



Cognex 3D-A1000 – Setting Up Ethernet/IP

2/21/2020

Legal Notices

Legal Notices

The software described in this document is furnished under license, and may be used or copied only in accordance with the terms of such license and with the inclusion of the copyright notice shown on this page. Neither the software, this document, nor any copies thereof may be provided to, or otherwise made available to, anyone other than the licensee. Title to, and ownership of, this software remains with Cognex Corporation or its licensor. Cognex Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Cognex Corporation. Cognex Corporation makes no warranties, either express or implied, regarding the described software, its merchantability, non-infringement or its fitness for any particular purpose.

The information in this document is subject to change without notice and should not be construed as a commitment by Cognex Corporation. Cognex Corporation is not responsible for any errors that may be present in either this document or the associated software.

Companies, names, and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, nor transferred to any other media or language without the written permission of Cognex Corporation. Copyright © 2020. Cognex Corporation. All Rights Reserved.

Portions of the hardware and software provided by Cognex may be covered by one or more U.S. and foreign patents, as well as pending U.S. and foreign patents listed on the Cognex web site at:
<http://www.cognex.com/patents>.

The following are registered trademarks of Cognex Corporation:

Cognex, 2D-MAX, Advantage, AlignPlus, Assemblyplus, Check it with Checker, Checker, Cognex Vision for Industry, Cognex VSOC, CVL, DataMan, DisplayInspect, DVT, EasyBuilder, Hotbars, IDMax, In-Sight, Laser Killer, MVS-8000, OmniView, PatFind, PatFlex, PatInspect, PatMax, PatQuick, SensorView, SmartView, SmartAdvisor, SmartLearn, UltraLight, Vision Solutions, VisionPro, VisionView

The following are trademarks of Cognex Corporation:

The Cognex logo, 1DMax, 3D-Locate, 3DMax, BGAI, CheckPoint, Cognex VSoC, CVC-1000, FFD, iLearn, In-Sight (design insignia with cross-hairs), In-Sight 2000, InspectEdge, Inspection Designer, MVS, NotchMax, OCRMax, PatMax RedLine, ProofRead, SmartSync, ProfilePlus, SmartDisplay, SmartSystem, SMD4, VisiFlex, Xpand

Portions copyright © Microsoft Corporation. All rights reserved.

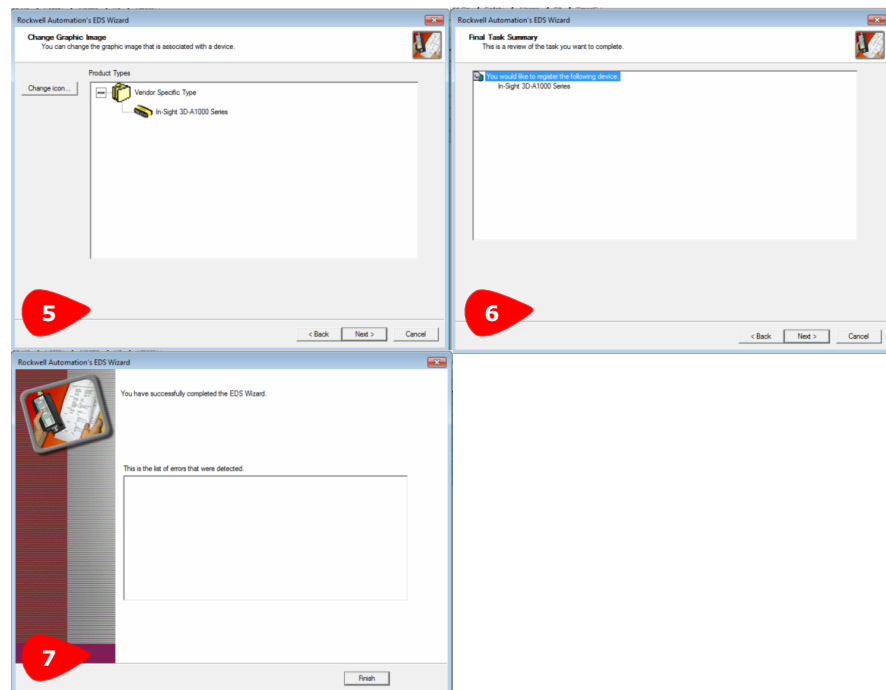
Portions copyright © MadCap Software, Inc. All rights reserved.

Other product and company trademarks identified herein are the trademarks of their respective owners.

CONTENTS

CONTENTS

Legal Notices	2
CONTENTS	3
1 Introduction	5
2 How to Use This Manual	6
3 EtherNet/IP System Information	7
4 EtherNet/IP I/O Assembly Data Format Assemblies	8
4.1 EtherNet/IP Input Assembly – Instance 11	8
4.2 EtherNet/IP Output Assembly – Instance 21	8
5 Industrial Communications Signals	9
5.1 Acquisition Signals	9
5.2 Acquisition Channels	9
5.3 Inspection Channels	9
5.4 Inspection Signals	10
5.5 System Signals	10
5.6 Cognex User Defined Types	10
6 Load/Enable/Test Communications via EtherNet/IP Protocol	11
6.1 Enabling EtherNet/IP in the 3D-A1000 Setup Wizard	11
6.2 Documentation and Sample Files	12
6.3 Loading the EDS File into RSLogix/Studio 5000	13



14

CONTENTS

6.4	Adding a Networked 3D-A1000 to Your PLC Program	14
6.5	Creating a 3D-A1000 PLC Program from Scratch	16
6.6	Loading the Sample PLC Program (A1000_EIP_Basic.ACD).....	19
6.7	Accessing Inspection Results	23
6.8	Sending Custom Parameters to the 3D-A1000.....	23

1 Introduction

This manual provides the necessary information to load, enable, and test communications via the EtherNet/IP protocol for the In-Sight 3D-A1000 device.

The In-Sight 3D-A1000 supports EtherNet/IP, an application-level protocol implemented on top of the Ethernet TCP/IP and UDP/IP layers. It shares its object model with ControlNet and DeviceNet through the Common Industrial Protocol (CIP).

EtherNet/IP provides an extensive range of messaging options and services for transferring data and I/O over Ethernet. All devices on an EtherNet/IP network present their data to the network as a series of data values called attributes. Attributes can be grouped with other related data values into sets – these are called Assemblies.

Follow these steps to enable the EtherNet/IP protocol for In-Sight 3D-A1000:

1. Using the Rockwell Software, configure it so that it recognizes the In-Sight 3D vision system.
See the `Readme_CopyRung.docx` document for the copy rung instructions.
2. Launch Rockwell Software's Electronic Data Sheets (EDS) Hardware Install Tool. It is available in the following location in the Windows Start Menu:
Programs > Rockwell Software > RSLinx > Tools > EDS Hardware Install Tool
3. Select the Electronic Data Sheets file (02A6030A0A010100.eds) for the vision system.

Note: The following information is included in the EDS file:



- Product Name = The type of the 3D vision system (such as 3D-A1000)
 - Catalog Number = The catalog number of the 3D vision system (such as 3D-A1000-XXXX)
 - Product Type (family) = In-Sight 3D Vision System
-

How to Use This Manual

2 How to Use This Manual

Throughout this manual, Warnings, Cautions, and Notes are used to draw your attention to important information in the body of the text. It is important to pay attention to the content provided under these headings.



WARNING: This symbol indicates a hazard that could cause death, serious personal injury or electrical shock.



CAUTION: This symbol indicates a hazard that could result in property damage.



Note: This symbol indicates additional information about a subject.

EtherNet/IP System Information

3 EtherNet/IP System Information

The following network identifiers are advertised on the EtherNet/IP stack:

Network Identifiers	In-Sight Vision System
Vendor ID	0x02A6
Device Type	0x30C
Product Code	0xC01
Serial Number	<mac4><mac5><mac6><0x0C>
Online Name	The name of the device (such as 3D-A1000)
SW Revision	1.1

Note: The following options are supported:



- Support for cyclic connections with 1 – 8000ms cycle times.
 - Support for cyclic connections with up to 500 bytes of data in both directions.
-

EtherNet/IP I/O Assembly Data Format Assemblies

4 EtherNet/IP I/O Assembly Data Format Assemblies

The EtherNet/IP I/O Assembly Data Format Assemblies, also known as the Assembly Object, allows data to and from each object to be sent or received over a single connection. Assembly Objects can be used to bind input or output data. The terms "input" and "output" are defined from the network's point of view; an "input" will produce data on the network (flowing from the 3D-A1000), and an "output" will consume data from the network (flowing to the 3D-A1000).

