

Learn-/Training Document  
  
Siemens Automation Cooperates with Education (SCE) | As of Version V15.1 SP1

**siemens.com/sce**

TIA Portal Module 092-300

OPC UA with SIMATIC S7-1500 as OPC Server and OPC SCOUT and SIMIT as OPC Clients

**Matching SCE trainer packages for this Learn-/Training Document**

SIMATIC controllers with SIMATIC STEP 7 BASIC V15

* **SIMATIC ET 200SP Distributed Controller CPU 1512SP F-1 PN Safety**

Order no.: 6ES7512-1SK00-4AB2

* **SIMATIC CPU 1516F PN/DP Safety with Software**

Order no.: 6ES7516-3FN00-4AB2

* **SIMATIC S7 CPU 1516 PN/DP with Software**  
  Order no.: 6ES7516-3AN00-4AB3
* **SIMATIC CPU 1512C-1 PN with Software**  
  Order no.: 6ES7512-1CK00-4AB6
* **SIMATIC CPU 1512C-1 PN with Software and PM 1507**  
  Order no.: 6ES7512-1CK00-4AB1
* **SIMATIC CPU 1512C-1 PN with Software and CP 1542-5 (CP PROFIBUS)**  
  Order no.: 6ES7512-1CK00-4AB7
* **SIMATIC CPU 1512C-1 PN with Software, PM 1507 and CP 1542-5 (CP PROFIBUS)**  
  Order no.: 6ES7512-1CK00-4AB2

**SIMATIC STEP 7 Software for Training**

* **SIMATIC STEP 7 Professional V15.1 - Single License**  
  Order no.: 6ES7822-1AA05-4YA5
* **SIMATIC STEP 7 Professional V15.1 - 6+20 User Classroom License**   
  Order no.: 6ES7822-1BA05-4YA5
* **SIMATIC STEP 7 Professional V15.1 - 6+20 User Upgrade License**  
  Order no.: 6ES7822-1AA05-4YE5
* **SIMATIC STEP 7 Professional V15.1 - Student License for 20 Users**  
  Order no.: 6ES7822-1AC05-4YA5

Please note that these trainer packages are replaced with successor packages when necessary.

An overview of the currently available SCE packages is provided under:[siemens.com/sce/tp](http://www.siemens.com/tp)

**Training courses**

For regional Siemens SCE training courses, contact your regional SCE representative:

[siemens.com/sce/contact](http://www.siemens.com/contact)

**Additional information regarding SCE**

[siemens.com/sce](http://www.siemens.com/sce)

**Information regarding use**

The SCE Learn-/Training Document for the integrated automation solution Totally Integrated Automation (TIA) was prepared for the program "Siemens Automation Cooperates with Education (SCE)" specifically for training purposes for public educational and R&D institutions. Siemens does not guarantee the contents.

This document is only to be used for initial training on Siemens products/systems. This means it can be copied in whole or in part and given to trainees/students for use within the scope of their training/course of study. Disseminating or duplicating this document and sharing its content is permitted within public training and advanced training facilities for training purposes or as part of a course of study.

Exceptions require written consent from the Siemens. Send all related requests to [scesupportfinder.i-ia@siemens.com](mailto:scesupportfinder.i-ia@siemens.com).

Offenders will be held liable. All rights including translation are reserved, particularly if a patent is granted or a utility model or design is registered.

Use for industrial customer courses is expressly prohibited. We do not consent to commercial use of the training documents.

We wish to thank the TU Dresden, especially Prof. Dr.-Ing. Leon Urbas and the Michael Dziallas Engineering company and all other involved persons for the support in the preparation of this SCE   
Learn-/Training Document.

**Table of Contents**

[1 Goal 5](#_Toc16504016)

[2 Requirement 5](#_Toc16504017)

[3 Required hardware and software 6](#_Toc16504018)

[4 Theory\* 7](#_Toc16504019)

[4.1 OPC UA General 7](#_Toc16504020)

[4.1.1 Overview 7](#_Toc16504021)

[4.1.2 What is OPC? 7](#_Toc16504022)

[4.2 OPC UA address space 9](#_Toc16504023)

[4.2.1 Nodes in the address space 9](#_Toc16504024)

[4.2.2 Available type of nodes in the address space 10](#_Toc16504025)

[4.2.3 Namespaces and node IDs 11](#_Toc16504026)

[4.2.4 Attributes of the nodes 12](#_Toc16504027)

[4.3 OPC UA Security 13](#_Toc16504028)

[4.3.1 Security levels 13](#_Toc16504029)

[4.3.2 Configuration options for the security 14](#_Toc16504030)

[4.3.3 Certificate exchange between client and server 15](#_Toc16504031)

[4.4 OPC UA server of the S7-1500 16](#_Toc16504032)

[4.4.1 Supported OPC UA services of the data access of S7-1500 16](#_Toc16504033)

[4.4.2 Performance when accessing numerous tags of the server 16](#_Toc16504034)

[4.4.3 Licensing concept 16](#_Toc16504035)

[4.5 Example programs for OPC UA clients 17](#_Toc16504036)

[4.5.1 OPC Scout V10 17](#_Toc16504037)

[4.5.2 SIMIT V9.1 18](#_Toc16504038)

[4.5.3 Excel with OPC Labs QuickOPC 19](#_Toc16504039)

[4.5.4 Node-RED 19](#_Toc16504040)

[5 Task 20](#_Toc16504041)

[6 Planning 20](#_Toc16504042)

[7 Structured step-by-step instructions 21](#_Toc16504043)

[7.1 Retrieving an existing project 21](#_Toc16504044)

[7.2 Settings OPC UA Server with SIMATIC S7-1500 22](#_Toc16504045)

[7.3 Save, compile, and download the S7 station 27](#_Toc16504046)

[7.4 Archiving the TIA Portal project 28](#_Toc16504047)

[7.5 Access to SIMATIC S7-1500 via OPC-UA with OPC Scout V10 29](#_Toc16504048)

[7.6 Access to SIMATIC S7-1500 via OPC-UA with SIMIT V9.1 32](#_Toc16504049)

[7.6.1 Copy SIMIT client certificate to the certificate memory 32](#_Toc16504050)

[7.6.2 SIMIT application with "OPC UA Client" coupling 36](#_Toc16504051)

[7.6.3 Checklist – step-by-step instructions 45](#_Toc16504052)

[8 Additional information 46](#_Toc16504053)

OPC UA with SIMATIC S7-1500 as OPC Server and OPC SCOUT and SIMIT as OPC Clients

# Goal

The following pages show how the data of SIMATIC S7-1500 can be accessed via OPC UA in a project with SIMATIC S7-1500.

OPC Scout V10 and SIMIT V9.1 are used as OPC UA clients.

# Requirement

This section builds on the section Global data blocks with SIMATIC S7-CPU 1516F-3 PN/DP. To complete this section, you can use the following project, for example: "SCE\_EN\_032-600\_Global\_Data\_Blocks….".

# Required hardware and software

**1** Engineering station: Requirements include hardware and operating system   
(for additional information, see Readme on the TIA Portal Installation DVDs)

**2** SIMATIC STEP 7 Professional software in TIA Portal – as of V15.1

**3** Software OPC Scout – as of V10

**4** Software SIMIT – as of V9.1 (with dongle or in Demo mode)

**5** SIMATIC S7-1500/S7-1200/S7-300 controller, e.g. CPU 1516F-3 PN/DP –   
as of firmware V2.1 with memory card

**6** Ethernet connection between engineering station and controller



**2** SIMATIC STEP 7 Professional (TIA Portal) as of V15.1



**3** OPC Scout as of V10



**1** Engineering station



**4** SIMIT as of V9.1

**6** Ethernet connection

****

**5** SIMATIC S7-1500 controller

As of firmware V2.1

# Theory\*

## OPC UA General

### Overview

The OPC Foundation (an interest group of well-known manufacturers for the definition of standard interfaces) has defined a large number of software interfaces in recent years to standardize the information flow from the process level to the management level. In the past, different OPC (=Open Platform Communications) specifications were developed according to the different requirements within an industrial application: Data Access (DA), Alarm & Events (A&E), Historical Data Access (HDA) and Data eXchange (DX). Access to process data is described in the DA specification, A&E describes an interface for event-based information including acknowledgment, HDA describes functions for archived data and DX defines server-to-server inter-station communication.

Based on the experiences of these classic OPC interfaces, the OPC Foundation has defined a new platform called OPC Unified Architecture (UA). The target of this standard is the generic description and the uniform access to all information that must be exchanged between systems or applications. This includes the functionality of all previous OPC interfaces. In addition, the option has been created to integrate the interface natively into the respective system, regardless of the operating system that is running on the system and regardless of the programming language used to create the system.

### What is OPC?

In the past, OPC was a collection of software interfaces for data exchange between PC applications and process devices. These software interfaces were defined according to the rules of Microsoft COM (Component Object Model) and were therefore easy to integrate on Microsoft operating systems. COM or DCOM (Distributed COM) provides the functionality of interprocess communication and organizes the exchange of information between applications, even by means of computer boundaries (DCOM). Therefore, an OPC client (COM client) can exchange information with an OPC server (COM server) using mechanisms of the Microsoft operating system.

The OPC server provides process information of a device available at its interface. The OPC client connects with the server and can access the offered data.

\* From SIEMENS application example "Client example for the OPC UA server of a SIMATIC S7‑1500" [Entry ID: 109737901](https://support.industry.siemens.com/cs/document/109737901), V1.0, 06/2018

The use of COM or DCOM means that OPC server and clients can only be operated on a Windows PC or in the local network and these must implement the communication to the corresponding automation system mostly by means of proprietary protocols. For network communication between client and server, additional tunneling tools often have to be used to get through firewalls or to bypass the complicated DCOM configuration. Moreover, the interface can only be accessed natively with C++ applications, .NET or JAVA applications can only access via a wrapper layer. In practice, these limitations lead to additional communication and software layers, which increase the configuration effort and complexity.

Due to the widespread use of OPC, the standard is increasingly used for general coupling of automation systems and no longer only for the original application as a driver interface in HMI and SCADA systems to access process information.

To solve these limitations in practice and to meet the additional requirements, in the last seven years the OPC Foundation has defined a new platform with the name OPC Unified Architecture. This provides a uniform basis for the exchange of information between components and systems. OPC UA is available as IEC 62541 standard and therefore forms the basis for other international standards.

OPC UA offers the following features:

* Combination of all previous OPC features and information, such as DA, A&E and HDA, in a generic interface.
* Use open and platform-independent protocols for the interprocess and network communication.
* Internet access and communication through firewalls.
* Integrated access control and security mechanisms at protocol and application level.
* Extensive mapping options for object-oriented models; objects may have tags and methods and trigger events.
* Extensible type system for objects and complex data types.
* Transport mechanisms and modeling rules form the basis for other standards.
* Scalability from small embedded systems to enterprise applications and from simple DA address spaces to complex, object-oriented models.

## OPC UA address space

The following descriptions explain the address space of an OPC UA server.

### Nodes in the address space

A node in the OPC UA address space is of a certain type (such as object, tag or method) and is described by a list of attributes. All nodes have common attributes such as a name or description and specific attributes, such as the value of a tag. The list of attributes is not expandable. Additional information about the node can be added as a property. Properties are a special type of tag. The nodes are connected to one another with references. The references are typified. There are two main groups: Hierarchical references such as HasComponent for the components of an object or non-hierarchical references such as HasTypeDefinition for a connection of an object instance to an object type.

The following figure shows an example for nodes and the connect reference:

Variable

Attributes

* Name
* Description

Reference

Data type

Attributes

* Name
* Description

Reference

Variable

Attributes

* Name
* Description

Reference

Object

Attributes

* Name
* Description

Reference

* HasComponent
* HasComponent
* HasComponent

Figure 1.1

### Available type of nodes in the address space

The following table shows the nodes types defined in the standard:

|  |  |
| --- | --- |
| **Node type** | **Description** |
| Object | An object is used as a typified container or folder for tags, methods and events. |
| Tag | Tags represent the data of objects or tags, or as attributes the properties of a node. |
| Method | Methods are components of objects and can have a list of input or output parameters. The parameters are described using defined attributes. |
| View | Views represent a part of the address space. A node is used as an entry point and as a filter when browsing. |
| Object type | Object types provide information on the structure or the components of an object. |
| Tag type | Tag types describe which attributes or data types can be found at an instance of a tag. |
| Reference type | Reference types define the possible types of references between nodes. |
| Data type | Data types describe the content of the value of a tag. |

Table 1.1

### Namespaces and node IDs

Each node in the OPC UA address space is uniquely identified by a node ID. This node ID consists of a namespace for distinguishing identifiers from different subsystems and an identifier, which can be either a numeric value, a string or a GUID. Strings are typically use for the ID. This is analogous to OPC Data Access, where the item ID as identifier is also a string. Numeric values are used for static namespaces, such as type system. OPC UA defines a namespace with corresponding namespace index for the nodes defined by the OPC Foundation. The OPC UA servers also define one or more namespaces with index. The namespaces defined by the server are variable and can change. It is therefore recommended that the client asks for the current namespaces when setting up the session.

