Created by Werner Krings, Tani GmbH
Date: 2019.7.1

Technical Data Opc Server

General

The Tani OPC Server is a multi protocol and multi OPC server. It allows access to coltrollers and devives from various manufactuters. It is easily to configure. It offers a lot of diagnostics functions.

Interfejsy OPC

- **OPC Pipe** Otwarty interfejs
- **OPC UA** (Unified Architecture)
- **OPC DA** (Classic OPC z technologią DCOM, tylko dla systemu Windows)

Maksymalna liczba klientów OPC jest zależna tylko od zasobów komputera. Komputer PC z 2014 roku może obsługiwać kilkaset połączeń.
Wszystkie interfejsy OPC działają lokalnie w jednym komputerze lub przez sieć
W przypadku OPC Classic, nie jest zalecane korzystanie z technologii DCOM w sieci, ale takie połączenie jest obsługiwane.
OPC UA wspiera szybki protokół binarny. Bezpieczeństwo nie jest zapewniane. Wykorzystywane są DataAccess / DataItems do 200K długości.

OPC UA functionality and limitations

The OPC UA implementation conforms to the specification 1.04.
An internal discovery server is active on standard, it supports multicast discovery also. It can be used as a global discovery server. Alternatively an external discovery server can be configured.

The session timeout will be limited to one hour.
AddNodes is supported with the following restrictions:

- AddNodes is possible only in Objects.Topics.Memory tree
- Reference type must be OpcUaId_Organizes
- NodeId can't be specified
- BrowseName can't contain a dot
- NodeClass must be Variable or Object
- NodeAttributes for Variable:
  - DisplayName: unspecified or equal to BrowseName
  - Description: unspecified or any text
  - Value: is ignored; new variables will always be initialized to 0 (if numeric) or "" (if string type)
  - DataType:
    - OpcUaType_Boolean
    - OpcUaType_SByte, OpcUaType_Byte
    - OpcUaType_Int16/32/64, OpcUaType_UInt16/32/64
    - OpcUaType_Float, OpcUaType_Double
    - OpcUaType_String
    - OpcUaType_DateTime
    - One of the structure types under Types -> DataTypes -> BaseDataType -> Structure -> UserStructures; these are the structures known to the PLC Engine core.
      - if the structure is given both here and via TypeDefinition, both settings must match
    - if unspecified, OpcUaType_Byte or the structure type of the TypeDefinition is used
  - ValueRank, ArrayDimensions: unspecified (= scalar), scalar or a one-dimensional array of any size
  - AccessLevel, UserAccessLevel: unspecified or (OpcUa_AccessLevels_CurrentRead | OpcUa_AccessLevels_CurrentWrite)
  - MinimumSamplingInterval: unspecified or 0
  - Historizing: unspecified or 0
  - WriteMask, UserWriteMask: unspecified or OpcUa_NodeAttributesMask_Value
- NodeAttributes for Object:
  - DisplayName: unspecified or equal to BrowseName
  - Description: unspecified or any text
  - EventNotifier, WriteMask, UserWriteMask: unspecified or 0
- TypeDefinition for Variable:
  - OpcUaId_BaseDataVariableType
  - one of the structure types under Types -> VariableTypes -> BaseVariableType -> BaseDataVariableType -> UserStructures; these are the structures known to the PLC Engine core.

- TypeDefinition for Object:
  - OpcUaId_FolderType

Możliwa jest wymiana danych pomiędzy różnymi interfejsami OPC (tunelowanie).

Controller Interfaces

All controllers will be connected over network. Often this is Ethernet, WLAN or other networks. All serial Ethernet and MPI Ethernet gateways for industrial controllers usage are supported.

Configuration Interfaces

The configuration can be done with the shipped configuration software or over OPC with the System topic.
The connection for the configuration is encrypted with TLS 1.2. The encryption can be switched off for usage in countries where encryption is forbidden.

Network Redundancy for connections to controllers and devices

Connections to devices and controllers are supporting network redundancy.
Double and triple redundancy can be selected.
Two redundancy operation modi are possible.
In **dynamic redundancy** any of the connections is working as master. If it breaks another connection becomes the master connection.
In **static redundancy** the first connection is the master. If it breaks another connection becomes the master. If the first connection works again it will become the master connection again.

