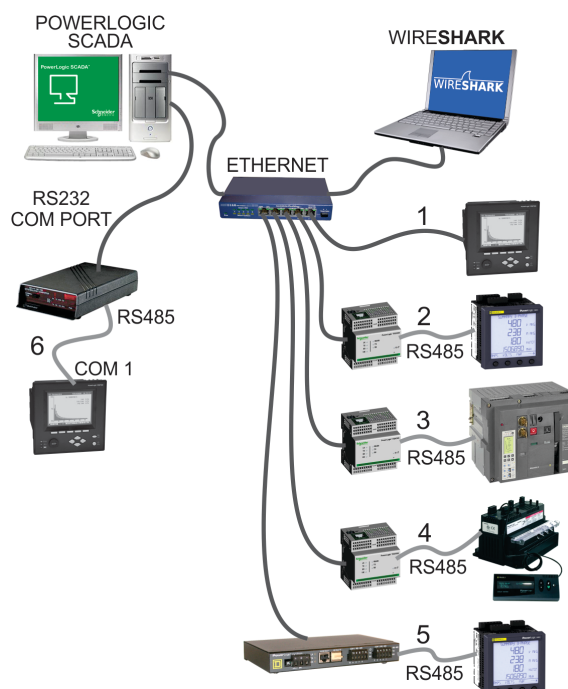
This section describes the response times of one system using several different types of communication protocols including Modbus, DNP3, and IEC 61850. Several different types of devices were used as well.

System Configuration

The figure below illustrates the test configuration. We used an isolated network with an Ethernet hub. On one computer, we installed PowerLogic SCADA 7.20 to poll the device. On the other computer, we installed WireShark to monitor communication times.

Test Procedure

We used the standard profile for each device type added to the system. Times represent one full read of system tags from the first variable tag request, to the device response, to the next request for the same tag.



Number	Device Type	Device Firmware Version	Gateway/ Converter	Gateway Info (19.2 K Baud)	Communication Protocol
Number	Device Type	Device Firmware Version	Gateway/ Converter	Gateway Info (19.2 K Baud)	Communication Protocol
1	ION7650	B0042 7650	None		DNP3
2	PM710	OS Version: 1.022	G3200PL	F/W Version 1.4	Modbus
3	MicroLogic Type A	Firmware Version 7.0	G3200PL	F/W Version 1.4	IEC61850
4	CM4000	OS Rev: 12:860	G3200PL	F/W Version 1.4	IEC61850
5	PM850	OSS Rev: 10.561	EGX400	F/W Version 5.500	Modbus
6	ION7650	B0042 7650	Black Box 109	Model No. IC109A-R2	DNP3

Test Results

Average Response for all tags in system: 2.605 seconds

Appendix B: Configuring PLSCADA Expert as an OPC-DA Server

Before you begin configuring OPC communications with PLSCADA Expert, refer to these help file locations:

- In the DriverReferenceHelp.chm help file (located in the PLSCADA Expert Bin folder), see the OPC Driver section.
- In the citectscada.chm help file (also in the Bin folder), see Using PLSCADA Expert > Using OPC Server DA2.0.

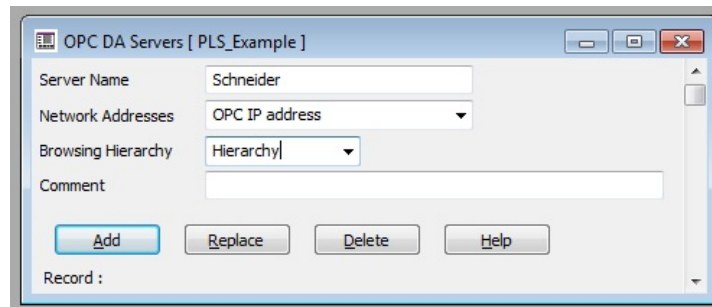
You can configure PLSCADA Expert to act as an OPC-DA server. In this mode, it will supply data to an OPC client, such as Matrikon OPC Explorer (a free download available at Matrikon.com). The following screen captures are from Matrikon Explorer, version 5.0.

NOTE: We used Matrikon in our tests and validation, but you may have one of the many other OPC products. The information in this document is specific to Matrikon products. Thus, the screens you see in your OPC client software may not be the same as the samples below.