When using EtherNet/IP for communicating with Rockwell PLCs, the I/O Assembly data attribute for the input and output data has the following formats:

Input Assembly	Output Assembly
Instance 11	Instance 21

4.1 EtherNet/IP Input Assembly – Instance 11

Instance 11 is supported using the In-Sight 3D-A1000 vision system.

Figure 4-1 EtherNet/IP Input Assembly – Instance 11

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
11	0	Run Mode	Processing Enabled	Reserved					System Ready
	1	Error	Command Failed	Command Completed	Command Executing	Reserved			
	2, 3	Reserved							
	4, 5	Command Result Code							
	6, 7	Error ID							
	8, 11	Soft Event Acknowledge 0-31							
	12, 15	Reserved							
	16	Results Buffer Overrun 0	Inspection Passed 0	Results Valid 0	Inspection Completed 0	Exposure Complete 0	Acquisition Error 0	Trigger Acknowledge Camera 0	Trigger Ready Camera 0
	17	Reserved							Set User Data Acknowledge 0
	18, 19	Acquisition ID 0							
	20, 21	Inspection Image ID 0							
	22, 23	Inspection Result Code 0							
	24	Results Buffer Overrun 1	Inspection Passed 1	Results Valid 1	Inspection Completed 1	Exposure Complete 1	Acquisition Error 1	Trigger Acknowledge Camera 1	Trigger Ready Camera 1
	25	Reserved							Set User Data Acknowledge 1
	26, 27	Acquisition ID 1							
	28, 29	Inspection Image ID 1							
	30, 31	Inspection Result Code 1							
	32	Results Buffer Overrun 2	Inspection Passed 2	Results Valid 2	Inspection Completed 2	Exposure Complete 2	Acquisition Error 2	Trigger Acknowledge Camera 2	Trigger Ready Camera 2
	33	Reserved							Set User Data Acknowledge 2
	34, 35	Acquisition ID 2							
	36, 37	Inspection Image ID 2							
	38, 39	Inspection Result Code 2							
	40	Results Buffer Overrun 3	Inspection Passed 3	Results Valid 3	Inspection Completed 3	Exposure Complete 3	Acquisition Error 3	Trigger Acknowledge Camera 3	Trigger Ready Camera 3
	41	Reserved							Set User Data Acknowledge 3
	42, 43	Acquisition ID 3							
	44, 45	Inspection Image ID 3							
	46, 47	Inspection Result Code 3							
	48, 499	Inspection Results 0...451							

4.2 EtherNet/IP Output Assembly – Instance 21

Instance 21 is supported using the In-Sight 3D-A1000 vision system.

Figure 4-2 EtherNet/IP Output Assembly-- Instance 21

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
21	0	Reserved	Clear Error	Reserved	Execute Command	Reserved		Buffer Results Enable	Trigger Enable
	1	Reserved							
	2, 3	Command ID							
	4, 7	Soft Event 0-31							
	8, 15	Reserved							
	16	Reserved				Set User Data 0	Clear Exposure Complete 0	Inspection Results Acknowledge 0	Trigger Camera 0
	17	Reserved				Set User Data 1	Clear Exposure Complete 1	Inspection Results Acknowledge 1	Trigger Camera 1
	18	Reserved				Set User Data 2	Clear Exposure Complete 2	Inspection Results Acknowledge 2	Trigger Camera 2
	19	Reserved				Set User Data 3	Clear Exposure Complete 3	Inspection Results Acknowledge 3	Trigger Camera 3
	20, 499	User Data 0...479							

Industrial Communications Signals

5 Industrial Communications Signals

The In-Sight 3D-A1000 vision system uses a fixed set of pre-named signals that can be used to transfer data and results to and/or from a PLC and the vision system.

For all the signals to be active, the vision system must be configured with the correct Factory Floor Protocol (Ethernet/IP) and the *System Ready*, *Processing Enabled*, and *Run Mode* bits must be *True*.

5.1 Acquisition Signals

There is a single acquisition signal for the In-Sight 3D-A1000 vision system.

Trigger Enable: This signal gates the acquisition for the In-Sight 3D-A1000 vision system. This field is set to enable triggering via the *Trigger* bit. Clear this bit to disable the Factory Floor Protocol triggering mechanism.

5.2 Acquisition Channels

The In-Sight 3D-A1000 vision systems use a set of signals related to acquisition. These signals have timing relationships, cause actions in the camera dedicated to the acquisition channel, and report the status of the camera. Although four channels are exposed, the camera operates only on Channel 0.

Figure 5-1 EtherNet/IP Acquisition Channels

Signal	Description
Trigger Ready 0	Indicates when the 3D vision system can accept a new trigger on one of the four cameras. This field is true when the <i>Trigger Enable</i> bit is set, the camera is not currently acquiring an image.
Trigger 0	Setting the <i>Trigger</i> bit will trigger an acquisition on a specific camera. The <i>Trigger Enable</i> bit must also be set.
Trigger Acknowledge 0	Indicates when the specified camera has been successfully triggered via the <i>Trigger</i> bit. The <i>Trigger Acknowledge</i> bit will stay set until the <i>Trigger</i> bit is cleared.
Exposure Complete 0	This bit is set when the specified camera's image sensor is no longer exposed to light. Used to indicate that the part being inspected can be moved from under the camera's field of view.
Acquisition Error 0	This bit is set when the camera fails to acquire an image for one of the following reasons: - The camera's <i>Trigger Mode</i> property is not set to <i>Factory Floor Protocol</i>. - The In-Sight 3D vision system is not in <i>Run Mode</i>. - The <i>\$System.ProcessingEnabled</i> tag is <i>False</i>. - The camera is currently acquiring an image. - The camera is out of image buffers. This bit is cleared when a successful acquisition is triggered, or the <i>Trigger Enable</i> bit is toggled. Note: If the <i>Trigger</i> bit is set while the <i>Trigger Enable</i> bit is Low, this will result in an acquisition error, because the <i>Trigger</i> bit is ignored.
Acquisition ID 0	This ID increments on the completion of every acquisition, regardless of the trigger source, and can be used to synchronize an acquisition with its <i>Inspection Result</i> . The acquisition ID is changed on the start of an acquisition. If the PLC is the trigger source, the <i>Acquisition ID</i> is set prior to setting the <i>Trigger Acknowledge</i> bit.
Inspection Image ID 0	These bytes are used to provide the acquisition IDs for each of the images associated with this set of results. This data is passed via <i>Acquire</i> block and the <i>Show ID Pins</i> output pin to a <i>Notify PLC</i> block that is configured to use the <i>InspectionComplete Notification Method</i> Type, or the <i>NotifyInspectionComplete</i> function is executed. You can specify multiple cameras in this call, and each <i>Inspection Channels</i> can report a single image ID from each camera that was used to generate a set of results.

5.3 Inspection Channels

There is a single group of signals used to transfer data to the In-Sight 3D-A1000 vision system and to send inspection result data to the PLC. It supports storing a copy of the *User Data* when the *Set User Data* bit is set and publishing a set of inspection results from the vision system. Note that the application itself currently has no support for User Data.

Figure 5-2 EtherNet/IP Inspection Channels

Signal	Description
Inspection Completed 0	This bit is toggled when the <i>Notify PLC</i> block or <i>NotifyInspectionComplete</i> function are executed. You specify which of the four inspection channels to send the results through.
Results Valid 0	This bit is set when the <i>Inspected Image ID</i> , <i>Inspection Result Code</i> and <i>Inspection Results</i> fields are valid. The bit is cleared when the <i>Inspection Results Acknowledge</i> bit is set.
Results Buffer Overrun 0	This bit is set when the <i>Enable Results Buffer</i> bit is set, and there are more results than the result queue can hold.
Inspection Result Code 0	These bytes are used to pass the user-defined result code. This code is defined in either the <i>Notify PLC</i> block or <i>NotifyInspectionComplete</i> function, using the <i>resultCode</i> parameter.
Inspection Passed 0	This bit indicates the pass and/or fail state of the inspection. This bit is defined in either the <i>Notify PLC</i> block or <i>NotifyInspectionComplete</i> function, using the <i>inspectionPassed</i> parameter.
Inspection Results Acknowledge 0	When the <i>Buffer Results Enable</i> bit is set, the <i>Inspection Results Acknowledge</i> bit acknowledges that the PLC has received the <i>Inspected Image ID</i> , <i>Inspection Result Code</i> and <i>Inspection Results</i> data.
Set User Data 0 Set User Data Acknowledge 0	These bits are used to signal that new <i>User Data</i> is available to be read in one of the four inspection channels in the vision system. After latching the entire <i>User Data</i> area into a holding buffer for the specified inspection channel, the vision system will indicate to the PLC that it has read the data, and it can start loading new data into the buffer, if required, by issuing the <i>Set User Data Acknowledge</i> signal.