The following figure explains the structure of a node ID:

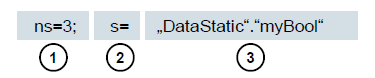


Figure 1.2

|  |  |
| --- | --- |
| 1. | Namespace index |
| 2. | Node ID type (s=String; i=Numeric: g=GUID) |
| 3. | ID |

Table 1.2

### Attributes of the nodes

The following table explains the most important node attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Node type** | **Description** |
| Node ID | All | The unique node ID within the namespace index |
| Namespace index | All | The namespace index to which the node is assigned. |
| Identifier type | All | The node ID type |
| Identifier | All | The unique node ID within the namespace index |
| Browse name | All | The browse name |
| Display name | All | The display name |
| Node class | All | The node class (object, variable, data type) |
| Description | All | Brief description of the node |
| Type definition | All | Reference to the data type description of the tag |
| Write mask | All | Write permissions on node attribute (0=no, 1=yes) without taking user groups into account |
| User write mask | All | Write permissions on node attribute (0=no, 1=yes) while taking the current user into account |
| Data type | Tag | Data type of the tag |
| Value Rank | Tag | Value type of the variable (any, scalar, vector, array) |
| ARRAY dimensions | Tag | Number or array dimensions |
| Access level | Tag | Access authorization (read, write, read/write) on the node |
| Minimum sampling interval | Tag | The smallest possible sampling interval of the tag on the server side |
| Historizing | Tag | Time lapse of the tag available on the server (yes, no) |

Table 1.3

## OPC UA Security

The following explanations explain the OPC UA security concept.

### Security levels

The following figure provides an overview of the security levels of

OPC UA:

OPC UA server

Application

Transport layer

UA Stack

OPC UA client

Application

Transport layer

UA Stack

Session

Secure channel

Socket level

Figure 1.3

The user authentication is carried out via the session. This is done, for example, by means of a user name and a password or using certificates. Mutual authentication of the application and a message-based backup of the communication are performed via a secure channel. Each message is signed and encrypted to ensure the integrity and confidentiality of the messages. These mechanisms are based on certificates (X509), which uniquely identify the applications by means of a Public Key Infrastructure (PKI) system.

On the socket level, a connection-oriented backup and socket connection via Secure Socket Layer (SSL) or Virtual Private Network (VPN) can be used in addition to or alternatively to the Secure Channel.

### Configuration options for the security

The following table describes the configuration options for the security mechanisms:

|  |  |
| --- | --- |
| **Option** | **Description** |
| Security policy | **None - No security is used in the Secure Channel.**  **Basic128Rsa15** – Set of encryption algorithms  **Basic256** – Set of advanced encryption algorithms. |
| Message security mode | **None – the messages are not backed up!**  **Sign** – the messages are signed.  **Sign&Encrypt** – the messages are signed and encrypted. |
| User authentication | **Anonymous** – No user authentication is required.  **User password** – The user authentication is carried out using a user name and a password.  **Certificate** - User authentication is performed using a certificate. |

Table 1.4

### Certificate exchange between client and server

If all participating applications implement the guidelines of OPC UA for security configuration, only one manual step (4) is required to exchange the certificates on the server, as the certificates are automatically exchanged between the applications and only an administrator has to accept the certificates.

The following figure illustrates the certificate exchange between client and server:

OPC UA client

OPC UA server

Session.Create

Server.der

Client.der

Client.der

Server.der

Figure 1.4

|  |  |
| --- | --- |
| **No.** | **Description** |
| 1. | When establishing a connection to the server (Session.Create), the client receives the server certificate via the server endpoint. |
| 2. | The client program can then decide how to handle the certificate: reject or accept it. |
| 3. | In the same process, the client sends its certificate to the server. This rejects the certificate first and stores it in a rejected folder. |
| 4. | The client certificate must then be accepted by an administrator on the server. In most cases, this is done by requiring an administrator to copy the client certificate from a rejected folder to a trusted folder. |

Table 1.5

***Note:***

* With the OPC UA server of the S7-1500, the client certificate must be loaded onto the controller before the server attempts to establish the connection via the TIA Portal in order to accept the certificate.

## OPC UA server of the S7-1500

The section provides an overview of some key data of the OPC UA server of the S7-1500. In addition, notes and tips are provided on how to use the server.

***Note:***

* For more detailed information on OPC UA server of the SIMATIC S7-1500, refer to the function manual: S7-1500, ET 200MP, ET 200SP, ET 200AL, ET 200pro Communication"   
  ([support.industry.siemens.com/cs/document//59192925](https://support.industry.siemens.com/cs/document/59192925)).

### Supported OPC UA services of the data access of S7-1500

The OPC UA server of SIMATIC S7-1500 supports the following services for the data access:

* Read
* Write
* Registered read/write
* Subscriptions

### Performance when accessing numerous tags of the server

If you want to read or write a large number of tags of a S7-1500, you can significantly improve performance by structuring the tags on the S7-1500. Use arrays and structures to declare the tags to be read/written.

Viewed individually, arrays offer the best performance. They are about 2 to 3 times faster than structures. These are about 10 to 100 faster than single accesses (with a number of about 1000 tags).

Use "Registered read/write" for recurring accesses to further increase performance.

### Licensing concept

|  |  |  |  |
| --- | --- | --- | --- |
| **CPU type** | ET 200SP CPU to S7-1513(F) | 1515/1516(F) | 1517/1518(F) |
| **Required license:** | Small | Medium | Large |

Table 1.6

Further details and information can be found in the manuals, which can be downloaded from [support.automation.siemens.com](http://support.automation.siemens.com/) and from the homepage of the OPC Foundation [opcfoundation.org](https://opcfoundation.org/).

## Example programs for OPC UA clients

A few OPC UA clients are presented as examples in the following section.

The **OPC Scout V10** and **SIMIT V9.1** software tools are included in the scope of delivery of the SCE Trainer packages with SIMATIC STEP 7 Professional V15.

**OPC Scout V10** is included on the DVD **"SIMATIC NET Networking for Industry PC Software V15"**. **SIMIT V9.1** is available as a separate DVD.

### OPC Scout V10

The **OPC Scout V10** is a support tool for the commissioning and testing of your OPC system.

The following OPC interfaces are supported:

* COM
* Data Access
* Alarms & Events
* XML (Data Access)
* OPC UA (OPC Unified Architecture)

Various functions are available for this purpose:

* Browsing and displaying the available OPC servers
* Browsing for objects with the "Discovery" function is supported for OPC UA.
* Testing connections and objects
* Monitoring items
* Reading and writing values
* Display alarms
* S7 connection diagnostics
* Creating and saving views of the objects to be acquired

### SIMIT V9.1

SIMIT is a **process simulation software** and has the following possible applications:

* Complete plant documentation
* Simulation of signals, devices and plant behavior
* Input and output simulator of test signals for an automation controller
* Testing and commissioning of automation software

SIMIT provides the following components to create a simulation:

* **Chart**  
  To build a simulation, the components available in the libraries are put together in the chart editor and suitable parameters are entered.
* **Visualization**  
  Visualizations provide an overview of your plant’s signals. Signals are visualized with controls (input and display objects) and graphical objects.
* **Coupling**  
  The coupling is the interface to the automation system and is required for signal exchange. In addition to couplings to PLCSIM, PLCSIM Advanced, PRODAVE, ... there is also a coupling with SIMIT as OPC UA client.

**Demo mode**

DEMO mode provides you with an idea of the handling and performance of SIMIT without having a valid license.

However, SIMIT has only a limited range of functions in the DEMO mode.

If you start SIMIT without a SIMIT dongle plugged into your computer, a message appears asking you if you want to launch SIMIT in DEMO mode. Confirm this message to start DEMO mode.

In demo mode it is possible to open, simulate and modify models already created. In addition, complete new models can also be created. The models created or modified in the demo mode can only be executed on the computer on which they were created.

SIMIT Simulation is limited to 45 minutes in demo mode, then the simulation must be restarted.

### Excel with OPC Labs QuickOPC

To access OPC server data from Excel, an OPC UA client library is required, which contains corresponding development components and commands.

The OPCLabs library is one such example, which can be easily integrated into an Excel worksheet.

The OPC Labs QuickOPC software with the OPCLabs library can be downloaded from the Internet at [opclabs.com](file:///C:\arbeit\00_GJ17_18\Schmitt\SCE_EN_102-101_RFID-Sensor\www.opclabs.com) . A free time-limited trial version is also available here.

***Note:***

* Please note and follow the licensing instructions for the OPC Labs QuickOPC software.

### Node-RED

Node-RED is a free tool or development environment to connect various hardware devices, APIs and online services. The software as originally developed by IBM as Proof-of-Concept and later released as Open Source software. Since then it has been continuously developed and is available free of charge to all users.

The program offers a Web interface, similar to the FBD or LAD for Siemens controllers, with which flow-based programming can be performed. The individual blocks available here are called "nodes" and are comparable to FCs or FBs. They offer inputs and outputs with which the individual nodes can be connected.

The data is transferred between the blocks in the form of messages.

In addition to the standard nodes there is an active community that develops additional nodes and makes these freely available. The public library is available on the Node-RED website: [flows.nodered.org](https://flows.nodered.org)

There is also the option to develop and use your own nodes. A documentation for this is available on the documentation page of the project: [nodered.org/docs/](https://nodered.org/docs/)

# Task

In this section, the OPC UA Server is activated and set up for the CPU from the section "SCE\_EN\_032-600\_Global\_Data\_Blocks S7-1500".

The data block "SPEED\_MOTOR[DB2]" in the CPU should be read and written with different OPC UA clients via the OPC UA server.

# Planning

The OPC UA server is set up in the properties of the CPU, which must have at least firmware version 2.1.

The security settings and the certificate and license management can also be made in these properties.

The programming device and the SIMATIC S7-1500 controller are interconnected via the **Ethernet interface**.

The data for the OPC UA server is enabled in the "SPEED\_ MOTOR[DB2]" data block.

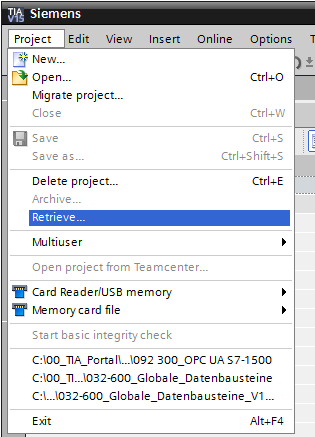
The **OPC Scout V10** and **SIMIT V9.1** software tools included in the scope of delivery of the SCE Trainer packages with SIMATIC Step 7 Professional V15 are used to test OPC UA access.

# Structured step-by-step instructions

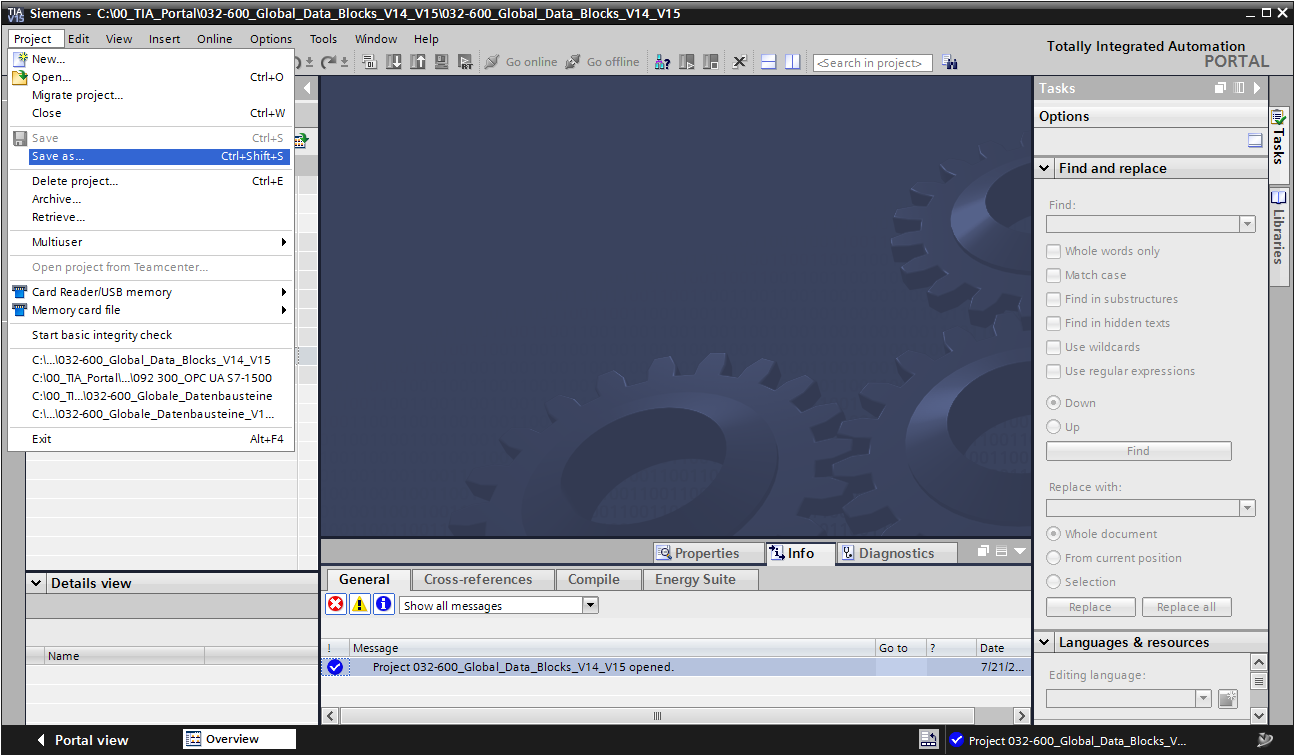
You can find instructions on how to carry out planning below. If you already have a good understanding of everything, it will be sufficient to focus on the numbered steps. Otherwise, follow the individual instructions in the steps below.

