

CyBro OPC

Data Access Server

User Manual

rev. 23

(applies to CyBro OPC v2.4.0 and later)

cybroTECH

© 2007 Cybrotech Ltd

Cybrotech Ltd
14 Brinell Way, Harfreys Industrial Estate
Great Yarmouth, Norfolk, Nr31 0LU - UK
tel: +44 1414 161884
www.cybrotech.co.uk
info@cybrotech.co.uk

Index

Index	1
General	2
Installation	2
Configuration	3
Activation code	3
Toolbar	4
Network monitor	5
PLC status	5
Tag status	5
Communication	5
Socket	6
Communication	7
Master/slave	7
Process priority	7
Port status	8
Port sharing	8
Tag quality	9
System tags	10
OPC system tags	10
CyBro system tags	11
Keyboard shortcuts	13
Technical Specifications	14

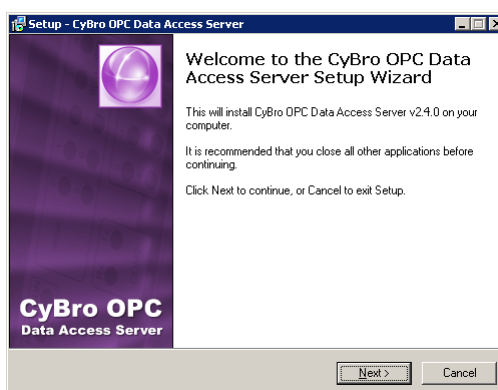
General

OPC stands for "OLE for Process Control". It is a specification standardized by OPC foundation (www.opcfoundation.org) which enables OPC client applications to access hardware specific data via OPC servers in a common, well defined way.

CyBro OPC Data Access Server enables OPC clients (SCADA/HMI or other) to connect to CyBro PLCs, using A-bus protocol. OPC client may read or write any location from the complete memory space.

Installation

To install CyBro OPC Server, start the installation archive and follow the instructions. Recommended install directory is C:\Program Files\CyBro OPC Server.



Installation does the following:

- unpacks CyBro OPC files into specified directory
- creates start menu group and icons
- registers OPC server

Note: Installation and setup should be performed with administrator privileges, otherwise OPC server will not function properly.

CyBro OPC Server may be installed on MS Windows 2000 or MS Windows XP. Recommended is MS Windows 2000 with the latest available service pack.

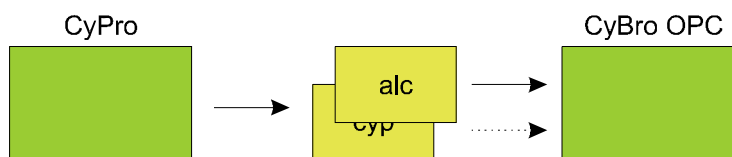
Installation occupies approximately 3Mb of disk space. RAM usage depends on number of connected PLCs, number of clients, and number of monitored variables, and in case of large network may be significant.

To upgrade CyBro OPC, install new release into the same directory, without uninstalling previous version. User settings will be preserved. Before upgrading, close related SCADA and shut down OPC server.

To uninstall CyBro OPC, start **Control Panel, Add/Remove Programs**, select CyBro OPC and press **Add/Remove** button.

Configuration

CyBro OPC is configured using the allocation file (.alc), generated by CyPro.



If project file (.cyp) with the same name exists in same directory, CyBro OPC will use it to read socket configuration and master/slave configuration. Project file is optional, CyBro OPC is functional without it, but in that case network monitor is not fully functional (no socket data is available), and master/slave configuration can not be automatically determined.

If allocation file is changed, CyBro OPC will automatically update configuration.

Configuration is saved automatically.

Activation code

To use CyBro OPC Server, proper activation code should be entered. Activation code may be obtained from your local distributor.

According to application size, activation code may be:

- small up to 100 tags
- medium up to 1000 tags
- large unlimited number of tags

Depending on distributor, only a "large" licence may be available.

Without the activation code, or when the number of used tags is greater than allowed number of tags, CyBro OPC server will work two hours, and then the warning message will pop up and server will stop. Period of two hours begins when the number of used tags becomes greater than allowed. Restarting the OPC server will provide another two hours. The number of cycles is not limited.

