ROBOTICS

Application manual

EtherNet/IP Scanner/Adapter

Trace back information:
Workspace 26B version a6
Checked in 2026-06-12
Skribenta version 5.6.019

Application manual

EtherNet/IP Scanner/Adapter

RobotWare 8.1

Document ID: 3HAC097985-001

Revision: B

The information in this manual is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this manual.

Except as may be expressly stated anywhere in this manual, nothing herein shall be construed as any kind of guarantee or warranty by ABB for losses, damage to persons or property, fitness for a specific purpose or the like.

In no event shall ABB be liable for incidental or consequential damages arising from use of this manual and products described herein.

This manual and parts thereof must not be reproduced or copied without ABB's written permission.

Keep for future reference.

Additional copies of this manual may be obtained from ABB.

Original instructions.

Table of contents

Table of contents

Overview of this manual

About this manual

This manual describes the following options and contains instructions on how to configure them in an OmniCore system.

- *3024-1 EtherNet/IP Scanner*
- *3024-2 EtherNet/IP Adapter*
- *3026-2 CIP Safety Adapter*

Usage

This manual should be used during installation and configuration of the options for EtherNet/IP Scanner/Adapter and CIP Safety Adapter.

Who should read this manual?

This manual is intended for

- Personnel responsible for installations and configurations of industrial network hardware/software
- Personnel responsible for I/O system configuration
- System integrators

Prerequisites

The reader should have the required knowledge of

- Mechanical installation work
- Electrical installation work
- System parameters and how to configure them
- RobotStudio

References

Document references

Reference	Document ID
Application manual - I/O Engineering	*3HAC082346-001*
Operating manual - RobotStudio	*3HAC032104-001*
Operating manual - OmniCore	*3HAC098383-001*
Product manual - OmniCore V250XT Type B	*3HAC087112-001*
Product manual - OmniCore V400XT	*3HAC081697-001*
Technical reference manual - System parameters	*3HAC098390-001*
Technical reference manual - RAPID Instructions, Functions and Data types	*3HAC098387-001*
Application manual - DeviceNet Controller/Device	*3HAC098568-001*
Application manual - Controller software OmniCore RobotWare 8	*3HAC098393-001*
Application manual - Functional safety and SafeMove	*3HAC098577-001*

Continues on next page

Continued

Other references

Reference	Description
www.odva.org	The web site of ODVA (Open DeviceNet Vendor Association).
EtherNet/IP™ Specification	ODVA Specification comprises two volumes from the library: *Volume One: Common Industrial Protocol (CIP) Specification* and *Volume Two: EtherNet/IP Adaptation of CIP*.

Revisions

Revision	Description
A	Released with RobotWare 8.0.
B	Released with RobotWare 8.1. • Information about simulated devices added in *Setting up your EtherNet/IP system on page 27* and *Configuring the internal scanner on page 47*. • Compatibility information added in *EtherNet/IP for OmniCore on page 16*. • Minor corrections.

Product documentation

Categories for user documentation from ABB Robotics

The user documentation from ABB Robotics is divided into a number of categories. This listing is based on the type of information in the documents, regardless of whether the products are standard or optional.

> **Tip**
>
> All documents can be found via ABB Robotics One, *https://one.robotics.abb.com/en/knowledge-hub*.

Product manuals

Manipulators, controllers, DressPack, and most other hardware is delivered with a **Product manual** that generally contains:

- Safety information.
- Installation and commissioning (descriptions of mechanical installation or electrical connections).
- Maintenance (descriptions of all required preventive maintenance procedures including intervals and expected life time of parts).
- Repair (descriptions of all recommended repair procedures including spare parts).
- Calibration.
- Troubleshooting.
- Decommissioning.
- Reference information (safety standards, unit conversions, screw joints, lists of tools).
- Spare parts list with corresponding figures (or references to separate spare parts lists).
- References to circuit diagrams.

Technical reference manuals

The technical reference manuals describe reference information for robotics products, for example lubrication, the RAPID language, and system parameters.

Application manuals

Specific applications (for example software or hardware options) are described in **Application manuals**. An application manual can describe one or several applications.

An application manual generally contains information about:

- The purpose of the application (what it does and when it is useful).
- What is included (for example cables, I/O boards, RAPID instructions, system parameters, software).
- How to install included or required hardware.

Continues on next page

Continued

- How to use the application.
- Examples of how to use the application.

Operating manuals

The operating manuals describe hands-on handling of the products. The manuals are aimed at those having first-hand operational contact with the product, that is production cell operators, programmers, and troubleshooters.

Safety

Safety regulations

Before beginning mechanical and/or electrical installations, ensure you are familiar with the safety information in the product manuals for the robot.

The integrator of the robot system is responsible for the safety of the robot system.

Network security

Network security

This product is designed to be connected to and to communicate information and data via a network interface. It is your sole responsibility to provide, and continuously ensure, a secure connection between the product and to your network or any other network (as the case may be).

You shall establish and maintain any appropriate measures (such as, but not limited to, the installation of firewalls, application of authentication measures, encryption of data, installation of anti-virus programs, etc) to protect the product, the network, its system and the interface against any kind of security breaches, unauthorized access, interference, intrusion, leakage and/or theft of data or information. ABB Ltd and its entities are not liable for damage and/or loss related to such security breaches, any unauthorized access, interference, intrusion, leakage and/or theft of data or information.

Terminology

Terms

Term	Explanation
Adapter	I/O device that is controlled by a scanner in an EtherNet network. Previously, ABB documentation used the term *slave*.
CIP	Common Industrial Protocol. Protocol that DeviceNet and EtherNet/IP are based on.
Client	See Scanner. Some documents use the term *client*, whereas the ABB documentation use the term *Scanner* for EtherNet/IP industrial network.
Closed communication systems	Fixed number or fixed maximum number of participants linked by a communication system with well-known and fixed properties, and where the risk of unauthorized access is considered negligible (see IEC 61784-1:2021 §3.1.6).
EDS	Electronic Data Sheet. EDS files contain the configuration details relevant to CIP devices.
Explicit Messages	An explicit message is a request or response oriented communication with other devices. These messages are mostly configuration data.
External adapter	Describes an EtherNet/IP adapter on the EtherNet/IP network connected to the OmniCore controller (not the robot controller acting as EtherNet/IP adapter).
External scanner	Describes an EtherNet/IP scanner on the EtherNet/IP network connected to the OmniCore controller (not the robot controller acting as EtherNet/IP scanner).
Implicit Messages	Implicit messages are exchanged between I/O connections. No messaging protocol is contained within the message data as with Explicit messaging. Implicit messages can be point to point (unicast) or multicast and are used to transmit application specific I/O data.
Internal adapter	Describes when the robot controller acts as an EtherNet/IP adapter on the EtherNet/IP network.
Internal scanner	Describes when the robot controller acts as an EtherNet/IP scanner on the EtherNet/IP network.
LAN	Connector for Local Area Network.
M12	An external Ehternet adapter, distributed by ABB, that is easier to configure in OmniCore than adapters from other suppliers.
Master	See term *Scanner*.
ODVA	Open DeviceNet Vendor Association. Organization for networks built on CIP, for example DeviceNet and EtherNet/IP.
RJ45	Standard Ethernet contact.

Continues on next page

Continued

Term	Explanation
Scanner	Controls other I/O devices (adapters) in an Ethernet network. Previously, ABB documentation used the term *Master*.
Server	See term *Adapter*. Some documents use the term *server*, whereas the ABB documentation use the term *adapter* for EtherNet/IP industrial network.
Slave	See term *Adapter*.
WAN	Port for Wide Area Network.

1 Introduction

1.1 What is EtherNet/IP?

General

EtherNet/IP is a communications link to connect industrial devices.

The EtherNet/IP (EtherNet Industrial Protocol) is managed by ODVA (Open DeviceNet Vendors Association). It is a well established industrial Ethernet communication system with good real-time capabilities. EtherNet/IP extends commercial off-the-shelf Ethernet to the CIP (Common Industrial Protocol)— the same upper-layer protocol and object model found in DeviceNet and ControlNet. CIP allows EtherNet/IP and DeviceNet system integrators and users to apply the same objects and profiles for plug-and-play interoperability among devices from multiple vendors and in multiple sub-nets. Combined, DeviceNet, ControlNet and EtherNet/IP promote transparency from sensors to the enterprise software.

Examples of applications

Here are some examples of EtherNet/IP applications:

- Peer-to-peer data exchange where an EtherNet/IP product can produce and consume messages
- Scanner/adapter operation defined as a proper subset of peer-to-peer
- An EtherNet/IP product can function as a client or server, or both

Standardization

EtherNet/IP is standardized according to the International standard IEC 61158 and EtherNet/IP devices are certified by ODVA for interoperability and conformance.

Data

The following table specifies a number of EtherNet/IP data.

Network type	Ethernet based Control Level network with CIP application protocol
Installation	Standard Off the Shelf (COTS) Ethernet cables and connectors. 10/100/1000 Mbit/s TX Ethernet cable or fibre optics. RJ45, M12 or fibre optic connectors.
Speed	10, 100, 1000 Mbit/s

EDS file

The configuration process is based on EDS files (Electronic Data Sheet) which are required for each EtherNet/IP device. EDS files are provided by the device manufacturers. It contains electronic descriptions of all relevant communication parameters and objects of the EtherNet/IP device.

1.2 EtherNet/IP for OmniCore

General

The EtherNet/IP network is running on the OmniCore main computer and does not require any additional hardware.

Options

With option *3024-1 EtherNet/IP Scanner*, the OmniCore controller can act as a scanner on the EtherNet/IP network.

With option *3024-2 EtherNet/IP Adapter*, the OmniCore controller can act as an adapter.

Note

In this manual, the *3024-2 EtherNet/IP Adapter* is referred to as *internal device*.

Option *3026-2 CIP Safety Adapter* is mandatory for the safe internal adapter device.

Compatibility

The EtherNet/IP adapter is ODVA conformant and certified according to EtherNet/IP specification (CT22-EN).

Specification overview

Item	Specification
Industrial Network type	EtherNet/IP
Conform to	EtherNet/IP protocol conformance test A-9
Data rate	10/100 Mbit
Connection type	Point to point, Multicast
Connection size	Maximum 505 input bytes and 505 output bytes
Transport Class	Class 1 I/O implicit
Trigger mode	Cyclic, Change of state[i]
Application type	Exclusive Owner, Input Only Connection, Listen Only Connection
Priority	Low, High, Scheduled, Urgent
Real time format	32-Bit Run/idle Header[ii], Heartbeat[ii], Modeless[iii]
EPI (Input and Output RPI) for CIP Safety Adapter	10 ms

i Only applicable for Scanner.
ii Only direction Scanner to Adapter.
iii Only direction Adapter to Scanner.

Application protocols

For information about application protocols and port numbers, see section *OmniCore application protocols* in *Operating manual - Integrator's guide OmniCore*.

Continues on next page

Multiport support

EtherNet/IP Adapter/Scanner in the robot controller is a multiport device that supports communication with multiple scanners and adapters on multiple networks simultaneously. No manual port selection is needed, as the device automatically detects which interface it is connected to. The adapter and scanner can operate on the **Public Network** (WAN1) and **I/O Network** (LAN). The **Private Network** (DEV/DEVICE/DEVICE2) can also be used by the scanner but only for Scalable I/O and Safe 24V Switching device.

> **Note**
>
> For information about available network interfaces and their corresponding labels, see the product manual for the respective OmniCore controller.