The connections of the redundancy should work on different netaork adapters.The adapters need different IP subnets for proberly work..

Obsługiwane sterowniki PLC i urządzenia

- Siemens Simatic S7-200, S7-300, S7-400, S7-1200 oraz S7-1500, VIPA Speed7, IBH SoftS7 i inne. Sterowniki, które nie są wyposażone w interfejs Ethernet mogą być podłączone przez adapter MPI/Profibus - Ethernet np. S7-LAN. S7 1500 + 1200 z przeglądania stron w Internecie i korzystania z "Optimized bloków danych".
- Siemens Simatic S5 przez TCP/IP lub OSI/H1. Sterowniki, które nie są wyposażone w interfejs Ethernet mogą być podłączone przez adapter AS 511 - Ethernet np. S5-LAN.
- Allen-Bradley CompactLogix i ControlLogix (wszystkie wersje)
- urządzenia z interfejsem Modbus/TCP:
  - Modicon
  - Schneider
  - Wago
  - Beckhoff
  - Phoenix Contact
- Raw Data (surowe dane)

Komunikacja w każdym przypadku realizowana jest w sieci Ethernet

BACnet

BACnet will be used over IP / UDP.
Maximum length of strings: 256 Byte
Status text elements are supported (state_text)
Supported charsets: UTF-8, UTF-16, Latin-1
Unions ("Choice") and structures ("Sequence") are existing for important values. All unimplemented instances will not be shown.
Enum values are represented as UINT32. Some special enum are handled as bool.
Values in "Octet-String" and "Bit-String" can be written in whole only.

BBMD (BACnet Broadcast Management Device) details

BBMD will be used during the connection establishing and the device search if the devices do not be all in the same collision domain. BACnet uses broadcast during ist connection establishing.
There are several procedures in BBMD:

- Search ussing broadcast.
- Search using the IP device address, receive the BACnet ID.
- Search using the BACnet id, receice the IP address.

Additionally BBMD can be used connecting older serial only installations to the IP network.

COV (Change Of Values) details

COV represents the event subsystem of BACnet. Events will be offered in browsing the variables, they will be subscribed. If the device will send the data the event will be generated.
Because BACnet is working with UDP the COV receive can not be guaranteed. Tani is offering an option: If no event will be received during the reconnection time from the configured connection it will be polled. If the value did not change no event is send for this polling.

BACnet - Writing values with priority-array

These object types have a priority-array in addition to their present-value property:

- analog-output
- analog-value
- binary-output
- binary-value
- multi-state-output
- multi-state-value
- access-door

The BACnet spec says:

- priority-array is read-only and contains 16 entries (that can be a valid value or NULL).
- present-value is read-write and contains 1 value (the non-NULL value with the lowest priority from priority-array, or the value from relinquish-default if no non-NULL value in priority-array exists).
- Writing to present-value uses an optional priority parameter to write to the correct entry in priority-array.

The Tani implementation works as follows:

- priority-array is read-write and contains 16 structure entries with 2 fields:
  * Value: the data value in this entry (or 0 if no valid value is present)
  * ValueValid: a boolean value; 1 if Value is valid, 0 if not (NULL value).
- Writing to an element of priority-array implicitly uses a "write present-value with priority" operation to change the desired value.
- Writing to priority-array[i].Value always creates a non-NULL entry.
- Writing 0 to priority-array[i].ValueValid creates a NULL entry.
- Writing 1 to priority-array[i].ValueValid creates a non-NULL entry with value 0 (this is usually not very useful).
- Writing to priority-array[i] (as a structured data type) creates a NULL entry when ValueValid is 0. Else a non-NULL entry with the specified Value is created.
- present-value is read-write and contains the value obtained by BACnet protocol.
- Writing to present-value doesn't transfer the priority parameter. The BACnet device will implicitly write to priority entry 16 in this case.

This mechanism was chosen to allow choosing the write priority via OPC without changing the read syntax for present-value property. This also allows writing NULL values via OPC.

Logger for diagnostics

The OPC Server contains a logger for diagnostics purposes during plant startup. The logger can be configured. The system load can be big if all controller data in big plants are logged.

