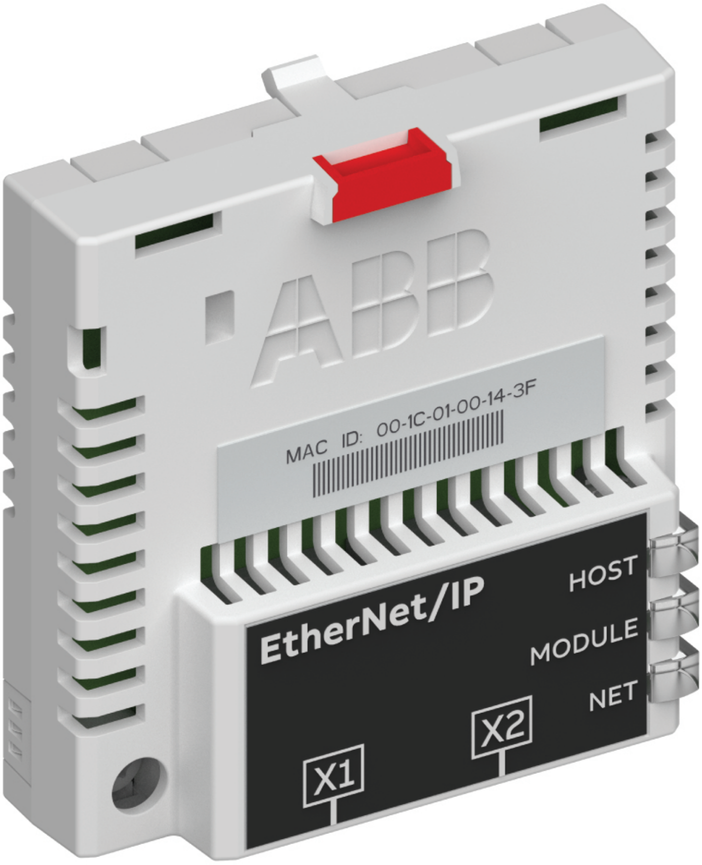
ABB INDUSTRIAL DRIVES

FEIP-21 EtherNet/IP fieldbus adapter module

User's manual



FEIP-21 EtherNet/IP fieldbus adapter module

User's manual

3AXD50000158621 Rev C
EN
Original instructions
EFFECTIVE: 2022-05-25

Table of contents

1 Safety instructions

2 Introduction to the manual

3 Overview of the Ethernet network and the FEIP-21 module

4 Mechanical installation

5 Electrical installation

6 EtherNet/IP – Start-up

7 EtherNet/IP – Communication profiles

8 EtherNet/IP – Communication protocol

15 Appendix C - Replacing FENA-xx module with FEIP-21

16 Appendix D - FEIP-21 configuration web pages

Further information

1

Safety instructions

Contents of this chapter

The chapter describes the warning symbols used in this manual and refers to the safety instructions which you must obey when you install or connect an option module. If you ignore the safety instructions, injury, death or damage can occur. Read this chapter before you start the installation.

Use of warnings and notes

Warnings tell you about conditions which can cause injury or death, or damage to the equipment. They also tell you how to prevent the danger. Notes draw attention to a particular condition or fact, or give information on a subject.

The manual uses these warning symbols:

WARNING!
Electricity warning tells about hazards from electricity which can cause injury or death, or damage to the equipment.

WARNING!
General warning tells about conditions other than those caused by electricity, which can cause injury or death, or damage to the equipment.

WARNING!
Electrostatic sensitive devices warning tells you about the risk of electrostatic discharge which can cause damage to the equipment.

Safety in installation and maintenance

WARNING!
Obey the safety instructions of the drive. If you ignore them, injury or death, or damage to the equipment can occur. If you are not a qualified electrical professional, do not do installation, commissioning or maintenance work.

2

Introduction to the manual

Contents of this chapter

This chapter introduces this manual.

Applicability

This manual applies to the FEIP-21 fieldbus adapter module, revision E or software version 1.20 or later.

Compatibility

Drives

The FEIP-21 fieldbus adapter module is compatible with:

- ACS880 primary control program version 2.51.0.0 and later
- ACS580 standard control program version 2.02.0.1 and later
- ACH580 HVAC control program 2.01.0.4 and later
- ACQ580 pump control program 2.03.0.3 and later
- ACS380 machinery control program version 2.02.0.1 and later
- ACS480 standard control program 2.02.0.3 and later
- ACH480 HVAC control program 2.06.0.2 and later
- ACS560 standard control program 2.08.0.0 and later

Note: Not all compatible drives are listed here. For details of compatibility, check the drive's firmware manual. You can check the current firmware version of the drive with parameter 07.05 Firmware version.

Protocol

The FEIP-21 module is compatible with Ethernet standards IEEE 802.3 and IEE 802.3u and it supports the EtherNet/IP protocol.

All EtherNet/IP clients that support:

- The CIP Networks Library, Volume 1, Common Industrial Protocol (CIP), Edition 3.0 May, 2006
- The CIP Networks Library, Volume 2, EtherNet/IP Adaptation of CIP, Edition 1.2 May, 2006
- Recommended Functionality for EtherNet/IP Devices Version 1.2, Feb., 2006

are compatible with the EtherNet/IP module.

Target audience

This manual is intended for people who plan the installation, install, start up, use and service the module. Before you do work on the module, read this manual and the applicable drive manual that contains the hardware and safety information for the product in question. You are expected to know the fundamentals of electricity, wiring, electrical components, and electrical schematic symbols.

Purpose of the manual

The manual provides information on installing, commissioning and using the FEIP-21 module.

Cybersecurity disclaimer

This product can be connected to and communicate information and data via a network interface. The HTTP protocol, which is used between the commissioning tool (Drive Composer) and the product, is an unsecured protocol. For independent and continuous operation of product such connection via network to commissioning tool is not necessary. However it is Customer's sole responsibility to provide and continuously ensure a secure connection between the product and Customer network or any other network (as the case may be). Customer shall establish and maintain any appropriate measures (such as but not limited to the installation of firewalls, prevention of physical access, 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.

Notwithstanding any other provision to the contrary and regardless of whether the contract is terminated or not, ABB and its affiliates are under no circumstances liable for damages and/or losses related to such security breaches, any unauthorized access, interference, intrusion, leakage and/or theft of data or information.

Terms and abbreviations

Term	Description
Control word	16-bit or 32-bit word from a controller to the controlled device with bit-coded control signals (sometimes called the Command word).

Term	Description
DHCP	Dynamic Host Control Protocol. A protocol for automating the configuration of IP devices. DHCP can be used to automatically assign IP addresses and related network information.
DLR	Device Level Ring. DLR network is a single-fault tolerant ring network topology intended for interconnection of automation devices.
Drive	Frequency converter for controlling AC motors
EDS file	Electronic Datasheet File identifies the properties of the device to the EtherNet/IP client. Each type of drive and application program requires its own EDS file.
EMC	Electromagnetic compatibility
FBA	Fieldbus adapter
FEIP-21	Optional Ethernet adapter module for EtherNet/IP™
MAC address	Media Access Control address
ODVA™	Open DeviceNet Vendor Association. ODVA is an independent organization that promotes interoperability between different manufacturers' EtherNet/IP products. ABB is an Associate Member at ODVA.
PLC	Programmable logic controller
Profile	Adaptation of a communication protocol for a certain application field (for example drives)
SNTP	Simple Network Time Protocol. A protocol to synchronize drive time with the network time server.
Status word	16-bit or 32-bit word from a controlled device to the controller with bit-coded status signals

Related documents

Manual	Code
Drive hardware manuals and guides	
ACS380-04 manuals	9AAK10103A6193
ACS480 manuals	9AKK106930A8739
ACS580-01 manuals	9AKK105713A8085
ACH580-01 manuals	9AKK10103A0587
ACQ580-01 manuals	9AKK106713A2709
ACS580-04 manuals	9AKK106930A9060
ACH580-04 manuals	9AKK106930A9059
ACQ580-04 manuals	9AKK106930A9053
ACS580-07 manuals	9AKK106930A5239
ACH580-07 manuals	9AKK106930A5241
ACQ580-07 manuals	9AKK106930A3150
ACS880-01 manuals	9AKK105408A7004
ACS880-04 manuals	9AKK105713A4819
ACS880-07 (45 to 710 kW) manuals	9AKK105408A8149
ACS880-07 (560 to 2800 kW) manuals	9AKK105713A6663
ACS880-07LC manuals	9AKK107680A9275
ACS880-11 manuals	9AKK106930A9565
ACS880-14 manuals	9AKK107045A8023
ACS880-17 (45 to 400 kW) manuals	9AKK106930A3466
ACS880-17 (160 to 3200 kW) manuals	9AKK106354A1499

Manual	**Code**
ACS880-17LC manuals	9AKK107492A4721
ACS880-31 manuals	9AKK106930A9564
ACS880-34 manuals	9AKK107045A8025
ACS880-37 (45 to 400 kW) manuals	9AKK106930A3467
ACS880-37 (160 to 3200 kW) manuals	9AKK106354A1500
ACS880-37LC manuals	9AKK107492A4722
Option manuals and guides	
FEIP-21 EtherNet/IP fieldbus adapter module user’s manual	3AXD50000158621
FEIP-21 EtherNet/IP fieldbus adapter module quick installation and start-up guide	3AXD50000158584

The links above contain lists of documents. You can find manuals and other product documents in PDF format on the Internet. See ABB Document Library. For manuals not available in the Document library, contact your local ABB representative.

FEIP-21 EtherNet/IP fieldbus adapter module User's manual

Fieldbus connectivity web page

3

Overview of the Ethernet network and the FEIP-21 module

Contents of this chapter

This chapter contains a short description of the Ethernet network and the topology supported by the FEIP-21 adapter module.

Ethernet network

Ethernet standards support a variety of physical media (coaxial cable, twisted pair, fiber optics) and topologies (bus and star).

The FEIP-21 module supports twisted pair as the physical media

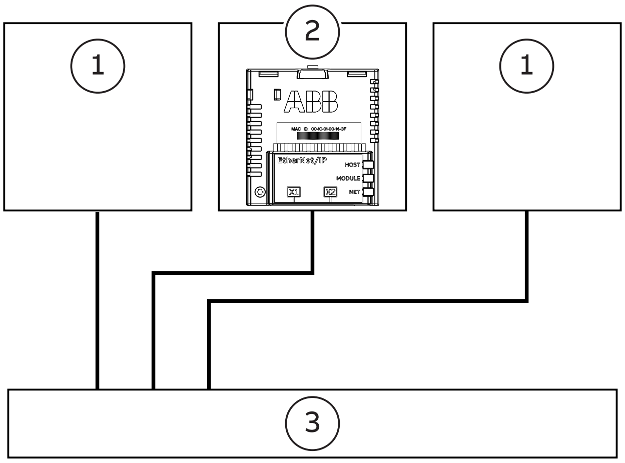
FEIP-21 supports star topology, daisy chain topology and ring topology (Device Level Ring, DLR)

The maximum length for an Ethernet segment on twisted pair media is 100 meters. All twisted pair media between the Ethernet node and the switch or router must be shorter than 100 meters, including media within patch panels.

Example topology of the Ethernet link

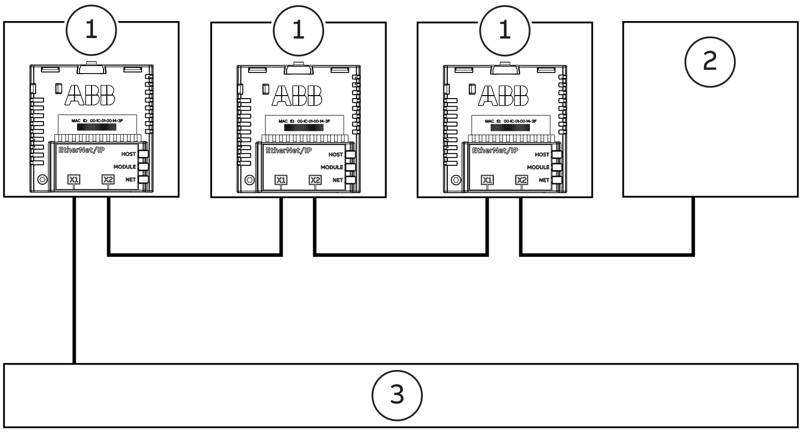
The figures below show example topologies for an Ethernet network with FEIP-21.

Star topology



1	Other device
2	ABB drive
3	Switch or router

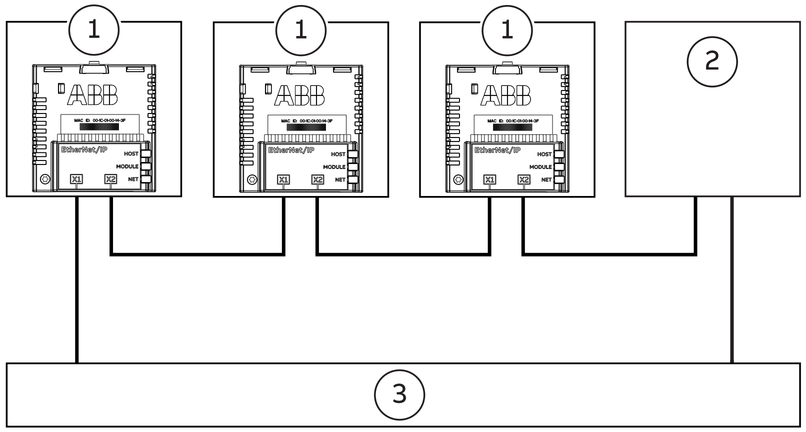
Daisy chain topology using integrated Ethernet switch



1	ABB drive
2	Other device
3	Network master device

Ring topology

For DLR configuration, see Configuring DLR topology for FEIP-21 (page 51).



1	ABB drive
2	Other device
3	Network master device

FEIP-21 module overview

The FEIP-21 adapter module is a plug-in device for ABB drives which enables the connection of the drive to an EtherNet/IP network.

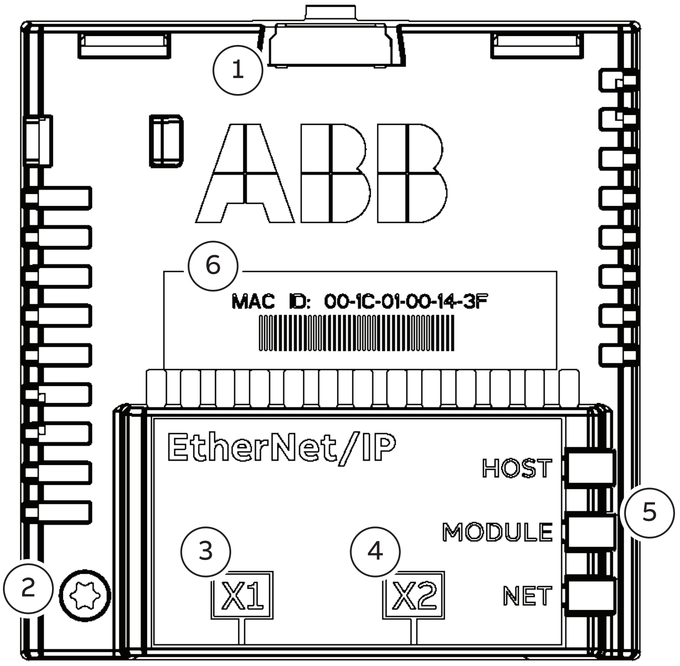
Through the adapter module you can:

- give control commands to the drive (for example, Start, Stop, Run enable)
- feed a motor speed or torque reference to the drive
- give a process actual value or a process reference to the PID controller of the drive
- read status information and actual values from the drive
- reset a drive fault
- read/write parameters of the drive
- connect Drive composer pro tool

The adapter module supports 10 Mbit/s and 100 Mbit/s data transfer rates and automatically detects the data transfer rate used in the network.

The adapter module is installed into an option slot on the drive control unit. See the drive manuals for module placement options.

FEIP-21 layout



No.	Description
1	Lock
2	Mounting screw (Grounding screw)
3	X1 connector to Ethernet
4	X2 connector for chaining another module
5	Diagnostic LEDs
6	MAC address

4

Mechanical installation

Contents of this chapter

This chapter contains a delivery checklist and instructions on installing the module.

Necessary tools and instructions

You will need a Torx TX10 screwdriver to secure the FEIP-21 module to the drive. See also the drive hardware manual.

Unpacking and examining the delivery

1. Open the option package.
2. Make sure that the package contains:
   - fieldbus module, type FEIP-21
   - quick guide.
3. Make sure that there are no signs of damage.

Before you start

Install the module to a free option slot on the drive control unit. Plastic pins, a lock and one screw hold the module in place. The screw also makes an electrical connection between the module and drive frame for cable shield grounding.

Note: Do not install the FEIP-21 module on the FEA-03 F-series extension adapter.

Note: Do not install the FEIP-21 when the drive is powered up.

When the module is installed, it makes the signal and power connection to the drive through a 20-pin connector.

Installing option modules

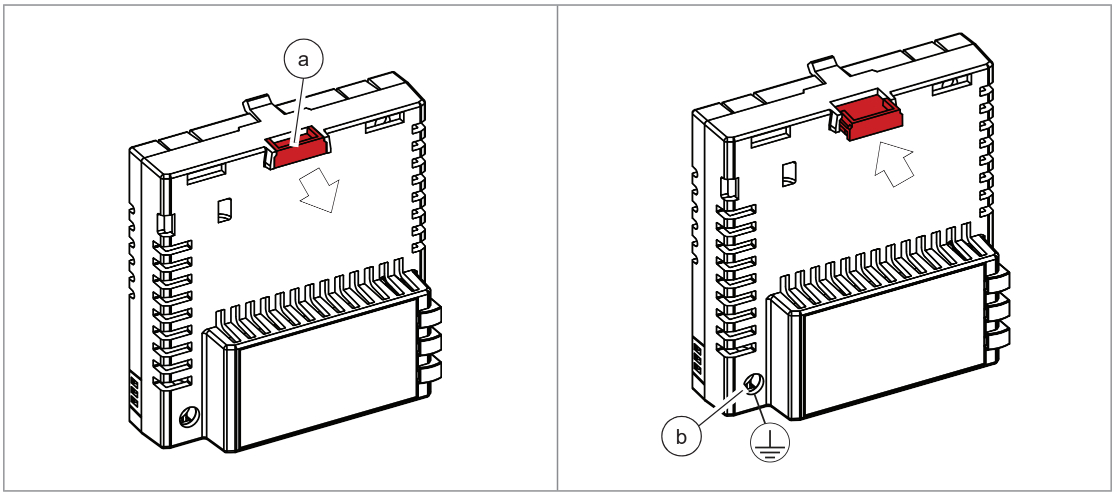
WARNING!
Obey the safety instructions of the drive. If you ignore them, injury or death, or damage to the equipment can occur.

Pay attention to the free space required by the cabling or terminals coming to the option modules.

1. Repeat the steps described in the electrical safety precautions of the drive.
2. Pull out the lock (a).
3. Install the module to a free option module slot on the control unit.
4. Push in the lock (a).
5. Tighten the grounding screw (b) to a torque of 0.8 N·m (7 lbf·in).

   Note: The screw tightens the connections and grounds the module. It is essential for fulfilling the EMC requirements and for proper operation of the module.

   WARNING!
   Do not use excessive force, or leave the screw too loose. Over-tightening can damage the screw or module. A loose screw can cause an operation failure.



6. Connect the wiring to the module. Obey the instructions given in this manual.

5

Electrical installation

Contents of this chapter

This chapter contains general cabling instructions and instructions on connecting the FEIP-21 module to the Ethernet network and the drive.

Necessary tools and instructions

See the drive hardware manual.

General cabling instructions

- Arrange the bus cables as far away from the motor cables as possible.
- Avoid parallel runs.
- Use bushings at cable entries.

Connecting the FEIP-21 to the Ethernet network

WARNING!
Obey the safety instructions of the drive. If you ignore them, injury or death, or damage to the equipment can occur.

Choose correct cable for your application, CAT5e and CAT6 cables are recommended for industrial applications. ABB recommends to use double shielded twisted pair cables, eg. "SF/FTP".

6

EtherNet/IP – Start-up

Contents of this chapter

This chapter contains:

- information on configuring the drive for operation with the adapter module
- drive-specific instructions on starting up the drive with the adapter module
- examples of configuring the client for communication with the adapter module.

Warnings

WARNING!
Obey the safety instructions given in this manual and the drive documentation.

Drive configuration

The information in this section applies to all drive types compatible with the adapter module, unless otherwise stated.

EtherNet/IP connection configuration

After the adapter module has been mechanically and electrically installed according to the instructions in chapters Mechanical installation and Electrical installation, you must prepare the drive for communication with the module.

Normally, you must adjust a parameter to activate the communication. For detailed procedure of activating the module for EtherNet/IP communication with the drive, see Starting up fieldbus communication for the drive (page 37).

Once communication between the drive and the adapter module has been established, several configuration parameters are copied to the drive. These parameters are shown in the tables below and must be checked first and adjusted where necessary. You can adjust the parameters via a drive control panel or a PC tool.

Note: Not all drives display descriptive names for the configuration parameters. To help you identify the parameters in different drives, the names displayed by each drive are given in gray boxes in the tables.

Note: The new parameter settings take effect only when you power up the module the next time or when you activate the fieldbus adapter refresh parameter.

FEIP-21 configuration parameters – group A (group 1)

Note: The actual parameter group number depends on the drive type. Group A (group 1) corresponds to:

- parameter group 51 in ACS380, ACS480, ACH580, ACQ580 and ACS580
- parameter group 51/54 (or 151/154 in some variants) in ACS880 if the adapter is installed as fieldbus adapter A/B.

No.	Name/ Value	Description	Default
01	FBA type	Read-only. Shows the fieldbus adapter type as detected by the drive. The value cannot be adjusted by the user. If the value is 0 = None, the communication between the drive and the module has not been established.	2222 = EtherNet/IP
02	Protocol/Profile	Selects the application protocol and communication profile for the network connection. The selections available for EtherNet/IP communication are listed below.	101 = EIP ABB Pro
	100 = EIP AC/DC	EtherNet/IP protocol: ODVA AC/DC drive profile	
	101 = EIP ABB Pro	EtherNet/IP protocol: ABB Drives profile	
	102 = EIP T16	EtherNet/IP protocol: Transparent 16-bit profile	
	103 = EIP T32	EtherNet/IP protocol: Transparent 32-bit profile	
03	Commrate	Sets the bit rate for the Ethernet interface. In FEIP-21 this parameter is used for configuring port 1. For configuring port 2, see parameter 14 Commrate port 2 (page 28).	0 = Auto
	0 = Auto	Autonegotiate	
	1 = 100 Mbps FD	100 Mbps, full duplex	
	2 = 100 Mbps HD	100 Mbps, half duplex	
	3 = 10 Mbps FD	10 Mbps, full duplex	
	4 = 10 Mbps HD	10 Mbps, half duplex	
04	IP configuration	Sets the method for configuring the IP address, subnet mask and gateway address for the adapter module.	1 = Dyn IP DHCP
	0 = Static IP	Configuration will be obtained from parameters 05…13.	

<table>
<tr><th>No.</th><th>Name/ Value</th><th>Description</th><th>Default</th></tr>
<tr><td></td><td>1 = Dyn IP DHCP</td><td>Configuration will be obtained via DHCP.</td><td></td></tr>
<tr><td>05</td><td>IP address 1</td><td>An IP address is assigned to each IP node on a network. An IP address is a 32-bit number that is typically represented in “dotted decimal” notation consisting of four decimal integers, on the range 0…255, separated by periods. Each integer represents the value of one octet (8-bits) in the IP address. Parameters 05…08 define the four octets of the IP address.</td><td>0</td></tr>
<tr><td></td><td>0 … 255</td><td>IP address</td><td></td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>...</td></tr>
<tr><td>08</td><td>IP address 4</td><td>See parameter 05 IP address 1 (page 27).</td><td>0</td></tr>
<tr><td></td><td>0 … 255</td><td>IP address</td><td></td></tr>
<tr><td>09</td><td>Subnet CIDR</td><td>Subnet masks are used for splitting networks into smaller networks called subnets. A subnet mask is a 32-bit binary number that splits the IP address into a network address and host address.
Subnet masks are typically represented in either dotted decimal notation or the more compact CIDR notation, as shown in the table below.</td><td>0</td></tr>
<tr><td></td><td colspan="3">
<table>
<tr><th>CIDR</th><th>Dotted decimal</th><th>CIDR</th><th>Dotted decimal</th></tr>
<tr><td>31</td><td>255.255.255.254</td><td>15</td><td>255.254.0.0</td></tr>
<tr><td>30</td><td>255.255.255.252</td><td>14</td><td>255.252.0.0</td></tr>
<tr><td>29</td><td>255.255.255.248</td><td>13</td><td>255.248.0.0</td></tr>
<tr><td>28</td><td>255.255.255.240</td><td>12</td><td>255.240.0.0</td></tr>
<tr><td>27</td><td>255.255.255.224</td><td>11</td><td>255.224.0.0</td></tr>
<tr><td>26</td><td>255.255.255.192</td><td>10</td><td>255.192.0.0</td></tr>
<tr><td>25</td><td>255.255.255.128</td><td>9</td><td>255.128.0.0</td></tr>
<tr><td>24</td><td>255.255.255.0</td><td>8</td><td>255.0.0.0</td></tr>
<tr><td>23</td><td>255.255.254.0</td><td>7</td><td>254.0.0.0</td></tr>
<tr><td>22</td><td>255.255.252.0</td><td>6</td><td>252.0.0.0</td></tr>
<tr><td>21</td><td>255.255.248.0</td><td>5</td><td>248.0.0.0</td></tr>
<tr><td>20</td><td>255.255.240.0</td><td>4</td><td>240.0.0.0</td></tr>
<tr><td>19</td><td>255.255.224.0</td><td>3</td><td>224.0.0.0</td></tr>
<tr><td>18</td><td>255.255.192.0</td><td>2</td><td>192.0.0.0</td></tr>
<tr><td>17</td><td>255.255.128.0</td><td>1</td><td>128.0.0.0</td></tr>
<tr><td>16</td><td>255.255.0.0</td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td></td><td>1 ... 31</td><td>Subnet mask in CIDR notation</td><td></td></tr>
<tr><td>10</td><td>GW address 1</td><td>IP gateways connect individual physical IP subnets into a unified IP network. When an IP node needs to communicate with an IP node on another subnet, the IP node sends the data to the IP gateway for forwarding. Parameters 10…13 define the four octets of the gateway address.</td><td>0</td></tr>
<tr><td></td><td>0 ... 255</td><td>GW address</td><td></td></tr>
</table>

<table>
<tr><th>No.</th><th>Name/ Value</th><th>Description</th><th>Default</th></tr>
<tr><td>...</td><td>...</td><td>...</td><td>...</td></tr>
<tr><td>13</td><td>GW address 4</td><td>See parameter 10 GW address 1 (page 27).</td><td>0</td></tr>
<tr><td></td><td>0 ... 255</td><td>GW address</td><td></td></tr>
<tr><td>14</td><td>Commrate port 2</td><td>Sets the bit rate for the Ethernet port 2. This parameter is used only with FEIP-21.</td><td>0 = Auto</td></tr>
<tr><td></td><td>0 = Auto</td><td>Autonegotiate</td><td></td></tr>
<tr><td></td><td>1 = 100 Mbps FD</td><td>100 Mbps, full duplex</td><td></td></tr>
<tr><td></td><td>2 = 100 Mbps HD</td><td>100 Mbps, half duplex</td><td></td></tr>
<tr><td></td><td>3 = 10 Mbps FD</td><td>10 Mbps, full duplex</td><td></td></tr>
<tr><td></td><td>4 = 10 Mbps HD</td><td>10 Mbps, half duplex</td><td></td></tr>
<tr><td>15</td><td>Service configuration</td><td>Disables services that are not required.
Each service is represented by a single bit. Bit 0, Lock configuration, can be used to prevent accidental changing of this parameter.
By default, all services are enabled and configuration is unlocked.</td><td></td></tr>
<tr><td></td><td colspan="3">
<table>
<tr><th>Bit</th><th>Name</th><th>Information</th></tr>
<tr><td>0</td><td>Lock configuration</td><td>Changing this bit to one will lock service configuration and no bit can be changed. Only resetting the fieldbus configuration will unlock the parameter. To reset the fieldbus configuration, choose "Reset all fieldbus settings" or "Clear all" in parameter 96.06 Parameter restore.</td></tr>
<tr><td>1</td><td>Disable IP config tool</td><td>When this bit is set, access from ABB IP Configuration tool is prevented.</td></tr>
<tr><td>2</td><td>Disable ETH tool network</td><td>When this bit is set, access from Ethernet tool network (eg, ABB Drive Composer tool) is prevented.</td></tr>
<tr><td>3</td><td>Disable ping response</td><td>When this bit is set, response to ICMP (ping) message is prevented.</td></tr>
<tr><td>4</td><td>Unsecured ETH tool network</td><td>When this bit is set, access from Ethernet tool network is unsecured.
<b>Note:</b> Drive Composer pro before V2.7 supports unsecured communication only.</td></tr>
<tr><td>5</td><td>Disable configuring web pages</td><td>When this bit is set, access to web pages is disabled.</td></tr>
</table>
</td></tr>
<tr><td></td><td>0000b…1111b</td><td>Service configuration</td><td></td></tr>
</table>

No.	Name/ Value	Description	Default
16	Module emulation	Enables the emulation of and old Ethernet/IP module (RETA-01) and selects the emulated old drive type. Select the emulated product code from the list below on basis of the drive and Ethernet module type to be emulated. FEIP-21 module reports the selected code to the PLC. It is also possible to use your own code. See section Emulation modes (page 35).	0
	0	No emulation. The FEIP-21 module reports the actual product code to the PLC.	
	202	ACH550-xx with RETA-01	
	201	ACS550-xx with RETA-01	
	171	ACS800 (AMAL) with RETA-01	
	149	ACS800 (Braking) with RETA-01	
	121	ACS800 (Centrifuge) with RETA-01	
	111	ACS800 (Crane) with RETA-01	
	112	ACS800 (CraneControl) with RETA-01	
	141	ACS800 (FCBTemplate) with RETA-01	
	210	ACS800 (MotionControl) with RETA-01	
	191	ACS800 (MultiBlock) with RETA-01	
	101	ACS800 (Standard) with RETA-01	
	151	ACS800 (StandardPMSM) with RETA-01	
	105	ACS800 (System) with RETA-01	
	156	ACS800 (SystemPMSM) with RETA-01	
	243	ACS800 (Winder) with RETA-01	
17	Revision emulation	Enables the user to define the firmware major and minor revisions that FEIP-21 will emulate. Revision emulation can be used to pass the major/minor revision check if they are set in the Ethernet IP scanner configuration. **Note:** Emulating the revision number might affect system behavior. By changing this parameter you accept the risk. See section Configuring revision emulation (page 36).	0 = Use the actual revision of FEIP-21
	0	Emulation is not used. Actual revision of FEIP-21 will be used.	
	0x1-0x9999	Emulate revision in hexadecimal. High byte sets the major revision, low byte sets the minor revision.	
18	Reserved	This parameter is not used.	