Limitations

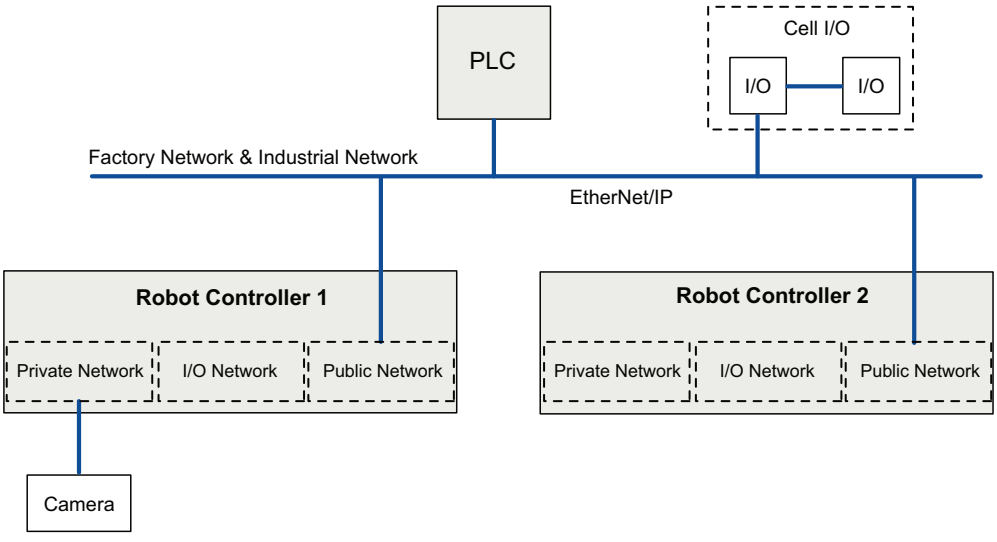
- EtherNet/IP cannot be used on WAN2
- EtherNet/IP Scanner cannot handle devices on different ports if they share the same subnet
- EtherNet/IP Adapter/Scanner can be used on **Public Network** and **I/O Network**
- The **Private Network** supports only the EtherNet/IP Scanner with Scalable I/O and Safe 24V Switching Device

Network connections

EtherNet/IP on public network

When the public network is used for connecting to an industrial network, the traffic shares the same media as the factory network and will share bandwidth with other non industrial network traffic.

The following figure illustrates the network when connecting a scanner and an adapter to the public network:

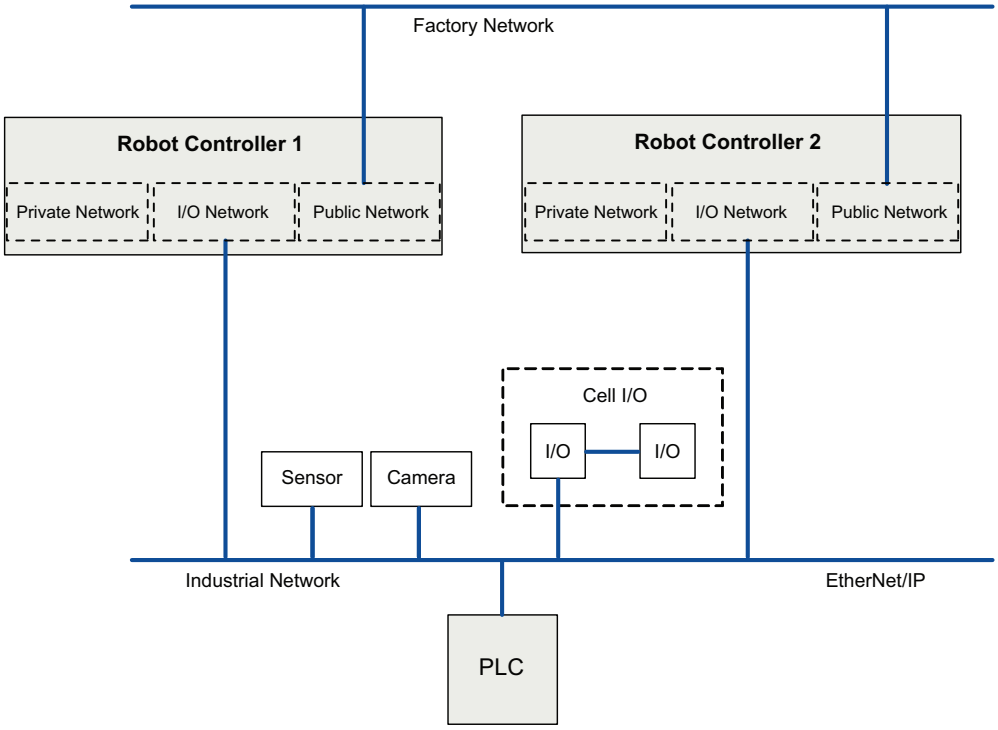


xx2300000128

Continues on next page

EtherNet/IP on I/O network

By connecting to the I/O network it is possible to connect several robot controllers to a dedicated industrial network.



xx2300000131

Note

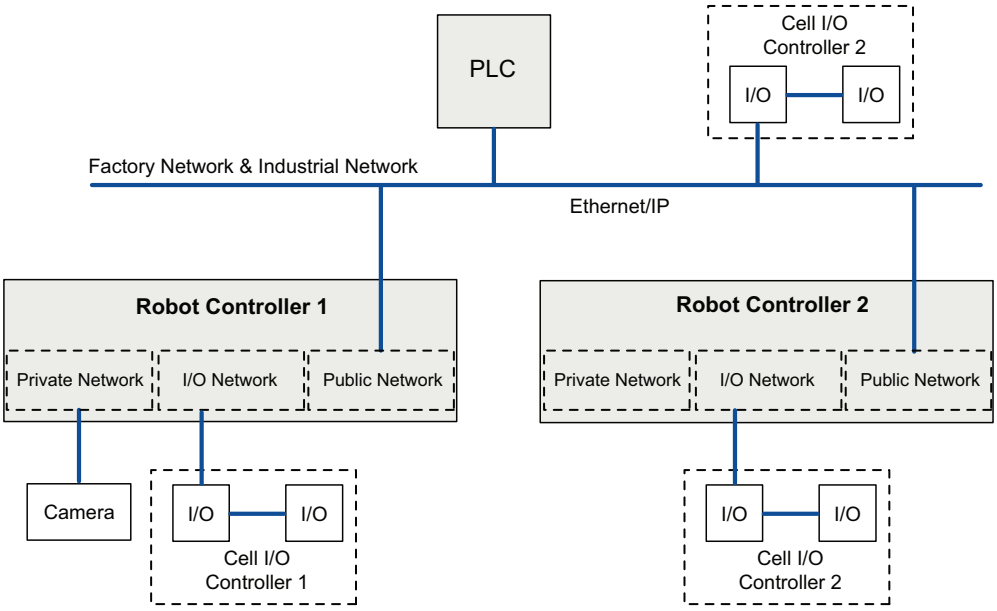
The I/O Network is not available for OmniCore E10.

EtherNet/IP on different networks

The EtherNet/IP scanner and adapter can be on different network interfaces. In this example, the EtherNet/IP scanner on robot controller 2 acts on multiple

Continues on next page

interfaces at the same time (public network and I/O network) and the adapter acts on the public network.



xx2400000604

This page is intentionally left blank

2 Hardware overview

2.1 Ethernet switches

Prerequisites

It is recommended that switches used in the I/O network support Quality of Service (QoS).

I/O devices mark their packets with a priority value. The priority value is used in order to get better I/O data throughput and shorter delays on the network.

Switches and routers are then able to differentiate the I/O device's critical from the other non-critical traffic. To do this, the switches and routers must support Quality of Service.

2.2 I/O devices

Limitations

It is possible to connect any type of EtherNet/IP compliant I/O device on the EtherNet/IP network. All I/O devices should comply with the EtherNet/IP standard and be conformance tested by ODVA. I/O devices may be mounted inside the OmniCore controller.

3 Software overview

3.1 Information about the adapter device

General

To use the EtherNet/IP adapter device, the OmniCore controller must be installed with the option *3024-2 EtherNet/IP Adapter*.

The EtherNet/IP adapter device can be used to:

- connect a PLC to the OmniCore controller.
- connect the OmniCore controller to another OmniCore controller which acts as a scanner.

Industrial Network

When the robot system is installed with the *3024-2 EtherNet/IP Adapter* option, a predefined industrial network with the name *EtherNetIP* is created at system startup.

Predefined internal adapter device

When the robot system is installed with the *3024-2 EtherNet/IP Adapter* option, a predefined internal adapter device with the name *EN_Internal_Device* is created at system startup. This internal device is used to define the internal adapter device in the OmniCore controller, which will enable a PLC to connect to the OmniCore controller. There can be only one internal adapter device defined in the OmniCore controller.

I/O device

The input and output map starts at bit 0.

EDS file

An Electronic Data Sheet file, EDS file, is available for the internal adapter device, matching the configuration of the predefined *EtherNet/IP Internal Adapter Device EN_Internal_Device*.

The EDS files can be obtained from RobotStudio or the OmniCore controller.

- **In the RobotWare installation folder in RobotStudio**:
  ...\RobotPackages\RobotControl_x.x.xxx\utility\service\EDS
- **On the OmniCore Controller**:
  ...\RobotWare\RobotControl_x.x.x-xxx\utility\service\EDS

In order to select the correct EDS file, see *Selecting the correct EDS file for OmniCore internal adapter on page 35*.

Note

Navigate to the RobotWare installation folder from the RobotStudio **Add-Ins** tab, by right-clicking on the installed RobotWare version in the **Add-Ins** browser and selecting **Open Package Folder**.

Continues on next page

Assembly

The internal adapter device has the following *Assembly* values.

Assembly	Value
Output	112
Input	100
Configuration	199

Behavior

Cyclic I/O connection is supported and the size of the I/O connection is defined by the predefined EtherNet/IP Internal Adapter Device, EN_Internal_Device.

Note

If the EtherNet/IP adapter device loses connection with its scanner, the configured input signals are cleared (reset to zero). The output signals are kept and are possible to change.

When the connection is re-established, the EtherNet/IP adapter device updates the input and output signals.

Note

Communication with configured EtherNet/IP Adapter devices is still possible with firewall setting set to ON. Communication initiated by the controller is always allowed.

3.2 Information about the internal scanner

General

To use the EtherNet/IP internal scanner, the OmniCore controller must be installed with the option *3024-1 EtherNet/IP Scanner*.

The EtherNet/IP internal scanner can be used to:

- connect EtherNet/IP I/O devices to the OmniCore controller.
- connect the OmniCore controller to another OmniCore controller which acts as an adapter.

Industrial Network

When the robot system is installed with the EtherNet/IP Scanner/Adapter option, a predefined industrial network *EtherNetIP* is created at system startup.

Device Templates

There are predefined device templates available for the internal scanner. Examples of present device templates are:

- *ABB OmniCore EtherNet/IP Adapter Device* is used on the scanner side when connecting to another OmniCore EtherNet/IP adapter.
- ABB Scalable I/O Device: 16DO/16DI (base version)
- ABB Scalable I/O Device + Digital: 16DO/16DI + 16DO/16DI
- ABB Scalable I/O Device + Analog: 16DO/16DI + 4DO/4DI
- ABB Scalable I/O Device + Relay: 16DO/16DI + 8DO/8DI

Number of allowed I/O devices

A maximum number of 50 user defined I/O devices can be defined in the OmniCore system, for more information see *Device Type* of *I/O System* section in*Technical reference manual - System parameters*.

The following are counted as user defined I/O devices:

- All EtherNet/IP adapter devices connected to the OmniCore EtherNet/IP scanner.
- Simulated EtherNet/IP I/O devices.

> **Note**
>
> The internal adapter device is not counted as a user defined I/O device.

It is possible to use ABB I/O devices or I/O devices from other vendors. Only the EtherNet/IP Scanner/Adapter option is required to run I/O devices from other vendors.

The values of input and output *Assembly* are used by the EtherNet/IP scanner to locate the input and output data in the I/O device. The *Assembly* values for different I/O devices are available in the EDS file and in the User Manual. We recommend you to refer the User Manual for the *Assembly* values of the I/O device.

Continues on next page

Connecting two OmniCore systems

When connecting two OmniCore systems, the internal adapter should be seen and configured as any other ordinary device from the other OmniCore system, which is acting as a scanner. See *Communication between two OmniCore controllers on page 59*.

4 Setting up your EtherNet/IP system

Log in with configuration grant

For configuration in I/O Engineering, the user grant **Modify configuration** is required. See *Operating manual - RobotStudio*, section *Managing user rights and write access on a controller*.

Start I/O Engineering

1 Start RobotStudio and connect to the robot system to configure.

2 In the ribbon of the **Controller** tab, select **I/O Engineering**.

See *Application manual - I/O Engineering* for more information.