Limits

Maximum number of configurable client connections: 4000.
Maximum length of a single item: 4GB.
Maximum number of elements each connection: 1 million.
Maximum number of elements (Items): 16 million.
Maximum OPC groups each connection: 100.
Maximum number of passive connection for each port is 999.
The OPC synchronous functions returning a bad quality immediately if the PLC connection is not established.
Changes in controller configuration will be checked all 10 seconds if the PLC does not offer a mechanism for this check during write.
Fields can be up to 64K in length each.
Multi dimensional arrays can have up to six dimensions.

Simulation of plc connections

Connections and their data can be simulated.
Read data creates the elements with value zero. Write will change the content. The changed values will be returned as new data.
Limits:

- Strings can be written up to 16 bytes
- Suffixes will not be handled
- Date and time are supported limited

Depending on the license the limits can be less.

Speed

The throughput will be mainly limited by the controller speed or the reaction time of OPC applications.
Read requests to the controller will be optimized as much the controller is supporting this. For that elements will be collected to blocks reading more than requested, but not for inputs and outputs. These optimizing can be affected by configuration separately for each connection. Optimizing can be switched off, too.
Write requests to the controller are collected or handled in that order the application did called the system.
On OPC all optimizing the individual OPC uses is supported.
The normal time in cyclic controller requests is 50ms.It can be faster if the controller polling interval is set to zero.
Only data are sent to OPC which did change in the controller between two read requests.

Field and text optimizings

The from version 1.8 existing field optimizings will prevent reading the long fields too often, the index is requested on standard only.This optimizing bases of the fact that the index does net changed too frequently.

Usage of memory

- Program code: A minimum of 6MB is used. The exactly memory usage is depending of the internal behavior of the operating systems. So dynamic libraries are loaded once for all running instances using them. Example: If the standard library is not loaded already it will use additional 4MB of memory.
- User data: The minimum data usage is 2MB internally. Additional the controller data are held in memory for comparing new data. Each item uses the length of data and additional 64 bytes. Each configured connection occupies 4KB.

Usage of computation time

The consumed computation time is depending on the load with communication. Most the time it will be waited for controller data or OPC application reaction.
All software is working with events. This maximizes the throughput and minimizes the usage of computation time.
Multiple CPU are supported. Up to ten CPU will be used, the main work will be handled by three CPU.

Installation

The installation does depending on the product install multiple parts separately. On uninstall not all products are deleted automatically. But all installed products can be deleted over the menu or the software part in the system control manager.
The user settings will be preserved and not deleted during uninstall.

Obsługiwane systemy operacyjne

- Windows XP, Vista, 7, 8, 8.1, 10. 32/64 Bit
- Windows Server 2008, 2012, 2016 and 2018
- Linux na komputerach Raspberry oraz Odroid
- Linux na komputerach Wiesemann & Theis pure.box 3
- Linux Debian, Ubuntu, Suse, Redhat i inne dystrybucje
- Linux 64 Bit jako kontener Docker or Kubernetes

- OPC DA wymaga systemu Microsoft Windows. Wszystkie obsługiwane systemy operacyjne Windows dla Intela i języki użytkownika będą obsługiwane. Najnowszy dodatek Service Pack musi być zainstalowany.
- Pod Windowsem serwer OPC pracuje jako usługa, Linux uruchamia go jako demon.
- Wersja Raspberry obsługuje wszystkie dystrybucje Linuksa oferowane na tej platformie.
- Wszystko inne będzie pracować na wielu systemach operacyjnych, głównie opartych na Linuksie.
- Pod Linuksem OPC Server potrzebuje systemu zgodnego z POSIX. Biblioteka standardowa potrzebuje V2.2 jako minimum. Oprogramowanie konfiguracyjne bazuje na KDE 4 i potrzebuje kdelibs. Proszę używać aktualnych dystrybucji takich jak Debian, Ubuntu, Suse, RedHat i podobne.
- Przeestowany jest: Windows Intel 32 i 64 bit, Linux Intel 32 i 64 bit, Linux MIPS CPU, Linux ARM 32 i 64 bit
- Uruchamianie w maszynach wirtualnych jest obsługiwane. Kontenery są także obsługiwane.

- Windows7 potrzebuje Service Pack 1 do obsługi sterownika H1.
- Wszystkie konfiguracje są kompatybilne ze wszystkimi serwerami OPC, a także powyższymi systemami operacyjnymi.