<table>
<tr><th>No.</th><th>Name/ Value</th><th>Description</th><th>Default</th></tr>
<tr><td>19</td><td>T16 scale</td><td>Defines the scaling for reference 1 and actual 1 with Transparent 16 profile. (Protocol.Profile = EIP T16)
Scaling also depends on the selected Reference type on 50.04 FBA A Ref 1 type and 50.34 FBA B Ref 1 type and 50.07 and 50.37 for the actual 1.
Ref type = Transparent
FBA_A/B_Ref1 = Ref1_from_PLC * (T16_Scale + 1)
Ref type = General
FBA_A/B_Ref1 = Ref1_from_PLC * (T16_Scale + 1) / 100</td><td>99</td></tr>
<tr><td></td><td>0 ... 65535</td><td>Reference multiplier/actual value divisor</td><td></td></tr>
<tr><td>20</td><td>Control timeout</td><td>Defines the control timeout value.
The EtherNet/IP protocol specifies connection timeout for I/O messaging (Class 1) and Connected explicit messaging (Class 3), but not Unconnected explicit messaging.
This parameter provides a timeout for Unconnected explicit messaging and for instances of Connected explicit messaging (Class 3), where the client breaks the connection in between requests.</td><td>0</td></tr>
<tr><td></td><td colspan="3">
<table>
<tr><th>Connection type</th><th>Control timeout</th><th>Timeout source</th></tr>
<tr><td>I/O messaging (Class 1)</td><td>0 … 65535</td><td>(Requested Packet Interval) X (Connection Timeout Multiplier)
<b>Note:</b> Timeout behavior may be modified by Watchdog Timeout Action attribute of Connection object.</td></tr>
<tr><td>Connected explicit messaging (Class 3)</td><td>0</td><td>(Requested Packet Interval) X (Connection Timeout Multiplier)
<b>Note:</b> Timeout behavior may be modified by Watchdog Timeout Action attribute of Connection object.</td></tr>
<tr><td></td><td>1 ... 65534</td><td>100ms X (Control Timeout Value) since last Control Event</td></tr>
<tr><td></td><td>65535</td><td>Never Timeout</td></tr>
<tr><td>Unconnected explicit messaging</td><td>0</td><td>Always Timeout
Control Timeout must be greater than zero to control drive with Unconnected Explicit Messaging.</td></tr>
<tr><td></td><td>1 ... 65534</td><td>100ms X (Control Timeout Value) since last Control Event</td></tr>
<tr><td></td><td>65535</td><td>Never Timeout</td></tr>
</table>
</td></tr>
<tr><td></td><td></td><td>Control timeout events:
• Write of an output assembly object instance
• Write of control bits (Run1, Run2, NetCtrl, NetRef and FaultReset)
• Write Speed Reference
• Write Torque Reference
• Reset Control Supervisor object
• Write Force Fault via Control Supervisor object
If a timeout occurs, the adapter module signals the drive that communication with the client has been lost. The drive configuration then determines how to respond.
Example: If the timeout is 250 ms and the drive is configured to fault on a communication failure with a delay of 500 ms, then the drive will fault 750 ms after communications is lost.</td><td></td></tr>
<tr><td></td><td>0 ... 65535</td><td>Control timeout value</td><td></td></tr>
</table>

No.	Name/ Value	Description	Default
21	Idle action	I/O connections may include a Run/Idle notification. This parameter determines the action the drive takes in response to an Idle notification.	0 = Off-line
	0 = Off-line	In the event of an Idle notification, the adapter module signals the drive that communication with the client has been lost. The drive configuration then determines how to respond. Example: If the timeout is 250 ms and the drive is configured to fault on a communication failure with a delay of 500 ms, then the drive will fault 750 ms after communications is lost.	
	1 = On-line	In the event of an Idle notification, the drive will continue to operate using the last command and references received.	
22	Stop function	Determines how the motor is to be stopped when a stop command is received via EtherNet/IP. This parameter only applies to the ODVA AC/DC drive profile.	0 = Ramp
	0 = Ramp	The motor decelerates along the active deceleration ramp.	
	1 = Coast	The motor comes to a stop by coasting.	
23	Speed scale	This parameter only applies to the ODVA AC/DC drive profile. The units of reference and actual speeds for the ODVA AC/DC drive profile are given by the formula below. Speed unit = RPM × $2^{(-1 \times \text{ODVA speed scale value})}$ **Note:** While a wide range of resolutions may be configured, the actual performance is limited to the performance capabilities of the drive. The table below shows the how the drive ODVA speed scale parameter values correspond to the ODVA speed scale units.	128

ODVA speed scale value[1]	Speed scale value of drive parameter[2]	Unit
-5	123	32 RPM
-4	124	16 RPM
-3	125	8 RPM
-2	126	4 RPM
-1	127	2 RPM
0 (default)	128	1 RPM
1	129	0.5 RPM
2	130	0.25 RPM
3	131	0.125 RPM
4	132	0.0625 RPM
5	133	0.03125 RPM

[1] Use the ODVA speed scale value when reading/writing parameter Speed scale via AC/DC-drive object, class 2Ah. When written via the AC/DC drive object, the new value takes effect immediately.

[2] Use the speed scale value of the drive parameter when reading/writing parameter Speed scale via the drive control panel, Drive parameter object, class 90h and Fieldbus configuration object, class 91h. When written via these methods, the new value takes effect after the drive is repowered or a “Fieldbus Adapter Parameter refresh” is given.

	0 … 255	Speed scale value of drive parameter	

No.	Name/ Value	Description	Default
24	Torque scale	This parameter only applies to the ODVA AC/DC drive profile. The units of reference and actual torques for the ODVA AC/DC drive profile are given by the formula below. Torque unit = N·m × $2^{(-1 \times \text{ODVA torque scale})}$ where: (N·m = Newton × Meter) **Note:** While a wide range of resolutions may be configured, the actual performance is limited to the performance capabilities of the drive. The table below shows how the drive ODVA torque scale parameter values correspond to the ODVA torque scale units.	128

ODVA torque scale value[1]	Torque scale value of drive parameter[2]	Unit
-5	123	32 RPM
-4	124	16 RPM
-3	125	8 RPM
-2	126	4 RPM
-1	127	2 RPM
0 (default)	128	1 RPM
1	129	0.5 RPM
2	130	0.25 RPM
3	131	0.125 RPM
4	132	0.0625 RPM
5	133	0.03125 RPM

[1] Use the ODVA torque scale value when reading/writing parameter Torque scale via AC/DC-drive object, class 2Ah. When written via the AC/DC drive object, the new value takes effect immediately.

[2] Use the torque scale value of the drive parameter when reading/writing parameter Torque scale via the drive control panel, Drive parameter object, class 90h and Fieldbus configuration object, class 91h. When written via these methods, the new value takes effect after the drive is repowered or a “Fieldbus Adapter Parameter refresh” is given.

No.	Name/ Value	Description	Default
	0 … 255	Torque scale value of drive parameter	
25 … 26	Reserved		
27	FBA A/B par refresh	Validates any changed adapter module configuration parameter settings. After refreshing, the value reverts automatically to 0 = Done. **Note:** This parameter cannot be changed while the drive is running.	0 = Done
	0 = Done	Refreshing done	
	1 = Refresh	Refreshing	

No.	Name/ Value	Description	Default

28	FBA A/B par table ver	Read-only. Displays the parameter table revision of the fieldbus adapter module mapping file stored in the memory of the drive, in format xyz, where
		x = major revision number
		y = minor revision number
		z = correction number
		OR
		in format axyz, where
		a = major revision number
		xy = minor revision numbers
		z = correction number or letter.
			N/A

		Parameter table revision	
29	FBA A/B drive type code	Read-only. Displays the drive type code of the fieldbus adapter module mapping file stored in the memory of the drive.	N/A
		Drive type code of the fieldbus adapter module mapping file	
30	FBA A/B mapping file ver	Read-only. Displays the fieldbus adapter module mapping file revision stored in the memory of the drive in decimal format.	N/A
		Mapping file revision	
31	D2FBA A/B comm status	Read-only. Displays the status of the fieldbus adapter module communication. **Note:** The value names may vary by drive.	0 = Idle or 4 = Offline
	0 = Idle	Adapter is not configured.	
	1 = Exec.init	Adapter is initializing.	
	2 = Time out	A timeout has occurred in the communication between the adapter and the drive.	
	3 = Conf.err	Adapter configuration error: The major or minor revision code of the common program revision in the fieldbus adapter module is not the revision required by the module or mapping file upload has failed more than three times.	
	4 = Off-line	Adapter is off-line.	
	5 = On-line	Adapter is on-line.	
	6 = Reset	Adapter is performing a hardware reset.	
32	FBA A/B comm SW ver	Read-only. Displays firmware patch and build number of the adapter module in the xxyy format, where: xx = patch number yy = build number Example: If the firmware version (<major>.<minor>.<patch>.<build>) is 3.10.200.13, the value C80D is displayed. If the version is 3.10.0.0, the value 0 is displayed. See also parameter 33 FBA A/B appl SW ver (page 34).	N/A

No.	Name/ Value	Description	Default
33	FBA A/B appl SW ver	Read-only. Displays firmware version of the adapter module in xxyy format, where: xx = major revision number yy = minor revision number Example: If the firmware version (<major>.<minor>.<patch>.<build>) is 3.10.200.13 or 3.10.0.0, the value 310 is displayed. See also parameter 32 FBA A/B comm SW ver (page 33).	N/A

FEIP-21 configuration parameters – group B (group 2)

Note: The actual parameter group number depends on the drive type. Group B (group 2) corresponds to:

- parameter group 53 in ACS380, ACS480, ACH580, ACQ580 and ACS580
- parameter group is typically 53/56 (153/156 in some variants) in ACS880 if the adapter is installed as fieldbus adapter A/B.

<table>
<tr><th>No.[1)]</th><th>Name/ Value</th><th>Description</th><th>Default</th></tr>
<tr><td>01</td><td>FBA A/B data out 1 (client to drive)</td><td>In output assembly instances that include drive parameters, this parameter specifies which parameter’s value will be placed in location Data out 1 value received by the drive from the EtherNet/IP client.
The content is defined by a decimal number in the range of 0 to 9999 as follows:
<table>
<tr><td>0</td><td>Not used</td></tr>
<tr><td>1 … 99</td><td>Virtual address area of drive control. Not used when the EtherNet/IP protocol is used.</td></tr>
<tr><td>101 … 9999</td><td>Parameter area of the drive</td></tr>
</table></td><td>0 = None</td></tr>
<tr><td></td><td>0 = None</td><td>Not used</td><td></td></tr>
<tr><td></td><td>101 … 9999</td><td>Parameter index with format xxyy, where
• xx is the parameter group number (1…99)
• yy is the parameter number index within that group (01…99).
<b>Note:</b> In ACS480, ACS580 and ACS880, choose Other to display a list of mappable drive parameters.</td><td></td></tr>
<tr><td>02 … 10</td><td>Data out 2 … Data out 10</td><td>See parameter 01 FBA A/B data out 1 (client to drive) (page 34).</td><td>0 = None</td></tr>
</table>

[1)] The number of parameters in this group may vary by drive type and drive firmware.

FEIP-21 configuration parameters – group C (group 3)

Note: The actual parameter group number depends on the drive type. Group C (group 3) corresponds to:

- parameter group 52 in ACS380, ACS480, ACH580, ACQ580 and ACS580
- parameter group is typically 52/55 (152/155 in some variants) in ACS880 and ACS880-M04 if the adapter is installed as fieldbus adapter A/B.

<table>
<tr><th>No.[1]</th><th>Name/ Value</th><th>Description</th><th>Default</th></tr>
<tr><td>01</td><td>FBA A/B data in1 (drive to client)</td><td>In input assembly instances that include drive parameters, this parameter specifies which parameter's value will be placed in location Data in 1 value sent by the drive to the EtherNet/IP client.
The content is defined by a decimal number in the range of 0 to 9999 as follows:<table><tr><td>0</td><td>Not used</td></tr><tr><td>1 ... 99</td><td>Virtual address area of drive control. Not used when the EtherNet/IP protocol is used.</td></tr><tr><td>101 ... 9999</td><td>Parameter area of the drive</td></tr></table></td><td>0 = None</td></tr>
<tr><td></td><td>0 = None</td><td>Not used</td><td></td></tr>
<tr><td></td><td>101 … 9999</td><td>Parameter index with format xxyy, where
• xx is the parameter group number (1…99)
• yy is the parameter number index within that group (01…99).
<b>Note:</b> In ACS480, ACS580 and ACS880, choose Other to display a list of mappable drive parameters.</td><td></td></tr>
<tr><td>02 … 10</td><td>Data in 2 … Data in 10</td><td>See parameter 01 FBA A/B data in1 (drive to client) (page 35).</td><td>0 = None</td></tr>
</table>

[1] The number of parameters in this group may vary by drive type and drive firmware.

Control locations

ABB drives can receive control information from multiple sources including digital inputs, analog inputs, the drive control panel and a fieldbus adapter module. ABB drives allow the user to separately determine the source for each type of control information (Start, Stop, Direction, Reference, Fault reset, etc.).

To give the fieldbus client the most complete control over the drive, you must select the adapter module as the source of this information. The drive-specific parameter setting examples below contain the drive control parameters relevant in the examples. For a complete parameter list, see the drive documentation.

Emulation modes

Emulation mode can be used in the following cases:

- when replacing an older drive with a new drive
- where the PLC program is configured to use RETA-01 and the drive is equipped with FEIP-21
- where the PLC program is configured to check device firmware major and minor revisions and FEIP-21 has different firmware revisions but the device must still be used.

FEIP-21 supports emulation of RETA-01 module and emulation of FEIP module software revisions (major and minor).

RETA-01 compatibility mode changes the assembly of FEIP-21 and emulates the product code of the selected drive, so that an old PLC configuration done for an old drive type with RETA-01 module can accept FEIP-01 module and new drive type connection without modifications.

Note: RETA-01 compatibility mode does not emulate the firmware version of RETA-01. For firmware version emulation, see Revision emulation (page 36).

Revision emulation

Revision emulation allows the user to define the firmware major and minor revisions which FEIP-21 will emulate, so that the PLC accepts the connection when replacing an ethernet adapter module (eg. FEIP-01, RETA-01 or FENA-01 or -11) with a FEIP module with different firmware revisions. If the PLC has major and minor revision checks disabled, you do not need to use revision emulation even if the replacement part revisions do not match.

Always verify correct drive operation when you use the emulation. Emulated revision might behave differently than the product used before.

Note: Revision emulation does not emulate other functions.

Configuring revision emulation

To emulate FENA-11/21 firmware revision:

1. Set parameter 51.17 Revision emulation in hexadecimal to match the FENA-11/21 label SW information or parameter 51.33 FBA A appl FW value.
    - Example: when the label states “SW: 3.25” or parameter 51.33 FBA A appl FW is set to 0x0325, set parameter 51.17 Revision emulation to 0x0325.

To emulate RETA-01 firmware revision:

1. Convert the RETA-01 label APPL information or parameter 51.33 FBA APPL FW REV value from decimal to hexadecimal separately for the major and minor revision numbers.
    - Example: when the label states “APPL 2.12” or parameter 51.33 FBA APPL FW REV is set to 0x0212, convert major revision 2 to hex 0x02 and minor revision 12 to 0x0C.
2. Combine the hexadecimal values (major revision 0x02 and minor revision 0x0C = 0x020C) and set the value to parameter 51.17 Revision emulation.

RETA-01 compatible assemblies

Profile	Assembly types
ODVA	• 20 - Basic Speed Control output assembly (Control Word, Set Speed) • 70 - Basic Speed Control input assembly (Status Word, Actual Speed) • 21 - Extended Speed Control assembly output (Extended Control Word, Set Speed) • 71 - Extended Speed Control assembly input (Extended Status Word, Actual Speed) • 121 - Extended Speed Control plus Drive parameters output assembly (Extended Control Word, Set Speed, dynamic mappings) • 171 - Extended Speed Control plus Drive parameters input assembly (Extended Status Word, Actual Speed, dynamic mappings)
ABB Drives profile	• 100 - Output ABB Drive Control assembly (Control Word, Set Speed) • 101 - Input ABB Drive Control assembly (Status Word, Actual Speed) • 102 - Output User Specific Control assembly (dynamic) • 103 - Input User Specific Control assembly (dynamic)

Configuring RETA-01 emulation mode

To configure RETA-01 emulation mode, follow the below steps:

1. Select a profile based on the used assemblies with parameter 51.02 Protocol Profile.
   - Select EIP AC/DC, if assemblies 20, 70, 21, 71, 121 or 171 are used in the RETA-01.
   - Select EIP ABB Pro, if assemblies 100, 101, 102 or 103 are used in the RETA-01.
2. Select the desired product code in parameter 51.16 Module emulation.
3. Refresh the parameters to configure other parameters needed for Ethernet/IP protocol.

Configuring data in and data out mapping parameters

For assemblies 20, 70, 21, 71, 100, or 101, the order of parameters is predefined. For dynamic assemblies 102, 103, 121, or 171, the assembly content can be configured. You must refresh the parameters after the mapping parameters are ready.

For more information, see RETA-01 Ethernet Adapter Module User's Manual (3AFE64539736 [English]).

Note: FEIP-21 determines the length of dynamic assemblies by refreshing the parameters. The connection from host does not succeed before the host and FEIP-21 have matching lengths of assemblies.

Note: RETA-01 compatibility mode cannot emulate drive parameters. The compatible parameter data is required if parameters other than Control word, Status word, Speed reference, and Speed actual are needed.

Fieldbus configuration object, class 91h

The fieldbus configuration object allows you to configure Data out and Date in configuration over Ethernet/IP.

- Use instance #1 for Data out parameters
- Use instance #2 for Data in parameters.

For more information, see section Starting up fieldbus communication for the drive.

Starting up fieldbus communication for the drive

1. Power up the drive.
2. Enable the communication between the adapter module and the drive by selecting the correct slot number in parameter 50.01 FBA A enable.
   The selection must correspond to the slot where the adapter module is installed. For example, if the adapter module is installed in slot 2, you must select slot 2.
3. With parameter 50.02 FBA A comm loss func, select how the drive reacts to a fieldbus communication break.
   Note that this function monitors both communication between the fieldbus master and the adapter module and communication between the adapter module and the drive.
4. With parameter 50.03 FBA A comm loss t out, define the time between communication break detection and the selected action.
5. Select application-specific values for the rest of the parameters in group 50, starting from 50.04.
   Examples of appropriate values are shown in the tables below.
6. Set the module configuration parameters in group 51.

At the minimum, select the communication protocol and profile with parameter 51.02 and configure the network settings with parameters 51.03…51.13.

7. Define the process data transferred to and from the drive in parameter groups 52 and 53.

   Note: The adapter module assigns the Control word, Status word, references 1…2 and actual values 1…2 automatically to cyclical communication according to the selected assembly instances.

8. Save the valid parameter values to permanent memory with parameter 96.07 Parameter save manually.

9. Validate the settings made in parameter groups 51, 52 and 53 with parameter 51.27 FBA A par refresh.

10. Set the relevant drive control parameters to control the drive according to the application.
    Examples of appropriate values are shown in the tables below.

Parameter setting examples – ACS380, ACS480, ACS580 and ACS880

Speed control using the ODVA AC/DC drive profile, Extended speed control assembly

This example shows how to configure a speed control application that uses the ODVA AC/DC drive profile, Extended speed control assembly. In addition, some application-specific data is added to the communication.

The start/stop commands and reference scaling are according to the ODVA AC/DC drive profile. For more information, see section ODVA AC/DC drive profile (page 56).

When Reference 1 (REF1) is used for speed control and the value of parameter 51.23 is 128, an ODVA speed reference value of ±30000 (decimal) corresponds to an equal amount of rpm in the drive. The speed reference value sent from the PLC is limited by parameter 30.12 Maximum speed in the forward direction and 30.11 Minimum speed in the reverse direction.

The minimum and maximum 16-bit integer values that can be given through the fieldbus are -32768 and 32767 respectively.

Bytes	Instance 121	Instance 171
0…1	Control word	Status word
2…3	Speed reference	Speed actual value
4…7	Constant speed 1 [32] [1]	Output power [32] [1]
8…11	Constant speed [32] [1]	DC voltage [32] [1]

[1] Example

The table below gives the recommended drive parameter settings.

Drive parameter	Setting for drives	Description
50.01 FBA A enable	1 = Option slot 1 [1]	Enables communication between the drive and the fieldbus adapter module.
50.02 FBA A comm loss func	1 = Fault [1]	Enables fieldbus A communication fault monitoring.

Drive parameter	Setting for drives	Description
50.03 FBA A comm loss t out	3.0 s[1]	Defines the fieldbus A communication break supervision time.
50.04 FBA A ref1 type	4 = Speed	Selects the fieldbus A reference 1 type and scaling.
51.01 FBA A type	2222 = EtherNet/IP[2]	Displays the type of the fieldbus adapter module.
51.02 Protocol/Profile	100 = EIP AC/DC	Selects the EtherNet/IP protocol and the ODVA AC/DC drive profile.
51.03 Commrate	0 = Auto[1]	Ethernet communication rate is negotiated automatically by the device.
51.04 IP configuration	0 = Static IP[1]	Configuration will be obtained from parameters 05...13.
51.05 IP address 1	192[1]	First part of the IP address
51.06 IP address 2	168[1]	Second part of the IP address
51.07 IP address 3	0[1]	Third part of the IP address
51.08 IP address 4	16[1]	Last part of the IP address
51.09 Subnet CIDR	24[1]	Sets the network mask as 255.255.255.0, allowing access only to the last subnet.
51.23 ODVA speed scale	128[1]	Sets the scaling for the ODVA speed reference.
52.01 FBA data in1	01.14[1]	Output power
52.03 FBA data in3	01.11[1]	DC voltage
53.01 FBA data out1	22.26[1]	Constant speed 1
53.03 FBA data out3	22.27[1]	Constant speed 2
51.27 FBA A par refresh	1 = Refresh	Validates the FEIP-21 configuration parameter settings.
20.01 Ext1 commands	12 = Fieldbus A	Selects the fieldbus A interface as the source of the start and stop commands for external control location 1.
22.11 Speed ref1 source	4 = FB A ref1	Selects the fieldbus A reference 1 as the source for speed reference 1.

[1] Example
[2] Read-only or automatically detected/set

The start sequence for the parameter example above is given below.

Control word:

- Reset the fieldbus communication fault (if active).
- Enter 0h (0 decimal) → READY.
- Enter 1h (1 decimal) → ENABLED (Running forward).
- Enter 2h (2 decimal) → ENABLED (Running reverse).

Configuring the client

After the adapter module has been initialized by the drive, you must prepare the client for communication with the module. An example of an Allen-Bradley® PLC is given below. If you are using another client system, refer to its documentation for more information.

The example applies to all drive types compatible with the module.

Before you start

Decide on these points before you start the client configuration.

Select profile

The communication profile determines which I/O assemblies and objects are available. See chapter EtherNet/IP – Communication profiles (page 55) for more information.

Select output and input assembly instances

EtherNet/IP devices implement multiple objects each with many attributes. While it is possible to write or read each attribute separately to control the drive, this is inefficient. Assembly object instances provide a means to group writes or reads of attributes. The selection of assembly objects is limited by the choice of the communication profile. This table gives a listing of the output and input assemblies.

Name	Output instance	Input instance	Size (bytes)	Size (16-bit words)	Profile
Basic Speed Control	20	70	4	2	ODVA AC/DC drive
Enhanced Speed Control	21	71	4	2	ODVA AC/DC drive
Basic Speed and Torque Control	22	72	6	3	ODVA AC/DC drive
Enhanced Speed and Torque Control	23	73	6	3	ODVA AC/DC drive
Basic Speed Control plus Drive Parameters	120	170	24	12	ODVA AC/DC drive
Enhanced Speed Control plus Drive Parameters	121	171	24	12	ODVA AC/DC drive
Basic Speed and Torque Control plus Drive Parameters	122	172	26	13	ODVA AC/DC drive
Enhanced Speed and Torque Control plus Drive Parameters	123	173	26	13	ODVA AC/DC drive
ABB Drives Profile w/ Set Speed	1	51	4	2	ABB Drives profile
ABB Drives Profile w/ Set Speed and Set Torque	2	52	6	3	ABB Drives profile
ABB Drives Profile w/ Set Speed plus Drive Parameters	101	151	24	12	ABB Drives profile
ABB Drives Profile w/ Set Speed and Set Torque plus Drive Parameters	102	152	26	13	ABB Drives profile

Name	Output instance	Input instance	Size (bytes)	Size (16-bit words)	Profile
Transparent16 w/One	11	61	4	2	Transparent16 profile
Transparent16 w/Two	12	62	6	3	Transparent16 profile
Transparent16 w/One plus Drive Parameters	111	161	24	12	Transparent16 profile
Transparent16 w/Two plus Drive Parameters	112	162	26	13	Transparent16 profile
Transparent32 w/One	21	71	8	4	Transparent32 profile
Transparent32 w/Two	22	72	12	6	Transparent32 profile
Transparent32 w/One plus Drive Parameters	121	171	28	14	Transparent32 profile
Transparent32 w/Two plus Drive Parameters	122	172	32	16	Transparent32 profile

Select connection method

EtherNet/IP provides a variety of connection methods to communicate between devices. Not all methods are supported by all devices. Refer to the client documentation to determine which method(s) are supported by the client.

Note: The choice of the connection method has a significant impact on the timeout behavior. Refer to configuration parameters 20 Control timeout and 21 Idle action for more information.

The FEIP-21 adapter module supports the following connection methods:

I/O connections

The adapter module supports Class 1 I/O connections. I/O connections are often also referred to as “Implicit Messaging”. I/O connections are typically established by configuring an I/O scanner to write and read assembly object instances.

Connected explicit messaging

The adapter module supports Class 3 connected explicit messaging. Class 3 connected explicit messages are typically established by using a “message instruction” to write or read an attribute.