Using the OPC server with 10 or less tags is free, no activation code is needed.

Without the activation code, CyBro OPC server may be used for development/testing purpose only.

Toolbar



Connect

Open communication port and connect to A-bus network.



Add project

Add CyPro project (cyp) to project tree. Although a single project file may enclose all PLC's in network, it is possible to open more than one project. If project contains PLC's that are not currently available, network update time will be considerably increased.



New monitor

Create and open a new monitor list. List may contain any number of PLC or system tags. Monitor is an OPC client. When selected, read request is sent to OPC server periodically every 300ms. When monitor is not selected, no requests are sent, so network performances are not affected.



Add to monitor

Add selected (one or more) tags to a current monitor. Current monitor is the last opened one. Current monitor is indicated by a star symbol (i.e. Monitor3*).



Refresh tag

Refresh (one or more) selected tags. Tags will be refreshed in first available communication cycle. Refresh is performed by setting a read request (■) to selected tag(s), so the actual response time depends on a number of connected clients and a number of pending requests.



Refresh PLC

Refresh all tags of selected PLC.



Set value

Set value of selected (one or more) tags. Value will be written in next available communication cycle. Refresh is performed by setting a write request (■) to selected tag(s), so the actual response time depends on a number of connected clients and a number of pending requests.



Tag color
and monitor view.

Change a color of selected (one or more) tags. Tag has the same color in PLC



Move tag up

Move tag up.



Move tag down

Move tag down.

Network monitor

Project tree on the leftmost pane may be used as a very effective monitor of network traffic. Network activity is showed in real time, using animated icons, so the network state is visible at a glance.

PLC status



Active

CyBro is active and there are no communication errors.



Inactive
tag Sys.ComStatus.

CyBro is inactive, there is no communication. For more details, check PLC system



SCADA request

Indicates that OPC client (SCADA) put a request to read/write one or more PLC tags.

Tag status



Reading
cycle.

Read request is pending. Tag will be updated in next available communication



Writing
cycle. Tag quality becomes

LOCAL_OVERRIDE until next read cycle is performed.



Idle

No requests are pending, tag will be refreshed in next background refresh cycle. If communication is busy because of SCADA requests, background refresh is not available, and idle tags are not updated (tag quality drops to LAST_KNOWN).

Communication



Reading

Indicates a data read cycle. Two devices are always activated at (almost) the same time, first one sending the read request, and second one responding with requested data. If only one indicator appears, no response is detected.



Writing

Indicates a data write cycle. Two devices are always activated at (almost) the same time, first one sending write request and data, and second one sending the acknowledge. If only one indicator appears, no response is detected.



Master

Indicates network master activity. Indicator appears when master actually send permission-to-send command (SEND_PERM). No indicator shown means no master is active, network is dead. Two or more indicators means more than one master is trying to communicate, so the network is not properly configured.



Token

Indicates PLC which received permission-to-send command. It is a pair with Master, and appears at the same time.

Socket



Transmit/receive Indicate socket activity. At the moment when socket is sent, both output and input sockets flicker at the same time.



Inactive Indicates inactive socket. In case of output socket, CyBro does not have permission (Token) to send the socket. In case of input socket error is somewhere else: no matching output socket or the socket is not sending.



Error

Indicates that socket sent on the network does not match. Testing is performed only for size; data type and order are not checked.

Communication

Master/slave

CyBro OPC server may be either network slave or network master.

By default, master/slave configuration is determined automatically - OPC is master when all CyBro's are slaves. Auto mode works only if project (cyp) file is available. Otherwise auto mode should be turned off, and configuration should be selected manually.

Note: CyBro OPC master is usable only when data is exchanged exclusively between OPC and CyBro, and there is no communication between CyBro's (there are no declared sockets). If project contains at least one socket, network master should be CyBro.