Configuration Process

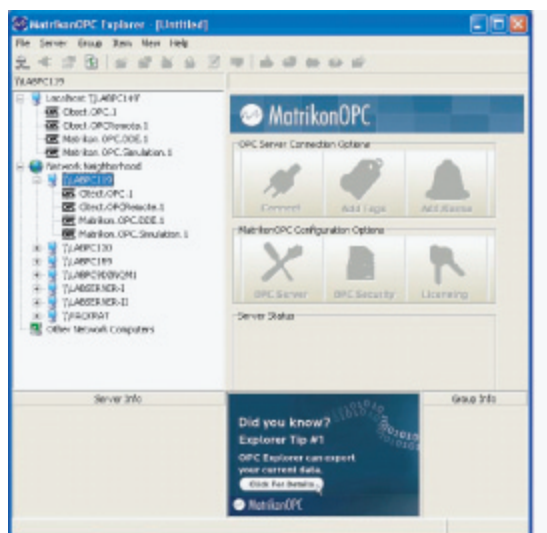
Follow these steps to select device profiles, create tags, and begin using the Matrikon tool:

1. From the Profile Editor, select the device profiles to be used for the project that will be used when PLSCADA Expert becomes an OPC-DA server.
2. From the Profile Wizard (Citect Project Editor > Tools > Profile Wizard), add the device. This will create the variable tags you need for the project.
3. To configure the OPC-DA server, complete the OPC DA Servers form in the Citect Project Editor > Servers.



4. Compile and run the project in PLSCADA Expert.
5. Launch the Matrikon OPC Explorer.

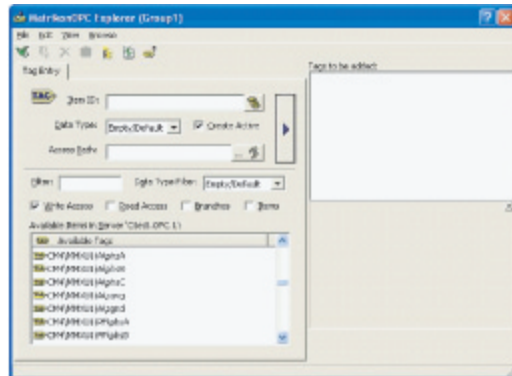
The Matrikon OPC Explorer screen displays. On the left side of the screen, a list of available OPC servers displays.



6. Highlight the server you want. The Connect button to the right of the list is enabled. Click Connect.

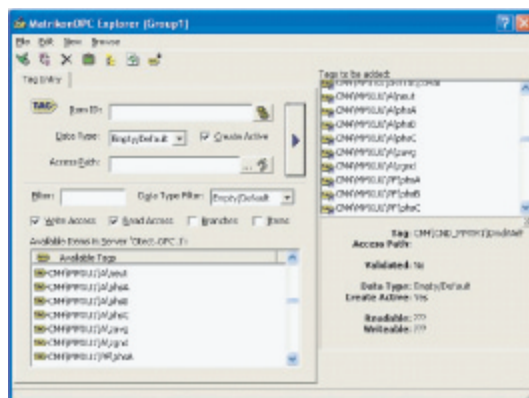
NOTE: If you are connecting to an OPC Server on a remote networked computer, and it does not display in the list, you must manually add the server. From the top toolbar, click Server > Add/Connect Server. This displays the form used to enter the host and server. Choose the server on that form and click OK to connect.

7. After you have connected to the server, click Add Tags to display a new pop-up box, which lists the available tags in the project that is running:

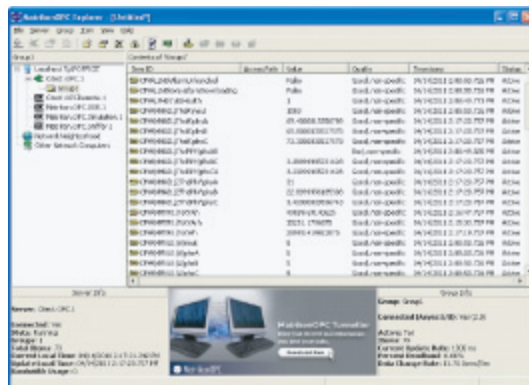


8. To add a single tag to the group, hover over the tag name and right click. Select Add to Tag List. To add all items to the tag list, right click and select Add All Items to Tag List.

Selected tags appear in the Tags to be added column on the right:



9. After you select all the tags you want, close the form: click File > Update and return. You return to the main setup page, where the tag values are displayed.