Configure IP settings

IP settings for the EtherNet/IP network used on the Public or I/O Network are defined in **Network Settings** in RobotStudio or on the FlexPendant. See *Operating manual - RobotStudio* and *Operating manual - Integrator's guide OmniCore*.

The IP address must be assigned statically since automatic IP address assignment (DHCP) is not supported for Ethernet/IP.

Note

The Private Network uses a fixed IP address of 192.168.125.1 and cannot be configured.

Configure firewall settings

The Firewall Management function is used to configure the network firewall on the controller. Configuration is done in RobotStudio under **Configuration\Communication\Firewall Manager** where pre-configured Network Services can be enabled or disabled. See *Operating manual - RobotStudio* and *Operating manual - Integrator's guide OmniCore*.

Configure the EtherNet/IP network properties

1 In the **Controller** tab in RobotStudio, select **I/O Engineering**. The **I/O Engineering** tab is displayed.

2 In the **Configuration** browser, select **EtherNetIP** (under I/O system).

3 In the **Properties** browser, you can configure the following:

Parameter	Description	Allowed values
Simulated	Select **Yes** or **No**, indicating if the industrial network and all its connected I/O devices should be treated as simulated. **Note** If **Simulated** is set to **Yes**, network configuration is not required.	The default value is No.

Continues on next page

Continued

Parameter	Description	Allowed values
Identification Label	This parameter is an optional way to provide a label that will help the operator to identify the internal device.	A string with maximum 80 characters.

Configure unique identifiers for CIP Safety

This chapter describes the unique identifiers for the CIP Safety device (Adapter). It is essential with proper configuration to uniquely identify a device on the CIP Safety network to maintain high integrity and safety standards.

Unique Identifiers

Each CIP Safety Adapter on the network must be assigned the following identifiers:

- Node ID (IP address of the device)
- Safety Network Number
-

CAUTION

The user should assign **SNN** numbers for each safety network or safety subnet that are unique system-wide.

Note

If a safety project is copied to another project within the same CIP Safety system, all safety network numbers (SNN) in the new project must be changed. SNN values must not be reused.

These identifiers are critical for distinguishing devices within the safety network and ensuring secure, reliable communication.

Multiport Device Support

The CIP Safety device (Adapter) in robot controller supports multiport communication, allowing it to communicate with multiple safety controllers simultaneously, while maintaining the integrity and independence of each safety connection.

To ensure proper operation, the unique identifiers must be configured for each available network interface on the robot controller:

- Public network
- I/O network
- Private network

Note

Refer to the specific robot controller's manual to identify the available network interfaces and their corresponding labels.

Continues on next page

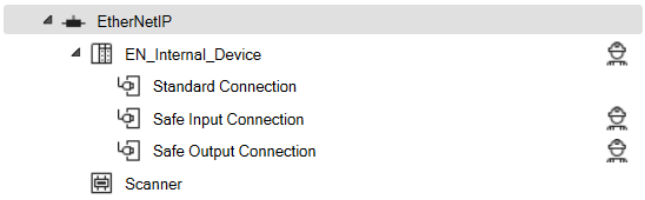
Note

Private network is reserved for Safe Scalable I/O and Safe 24V Switching Device, and shall not be used by the CIP Safety Adapter.

Configuration is done at the network entry level, EtherNet/IP.

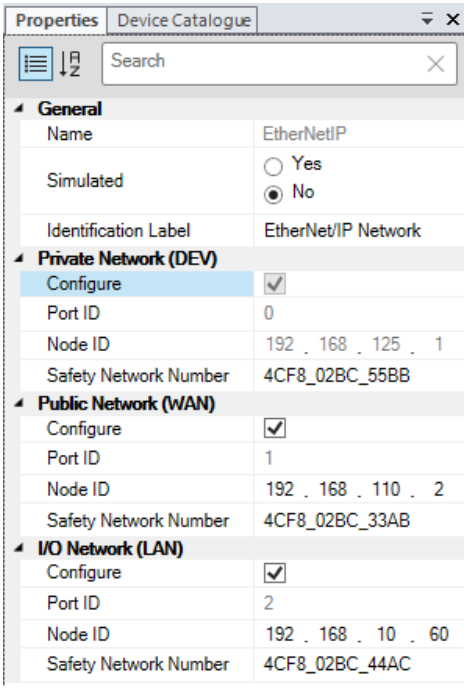
1 In the **Controller** tab in RobotStudio, select **I/O Engineering**.

2 In the **Configuration** browser, select **EtherNetIP**.



xx2400001589

3 Configure the parameters for each network interface in the **Properties** window:



xx2400001621

Parameter	Description
Configure	Select the checkbox to configure the network for CIP Safety. Clear the checkbox to remove CIP Safety from the configuration.

Continues on next page

Continued

Parameter	Description
Node ID	The same as the IP address of the safety device. **Note** The IP address for the Private Network is default. This network interface is reserved for Safe Scalable I/O and Safe 24V Switching Device, and shall not be used by the CIP Safety Adapter. **Note** WAN and LAN are network interfaces that can be configured for the CIP Safety Adapter.
Safety Network Number	The *Safety Network Number (SNN)* provides a unique network identifier for each network in the safety system. Typically, it is generated automatically by the configuration tool or selected arbitrary by the user. This value is a 6 byte long, written as a hexadecimal string. The values 0000_0000_0000 and FFFF_FFFF_FFFF are illegal. To generate **SNN** in Studio 5000, see *Configuring the ABB CIP Safety Adapter in Studio 5000® on page 71* **CAUTION** The user should assign **SNN** numbers for each safety network or safety subnet that are unique system-wide. **Note** If a safety project is copied to another project within the same CIP Safety system, all safety network numbers (SNN) in the new project must be changed. SNN values must not be reused.

Note

The **Port ID** identifies the port in the safety configuration. This ID is referenced in the safety report.

Importing EDS files

An EDS file contains data about a device. It is necessary to add EDS files for all devices that should be added. It is also possible to add EDS files for any devices that may be added later.

1. In the **I/O Engineering** tab, select **Import Device Descriptions** > **EDS Files**, or right-click on **EtherNetIP** (under I/O system) and select **Manage EDS files**.
2. The **EDS Files** window is displayed. Click **Import** and browse for an EDS file, or **Import, Folder** to import a complete folder with EDS files.

Continues on next page

Continued

Note

The **Used EDS files...** window shows all EDS files that are used in the current I/O project.

The **Imported EDS files...** window shows all EDS files that are imported into the I/O project but are not used in the configuration.

Note

When a new I/O project is opened, only the used EDS files will be shown in the **Imported EDS files...** window. Click **Update** to display all previously imported, but not used, EDS files.

This page is intentionally left blank

5 Configuring the internal adapter

Introduction

This chapter describes how to configure the internal adapter in RobotStudio.

To establish a connection with a PLC (scanner), the adapter must also be configured within the PLC program. See the scanner manufacturer's manual for instructions on how to configure the OmniCore adapter using the EDS file. To select the correct EDS file for the Omnicore adapter, see *Selecting the correct EDS file for OmniCore internal adapter on page 35*.

Continues on next page

5.1 Configuration prerequisites

Prerequisites

Before configuring the internal adapter device, make sure to set up your system according to *Setting up your EtherNet/IP system on page 27*.

The option *3024-2 EtherNet/IP Adapter* is needed for EtherNet/IP adapter device configuration.

5.2 Selecting the correct EDS file for OmniCore internal adapter

EDS file selection

OmniCore products support different communication options depending on the configuration. To ensure proper integration with your PLC, it is important to select the correct EDS file and to understand when to enable **Compatibility Mode**.

OmniCore products support two types of internal adapters:

- EtherNet/IP Adapter (3024-2)
- CIP Safety Adapter (3026-2)

Note

The actual EDS file name depends on the controller configuration selected during installation. See *EDS file reference by controller configuration on page 35* to find the correct file for your setup.

Option installed	EDS file to be selected
EtherNet/IP Adapter (3024-2)	OmnicoreXXX.eds
CIP Safety Adapter (3026-2)	OmnicoreXXXSafe.eds
EtherNet/IP Adapter (3024-2) and CIP Safety Adapter (3026-2)	If both options are installed, the CIP Safety option takes precedence. In this case, the product will behave as a CIP Safety device in identity and electronic keying operations. In this case, **Compatibility Mode** must be enabled to use the standard file OmnicoreXXX.eds.
ABB Safety I/O modules (Safe Scalable I/O and Safe 24V Switching Device)	If ABB Safety I/O modules (Safe Scalable I/O and Safe 24V Switching Device) are installed, the CIP Safety option takes precedence. In this case, the product will behave as a CIP Safety device in identity and electronic keying operations. In this case, **Compatibility Mode** must be enabled to use the standard file OmnicoreXXX.eds.

Compatibility mode

This allows the PLC or scanner to communicate using the standard OmnicoreXXX.eds file, even though safety features are present.

To enable **Compatibility Mode**:

- Set the **Compatibility Bit** in the **Electronic Keying** settings of your PLC or scanner.
- Refer to your PLC vendor's documentation for detailed instructions.

EDS file reference by controller configuration

OmniCore controller	EtherNet/IP Adapter (3024-2)	CIP Safety Adapter (3026-2)
E10 Type A	OmniCore_E10A.eds	OmniCoreSafe_E10A.eds
C30 Type B (with option *3091-1 Extension switch*)	OmniCore_C30B.eds	OmniCoreSafe_C30B.eds

Continues on next page

OmniCore controller	EtherNet/IP Adapter (3024-2)	CIP Safety Adapter (3026-2)
C30 Type B	OmniCore_C30B_Basic.eds	OmniCoreSafe_C30B_Basic.eds
C90XT Type A	OmniCore.eds	OmniCoreSafe.eds
V250XT Type B		
V400XT		

Note

Currently, Studio 5000 Logix Designer does not support multiport devices, which means the OmniCore EDS file cannot be parsed using this tool. To configure the Ethernet/IP adapter for OmniCore controller E10 Type A, use the EDS file named OmniCore_E10A_Studio5000.eds.

When configuring the CIP Safety adapter, use generic profiles present in Studio5000, see *Configuring the ABB CIP Safety Adapter in Studio 5000® on page 71*.

5.3 Configure the internal adapter properties

Configure the internal adapter properties

1 In the **Configuration** browser, expand **EtherNetIP** and select the internal adapter.

2 In the **Properties** browser, you can configure the following:

Parameter	Description	Allowed values
Name	Enter the name to be used for the device.	
Identification Label	This parameter is an optional way to provide a label that will help the operator to identify the device.	A string with maximum 80 characters.
Connection Input Size	Enter the input size.	0 - 505
Connection Output Size	Enter the output size.	0 - 505

3 Select **I/O Project**/**Save Project** to save the changes.

4 Select **Read and Write** to write the configuration to the controller. See *Application manual - I/O Engineering* for more information.

5 Restart the controller.

5.4 Configuring the safe internal adapter device (CIP Safety)

Introduction

When using CIP Safety Adapter, all I/O communication for the SafeMove functionality is done via a safety PLC using CIP Safety (Safe EtherNet/IP).

The safety PLC acts as a CIP Safety controller (Scanner) and the robot controller acts as a CIP Safety device (Adapter). See *How to establish CIP Safety communication with a PLC on page 69*.

Prerequisites

Required options are *3024-2 EtherNet/IP Adapter* and *3026-2 CIP Safety Adapter*.

If working with (CIP Safety), log in as a safety user (the user grant **Safety Services** is required). See *Operating manual - RobotStudio*, section *Managing user rights and write access on a controller*.

CAUTION

Before installing a new device into the safety network, the user must ensure that any pre-existing configuration is cleared from the new device.

Note

The safety function integrator shall carefully consider implications of mixing different SIL/PL level devices on the network. The device with the lowest SIL/PL rating determines the level for the closed communication system[1] and all safety functions using it.

Limitations

- The CIP Safety internal adapter does not support multicast connections.
- When configuring *Requested Packet Interval (RPI)* on the PLC, the value must be larger or equal to 10 milliseconds.
- The size of safety data assemblies is fixed to 8 bytes and not configurable.
- The CIP Safety internal adapter only supports *Type 2 SafetyOpen* from external PLCs, meaning it can only be configured from I/O Engineering.

Note

The CIP Safety internal adapter supports PLd and shall only be used in closed communication systems.