Note: When using Class 3 explicit messaging, some EtherNet/IP clients may close the connection after the MSG instruction is done. This will cause the module to behave as if it were controlled via unconnected explicit messaging.

Unconnected explicit messaging

The adapter module supports unconnected explicit messaging. Unconnected explicit messages are typically established by using a “message instruction” to write or read an attribute.

Note: EtherNet/IP does not provide a timeout means for unconnected explicit messaging. To use unconnected explicit messaging for control, refer to configuration parameter 20 Control timeout (page 30).

EDS files

Electronic Data Sheet (EDS) files specify the properties of the device for the EtherNet/IP client. The client identifies the device by means of the product code, device type and major revision attributes.

To enable the use of different ABB drive types on the same EtherNet/IP network, a unique product code has been given to each drive type and application combination.

EDS files are available from the Document library (http://new.abb.com/drives/ethernet-ip).

Note: Only one EDS file with the same EtherNet/IP product code can be installed in the PLC at a time.

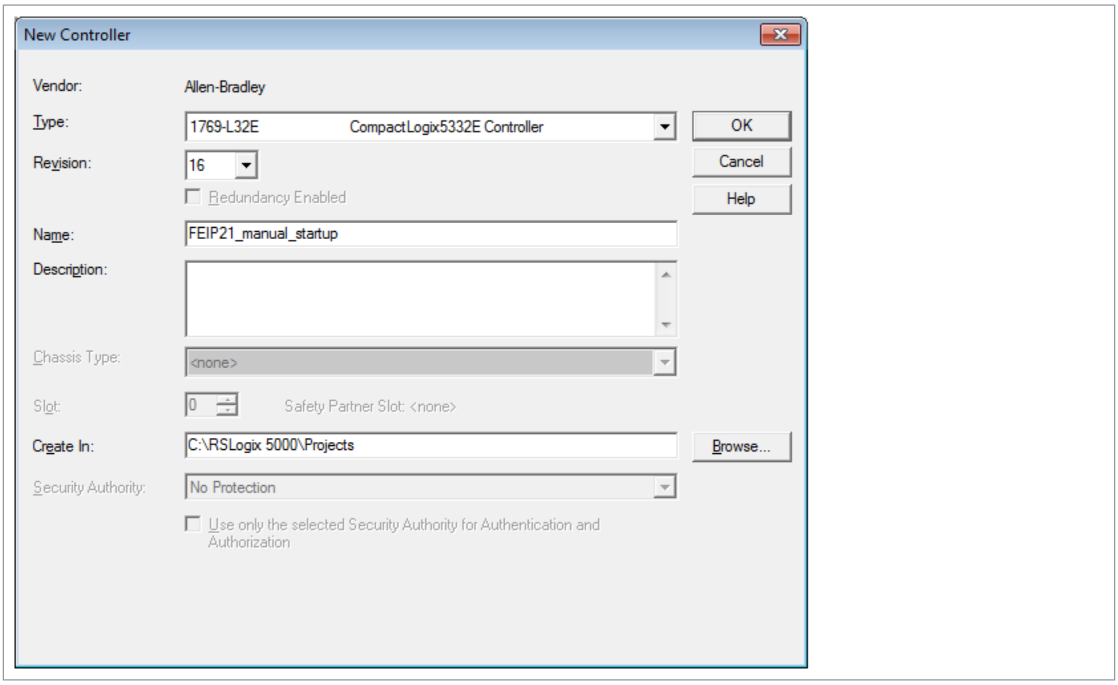
Configuring an Allen-Bradley® PLC

Example 1: RSLogix 5000

This example shows how to prepare an Allen-Bradley® Control-Logix5555™ PLC for communication with the adapter module by using the RSLogix 5000® software as the configuration tool.

1. Start the RSLogix software and open/create an RSLogix project.

   Note: It is assumed that the PLC configuration has already been established in the RSLogix project.



New Controller

Vendor: Allen-Bradley

Type: 1769-L32E CompactLogix5332E Controller

Revision: 16

□ Redundancy Enabled

Name: FEIP21_manual_startup

Description:

Chassis Type: <none>

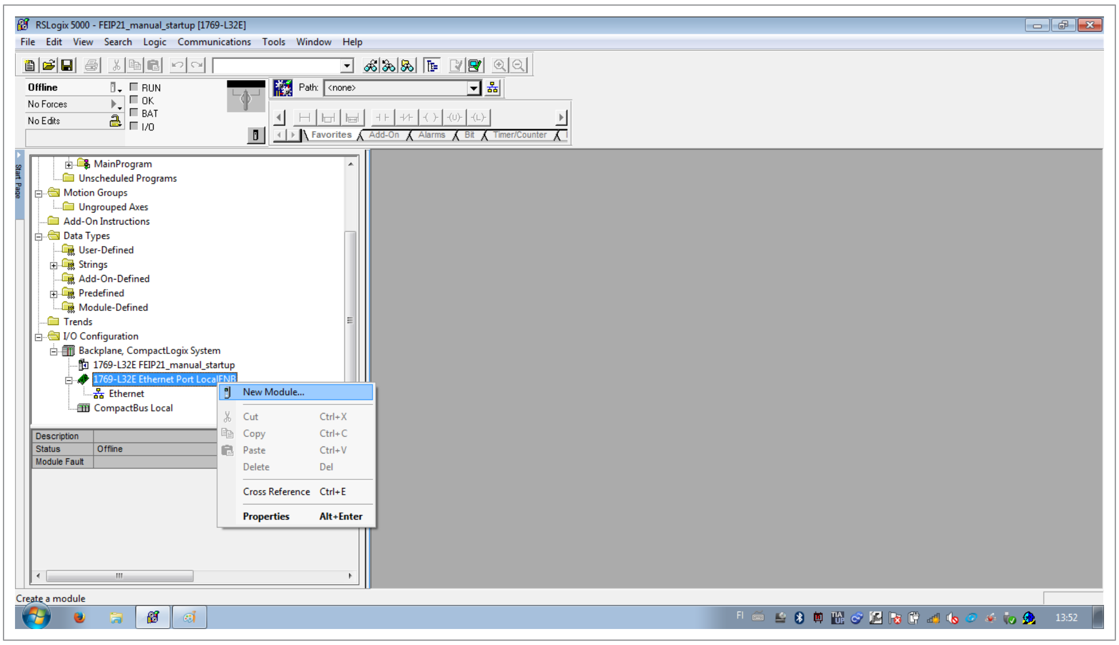
Slot: 0 Safety Partner Slot: <none>

Create In: C:\RSLogix 5000\Projects

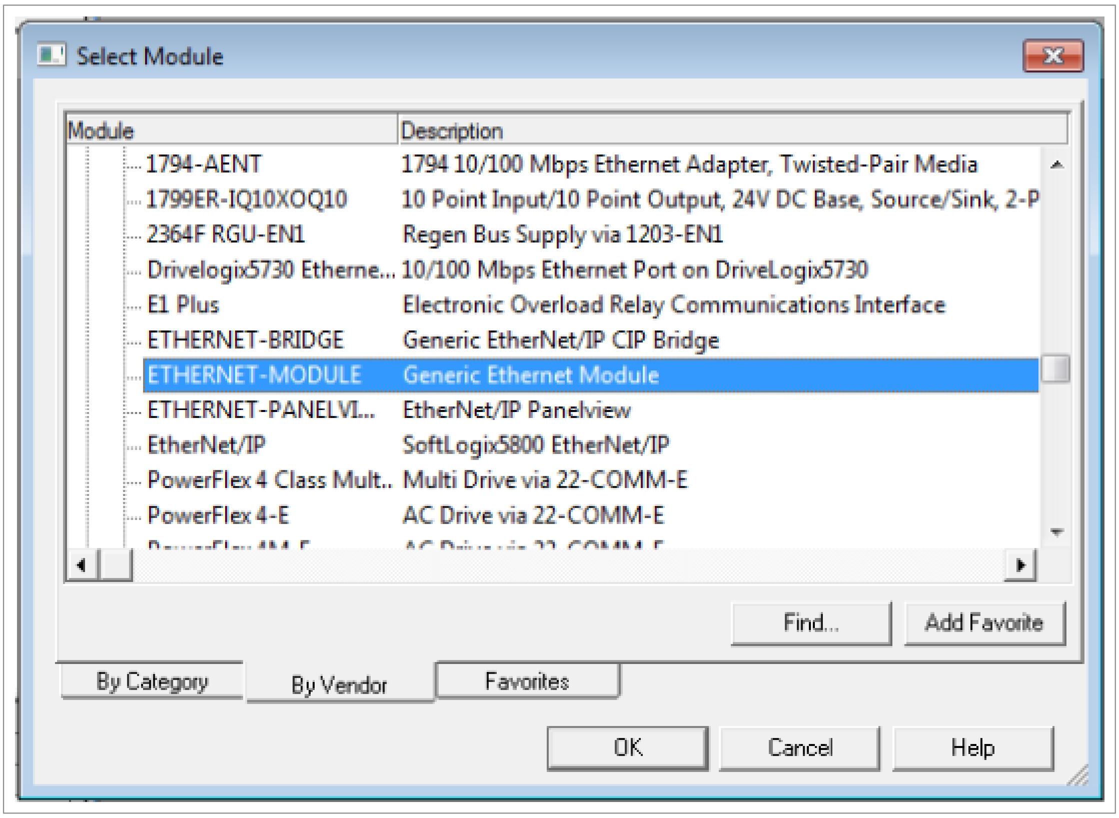
Security Authority: No Protection

□ Use only the selected Security Authority for Authentication and Authorization

2. In the RSLogix I/O, right-click the EtherNet/IP communication module and select New Module.



3. In the Select Module window, select ETHERNET-MODULE.



Module	Description
1794-AENT	1794 10/100 Mbps Ethernet Adapter, Twisted-Pair Media
1799ER-IQ10XOQ10	10 Point Input/10 Point Output, 24V DC Base, Source/Sink, 2-P
2364F RGU-EN1	Regen Bus Supply via 1203-EN1
Drivelogix5730 Etherne...	10/100 Mbps Ethernet Port on DriveLogix5730
E1 Plus	Electronic Overload Relay Communications Interface
ETHERNET-BRIDGE	Generic EtherNet/IP CIP Bridge
ETHERNET-MODULE	Generic Ethernet Module
ETHERNET-PANELVI...	EtherNet/IP Panelview
EtherNet/IP	SoftLogix5800 EtherNet/IP
PowerFlex 4 Class Mult..	Multi Drive via 22-COMM-E
PowerFlex 4-E	AC Drive via 22-COMM-E

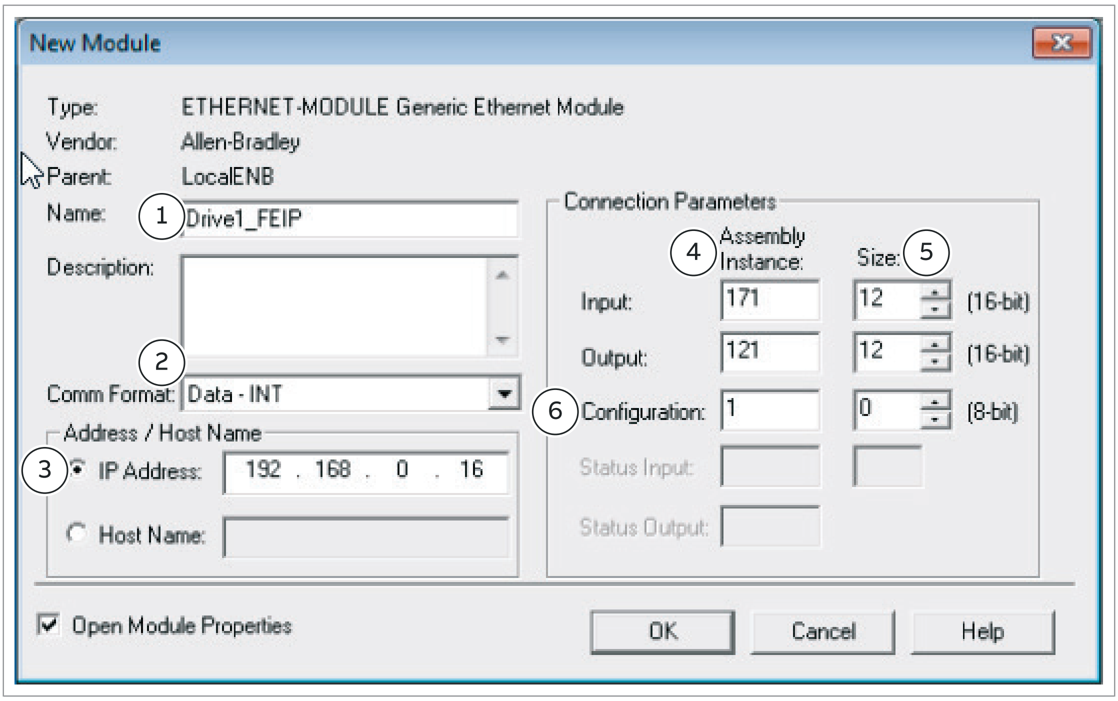
4. Select the input and output assembly instances and the PLC I/O memory size to be used.
The table below shows the available combinations. The example below uses the ODVA AC/DC assembly instances 121 and 171.

Input assembly instances	Output assembly instances	PLC word settings
70	20	2
71	21	2

Input assembly instances	Output assembly instances	PLC word settings
72	22	3
73	23	3
170	120	12
171	121	12
172	122	13
173	123	13
51	1	2
52	2	3
151	101	12
152	102	13
61	11	2
62	12	3
161	111	12
162	112	13

For more information on the input/output assembly instances, see section Select output and input assembly instances (page 40).

5. Enter the following information.
   The example below uses ODVA AC/DC assembly instances 121 and 171. The PLC will transmit and receive 12 words.



New Module

Type: ETHERNET-MODULE Generic Ethernet Module
Vendor: Allen-Bradley
Parent: LocalENB
Name: (1) Drive1_FEIP
Description:
Comm Format: (2) Data - INT
Address / Host Name
(3) ● IP Address: 192 . 168 . 0 . 16
○ Host Name:

Connection Parameters

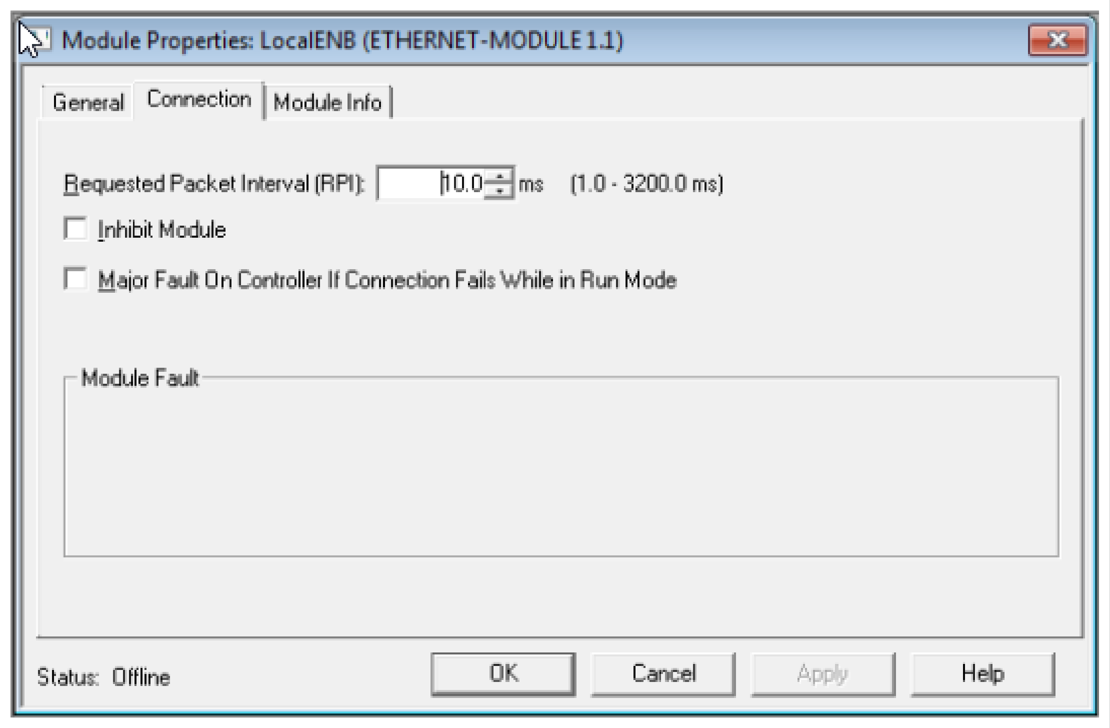
	(4) Assembly Instance:	Size: (5)	
Input:	171	12	(16-bit)
Output:	121	12	(16-bit)
(6) Configuration:	1	0	(8-bit)
Status Input:			
Status Output:			

☑ Open Module Properties

- Type a name for the adapter module. (1)
- FEIP uses 16-bit words. Change Comm Format to Data - INT (16 bits). (2)
- Type the IP address of the adapter module. (3)

- Type the Input and Output Assembly Instance numbers. (4)
- Select the sizes of the Input and Output words for the adapter module. (5)
- Set Configuration as 1 and Size as 0. (6)

6. Click OK.
   The adapter module is now added to the PLC I/O.
7. Click the FEIP-21 module to open the Module Properties window.
8. On the Connection tab, select the Requested Packet Interval (RPI) for the adapter module I/O communication.



Module Properties: LocalENB (ETHERNET-MODULE 1.1)

Requested Packet Interval (RPI): 10.0 ms (1.0 - 3200.0 ms)

□ Inhibit Module

□ Major Fault On Controller If Connection Fails While in Run Mode

Status: Offline

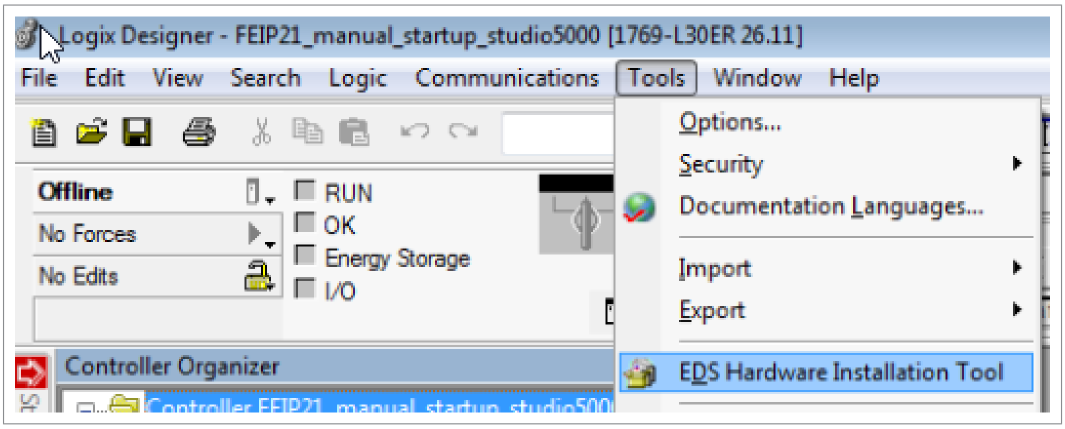
9. Download the new configuration to the PLC.
   The PLC is now ready for communication with the adapter module.

Example 2: Studio 5000

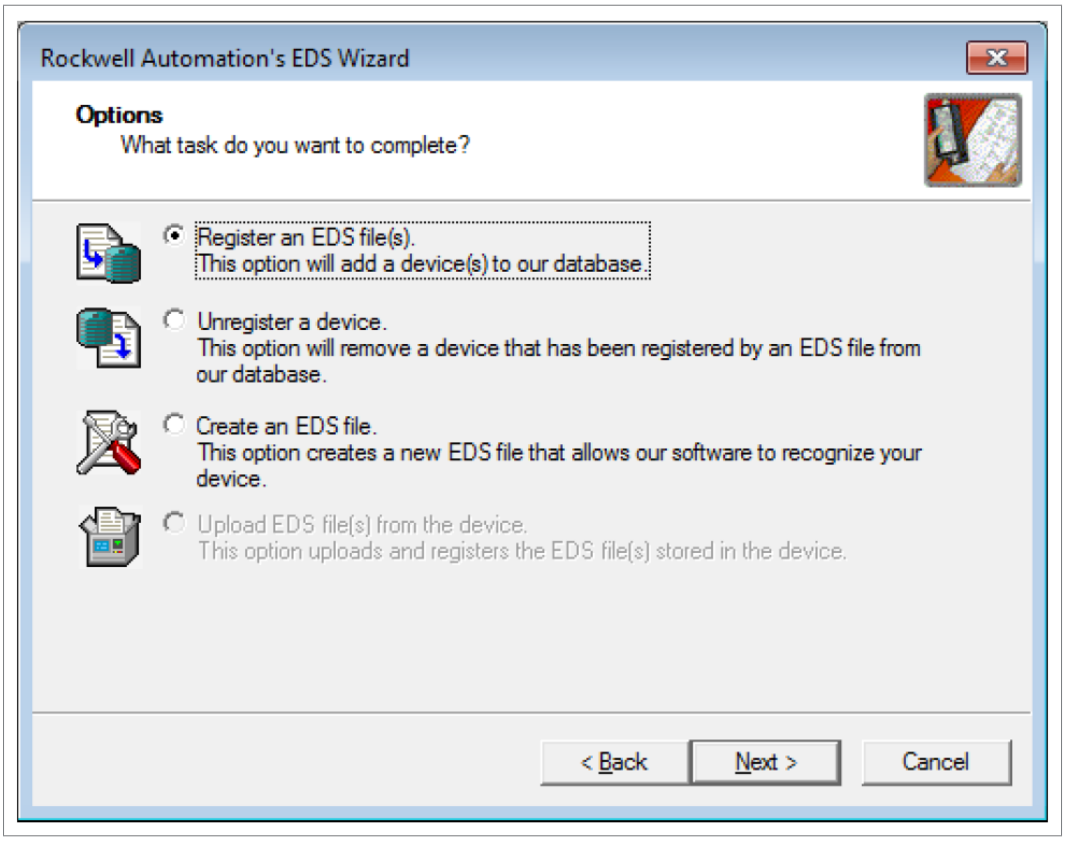
This example shows how to prepare an Allen-Bradley® CompactLogix™ PLC for communication with the adapter module using the Studio 5000® software as the configuration tool.

1. Start the RSLogix software and open/create an RSLogix project.

   Note: It is assumed that the PLC configuration was already established in the Studio 5000® project.
2. If EDS file for the correct device is not installed, use the EDS hardware installation tool. To register a new EDS file:
   - Select Tools → EDS Hardware Installation Tool.


Logix Designer - FEIP21_manual_startup_studio5000 [1769-L30ER 26.11]
File Edit View Search Logic Communications Tools Window Help
Offline
No Forces
No Edits
RUN
OK
Energy Storage
I/O
Options...
Security
Documentation Languages...
Import
Export
EDS Hardware Installation Tool
Controller Organizer
Controller FEIP21_manual_startup_studio500

- Select the option Register an EDS file(s). Click Next.



Options

What task do you want to complete?

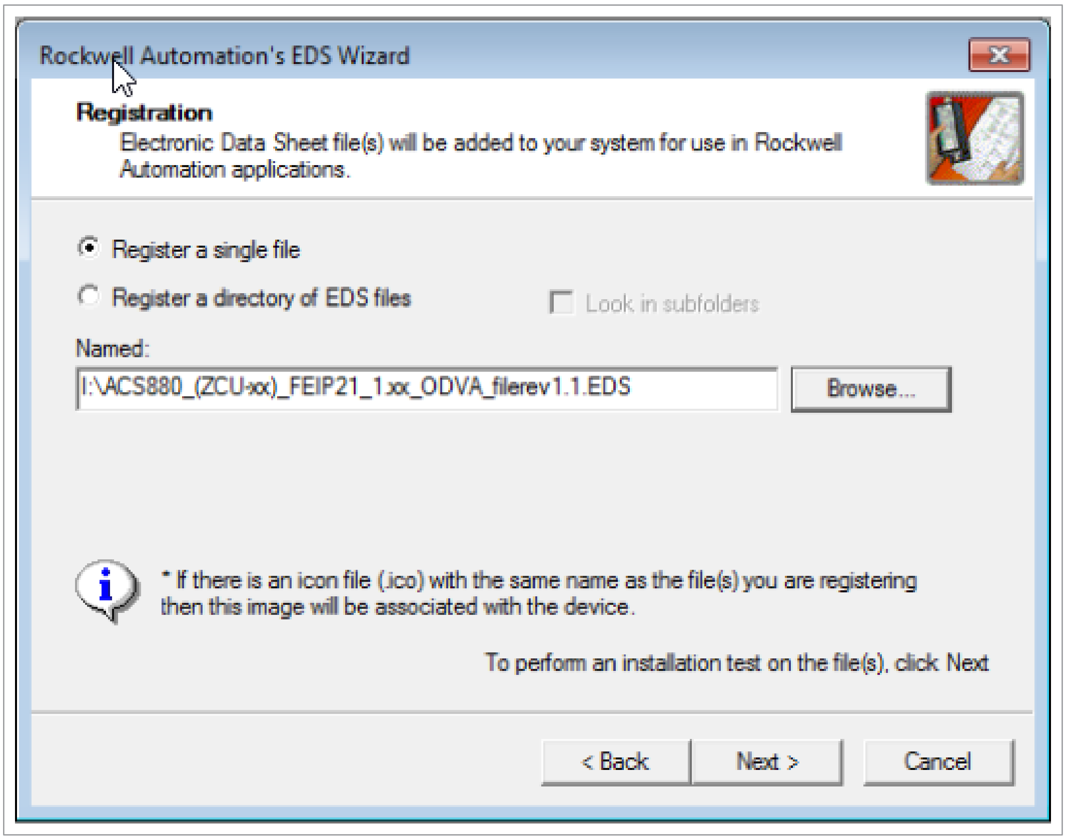
● Register an EDS file(s).
This option will add a device(s) to our database.

○ Unregister a device.
This option will remove a device that has been registered by an EDS file from our database.

○ Create an EDS file.
This option creates a new EDS file that allows our software to recognize your device.

○ Upload EDS file(s) from the device.
This option uploads and registers the EDS file(s) stored in the device.

- Browse to FEIP-21 EDS file and select the file. Click Next to register the EDS file.



Rockwell Automation's EDS Wizard

Registration

Electronic Data Sheet file(s) will be added to your system for use in Rockwell Automation applications.

● Register a single file

○ Register a directory of EDS files □ Look in subfolders

Named:

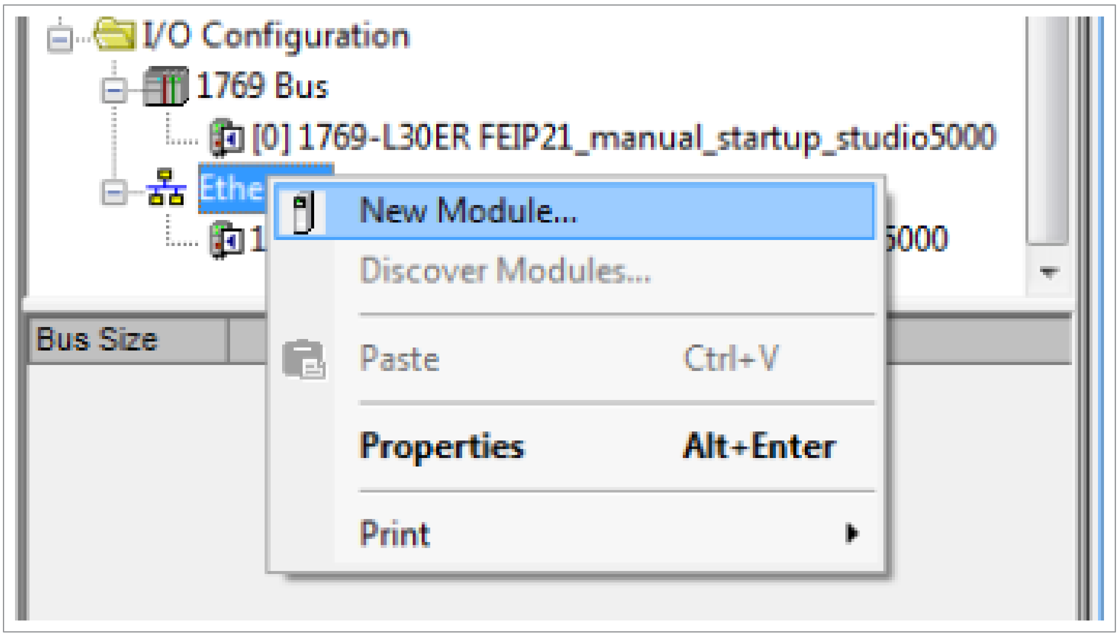
I:\ACS880_(ZCU-xx)_FEIP21_1.xx_ODVA_filerev1.1.EDS

* If there is an icon file (.ico) with the same name as the file(s) you are registering then this image will be associated with the device.

