OPC Unified Architectrue (OPC UA)

Laura Mons

DFKI

09.July.2020

Content

1. Introduction
2. Open62541
3. Demo

Content

1. Introduction
2. Open62541
3. Demo

What is OPC

- OPC - Open Platform Communications

What is OPC

- OPC - Open Platform Communications
- Tries to provide a universal approach for communication via bus systems or protocols

What is OPC

- OPC - Open Platform Communications
- Tries to provide a universal approach for communication via bus systems or protocols
- OPC Compliance Tests available to ensure compatibility

What is OPC

- OPC - Open Platform Communications
- Tries to provide a universal approach for communication via bus systems or protocols
- OPC Compliance Tests available to ensure compatibility
- Constructed to communicate between sensors or other IoT devices in heterogeneous networks

What is OPC

- OPC - Open Platform Communications
- Tries to provide a universal approach for communication via bus systems or protocols
- OPC Compliance Tests available to ensure compatibility
- Constructed to communicate between sensors or other IoT devices in heterogeneous networks
- OPC standards include:
  1. Real Time Data
  2. Data archiving
  3. Alarm messages
  4. Control (command transmission)

What is OPC UA

- OPC UA - Open Platform Communications Unified Architecture

What is OPC UA

- OPC UA - Open Platform Communications Unified Architecture
- Current specification of OPC

What is OPC UA

- OPC UA - Open Platform Communications Unified Architecture
- Current specification of OPC
- Enables machine readable description of machine data

What is OPC UA

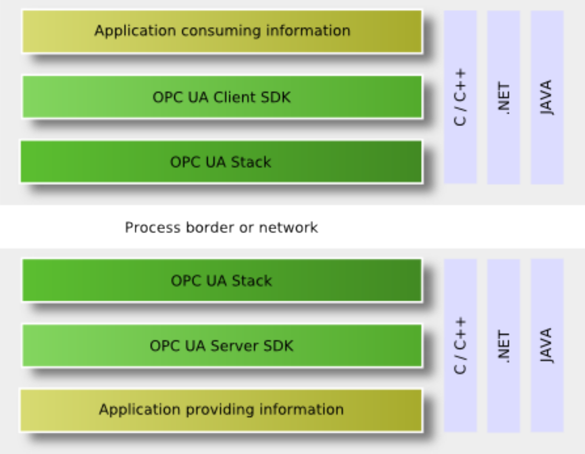
- OPC UA - Open Platform Communications Unified Architecture
- Current specification of OPC
- Enables machine readable description of machine data
- Not tied to COM/DCOM anymore

  ⇒ Eliminate drawbacks such as binding to Windows, security issues, or no configurable timeouts

What is OPC UA

- OPC UA - Open Platform Communications Unified Architecture
- Current specification of OPC
- Enables machine readable description of machine data
- Not tied to COM/DCOM anymore
  - ⇒ Eliminate drawbacks such as binding to Windows, security issues, or no configurable timeouts
- Own communication stack for OPC UA includes
  1. Portable ANSI-C implementation as well as C++ and Java
  2. Scalability from embedded control software to operational or management information systems
  3. Compileable for both multithreaded and single-threaded / single-task operation, which is important for porting to embedded devices.
  4. Security implementation based on the latest standards
  5. Configurability of timeouts for each service call
  6. Chunking large data packets

OPC - UA - Communication Stack


Application consuming information
OPC UA Client SDK
OPC UA Stack
C / C++
.NET
JAVA
Process border or network
OPC UA Stack
OPC UA Server SDK
Application providing information
C / C++
.NET
JAVA

https://documentation.unified-automation.com/uasdkhp/1.0.0/html/softwarelayers.png

Choosing an Open Source Library

- Some open source implementations available: https://github.com/open62541/open62541/wiki/List-of-Open-Source-OPC-UA-Implementations
- Limited availability of open source C++ libraries with client framework

Choosing an Open Source Library

- Some open source implementations available: https://github.com/open62541/open62541/wiki/List-of-Open-Source-OPC-UA-Implementations
- Limited availability of open source C++ libraries with client framework
- Identified three possible candidates:
    - FreeOPCUa with LGOL License

Choosing an Open Source Library

- Some open source implementations available: https://github.com/open62541/open62541/wiki/List-of-Open-Source-OPC-UA-Implementations
- Limited availability of open source C++ libraries with client framework
- Identified three possible candidates:
    - FreeOPCUa with LGOL License → Build failing at time of evaluation and no releases, sparse documentation, no historical data access

Choosing an Open Source Library

- Some open source implementations available: https://github.com/open62541/open62541/wiki/List-of-Open-Source-OPC-UA-Implementations
- Limited availability of open source C++ libraries with client framework
- Identified three possible candidates:
    - FreeOPCUa with LGOL License → Build failing at time of evaluation and no releases, sparse documentation, no historical data access
    - ASNeG with Apache License

Choosing an Open Source Library

- Some open source implementations available: https://github.com/open62541/open62541/wiki/List-of-Open-Source-OPC-UA-Implementations
- Limited availability of open source C++ libraries with client framework
- Identified three possible candidates:
  - FreeOPCUa with LGOL License → Build failing at time of evaluation and no releases, sparse documentation, no historical data access
  - ASNeG with Apache License → No documentation of client side, no historical data access

Choosing an Open Source Library

- Some open source implementations available: https://github.com/open62541/open62541/wiki/List-of-Open-Source-OPC-UA-Implementations
- Limited availability of open source C++ libraries with client framework
- Identified three possible candidates:
  - FreeOPCUa with LGOL License → Build failing at time of evaluation and no releases, sparse documentation, no historical data access
  - ASNeG with Apache License → No documentation of client side, no historical data access
  - Open62541 with MPL-2.0 License

Choosing an Open Source Library

- Some open source implementations available: https://github.com/open62541/open62541/wiki/List-of-Open-Source-OPC-UA-Implementations
- Limited availability of open source C++ libraries with client framework
- Identified three possible candidates:
    - FreeOPCUa with LGOL License → Build failing at time of evaluation and no releases, sparse documentation, no historical data access
    - ASNeG with Apache License → No documentation of client side, no historical data access
    - Open62541 with MPL-2.0 License → Utilization in C++ will generate warnings, but still possible

Content

1 Introduction

2 Open62541

3 Demo

Overview

- Available: https://open62541.org/
- Written in common subset of C99 and C++98, usable with all major compilers
- Supports most common OPC UA client and server services

Overview

- Available: https://open62541.org/
- Written in common subset of C99 and C++98, usable with all major compilers
- Supports most common OPC UA client and server services
- We take a look at the following client side services:
    - Read
    - Write
    - HistoryRead

Content

1. Introduction
2. Open62541
3. Demo

Test Servers

1. To test the read and write behavior of the client, we utilized one of the server implementations provided by open62541 developers

Test Servers

1. To test the read and write behavior of the client, we utilized one of the server implementations provided by open62541 developers
2. To test the HistoryRead behavior of the client, we utilized Prosys OPC UA Simulation Server available at: https://www.prosysopc.com/products/opc-ua-simulation-server/

Demonstration of a Simple Client and a Client with Historical Read

Available: https://github.com/eleicha/OPCTest