Reset ownership

The first scanner (originator) that successfully establishes a producing connection to the ABB CIP safety adapter becomes the owner of that adapter's inputs. It is necessary to reset the ownership when the originator's SNN or NodeID is changed.

1 Fixed number or fixed maximum number of participants linked by a communication system with well-known and fixed properties, and where the risk of unauthorized access is considered negligible (see IEC 61784-1:2021 §3.1.6).

Continues on next page

The ownership is established to prevent errant or unauthorized connections from hijacking an input resource in a validated safety system. Only one owner is allowed.

It is possible to reset the ownership of the internal CIP safety adapter by executing the **Reset CIP Safety** function.

On the FlexPendant start screen, select **Settings** > **Safety Controller**. In the **Configuration** tab, select **Reset CIP Safety**. The CIP Safety configuration will be restored except from the ownership.

> **Note**
>
> It is the responsibility of the user to guarantee the safety of the system after resetting CIP Safety. The user needs to guarantee that the desired ownership is established after the reset.

Resetting the safety controller

Use this procedure to reset the safety controller to factory settings from the FlexPendant:

1. On the start screen, tap **Settings**, and then select **Backup & Recovery** from the menu.
2. On the sidebar tap **Reset user data**.
3. Select the check box **Reset safety settings**.
4. Tap **Reset**.
5. Synchronize the safety controller with the robot controller, see *Application manual - Functional safety and SafeMove*.
6. A new safety configuration can now be loaded and validated, see *Application manual - Functional safety and SafeMove*.

> **Note**
>
> It is also possible to reset the safety controller to factory settings from RobotStudio, see *Application manual - Functional safety and SafeMove*.

Configure the CIP safety adapter

This procedure describes how to configure the CIP safety adapter. The procedure consists of three main topics:

- Configure the unique identification of the CIP Safety Adapter on the network, see *Configure unique identifiers for CIP Safety on page 28*.
- Configure I/O settings, see *Configure I/O settings on page 40*.
- Configure I/O signals, see *Configure I/O signals on page 40*.

> **Note**
>
> The CIP safety adapter is added automatically for systems with the option *3026-2 CIP Safety Adapter*.

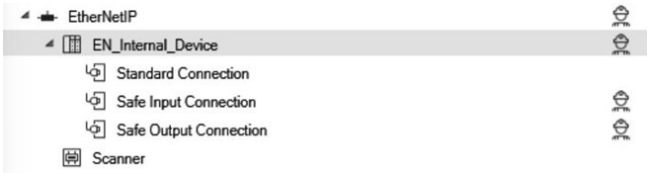
Continues on next page

5.4 Configuring the safe internal adapter device (CIP Safety)
Continued

Configure I/O settings

This is done at the device entry level, **EN_Internal_Device**.

1. In the **Controller** tab in RobotStudio, select **I/O Engineering**.
2. The **Configuration** browser displays the CIP safety adapter (EN_Internal_Device), and three connections (without signals):



xx2500000975

3. Select the **Standard Connection** for **EN_Internal_Device** and configure the parameters in the **Properties** window.

   You can, for example, change the **Input/Output size**.

Configure I/O signals

1. In the **Configuration** browser, select a connection (Standard/Safe Input/Safe Output) and add signals in the **Signal Editor**.

   For detailed information about how to work with signals, see *Application manual - I/O Engineering*.
2. Select **I/O Project/Save Project** to save the changes.
3. Select **Read and Write** to write the configuration to the controller. See *Application manual - I/O Engineering* for more information.
4. Restart the controller.

Copy SCID to Studio 5000®

The configuration signature, also called *Safety Configuration ID (SCID)*, uniquely identifies the configuration of the *ABB CIP Safety Adapter* and can be used to confirm the integrity of the adapter configuration over time.

The signature is checked whenever an originator tries to connect to the adapter. If the signatures match, the connection is established. If the signature does not match, the error response *Configuration signature mismatch* is returned.

The signature is printed in the safety report and must be copied to any external safety scanner (PLC) that wants to connect to the robot controller.

CAUTION

The **SCID** in the safety report shall be compared with the transferred SCID read from the Live view in I/O Engineering and the SCID entered in Studio 5000®.

CAUTION

When configuring safety connections without the configuration signature, the user is responsible for ensuring that originators (scanners) and targets (adapters) have the correct configurations.

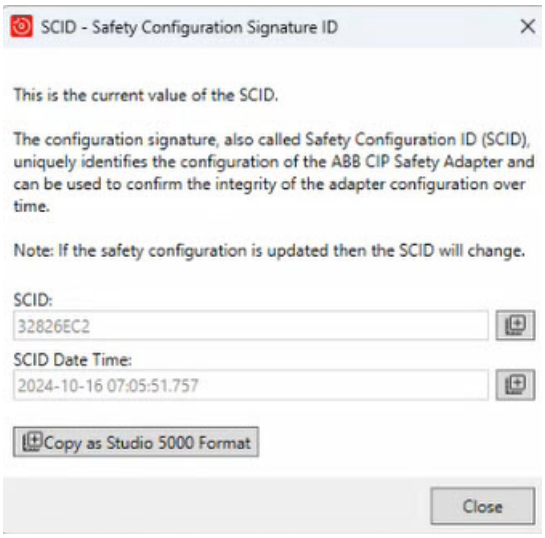
Continues on next page

CAUTION

The configuration signature should only be considered verified after user testing. User testing is the means by which all downloads are validated.

Use this procedure to copy the configuration signature (**SCID**) to Studio 5000®:

1. Open the **Live** view in I/O Engineering.
2. Right-click the internal device, select **Show SCID**.
3. In the **SCID - Safety Configuration Signature ID** window, select **Copy as Studio 5000 Format** to copy the **SCID** to the clipboard.



xx2500000713

4. Paste the **SCID** in Studio 5000®.

   For more information, see *Configuring the ABB CIP Safety Adapter in Studio 5000® on page 71*.

Validate the safe fieldbus and signal configuration

Validate the safe fieldbus parameters, including I/O settings and signals used for safety interlocking, by comparing the **ABB Safety Configuration Report** with the configured values in I/O Engineering.

The user must visually verify that the data in the safety report is correct and that it is the same as entered in I/O Engineering. The SCID in the safety report shall also be compared with the transferred SCID read from the **Live** view in I/O Engineering and the SCID entered in Studio 5000®.

After controller restart the configuration is applied. The user must verify that no safe fieldbus related event logs were generated, that the status of the connection in the originator indicates "running", and that the connection to the intended adapter device has been established.

Continues on next page

CAUTION

The user must visually verify that the data in the safety report is correct and that it is the same as entered in I/O Engineering.

WARNING

All downloaded configurations must be validated by user testing before the installation can be regarded as safe. User testing is the means by which all downloads are validated.

Use this procedure to verify the configuration parameters:

1. In I/O Engineering, write the configuration to the controller.
2. In the **Controller** tab in RobotStudio, select **Safety > Visual SafeMove**.
3. In the **Visual SafeMove** tab, select **Show Report > Controller configuration**.
4. The **ABB Safety Configuration Report** is displayed.
5. Verify that the following configuration parameters matches the safety report:
   - General Information:
     - **Configuration Signature - ID**
     - **Configuration Signature - Date**
     - **Configuration Signature - Time**
   - CIP Safety:
     - Electronic Key:

       Note

       Electronic key values shall be compared with values specified in *Module parameters on page 70*.

       # **Vendor ID**
       # **Device Type**
       # **Product Code**
       # **Major Revision**
       # **Minor Revision**
     - Port:
       # **Id**
       # **Node Id**
       # **Safety Network Number**
     - Devices:
       # **Device Name**
       # **Signals**
6. If the parameters in the report are correct, the status of the configuration can be set to validated or locked. For more information about configuration status, see *Application manual - Functional safety and SafeMove*.

Continues on next page

Note

Do not set the status of the configuration to validated or locked until the validation is performed. For more information about configuration status, see *Application manual - Functional safety and SafeMove*.

5.5 Working with signals

Overview

I/O signals can be added to I/O devices in a project.

For detailed descriptions of all signal parameters, see *Technical reference manual - System parameters*.

For more information about working with signals, see *Application manual - I/O Engineering*.

5.6 Saving the configuration

Save configuration and write to controller

1 Select **I/O Project**/**Save Project** to save the changes.

2 Select **Read and Write** to write the configuration to the controller.

See *Application manual - I/O Engineering* for more information.

This page is intentionally left blank

6 Configuring the internal scanner

6.1 Configuration prerequisites

Prerequisites

Before configuring the internal scanner, make sure to set up your system according to *Setting up your EtherNet/IP system on page 27*.

The option *3024-1 EtherNet/IP Scanner* is needed for EtherNet/IP internal scanner.

Set the IP address on the external device. See user manual from the vendor of the external device.

6.2 Configuration guidelines

Guideline to select Production Trigger for I/O Units

OmniCore EtherNet/IP Scanner supports communication with I/O units using **Cyclic** or **Change of State** connections.

- **Cyclic**: Data is sent at a fixed interval defined by the RPI.
- **Change-of-State (COS)**: Data is sent when a change of state of the channel occurs or at least every RPI. COS can therefore be used to achieve faster response times and reduce both Controller load and network traffic.

  It is also recommended to set **Production Inhibit Time** for COS connections. See *Guideline to select Production Inhibit Time for Change of State on page 49*.

Recommendation for production trigger in OmniCore is to use **Change of State** for application with high demand on response times. This is preferable instead of using **Cyclic** with fast cycle times.

> **Note**
>
> COS can only be used if the external I/O unit supports it.

> **Note**
>
> All ABB Scalable I/O supports COS.

Guideline to select RPI for devices

RPI (Requested Packet Interval) is the rate at which cyclic I/O data is exchanged. It applies to both produced (sent) and consumed (received) data thus the settings Output RPI and Input RPI.

Select RPI for Cyclic production trigger

Choosing the right RPI is a trade-off between response times and network/controller load.

OmniCore supports minimum RPI 2 ms for application with max 10 devices and minimum RPI 8 ms otherwise.

> **Note**
>
> For the OmniCore controller, the application complexity (use of RAPID and motion control) however restricts the practical minimum RPI.

- **Determine application requirements**

  Do not set the RPI to a lower value than the response time your application requires. For fast response times consider using Change of State connection.
- **Consider network bandwidth and Controller load**

  To many low RPI devices can cause jitter.
- **Device capability**

Continues on next page

Some devices have a minimum supported RPI. Exceeding device capability can lead to loss of connections.

- **Network topology**

  Number of network switches between the OmniCore EtherNet/IP Scanner and the external device affects the minimum RPI as each switch introduces small delays.

Select RPI for Change of State connection

For a change of state connection, the **RPI** serves more like a heartbeat.

It defines the maximum time between packets, even if no data changes.

If no data change occurs within the **RPI**, the device must still send a packet to indicate it is alive and the connection is healthy.

Recommendation for a **Change of State** connection is to choose an **RPI** that reflects how quickly you need to detect a loss of communication (in combination with the **Connection Timeout Multiplier**).

If you want to detect a failure within 1000 ms, set **RPI** to 250 ms and **Connection Timeout Multiplier** to x4.

Guideline to select Production Inhibit Time for Change of State

Production Inhibit Time is the minimum time interval that must elapse between two consecutive I/O data transmissions from the device to the scanner. It prevents the OmniCore EtherNet/IP Scanner and the external device from sending data too frequently, which helps avoid too high network load and too high CPU load in the OmniCore controller.

6.3 Configuring the internal scanner properties

1. In the **Controller** tab in RobotStudio, select **I/O Engineering**. The **I/O Engineering** tab is displayed.
2. In the **Configuration** browser, select the **EthernetIP** node.
3. In the **Properties** browser, you can configure the following:

Parameter	Description	Allowed values
Simulated	Select **Yes** or **No**, indicating if the industrial network and all its connected I/O devices should be treated as simulated.  **Note** If **Simulated** is set to **Yes**, network configuration is not required.	The default value is No.
Identification Label	This parameter is an optional way to provide a label that will help the operator to identify the internal device.	A string with maximum 80 characters.