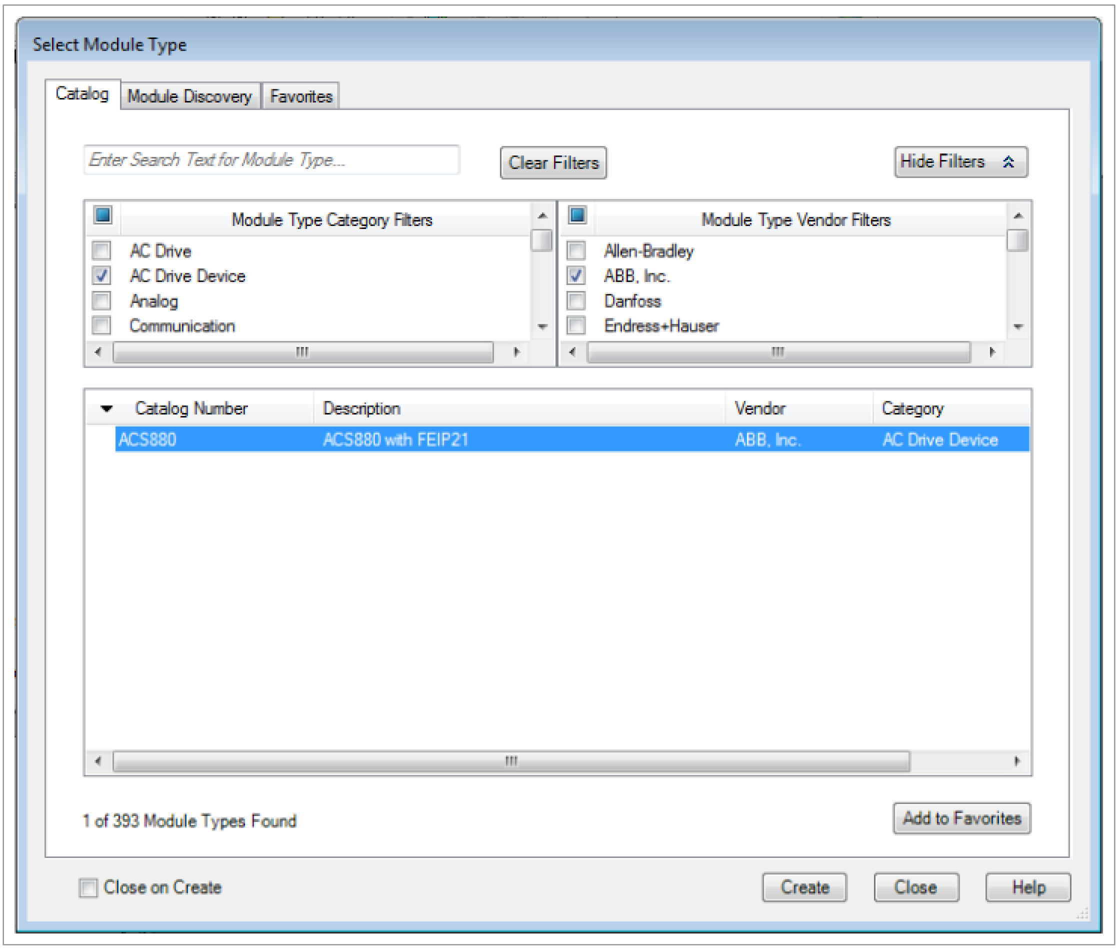
To perform an installation test on the file(s), click Next

- Click Next and Finish to finalize registration. After the EDS file is successfully registered the device can be used in the PLC project.

3. Add new devices to EtherNet/IP bus by right-clicking Ethernet under I/O Configuration and selecting New Module.



4. In the Select Module Type window, select ACS880 module (AC Drive Device). You can easily find the ABB devices using filters. Click Create to add a new module.



Select Module Type

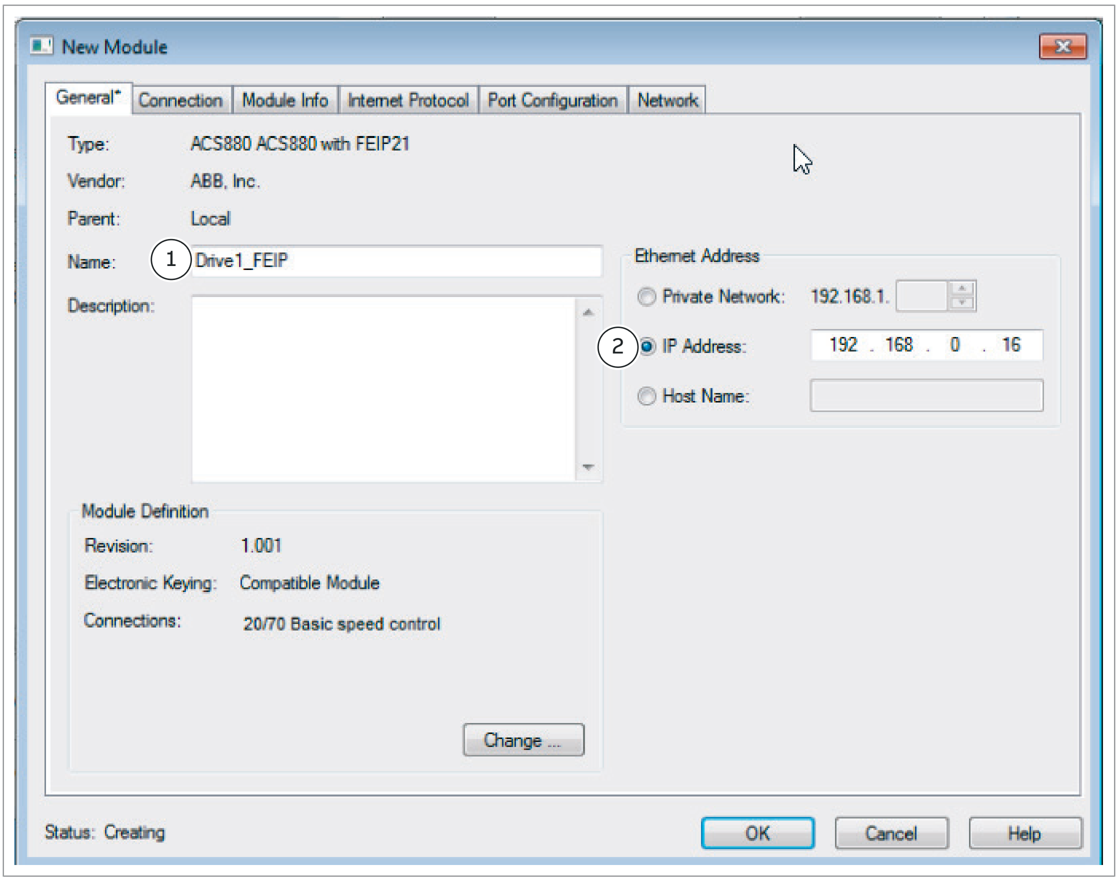
Module Type Category Filters	Module Type Vendor Filters
□ AC Drive	□ Allen-Bradley
☑ AC Drive Device	☑ ABB, Inc.
□ Analog	□ Danfoss
□ Communication	□ Endress+Hauser

Catalog Number	Description	Vendor	Category
ACS880	ACS880 with FEIP21	ABB, Inc.	AC Drive Device

1 of 393 Module Types Found

□ Close on Create

5. Enter the following information to configure the IP address and module name.



New Module

Type: ACS880 ACS880 with FEIP21

Vendor: ABB, Inc.

Parent: Local

Name: Drive1_FEIP

Description:

Ethernet Address

○ Private Network: 192.168.1.

● IP Address: 192 . 168 . 0 . 16

○ Host Name:

Module Definition

Revision: 1.001

Electronic Keying: Compatible Module

Connections: 20/70 Basic speed control

Status: Creating

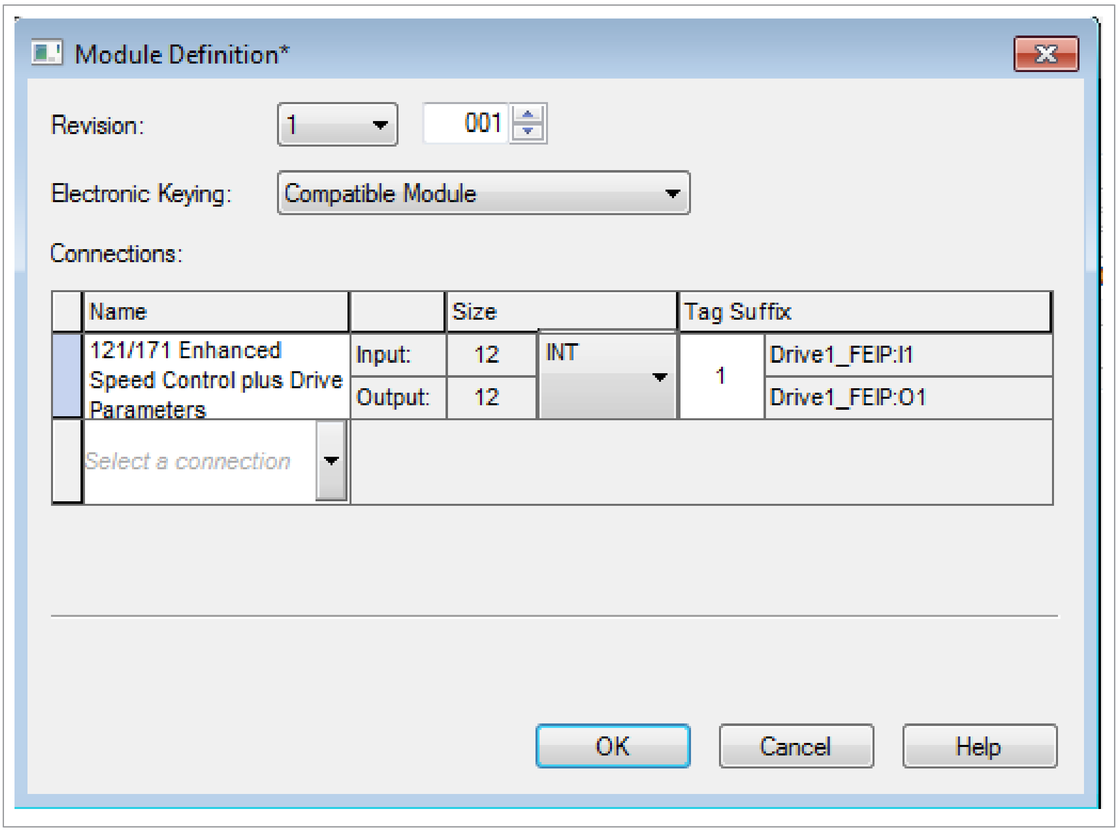
- Type a name for the adapter module. (1)
- Type the IP address of the adapter module. (2)

6. Click Change, to select the input and output assembly instances and the PLC I/O memory size to be used. The table below shows the available combinations.

Input assembly instances	Output assembly instances	PLC word setting
70	20	2
71	21	2
72	22	3
73	23	3
170	120	12
171	121	12
172	122	13
173	123	13
51	1	2
52	2	3
151	101	12
152	102	13
61	11	2
62	12	3
161	111	12
162	112	13

For more information on the input/output assembly instances, see section Select output and input assembly instances (page 40).

FEIP-21 uses 16-bit words. Change the size to INT (16 bits). The example below uses ODVA AC/DC assembly instances 121 and 171. The PLC transmits and receives 12 words.



Module Definition*

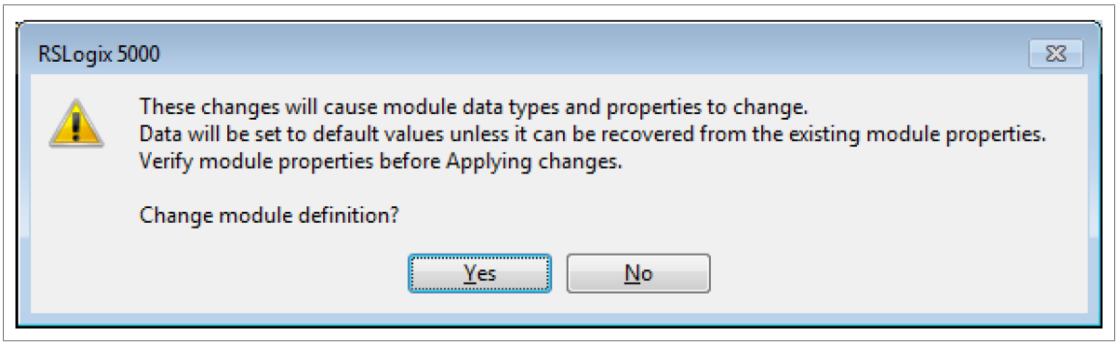
Revision: 1 001

Electronic Keying: Compatible Module

Connections:

Name		Size		Tag Suffix	
121/171 Enhanced Speed Control plus Drive Parameters	Input:	12	INT	1	Drive1_FEIP:I1
	Output:	12			Drive1_FEIP:O1
Select a connection					

7. Click OK and confirm selection to change the module data types.

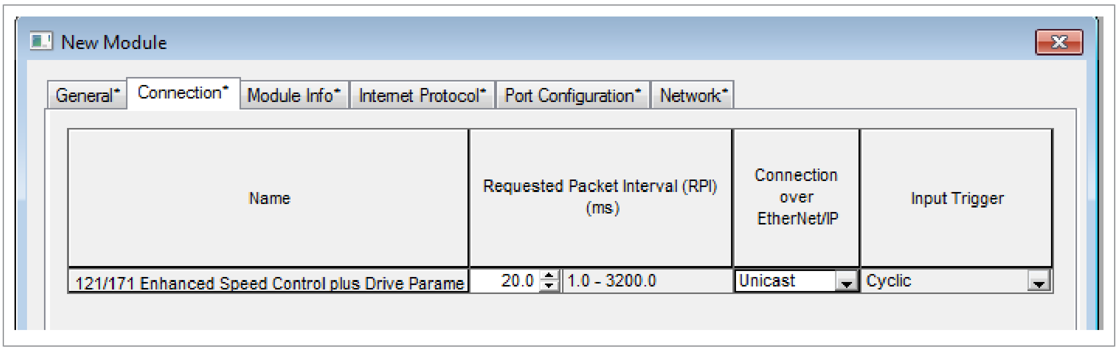


RSLogix 5000

These changes will cause module data types and properties to change.
Data will be set to default values unless it can be recovered from the existing module properties.
Verify module properties before Applying changes.

Change module definition?

8. On the Connection tab, select the Requested Packet Interval (RPI) for the adapter module I/O communication.



New Module

Name	Requested Packet Interval (RPI) (ms)		Connection over EtherNet/IP	Input Trigger
121/171 Enhanced Speed Control plus Drive Parame	20.0	1.0 - 3200.0	Unicast	Cyclic

9. Click OK. The adapter module is now added to the PLC I/O. You can add more modules by choosing Create or exit the window by choosing Close.

10. Download the new configuration to the PLC. The PLC is now ready for communication with the adapter module.

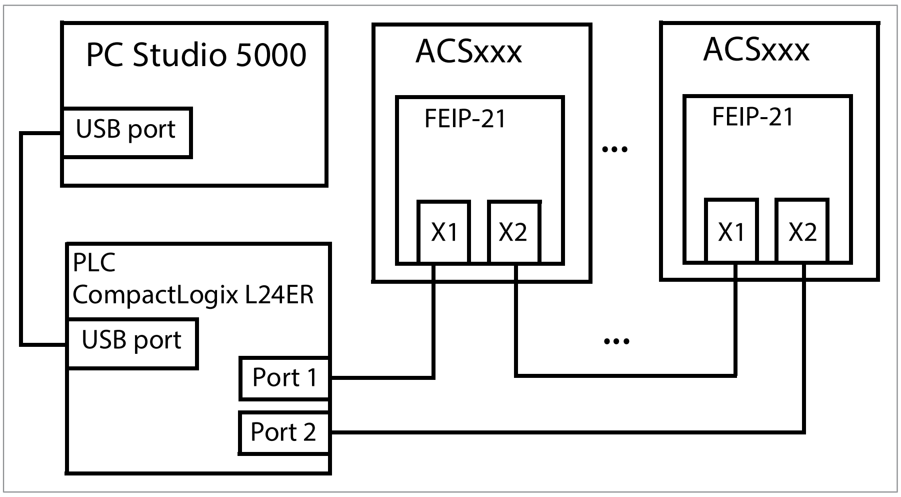
Configuring DLR topology for FEIP-21

This example shows how to prepare an Allen-Bradley® CompactLogix™ PLC for DLR topology with FEIP-21 adapter modules. After installing the devices on the DLR network, at least one supervisor node must be configured. Configuration can be done by using the Studio 5000® Logix Designer or RSLinx® Classic Lite software.

Note: The examples below uses ACSxxx drive. You can also use this configuration with other drives that supports FEIP-21.

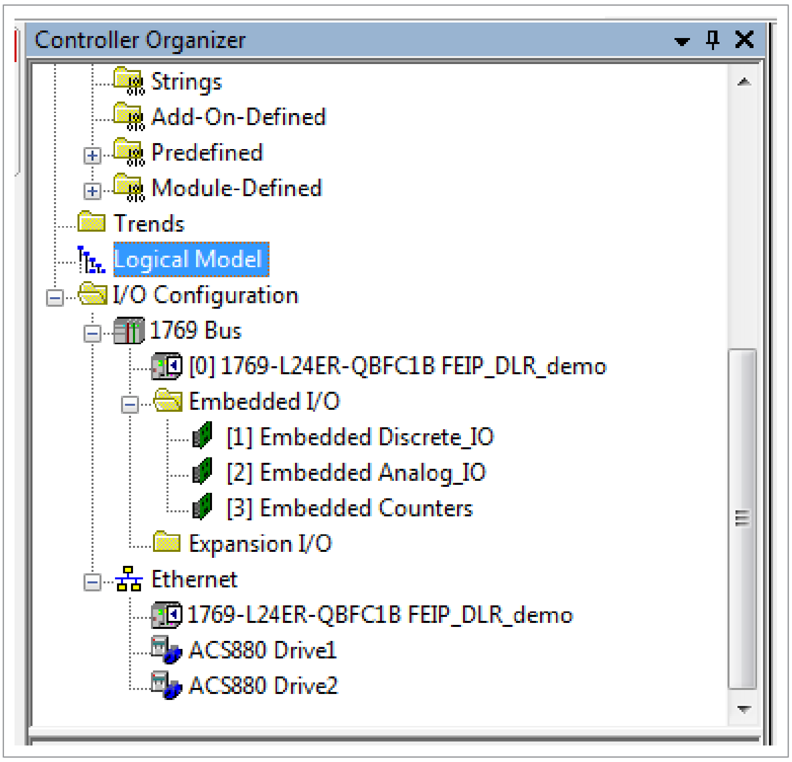
Setup using Logix Designer

1. Open the Studio 5000® software. Test setup uses an Allen Bradley PLC connected in a ring topology with two FEIP-21 fieldbus Ethernet modules. The topology used in the example is shown below. More devices can be added, but the recommended maximum number of nodes on a single DLR network is 50.

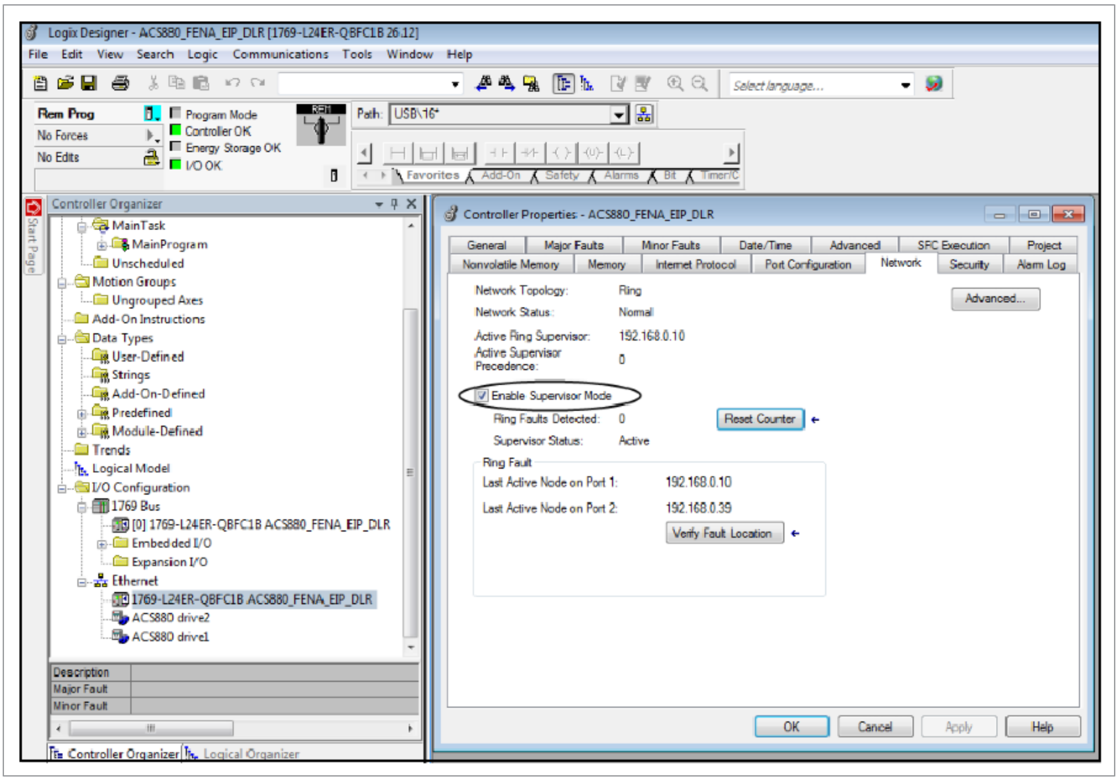


Note: It is assumed that the PLC configuration was already established in the Studio 5000® project and the EDS file(s) are installed and at least two FEIP-21 modules are added to the project.

For more information on adding modules to a project and installing EDS files, see chapter Configuring an Allen-Bradley® PLC (page 42).



2. Download the project to the PLC.
3. Go online with the PLC and leave it in Program mode.
4. Double-click the module in the I/O Configuration. In the Controller Properties window, open Network tab and select Enable Supervisor Mode. Click OK.



Controller Properties - ACS880_FENA_EIP_DLR

Network Topology: Ring
Network Status: Normal
Active Ring Supervisor: 192.168.0.10
Active Supervisor Precedence: 0

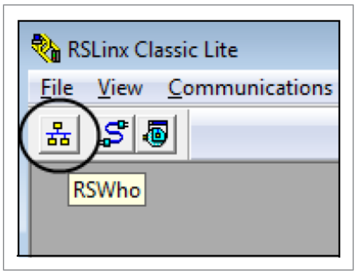
☑ Enable Supervisor Mode
Ring Faults Detected: 0
Supervisor Status: Active

Ring Fault
Last Active Node on Port 1: 192.168.0.10
Last Active Node on Port 2: 192.168.0.39

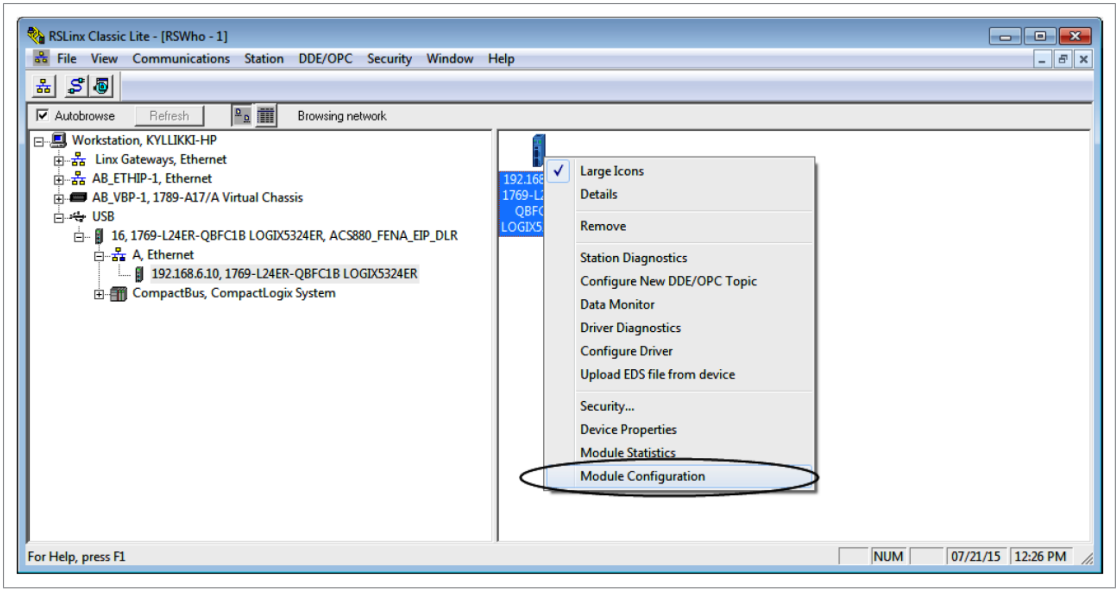
Setup using RSLinx® Classic

You can configure and enable DLR supervisor via RSLinx®Classic.

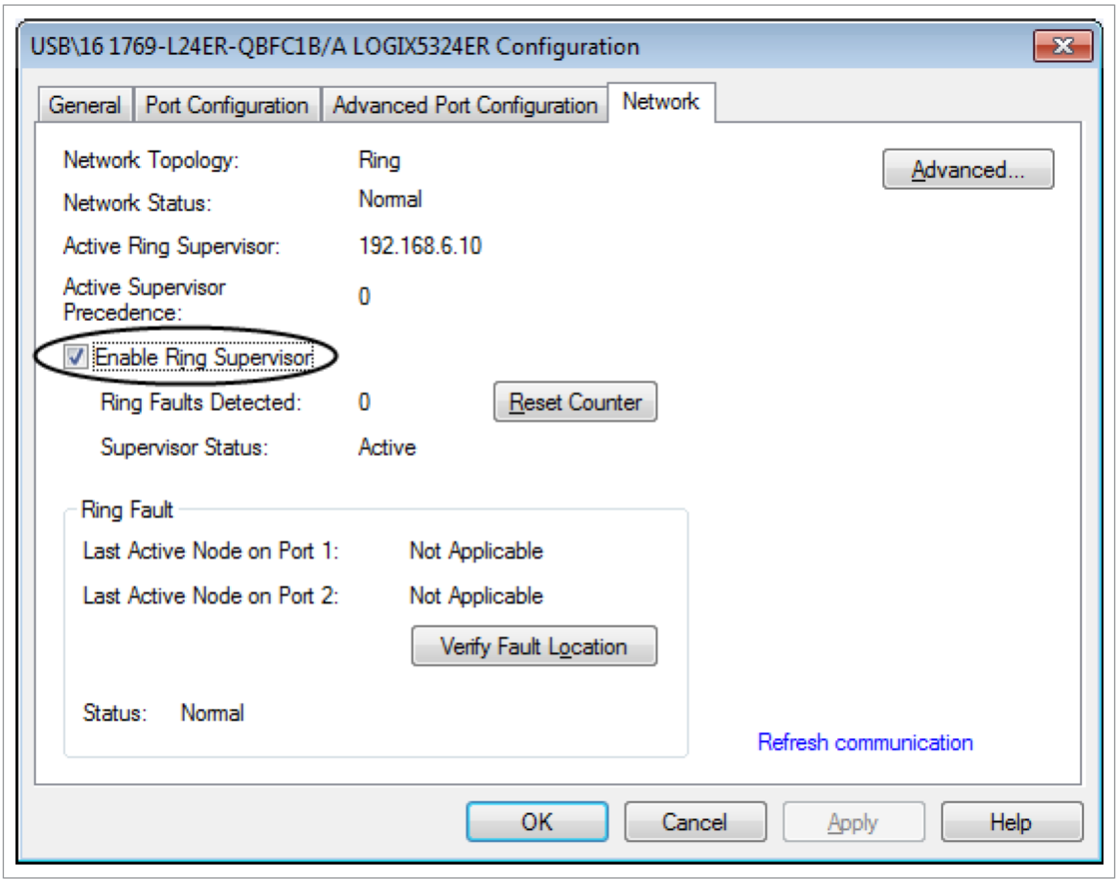
1. Open the RSLinx®Classic software.
2. Browse to the DLR network.