## Retrieving an existing project

* Before you can expand the "SCE\_EN\_032-600\_Global\_Data\_Blocks..." project from section "SCE\_EN\_032-600\_Global\_Data\_Blocks", you must retrieve this project from the archive. To retrieve an existing project that has been archived, you must select the relevant archive with → Project → Retrieve in the project view. Confirm your selection with "Open". (→ Project → Retrieve → Select a .zap archive … → Open)

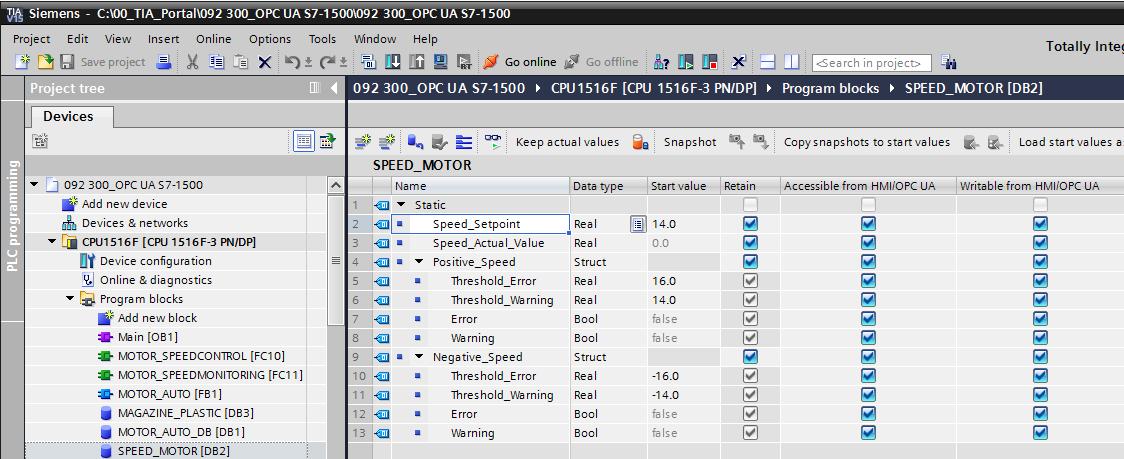


* The next step is to select the target directory where the retrieved project will be stored. Confirm your selection with "OK". (→ Target directory … → OK)
* Save the opened project under the name 092‑300\_OPC UA S7-1500.   
  (→ Project → Save as … → 092‑300\_OPC UA S7-1500 → Save)

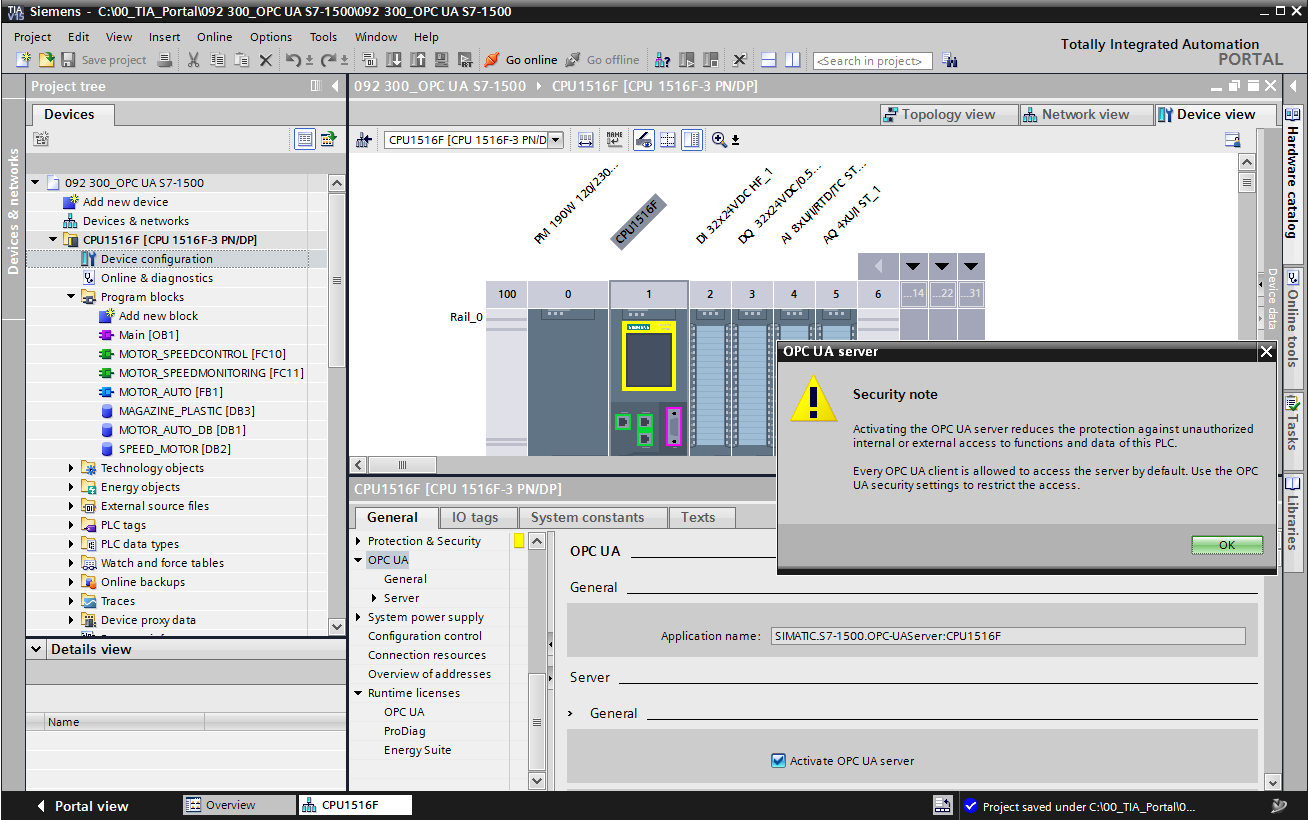


## Settings OPC UA Server with SIMATIC S7-1500

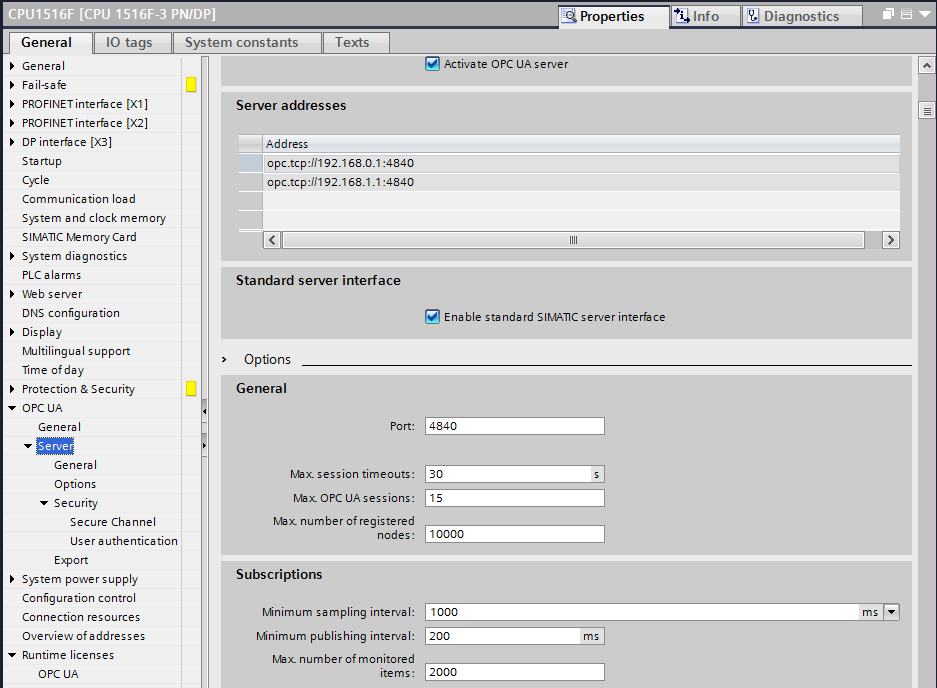
* Make sure that access to the data in the data block "**SPEED\_ MOTOR[DB2]**" is enabled by means of OPC UA. (→ SPEED\_MOTOR[DB2] →  Accessible from HMI/OPC UA →  Writeable from HMI/OPC UA)



* Enable the **"OPC UA server"** in the **"Device configuration"** of **"CPU\_1516F"** and confirm the security note. (→ CPU\_1516F → Device configuration → OPC UA →  Activate OPC UA server → OK)

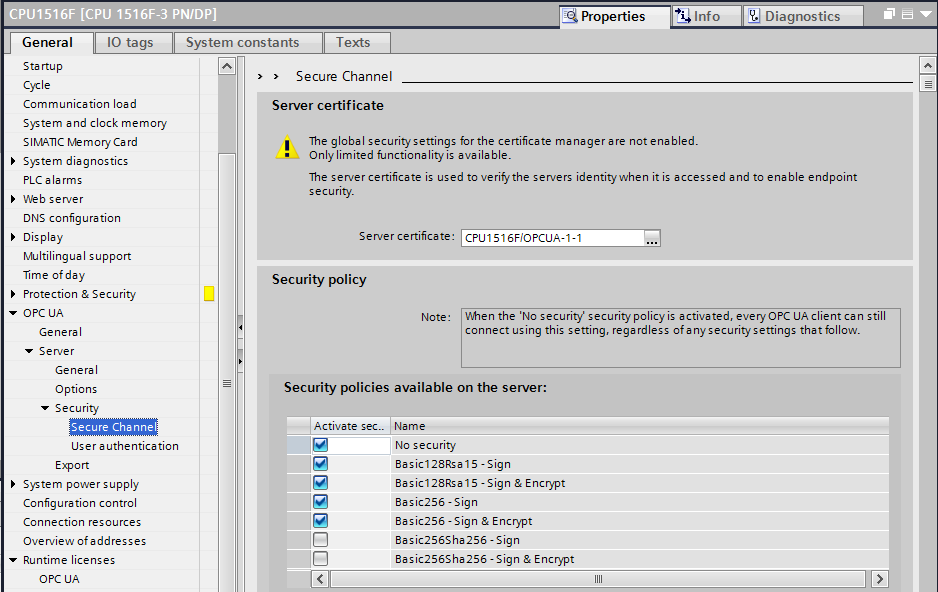


* In the **"Settings"** for the **"Server"**, select the settings displayed here for time response and number of sessions and nodes. Make a note of "**Port number**" and **"Server addresses"**, also called URLs of the server. (→ OPC UA → Server → Settings)