6.4 Adding devices to the internal scanner

1. In the **Configuration** browser, expand **EtherNetIP** and select the internal scanner.
2. Select the **Device Catalogue** tab to show a list of available devices.

> **Note**
>
> The EDS files that have been imported to the project define what devices can be selected. See *Importing EDS files*.

3. Double-click on a device in the list to add it to the internal scanner.
4. All necessary device information is retrieved from the EDS file, but some parameters can be changed in the **Properties** tab if needed. See table below in *on page 52* for a description of each parameter.

6.5 Configuring EtherNet/IP generic devices

Configuring generic devices is useful when no EDS file is available.

1. In the **Configuration** browser, expand **EtherNetIP** and select the internal scanner.
2. In the **Device Catalogue**, double-click on **EtherNetIP Generic Device** to add a generic device.
3. In the **Properties** tab, specify the properties for the generic device:

> **Note**
>
> A red frame around a property field means that the property is not specified, or causes a validation error.

Parameter	Description	Allowed values
Name	The name of the internal scanner device is used as a reference to the specific device when configuring the I/O signals and device commands.	A string with maximum 32 characters. The string must follow the RAPID rules described in *Technical reference manual - RAPID Overview*. The name must be unique among all named objects in the I/O system configuration. **Note** Names differing only in upper and lower case are considered to be equal.
Identification Label	This parameter is an optional way to provide a label that will help the operator to identify the internal scanner device.	A string with maximum 80 characters.
Vendor Name	The name of the I/O device vendor.	A string with maximum 80 characters.
Product Name	The product name for this I/O device according to industrial network type standard.	A string with maximum 80 characters.
Vendor ID	This parameter is used as an identification of the I/O device to secure communication to the correct device.	Allowed values are the integers 0-65535.
Product Code	This parameter is used as an identification of the I/O device to secure communication to the correct device.	Allowed values are the integers 0-65535.
Device Type	This parameter is used as an identification of the I/O device to secure communication to the correct device.	Allowed values are the integers 0-65535.
Major Revision	Used together with **Minor Revision** to define the revision of the device.	Allowed value is an integer between 0 and 127.
Minor Revision	Used together with **Major Revision** to define the revision of the device.	Allowed value is an integer between 0 and 127.
Compatibility	This parameter makes it possible to install devices that can emulate the exact device.	

Continues on next page

Parameter	Description	Allowed values
Trust Level	Select an existing trust level that defines the behavior for external devices at different execution situations in the robot controller. See *Application manual - I/O Engineering* for more information about how to create trust levels.	
Simulated	Select **Yes** or **No**, indicating if the industrial network and all its connected I/O devices should be treated as simulated. **Note** If **Simulated** is set to **Yes**, network configuration is not required.	The default value is No.
State when System Startup	Defines the logical state that the robot system shall try to set for the external device at system startup. The available options are: • Establish communication (Activated) • Don't establish communication (Deactivated) • Restore the previously stored logical state for the external device at system shutdown (Last State)	
Output Assembly	Specifies where the output data for an I/O device is located. The output assembly is vendor specific.	Allowed values are the integers 0-65535.
Output Size	Defines the output data size in bytes for an I/O device.	Allowed values are the integers 0-505 (0-4040 signal bits), specifying the data size in bytes.
Output RPI	*Output RPI* (Originator to Target Request Packet Interval) is the time between I/O packets from the scanner to the I/O device. Use this parameter to decide at which interval the scanner shall produce output data to the I/O device. The Request Packet Interval is specified in micro seconds. **Note** In case of connection problems, it is recommended to increase the Connection Timeout Multiplier.	The minimum limit is 2 and maximum limit is 4.294967E+09.
Input Assembly	Input Assembly specifies where the input data for an I/O device is located. The input assembly is vendor specific.	Allowed values are the integers 0-65535.
Input Size	Defines the input data size in bytes for an I/O device.	Allowed values are the integers 0-505 (0-4040 signal bits), specifying the data size in bytes.

Continues on next page

Parameter	Description	Allowed values
Input RPI	*Input RPI* (Target to Originator Request Packet Interval) is the time between I/O packets from the I/O device to the scanner. Use this parameter to decide at which interval the scanner shall consume input data from the I/O device. The Request Packet Interval is specified in micro seconds. **Note** In case of connection problems, it is recommended to increase the Connection Timeout Multiplier.	The minimum limit is 2 and maximum limit is 4.294967E+09.
Input Connection Type	The *Input Connection Type* parameter specifies how I/O data is send from the I/O device to the scanner. There are two different connection types: • **Point-to-point (Unicast)**: A connection where the data is send from one point to another point. In this case there is just one sender and one receiver. • **Multicast**: A connection where the data is send from one or more points to a set of other points. In this case there is one sender and multiple receivers. **Note** Some EtherNet/IP I/O devices might not support Point-to-point as input connection type.	
Configuration Assembly	Specifies where the configuration data for a device is located. The configuration assembly is vendor specific.	Allowed values are the integers 0-65535.
Configuration Size	*Configuration Size* specifies the size of the *Configuration Assembly*.	Integer between 0 and 400, specifying the data size in bytes.

Continues on next page

Parameter	Description	Allowed values
Ownership	The *Ownership* parameter specifies how the I/O connection shall act between the scanner and the I/O device. There are three different types of Ownership: • Exclusive Owner: An I/O connection where the data of an I/O device can be controlled only by one scanner. • Input Only: An I/O connection where only the scanner can receive input data from an I/O device. There is no output data. • Listen Only: An I/O connection where only the scanner can receive input data from an I/O device. This type of *Ownership* can only be attached to an connection of type; Exclusive Owner or Input Only. If this underlying connection closes, then the connection with Ownership of type; Listen Only will also be closed. There is no output data. **Note** Some EtherNet/IP devices might not support the Input Only connection.	Allowed values are Exclusive Owner, Input Only, or Listen Only.
Connection Timeout Multiplier	*Connection Timeout Multiplier* specifies the multiplier applied to the expected packet rate value to derive the value for the Inactivity/Watchdog Timer. The *Connection Timeout Multiplier* is a number among 4, 8, 16, 32, 64, 128, 256. It is used together with RPI to calculate the timeout on connections. *RPI* multiplied by *Connection Timeout Multiplier* gives the maximum time before dropping the connection.	Allowed values are 4, 8, 16, 32, 64, 128, 256, 512.
Connection Priority	The *Connection Priority* parameter specifies how I/O data is prioritized on the network. Network priority is accomplished by using Quality of Service (QoS) mechanisms in the device.	Allowed values are Low, High, Schedule, Urgent.
Production Trigger	Select **Change of State** or **Cyclic** indicating the type of I/O connection to be used.	
Production Inhibit Time	**Production Inhibit Time** is used together with the production trigger **Change of State** indicating the frequency with which a signal change can occur. This value is calculated as Request Packet Interval (RPI) divided by 4.	

6.6 Saving the configuration

Save configuration and write to controller

1. Select **I/O Project**/**Save Project** to save the changes.
2. Select **Read and Write** to write the configuration to the controller.

See *Application manual - I/O Engineering* for more information.

7 Additional configuration

7.1 QuickConnect

Overview

The *QuickConnect* functionality provides the connection between the EtherNet/IP scanner and the device to quickly disconnect and reconnect to the Ethernet network, both mechanically and logically. With the *QuickConnect* functionality activated, the device will be connected and operational by the EtherNet/IP Scanner under 500 ms.

Requirements

A QuickConnect system requires an electrical lock signal that indicates, when power has been applied to the QuickConnect devices. This signal must be implemented by the system builder and is used to start the QuickConnect sequence.

Additional system component requirements:

- Managed network switch(es)
- QuickConnect device(s):

  A QuickConnect device has *QuickConnect* functionality disabled as default. This functionality must be activated for proper function.

  It can be done using:

  - Configuration data via the configuration *Assembly*
  - Third party tool before connecting the module to OmniCore controller Ethernet/IP scanner
  - EtherNet/IP command. See *Enabling device for QuickConnect operation on page 58*.

Note

When connecting QuickConnect devices, it is essential that network switches allow *gratuitous ARP* to exist on the network. Gratuitous ARP is issued by QuickConnect devices during startup to inform other network devices that they are ready to join the network.

Note

While using *QuickConnect*, make sure to turn off autonegotiation on the link that is disconnected. For example, in the connector on the switch (or in the connector for the OmniCore controller) and in the connector on the I/O device.

An error message appears if trying to activate or deactivate the *QuickConnect* functionality on an I/O device that does not support *QuickConnect*.

Sequence

1 The OmniCore controller deactivates current connections to QuickConnect devices, and the robot arm physically disengages the current tool.

Continues on next page

2 The robot arm physically attaches to the new QuickConnect devices.
3 The new QuickConnect devices power up.
4 The OmniCore controller acknowledges a successful attachment to a new tool via an electrical lock signal.
5 Upon receiving the electrical lock signal, the OmniCore controller waits a specific time[I] for the QuickConnect devices to power up before activating the devices.

[I] The QuickConnect time can be found in the EDS file for the QuickConnect device.

Enabling device for QuickConnect operation

This instruction describes how to enable a device for QuickConnect through RobotStudio.

> **Note**
>
> Before enabling the device through RobotStudio, read the supplier manual for the specific device and follow those instructions. Only continue with the procedures in RobotStudio if necessary.

Setting the QuickConnect attribute using the IO command

1 Start RobotStudio and connect to the OmniCore controller. Request write access.
2 In the **Controller** tab, select **I/O Engineering**.
3 In the **Configuration** browser, expand the internal scanner device and select **IO Commands**.
4 In the **Device Catalogue**, double-click **QuickConnect** to add the command to the device.
5 The command details can be viewed in the **Properties** tab. See *IO command system parameters on page 61* for detailed information about the parameters.
6 Save the I/O project and restart the controller.

7.2 Communication between two OmniCore controllers

General

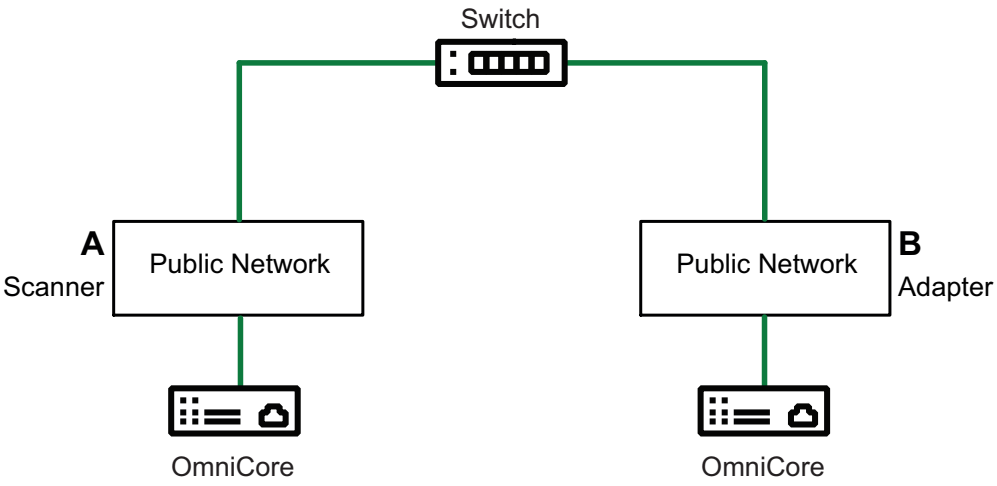
When two OmniCore controllers are connected to each other through EtherNet/IP, one of them must be acting as an adapter device and the other one must be acting as a scanner.

Note

It is possible to configure both the scanner and an adapter device in the same OmniCore controller.

Illustration

The following figure illustrates communication between two OmniCore controllers.



xx1900000773

Note

The switch is optional. You can use an Ethernet cable when there is no switch.

Limitations

The address specified in the *Industrial Network* cannot be the same on the two controllers since they shall be interconnected.

Configuring the scanner/adapter controllers

The following procedures describe the configuration of a hardware setup like the one illustrated in section *Illustration on page 59*.

1 Configure the EtherNet/IP industrial network address for both the OmniCore controllers. See *Technical reference manual - System parameters*.

Continues on next page

> **Note**
>
> Be sure to use different EtherNet/IP addresses for the two OmniCore controllers to avoid duplicated addresses on the interconnected network.