3. Open the Module Configuration by right-clicking on the ring supervisor in the list.



4. On the Network tab, select Enable Ring Supervisor, to enable DLR messages in the ring.



USB\16 1769-L24ER-QBFC1B/A LOGIX5324ER Configuration

Network Topology: Ring

Network Status: Normal

Active Ring Supervisor: 192.168.6.10

Active Supervisor Precedence: 0

☑ Enable Ring Supervisor

Ring Faults Detected: 0

Supervisor Status: Active

Ring Fault

Last Active Node on Port 1: Not Applicable

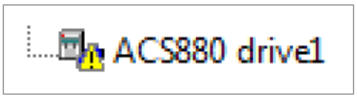
Last Active Node on Port 2: Not Applicable

Status: Normal

5. Click Advanced... to configure DLR parameters, such as Beacon Interval and Beacon Timeout.

   Note: It is recommended to use the default values.

6. Go back to Logix Designer and make sure that none of the FEIP-21 modules are faulted, that is no warning symbols are displayed.



ACS880 drive1

7

EtherNet/IP – Communication profiles

Contents of this chapter

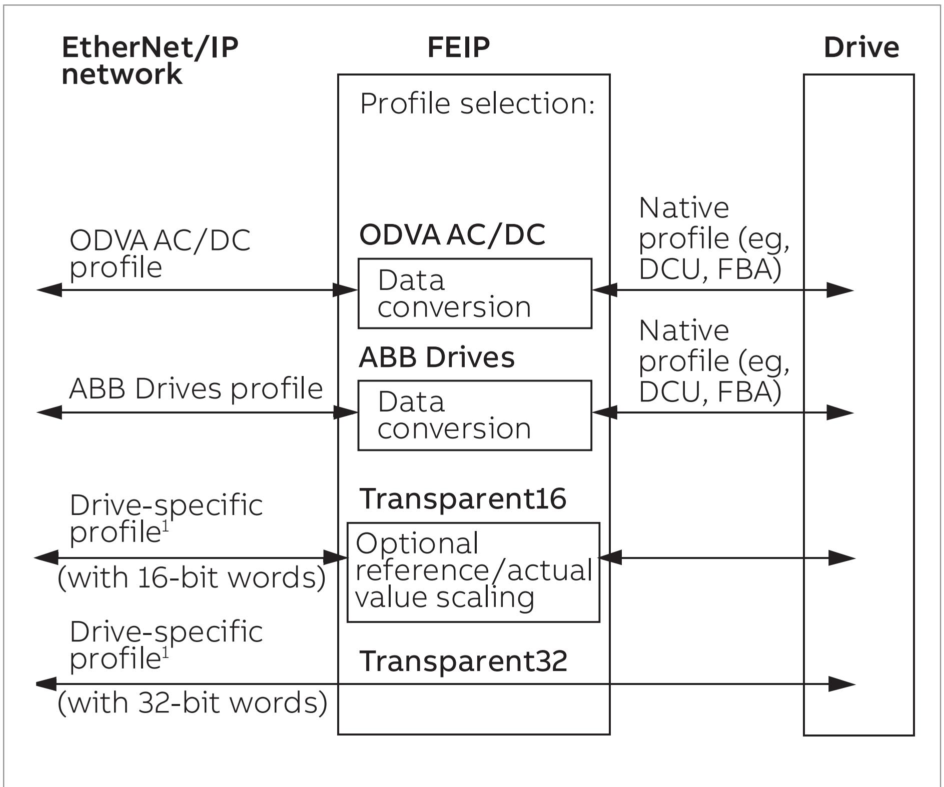
This chapter describes the communication profiles used in the communication between the EtherNet/IP client, the adapter module and the drive.

Communication profiles

Communication profiles are ways of conveying control commands (Control word, Status word, references and actual values) between the master station and the drive.

With the FEIP-21 adapter module, the EtherNet/IP network may employ either the ODVA AC/DC drive profile or the ABB Drives profile. Both are converted to the native profile (eg, DCU or FBA) by the adapter module. In addition, two Transparent modes – for 16-bit and 32-bit words respectively – are available. With the Transparent modes, no data conversion takes place.

The figure below illustrates the profile selection:


EtherNet/IP network
FEIP
Drive
Profile selection:
ODVA AC/DC profile
ODVA AC/DC
Data conversion
Native profile (eg, DCU, FBA)
ABB Drives profile
ABB Drives
Data conversion
Native profile (eg, DCU, FBA)
Drive-specific profile1 (with 16-bit words)
Transparent16
Optional reference/actual value scaling
Drive-specific profile1 (with 32-bit words)
Transparent32

[1] Can be used if the native profile is supported by the drive.

The following sections describe the Control word, the Status word, references and actual values for the ODVA AC/DC drive and ABB Drives communication profiles. Refer to the drive manuals for details on the native profiles.

ODVA AC/DC drive profile

This section briefly describes the ODVA AC/DC drive profile. Additional information is available at www.odva.org.

An EtherNet/IP node is modeled as a collection of abstract objects. Each object represents the interface to and behavior of a component within the product. The ODVA AC/DC drive profile defines a collection of objects suitable for the control of AC and DC drives. The objects supported by the adapter module are listed in section Class objects (page 94).

Objects are defined by:

- Service
- Class
- Instance
- Attribute
- Behavior.

For example, to set the drive speed reference, the Set_Attribute_Single service can be requested for the SpeedRef attribute of the AC/DC drive object class. The resulting behavior is that the reference speed of the drive is set to the requested value.

This is an example of explicit messaging, where each attribute of a class is set individually. While this is allowed, it is inefficient. Instead, implicit messaging using input and output assembly Instances is recommended. Implicit messaging allows the

EtherNet/IP client to set or get predefined groups of attributes in a single message exchange. The assembly instances supported by the adapter module are listed and defined in section Assembly objects (page 67).

ODVA output attributes

This section briefly describes the instances found in the output assemblies of the ODVA AC/DC drive profile. Note that all output assembly instances do not support all attributes listed here.

In the ODVA EtherNet/IP specification the word *output* is used to describe data flow from the network into a device (such as the adapter module).

Run Forward & Run Reverse (Control supervisor object)

These attributes are used to assert run and stop commands to the Control supervisor object state machine according to the following Run/Stop event matrix. See sectionState (Control supervisor object) (page 59).

RunFwd (Run1)	RunRev (Run2)	Trigger event	Run type
0	0	Stop	N/A
0 → 1	0	Run	RunFwd
0	0 → 1	Run	RunRev
0 → 1	0 → 1	No action	N/A
1	1	No action	N/A
0 → 1	1	Run	RunRev
1	1 → 0	Run	RunFwd

Fault Reset (Control supervisor object)

This attribute resets a drive fault on a transition from zero to one if the condition that caused the fault has been cleared.

Net Ctrl (Control supervisor object)

This attribute requests that the drive Run/Stop command be supplied locally (Net Ctrl = 0) or by the network (Net Ctrl = 1).

Net Ref (AC/DC drive object)

This attribute requests that the drive speed and torque references be supplied locally (Net Ref = 0) or by the network (Net Ref = 1).

Speed Reference (AC/DC drive object)

This attribute is the speed reference for the drive. The units are scaled by the Speed Scale attribute of the AC/DC drive object. See parameter 23 Speed scale (page 31) for details.

Scalar mode

When the drive is operating in the scalar mode, the adapter module provides the drive with a frequency reference. The ODVA AC/DC drive profile uses rpm units for the speed reference. The drive frequency reference is calculated as follows:

$$\text{Dfr} = \frac{\text{Osr} \times \text{Us} \times \text{Mf}}{\text{Mss}}$$

where

Dfr = Drive Frequency Reference in Hz

Osr = ODVA Speed Reference

Us = ODVA Speed Unit (see parameter 23 Speed scale (page 31))

Mf = Motor Nominal Frequency in Hz

Mss = Motor Synchronous Speed in rpm (not Motor Nominal Speed).

For example, for a 4-pole 60 Hz motor (Mss = 1800 rpm) with a unit of 1 rpm and an ODVA Speed Reference of 900, the drive frequency reference is:

$$\text{Dfr} = \frac{\text{Osr} \times \text{Us} \times \text{Mf}}{\text{Mss}} = \frac{900 \times 1\text{rpm} \times 60\text{Hz}}{1800\text{rpm}} = 30\text{Hz}$$

Vector mode

When the drive is operating in the vector mode, the adapter module provides the drive with a speed reference. The ODVA AC/DC drive profile uses rpm units for the speed reference. The drive speed reference is calculated as follows:

$$\text{Dsr} = \text{Osr} \times \text{Us}$$

where

Dsr = Drive Speed Reference in rpm

Osr = ODVA Speed Reference

Us = ODVA Speed Unit (see parameter 23 Speed scale (page 31)).

For example, for an ODVA Speed Reference of 900 rpm with a unit of 0.5 rpm, the drive speed reference is:

$$\text{Dsr} = \text{Osr} \times \text{Us} = 900 \times 0.5\text{rpm} = 450\text{rpm}$$

Torque Reference (AC/DC drive object)

This attribute is the torque reference for the drive. The units are scaled by the Torque Scale attribute of the AC/DC drive object. See parameter 24 Torque scale (page 32) for details.

The adapter module provides the drive with a torque reference in percent of the motor nominal torque. The ODVA AC/DC drive profile uses Newton-meter (N·m) units for the torque reference. The drive torque reference is calculated as follows:

$$\text{Dtr} = \frac{100 \times \text{Otr} \times \text{Ut}}{\text{Mt}}$$

where

Dtr = Drive Torque Reference in Percent of Motor Nominal Torque

Otr = ODVA Torque Reference

Ut = ODVA Torque Unit (see 24 Torque scale on page 128)

Mt = Motor Nominal Torque in N·m.

For example, for a 1000 N·m Motor Nominal Torque with a unit of 1 N·m and an ODVA Torque Reference of 500, the drive torque reference is:

$$\text{Dtr} = \frac{100 \times \text{Otr} \times \text{Ut}}{\text{Mt}} = \frac{100 \times 500 \times 1\ \text{Nm}}{1000\ \text{Nm}} = 50$$

ODVA input attributes

This section briefly describes the instances found in the ODVA AC/DC drive profile's input assemblies. Note that all input assembly instances do not support all attributes listed here.

In the ODVA EtherNet/IP specification the word *input* is used to describe data flow from a device (such as the adapter module) to the network.

Faulted (Control supervisor object)

This attribute indicates that the drive has experienced a fault. The fault code may be read from the FaultCode attribute of the Control supervisor object.

Warning (Control supervisor object)

This attribute indicates that the drive is experiencing a warning condition. The warning code may be read from the WarnCode attribute of the Control supervisor object.

Running Forward (Control supervisor object)

This attribute indicates that the drive is running in the forward direction.

Running Reverse (Control supervisor object)

This attribute indicates that the drive is running in the reverse direction.

Ready (Control supervisor object)

This attribute indicates that the Control supervisor object state machine is in the Ready, Running or Stopping state. See State (Control supervisor object) (page 59).

Ctrl From Net (Control supervisor object)

This attribute indicates if the Run/Stop command is being supplied locally (Ctrl From Net = 0) or by the network (Ctrl From Net = 1).

Ref From Net (AC/DC drive object)

This attribute indicates if the speed and torque references are being supplied locally (Ref From Net = 0) or by the network (Ref From Net = 1).

At Reference (AC/DC drive object)

This attribute indicates that the drive is operating at the specified speed or torque reference.

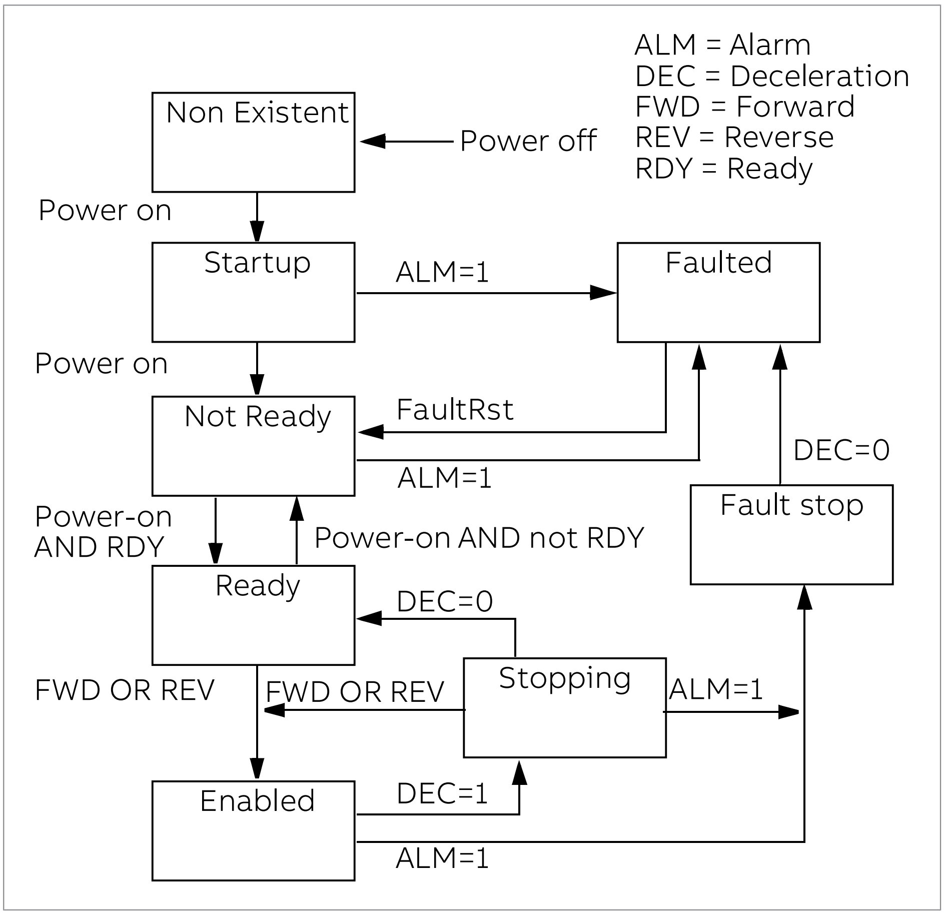
State (Control supervisor object)

This attribute indicates the current state of the Control supervisor object.

State	Description	State	Description
0	Vendor-specific	4	Enabled
1	Startup	5	Stopping

State	Description	State	Description
2	Not ready	6	Fault stop
3	Ready	7	Faulted

The ODVA state transition diagram is shown below:



Speed Actual (AC/DC drive object)

This attribute indicates the actual speed at which the drive is operating. The units are scaled by the SpeedScale attribute of the AC/DC drive object. See parameter 23 Speed scale (page 31) for details.

Scalar mode

When the drive is operating in the scalar mode, the drive provides the adapter module with a frequency actual. The ODVA AC/DC drive profile uses rpm units for the speed actual. The ODVA Speed Actual is calculated as follows:

$$\text{Osa} = \frac{\text{Dfa} \times \text{Mss}}{\text{Mf} \times \text{Us}}$$

where

Osa = ODVA Speed Actual

Dfa = Drive Frequency Actual in Hz

Us = ODVA Speed Unit (see parameter 23 Speed scale (page 31))

Mf = Motor Nominal Frequency in Hz

Mss = Motor Synchronous Speed in rpm (not Motor Nominal Speed).

For example, for a 4-pole 60 Hz motor (Mss = 1800 rpm) with a unit of 1 rpm and a Drive Frequency Actual of 30 Hz, the ODVA Speed Actual is:

$$\mathrm{Osa} = \frac{\mathrm{Dfa} \times \mathrm{Mss}}{\mathrm{Mf} \times \mathrm{Us}} = \frac{30\mathrm{Hz} \times 1800\mathrm{rpm}}{60\mathrm{Hz} \times 1\mathrm{rpm}} = 900$$

Vector mode

When the drive is operating in the vector mode, the drive provides the adapter module with a speed actual. The ODVA AC/DC drive profile uses rpm units for the speed actual. The ODVA Speed Actual is calculated as follows:

$$\mathrm{Osa} = \frac{\mathrm{Dsa}}{\mathrm{Us}}$$

where

Dsa = Drive Speed Actual in rpm

Osa = ODVA Speed Actual

Us = ODVA Speed Unit (see parameter 23 Speed scale (page 31)).

For example, for a Drive Speed Actual of 900 rpm with a unit of 0.5 rpm, the ODVA Speed Actual is:

$$\mathrm{Osa} = \frac{\mathrm{Dsa}}{\mathrm{Us}} = \frac{450\mathrm{rpm}}{0.5\mathrm{rpm}} = 900$$

Torque Actual (AC/DC drive object)

This attribute indicates the actual torque at which the drive is operating. The units are scaled by the Torque Scale attribute of the AC/DC drive object. See parameter 24 Torque scale (page 32) for details.

The drive provides the adapter module with a torque actual in percent of the Motor Nominal Torque. The ODVA AC/DC drive profile uses Newton-meter (N·m) units for the torque actual. The ODVA Torque Actual is calculated as follows:

$$\mathrm{Ota} = \frac{\mathrm{Dta} \times \mathrm{Mt}}{100 \times \mathrm{Ut}}$$

where

Dta = Drive Torque Actual in Percent of Motor Nominal Torque

Ota = ODVA Torque Actual

Ut = ODVA Torque Unit (see parameter 24 Torque scale (page 32))

Mt = Motor Nominal Torque in N·m.

For example, for a 1000 N·m Motor Nominal Torque with a unit of 1 N·m and a drive torque actual of 50%, the ODVA Torque Actual is:

$$\mathrm{Ota} = \frac{\mathrm{Dta} \times \mathrm{Mt}}{100 \times \mathrm{Ut}} = \frac{50 \times 1000\mathrm{Nm}}{100 \times 1\mathrm{Nm}} = 500$$

ABB Drives communication profile

Control word and Status word

The Control word is the principal means for controlling the drive from a fieldbus system. It is sent by the fieldbus client station to the drive through the adapter module. The drive switches between its states according to the bit-coded instructions in the Control word and returns status information to the client in the Status word.

The contents of the Control word and the Status word are detailed below. The drive states are presented in section State machine (page 64).

Control word contents

The table below shows the contents of the Control word for the ABB Drives communication profile. The upper case boldface text refers to the states shown in section State machine (page 64).

Bit	Name	Value	STATE/Description
0	OFF1_CONTROL	1	Proceed to READY TO OPERATE.
		0	Stop along currently active deceleration ramp. Proceed to OFF1 ACTIVE; proceed to READY TO SWITCH ON unless other interlocks (OFF2, OFF3) are active.
1	OFF2_CONTROL	1	Continue operation (OFF2 inactive).
		0	Emergency OFF, coast to stop. Proceed to OFF2 ACTIVE, proceed to SWITCH-ON INHIBITED.
2	OFF3_CONTROL	1	Continue operation (OFF3 inactive).
		0	Emergency stop, stop within time defined by drive parameter. Proceed to OFF3 ACTIVE; proceed to SWITCH-ON INHIBITED. ⚠ **WARNING!** Ensure that motor and driven machine can be stopped using this stop mode.
3	INHIBIT_OPERATION	1	Proceed to OPERATION ENABLED. **Note:** Run enable signal must be active; see drive documentation. If the drive is set to receive the Run enable signal from the fieldbus, this bit activates the signal.
		0	Inhibit operation. Proceed to OPERATION INHIBITED.
4	RAMP_OUT_ZERO	1	Normal operation. Proceed to RAMP FUNCTION GENERATOR: OUTPUT ENABLED.
		0	Force Ramp Function Generator output to zero. Drive ramps to stop (current and DC voltage limits in force).
5	RAMP_HOLD	1	Enable ramp function. Proceed to RAMP FUNCTION GENERATOR: ACCELERATOR ENABLED.
		0	Halt ramping (Ramp Function Generator output held).
6	RAMP_IN_ZERO	1	Normal operation. Proceed to OPERATION. **Note:** This bit is effective only if the fieldbus interface is set as the source for this signal by drive parameters.
		0	Force Ramp Function Generator input to zero.

Bit	Name	Value	STATE/Description
7	RESET	0 → 1	Fault reset if an active fault exists. Proceed to SWITCH-ON INHIBITED. **Note:** This bit is effective only if the fieldbus interface is set as the source for this signal by drive parameters.
		0	Continue normal operation.
8 … 9	Drive-specific (For information, see the drive documentation.)		
10	REMOTE_CMD	1	Fieldbus control enabled.
		0	Control word and reference not getting through to the drive, except for CW bits OFF1, OFF2 and OFF3.
11	EXT_CTRL_LOC	1	Select External Control Location EXT2. Effective if control location parameterized to be selected from fieldbus.
		0	Select External Control Location EXT1. Effective if control location parameterized to be selected from fieldbus.
12 … 15	Reserved		

Status word contents

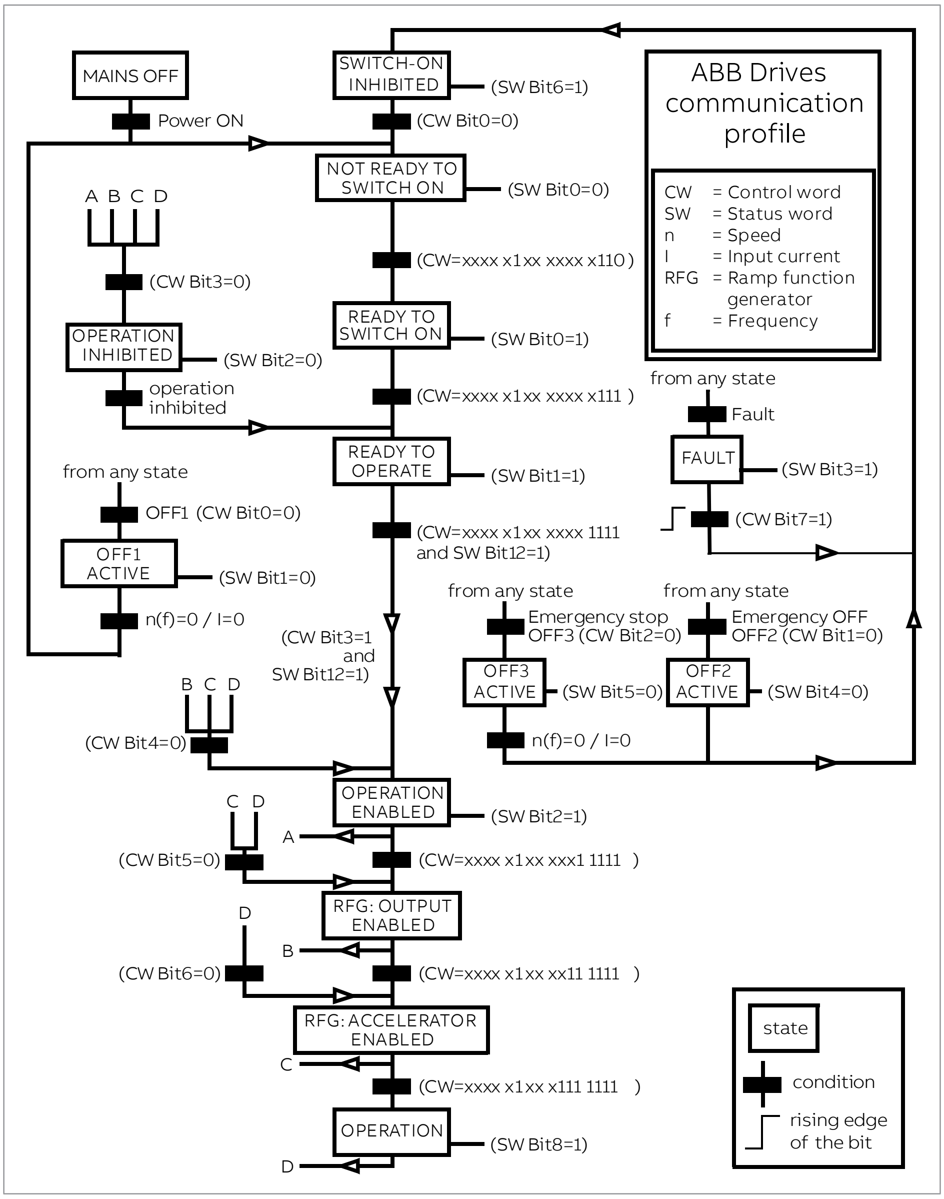
The table below shows the contents of the Status word for the ABB Drives communication profile. The upper case boldface text refers to the states shown in section State machine (page 64).

Bit	Name	Value	STATE/Description
0	RDY_ON	1	READY TO SWITCH ON
		0	NOT READY TO SWITCH ON
1	RDY_RUN	1	READY TO OPERATE
		0	OFF1 ACTIVE
2	RDY_REF	1	OPERATION ENABLED
		0	OPERATION INHIBITED
3	TRIPPED	1	FAULT
		0	No fault
4	OFF_2_STA	1	OFF2 inactive
		0	OFF2 ACTIVE
5	OFF_3_STA	1	OFF3 inactive
		0	OFF3 ACTIVE
6	SWC_ON_INHIB	1	SWITCH-ON INHIBITED
		0	–
7	ALARM	1	Warning/Alarm
		0	No warning/alarm

Bit	Name	Value	STATE/Description
8	AT_SETPOINT	1	OPERATION. Actual value equals reference (= is within tolerance limits, ie, in speed control, speed error is 10% max. of nominal motor speed).
		0	Actual value differs from reference (= is outside tolerance limits).
9	REMOTE	1	Drive control location: REMOTE (EXT1 or EXT2)
		0	Drive control location: LOCAL
10	ABOVE_LIMIT	1	Actual frequency or speed equals or exceeds supervision limit (set by drive parameter). Valid in both directions of rotation.
		0	Actual frequency or speed within supervision limit
11	EXT_CTRL_LOC	1	External Control Location EXT2 selected. **Note:** For ACS880: This bit is effective only if the fieldbus interface is set as the target for this signal by drive parameters. User bit 0 selection (06.33).
		0	External Control Location EXT1 selected
12	EXT_RUN_ENABLE	1	External Run Enable signal received. **Note:** For ACS880: This bit is effective only if the fieldbus interface is set as the target for this signal by drive parameters. User bit 1 selection (06.34)
		0	No External Run Enable signal received
13 … 14	Reserved		
15	FBA_ERROR	1	Communication error detected by fieldbus adapter module
		0	Fieldbus adapter communication OK

State machine

The state machine for the ABB Drives communication profile is shown below.



References

References are 16-bit words containing a sign bit and a 15-bit integer. A negative reference (indicating reversed direction of rotation) is formed by calculating the two’s complement from the corresponding positive reference.

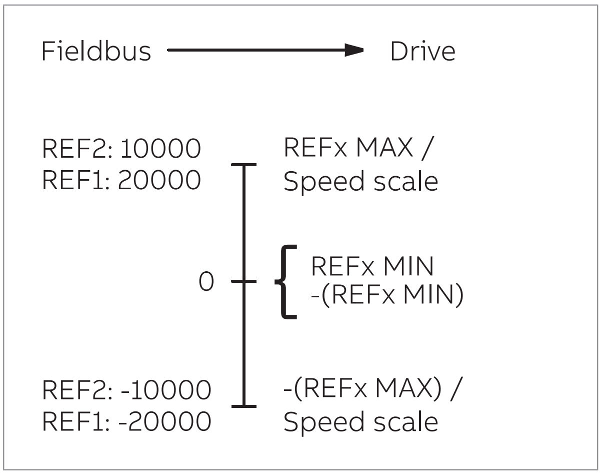
ABB drives can receive control information from multiple sources including analog and digital inputs, the drive control panel and a fieldbus adapter module (for example, FEIP-21). To have the drive controlled through the fieldbus, you must select the module as the source for control information, for example, reference.

Scaling

References are scaled as shown below.

Note: The values of REF1 MAX and REF2 MAX are set with drive parameters. See the drive manuals for further information.

In ACS380, ACS480, ACH580, ACQ580, ACS580 and ACS880, the speed reference (REFx) in decimal (0…20000) corresponds to 0…100% of the speed scaling value (as defined with a drive parameter).