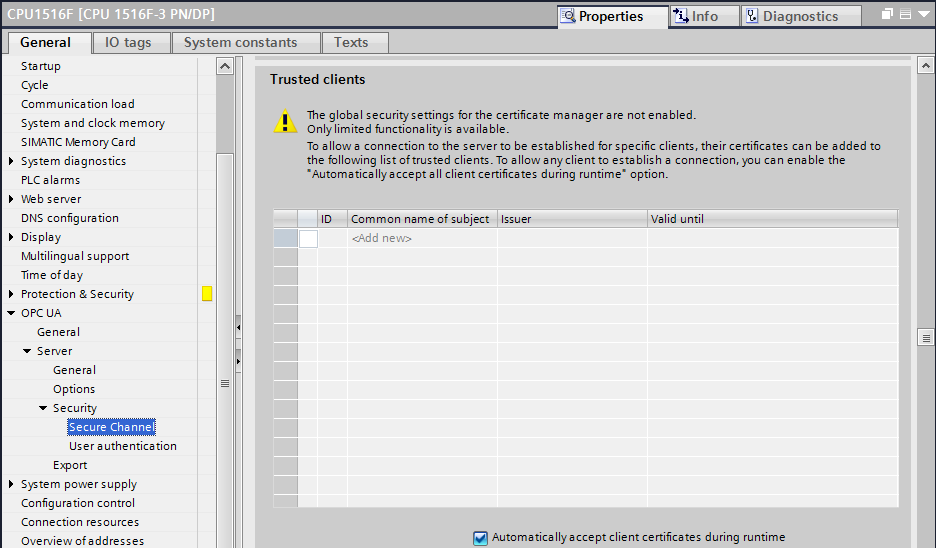


***Note:***

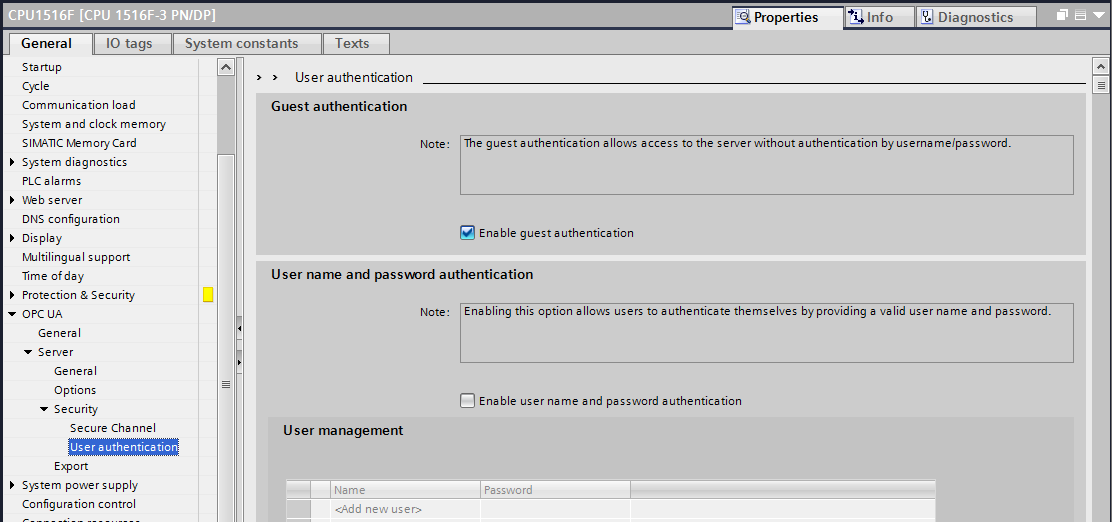
* Leave the option "Standard SIMATIC Server Interface" selected so that OPC UA clients can automatically connect to the OPC UA server of the CPU and exchange data with it.
* To simplify OPC UA access, we also allow the **"No Security"** variant for test purposes in the **"Security policy"**. (→ OPC UA → Server → Security → Secure Channel →  No Security)



* With **"Trusted clients"** we allow **"Automatically accept client certificates during runtime"**. (→ OPC UA → Server → Security → Secure Channel →  Automatically accept client certificates during runtime)



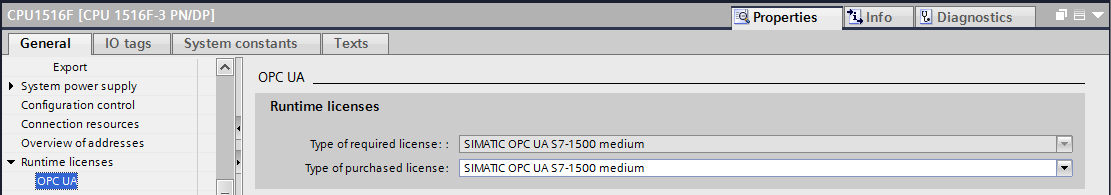
* To simplify OPC UA access, we also allow a **"Guest authentication"** for test purposes and leave the **"User name and password authentication"** disabled. (→ OPC UA → Server → Security → User authentication →  Enable guest authentication)



* The settings of the OPC UA Server interface can also be exported to support offline configuration of OPC UA clients. (→ OPC UA → Server → Export → Exporting an OPC UA XML file)

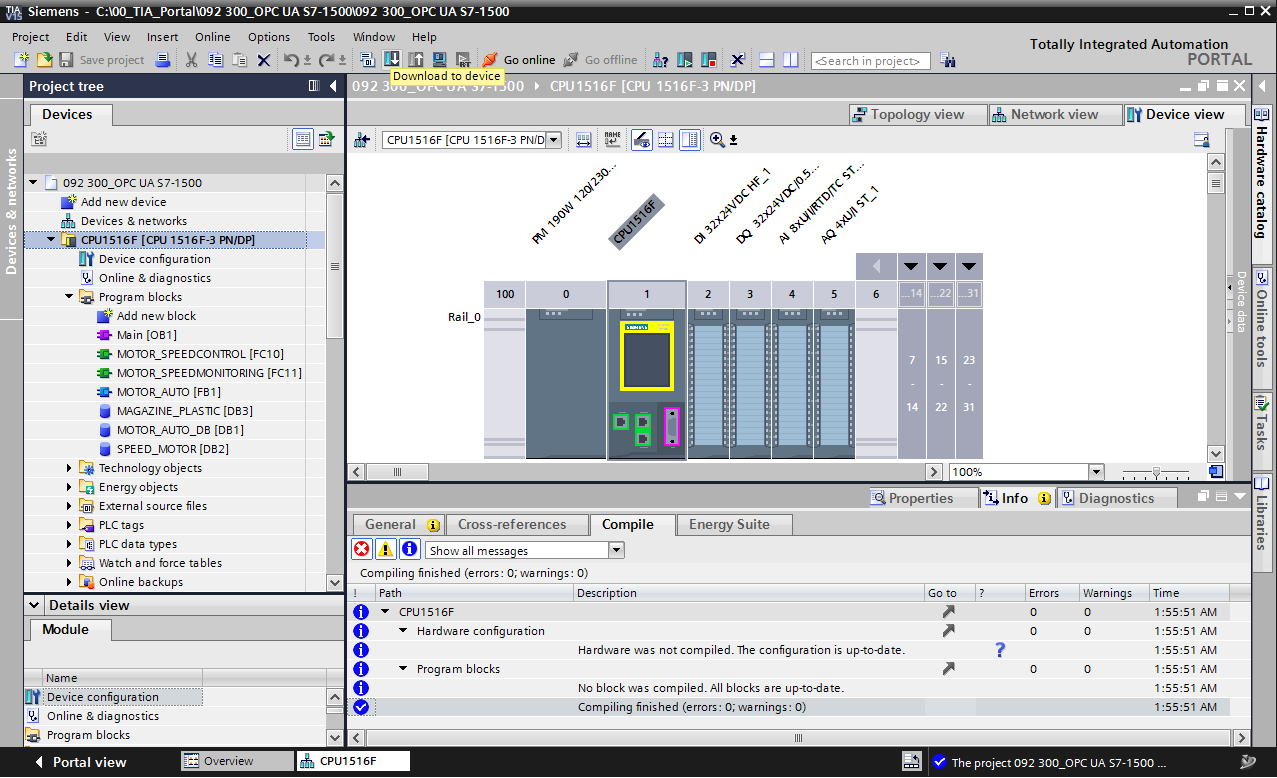


* Now also select the required **"Runtime licenses"**. (→ Runtime licenses → OPC UA → Type of purchased license → SIMATIC OPC UA S7-1500 medium)



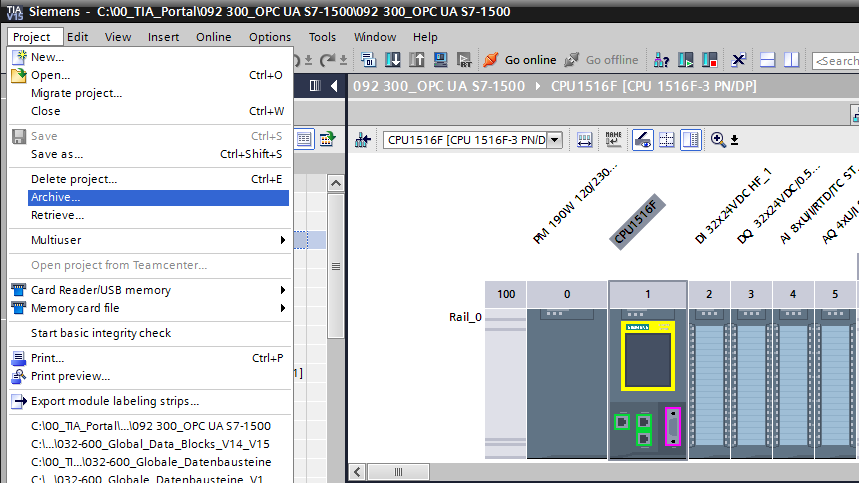
## Save, compile, and download the S7 station

* Click on the folder **"CPU\_1516F [CPU1516F-3 PN/DP]"**, compile the complete station and save the project now. After successful compilation and saving, download the station to the controller. (→ CPU\_1516F [CPU1516F-3 PN/DP] →  →  → )



## Archiving the TIA Portal project

* As the final step, you need to also archive the complete TIA Portal project. Select the   
  → 'Archive ...' command in the → 'Project' menu item. Open a folder in which you want to archive your project and save your project as the file type 'TIA Portal project archive'.   
  (→ Project → Archive → TIA Portal project archive → SCE\_EN\_092-300 OPC UA S7-1500…. → Save)

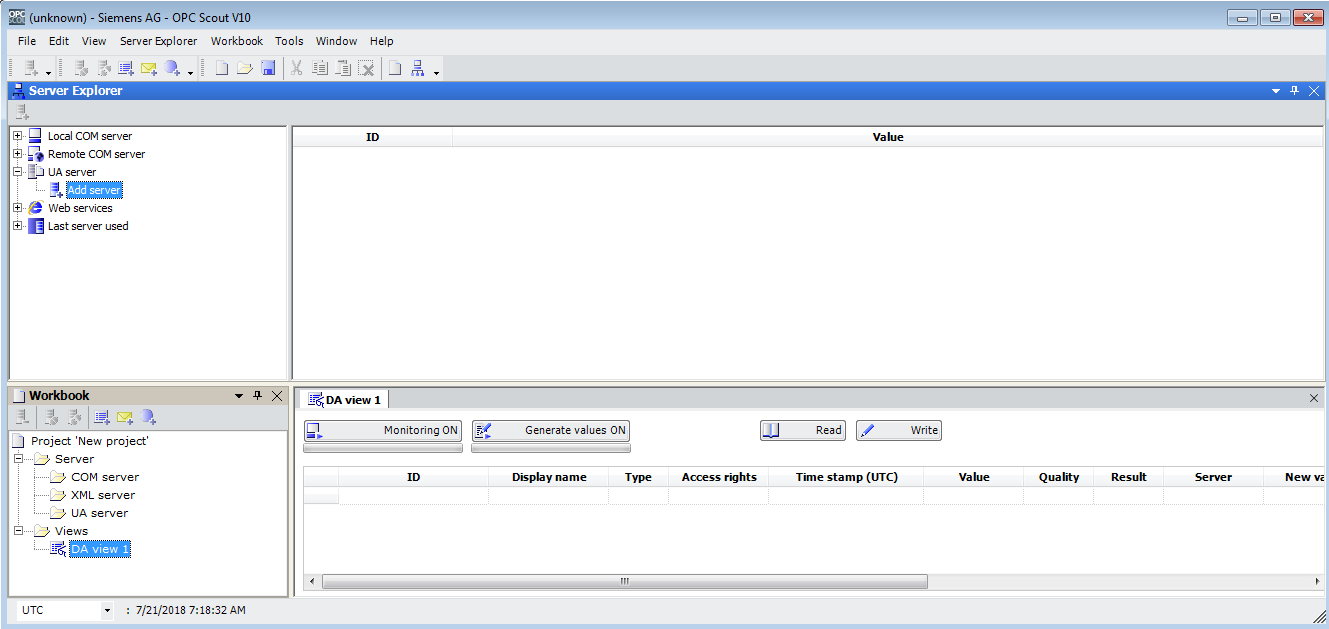


## Access to SIMATIC S7-1500 via OPC-UA with OPC Scout V10

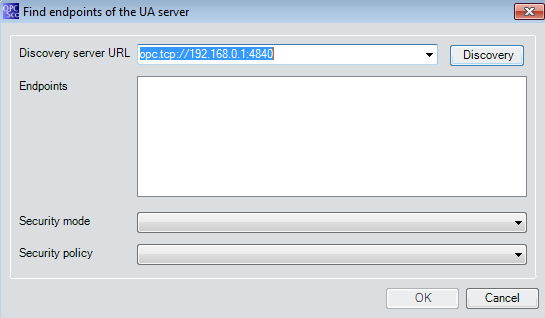
* Open the **"OPC Scout V10"** from the desktop of your programming device/PC. (→OPC Scout V10)