Industrial Communications Signals

5.4 Inspection Signals

The following signals are used to send and/or receive data on the In-Sight 3D vision system and the connected PLC.

Figure 5-3 EtherNet/IP Inspection Signals

Signal	Description
Buffer Results Enable	When the <i>Buffer Results Enable</i> bit is set, the vision system will queue up to 16 sets of results, per inspection Channel. Each result must be explicitly acknowledged using the <i>Inspection Results Acknowledge</i> signal.
Inspection Results	These bytes are used to create a collection of vision inspection tool results. You define the size of the <i>Inspection Results</i> using the <i>resultData</i> parameter, for each inspection. This allows you to define, for example, the <i>Inspection Results 0</i> section to be larger than the <i>Inspection Results 2</i> section. Note: For more information about the <i>resultData</i> parameter, see the <i>NotifyInspectionComplete</i> topic.
User Data	This is data that is sent to the vision system from the PLC, and the size is user-defined for each inspection Channel. This allows you to define, for example, the <i>User Data 0</i> section to be larger than the <i>User Data 2</i> section. When the PLC sends the <i>User Data</i> for inspection Channel 1, the data should only be written into the area reserved for inspection Channel 1, and then the <i>Set User Data 1</i> bit should be set.

5.5 System Signals

These signals are used to send and/or receive information about the In-Sight 3D vision system status. Note that the application currently does not support Soft Events.

Figure 5-4 EtherNet/IP System Signals

Signal	Description
Run Mode	This bit indicates that the vision system is in <i>Run Mode</i> and the <i>\$System.ProcessingEnabled</i> tag is <i>True</i> .
Processing Enabled	This bit indicates the state of the <i>\$System.ProcessingEnabled</i> .
Soft Event 0-31 Soft Event Acknowledge 0-31	The <i>Soft Event</i> allows the PLC to trigger a script associated with a <i>Soft Event</i> script point. The rising edge of this bit will trigger the associated <i>Soft Event</i> script, and the falling edge will trigger the associated <i>Soft Event Off</i> script. The <i>Soft Event Acknowledge</i> will be generated after the associated script finishes its execution. Note: For more information about the <i>Soft Event</i> script point, see the <i>FactoryProtocol Script Points</i> topic.
Error Error ID Clear Error	The <i>Error</i> bit is set in conjunction with either the <i>Notify PLC</i> block set to the <i>Error Notification Method Type</i> , or the execution of the <i>NotifyError</i> function. The <i>Error ID</i> field holds a user-defined error code, defined using the <i>errorid</i> parameter. The <i>Clear Error</i> bit can be used to acknowledge the current error code, which will clear the <i>Error</i> and <i>Error ID</i> signals. If another <i>Error ID</i> signal was sent, the <i>Error</i> bit and <i>Error ID</i> signal will be set to queued error ID when the <i>Clear Error</i> signal is dropped.
System Ready	This bit indicates when the In-Sight 3D vision system has fully started and is ready to accept and process triggers.

5.6 Cognex User Defined Types

Cognex provides user defined types (UDTs) with the 3D-A1000. The UDTs simplify access to the signals described above by reducing byte offset computation and improving code readability. See *Section 6.2* for instructions describing how to obtain the UDTs.

There are four UDTs, as shown in *Figure 5-5* and described as follows:

- **A1000_IN** maps to the entire input assembly, including the status data (bytes 0 → 47) and the inspection results (bytes 48 and onward), in accordance with *Figure 4-1*. More details regarding accessing specific inspection results can be found at the end of *Section 6.6*.
- **A1000_STATUS** maps to the status data portion of the **A1000_IN** UDT.
- **A1000_OUT** maps to the output assembly, including the control signals (bytes 0 → 19) and user data (bytes 20 and onward), in accordance with *Figure 4-2*.
- **A1000_CONTROL** maps to the control signals portion of the **A1000_OUT** UDT.

Figure 5-5 EtherNet/IP UDTs

Name	Alias For	Base Tag	Data Type
A1000D_in			A1000_IN
A1000D_in.Status			A1000_STATUS
A1000D_in.InspectionResults			SINT[452]
A1000D_out			A1000_OUT
A1000D_out.Control			A1000_CONTROL
A1000D_out.UserData			SINT[480]

NOTE: If you choose to *not* use the predefined UDTs, access to the inspection results begins at byte 48 of the input assembly.

Load/Enable/Test Communications via EtherNet/IP

6 Load/Enable/Test Communications via EtherNet/IP Protocol

This section details the steps needed to load, enable, and test EtherNet/IP communications.

6.1 Enabling EtherNet/IP in the 3D-A1000 Setup Wizard

COGNEX 3D-A1000 Dimensioning System

Get Started > Application Details > Baseline > Optimize > **Communications**

Industrial Protocol

- ☐ TCP Server
- ☐ TCP Client
- ☐ DataMan MultiReader Sync
- ☒ **Industrial Protocols**
- ☐ Discrete I/O and Encoder

EtherNet/IP

3D-A1000 can communicate over standard Industrial Protocols

Output Delay

- ☒ None
- ☐ Time-Based 3000 ms
- ☐ Distance-Based 300 mm

Output delay applies only to hardware triggers. The delay is in relation to the time or location where the trigger occurs.

Sample Output String:

1,772,248,453<CR><LF>

Test Outputs

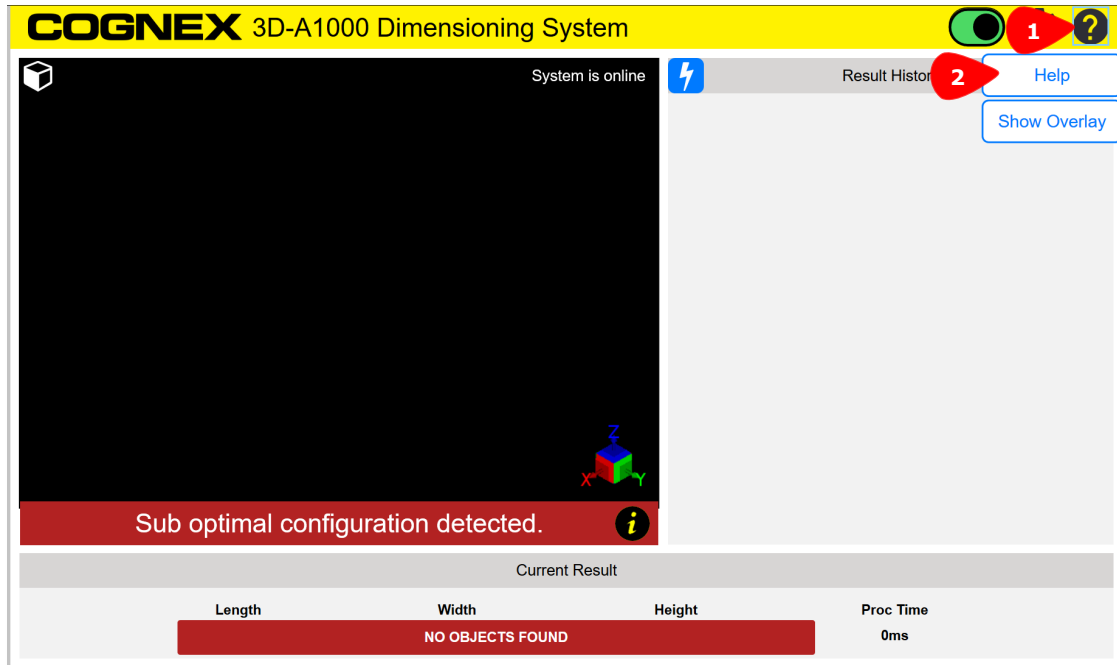
< Previous **Finish**

Note: A reboot will be required after completing the change

Load/Enable/Test Communications via Ethernet/IP

6.2 Documentation and Sample Files

The 3D-A1000 includes all necessary PLC files, which are available through the device's help page.