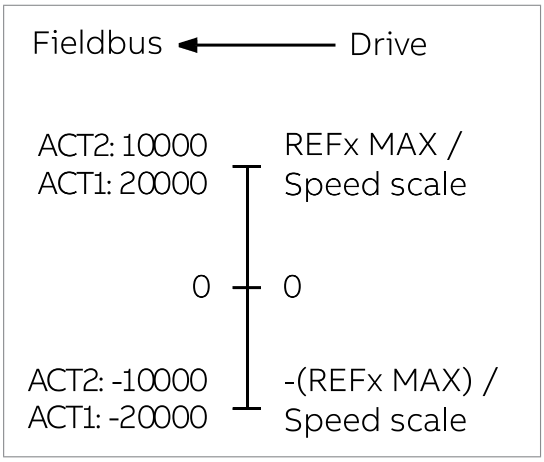
Actual values

Actual values are 16-bit words containing information on the operation of the drive. The functions to be monitored are selected with a drive parameter.

Scaling

Actual values are scaled as shown below.

Note: The values of REF1 MAX and REF2 MAX are set with drive parameters. See the drive manuals for further information.



8

EtherNet/IP – Communication protocol

Contents of this chapter

This chapter describes the EtherNet/IP communication protocol for the adapter module.

EtherNet/IP

EtherNet/IP is a variant of the Common Industrial Protocol (CIP) family of communication protocols intended for supervision and control of automation equipment. Specifically, it covers the use of CIP messaging over an IP network, typically using Ethernet as the media.

The FEIP-21 adapter module acts as a server on an EtherNet/IP network with support for the ODVA AC/DC drive, ABB Drives and Transparent profiles. Two simultaneous EtherNet/IP connections are supported, that is, two clients can be connected to the adapter module at a time.

Further information on the EtherNet/IP protocol is available at www.odva.org.

Object modeling and functional profiles

One of the main features of EtherNet/IP is object modeling. A group of objects can be described with a Functional Profile. The FEIP-21 adapter module realizes the ODVA AC/DC drive Functional Profile with additional features.

Assembly objects

I/O assembly instances may also be referred to as Block Transfer of data. Intelligent devices realizing a Functional Profile, such as FEIP-21, have several objects. Since it is not possible to transmit more than one object data through a single connection, it is practical and more efficient to group attributes from different objects into a single

I/O connection using the assembly object. The assembly object acts as a tool for grouping these attributes.

The assembly selections described above are, in fact, instances of the assembly object class. The adapter module uses static assemblies (in other words, fixed groupings of different object data only). The following tables describe the assembly instances supported by the adapter module.

Basic speed control assembly

The Basic speed control assembly is defined by the ODVA AC/DC drive profile. The format of the output assembly is:

Instance 20 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0						Fault Reset		Run Fwd
1								
2	Speed Reference (Low Byte)							
3	Speed Reference (High Byte)							

The format of the input assembly is:

Instance 70 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0						Running1 (Fwd)		Faulted
1								
2	Speed Actual (Low Byte)							
3	Speed Actual (High Byte)							

Basic speed control plus drive parameters assembly

The Basic speed control plus drive parameters assembly, defined by ABB, adds configurable drive parameters to the Basic speed control assembly of the ODVA AC/DC drive profile.

The format of the output assembly is:

Instance 120 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0						Fault reset		Run Fwd
1								
2	Speed Actual (Low Byte)							
3	Speed Actual (High Byte)							
4	DATA OUT 1 Value (Low Byte)							

Instance 120 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
5	DATA OUT 1 Value (High Byte)							
6	DATA OUT 2 Value (Low Byte)							
7	DATA OUT 2 Value (High Byte)							
8	DATA OUT 3 Value (Low Byte)							
9	DATA OUT 3 Value (High Byte)							
10	DATA OUT 4 Value (Low Byte)							
11	DATA OUT 4 Value (High Byte)							
12	DATA OUT 5 Value (Low Byte)							
13	DATA OUT 5 Value (High Byte)							
14	DATA OUT 6 Value (Low Byte)							
15	DATA OUT 6 Value (High Byte)							
16	DATA OUT 7 Value (Low Byte)							
17	DATA OUT 7 Value (High Byte)							
18	DATA OUT 8 Value (Low Byte)							
19	DATA OUT 8 Value (High Byte)							
20	DATA OUT 9 Value (Low Byte)							
21	DATA OUT 9 Value (High Byte)							
22	DATA OUT 10 Value (Low Byte)							
23	DATA OUT 10 Value (High Byte)							

The format of the input assembly is:

Instance 170 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0						Running1 (Fwd)		Faulted
1								
2	Speed Actual (Low Byte)							
3	Speed Actual (High Byte)							
4	DATA IN 1 Value (Low Byte)							
5	DATA IN 1 Value (High Byte)							
6	DATA IN 2 Value (Low Byte)							
7	DATA IN 2 Value (High Byte)							
8	DATA IN 3 Value (Low Byte)							

Instance 170 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
9	DATA IN 3 Value (High Byte)							
10	DATA IN 4 Value (Low Byte)							
11	DATA IN 4 Value (High Byte)							
12	DATA IN 5 Value (Low Byte)							
13	DATA IN 5 Value (High Byte)							
14	DATA IN 6 Value (Low Byte)							
15	DATA IN 6 Value (High Byte)							
16	DATA IN 7 Value (Low Byte)							
17	DATA IN 7 Value (High Byte)							
18	DATA IN 8 Value (Low Byte)							
19	DATA IN 8 Value (High Byte)							
20	DATA IN 9 Value (Low Byte)							
21	DATA IN 9 Value (High Byte)							
22	DATA IN 10 Value (Low Byte)							
23	DATA IN 10 Value (High Byte)							

Extended speed control assembly

The Extended speed control assembly is defined by the ODVA AC/DC drive profile. The format of the output assembly is:

Instance 21 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0		NetRef	NetCtrl			Fault Reset	Run Rev	Run Fwd
1								
2	Speed Reference (Low Byte)							
3	Speed Reference (High Byte)							

The format of the input assembly is:

Instance 71 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	At Reference	Ref From Net	Ctrl From Net	Ready	Running2 (Rev)	Running1 (Fwd)	Warning	Faulted
1	Drive State. See section State (Control supervisor object) (page 59).							

Instance 71 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
2	Speed Actual (Low Byte)							
3	Speed Actual (High Byte)							

Extended speed control plus drive parameters assembly

The Extended speed control plus drive parameters assembly, defined by ABB, adds configurable drive parameters to the Extended speed control assembly of the ODVA AC/DC drive profile.

The format of the output assembly is:

Instance 121 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0		Net Ref	Net Ctrl			Fault reset	Run Rev	Run Fwd
1								
2	Speed Actual (Low Byte)							
3	Speed Actual (High Byte)							
4	DATA OUT 1 Value (Low Byte)							
5	DATA OUT 1 Value (High Byte)							
6	DATA OUT 2 Value (Low Byte)							
7	DATA OUT 2 Value (High Byte)							
8	DATA OUT 3 Value (Low Byte)							
9	DATA OUT 3 Value (High Byte)							
10	DATA OUT 4 Value (Low Byte)							
11	DATA OUT 4 Value (High Byte)							
12	DATA OUT 5 Value (Low Byte)							
13	DATA OUT 5 Value (High Byte)							
14	DATA OUT 6 Value (Low Byte)							
15	DATA OUT 6 Value (High Byte)							
16	DATA OUT 7 Value (Low Byte)							
17	DATA OUT 7 Value (High Byte)							
18	DATA OUT 8 Value (Low Byte)							
19	DATA OUT 8 Value (High Byte)							
20	DATA OUT 9 Value (Low Byte)							
21	DATA OUT 9 Value (High Byte)							
22	DATA OUT 10 Value (Low Byte)							

Instance 121 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
23	DATA OUT 10 Value (High Byte)							

The format of the input assembly is:

Instance 171 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	At Reference	Ref From Net	Ctrl From Net	Ready	Running2 (Rev)	Running1 (Fwd)	Warning	Faulted
1	Drive State See section State (Control supervisor object) (page 59).							
2	Speed Actual (Low Byte)							
3	Speed Actual (High Byte)							
4	DATA IN 1 Value (Low Byte)							
5	DATA IN 1 Value (High Byte)							
6	DATA IN 2 Value (Low Byte)							
7	DATA IN 2 Value (High Byte)							
8	DATA IN 3 Value (Low Byte)							
9	DATA IN 3 Value (High Byte)							
10	DATA IN 4 Value (Low Byte)							
11	DATA IN 4 Value (High Byte)							
12	DATA IN 5 Value (Low Byte)							
13	DATA IN 5 Value (High Byte)							
14	DATA IN 6 Value (Low Byte)							
15	DATA IN 6 Value (High Byte)							
16	DATA IN 7 Value (Low Byte)							
17	DATA IN 7 Value (High Byte)							
18	DATA IN 8 Value (Low Byte)							
19	DATA IN 8 Value (High Byte)							
20	DATA IN 9 Value (Low Byte)							
21	DATA IN 9 Value (High Byte)							
22	DATA IN 10 Value (Low Byte)							
23	DATA IN 10 Value (High Byte)							

Basic speed and torque control assembly

The Basic speed and torque control assembly is defined by the ODVA AC/DC drive profile. The format of the output assembly is:

Instance 22 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0						Fault Re-set		Run Fwd
1								
2	Speed Reference (Low Byte)							
3	Speed Reference (High Byte)							
4	Torque Reference (Low Byte)							
5	Torque Reference (High Byte)							

The format of the input assembly is:

Instance 72 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0						Running1 (Fwd)		Faulted
1								
2	Speed Actual (Low Byte)							
3	Speed Actual (High Byte)							
4	Torque Actual (Low Byte)							
5	Torque Actual (High Byte)							

Basic speed and torque control plus drive parameters assembly

The Basic speed and torque control plus drive parameters assembly, defined by ABB, adds configurable drive parameters to the Basic speed and torque control assembly of the ODVA AC/DC drive profile.

The format of the output assembly is:

Instance 122 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0						Fault reset		Run Fwd
1								
2	Speed Reference (Low Byte)							
3	Speed Reference (High Byte)							
4	Torque Reference (Low Byte)							
5	Torque Reference (High Byte)							

Instance 122 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
6	DATA OUT 1 Value (Low Byte)							
7	DATA OUT 1 Value (High Byte)							
8	DATA OUT 2 Value (Low Byte)							
9	DATA OUT 2 Value (High Byte)							
10	DATA OUT 3 Value (Low Byte)							
11	DATA OUT 3 Value (High Byte)							
12	DATA OUT 4 Value (Low Byte)							
13	DATA OUT 4 Value (High Byte)							
14	DATA OUT 5 Value (Low Byte)							
15	DATA OUT 5 Value (High Byte)							
16	DATA OUT 6 Value (Low Byte)							
17	DATA OUT 6 Value (High Byte)							
18	DATA OUT 7 Value (Low Byte)							
19	DATA OUT 7 Value (High Byte)							
20	DATA OUT 8 Value (Low Byte)							
21	DATA OUT 8 Value (High Byte)							
22	DATA OUT 9 Value (Low Byte)							
23	DATA OUT 9 Value (High Byte)							
24	DATA OUT 10 Value (Low Byte)							
25	DATA OUT 10 Value (High Byte)							

The format of the input assembly is:

Instance 172 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0						Running1 (Fwd)		Faulted
1								
2	Speed Actual (Low Byte)							
3	Speed Actual (High Byte)							
4	Torque Actual (Low Byte)							
5	Torque Actual (High Byte)							
6	DATA IN 1 Value (Low Byte)							
7	DATA IN 1 Value (High Byte)							

Instance 172 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
8	DATA IN 2 Value (Low Byte)							
9	DATA IN 2 Value (High Byte)							
10	DATA IN 3 Value (Low Byte)							
11	DATA IN 3 Value (High Byte)							
12	DATA IN 4 Value (Low Byte)							
13	DATA IN 4 Value (High Byte)							
14	DATA IN 5 Value (Low Byte)							
15	DATA IN 5 Value (High Byte)							
16	DATA IN 6 Value (Low Byte)							
17	DATA IN 6 Value (High Byte)							
18	DATA IN 7 Value (Low Byte)							
19	DATA IN 7 Value (High Byte)							
20	DATA IN 8 Value (Low Byte)							
21	DATA IN 8 Value (High Byte)							
22	DATA IN 9 Value (Low Byte)							
23	DATA IN 9 Value (High Byte)							
24	DATA IN 10 Value (Low Byte)							
25	DATA IN 10 Value (High Byte)							

Extended speed and torque control assembly

The Extended speed and torque control assembly is defined by the ODVA AC/DC drive profile. The format of the output assembly is:

Instance 23 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0		Net Ref	Net Ctrl			Fault Reset	Run Rev	Run Fwd
1								
2	Speed Reference (Low Byte)							
3	Speed Reference (High Byte)							
4	Torque Reference (Low Byte)							
5	Torque Reference (High Byte)							

The format of the input assembly is:

Instance 73 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	At Reference	Ref From Net	Ctrl From Net	Ready	Running2 (Rev)	Running1 (Fwd)	Warning	Faulted
1	Drive State See section State (Control supervisor object) (page 59).							
2	Speed Actual (Low Byte)							
3	Speed Actual (High Byte)							
4	Torque Actual (Low Byte)							
5	Torque Actual (High Byte)							

Extended speed and torque control plus drive parameters assembly

The Extended speed and torque control plus drive parameters assembly, defined by ABB, adds configurable drive parameters to the Extended speed and torque control assembly of the ODVA AC/DC drive profile.

The format of the output assembly is:

Instance 123 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0		NetRef	NetCtl			Fault reset	Run Rev	Run Fwd
1								
2	Speed Reference (Low Byte)							
3	Speed Reference (High Byte)							
4	Torque Reference (Low Byte)							
5	Torque Reference (High Byte)							
6	DATA OUT 1 Value (Low Byte)							
7	DATA OUT 1 Value (High Byte)							
8	DATA OUT 2 Value (Low Byte)							
9	DATA OUT 2 Value (High Byte)							
10	DATA OUT 3 Value (Low Byte)							
11	DATA OUT 3 Value (High Byte)							
12	DATA OUT 4 Value (Low Byte)							
13	DATA OUT 4 Value (High Byte)							
14	DATA OUT 5 Value (Low Byte)							
15	DATA OUT 5 Value (High Byte)							
16	DATA OUT 6 Value (Low Byte)							
17	DATA OUT 6 Value (High Byte)							

Instance 123 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
18	DATA OUT 7 Value (Low Byte)							
19	DATA OUT 7 Value (High Byte)							
20	DATA OUT 8 Value (Low Byte)							
21	DATA OUT 8 Value (High Byte)							
22	DATA OUT 9 Value (Low Byte)							
23	DATA OUT 9 Value (High Byte)							
24	DATA OUT 10 Value (Low Byte)							
25	DATA OUT 10 Value (High Byte)							

The format of the input assembly is:

Instance 173 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	At Reference	Ref From Net	Ctrl From Net	Ready	Running2 (Rev)	Running1 (Fwd)	Warning	Faulted
1	Drive State See section State (Control supervisor object) (page 59).							
2	Speed Actual (Low Byte)							
3	Speed Actual (High Byte)							
4	Torque Actual (Low Byte)							
5	Torque Actual (High Byte)							
6	DATA IN 1 Value (Low Byte)							
7	DATA IN 1 Value (High Byte)							
8	DATA IN 2 Value (Low Byte)							
9	DATA IN 2 Value (High Byte)							
10	DATA IN 3 Value (Low Byte)							
11	DATA IN 3 Value (High Byte)							
12	DATA IN 4 Value (Low Byte)							
13	DATA IN 4 Value (High Byte)							
14	DATA IN 5 Value (Low Byte)							
15	DATA IN 5 Value (High Byte)							
16	DATA IN 6 Value (Low Byte)							
17	DATA IN 6 Value (High Byte)							
18	DATA IN 7 Value (Low Byte)							

Instance 173 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
19	DATA IN 7 Value (High Byte)							
20	DATA IN 8 Value (Low Byte)							
21	DATA IN 8 Value (High Byte)							
22	DATA IN 9 Value (Low Byte)							
23	DATA IN 9 Value (High Byte)							
24	DATA IN 10 Value (Low Byte)							
25	DATA IN 10 Value (High Byte)							

ABB Drives profile with set speed assembly

The ABB Drives profile with set speed assembly is defined by ABB. The format of the output assembly is:

Instance 1 (ABB Drives profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Reset	Ramp in Zero	Ramp Hold	Ramp Out Zero	Inhibit Operation	Off 3 Control	Off 2 Control	Off 1 Control
1					Ext Ctrl Loc	Remote Cmd		
2	Set Speed (Low Byte)							
3	Set Speed (High Byte)							

The format of the input assembly is:

Instance 51 (ABB Drives profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Alarm	Swc On Inhib	Off 3 Sta	Off 2 Sta	Tripped	Rdy Ref	Rdy Run	Rdy On
1	Fieldbus Error			Ext Run Enable	Ext Ctrl Loc	Above Limit	Remote	At Setpoint
2	Actual Speed (Low Byte)							
3	Actual Speed (High Byte)							

ABB Drives profile with set speed plus drive parameters assembly

The ABB Drives profile with set speed plus drive parameters assembly, defined by ABB, adds configurable drive parameters to the ABB Drives profile with set speed of the ABB Drives profile.

The format of the output assembly is:

Instance 101 (ABB Drives profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Reset	Ramp in Zero	Ramp Hold	Ramp Out Zero	Inhibit Operation	Off 3 Control	Off 2 Control	Off 1 Control
1					Ext Ctrl Loc	Remote Cmd		
2	Set Speed (Low Byte)							
3	Set Speed (High Byte)							
4	DATA OUT 1 Value (Low Byte)							
5	DATA OUT 1 Value (High Byte)							
6	DATA OUT 2 Value (Low Byte)							
7	DATA OUT 2 Value (High Byte)							
8	DATA OUT 3 Value (Low Byte)							
9	DATA OUT 3 Value (High Byte)							
10	DATA OUT 4 Value (Low Byte)							
11	DATA OUT 4 Value (High Byte)							
12	DATA OUT 5 Value (Low Byte)							
13	DATA OUT 5 Value (High Byte)							
14	DATA OUT 6 Value (Low Byte)							
15	DATA OUT 6 Value (High Byte)							
16	DATA OUT 7 Value (Low Byte)							
17	DATA OUT 7 Value (High Byte)							
18	DATA OUT 8 Value (Low Byte)							
19	DATA OUT 8 Value (High Byte)							
20	DATA OUT 9 Value (Low Byte)							
21	DATA OUT 9 Value (High Byte)							
22	DATA OUT 10 Value (Low Byte)							
23	DATA OUT 10 Value (High Byte)							

The format of the input assembly is:

Instance 151 (ABB Drives profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Alarm	Swc On Inhib	Off 3 Sta	Off 2 Sta	Tripped	Rdy Ref	Rdy Run	Rdy On
1	Fieldbus Error			Ext Run Enable	Ext Ctrl Loc	Above Limit	Remote	At Setpoint

Instance 151 (ABB Drives profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
2	Actual Speed (Low Byte)							
3	Actual Speed (High Byte)							
4	DATA IN 1 Value (Low Byte)							
5	DATA IN 1 Value (High Byte)							
6	DATA IN 2 Value (Low Byte)							
7	DATA IN 2 Value (High Byte)							
8	DATA IN 3 Value (Low Byte)							
9	DATA IN 3 Value (High Byte)							
10	DATA IN 4 Value (Low Byte)							
11	DATA IN 4 Value (High Byte)							
12	DATA IN 5 Value (Low Byte)							
13	DATA IN 5 Value (High Byte)							
14	DATA IN 6 Value (Low Byte)							
15	DATA IN 6 Value (High Byte)							
16	DATA IN 7 Value (Low Byte)							
17	DATA IN 7 Value (High Byte)							
18	DATA IN 8 Value (Low Byte)							
19	DATA IN 8 Value (High Byte)							
20	DATA IN 9 Value (Low Byte)							
21	DATA IN 9 Value (High Byte)							
22	DATA IN 10 Value (Low Byte)							
23	DATA IN 10 Value (High Byte)							

ABB Drives profile with set speed and set torque assembly

The ABB Drives profile with set speed and set torque assembly is defined by ABB. The format of the output assembly is:

Instance 2 (ABB Drives profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Reset	Ramp in Zero	Ramp Hold	Ramp Out Zero	Inhibit Operation	Off 3 Control	Off 2 Control	Off 1 Control
1					Ext Ctrl Loc	Remote Cmd		
2	Set Speed (Low Byte)							
3	Set Speed (High Byte)							

Instance 2 (ABB Drives profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
4	Set Torque (Low Byte)							
5	Set Torque (High Byte)							

The format of the input assembly is:

Instance 52 (ABB Drives profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Alarm	Swc On Inhib	Off 3 Sta	Off 2 Sta	Tripped	Rdy Ref	Rdy Run	Rdy On
1	Fieldbus Error			Ext Run Enable	Ext Ctrl Loc	Above Limit	Remote	At Setpoint
2	Actual Speed (Low Byte)							
3	Actual Speed (High Byte)							
4	Actual Torque (Low Byte)							
5	Actual Torque (High Byte)							

ABB Drives profile with set speed and set torque plus drive parameters assembly

The ABB Drives profile with set speed and set torque plus drive parameters assembly, defined by ABB, adds configurable drive parameters to the ABB Drives profile with set speed and set torque of the ABB Drives profile.

The format of the output assembly is:

Instance 102 (ABB Drives profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Reset	Ramp in Zero	Ramp Hold	Ramp Out Zero	Inhibit Operation	Off 3 Control	Off 2 Control	Off 1 Control
1					Ext Ctrl Loc	Remote Cmd		
2	Set Speed (Low Byte)							
3	Set Speed (High Byte)							
4	Set Torque (Low Byte)							
5	Set Torque (High Byte)							
6	DATA OUT 1 Value (Low Byte)							
7	DATA OUT 1 Value (High Byte)							
8	DATA OUT 2 Value (Low Byte)							
9	DATA OUT 2 Value (High Byte)							
10	DATA OUT 3 Value (Low Byte)							

Instance 102 (ABB Drives profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
11	DATA OUT 3 Value (High Byte)							
12	DATA OUT 4 Value (Low Byte)							
13	DATA OUT 4 Value (High Byte)							
14	DATA OUT 5 Value (Low Byte)							
15	DATA OUT 5 Value (High Byte)							
16	DATA OUT 6 Value (Low Byte)							
17	DATA OUT 6 Value (High Byte)							
18	DATA OUT 7 Value (Low Byte)							
19	DATA OUT 7 Value (High Byte)							
20	DATA OUT 8 Value (Low Byte)							
21	DATA OUT 8 Value (High Byte)							
22	DATA OUT 9 Value (Low Byte)							
23	DATA OUT 9 Value (High Byte)							
24	DATA OUT 10 Value (Low Byte)							
25	DATA OUT 10 Value (High Byte)							

The format of the input assembly is:

Instance 152 (ABB Drives profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Alarm	Swc On Inhib	Off 3 Sta	Off 2 Sta	Tripped	Rdy Ref	Rdy Run	Rdy On
1	Fieldbus Error			Ext Run Enable	Ext Ctrl Loc	Above Limit	Remote	At Setpoint
2	Actual Speed (Low Byte)							
3	Actual Speed (High Byte)							
4	Actual Torque (Low Byte)							
5	Actual Torque (High Byte)							
6	DATA IN 1 Value (Low Byte)							
7	DATA IN 1 Value (High Byte)							
8	DATA IN 2 Value (Low Byte)							
9	DATA IN 2 Value (High Byte)							
10	DATA IN 3 Value (Low Byte)							
11	DATA IN 3 Value (High Byte)							

Instance 152 (ABB Drives profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
12	DATA IN 4 Value (Low Byte)							
13	DATA IN 4 Value (High Byte)							
14	DATA IN 5 Value (Low Byte)							
15	DATA IN 5 Value (High Byte)							
16	DATA IN 6 Value (Low Byte)							
17	DATA IN 6 Value (High Byte)							
18	DATA IN 7 Value (Low Byte)							
19	DATA IN 7 Value (High Byte)							
20	DATA IN 8 Value (Low Byte)							
21	DATA IN 8 Value (High Byte)							
22	DATA IN 9 Value (Low Byte)							
23	DATA IN 9 Value (High Byte)							
24	DATA IN 10 Value (Low Byte)							
25	DATA IN 10 Value (High Byte)							

Transparent 16 with one assembly

The Transparent 16 with one assembly, defined by ABB, provides unaltered 16-bit access to the configured drive profile.

The format of the output assembly is:

Instance 11 (Transparent 16 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Drive Profile 16-bit Control Word (Low Byte)							
1	Drive Profile 16-bit Control Word (High Byte)							
2	Drive Profile 16-bit Reference 1 Word (Low Byte)							
3	Drive Profile 16-bit Reference 1 Word (High Byte)							

The format of the input assembly is:

Instance 61 (Transparent 16 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Drive Profile 16-bit Status Word (Low Byte)							
1	Drive Profile 16-bit Status Word (High Byte)							
2	Drive Profile 16-bit Actual 1 Word (Low Byte)							
3	Drive Profile 16-bit Actual 1 Word (High Byte)							

Transparent 16 with one assembly plus drive parameters

The Transparent 16 with one assembly plus drive parameters, defined by ABB, adds configurable drive parameters to the Transparent 16 with one assembly.

The format of the output assembly is:

Instance 111 (Transparent 16 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Drive Profile 16-bit Control Word (Low Byte)							
1	Drive Profile 16-bit Control Word (High Byte)							
2	Drive Profile 16-bit Reference 1 Word (Low Byte)							
3	Drive Profile 16-bit Reference 1 Word (High Byte)							
4	DATA OUT 1 Value (Low Byte)							
5	DATA OUT 1 Value (High Byte)							
6	DATA OUT 2 Value (Low Byte)							
7	DATA OUT 2 Value (High Byte)							
8	DATA OUT 3 Value (Low Byte)							
9	DATA OUT 3 Value (High Byte)							
10	DATA OUT 4 Value (Low Byte)							
11	DATA OUT 4 Value (High Byte)							
12	DATA OUT 5 Value (Low Byte)							
13	DATA OUT 5 Value (High Byte)							
14	DATA OUT 6 Value (Low Byte)							
15	DATA OUT 6 Value (High Byte)							
16	DATA OUT 7 Value (Low Byte)							
17	DATA OUT 7 Value (High Byte)							
18	DATA OUT 8 Value (Low Byte)							
19	DATA OUT 8 Value (High Byte)							
20	DATA OUT 9 Value (Low Byte)							
21	DATA OUT 9 Value (High Byte)							
22	DATA OUT 10 Value (Low Byte)							
23	DATA OUT 10 Value (High Byte)							

The format of the input assembly is:

Instance 161 (Transparent 16 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Drive Profile 16-bit Status Word (Low Byte)							

Instance 161 (Transparent 16 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
1	Drive Profile 16-bit Status Word (High Byte)							
2	Drive Profile 16-bit Actual 1 Word (Low Byte)							
3	Drive Profile 16-bit Actual 1 Word (High Byte)							
4	DATA IN 1 Value (Low Byte)							
5	DATA IN 1 Value (High Byte)							
6	DATA IN 2 Value (Low Byte)							
7	DATA IN 2 Value (High Byte)							
8	DATA IN 3 Value (Low Byte)							
9	DATA IN 3 Value (High Byte)							
10	DATA IN 4 Value (Low Byte)							
11	DATA IN 4 Value (High Byte)							
12	DATA IN 5 Value (Low Byte)							
13	DATA IN 5 Value (High Byte)							
14	DATA IN 6 Value (Low Byte)							
15	DATA IN 6 Value (High Byte)							
16	DATA IN 7 Value (Low Byte)							
17	DATA IN 7 Value (High Byte)							
18	DATA IN 8 Value (Low Byte)							
19	DATA IN 8 Value (High Byte)							
20	DATA IN 9 Value (Low Byte)							
21	DATA IN 9 Value (High Byte)							
22	DATA IN 10 Value (Low Byte)							
23	DATA IN 10 Value (High Byte)							

Transparent 16 with two assembly

The Transparent 16 with two assembly, defined by ABB, provides unaltered 16-bit access to the configured drive profile.