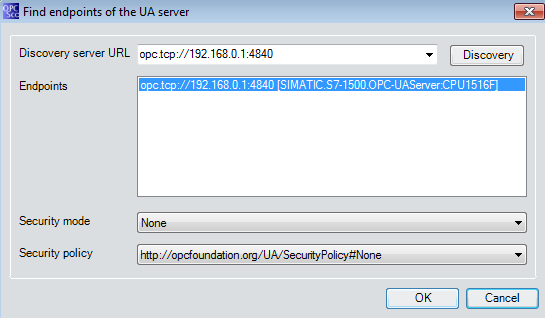
* In the left window, select the **"UA server"** and there **"Add server"**. (→ UA server → Add server)



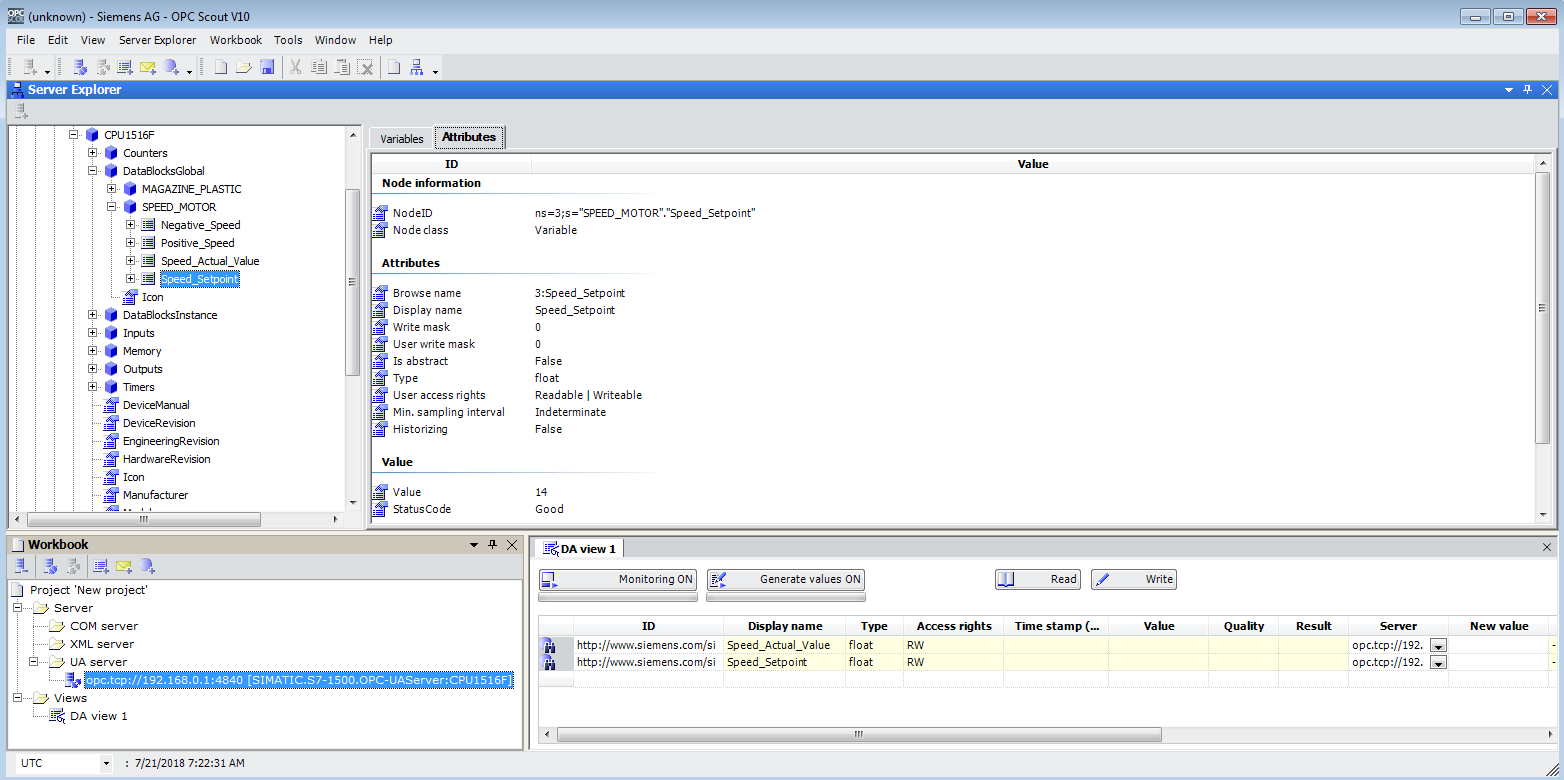
* Enter the server URL from the configuration settings of the OPC server in the SIMATIC S7-1500 and then click on **"Discovery".** (→ opc.tcp://192.168.0.1:4840 → Discovery)



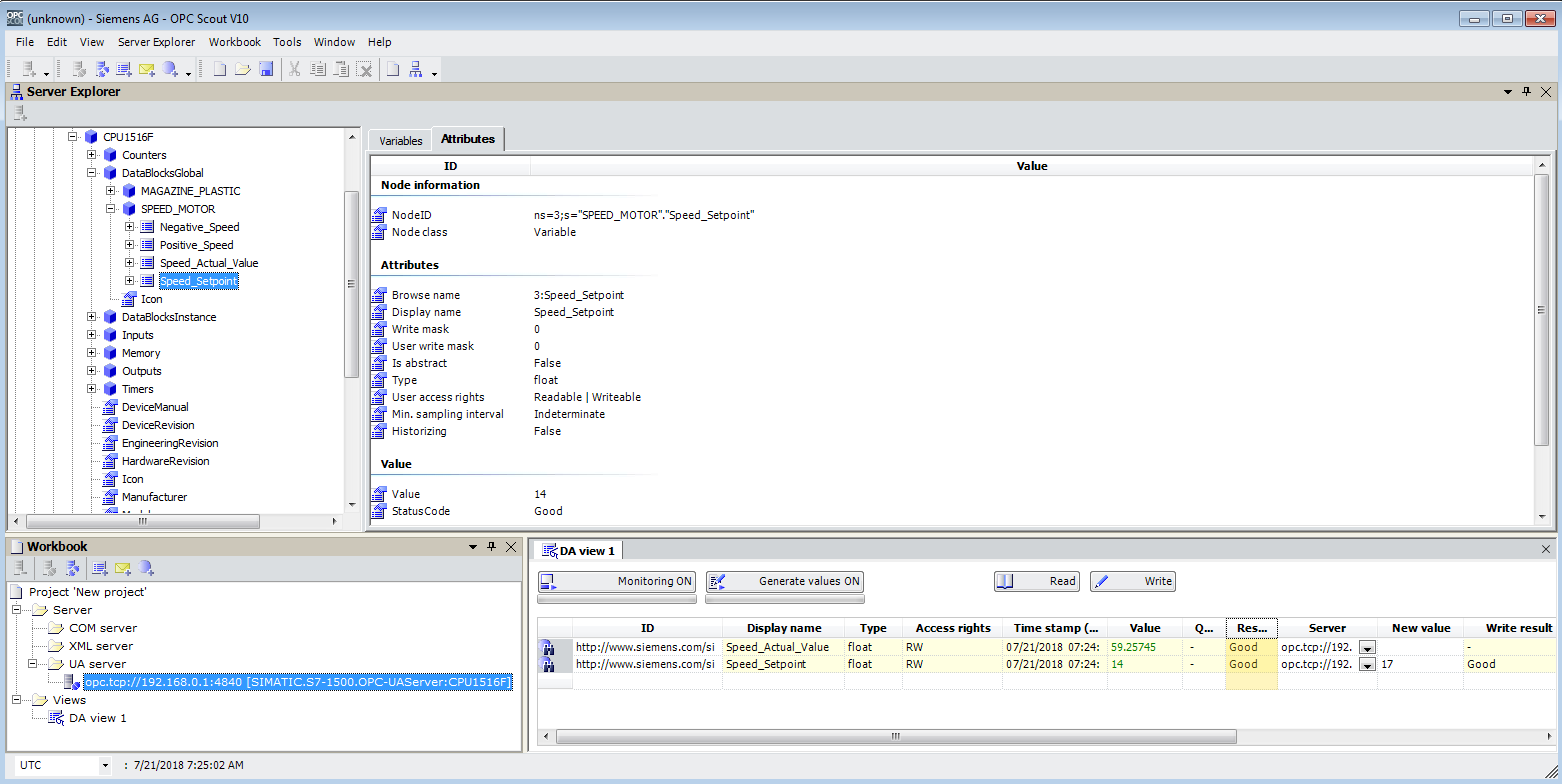
* If the end point has been found with your entries for the server URL, you can apply this with **"OK"**. (→ OK)



* In the structure of your OPC server, find the tags of your **"SPEED\_MOTOR"** data block under **"DataBlocksGlobal"**. You can drag them to the "**DA view"** area for monitoring and change. (→ UA server → opc.tcp://192.168.0.1:4840 → Objects → CPU\_1516F → DataBlocksGlobal → SPEED\_ MOTOR → Actual Speed Value → Speed Setpoint → DA view)



* In the "DA view" you can now select the tags in the **"SPEED\_ MOTOR"** data block via OPC UA "" and select new values "" there.



## Access to SIMATIC S7-1500 via OPC-UA with SIMIT V9.1

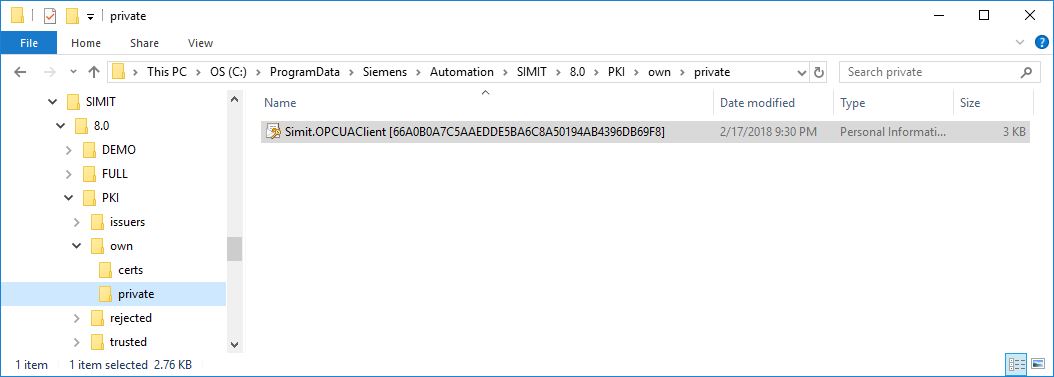
### Copy SIMIT client certificate to the certificate memory

For data exchange over OPC UA, certificates confirm the identity of the connection partner. The certificates are automatically exchanged by the OPC UA client and OPC UA server when a connection is first established. Before each subsequent connection, the system checks whether the certificates are still valid.

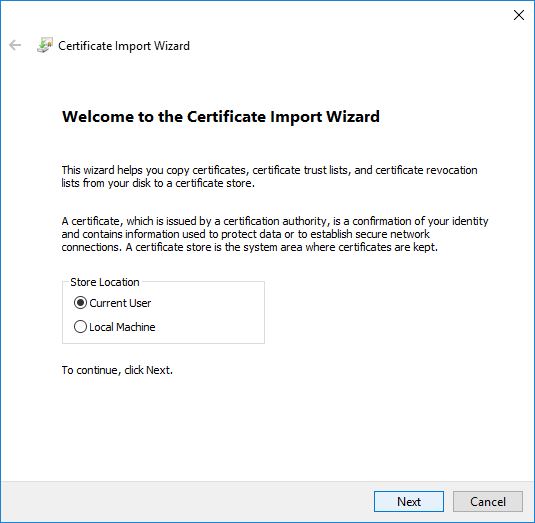
During the installation of SIMIT, the private key (certificate) of the OPC UA client SIMIT generated during the installation is stored in the directory "C:\ProgramData\Siemens\Automation\ SIMIT\8.0\PKI\own\private".

The private key is only generated once and is not overwritten when the software is updated.