CONTACT COGNEX

HAVE QUESTIONS?

Cognex representatives are available to assist.

Cognex Headquarters

Cognex Corporation
One Vision Drive
Natick, MA 01760-2059

Sales : +1-855-426-4639

Support : +1-508-650-4100

Video Tutorials

- [Getting Started](#)
- [Application Details](#)
- [Autotune](#)
- [Baseline](#)
- [Communications](#)
- [CAT](#)

Supplemental Documents

- [Release 1.0 Box Guide Reference](#)
- [Cognex 3D-A1000 EIP](#)
- [Cognex 3D-A1000 Home](#)
- [Cognex 3D-A1000 Setup Tool](#)
- [A1000R Drawing \(PDF\)](#)
- [A1000R Drawing \(Step\)](#)
- [A1000R Mounting Bracket \(Step\)](#)

Supplemental Software

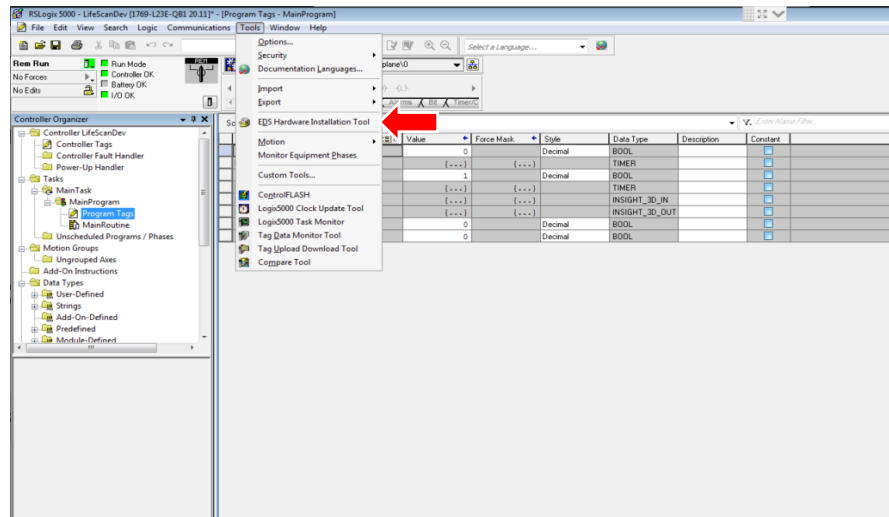
- [Download Dataman Setup Tool](#)
- [EDS Installation Files](#)
- [RSLogix/Studio 5000 UDTs](#)
- [RSLogix/Studio 5000 Example Program](#)

Load/Enable/Test Communications via EtherNet/IP

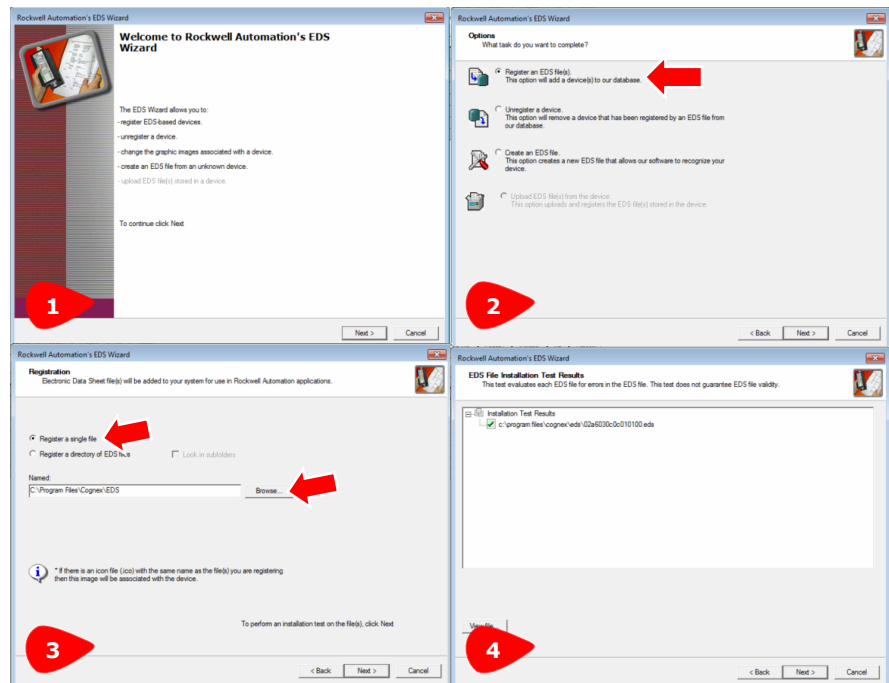
6.3 Loading the EDS File into RSLogix/Studio 5000

1. Download and unzip the EDS installation and UDT files onto your PC running RSLogix/Studio 5000.

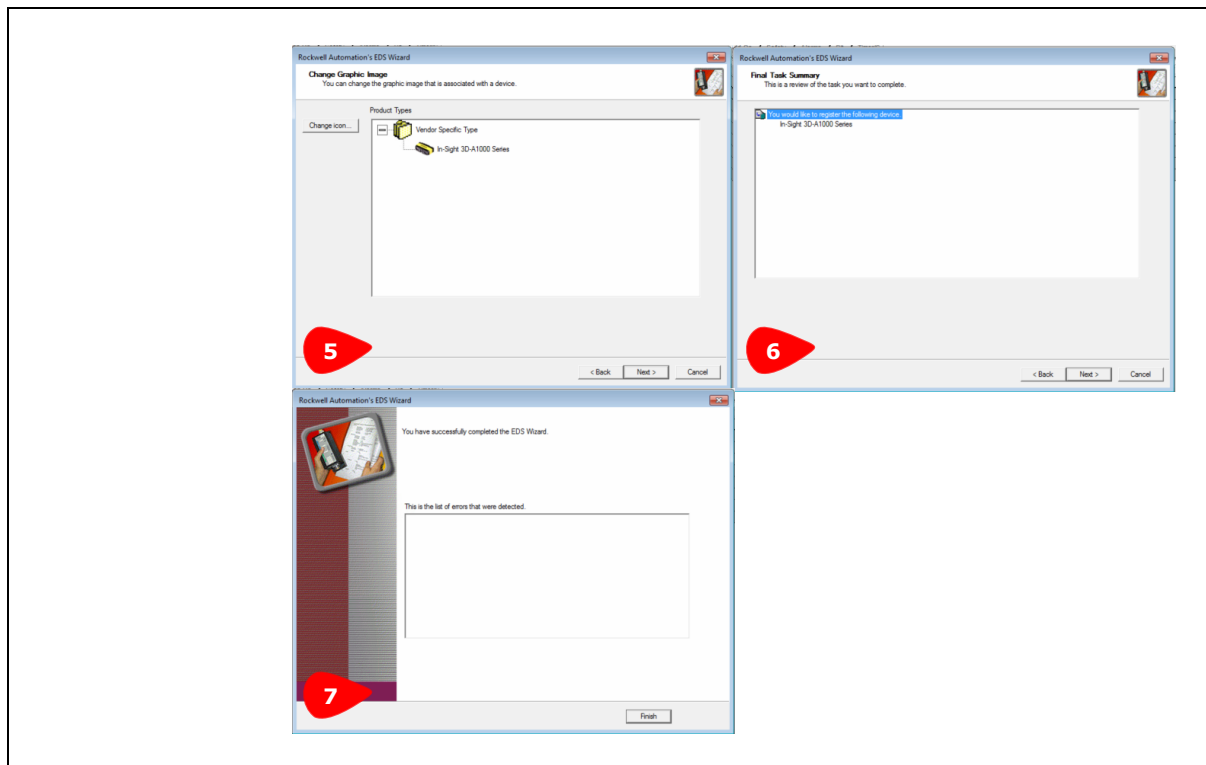
2. Navigate to the **Tools** menu, and select the **EDS Hardware Installation Tool**.