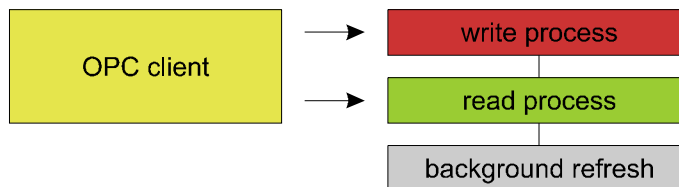


Process priority

Whether OPC is master or slave, it performs three different communication processes:

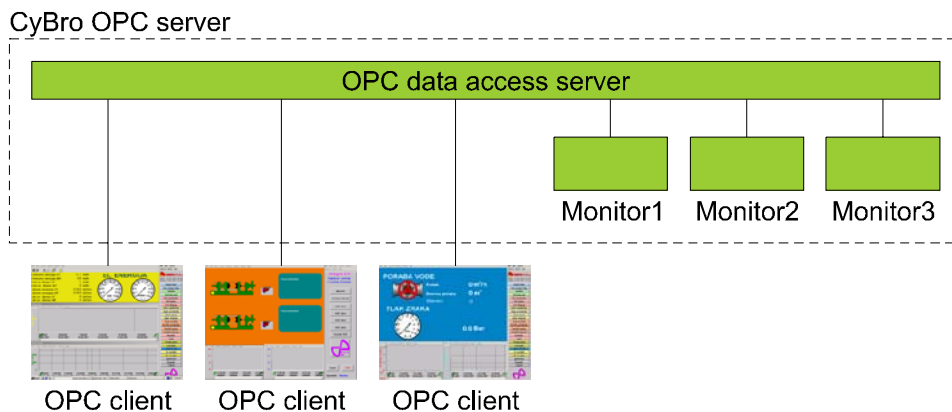
- write process (request to write data to PLC)
- read process (request to read data from PLC)
- background refresh (when no request is pending, refresh all tags)

Write process has top priority, read process is below, and background refresh has lowest priority.



Read and write processes are initiated by OPC client requests (SCADA). If there are no read/write requests, only a background refresh is active. If read/write requests from SCADA are very frequent, background refresh will be stopped.

CyBro OPC data monitor is in essentially an OPC client.



Port status

Cable indicator icon on the status bar displays the status of the communication port:



Can't open communication port. The reason may be a wrong port number, or the port may already be used by another application.



Communication port is open, but the cable is not detected. The reason may be a wrong port number, or bad communication cable. Connect cable to another DB-9 connector, or use another cable.

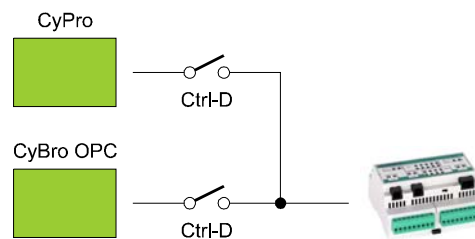


Communication port open successfully and CyBro cable is connected to the port. If communication is not working, check network master settings.

Port sharing

CyBro network is connected to PC using serial port. Unlike USB or Ethernet, serial port can not be shared among applications, only one application at the time can open the port.

To overcoming the problem, CyBro OPC and CyPro employ a special protocol to take over the communication port. When performs any communication command in CyPro, it will take over the port. Port will not be returned automatically - user needs to return back to CyBro OPC and manually connect to network.



Both applications have the same keyboard shortcut (Ctrl-D) to connect/disconnect the network.

Tag quality

According to OPC specification, each tag has **value** and **quality**.

Value is an actual tag value, quantity.

Range of possible values depends on tag type (bit, integer, long, real).

Quality is a property independent of value, and specifies how reliable the actual value is. Quality depends on how and when the value is obtained. When communication is uninterrupted, quality is always "Good". If communication channel is broken, quality first become "Uncertain", then "Bad".

A range of possible qualities is defined by OPC specification. There are three main categories, "Good", "Uncertain" and "Bad. Each category can contain additional info field, which may give a more detailed explanation about the problem cause.

CyBro OPC server implements the following qualities:

Good Value is updated regularly.

Good, local override Value is locally overridden, using function "Set Value". Tag status is red (pending to write). After the write command is completed, status becomes green (reading) or gray (idle). When tag value is updated next time, quality changes back to "Good".