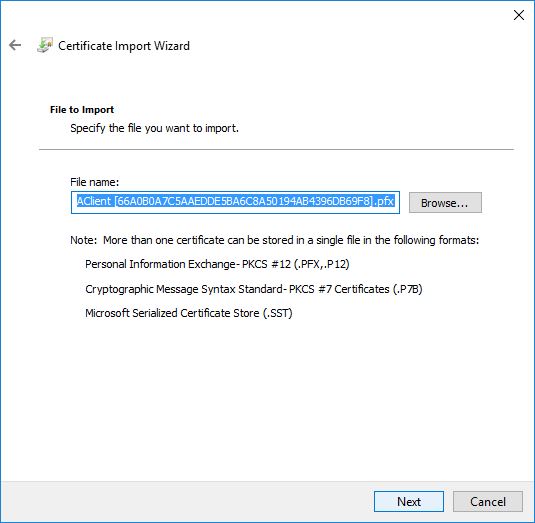
Before a connection can be established to the OPC UA server of the CPU SIMATIC S7-1500, the certificate generated during the installation of SIMIT must be copied into the certificate store of the user. For this purpose, there is a wizard which can be started by double-clicking on the single file **"Simit.OPCua Client [....].pfx"** in the folder **"C:\ProgramData\Siemens\ Automation\SIMIT\8.0\PKI\own\private"**. →C:\ProgramData\ Siemens \Automation\SIMIT\8.0\PKI\own\private → Simit.OPCUAClient [….].pfx)



* In the first dialog, specify the storage location for the certificate. (→ Current user → Next)

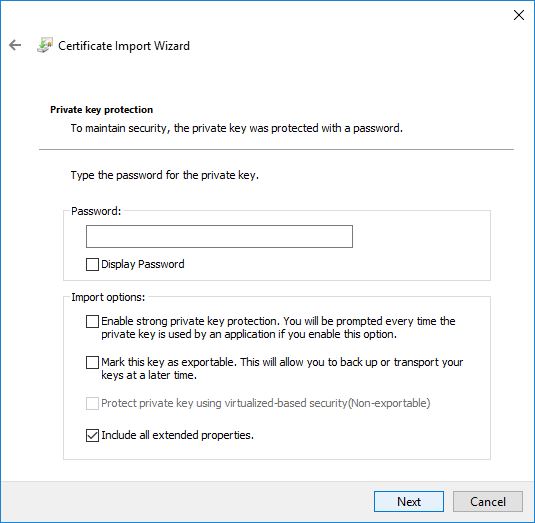


* In the second dialog, confirm the file name of the previously selected certificate. (→ Next)

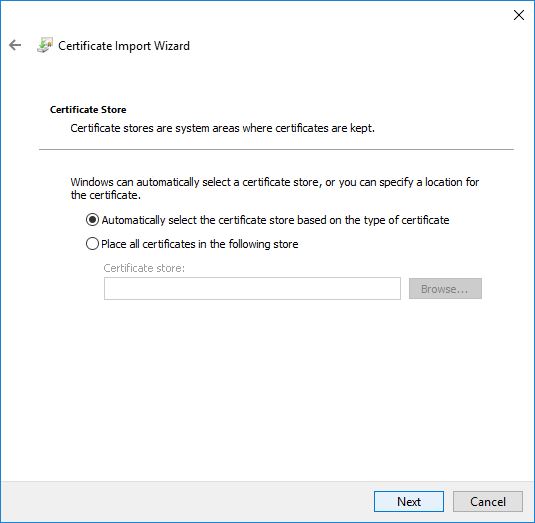


* In the following dialog, you can enter a password for the private key and select additional import options. Accept the default settings without password.

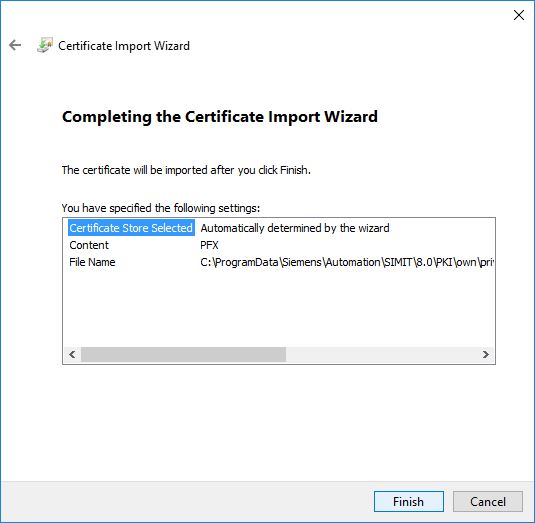
(→ Next)

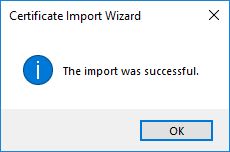


* Allow Windows to select the certificate store automatically. (→ Next)



* The selected settings for the import are then listed again. Start the import with **"Finish"** and close the alarm window with **"OK"**. (→ Finish → OK)



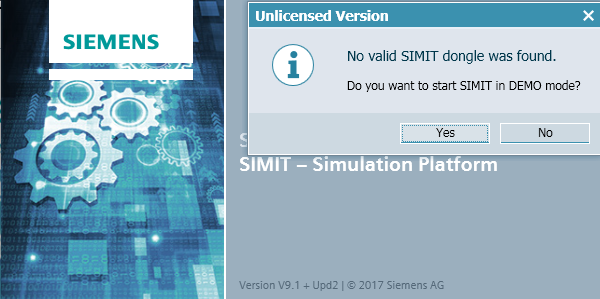


### SIMIT application with "OPC UA Client" coupling

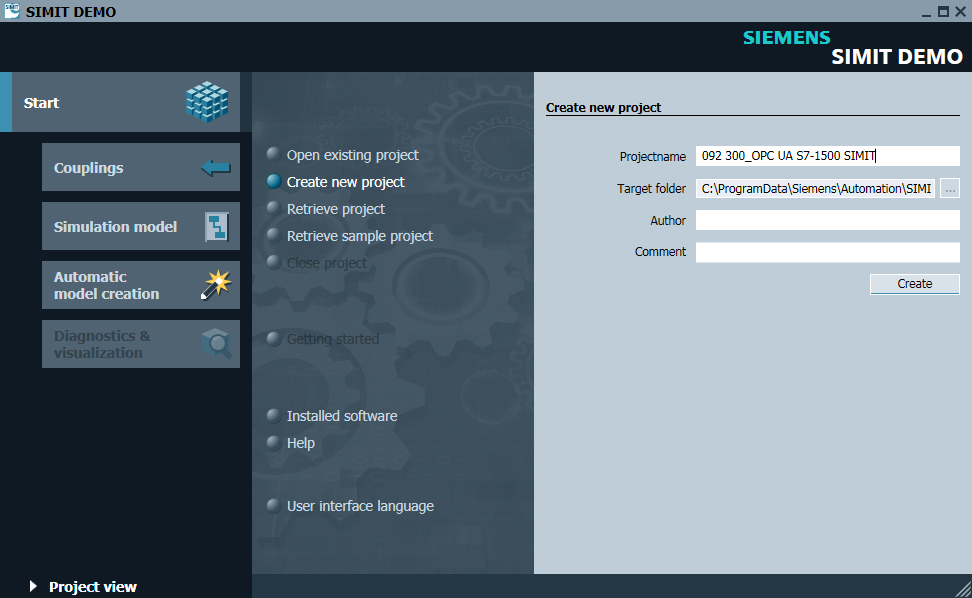
* Start SIMIT from the desktop of your computer by double-clicking on the logo for the **"SIMIT SP"** application (→ SIMIT SP)



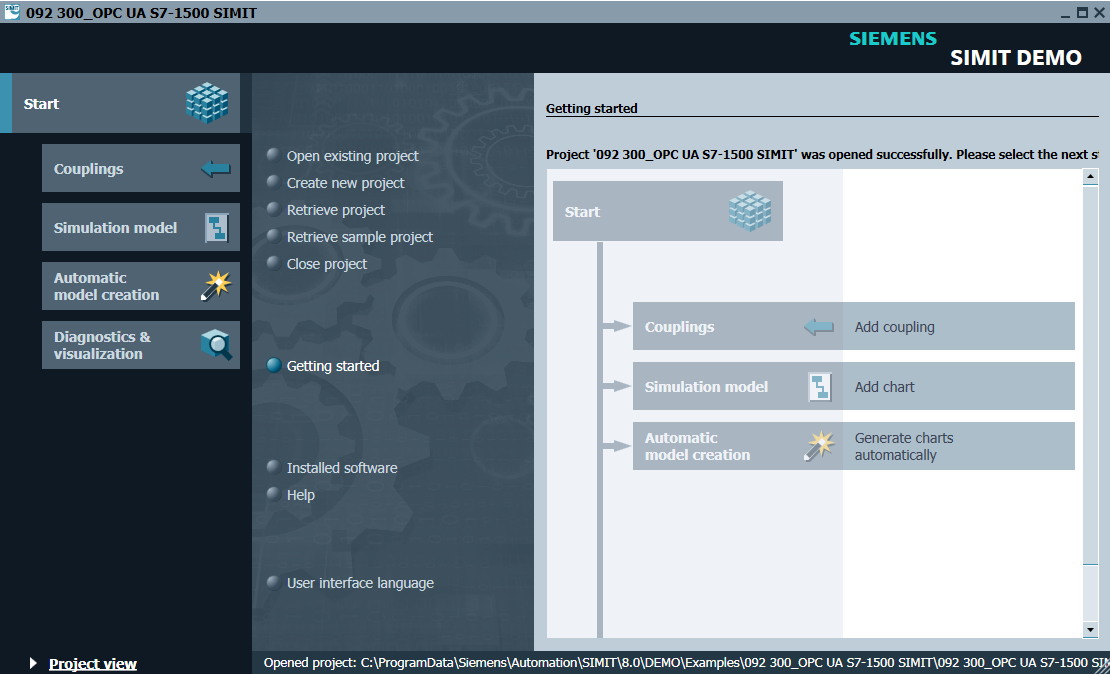
* Confirm that you want to start SIMIT in **"DEMO mode"**. (→ Yes)



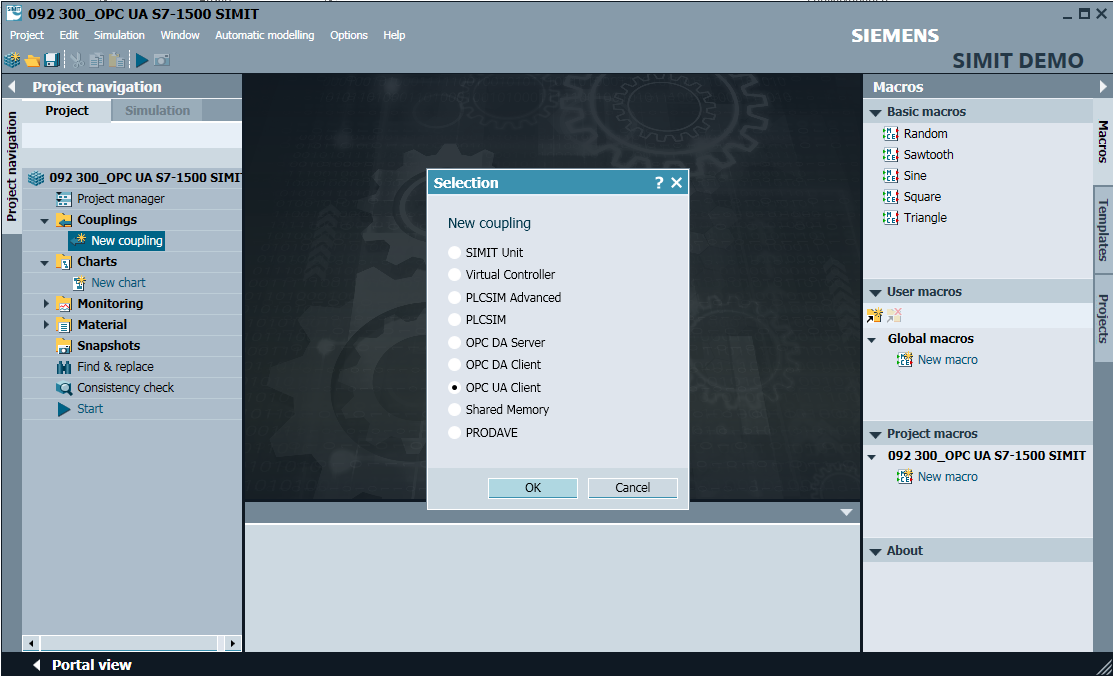
* Create a new project **"092 300\_OPC UA S7-1500 SIMIT"** project. (→ Create new project → 092 300\_OPC UA S7-1500 SIMIT → Create)