3. Step through the tool's wizard as shown. When prompted, select the EDS file that you unzipped in Step 1.

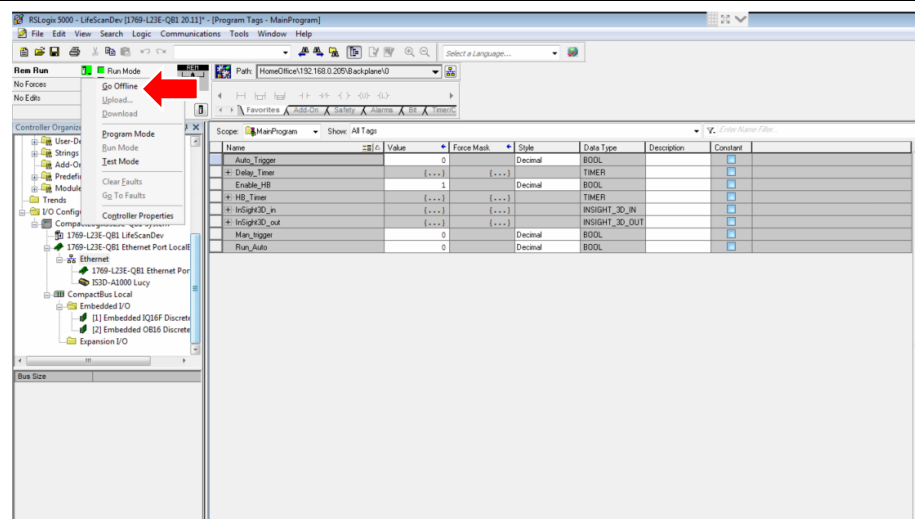


Load/Enable/Test Communications via EtherNet/IP



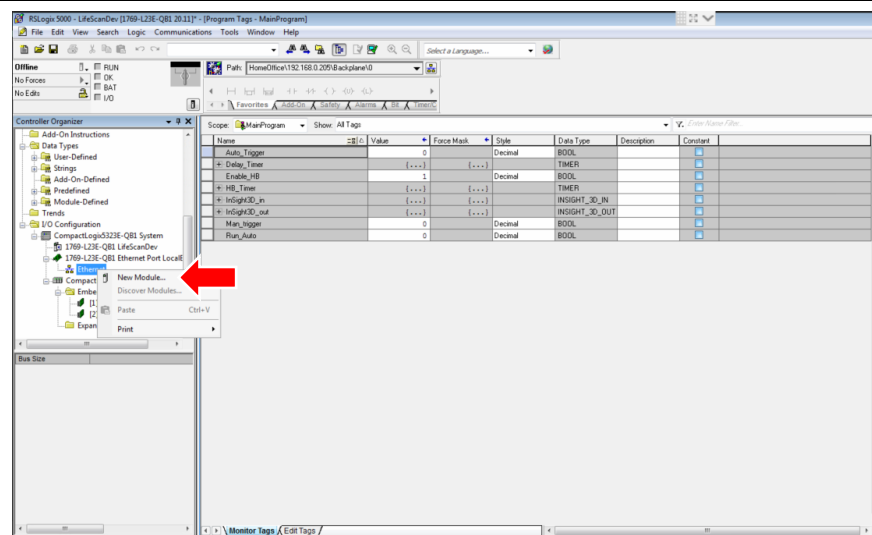
6.4 Adding a Networked 3D-A1000 to Your PLC Program

1. Make sure the PLC is offline.

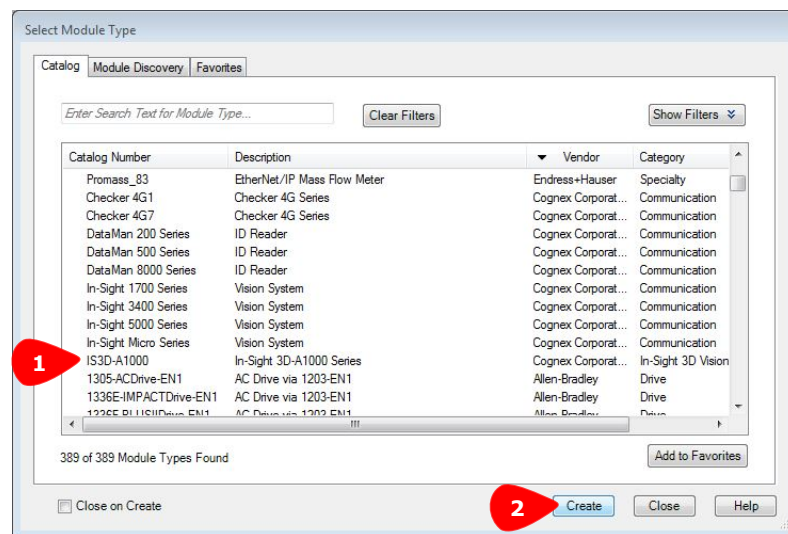


Load/Enable/Test Communications via EtherNet/IP

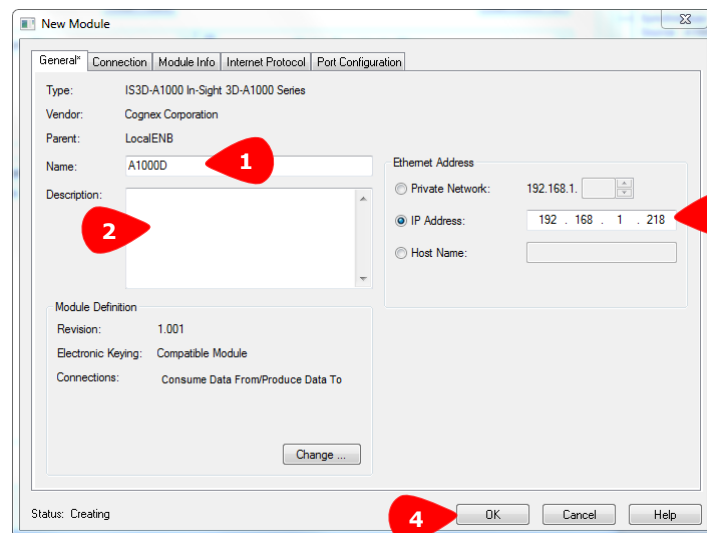
2. In the Controller Organizer under the I/O Configuration sub-section, locate the correct network controller interface, right click and select *New Module...*



3. Under the Catalog tab, locate and select the IS3D-A1000 entry and click *Create*.

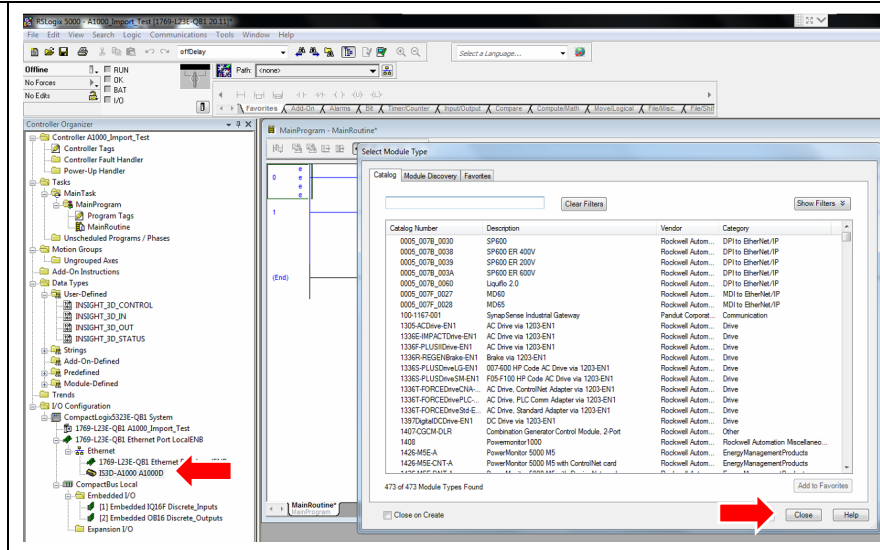


4. In the resulting *New Module* window, fill in the following fields:
 - Name
 - Description (optional)
 - The IP address of the 3D-A1000
 Click OK.



Load/Enable/Test Communications via EtherNet/IP

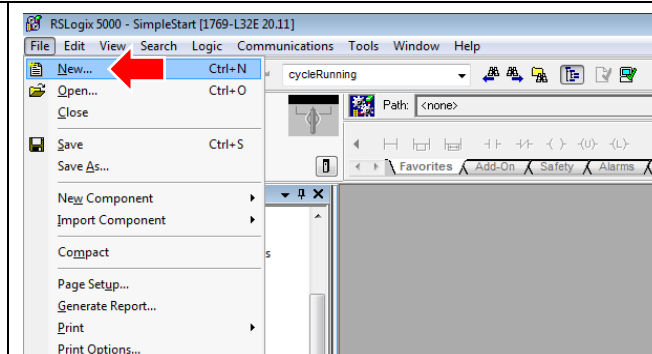
5. You should now see your unit in the network tree. If the *Select Module Type* window is still open, click *Close*.