2 Configure the EtherNet/IP adapter device according to the configuration procedure for the EtherNet/IP internal adapter device. See *Configuring the internal adapter on page 33*.

3 Configure the EtherNet/IP scanner to connect to the EtherNet/IP internal adapter device. See *Configuring the internal scanner on page 47*.

> **Note**
>
> Use ABB EtherNet/IP Adapter Device template when configuring the EtherNet/IP scanner to connect to the EtherNet/IP Adapter.

4 Configure signals on the created device.

5 Physically interconnect the two OmniCore controllers.

6 Restart the adapter controller.

7 Restart the scanner controller.

The scanner will now connect to the internal adapter controller.

8 Now it is possible to set output signals on one controller.

The output signals shall appear as inputs on the other controller.

8 Explicit messaging services

8.1 Information

General

It is possible to configure I/O devices through explicit messaging services. This could be done either at startup by defining the IO command to the configured device, or at runtime from RAPID through the *Fieldbus Command Interface* (FCI).

See *Explicit messaging services (IO command) at startup on page 64* and *Explicit messaging services (IO command) via RAPID on page 65*.

> **Note**
>
> For information about which explicit messaging services are available for a specific I/O device and how to set the parameters, refer to the supplier documentation of the I/O device and the *Common Industrial Protocol (CIP) Specification*, see *References on page 7*.

IO command system parameters

The IO command specific system parameters are:

- Path (`Path`).
- Service (`Service`).
- Download Order (`-OrderNr`).

> **Note**
>
> For more information, see *Technical reference manual - System parameters*.

The *Path* parameter

Following is a short description of the syntax used in the `Path` parameter.

```
"Path length, 20 Class 24 Instance 30 Attribute, Data type, Data
     type length"
```

The following table provides a description of the parameters used in the syntax:

Parameter	Description
Path length	The byte count for the "20 64 24 01 30 05" string. This is an optional parameter.
Class	The EtherNet/IP class number.
Instance	The instance number of the class.
Attribute	The attribute of the specified instance.
Data type	The data format of the attribute. This is an optional parameter.
Data type length	The length in bytes of the specified Data type. The highest allowed value is 0x20 (32 bytes). This parameter is ignored, but is accepted if entered.

Continues on next page

The following table provides a list of the allowed data types for the parameter *Data type*:

Data Type	Value	Description
CIP_EXPL_BOOL	C1	Logical Boolean with values TRUE and FALSE
CIP_EXPL_SINT	C2	Signed 8-bit integer value
CIP_EXPL_INT	C3	Signed 16-bit integer value
CIP_EXPL_USINT	C6	Unsigned 8-bit integer value
CIP_EXPL_UINT	C7	Unsigned 16-bit integer value
CIP_EXPL_UDINT	C8	Unsigned 32-bit integer value
CIP_EXPL_REAL	CA	32-bit floating point value
CIP_EXPL_STRING	D0	Character string (1 byte per character)
CIP_EXPL_BYTE	D1	Bit string - 8-bits
CIP_EXPL_WORD	D2	Bit string - 16-bits
CIP_EXPL_DWORD	D3	Bit string - 32-bits
CIP_EXPL_SHORT_STRING	DA	Character string (1 byte per character, 1 byte length indicator)

The following table provides a list of what delimiter to use for the parameter *Value*, if the data is an array:

Data Type	Delimiter	Example
CIP_EXPL_BOOL CIP_EXPL_SINT CIP_EXPL_INT CIP_EXPL_USINT CIP_EXPL_UINT CIP_EXPL_UDINT CIP_EXPL_REAL CIP_EXPL_BYTE CIP_EXPL_WORD CIP_EXPL_DWORD	The values are delimited by space.	"123 214 125 2 44" An array of 5 elements. The Data Type specifies the type of each element.
CIP_EXPL_STRING CIP_EXPL_SHORT_STRING	The values are delimited by semicolon.	"Hello;This;Is;My;Name" An array of 5 elements of string type.

The *Service* parameter

The *Service* parameter describes what type of operation that should be performed against the specified *Path* parameter.

Following are the allowed values for *Service*:

Operation	Value	Description
Set Attribute Single	16	Set the value specified in parameter *Value* of the *EtherNet/IP Command*.
Reset	5	Performs a reset of the specified device.

Continues on next page

The *Download Order* parameter

The *Download Order* parameter is used to specify in what order the commands are sent to the I/O device.

If an EtherNet/IP Command is rejected by the I/O device, the EtherNet/IP scanner will generate an event message with the error code returned by the I/O device.

8.2 Explicit messaging services (IO command) at startup

Information

It is possible to configure IO commands that will be sent to a device at startup.

The command is specific to the I/O device and will only be sent to the assigned I/O device.

Configuring IO commands in I/O Engineering

1. In the **Configuration** browser, expand the internal scanner device and select **IO Commands**.
2. In the **Device Catalogue**, double-click **EtherNetIP Generic Command** to add the command to the device.
3. The command details can be configured in the **Properties** tab. See *IO command system parameters on page 61* for detailed information about the parameters.

Example using *IO Command*

The following is a configuration example from RobotStudio that sends four IO commands at startup to an I/O device, *EN_Device*. There are four different specific commands that are sent to the device to perform specific operations on it. The example shows how to use the *Path* and *Service* parameters.

Parameter				
Name	RackSize	LastRack	StructuredType	ArrayOfWords
Device	EN_Device	EN_Device	EN_Device	EN_Device
Download Order	1	2	3	4
Path	6, 20 64 24 01 30 04, C6	6, 20 64 24 01 30 05, C1	20 64 24 01 30 12	6, 20 64 24 01 30 13, C7
Service	Set Attribute Single	Set Attribute Single	Set Attribute Single	Set Attribute Single
Value	0	0	45 00 35 00 255 255 255 255	100 200 300

Note

If a class, instance, or attribute below 0x10 is specified, it is important to include a "0" before the value. For example, the value 8 is written as 08 in the *Path* string.

Note

The path to different commands can be found in the EtherNet/IP specification from ODVA or in the vendor manual for the device.

8.3 Explicit messaging services (IO command) via RAPID

Information

For more information about the RAPID instructions, see *Technical reference manual - RAPID Instructions, Functions and Data types*.

Example

In this example, data packed as a `rawbytes` variable is read from an EtherNet/IP I/O device.

```
PROC get_quickconnect_value()
  VAR iodev dev;
  VAR rawbytes rawdata_out;
  VAR rawbytes rawdata_in;
  VAR num input_int;
  VAR byte return_status;
  VAR byte return_errcodecnt;
  VAR num return_errcode;
  VAR byte value;

  ! Empty contents of rawdata_out and rawdata_in
  ClearRawBytes rawdata_out;
  ClearRawBytes rawdata_in;

  ! Add Fieldbus command header to rawdata_out with service
       "GET_ATTRIBUTE_SINGLE" and path to QuickConnect attribute
       on I/O unit.
  PackDNHeader "0E", "6,20 F5 24 01 30 0C", rawdata_out;

  ! Open FCI device
  Open "/FCI1:" \File:="TheUnit", dev \Bin;

  ! Write the contents of rawdata_out to dev
  WriteRawBytes dev, rawdata_out \NoOfBytes :=
       RawBytesLen(rawdata_out);

  ! Read the answer from dev
  ReadRawBytes dev, rawdata_in;

  ! Close FCI device
  Close dev;

  ! Unpack rawdata_in to the variable return_status
  UnpackRawBytes rawdata_in, 1, return_status \Hex1;

  ! The first byte is always the general status byte. 0 means
       success, see the CIP standard error codes.
  IF return_status = 0 THEN
    TPWrite "Status OK from device. Status code:
         "\Num:=return_status;
    ! Unpack the read data value that follows the status byte.
```

Continues on next page

8.3 Explicit messaging services (IO command) via RAPID
Continued

```
    UnpackRawBytes rawdata_in, 2, value \Hex1;
    TPWrite "Read value: " \Num:=value;
  ELSE
    ! If the general status was not ok there is extended error
         information that can be retreived. First byte, after the
         general status byte, tells how many extended error words
         can be found.
    UnpackRawBytes rawdata_in, 2, return_errcodecnt \Hex1;
    ! Unpack the number of extended status words. In this example
         only the first one is unpacked.
    UnpackRawBytes rawdata_in, 3, return_errcode \IntX := UINT;
    TPWrite "Error code from device: "\Num:=return_status;
    TPWrite "Additional error code count from device:
         "\Num:=return_errcodecnt;
    TPWrite "Additional error code from device:
         "\Num:=return_errcode;
  ENDIF
ENDPROC
```

9 Troubleshooting

9.1 Frequently asked questions

Is it recommended to configure the WAN connector and the selected LAN connector on the same subnet?

No, the EtherNet/IP address for the WAN connector must belong to another subnet than the address of the selected LAN connector.

For example, if the address of the WAN connector is 111.122.133.144, the address for the selected LAN connector cannot be 111.122.133.145 if the subnet mask 255.255.255.0 (but it can be 111.122.**134**.145) is being used.

How to identify the desired EtherNet/IP I/O devices on the network?

There are two ways to identify the EtherNet/IP devices on the industrial network.

There is a list with the I/O device names, IP addresses and corresponding MAC addresses. In the window **Inputs and Outputs**, tap **View** and select **Industrial Networks**. Select the desired EtherNet/IP network and tap **I/O Device Identification** in the command bar. A window will be displayed with all the devices on the selected EtherNet/IP industrial network and their corresponding IP and MAC addresses.

Another way is to open the **Inputs and Outputs** window, tap **View** and select **Devices**. Select the EtherNet/IP device to be identified, tap **Actions** in the command bar and select **Unit Identification**. A message box will be shown displaying the MAC address for the selected device.

Can tool change be done without using dedicated QuickConnect I/O devices?

Yes. If time is really not important, there are many different I/O devices available today which can serve as tool changer equipment.

The only requirement for proper and deterministic behaviour is that, the I/O device must issue gratuituos ARP requests when powered on. If so, the connection time will be determined by the I/O device startup time.

If the I/O device does not support gratuitous ARP, the connection time will be dependent on the refresh of ARP timers in the robot communication software. Typically 20 seconds can be expected but it can take up to some minutes. Also, the error log "71058 Lost communication with I/O device" will be issued.

9.2 Error log descriptions

Error log "71367 No communication with I/O device" is shown after startup

	Action
1	Check cabling.
2	Ensure that the device address matches the configuration.
3	Ensure that all addresses are unique, and not used by more than one device.
4	If the address is changed, the power supply to the device must be cycled (switched OFF and then back ON) to ensure the address has been changed.
5	Verify that the configured *Input Assembly* and *Output Assembly* correspond to the data in the EDS file for your I/O device.
6	Verify the configured *Input Size* and *Output Size*.
7	Verify if the device needs the *Configuration Assembly*. See the EDS file. Too low Request Packet Interval is configured. See the manual of device.

Error log "71058 Lost communication with I/O device" is shown when activating QuickConnect I/O devices

This error occurs when connecting to the device when it is not ready to join the network.

	Action
1	Check that the QuickConnect device is activated for *QuickConnect* functionality.
2	Check if the electrical lock signal is working as expected.
3	Check if the OmniCore controller waits for sufficient amount of time after electrical lock has been engaged before connecting to the device.
4	Check if the network allows the gratuitous ARP request correctly. This is essential when doing QuickConnect on devices with the same IP address. The Spanning Tree Protocol should be disabled at the switch.

10 Reference material

10.1 How to establish CIP Safety communication with a PLC

10.1.1 Introduction

General

This section explains how to establish CIP Safety communication between an Allen-Bradley controller acting as a CIP Safety Scanner, and a robot controller acting as a CIP Safety Adapter.

Note

Currently, it is not possible to parse OmniCore safe EDS file with the Studio5000 Logix Designer Tool, since the tool does not support multiport devices. Therefore, the user cannot import the EDS file of the ABB CIP Safety adapter and use it for CIP Safety applications. The only way to establish CIP Safety communication with the ABB adapter is to use the generic profiles present in Logix Designer.

CAUTION

This description is only for informative purposes without any warranties, the original manual provided by the manufacturer of the scanner has precedence. The user must follow any and all safety requirements and advises provided by the manufacturer of the scanner.