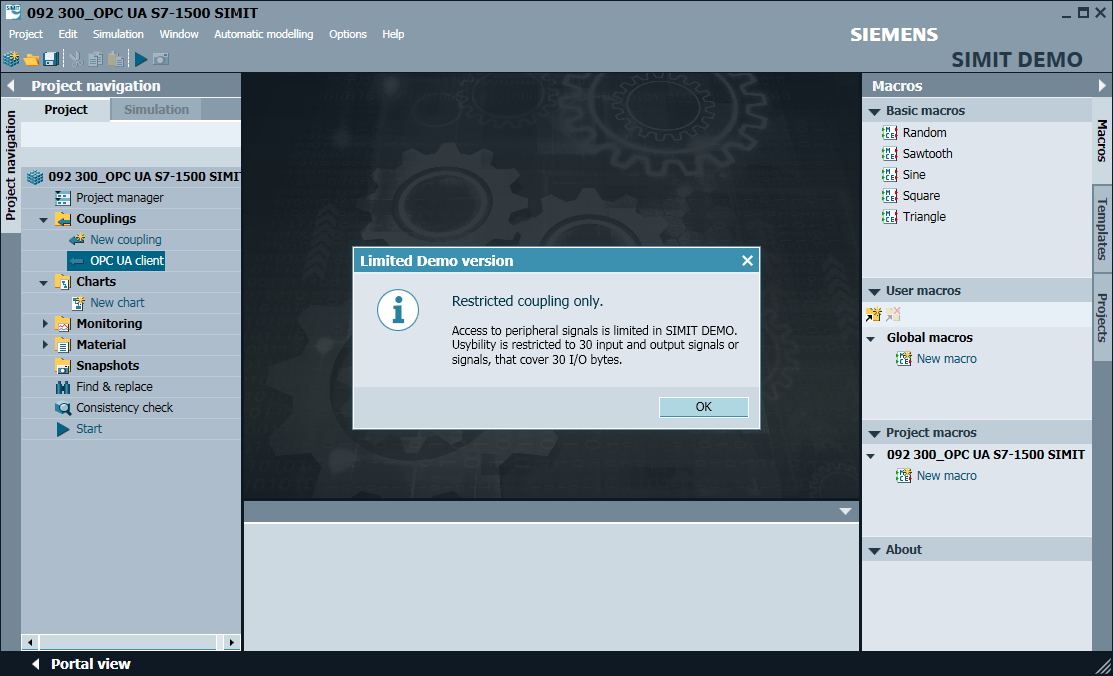
* Click here to go to the **"Project view"**. (→ Project view)



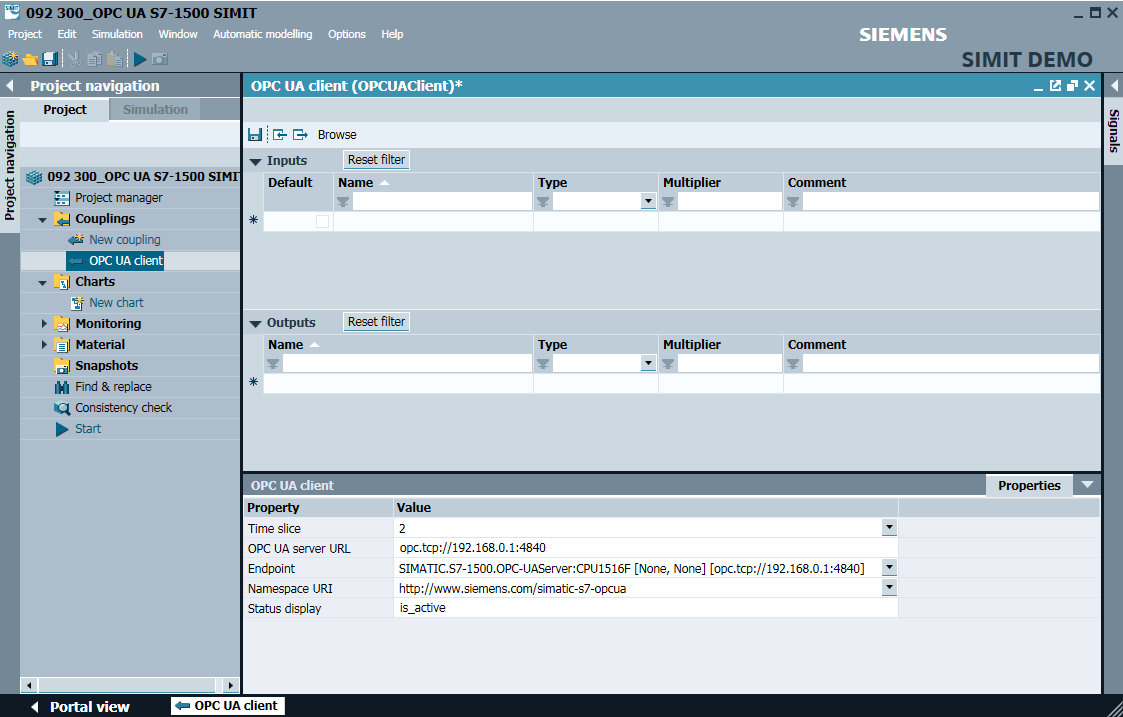
* Specify a **"New coupling" "OPC UA Client"** for your project under **"Couplings"**. (→ Couplings → New coupling → OPC UA Client → OK)

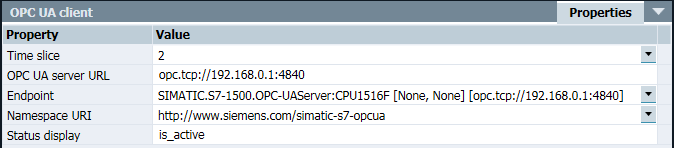


* Open the settings for the **"OPC UA client"** with a double click and confirm the note for restricted couplings in SIMIT DEMO. (→ OPC UA Client → OK)

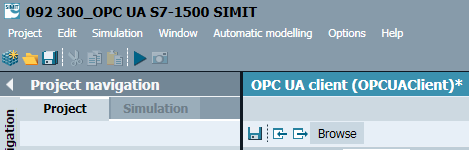


* In the **"Properties"** of the **"OPC UA client"**, enter the server URL from the configuration settings of the OPC server in SIMATIC S7-1500. Select the end point and namespace as shown here. (→ OPC UA client → properties)



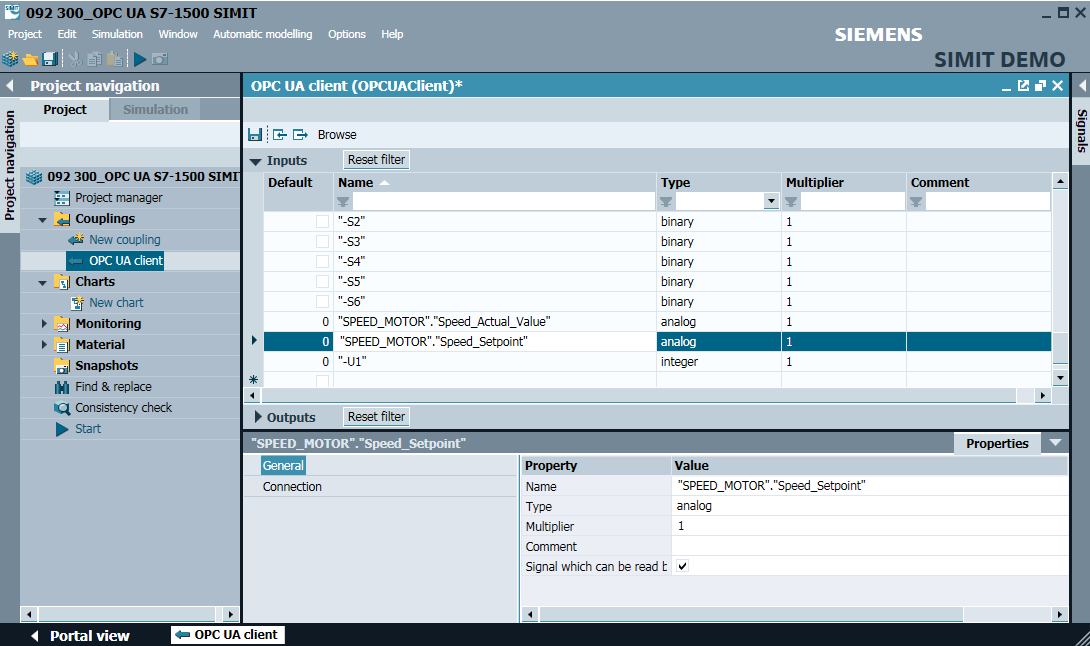


* With the next step, you use **"Browse"** to start the import of the tags enabled for OPC UA into SIMATIC S7-1500. (→ Browse)

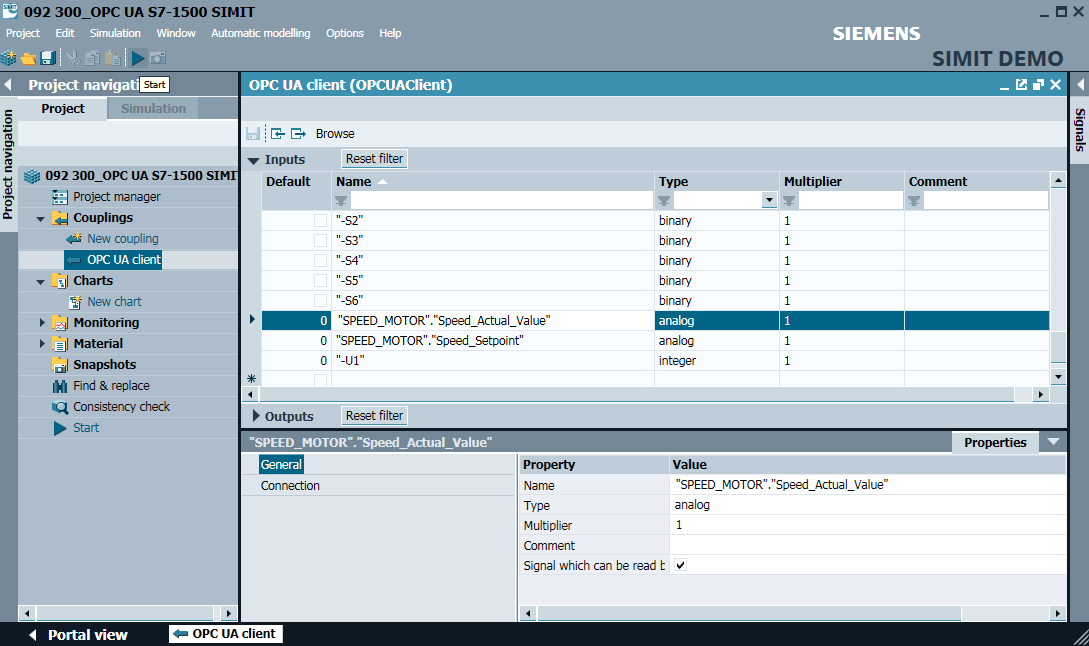


* The "Actual speed value" and "Speed setpoint" tags from the global data block "SPEED\_MOTOR" are used as "Inputs" with the name "SPEED\_MOTOR"." Actual\_Speed\_Value" and "SPEED\_MOTOR"."Speed\_Setpoint". Enable the two tags in the "Properties" under "General" "Readback- capable signal".