Uncertain, last usable First timeout, last updated value is 5 to 10 seconds old. Possible reason may be a slower network update time, or some communication errors. This is just a warning, not an error.

Bad Value is unusable, no successful read was performed since OPC server started.

Bad, last known Second timeout, last updated value is more that 10 seconds old. Communication is broken, so the value is not reliable any more.

Bad, out of service Activation code expired. More tags than allowed by license is used, development timeout (2 hours) expired.

To adjust "Uncertain" and "Bad" timeouts, open **Settings**, tab **Advanced**.

System tags

System tags are virtual tags created and handled by OPC server. Their purpose is to show information about OPC server/client and the network.

Two kinds of system tags are available - those belonging to OPC itself, and those belonging to each PLC.

Value of a system tag can not be changed by user, but there is a command, "Reset sys counters", which resets values of all system tags. It is not possible to reset a single tag.

OPC system tags

ServerStartedAt.....	Time and date when OPC server is started.
ServerUptime	Time elapsed since OPC server is started.
LicenseType	License type: "not licensed" (<10 tags), "small" (<100 tags), "medium" (<1000 tags) and "large" (unlimited number of tags).
LicenseStatus.....	License status: "ok", "30 min left", "expired".
TagLimit	Total number of monitored tags allowed by given license.
ClientsConnected	Number of connected OPC clients.
MonitoredTags	Total number of tags monitored by all connected OPC clients.
ProjectCount.....	Number of loaded CyPro projects.
PlcCount.....	Total number of PLC's in all loaded projects.
ComStatus	Communication status: "disconnected", "ok", "idle (no tags)", "error: can't open COM port", "error: no communication", "error: master running, but no requests to OPC", "error: some PLCs are ok, some are not responding", "error: no PLC is responding".
ComReceiveCount	Total number of good communication messages received.
ComTransmitCount	Total number of communication messages sent.
ComErrorCount	Total number of communication errors (including timeouts, bad messages and negative acknowledges).
LastReceivedAt	Timestamp of last good communication message received.
LastReceivedNad	Device from which last good communication message is received.
LastResponseTime	Response time of last good communication cycle.
MaximumResponseTime.....	Maximum response time encountered.
MaximumResponseTimeAt	Timestamp when maximum response time is encountered.
MaximumResponseTimeNad	Device for which maximum response time is encountered.
ComTimeoutCount	Total number of communication timeouts.
LastTimeoutAt	Timestamp of last detected OPC communication timeout.
LastTimeoutNad	Last device that caused communication timeout.
BadMessageCount.....	Total number of bad communication messages received.
LastBadMessageAt	Timestamp of last detected bad communication message.
LastBadMessageNad	Last device from which bad communication message is received.
NegativeAckCount.....	Total number of negative acknowledges received.
LastNegativeAckAt	Timestamp of last detected negative acknowledge.
LastNegativeAckNad	Last device from which negative acknowledge is received.
TagsToRead	Total number of tags waiting to be read (excluding background refresh).
TagsToWrite.....	Total number of tags waiting to be written.

ReadCount	Number of OPC read cycles (excluding background refresh).
LastReadAt.....	Timestamp of last succesfull OPC read (excluding background refresh).
LastReadNad	PLC accessed in last succesfull OPC read (excluding background refresh).
ReadErrorCount	Number of OPC read errors (excluding background refresh).
WriteCount	Number of OPC write cycles.
LastWriteAt.....	Timestamp of last succesfull OPC write.
LastWriteNad.....	PLC accessed in last succesfull OPC write.
WriteErrorCount	Number of OPC write errors.
BackgroundRefreshCount	Number of OPC background refresh cycles.
LastBackgroundRefreshAt.....	Timestamp of last succesfull OPC background refresh.
LastBackgroundRefreshNad	PLC accessed in last succesfull OPC background refresh.
BackgroundRefreshErrorCount	Number of OPC background refresh errors.
LastBackgroundRefreshCycle	Cycle time to background refresh all tags of all PLC's.
MasterNad.....	Master NAD.
MasterRequestCount	Number of requests sent from master to any device in the network.
MasterToOpcRequestCount.....	Number of requests sent from network master to OPC.
LastMasterCycle.....	Time between last two requests sent from network master to any PLC.
LastMasterToOpcCycle.....	Time between last two requests sent from network master to OPC server.
LastNetworkSlaveNad	Last device accessed by network master.
NetworkResponseCount.....	Total number of responses sent to network master.
LastNetworkResponseAt	Timestamp of last response to network master.
LastNetworkResponseNad.....	Last device responded to network master.
NetworkSocketCount.....	Total number of sockets sent in the network.
LastNetworkSocketAt	Timestamp of last socket sent.
LastNetworkSocketNad	Last device which sent a socket.
TimeSinceLastMasterRequest	Time since last request from master received.
TimeSinceLastMasterActivity	Time since last communication activity detected.