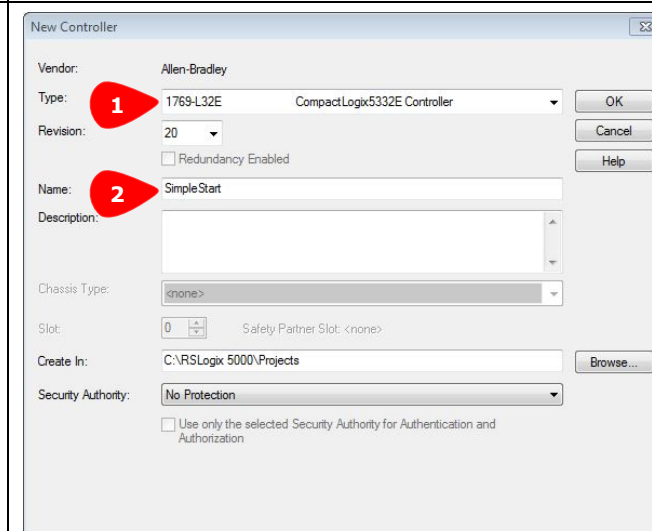
6.5 Creating a 3D-A1000 PLC Program from Scratch

This section describes the necessary steps to create a simple PLC program that triggers the 3D-A1000 device, provides basic handshaking, and receives result data. For a prewritten sample program, please refer to Section 6.6.

1. Start a new project in RSLogix/Studio 5000.



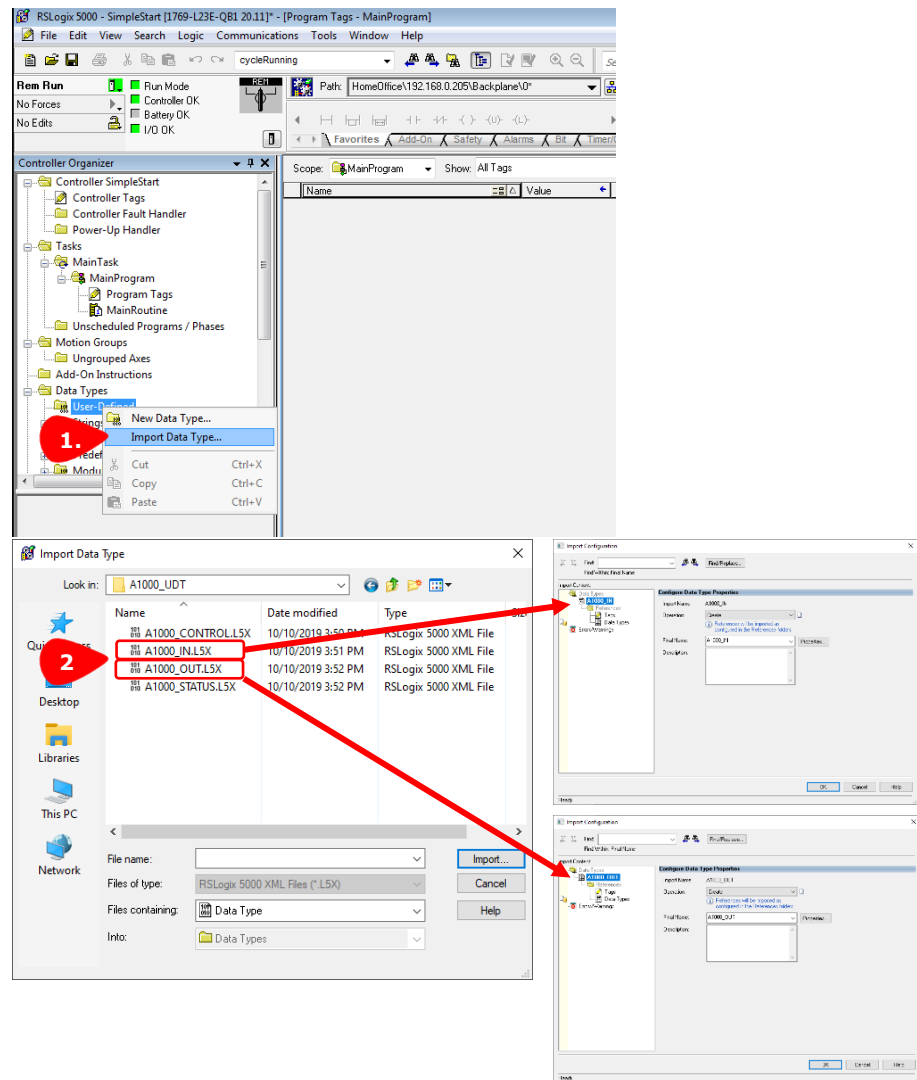
2. Select your controller, name your project, and complete the steps in Section 6.3 and Section 6.4 above.



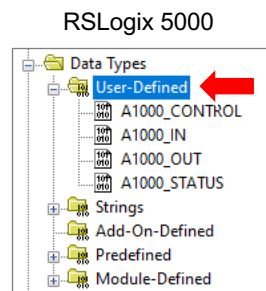
Load/Enable/Test Communications via EtherNet/IP

3. In the *Controller Organizer*, expand the *Data Types* tree, right-click *User-Defined*, and select *Import Data Type...* from the menu. When prompted, select the *A1000_IN.L5X* and *A1000_OUT.L5X* UDT files that you unzipped in Step 1 of *Section 6.3*, as shown here.

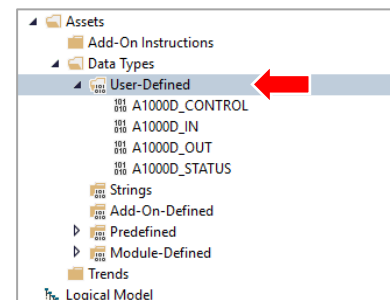
Note that the STATUS UDT is automatically created when importing the IN UDT and that the CONTROL UDT is automatically created when importing the OUT UDT.



4. After the UDT import, your *Data Types* tree should look like this:



Studio 5000



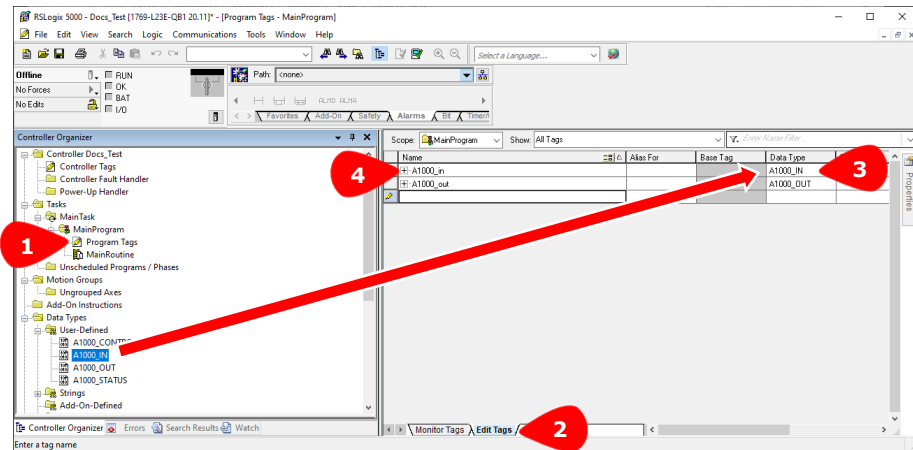
Load/Enable/Test Communications via EtherNet/IP

- Next, add local tags for the A1000 to your *MainProgram*. In the *Tasks* tree, expand *MainTask* → *MainProgram*.

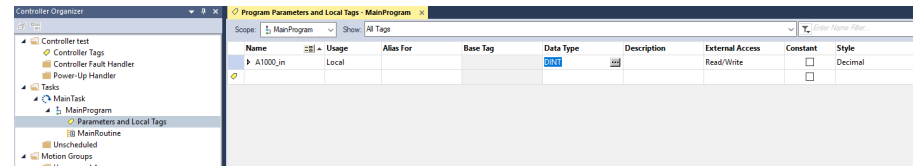
In RSLogix 5000, double-click *Program Tags*. In Studio 5000, double-click on *Parameters and Local Tags*.