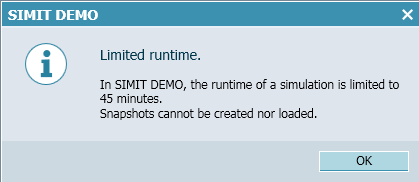
(→ "SPEED\_MOTOR"."Actual\_Speed\_Value" → Signal which can be read back  → "SPEED\_ MOTOR"."Actual\_Speed\_Value" → Signal which can be read back )



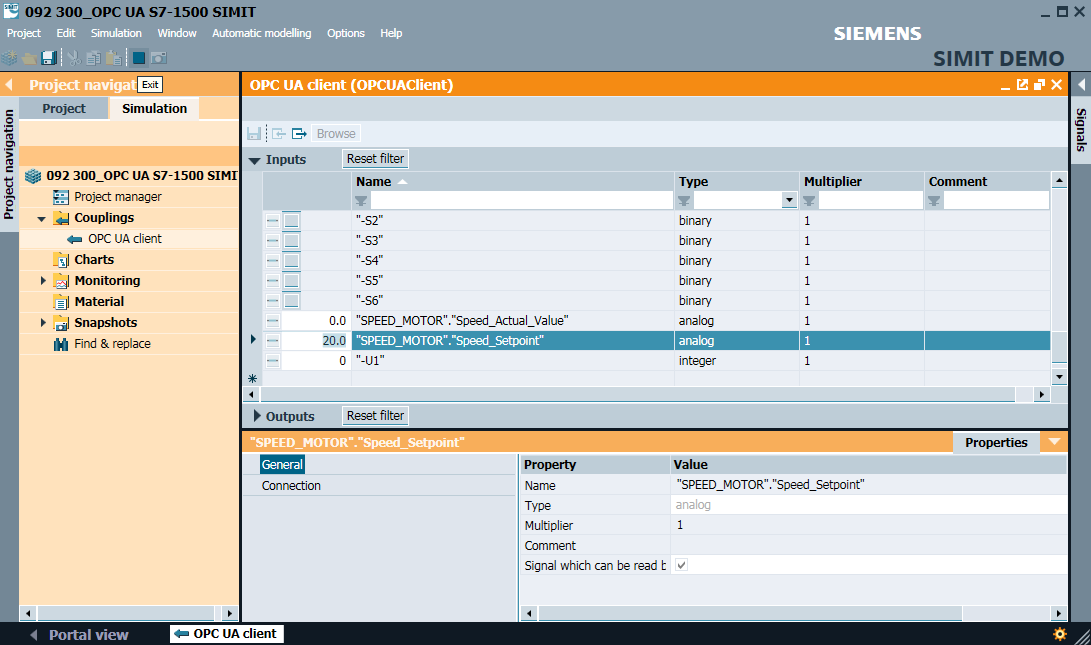
* Select **"** **Save all"** and **"** **Start"** the simulation. (→  → )



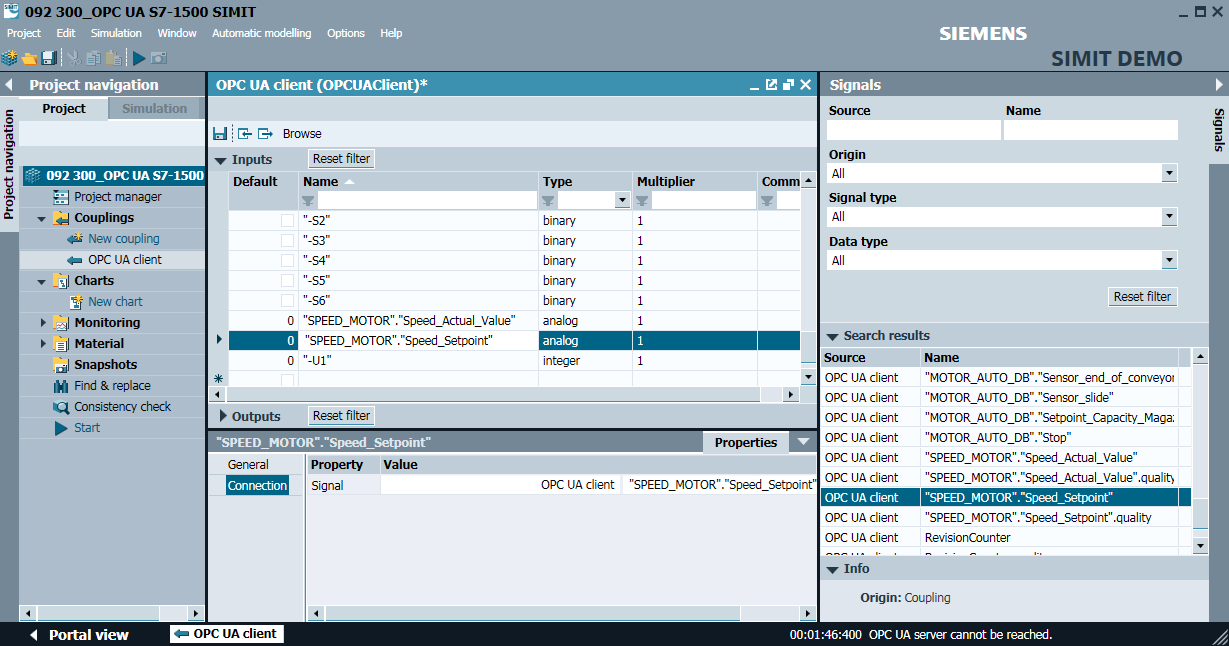
* Confirm the message on limited runtime in SIMIT DEMO. (→ OK)



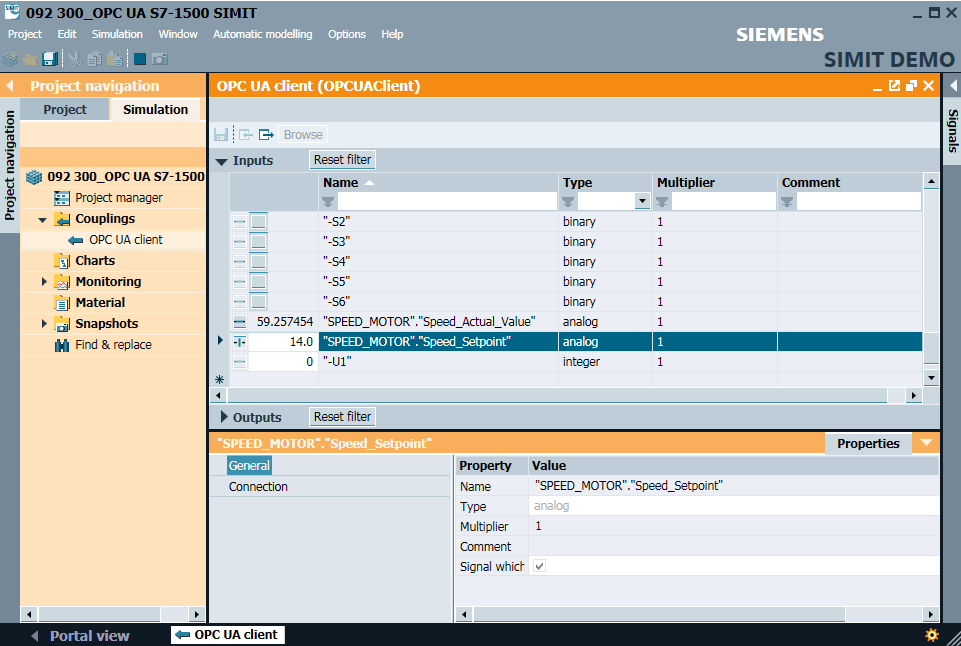
* In the I/O field in front of the "SPEED\_MOTOR"."Speed\_Setpoint", this can now already be modified and written to the controller with **"Enter"**. Cyclic reading is not yet possible. To do this, the simulation must first be closed again by clicking on "". (→ 20.0 → Enter → )



* Assign the corresponding signals of the **"Source" "OPC UA Client"** to the **"interconnections"** of the two tags **"SPEED\_MOTOR". "Actual\_Speed\_Value"** and **"SPEED\_MOTOR"."Speed\_Setpoint"** from **"Signals"**. This is done using drag-and-drop operation as shown below. (→ "SPEED\_MOTOR"."Actual\_Speed\_Value" → Connection → OPC UA client "SPEED\_MOTOR"."Actual\_Speed\_Value" → "SPEED\_MOTOR"." Speed\_Setpoint" → Connection → OPC UA Client "SPEED\_MOTOR"."Actual\_Speed\_ Value")



* Save your project with **" Save All"** and **" Start"** the simulation again. The current values from the controller are now displayed in the I/O field in front of the tags "SPEED\_MOTOR"."Actual\_Speed\_Value" and "SPEED\_MOTOR"."Speed\_Setpoint". Of course, you can still modify the "SPEED\_ MOTOR"."Speed\_Setpoint" tag. This is done by clicking on the "" field in front of the tag so that it enables the writing of the tag in the "" view. Now you can enter the desired value and write it to the controller using **"Enter"**. (→  →  →  → 13 → Enter)







### Checklist – step-by-step instructions

The following checklist helps trainees/students to independently check whether all steps of the step-by-step instruction have been carefully completed and enables them to successfully complete the module on their own.

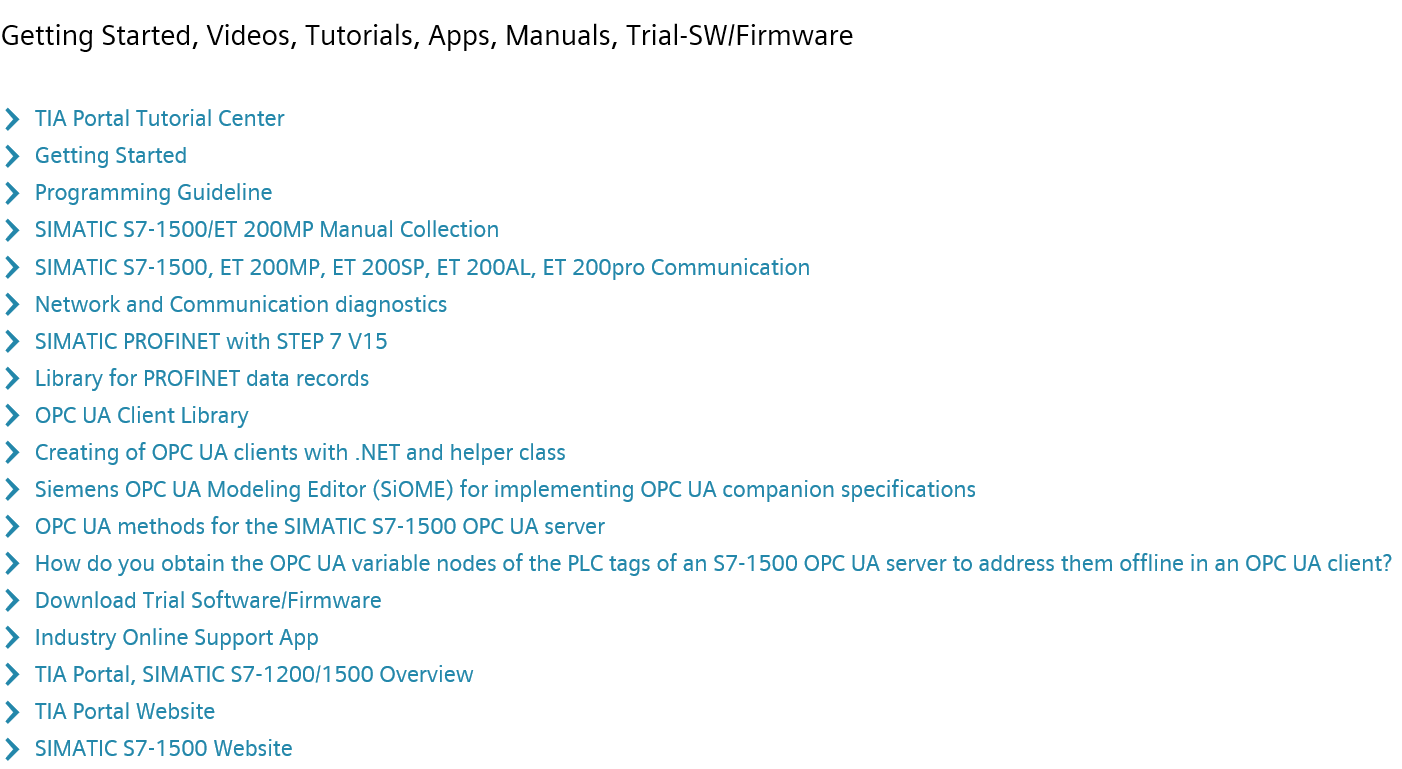
|  |  |  |
| --- | --- | --- |
| **No.** | **Description** | **Checked** |
| 1 | Values in the "SPEED\_MOTOR" data block for read and write access enabled via OPC UA |  |
| 2 | Enabling the OPC UA server in the CPU |  |
| 3 | Security settings made for the OPC UA server |  |
| 4 | Runtime license selected in the CPU |  |
| 5 | Compiling successful and without error message |  |
| 6 | Download successful and without error message |  |
| 7 | Project successfully archived |  |
| 8 | Successful test of OPC UA access with the OPC Scout |  |
| 9 | Successful test of the OPC UA access with SIMIT |  |

Table 1.7

# Additional information

You can find additional information as an orientation aid to familiarize yourself or deepen your knowledge, for example: Getting Started, videos, tutorials, apps, manuals, programming guidelines and trial software / firmware, under the following link:   
  
[siemens.com/sce/opc](http://www.siemens.com/sce/opc)

**"Additional information" preview**



Additional information

Siemens Automation Cooperates with Education  
**siemens.com/sce**

SCE Learn-/Training Document  
**siemens.com/sce/module**

SCE Trainer Packages  
**siemens.com/sce/tp**

SCE Contact Partners   
**siemens.com/sce/contact**

Digital Enterprise  
**siemens.com/digital-enterprise**

Industrie 4.0   
**siemens.com/future-of-manufacturing**

Totally Integrated Automation (TIA)  
**siemens.com/tia**

TIA Portal  
**siemens.com/tia-portal**

SIMATIC Controller  
**siemens.com/controller**

SIMATIC Technical Documentation   
**siemens.com/simatic-docu**

Industry Online Support  
**support.industry.siemens.com**

Industry Mall catalog and ordering system   
**mall.industry.siemens.com**

Siemens  
Digital Industries, FA   
P.O. Box 4848  
90026 Nuremberg  
Germany

Errors excepted and subject to change without prior notice.  
© Siemens 2019

**siemens.com/sce**