The format of the output assembly is:

Instance 12 (Transparent 16 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Drive Profile 16-bit Control Word (Low Byte)							
1	Drive Profile 16-bit Control Word (High Byte)							
2	Drive Profile 16-bit Reference 1 Word (Low Byte)							

Instance 12 (Transparent 16 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
3	Drive Profile 16-bit Reference 1 Word (High Byte)							
4	Drive Profile 16-bit Reference 2 Word (Low Byte)							
5	Drive Profile 16-bit Reference 2 Word (High Byte)							

The format of the input assembly is:

Instance 62 (Transparent 16 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Drive Profile 16-bit Status Word (Low Byte)							
1	Drive Profile 16-bit Status Word (High Byte)							
2	Drive Profile 16-bit Actual 1 Word (Low Byte)							
3	Drive Profile 16-bit Actual 1 Word (High Byte)							
4	Drive Profile16-bit Actual 2 Word (Low Byte)							
5	Drive Profile 16-bit Actual 2 Word (High Byte)							

Transparent 16 with two assembly plus drive parameters

The Transparent 16 with two assembly plus drive parameters, defined by ABB, adds configurable drive parameters to the Transparent 16 with two assembly.

The format of the output assembly is:

Instance 112 (Transparent 16 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Drive Profile 16-bit Control Word (Low Byte)							
1	Drive Profile 16-bit Control Word (High Byte)							
2	Drive Profile 16-bit Reference 1 Word (Low Byte)							
3	Drive Profile 16-bit Reference 1 Word (High Byte)							
4	Drive Profile 16-bit Reference 2 Word (Low Byte)							
5	Drive Profile 16-bit Reference 2 Word (High Byte)							
6	DATA OUT 1 Value (Low Byte)							
7	DATA OUT 1 Value (High Byte)							
8	DATA OUT 2 Value (Low Byte)							
9	DATA OUT 2 Value (High Byte)							
10	DATA OUT 3 Value (Low Byte)							
11	DATA OUT 3 Value (High Byte)							
12	DATA OUT 4 Value (Low Byte)							

Instance 112 (Transparent 16 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
13	DATA OUT 4 Value (High Byte)							
14	DATA OUT 5 Value (Low Byte)							
15	DATA OUT 5 Value (High Byte)							
16	DATA OUT 6 Value (Low Byte)							
17	DATA OUT 6 Value (High Byte)							
18	DATA OUT 7 Value (Low Byte)							
19	DATA OUT 7 Value (High Byte)							
20	DATA OUT 8 Value (Low Byte)							
21	DATA OUT 8 Value (High Byte)							
22	DATA OUT 9 Value (Low Byte)							
23	DATA OUT 9 Value (High Byte)							
24	DATA OUT 10 Value (Low Byte)							
25	DATA OUT 10 Value (High Byte)							

Instance 162 (Transparent 16 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Drive Profile 16-bit Status Word (Low Byte)							
1	Drive Profile 16-bit Status Word (High Byte)							
2	Drive Profile 16-bit Actual 1 Word (Low Byte)							
3	Drive Profile 16-bit Actual 1 Word (High Byte)							
4	Drive Profile 16-bit Actual 2 Word (Low Byte)							
5	Drive Profile 16-bit Actual 2 Word (High Byte)							
6	DATA IN 1 Value (Low Byte)							
7	DATA IN 1 Value (High Byte)							
8	DATA IN 2 Value (Low Byte)							
9	DATA IN 2 Value (High Byte)							
10	DATA IN 3 Value (Low Byte)							
11	DATA IN 3 Value (High Byte)							
12	DATA IN 4 Value (Low Byte)							
13	DATA IN 4 Value (High Byte)							
14	DATA IN 5 Value (Low Byte)							
15	DATA IN 5 Value (High Byte)							
16	DATA IN 6 Value (Low Byte)							

Instance 162 (Transparent 16 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
17	DATA IN 6 Value (High Byte)							
18	DATA IN 7 Value (Low Byte)							
19	DATA IN 7 Value (High Byte)							
20	DATA IN 8 Value (Low Byte)							
21	DATA IN 8 Value (High Byte)							
22	DATA IN 9 Value (Low Byte)							
23	DATA IN 9 Value (High Byte)							
24	DATA IN 10 Value (Low Byte)							
25	DATA IN 10 Value (High Byte)							

Transparent 32 with one assembly

The Transparent 32 with one assembly, defined by ABB, provides unaltered 32-bit access to the configured drive profile.

The format of the output assembly is:

Instance 21 (Transparent 32 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Drive Profile 32-bit Control Word (Low Byte)							
1	Drive Profile 32-bit Control Word							
2	Drive Profile 32-bit Control Word							
3	Drive Profile 32-bit Control Word (High Byte)							
4	Drive Profile 32-bit Reference 1 Word (Low Byte)							
5	Drive Profile 32-bit Reference 1 Word							
6	Drive Profile 32-bit Reference 1 Word							
7	Drive Profile 32-bit Reference 1 Word (High Byte)							

The format of the input assembly is:

Instance 71 (Transparent 32 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Drive Profile 32-bit Status Word (Low Byte)							
1	Drive Profile 32-bit Status Word							
2	Drive Profile 32-bit Status Word							
3	Drive Profile 32-bit Status Word (High Byte)							
4	Drive Profile 32-bit Actual 1 Word (Low Byte)							

Instance 71 (Transparent 32 profile)								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
5	Drive Profile 32-bit Actual 1 Word							
6	Drive Profile 32-bit Actual 1 Word							
7	Drive Profile 32-bit Actual 1 Word (High Byte)							

Transparent 32 with one assembly plus drive parameters

The Transparent 32 with one assembly plus drive parameters, defined by ABB, adds configurable drive parameters to the Transparent 32 with one assembly.

The format of the output assembly is:

Instance 121 (Transparent 32 profile)								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Drive Profile 32-bit Control Word (Low Byte)							
1	Drive Profile 32-bit Control Word							
2	Drive Profile 32-bit Control Word							
3	Drive Profile 32-bit Control Word (High Byte)							
4	Drive Profile 32-bit Reference 1 Word (Low Byte)							
5	Drive Profile 32-bit Reference 1 Word							
6	Drive Profile 32-bit Reference 1 Word							
7	Drive Profile 32-bit Reference 1 Word (High Byte)							
8	DATA OUT 1 Value (Low Byte)							
9	DATA OUT 1 Value (High Byte)							
10	DATA OUT 2 Value (Low Byte)							
11	DATA OUT 2 Value (High Byte)							
12	DATA OUT 3 Value (Low Byte)							
13	DATA OUT 3 Value (High Byte)							
14	DATA OUT 4 Value (Low Byte)							
15	DATA OUT 4 Value (High Byte)							
16	DATA OUT 5 Value (Low Byte)							
17	DATA OUT 5 Value (High Byte)							
18	DATA OUT 6 Value (Low Byte)							
19	DATA OUT 6 Value (High Byte)							
20	DATA OUT 7 Value (Low Byte)							
21	DATA OUT 7 Value (High Byte)							
22	DATA OUT 8 Value (Low Byte)							

Instance 121 (Transparent 32 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
23	DATA OUT 8 Value (High Byte)							
24	DATA OUT 9 Value (Low Byte)							
25	DATA OUT 9 Value (High Byte)							
26	DATA OUT 10 Value (Low Byte)							
27	DATA OUT 10 Value (High Byte)							

The format of the input assembly is:

Instance 171 (Transparent 32 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Drive Profile 32-bit Status Word (Low Byte)							
1	Drive Profile 32-bit Status Word							
2	Drive Profile 32-bit Status Word							
3	Drive Profile 32-bit Status Word (High Byte)							
4	Drive Profile 32-bit Actual 1 Word (Low Byte)							
5	Drive Profile 32-bit Actual 1 Word (High Byte)							
6	Drive Profile 32-bit Actual 1 Word							
7	Drive Profile 32-bit Actual 1 Word (High Byte)							
8	DATA IN 1 Value (Low Byte)							
9	DATA IN 1 Value (High Byte)							
10	DATA IN 2 Value (Low Byte)							
11	DATA IN 2 Value (High Byte)							
12	DATA IN 3 Value (Low Byte)							
13	DATA IN 3 Value (High Byte)							
14	DATA IN 4 Value (Low Byte)							
15	DATA IN 4 Value (High Byte)							
16	DATA IN 5 Value (Low Byte)							
17	DATA IN 5 Value (High Byte)							
18	DATA IN 6 Value (Low Byte)							
19	DATA IN 6 Value (High Byte)							
20	DATA IN 7 Value (Low Byte)							
21	DATA IN 7 Value (High Byte)							
22	DATA IN 8 Value (Low Byte)							
23	DATA IN 8 Value (High Byte)							

Instance 171 (Transparent 32 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
24	DATA IN 9 Value (Low Byte)							
25	DATA IN 9 Value (High Byte)							
26	DATA IN 10 Value (Low Byte)							
27	DATA IN 10 Value (High Byte)							

Transparent 32 with two assembly

The Transparent 32 with two assembly, defined by ABB, provides unaltered 32-bit access to the configured drive profile.

The format of the output assembly is:

Instance 22 (Transparent 32 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Drive Profile 32-bit Control Word (Low Byte)							
1	Drive Profile 32-bit Control Word							
2	Drive Profile 32-bit Control Word							
3	Drive Profile 32-bit Control Word (High Byte)							
4	Drive Profile 32-bit Reference 1 Word (Low Byte)							
5	Drive Profile 32-bit Reference 1 Word							
6	Drive Profile 32-bit Reference 1 Word							
7	Drive Profile 32-bit Reference 1 Word (High Byte)							
8	Drive Profile 32-bit Reference 2 Word (Low Byte)							
9	Drive Profile 32-bit Reference 2 Word							
10	Drive Profile 32-bit Reference 2 Word							
11	Drive Profile 32-bit Reference 2 Word (High Byte)							

The format of the input assembly is:

Instance 72 (Transparent 32 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Drive Profile 32-bit Status Word (Low Byte)							
1	Drive Profile 32-bit Status Word							
2	Drive Profile 32-bit Status Word							
3	Drive Profile 32-bit Status Word (High Byte)							
4	Drive Profile 32-bit Actual 1 Word (Low Byte)							
5	Drive Profile 32-bit Actual 1 Word							

Instance 72 (Transparent 32 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
6	Drive Profile 32-bit Actual 1 Word							
7	Drive Profile 32-bit Actual 1 Word (High Byte)							
8	Drive Profile 32-bit Actual 2 Word (Low Byte)							
9	Drive Profile 32-bit Actual 2 Word							
10	Drive Profile 32-bit Actual 2 Word							
11	Drive Profile 32-bit Actual 2 Word (High Byte)							

Transparent 32 with two assembly plus drive parameters

The Transparent 32 with two assembly plus drive parameters, defined by ABB, adds configurable drive parameters to the Transparent 32 with two assembly.

The format of the output assembly is:

Instance 122 (Transparent 32 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Drive Profile 32-bit Control Word (Low Byte)							
1	Drive Profile 32-bit Control Word							
2	Drive Profile 32-bit Control Word							
3	Drive Profile 32-bit Control Word (High Byte)							
4	Drive Profile 32-bit Reference 1 Word (Low Byte)							
5	Drive Profile 32-bit Reference 1 Word							
6	Drive Profile 32-bit Reference 1 Word							
7	Drive Profile 32-bit Reference 1 Word (High Byte)							
8	Drive Profile 32-bit Reference 2 Word (Low Byte)							
9	Drive Profile 32-bit Reference 2 Word							
10	Drive Profile 32-bit Reference 2 Word							
11	Drive Profile 32-bit Reference 2 Word (High Byte)							
12	DATA OUT 1 Value (Low Byte)							
13	DATA OUT 1 Value (High Byte)							
14	DATA OUT 2 Value (Low Byte)							
15	DATA OUT 2 Value (High Byte)							
16	DATA OUT 3 Value (Low Byte)							
17	DATA OUT 3 Value (High Byte)							
18	DATA OUT 4 Value (Low Byte)							
19	DATA OUT 4 Value (High Byte)							

Instance 122 (Transparent 32 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
20	DATA OUT 5 Value (Low Byte)							
21	DATA OUT 5 Value (High Byte)							
22	DATA OUT 6 Value (Low Byte)							
23	DATA OUT 6 Value (High Byte)							
24	DATA OUT 7 Value (Low Byte)							
25	DATA OUT 7 Value (High Byte)							
26	DATA OUT 8 Value (Low Byte)							
27	DATA OUT 8 Value (High Byte)							
28	DATA OUT 9 Value (Low Byte)							
29	DATA OUT 9 Value (High Byte)							
30	DATA OUT 10 Value (Low Byte)							
31	DATA OUT 10 Value (High Byte)							

The format of the input assembly is:

Instance 172 (Transparent 32 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Drive Profile 32-bit Status Word (Low Byte)							
1	Drive Profile 32-bit Status Word							
2	Drive Profile 32-bit Status Word							
3	Drive Profile 32-bit Status Word (High Byte)							
4	Drive Profile 32-bit Actual 1 Word (Low Byte)							
5	Drive Profile 32-bit Actual 1 Word							
6	Drive Profile 32-bit Actual 1 Word							
7	Drive Profile 32-bit Actual 1 Word (High Byte)							
8	Drive Profile 32-bit Actual 2 Word (Low Byte)							
9	Drive Profile 32-bit Actual 2 Word							
10	Drive Profile 32-bit Actual 2 Word							
11	Drive Profile 32-bit Actual 2 Word (High Byte)							
12	DATA IN 1 Value (Low Byte)							
13	DATA IN 1 Value (High Byte)							
14	DATA IN 2 Value (Low Byte)							
15	DATA IN 2 Value (High Byte)							
16	DATA IN 3 Value (Low Byte)							

Instance 172 (Transparent 32 profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
17	DATA IN 3 Value (High Byte)							
18	DATA IN 4 Value (Low Byte)							
19	DATA IN 4 Value (High Byte)							
20	DATA IN 5 Value (Low Byte)							
21	DATA IN 5 Value (High Byte)							
22	DATA IN 6 Value (Low Byte)							
23	DATA IN 6 Value (High Byte)							
24	DATA IN 7 Value (Low Byte)							
25	DATA IN 7 Value (High Byte)							
26	DATA IN 8 Value (Low Byte)							
27	DATA IN 8 Value (High Byte)							
28	DATA IN 9 Value (Low Byte)							
29	DATA IN 9 Value (High Byte)							
30	DATA IN 10 Value (Low Byte)							
31	DATA IN 10 Value (High Byte)							

Class objects

The following table lists the data types used in the class object descriptions of this manual.

Legend	**Data type**
UINT8	Unsigned Integer 8 bit
UINT16	Unsigned Integer 16 bit
SINT16	Signed Integer 16 bit
UINT32	Unsigned Integer 32 bit
BOOL	Boolean value

Note: The adapter module is designed to provide EtherNet/IP communications for a variety of drives with different capabilities. Default, minimum and maximum values for attributes necessarily vary based upon the capabilities of the drive to which the module is attached and are not documented herein. Default, minimum and maximum values for attributes may be found in the:

- drive manuals
- Electronic Data Sheet Files (EDS) for the drive.

Note that the units of the attributes may differ from those of the parameters documented elsewhere, and those differences must be considered when interfacing to the drive via the module.

The table below shows the service names of the class objects.

Service	Name
GET	0x0E Get_Attribute_Single
SET	0x10 Set_Attribute_Single
SET ALL	0x02 Set_Attribute_All
GET ALL	0x01 Get_Attribute_All

Identity object, class 01h

This object provides identification of and general information about the device.

Class attributes (Instance #0)

#	Attribute name	Services	Description	Data type
1	Revision	Get	Revision of the identity object	Array of UINT8

Instance attributes (Instance #1)

#	Attribute name	Services	Description	Data type
1	Vendor ID	Get	Identification of the device vendor	UINT16
2	Device Type	Get	Identification of the general product type	UINT16
3	Product Code	Get	Assigned vendor code to describe the device	UINT16
4	Revision	Get	Revision of the item the identity object represents	Array[UINT8 UINT8]
5	Status	Get	Summary status of the device	UINT16
6	ODVA Serial Number	Get	Serial number of the EtherNet/IP module	UINT32
7	Product Name	Get	Product identification. Max 32 characters.	Short String

Reset service (Service code 05h)

Value (reset type)	Type of reset
0	Reset the adapter

Value (reset type)	Type of reset
1	Reset the adapter (* and factory default configuration)
2	Reset the adapter (* and set the out-of-box configuration with the exception of communication link parameters)

* not implemented

Attribute explanations

Vendor ID

Vendor IDs are managed by the Open DeviceNet Vendor Association, Inc. (ODVA). The ABB Vendor ID is 46.

Device Type

The list of device types is managed by ODVA. It is used to identify the device profile that a particular product is using.

Drive Type	Profile	Device Type	Value
AC	ODVA AC/DC Drive	ODVA AC Drive	02h
	ABB Drives Profile	ABB AC Drive	64h
	Transparent 16	ABB AC Drive	64h
	Transparent 32	ABB AC Drive	64h
DC	ODVA AC/DC Drive	ODVA DC Drive	13h
	ABB Drives Profile	ABB DC Drive	65h
	Transparent 16	ABB DC Drive	65h
	Transparent 32	ABB DC Drive	65h

Product Code

Every ABB drive type or application of the drive has a dedicated product code. The product code is 100 + the value of parameter 29 FBA A/B drive type code (page 33).

Revision

Revision attribute, which consists of Major and Minor Revisions, identifies the revision of the item the identity object represents.

Status

This attribute represents the current status of the entire device. Its value changes as the state of the device changes. The Status attribute is a WORD, with the following bit definitions:

Bit(s)	Type/ Name	Definition
0	Owned	TRUE indicates the device (or an object within the device) has an owner. Within the Master/Slave paradigm the setting of this bit means that the Predefined Master/Slave Connection Set has been allocated to a master. Outside the Master/Slave paradigm the meaning of this bit is to be defined.
1		Reserved, set to 0

Bit(s)	Type/ Name	Definition
2	Configured	TRUE indicates that the application of the device has been configured to do something that differs from the “out–of–box” default. This does not include configuration of the communications.
3		Reserved, set to 0
4, 5, 6, 7		Vendor-specific
8	Minor Recoverable Fault	TRUE indicates the device detected a recoverable problem. The problem does not cause the device fault state.
9	Minor Unrecoverable Fault	TRUE indicates the device detected a unrecoverable problem. The problem does not cause the device fault state.
10	Major Recoverable Fault	TRUE indicates the device detected a problem which caused the device to transfer into the “Major Recoverable Fault” state.
11	Major Unrecoverable Fault	TRUE indicates the device detected a problem which caused the device to transfer into the “Major Unrecoverable Fault” state.
12, 13, 14, 15		Reserved, set to 0

ODVA Serial Number

This attribute is a number used in conjunction with the Vendor ID to form a unique identifier for each device on EtherNet/IP. The value of this attribute is 02000000h plus the SERNO value from the device label.

Product Name

This text string should represent a short description of the product/product family represented by the product code in attribute 3.

Motor data object, class 28h

The Motor data object can only be used if the ODVA AC/DC drive profile is in use.

The object serves as a database for motor parameters. Different motor types require different data to describe the motor. For example, AC induction motors do not need field current data like a DC motor to describe the motor.

Motor class	Motor types in class
AC motors	3 - PM synchronous 6 - Wound rotor induction 7 - Squirrel cage induction motor
DC motors	1 - PM DC motor 2 - FC DC motor

Class attributes (Instance #0)

#	Attribute name	Services	Description	Data type
1	Revision	Get	Revision of the CIP Object Class Definition upon which the implementation is based	Array of UINT8

Instance attributes (Instance #1)

#	Attribute name	Services	Description	Motor type	Data type
3	Motor type	Get	See the table above.	AC	UINT8
6	Rated Current	Get, Set	Rated Stator Current from motor name plate Units: [100mA]	AC/DC	UINT16
7	Rated Voltage	Get, Set	Rated Base Voltage from motor name plate Units: [V]	AC/DC	UINT16
8	Rated Power	Get, Set	Rated Power at Rated Frequency Units: [W]	AC/DC	UINT32
9	Rated frequency	Get, Set	Rated Electrical Frequency Units: [Hz]	AC	UINT16
12	Pole Count	Get	Number of poles in the motor	AC	UINT16
15	Base Speed	Get, Set	Nominal speed at rated frequency from nameplate Units [RPM]	AC/DC	UINT16

Control supervisor object, class 29h

The Control supervisor object can only be used if the ODVA AC/DC drive profile is in use.

The object models all the management functions for devices within the 'Hierarchy of Motor Control Devices'. The behavior of motor control devices is described by:

- AC/DC-drive object, class 2Ah (page 100) and
- Run/Stop event matrix under Run Forward & Run Reverse (Control supervisor object) (page 57).

See also section State (Control supervisor object) (page 59).

Note: If assembly instances are used, they override this object, for example, upon drive power-up.

Class attributes (Instance #0)

#	Attribute name	Services	Description	Data type
1	Revision	Get	Revision of the CIP Object Class Definition upon which the implementation is based	Array of UINT8

Instance attributes (Instance #1)

#	Attribute name	Services	Description	Data type
3	Run 1 (RunFwd)	Get, Set	0 = Stop, 1 = Run See Run Forward & Run Reverse (Control supervisor object) (page 57).	BOOL
4	Run 2 (RunRev)	Get, Set	0 = Stop, 1 = Run See Run Forward & Run Reverse (Control supervisor object) (page 57).	BOOL
5	Net Control	Get, Set	0 = Local Control, 1 = Network Control	BOOL
6	State	Get	State of Object. See section State (Control supervisor object) (page 59).	UINT8
7	Running 1 (Fwd)	Get	0 = Stopped, 1 = Running	BOOL
8	Running 2 (Rev)	Get	0 = Stopped, 1 = Running	BOOL
9	Ready	Get	1 = Ready, Enabled or Stopping; 0 = Other state	BOOL
10	Faulted	Get	0 = Not faulted, 1 = Fault occurred	BOOL
11	Warning	Get	0 = No Warnings present, 1 = Warning	BOOL
12	FaultRst	Get, Set	0 → 1 Fault Reset	BOOL
13	Fault Code	Get	The fault that caused the last transition to the Faulted state. DRIVECOMM codes are reported. See the drive manual for further information on DRIVECOMM codes.	UINT16

#	Attribute name	Services	Description	Data type
14	Warning Code	Get	Code word indicating the warning present. If multiple warnings are present, the lowest code value is displayed. DRIVECOMM codes are reported. See the drive manual for further information on DRIVECOMM codes.	UINT16
15	CtlFromNet	Get	0 = NetControl disabled 1 = NetControl enabled	BOOL
16	DNFaultMode	Get, Set	2 = Vendor specified	UINT8
17	ForceFault	Get, Set	0 → 1 forces the drive to fault	BOOL

AC/DC-drive object, class 2Ah

The AC/DC-drive object can only be used if the ODVA AC/DC drive profile is in use.

The object models the functions specific to an AC or DC Drive.

Class attributes (Instance #0)

#	Attribute name	Services	Description	Data type
1	Revision	Get	Revision of the CIP Object Class Definition upon which the implementation is based	Array of UINT8

Instance attributes (Instance #1)

#	Attribute name	Services	Description	Data type
3	At Reference	Get	Frequency arrival	BOOL

#	Attribute name	Services	Description	Data type
4	NetRef	Get, Set	Requests torque or speed reference to be local or from the network. 0 = Set Reference not DN Control 1 = Set Reference at DN Control Note that the actual status of torque or speed reference is reflected in attribute 29, RefFromNet.	BOOL
6	Drive mode	Get, Set	0 = Vendor-specific	UINT8
7	Speed Actual	Get	Units = See parameter 23 Speed scale (page 31).	SINT16
8	SpeedRef	Get, Set	Units = See parameter 23 Speed scale (page 31).	SINT16
11	Torque Actual	Get	Units = See parameter 24 Torque scale (page 32).	SINT16
12	TorqueRef	Get, Set	Units = See parameter 24 Torque scale (page 32).	SINT16
18	AccelTime	Get, Set	Units = milliseconds	UINT16
19	DecelTime	Get, Set	Units = milliseconds	UINT16
22	Speed Scale	Get, Set	Speed scaling factor. See parameter 23 Speed scale (page 31).	UINT8
24	Torque Scale	Get, Set	Torque scaling factor. See parameter 24 Torque scale (page 32).	UINT8
29	Ref From Net	Get	Reflecting attribute 4	BOOL

Drive parameter object, class 90h

With the FEIP-21 adapter module, drive parameters can also be accessed via Explicit Messaging. Explicit Messaging makes use of objects consisting of three parts: class, instance and attribute.

Note: When you use the drive parameter object to update the fieldbus configuration groups, changes to the fieldbus configuration will only take effect when the module is powered up the next time or when a 'Fieldbus Adapter parameter refresh' is given.

Class is always 144 (90h). Instance and attribute correspond to the drive parameter group and index in the following way:

- Instance = Parameter group (0…99) (ACx880/580: 0...255)
- Attribute = Parameter index (01…99) (ACx880/580: 0...255)

For example, Parameter 99.01 is accessed as follows:

- Class = 144 = 90h
- Instance = 99 = 63h
- Attribute = 1 = 01h

Fieldbus configuration object, class 91h

The fieldbus configuration object allows you to configure the fieldbus configuration groups without the need to know the drive-specific groups associated with the configuration groups.

Note: When you use the fieldbus configuration object to update the fieldbus configuration groups, changes to the fieldbus configuration will only take effect when a reset service is requested of the Identity Object, the module is powered up the next time or when a 'Fieldbus Adapter parameter refresh' is given.

Class attributes

#	Attribute name	Services	Description	Data type
1	Revision	Get	Revision of the Configuration Object	Array of UINT8

Instance #1: FEIP-21 configuration parameters group A (group 1)

The actual parameter group number depends on the drive type. Group A (group 1) corresponds to:

- parameter group 51 in ACS380, ACS480, ACH580, ACQ580 and ACS580
- parameter group is typically 51/54 (group 151/154 in some variants) in ACS880 if the adapter is installed as fieldbus adapter A/B.

#	Attribute name	Services	Description	Data type
1	Configuration Group A (Group 1) - Parameter 1	Get, Set	See FBA type (page 26) .	UINT16
2	Configuration Group A (Group 1) - Parameter 2	Get, Set	See Protocol/Profile (page 26).	UINT16
3	Configuration Group A (Group 1) - Parameter 3	Get, Set	See Comm-rate (page 26).	UINT16
4	Configuration Group A (Group 1) - Parameter 4	Get, Set	See IP configuration (page 26).	UINT16

#	Attribute name	Services	Description	Data type
5	Configuration Group A (Group 1) - Parameter 5	Get, Set	See IP address 1 (page 27).	UINT16
6	Configuration Group A (Group 1) - Parameter 6	Get, Set	See IP address 1 (page 27).	UINT16
7	Configuration Group A (Group 1) - Parameter 7	Get, Set	See IP address 1 (page 27).	UINT16
8	Configuration Group A (Group 1) - Parameter 8	Get, Set	See IP address 1 (page 27).	UINT16
9	Configuration Group A (Group 1) - Parameter 9	Get, Set	See Subnet CIDR (page 27).	UINT16
10	Configuration Group A (Group 1) - Parameter 10	Get, Set	See GW address 1 (page 27).	UINT16
11	Configuration Group A (Group 1) - Parameter 11	Get, Set	See GW address 1 (page 27).	UINT16
12	Configuration Group A (Group 1) - Parameter 12	Get, Set	See GW address 1 (page 27).	UINT16
13	Configuration Group A (Group 1) - Parameter 13	Get, Set	See GW address 1 (page 27).	UINT16
14	Configuration Group A (Group 1) - Parameter 14	Get, Set	See Commrate port 2 (page 28).	UINT16
15	Configuration Group A (Group 1) - Parameter 15	Get, Set	See Service configuration (page 28).	UINT16
16	Configuration Group A (Group 1) - Parameter 16	Get, Set	See Reserved (page 29).	UINT16
17	Configuration Group A (Group 1) - Parameter 17	Get, Set	See Reserved (page 29).	UINT16
18	Configuration Group A (Group 1) - Parameter 18	Get, Set	See Reserved (page 29).	UINT16
19	Configuration Group A (Group 1) - Parameter 19	Get, Set	See T16 scale (page 30).	UINT16
20	Configuration Group A (Group 1) - Parameter 20	Get, Set	See Control timeout (page 30).	UINT16

#	Attribute name	Services	Description	Data type
21	Configuration Group A (Group 1) - Parameter 21	Get, Set	See Idle action (page 31).	UINT16
22	Configuration Group A (Group 1) - Parameter 22	Get, Set	See Stop function (page 31).	UINT16
23	Configuration Group A (Group 1) - Parameter 23	Get, Set	See Speed scale (page 31).	UINT16
24	Configuration Group A (Group 1) - Parameter 24	Get, Set	See Torque scale (page 32).	UINT16
25	Configuration Group A (Group 1) - Parameter 25	Get, Set	See Reserved (page 32).	UINT16
26	Configuration Group A (Group 1) - Parameter 26	Get, Set	See Reserved (page 32).	UINT16
27	Configuration Group A (Group 1) - Parameter 27	Get, Set	See FBA A/B par refresh (page 32).	UINT16
28	Configuration Group A (Group 1) - Parameter 28	Get	See FBA A/B par table ver (page 33).	UINT16
29	Configuration Group A (Group 1) - Parameter 29	Get	See FBA A/B drive type code (page 33).	UINT16
30	Configuration Group A (Group 1) - Parameter 30	Get	See FBA A/B mapping file ver (page 33).	UINT16
31	Configuration Group A (Group 1) - Parameter 31	Get	See D2FBA A/B comm status (page 33).	UINT16
32	Configuration Group A (Group 1) - Parameter 32	Get	See FBA A/B comm SW ver (page 33).	UINT16
33	Configuration Group A (Group 1) - Parameter 33	Get	See FBA A/B appl SW ver (page 34).	UINT16

Instance #2: FEIP-21 configuration parameters group B (group 2)

The actual parameter group number depends on the drive type. Group B (group 2) corresponds to:

- parameter group 53 in ACS380, ACS480, ACH580, ACQ580 and ACS580
- parameter group is typically 53/56 (group 153/156 in some variants) in ACS880 if the adapter is installed as fieldbus adapter A/B.