Appendix C: Configuring PLSCADA Expert as an OPC-DA Client

Before you begin configuring OPC communications with PLSCADA Expert, refer to the online help files in these locations:

- In the DriverReferenceHelp.chm help file (located in the PLSCADA Expert Bin folder), see the OPC Driver section.
- In the citectscada.chm help file (also in the Bin folder), see Using PLSCADA Expert > Using OPC Server DA2.0.

You can configure PLSCADA Expert to act as an OPC-DA client. In this mode, it will draw data from an OPC server, such as the one Matrikon OPC Explorer uses. The following screen captures are from Matrikon Explorer, version 5.0, which uses an OPC Simulation Server.

NOTE: We used Matrikon in our tests and validation, but you may have one of the many other OPC products. The information in this document is specific to Matrikon products. Thus, the screens you see in your OPC client software may not be the same as the samples below.

Configuration Process

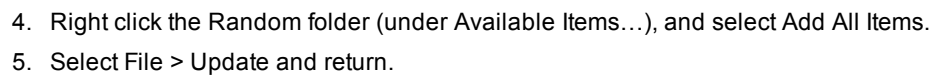
Follow these steps to create OPC tags in PLSCADA Expert:

1. Launch Matrikon Explorer to see tags that are available. Select the OPC Server to which you want to connect.

For this example, we are using Matrikon.OPC.Simulation.1

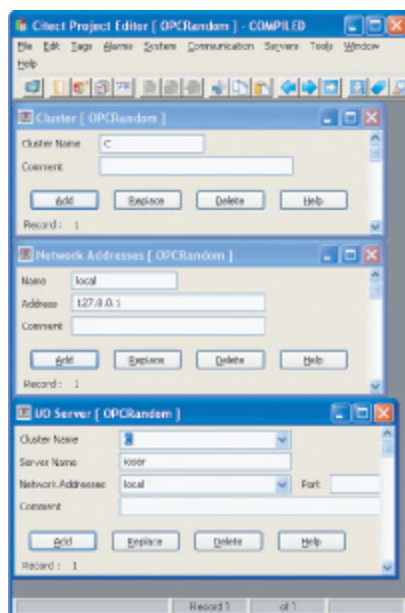


2. Connect to the Server Matrikon.OPC.Simulation.1 on the remote computer.
3. Click Add Tags to display the Tag Entry tab:

[illegible]

6. Create a project using Citect SCADA Explorer.
7. Using Citect Project Editor, add the following items:
 - cluster
 - network address
 - I/O Server

8. Add a board.



NOTE: Type the IP address of the remote OPC Server in the Special Opt field. The address field is used to specify the update interval in milliseconds. Type zero (0) here to use the default value.

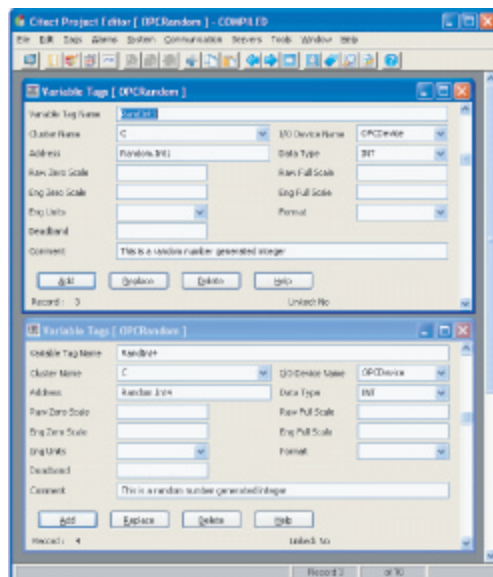
9. Create a port.

10. Create an I/O device which references the OPC Server name. Be sure to select OPC for the Protocol.

11. Create the variable tags:

- Add a tag name.
- Select the OPC I/O device you created earlier.
- The address is the tag name given by the OPC server.

One example in this case is Random.Int1, as shown in Matrikon Explorer display earlier.



12. Compile and run the project.

13. You can display the newly created PLSCADA Expert OPC tag values on a graphics page.

Performance Note: Using the setup described above with the default refresh rate (0), test results show that approximately 50,000 tags can be updated in less than one second. This was on a computer with an Intel Pentium dual-core processor running at 2.8 GHZ and 2 GB of RAM.