List of parameters

Connection parameters

Parameter	Setting
IP Address	User defined. The controller and the corresponding module in the PLC's scan list must have the same IP addresses.
Safety Network Number (SNN)	The *Safety Network Number (SNN)* provides a unique network identifier for each network in the safety system and is generated by the safety PLC. The SNN in the controller and the PLC must match.
Assembly Size	Fixed size 8 bytes
NodeID	The same as the IP Address.
RPI	*Requested Packet Interval (RPI)* ≥10ms

Assembly instance IDs

Parameter	Setting
Safety input	121
Safety output	120
Non-safety input	100

Continues on next page

10.1.1 Introduction
Continued

Parameter	Setting
Non-safety output	112
Null path	199
Configuration	199

Module parameters

Parameter	Setting	
Vendor	75	
Device type (product type)	101	
Product code	E10 Type A	3
	C30 Type B (with option *3091-1 Extension switch*)	5
	C30 Type B	4
	C90XT Type A	2
	V250XT Type B	2
	V400XT	2
Revision (major/minor)	2/50	

10.1.2 Configuring the ABB CIP Safety Adapter in Studio 5000®

Prerequisites

Configure the ABB CIP Safety Adapter in I/O Engineering, see *Configuring the safe internal adapter device (CIP Safety) on page 38* and *Application manual - I/O Engineering*.

Start the *Studio 5000 Logix Designer*. Before proceeding with the configuration make sure that the PLC controller is in offline mode.

Adding a new generic module

Use this procedure to add a new generic module to the scan list of the PLC controller.

1. In the **Controller Organizer** window, right click on the Ethernet item in the I/O Configuration folder. Select **New Module**.
2. Type "generic" in the search window to filter desired modules.
3. Select one of the available generic module profiles, *ETHERNET-SAFETYMODULE* for CIP safety communication only, or *ETHERNET-SAFETY-STANDARD-MODULE* for CIP safety and non-safety communication capabilities.
4. Click the **Create** button. A new **Module Properties** window appears, allowing for configuring of a new Ethernet/IP module.

Configuring the general properties

Use this procedure to configure the general properties of the ABB CIP Safety module.

1. Go to the **General** tab.
2. Enter the name of the module, for example *ABB safety adapter*.
3. Enter the IP address where the module is located, for example *192.168.0.111*.
4. Enter, or generate, the *Safety Network Number (SNN)*, for example *1234_5678_9ABC*. Note the underscore symbols.

   CAUTION

   The user should assign **SNN** numbers for each safety network or safety subnet that are unique system-wide.

5. Go to the **Module Definition** dialog box by clicking the **Change** button in the lower right corner.
6. In the **Module Definition** dialog box, use the **General** tab to configure vendor, product type, product code, and minor/major revision numbers. See *Module parameters on page 70*.
7. Select **Electronic Keying** to *Compatible Module*.

Continues on next page

10.1.2 Configuring the ABB CIP Safety Adapter in Studio 5000®
Continued

8 In the **Module Definition** dialog box, use the **Connections** tab to configure size and assembly instances used by the ABB robot module, see the following table:

Connection	Input		Output		Configuration	
	Assembly instance	Size (bytes)	Assembly instance	Size (bytes)	Assembly instance	Size (bytes)
Safety input	121	8	199	-	199	-
Safety output	199	-	120	8	-	-
Standard	100	User defined	112	User defined	199	0

9 Accept the module definition by clicking **OK**. Check the module definition details in the **General** tab of the **Module Properties** window.

10 Open the **Connection** tab of the **Module Properties** window and enter the desired RPI values. Note that the RPI for output connection is read-only and set by the safety task in the **Controller Organizer**, see **Safety Task -> Configuration -> Period**.

11 If configuration signature shall be used, open the **Safety** tab of the **Module Properties** window to configure the **Configuration Signature** settings. Enter the **ID**, **Date**, and **Time** fields with the values obtained from the ABB safety report or I/O Engineering. See *Copy SCID to Studio 5000® on page 40*.

Note

When configuring safety connections without the configuration signature, the user is responsible for ensuring that originators (scanners) and targets (adapters) have the correct configurations.

12 Accept the module configuration by clicking **Apply** in the **Module Properties** window.

Downloading the configuration

Use this procedure to download the configuration to the PLC controller.

1 From the main menu bar, select **Communication -> Go Online** or **Communication-> Download** to download the configuration to the controller.

Checking the status

Use this procedure to check the status of the communication.

1 Go to **Controller Organizer -> I/O Configuration -> Ethernet**, then right click on the ABB robot module and select **Properties**.

Tip

You can also open the **Properties** window by double-clicking on the ABB robot module.

Continues on next page

2 Check the status and fault messages of the module by examining the **Connection** tab.

10.2 Object dictionary

Overview

This chapter describes the CIP objects supported by OmniCore Ethernet/IP Scanner/Adapter.

The following table shows which objects are implemented.

Object Class	Class Code (Hex)	Number of Instances	Remark
Identity	0x01	1	
Message Router	0x02	1	
Assembly	0x04	4	
Connection Manager	0x06	1	
Safety Supervisor	0x39	1	Safety device type only
Safety Validator	0x3A	2	Safety device type only
Quality of Service	0x48	1	
Port	0xF4	2	
TCP/IP Interface	0xF5	1	
Ethernet Link	0xF6	2	

Description of each object in the table is described in the subchapters below.

Identity Object

The Identity object (Class Code 01hex) provides general information about the device, such as revision and serial number.

Class attributes

Attribute ID (hex)	Access rule	Attribute name	Data type	Default value	Description
1	Get	Revision	U16	02	

Instance attributes

Attribute ID (hex)	Access rule	NV	Attribute name	Data type	Default value	Description
1	Get	NV	Vendor	U16	75	ABB Robotics
2	Get	NV	Device Type	U16	100 or 101	Vendor Specific
3	Get	NV	Product Code	U16	3	E10 Type A
					5	C30 Type B (with option *3091-1 Extension switch*)
					4	C30 Type B
					2	C90XT Type A
					2	V250XT Type B
					2	V400XT

Continues on next page

Attribute ID (hex)	Access rule	NV	Attribute name	Data type	Default value	Description
4	Get	NV	Revision	STRUCT of:		
			Major Revision	U8		
			Minor Revision	U8		
5	Get	V	Status	U16		
6	Get	NV	Serial Number	U32		
	Get	NV	Product Name	STRING	OmniCore	
7	Get	NV	Product Name	STRING	OmniCore E10A or OmniCore Safe E10A	E10 Type A
					OmniCore C30B or OmniCore Safe C30B	C30 Type B (with option *3091-1 Extension switch*)
					OmniCore C30B Basic or OmniCore Safe C30B Basic	C30 Type B
					OmniCore or OmniCore Safe	C90XT Type A
					OmniCore or OmniCore Safe	V250XT Type B
					OmniCore or OmniCore Safe	V400XT
8	Get	V	State	U8		

Services

Service code (hex)	Implemented		Service name	Description
	Class	Instance		
01	NO	YES	Get Attributes All	
0E	NO	YES	Get Attribute Single	

Message Router Object

The Message Router object (Class Code: 02hex) exists only as an internal message router. Attributes or service are usable by the client.

Class attributes

No class attributes are supported.

Instance attributes

No instance attributes are supported.

Continues on next page

Services

No services are supported.

Assembly Object

The Assembly object (Class Code: 04hex) is used to collect data from multiple objects in the device to one or several data arrays. This enables a lot of different data being sent over one communication channel.

Class attributes

Attribute ID (hex)	Access rule	Attribute name	Data type	Default value	Description
1	Get	Revision	U16	02	

Instance attributes

Attribute ID (hex)	Access rule	NV	Attribute name	Data type	Default value	Description
3	Get	V	Data	ARRAY of: U8		

Services

Service code (hex)	Implemented		Service name	Description
	Class	Instance		
0E	NO	YES	Get Attribute Single	

Data mapping

Instance Number	Type	Description
100	Input data	Standard input data. Size dependent on configuration in I/O Engineering.
112	Output data	Standard output data. Size dependent on configuration in I/O Engineering.
120	Safety Input data	Safety input data. Size 8 bytes.
121	Safety Output data	Safety output data. Size 8 bytes.
199	Configuration	Null instance for configuration data. 0 bytes.

Non-Safety Assemblies

Assembly Instance nr	Assembly Name	Name	Data type	Description
100	Inputs	Input Element Array for non-safety modules.	ARRAY of BYTE	The order of the data is dependent of the signal configuration in I/O Engineering.
112	Outputs	Outputs Element Array on non-safety inputs.	ARRAY of BYTE	The order of the data is dependent of the signal configuration in I/O Engineering.

Continues on next page

Safety Assemblies

Assembly Instance nr	Assembly Name	Name	Data type	Description
120	Safety Inputs	Input Element Array on safety inputs.	ARRAY of BYTE	The order of the data is dependent of the signal configuration in I/O Engineering.
121	Safety Outputs	Outputs Element Array on safety inputs.	ARRAY of BYTE	The order of the data is dependent of the signal configuration in I/O Engineering.

Connection Manager Object

The Connection Manager (Class Code: 06hex) is object manages connected explicit and implicit communication channels.

Class attributes

No class attributes are supported.

Instance attributes

No instance attributes are supported.

Services

Service code (hex)	Implemented		Service name	Description
	Class	Instance		
4E	NO	YES	Forward Open	
54	NO	YES	Forward Close	

Safety Supervisor Object

The Safety Supervisor Object (Class Code: 39hex) manages the states of a CIP Safety device and provides access to safe identification and configuration data. Detailed information about this CIP object can be found in the CIP Networks Library Volume 5.

Class attributes

Attribute ID (hex)	Access rule	Attribute name	Data type	Default value	Description
1	Get	Revision	UINT	01	Revision of the Safety Supervisor Object Class Definition.

Continues on next page

10.2 Object dictionary
Continued

Instance attributes

Attribute ID (hex)	Access rule	NV	Attribute name	Data type	Default value	Description
11	Get	V	Device Status	USINT		1: Self-testing 2: Idle 3: Self-test exception 4: Executing 5: Abort 6: Critical Fault 7: Configuring 8: Waiting for TUNID
12	Get	V	Exception Status	BYTE		Indicates the status of alarms and warnings for the device. Bit 7 is always zero, so only the Basic Method is supported. (Exception Detail Alarm and Warning not supported)
15	Get/Set	NV	Alarm Enable	Bool		Enables or disables the Safety Superior Object's process of setting Exception Status bits.
16	Get/Set	NV	Warning Enable	Bool		
25	Get	NV	Configuration UNID	10 Octets		CFUNID – Identifies the owner of a Device Configuration. all 0xFF = Tool-only configuration 0 = unowned, accept any owner
26	Get	NV	Safety Configuration Identifier (SCID)	10 Octets		The SCID is comprised of the Safety Configuration CRC + Safety Configuration Time Stamp. This is a signature of the Configuration.
28	Get	NV	Output Connection Point Owner	Struct of:		
			Number of Array Entries	UINT		Number of OCPUNID struct entries
			Output Owners	Array of Structs		
			OCPUNID	10 Octets		The owner UNID for the output resource
			ePath Size	USINT		Path size, number of bytes
			Application Re-source	Packed ePath		The path to owned re-sources
30	Get	NV	Target UNID List	Struct of:		
			Number of Target UNIDs	USINT		Number of Target UNID list Entries
			Target UNIDs	Array of Structs		List of (Port number, Target UNID) pairs
			Port Number	UINT		
			Target UNID	10 Octets		Target UNID associated with a specific CIP Port Number

Continues on next page

Services

Service code (hex)	Implemented		Service name	Description
	Class	Instance		
0x0E	YES	YES	Get Attribute Single	Returns the contents of the specified attribute
0x10	NO	YES	Set Attribute Single	Modifies an attribute value

Safety Validator Object

The Safety Validator Object (Class Code: 3Ahex) coordinates and maintains reliable safety connections between client and server safety applications. Detailed information about this CIP object can be found in the CIP Networks Library Volume 5.