Select the *Edit Tags* tab, drag the A1000_IN UDT to *Data Type* to create a new tag, and name the tag, as shown here. Repeat for the A1000_OUT UDT.

RSLogix 5000



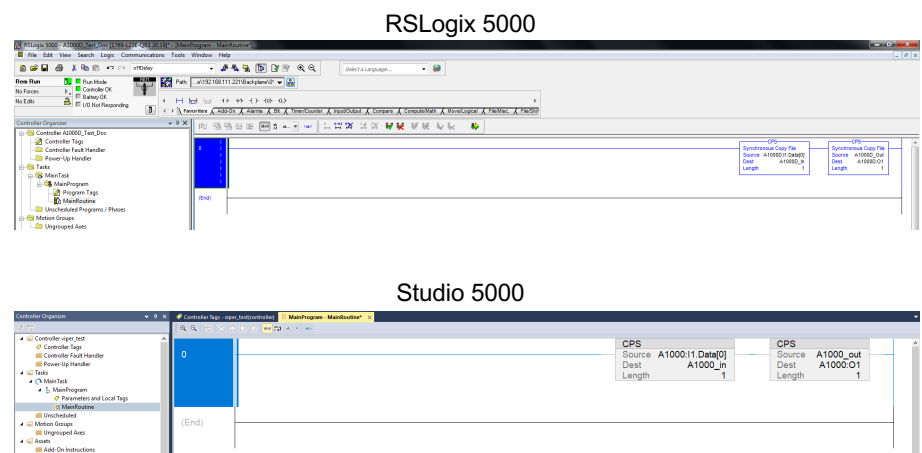
Studio 5000



Load/Enable/Test Communications via EtherNet/IP

6. Open the MainProgram's MainRoutine and add two CPS copy instructions to the first rung.
Note: The CPS instruction can be found under the File/Misc element group.

In the first CPS instruction, copy the A1000:I1 Controller Tag to the input tag from Step 5. In the second CPS instruction, copy the output tag from Step 5 to the A1000:O1 Controller Tag, as shown here.



This forms the basis of the PLC program and enables the use of the custom signals and channels defined within the UDTs. Please refer to *Section 4* and *Section 5* for specifics regarding the definition and meaning of the signals and channels.

6.6 Loading the Sample PLC Program (A1000_EIP_Basic.ACD)

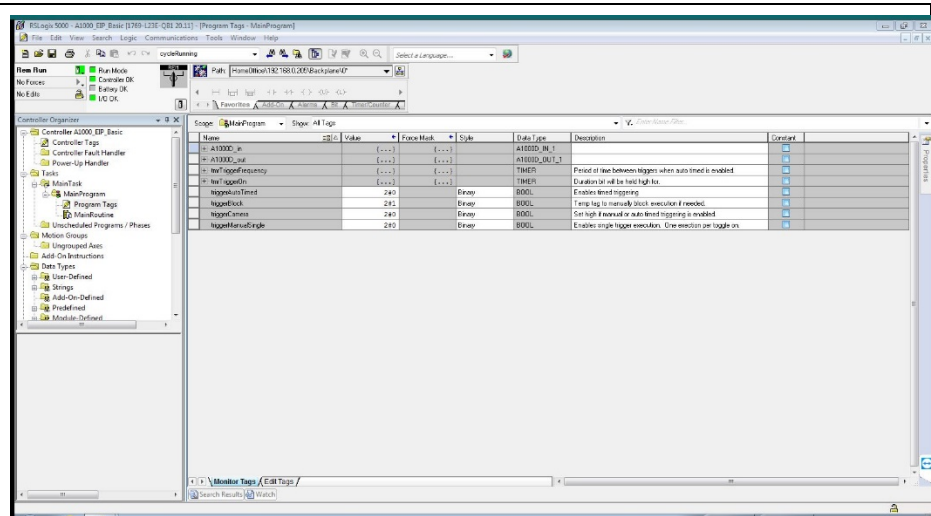
Included with the 3D-A1000 device is a simple six-rung program that will enable you to trigger your device manually via EtherNet/IP or in a timed loop and receive data back from the device. See *Section 6.2* to download the sample PLC program.

The following details each rung's functionality:

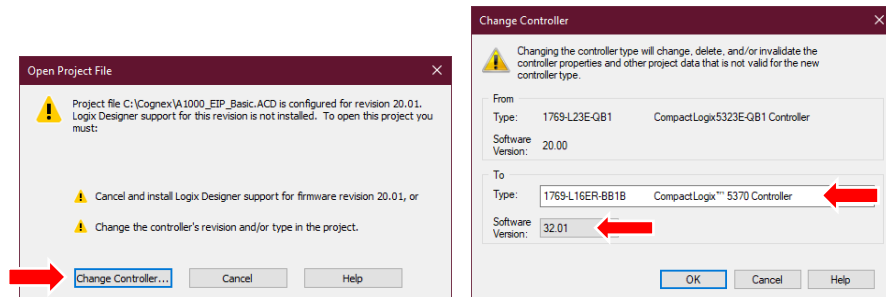
- Rung 0 copies the memory from PLC to Device and Device to PLC.
- Rung 1 ensures the camera is online from the status bits and enables the trigger.
- Rung 2 allows the user to toggle a bit to either manually trigger the device or send a repeating trigger to the camera using a timed loop.
- Rung 3 contains a timer that controls the duration that the trigger bit should be held high.
- Rung 4 contains a timer that controls the trigger rate if a timed loop is selected in Rung 2. The user should set the Preset value to control the frequency.
- Rung 5 executes the trigger.
- Rung 6 copies register values into local L/W/H, Rotational, and Status bit tags.

Load/Enable/Test Communications via EtherNet/IP

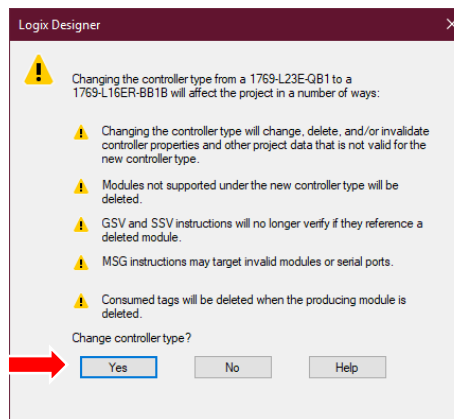
7. Load the program into RSLogix/Studio 5000. With the PLC offline, follow the steps below to set the name and IP address of the 3D-A1000.



Note that for Studio 5000 / Logix Designer, if your installation does not support Revision 20.01, you must change the controller's revision and/or type to open the project.

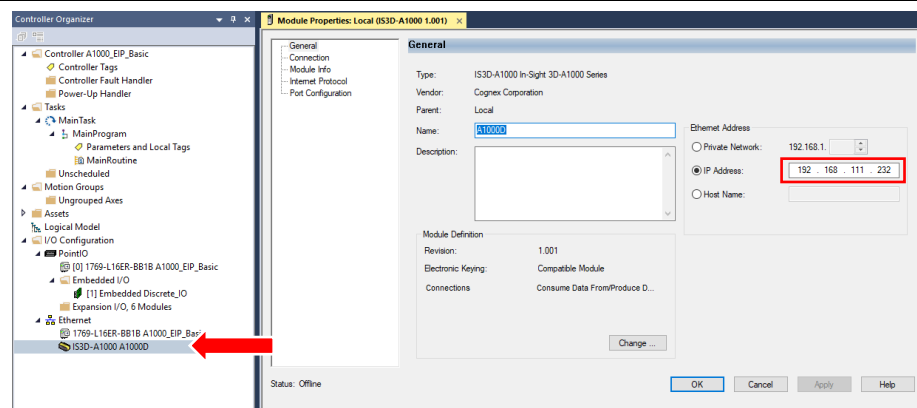


If you choose to change the controller type, you must confirm the change. Click "Yes" to confirm.