CyBro system tags

NAD.....	Actual CyBro network address (may be NAD or NAD alias).
TimedOut	True if CyBro is not responding to communication requests.
Master	True if CyBro is network master, false otherwise.
ComStatus	Communication status: "ok", "disconected", "idle (no tags)", "error: can't open COM port", "error: no communication", "error: master running, but no requests to OPC", "error: PLC is not responding".
IPAddress.....	Current IP address of connected CyBro (Ethernet only).
ComReceiveCount	Total number of good communication messages received from this PLC.
ComTransmitCount	Total number of communication messages sent to this PLC.
ComErrorCount	Total number of communication errors from this PLC (including timeouts, bad messages and negative acknowledges).
LastReceivedAt	Timestamp of last good communication message received.
LastResponseTime	Response time of last good communication cycle.

MaximumResponseTime.....	Maximum response time encountered.
MaximumResponseTimeAt	Timestamp when maximum response time is encountered.
ComTimeoutCount	Total number of communication timeouts.
LastTimeoutAt	Timestamp of last detected OPC communication timeout.
BadMessageCount.....	Total number of invalid communication messages received.
LastBadMessageAt	Timestamp of last detected invalid communication message.
NegativeAckCount.....	Total number of negative acknowledges received.
LastNegativeAckAt	Timestamp of last detected negative acknowledge.
TagsToRead	Number of tags waiting to be read (excluding background refresh).
TagsToWrite.....	Number of tags waiting to be written.
ReadCount	Number of read cycles (excluding background refresh).
ReadErrorCount	Number of read errors (excluding background refresh).
WriteCount	Number of write cycles.
WriteErrorCount	Number of write errors.
BackgroundRefreshCount	Number of background refresh cycles.
BackgroundRefreshErrorCount	Number of background refresh errors.
MonitoredTags	Total number of tags monitored by all connected OPC clients.
SystemStatus	System status: "loader", "kernel".
PLCStatus	PLC status: "stop", "pause", "run", "error".

Keyboard shortcuts

Ctrl-O.....	Add project
Ctrl-N.....	New monitor
Ctrl-F4	Close monitor
Ctrl-A	Select all tags
Ctrl-M	Add selected tag(s) to monitor
Delete.....	Remove selected tag(s) from monitor
Ctrl-D.....	Connect/disconnect
Tab.....	Select active window (project tree or tag list)
Up/Down.....	Move selection up/down
Enter.....	Set tag(s) value
Space	Toggle tag(s) value (bit only)
Ctrl-Up/Down.....	Move selected tag(s) up/down
F9.....	Refresh selected tag(s)
F10	Refresh all PLC tags
F11	Refresh all currently visible tags
F5.....	Settings
Alt-F4.....	Close OPC window
Ctrl-Alt-F4.....	Shutdown OPC server

Technical Specifications

OPC Server ID: CyProOPC.DA2

Supported OPC versions:..... 1.0, 1.0a and 2.0

Supported PLCs: CyBro-1: CyPro v1.9.0
CyBro-2: CyPro v2.4.0 and above

Communication: Serial (RS232/485, 19200, N, 8, 1), Ethernet/LAN, Ethernet/WAN

Communication protocol:..... A-bus

Used A-bus commands: RD_DATA, WR_DATA, RD_RANDOM, WR_RANDOM, RD_STATUS