Class attributes

Attribute ID (hex)	Access rule	Attribute name	Data type	Default value	Description
1	Get	Revision	UINT	-	Revision of the Safety Validator Object Class Definition.
8	Get	Safety Connection Fault Count	UINT	-	Diagnostic Counter that is a running count of Safety Connection Faults

Instance attributes

Attribute ID (hex)	Access rule	NV	Attribute name	Data type	Default value	Description
1	Get	V	Safety Validator State	USINT		1: Self-testing 2: Idle 3: Self-test exception 4: Executing 5: Abort 6: Critical Fault 7: Configuring 8: Waiting for TUNID
2	Get	V	Safety Validator Type	USINT		Safety Validator type used in this instance. 0 = unallocated 1= Single-cast 2 = Multi-cast 3-127 = Reserved

Continues on next page

Attribute ID (hex)	Access rule	NV	Attribute name	Data type	Default value	Description
12	Get/Set	V	Max Data Age	UINT	-	Diagnostic which holds the largest Data Age detected in 128 µS increments. Attribute only used for Safety Consumers.

Services

Service code (hex)	Implemented		Service name	Description
	Class	Instance		
0x0E	YES	YES	Get Attribute Single	Returns the contents of the specified attribute
0x10	NO	YES	Set Attribute Single	Modifies an attribute value
0x4B	YES	NO	Reset all error counters	Used to reset class attribute 8 in all instances

Quality of Service Object

The Quality of Service object (Class Code: 48hex) provides a mechanism to prioritize TCP/UDP/IP packets which the OmniCore produces on the network. OmniCore supports prioritization through Differentiated Services Code Points (DSCP).

Class attributes

Attribute ID (hex)	Access rule	Attribute name	Data type	Default value	Description
1	Get	Revision	UINT	01	Revision of the QoS Object Class Definition.

Instance attributes

Attribute ID (hex)	Access rule	NV	Attribute name	Data type	Default value	Description
4	Set	NV	DSCP Urgent	U8	55	
5	Set	NV	DSCP Scheduled	U8	47	
6	Set	NV	DSCP High	U8	43	
7	Set	NV	DSCP Low	U8	31	
8	Set	NV	DSCP Explicit	U8	27	

Services

Service code (hex)	Implemented		Service name	Description
	Class	Instance		

Continues on next page

Service code (hex)	Implemented		Service name	Description
0E	YES	YES	Get Attribute Single	
10	NO	YES	Set Attribute Single	

Port Object

The Port object (Class Code: F4hex) contains information about the Ports related data. It helps determine associated objects involved in the path of communication.

Class attributes

Attribute ID (hex)	Access rule	NV	Attribute name	Data type	Default value	Description
1	Get	NV	Revision	U16	4	
2	Get	NV	Max Instance	U16	4	
3	Get	NV	Number of Instances	U16	3	
8	Get	V	Entry Port	U16		Returns the instance of the Port Object that describes the port through which this request entered the device
9	Get	V	Port Instance Info	Array of structs of	Array of structures containing instance attributes 1 and 2 from each instance	The array is indexed by instance number starting with zero, up to the maximum instance number. The values for instance zero and any non-instantiated instances shall be zero.
			Port Type	U16	Enumerates the type of port	
			Port Number	U16	CIP port number associated with this port	

Continues on next page

10.2 Object dictionary
Continued

Instance attributes

Attribute ID (hex)	Access rule	NV	Attribute name	Data type	Default value	Description
1	Get	NV	Port Type	U16		Enumerates the type of port
2	Get	NV	Port Number	U16		CIP port number associated with this port
3	Get	NV	Logical Link Object	STRUCT of:		
			Path Length	U16		
			Link Path	Padded EPATH		
4	Get	NV	Port Name	SHORT_ STRING		DEV, Public Network or LAN
7	Get	NV	Port Number and Node Address	Padded EPATH	0x1200	
10	Get	NV	Port Routing Capabilities	U32	0x00000000	No routing supported
11	Get	NV	Associated Communication Objects	STRUCT of:		List of communication object instances associated with this Port Object
				U16		Number of entries in array
				ARRAY of STRUCT of		
				U16		Number of 16 bit words in the following path
				Padded EPATH		Logical path segments that identify an associated communication object instance

Services

Service code (hex)	Implemented		Service name	Description
	Class	Instance		
0E	YES	YES	Get Attribute Single	
10	NO	NO	Set Attribute Single	

TCP/IP Interface Object

The TCP/IP Interface object (Class Code: F5hex) contains information about the TCP/IP related data. Most of the data is derived from the controller configuration in RobotStudio or I/O Engineering.

Class attributes

Attribute ID (hex)	Access rule	NV	Attribute name	Data type	Default value	Description
1	Get	NV	Revision	U16	4	
2	Get	NV	Max Instance	U16	4	

Continues on next page

Attribute ID (hex)	Access rule	NV	Attribute name	Data type	Default value	Description
3	Get	NV	Number of Instances	U16	3	

Instance attributes

Attribute ID (hex)	Access rule	NV	Attribute name	Data type	Default value	Description
1	Get	V	Status	U32		
2	Get	NV	Configuration Capability	U32	0x00000020	Hardware configurable
3	Get	NV	Configuration Control	U32	0x00000000	Use statically-assigned IP configuration
4	Get	NV	Physical Link Object	STRUCT of:		No path used when multiple Eth interfaces to same TCP/IP interface.
			Path Size	U16	0	
			Path	Padded EPATH	-	
5	Get	NV	Interface Configuration	STRUCT of:		
			IP Address	U32		
			Network Mask	U32		
			Gateway Address	U32		
			Name Server	U32		
			Name Server 2	U32		
			Domain Name	STRING		
6	Get	NV	Host Name	STRING		
7	Get	NV	Safety Network Number	6 Octets		
13	Set	NV	Encapsulation Inactivity Timeout	U16	120 s	

Services

Service code (hex)	Implemented		Service name	Description
	Class	Instance		
0E	YES	YES	Get Attribute Single	
10	NO	YES	Set Attribute Single	

Ethernet Link Object

The Ethernet Link object (Class Code: F6hex) contains information and settings for the physical interfaces. Device has 4 physical interfaces which through an embedded switch lead to different TCP/IP interface (IP addresses). Each physical interface has one instance of the Ethernet link object.

Continues on next page

Class attributes

Attribute ID (hex)	Access rule	NV	Attribute name	Data type	Default value	Description
1	Get	NV	Revision	U16	4	
2	Get	NV	Max Instance	U16	4	
3	Get	NV	Number of Instances	U16	4	

Instance attributes

Attribute ID (hex)	Access rule	NV	Attribute name	Data type	Default value	Description
1	Get	V	Interface Speed	U32		
2	Get	V	Interface Flags	U32		
3	Get	NV	Physical Address	ARRAY of: 6 U8		
10	Get	NV	Interface Label	SHORT_ STRING		Instance1: DEV Instance2: WAN1 Instance3: LAN
11	Get	NV	Interface Capability	STRUCT of:		Supports auto-negotiate, auto-MDIX, manual Speed/Duplex. Speed/Duplex pairs supported: 100, Half 100, Full 10, Half 10, Full
			Capability Bits	U32	0xE	
			Speed/Duplex Options	STRUCT of:		
			Speed/Duplex Array Count	U8	4	
			Speed/Duplex Array	ARRAY of STRUCT of:		
			Interface Speed	U16		
			Interface Duplex Mode	U8		

Services

Service code (hex)	Implemented		Service name	Description
	Class	Instance		
0E	YES	YES	Get Attribute Single	
10	NO	NO	Set Attribute Single	

LLDP Management Object

The LLDP Management object (Class Code: 0x109) is used to control and configure the LLDP behavior of the device. It allows LLDP to be enabled or disabled on specific Ethernet ports using the LLDP Enable attribute. It also manages how

Continues on next page

frequently LLDP messages are transmitted (msgTxInterval) and determines how long neighboring devices should consider the received LLDP data valid, based on the calculated Time-To-Live (msgTxInterval × msgTxHold).

Instance attributes

Attribute ID (hex)	Access rule	NV	Attribute name	Data type	Default value	Description
1	Set	NV	LLDP Enable	STRUCT of:		
			LLDP Enable Array Length	UINT		Number of bits defined in the LLDP Enable Array.
			LLDP Enable Array	ARRAY of BYTE	Default enabled for all ports.	Array of flags that indicate whether LLDP is enabled or disabled for a port. Bit 0 Global Enable Tx & Rx, Bit 1-N Port Enable Tx
2	Set	NV	msgTxInterval	UINT	30	The interval in seconds at which LLDP frames are sent from the device.
3	Set	NV	msgTxHold	USINT	4	A multiplier of msgTxInterval to determine the value of the TTL TLV (Time-To-Live Type-Length-Value) sent to neighbouring devices.
4	Get	V	LLDP Datastore	UDINT	LLDP Data Table Object	An indication of the retrieval methods for the LLDP Datastore supported by this device Bit: 0 = LLDP Data Table Object 1 = SNMP1 2 = NETCONF YANG
5	Get	V	Last Change			The value of sysUpTime taken the last time any entry (ignoring TTL) in the LLDP Datastore changed.

Services

Service code (hex)	Implemented		Service name	Description
	Class	Instance		
0E	N/A	YES	Get Attribute Single	
10	N/A	YES	Set Attribute Single	

LLDP Data Table Object

The LLDP Data Table object (Class Code: 0x10A) stores information received from neighboring devices that send LLDP messages. Each detected neighbor is represented by a separate instance, containing details such as CIP identity parameters, capabilities, Time to Live value and other data.

Continues on next page

10.2 Object dictionary
Continued

Class attributes

Attribute ID (hex)	Access rule	NV	Attribute name	Data type	Default value	Description
2	Get	V	Max Instance	UINT		
3	Get	V	Number of Instances	UINT		

Instance attributes

Attribute ID (hex)	Access rule	NV	Attribute name	Data type	Default value	Description
1	Get	V	Ethernet Link Instance Number	UINT		The instance number of the Ethernet Link Object that matches the physical Ethernet port where the LLDP frame was received, if known.
2	Get	V	MAC Address	ETH_MAC_ADDR		
3	Get	V	Interface Label	SHORT_STRING		
4	Get	V	Time to Live	UINT		The number of seconds the neighboring information is to be considered valid.
5	Get	V	System Capabilities TLV	STRUCT of:		Set of flags to indicate the system capabilities of the neighboring device, for example if it is a router or a switch.
			System Capabilities	WORD		
			Enabled Capabilities	WORD		
6	Get	V	IPv4 Management Addresses	STRUCT of:		
			Management Address Count	USINT		Number of implemented management addresses.
			Management Address	ARRAY of UDINT		
7	Get	V	CIP Identification	STRUCT of:		The CIP Identification of the neighboring device.
			Vendor ID	UINT		
			Device Type	UINT		
			Product Code	UINT		
			Major Revision	BYTE		
			Minor Revision	USINT		
			CIP Serial Number	UDINT		

Continues on next page

Attribute ID (hex)	Access rule	NV	Attribute name	Data type	Default value	Description
8	Get	V	Additional Ethernet Capabilities	STRUCT of:		Ethernet Preemption Support from the neighboring device.
			Preemption Support	BOOL		
			Preemption Status	BOOL		
			Preemption Active	BOOL		
			Additional Fragment Size	USINT		
9	Get	V	Last Change	UDINT		The value of sysUpTime taken the last time any attribute in this instance changed.

Services

Service code (hex)	Implemented		Service name	Description
	Class	Instance		
0E	YES	YES	Get Attribute Single	
10	NO	YES	Set Attribute Single	

This page is intentionally left blank

Index

ABB Robotics Sweden AB
S-721 71 VÄSTERÅS, Sweden
Telephone +46 10-732 50 00

ABB Robotics Norway AS
Nordlysvegen 7, N-4340 BRYNE, Norway
Box 265, N-4349 BRYNE, Norway
Telephone: +47 22 87 2000

ABB Robotics (Shanghai) Limited
No. 39, 99 Miaoqiao Road,
Pudong New Area,
Shanghai, 201315, China
Telephone: +86 21 6105 6666

ABB Robotics LLC
1250 Brown Road
Auburn Hills, MI 48326
USA
Telephone: +1 248 391 9000

abb.com/robotics

3HAC097985-001, Rev B, en