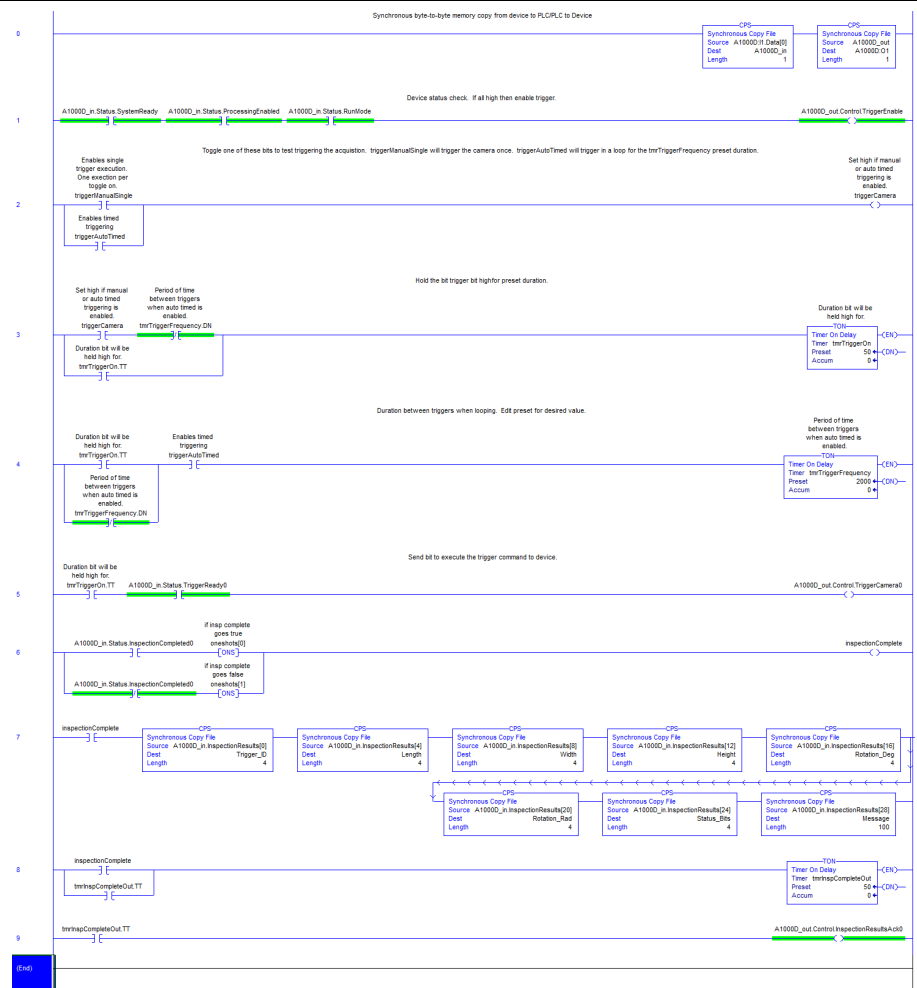


Load/Enable/Test Communications via EtherNet/IP

8. Change the IP address of your A1000 module to match that of your device.

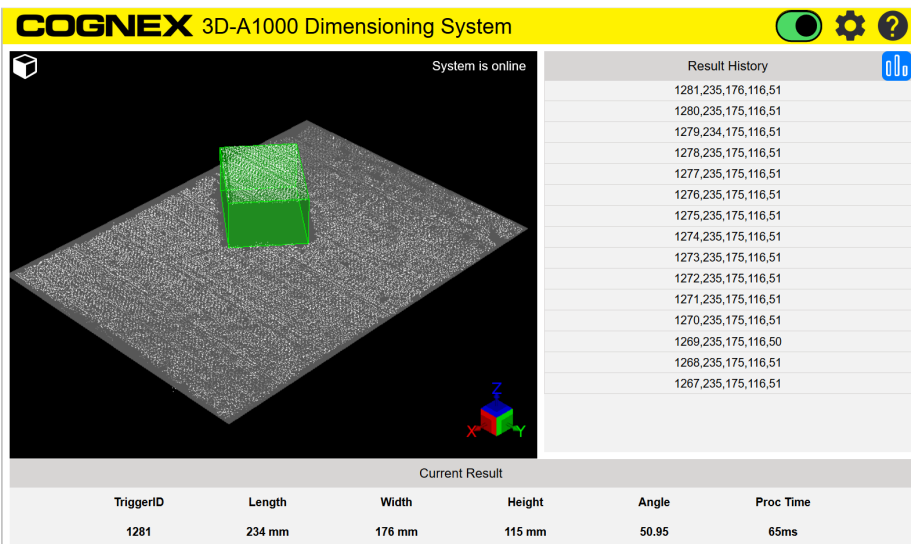


9. Download the program and set the PLC online.



Load/Enable/Test Communications via EtherNet/IP

10. Make sure the 3D-A1000 is online, then send a trigger from the PLC by enabling one of the two trigger bits. Observe the application responding to the trigger.



6.7 Accessing Inspection Results

The 3D-A1000 Dimensioner places the inspection result data into the input assembly starting at byte offset 48 (the *Inspection Results* bytes) as described in *Figure 4-1*. Alternatively, if the user has followed the steps in *Section 6.5* or *Section 6.6* to create the PLC program using a CPS instruction to copy the data into an **A1000_IN** UDT tag, the result data can be accessed via byte 0 of the predefined *InspectionResults* sub-type of the tag (see *Section 5.6* for more information on 3D-A1000 UDTs).

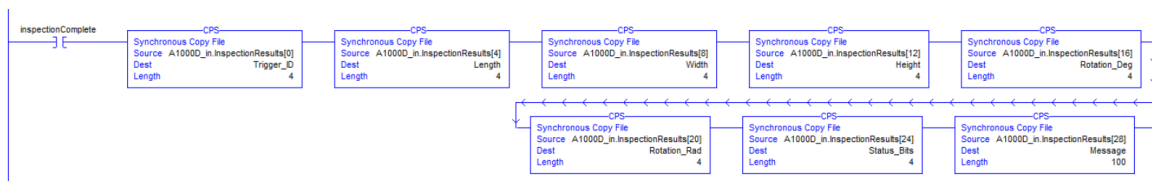
Listed below are the available data items, assuming the UDT has been used.

IMPORTANT NOTE: If the user prefers to directly access this data via the input assembly instead of using the UDT sub-type, an additional byte offset of 48 should be added to the UDT byte offset.

IMPORTANT NOTE: Make sure to set the correct data type (noted below) for any destination tags as failure to do so may cause unexpected results.

- Trigger ID – A1000D_In.InspectionResults[0] (DINT – 4 bytes)
- Length (mm) – A1000D_In.InspectionResults [4] (REAL – 4 bytes)
- Width (mm) – A1000D_In.InspectionResults [8] (REAL – 4 bytes)
- Height (mm) – A1000D_In.InspectionResults [12] (REAL – 4 bytes)
- Rotation (Deg) – A1000D_In.InspectionResults [16] (REAL – 4 bytes)
- Rotation (Rad) – A1000D_In.InspectionResults [20] (REAL – 4 bytes)
- Status Bits – A1000D_In.InspectionResults [24] (DINT – 4 bytes)
 - Bit 0 – Item Found
 - Bit 1 – Out of Spec
 - Bit 2 ... 31 – Future Expansion
- Result string length – A1000D_In.InspectionResults [28] (DINT – 4 bytes)
- Result string – A1000D_In.InspectionResults [32] (STRING – 100 bytes)
 - Note that access to the string data should be accomplished by including the preceding length data, i.e., at offset 28; see example below.

These data assembly registers can be read by adding synchronous copy file (CPS) commands to the PLC program, copying from the desired register(s) listed above to destination tag(s). The following rung shows example CPS instructions:



6.8 Sending Custom Parameters to the 3D-A1000

The 3D-A1000 Dimensioner supports the transfer of user data via the output assembly's *User Data* bytes as described in *Figure 4-2*.

Load/Enable/Test Communications via EtherNet/IP

- If the user has followed the steps in *Section 6.5* or *Section 6.6* to create the PLC program employing a CPS instruction to copy the data from a **A1000_OUT** UDT tag, the user data can be written starting at byte 0 of the predefined *UserData* sub-type of the tag (see *Section 5.6* for more information on 3D-A1000 UDTs).
- If the user prefers to directly write this data via the output assembly instead of using the UDT sub-type, a byte offset of 20 should be used.

Following transfer of the data into the *User Data* bytes, a notification should be sent to the 3D-A1000 that this data is available by raising the *Set User Data 0* bit.

- This bit is available via the *Control* sub-type of the **A1000_OUT** UDT tag via the *SetUserData0* tag.
- If the user prefers to set this bit directly in the output assembly instead of using the UDT sub-type, byte 16 / bit 3 should be used.