#	Attribute name	Services	Description	Data type
1	Configuration Group B (Group 2) - Parameter 1	Get, Set	See FBA A/B data out 1 (client to drive) (page 34).	UINT16 / UINT32 ACx880 / 580
2	Configuration Group B (Group 2) - Parameter 2	Get, Set	See FBA A/B data out 1 (client to drive) (page 34).	UINT16 / UINT32 ACx880 / 580
3	Configuration Group B (Group 2) - Parameter 3	Get, Set	See FBA A/B data out 1 (client to drive) (page 34).	UINT16 / UINT32 ACx880 / 580
4	Configuration Group B (Group 2) - Parameter 4	Get, Set	See FBA A/B data out 1 (client to drive) (page 34).	UINT16 / UINT32 ACx880 / 580
5	Configuration Group B (Group 2) - Parameter 5	Get, Set	See FBA A/B data out 1 (client to drive) (page 34).	UINT16 / UINT32 ACx880 / 580
6	Configuration Group B (Group 2) - Parameter 6	Get, Set	See FBA A/B data out 1 (client to drive) (page 34).	UINT16 / UINT32 ACx880 / 580
7	Configuration Group B (Group 2) - Parameter 7	Get, Set	See FBA A/B data out 1 (client to drive) (page 34).	UINT16 / UINT32 ACx880 / 580
8	Configuration Group B (Group 2) - Parameter 8	Get, Set	See FBA A/B data out 1 (client to drive) (page 34).	UINT16 / UINT32 ACx880 / 580
9	Configuration Group B (Group 2) - Parameter 9	Get, Set	See FBA A/B data out 1 (client to drive) (page 34).	UINT16 / UINT32 ACx880 / 580
10	Configuration Group B (Group 2) - Parameter 10	Get, Set	See FBA A/B data out 1 (client to drive) (page 34).	UINT16 / UINT32 ACx880 / 580

Instance #3: FEIP-21 configuration parameters group C (group 3)

The actual parameter group number depends on the drive type. Group C (group 3) corresponds to:

- parameter group 52 in ACS380, ACS480, ACH580, ACQ580 and ACS580
- parameter group is typically 52/55 (group 152/155 in some variants) in ACS880 if the adapter is installed as fieldbus adapter A/B.

#	Attribute name	Services	Description	Data type
1	Configuration Group C (Group 3) - Parameter 1	Get, Set	See FBA A/B data in1 (drive to client) (page 35).	UINT16 / UINT32 ACx880 / 580
2	Configuration Group C (Group 3) - Parameter 2	Get, Set	See FBA A/B data in1 (drive to client) (page 35).	UINT16 / UINT32 ACx880 / 580
3	Configuration Group C (Group 3) - Parameter 3	Get, Set	See FBA A/B data in1 (drive to client) (page 35).	UINT16 / UINT32 ACx880 / 580

#	Attribute name	Services	Description	Data type
4	Configuration Group C (Group 3) - Parameter 4	Get, Set	See FBA A/B data in1 (drive to client) (page 35).	UINT16 / UINT32 ACx880 / 580
5	Configuration Group C (Group 3) - Parameter 5	Get, Set	See FBA A/B data in1 (drive to client) (page 35).	UINT16 / UINT32 ACx880 / 580
6	Configuration Group C (Group 3) - Parameter 6	Get, Set	See FBA A/B data in1 (drive to client) (page 35).	UINT16 / UINT32 ACx880 / 580
7	Configuration Group C (Group 3) - Parameter 7	Get, Set	See FBA A/B data in1 (drive to client) (page 35).	UINT16 / UINT32 ACx880 / 580
8	Configuration Group C (Group 3) - Parameter 8	Get, Set	See FBA A/B data in1 (drive to client) (page 35).	UINT16 / UINT32 ACx880 / 580
9	Configuration Group C (Group 3) - Parameter 9	Get, Set	See FBA A/B data in1 (drive to client) (page 35).	UINT16 / UINT32 ACx880 / 580
10	Configuration Group C (Group 3) - Parameter 10	Get, Set	See FBA A/B data in1 (drive to client) (page 35).	UINT16 / UINT32 ACx880 / 580

Instance #10: SNTP configuration

#	Attribute name	Services	Description	Data type
1	UTC time offset	Get, Set	SNTP time offset to UTC time. Value in minutes [-1440, 1440] to offset the UTC time received from SNTP.	SINT16

TCP/IP interface object, class F5h

This object provides the mechanism to configure the TCP/IP network interface of the device.

Class attributes (Instance #0)

#	Attribute name	Services	Description	Data type
1	Revision	Get	Revision of the TCP/IP Interface Object Class Definition upon which the implementation is based	Array of UINT8

Instance attributes (Instance #1)

#	Attribute name	Services	Description	Data type
1	Interface status	Get	See *Interface Status attribute (#1) bits* (page 107).	DWORD
2	Configuration Capability	Get	See *Configuration Capability attribute (#2) bits* (page 108).	DWORD
3	Configuration Control	Get	See *Configuration Control attribute (#3) bits* (page 108).	DWORD
4	Physical Link Object	Get	Path to physical link object	STRUCT of:
	Path Size		Path size	UINT
	Path		Logical segments identifying the physical link object	Padded EPATH
5	Interface Configuration	Get		STRUCT of:
	IP Address		IP Address	UDINT
	Network Mask		Network Mask	UDINT
	Gateway Address		Gateway Address	UDINT
	Unused			UDINT
	Unused			UDINT
	Default Domain Name		Default Domain Name for unqualified host names.	STRING
6	Host Name	Get / Set	Host name	STRING
8	TTL Value	Get / Set	TTL value for EtherNet/IP multi cast packets	USINT 1 ... 255
13	NV Encapsulation Inactivity	Get / Set	Encapsulation inactivity timeout	UINT 0 = disabled 1 ... 3600 timeout in seconds

Attribute explanations

Interface Status attribute (#1) bits

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>0 ... 3</td><td>Interface configuration status</td><td>Indicates the status of the Interface Configuration attribute.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>The Interface Configuration attribute has not been configured.</td></tr>
<tr><td>1</td><td>The Interface Configuration attribute contains valid configuration obtained from BOOTP, DHCP or non-volatile storage.</td></tr>
<tr><td>2</td><td>The IP address member of the Interface Configuration attribute contains valid configuration obtained from hardware settings (e.g., push-wheel, thumbwheel).</td></tr>
<tr><td>3 ... 15</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>4</td><td>Mcast pending</td><td>Indicates a pending configuration change in the TTL Value and/or Mcast Config attributes. This bit is set when either the TTL Value or Mcast Config attribute is set, and cleared the next time the device starts.</td></tr>
<tr><td>5 ... 31</td><td></td><td>Reserved, set to 0</td></tr>
</table>

Configuration Capability attribute (#2) bits

Bit	Name	Description
0	BOOTP client	1 (True) = The device is capable of obtaining its network configuration via BOOTP.
1	DNS client	1 (True) = The device is capable of resolving host names by querying a DNS server.
2	DHCP client	1 (True) = The device is capable of obtaining its network configuration via DHCP.
3	DCHP-DNS update	1 (True) = The device is capable of sending its host name in the DHCP request as documented in Internet draft <draft-ietf-dhc-dhcp-dnc-12.txt>
4	Configuration settable	1 (True) = The Interface Configuration attribute is settable. Some devices, e.g., a PC or workstation, may not allow interface configuration to be set via the TCP/IP interface object.
5	Hardware configurable	1 (True) = The IP address member of the Interface Configuration attribute can be obtained from hardware settings (e.g., push-wheel, thumb-wheel).
		0 (False) = The Status instance attribute (1) Interface configuration status field value shall never be 2. (The Interface configuration attribute contains valid configuration obtained from hardware settings.)
6 ... 31		Reserved, set to 0

Configuration Control attribute (#3) bits

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>0 ... 3</td><td>Start-up configuration</td><td>Determines how the device obtains its initial configuration and start-up.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>The device uses the interface configuration values previously stored (eg, in non-volatile memory or via hardware switches).</td></tr>
<tr><td>1</td><td>The device obtains its interface configuration values via BOOTP.</td></tr>
<tr><td>2</td><td>The device obtains its interface configuration values via DHCP upon startup.</td></tr>
<tr><td>3 ... 15</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>4</td><td>DNS enable</td><td>1 (True) = The device resolves host names by querying a DNS server.</td></tr>
<tr><td>5 ... 31</td><td></td><td>Reserved, set to 0.</td></tr>
</table>

Ethernet link object, class F6h

This object maintains link-specific counters and status information for the Ethernet communication interface.

Class attributes (Instance #0)

#	Attribute name	Services	Description	Data type
1	Revision	Get	Revision of the Ethernet Link Object Class Definition upon which the implementation is based	Array of UINT8

Instance attributes (Instance #1)

<table>
<tr><th>#</th><th>Attribute name</th><th>Services</th><th>Description</th><th>Data type</th></tr>
<tr><td>1</td><td>Interface Speed</td><td>Get, Set</td><td>10 or 100 Mbps</td><td>UDINT</td></tr>
<tr><td>2</td><td>Interface Flags</td><td>Get, Set</td><td>Interface status flags:
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Link status</td></tr>
<tr><td>1</td><td>Half/Full duplex</td></tr>
<tr><td>2 ... 4</td><td>Negotiation status</td></tr>
<tr><td>5</td><td>Manual setting requires reset</td></tr>
<tr><td>6</td><td>Local hardware fault</td></tr>
<tr><td>7 ... 31</td><td>Reserved</td></tr>
</table></td><td>DWORD</td></tr>
</table>

#	Attribute name	Services	Description	Data type
3	Physical Address	Get	Ethernet MAC address of the module	ARRAY 6XUSINT

Connection object, class 05h

Do not modify this object. This object is only used while establishing the connection between the adapter module and the PLC.

The connection class allocates and manages the internal resources associated with both I/O and explicit messaging connections. The specific instance generated by the connection class is referred to as connection instance or connection object.

Connection object states

State	Description	State	Description
00	Non-Existent	03	Established
01	Configuring	04	Timed Out
02	Waiting for Connection ID	05	Deferred Delete

Class attributes

#	Attribute name	Services	Description	Data type
1	Revision	Get	Revision of the connection object	Array of UINT8

Instance attributes

Instance number	Description
1	Explicit messaging connection
2	Polled I/O connection
4	Change-of-State / Cyclic I/O connection

#	Attribute name	Services	Description	Data type
1	State	Get	State of the object. See the *Connection object states* (page 110).	UINT8
2	Instance Type	Get	Indicates either I/O (1) or messaging connection (0).	UINT8
3	Transport Class Trigger	Get	Defines the behavior of the connection.	UINT8
4	Produced Cnxn Id	Get	Placed in CAN Identifier Field when the connection transmits.	UINT16
5	Consumed Cnxn Id	Get	CAN Identifier Field value that denotes the message to be received	UINT16

#	Attribute name	Services	Description	Data type
6	Comm Characteristics	Get	Defines the Message Group(s) across which productions and consumptions are associated in this connection.	UINT8
7	Produced Connection Size	Get	Maximum number of bytes transmitted across this connection	UINT16
8	Consumed Connection size	Get	Maximum number of bytes received across this connection	UINT16
9	Expected Packet Rate	Get, Set	Defines the timing associated with this connection in milliseconds. A value of 0 deactivates the associated timers.	UINT16
12	Watchdog Timeout Action	Get, Set	Defines how to handle Inactivity/Watchdog timeouts.	UINT8
13	Produced Connection Path Length	Get	Number of bytes in the produced_connection_path length attribute	UINT16
14	Produced Connection Path	Get	Application object producing data on this connection	Array of UINT8
15	Consumed Connection Path Length	Get	Number of bytes in the consumed_connection_path length attribute	UINT16
16	Consumed Connection Path	Get	Specifies the application object(s) that are to receive the data consumed by this connection object.	Array of UINT8
17	Production Inhibit Time	Get	Defines the minimum time between new data production in milliseconds.	UINT16

Acknowledge handler object, class 2Bh

The acknowledge handler object is used to manage the reception of message acknowledgements. This object communicates with a message producing application object within the device. The acknowledge handler object notifies the producing application of acknowledge reception, acknowledge timeouts and production retry limit.

Class attributes (Instance #0)

#	Attribute name	Services	Description	Data type
1	Revision	Get	Revision of the CIP Object Class Definition upon which the implementation is based	Array of UINT8

Instance attributes (Instance #1)

#	Attribute name	Services	Description	Data type
1	Acknowledge Timer	Get, Set	Time in milliseconds to wait for acknowledge before resending	UINT16
2	Retry Limit	Get, Set	Number of Acknowledge Timeouts to wait before informing the producing application of a Retry-Limit_Reached event	UINT8
3	COS Producing Connection Instance	Get	Connection Instance Id which contains the path of the producing I/O application object which will be notified of Acknowledge Handler events	UINT16

9

EtherNet/IP – Diagnostics

Contents of this chapter

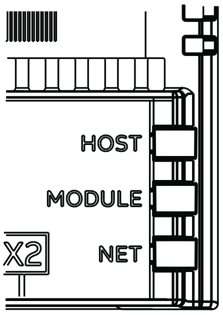
This chapter explains how to trace faults with the status LEDs on the adapter module when the module is used for EtherNet/IP communication.

Fault and warning messages

For the fault and warning messages concerning the adapter module, see the drive firmware manual.

LEDs

The adapter module is equipped with three bicolor diagnostic LEDs. The LEDs are described below.



Name	**Color**	**Function**
HOST	Flashing green	Establishing communication to host
	Green	Connection to host OK
	Flashing red	Communication to host lost temporarily
	Flashing orange, alternating with the MODULE flashing orange	Internal file system error. The error may be cleared by cycling drive power. If the error persists, contact your local ABB representative.
MODULE	Off	There is no power applied to the device.
	Green	Device is operating in a normal condition.
	Flashing green	Device needs commissioning due to configuration missing, incomplete or incorrect. The device may be in the Standby state. This may be caused by the adapter waiting for a response from a DHCP server or Duplicate Address Detection to complete.
	Flashing red	Ethernet interface is disabled. Duplicate Address Detection may have detected a duplicate address. Check the IP configuration and either initiate a Fieldbus Adapter parameter refresh or cycle power to the drive.
	Flashing red-green	Device is in Self Test.
	Flashing orange, alternating with the HOST flashing orange	Internal file system error. The error may be cleared by cycling drive power. If the error persists, contact your local ABB representative.
NETWORK / NET	Off	Device is not on-line. • The device has not completed the Duplicate Address Detection yet. • The device may not be powered; look at the MODULE status LED.
	Flashing green	Device is on-line but has no connections in the established state. • The device has passed Duplicate Address Detection, is on-line, but has no established connections to other nodes.
	Green	Device is on-line and has connections in the established state.
	Flashing red	One or more I/O connections are in the Timed-out state.
	Red	Failed communication device. The device has detected an error that has rendered it incapable of communicating on the network (Duplicate MAC ID or IP address detected).

10

NONE – Start-up

Contents of this chapter

This chapter contains:

- information on configuring the drive for operation with the adapter module
- drive-specific instructions on starting up the drive with the adapter module
- information on configuring the client for communication with the adapter module.

Warnings

WARNING!
Obey the safety instructions given in this manual and the drive documentation.

Drive configuration

The information in this section applies to all drive types compatible with the adapter module, unless otherwise stated.

Connection configuration using NONE protocol

After the adapter module is mechanically and electrically installed according to the instructions in chapters Mechanical installation and Electrical installation, you must prepare the drive for communication with the module.

The detailed procedure of activating the module using the NONE protocol with the drive depends on the drive type. Normally, you must adjust a parameter to activate the communication. See the drive-specific start-up sections Starting up fieldbus communication (page 120).

Once communication between the drive and the adapter module is established, several configuration parameters are copied to the drive. These parameters are shown in the tables below and must be checked first and adjusted where necessary. You can adjust the parameters via a drive control panel or a PC tool.

Note:

- Not all drives will display the descriptive names for the configuration parameters.
- The new parameter settings take effect only when you power up the module the next time or when you activate the fieldbus adapter refresh parameter.
- Use the NONE protocol selection when no fieldbus protocol is required. For example, when only the Ethernet tool network is used or for synchronizing time.

FEIP-21 configuration parameters – group A (group 1)

Note: The actual parameter group number depends on the drive type. Group A (group 1) corresponds to:

- parameter group 51 in ACS380 and ACS580.
- parameter group 51 in ACS880 if the adapter is installed as fieldbus adapter A or group 54 if the adapter is installed as fieldbus adapter B.

No.	Name/Value	Description	Default
01	FBA TYPE	**Read-only.** Shows the fieldbus adapter type as detected by the drive. The value cannot be adjusted by the user. If the value is **0** = None, the communication between the drive and the module has not been established.	**2222** = EtherNet/IP
02	Protocol/Profile	Select NONE to disable the EtherNet/IP protocol. Other services (ABB IP Configuration tool, Ethernet tool network, etc.) are still available, see Service configuration (page 118).	**2222** = EtherNet/IP
	200 = NONE	NONE protocol	
03	Commrate	Sets the bit rate for the Ethernet interface.	**0** = Auto
	0 = Auto	Autonegotiate	
	1 = 100 Mbps FD	100 Mbps, full duplex	
	2 = 100 Mbps HD	100 Mbps, half duplex	
	3 = 10 Mbps FD	10 Mbps, full duplex	
	4 = 10 Mbps HD	10 Mbps, half duplex	
04	IP configuration	Sets the method for configuring the IP address, subnet mask and gateway address for the module.	**1** = Dyn IP DHCP
	0 = Static IP	Configuration is obtained from parameters 05…13.	
	1 = Dyn IP DHCP	Configuration is obtained via DHCP.	

<table>
<tr><th>No.</th><th>Name/Value</th><th>Description</th><th>Default</th></tr>
<tr><td rowspan="5">05
…
08</td><td>IP address 1</td><td>An IP address is assigned to each IP node on a network. An IP address is a 32-bit number that is typically represented in “dotted decimal” notation consisting of four decimal integers, on the range 0…255, separated by periods. Each integer represents the value of one octet (8-bits) in the IP address. Parameters define the four octets of the IP address.</td><td>0</td></tr>
<tr><td>0...255</td><td>IP address</td><td></td></tr>
<tr><td>...</td><td>...</td><td>...</td></tr>
<tr><td>IP address 4</td><td>See parameter 05 IP address 1</td><td>0</td></tr>
<tr><td>0...255</td><td>IP address</td><td></td></tr>
<tr><td rowspan="3">09</td><td>Subnet CIDR</td><td>Subnet masks are used for splitting networks into smaller networks called subnets. A subnet mask is a 32-bit binary number that splits the IP address into a network address and host address.
Subnet masks are typically represented in either dotted decimal notation or the more compact CIDR notation, as shown in the table below.</td><td>0</td></tr>
<tr><td colspan="3">
<table>
<tr><th>CIDR</th><th>Dotted decimal</th><th>CIDR</th><th>Dotted decimal</th></tr>
<tr><td>31</td><td>255.255.255.254</td><td>15</td><td>255.254.0.0</td></tr>
<tr><td>30</td><td>255.255.255.252</td><td>14</td><td>255.252.0.0</td></tr>
<tr><td>29</td><td>255.255.255.248</td><td>13</td><td>255.248.0.0</td></tr>
<tr><td>28</td><td>255.255.255.240</td><td>12</td><td>255.240.0.0</td></tr>
<tr><td>27</td><td>255.255.255.224</td><td>11</td><td>255.224.0.0</td></tr>
<tr><td>26</td><td>255.255.255.192</td><td>10</td><td>255.192.0.0</td></tr>
<tr><td>25</td><td>255.255.255.128</td><td>9</td><td>255.128.0.0</td></tr>
<tr><td>24</td><td>255.255.255.0</td><td>8</td><td>255.0.0.0</td></tr>
<tr><td>23</td><td>255.255.254.0</td><td>7</td><td>254.0.0.0</td></tr>
<tr><td>22</td><td>255.255.252.0</td><td>6</td><td>252.0.0.0</td></tr>
<tr><td>21</td><td>255.255.248.0</td><td>5</td><td>248.0.0.0</td></tr>
<tr><td>20</td><td>255.255.240.0</td><td>4</td><td>240.0.0.0</td></tr>
<tr><td>19</td><td>255.255.224.0</td><td>3</td><td>224.0.0.0</td></tr>
<tr><td>18</td><td>255.255.192.0</td><td>2</td><td>192.0.0.0</td></tr>
<tr><td>17</td><td>255.255.128.0</td><td>1</td><td>128.0.0.0</td></tr>
<tr><td>16</td><td>255.255.0.0</td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>1...31</td><td>Subnet mask in CIDR notation</td><td></td></tr>
</table>

<table>
<tr><th>No.</th><th>Name/Value</th><th>Description</th><th>Default</th></tr>
<tr><td>10
…
13</td><td>GW address 1</td><td>IP gateways connect individual physical IP subnets into a unified IP network. When an IP node needs to communicate with an IP node on another subnet, the IP node sends the data to the IP gateway for forwarding. Parameters define the four octets of the gateway address.</td><td>0</td></tr>
<tr><td></td><td>0…255</td><td>GW address</td><td></td></tr>
<tr><td></td><td>…</td><td>…</td><td>…</td></tr>
<tr><td></td><td>GW address 4</td><td>See parameter 10 GW address 1.</td><td>0</td></tr>
<tr><td></td><td>0.255</td><td>GW address</td><td></td></tr>
<tr><td>14</td><td>Commrate port 2</td><td>Sets the bit rate for the Ethernet port 2.</td><td><b>0</b> = Auto</td></tr>
<tr><td></td><td><b>0</b> = Auto</td><td>Autonegotiate</td><td></td></tr>
<tr><td></td><td><b>1</b> = 100 Mbps FD</td><td>100 Mbps, full duplex</td><td></td></tr>
<tr><td></td><td><b>2</b> = 100 Mbps HD</td><td>100 Mbps, half duplex</td><td></td></tr>
<tr><td></td><td><b>3</b> = 10 Mbps FD</td><td>10 Mbps, full duplex</td><td></td></tr>
<tr><td></td><td><b>4</b> = 10 Mbps HD</td><td>10 Mbps, half duplex</td><td></td></tr>
<tr><td>15</td><td>Service configuration</td><td>Disable services that are not required.
Each service is represented by a single bit. Bit 0, Lock configuration, can be used to prevent accidental changing of this parameter.
By default, all services are enabled and configuration is unlocked.</td><td></td></tr>
<tr><td></td><td colspan="3">
<table>
<tr><th>Bit</th><th>Name</th><th>Information</th></tr>
<tr><td>0</td><td>Lock configuration</td><td>Changing this bit to one will lock service configuration and no bit can be changed. Only resetting the fieldbus configuration will unlock the parameter. To reset the fieldbus configuration, choose "Reset all fieldbus settings" or "Clear all" in parameter 96.06 Parameter restore.</td></tr>
<tr><td>1</td><td>Disable IP config tool</td><td>When this bit is set, access from ABB IP Configuration tool is prevented.</td></tr>
<tr><td>2</td><td>Disable ETH tool network</td><td>When this bit is set, access from Ethernet tool network (eg, ABB Drive Composer tool) is prevented.</td></tr>
<tr><td>3</td><td>Disable ping response</td><td>When this bit is set, response to ICMP (ping) message is prevented.</td></tr>
<tr><td>4</td><td>Unsecured ETH tool network</td><td>When this bit is set, access from Ethernet tool network is unsecured.
<b>Note:</b> Drive Composer pro before V2.7 supports unsecured communication only.</td></tr>
<tr><td>5</td><td>Disable configuring web pages</td><td>When this bit is set, access to web pages is disabled.</td></tr>
</table>
</td></tr>
<tr><td></td><td>0000b…1111b</td><td>Service configuration</td><td></td></tr>
</table>

No.	Name/Value	Description	Default
16 … 26	Reserved	These parameters are not used by the adapter module when using the NONE protocol.	N/A
27	FBA A/B par refresh	Validates any changed adapter module configuration parameter settings. After refreshing, the value reverts automatically to **0** = Done. **Note:** This parameter cannot be changed while the drive is running.	**0** = Done
	0 = Done	Refreshing done	
	1 = Refresh	Refreshing	
28	FBA A/B par table ver	**Read-only.** Displays the parameter table revision of the fieldbus adapter module mapping file stored in the memory of the drive. In format **xyz**, where **x** = major revision number **y** = minor revision number **z** = correction number OR in format **axyz**, where **a** = major revision number **xy** = minor revision numbers **z** = correction number or letter.	N/A
		Parameter table revision	
29	FBA A/B drive type code	**Read-only**. Displays the drive type code of the fieldbus adapter module mapping file stored in the memory of the drive.	N/A
		Drive type code of the fieldbus adapter module mapping file	
30	FBA A/B mapping file ver	**Read-only.** Displays the fieldbus adapter module mapping file revision stored in the memory of the drive in decimal format.	N/A
		Mapping file revision	
31	D2FBA A/B comm status	**Read-only.** Displays the status of the fieldbus adapter module communication. **Note:** The value names may vary by drive.	**0** = Idle or **4** = Offline or **2** = Time out
	0 = Idle	Adapter is not configured.	
	1 = Exec.init	Adapter is initializing.	
	2= Time out	A timeout has occurred in the communication between the adapter and the drive.	
	3= Conf.err	There is an internal error in the communication between the adapter and the drive. Contact your local ABB representative.	
	4 = Off-line	Adapter is off-line.	
	5 = On-line	Adapter is on-line.	
	6 = Reset	Adapter is performing a hardware reset.	

No.	Name/Value	Description	Default
32	FBA A/B comm SW ver	**Read-only.** Displays patch and build numbers of the adapter module's firmware version in **xxyy** format, where: **xx** = patch number **yy** = build number. Example: If the firmware version (<major>.<minor>.<patch>.<build>) is 3.10.200.13, the value C80D is displayed. If the version is 3.10.0.0, the value 0 is displayed. See also parameter 33.	N/A
33	FBA A/B appl SW ver	**Read-only.** Displays major and minor revision numbers of the adapter module's firmware version in **xxyy** format, where: **xx** = major revision number **yy** = minor revision number Example: If the firmware version (<major>.<minor>.<patch>.<build>) is 3.10.200.13 or 3.10.0.0, the value 310 is displayed. See also parameter 32.	N/A

Starting up fieldbus communication

1. Power up the drive.
2. Enable communication between the adapter module and the drive by selecting the correct slot number in parameter 50.01 FBA A enable.

   The selection must correspond to the slot where the adapter module is installed. For example, if the adapter module is installed in slot 1, you must select slot 1.
3. Set the module configuration parameters in group 51.
   - select the communication protocol and profile with parameter 51.02, and
   - configure the network settings with parameters 51.03…51.13.
4. Save the valid parameter values to permanent memory with parameter 96.07 Parameter save manually.
5. Validate the settings made in parameter groups 51 with parameter 51.27 FBA A par refresh.

11

NONE – Diagnostics

Contents of this chapter

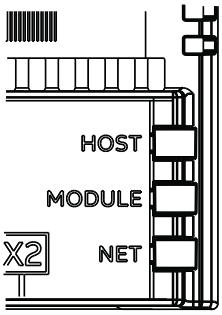
This chapter explains how to trace faults with the status LEDs on the adapter module using the NONE protocol.

Fault and warning messages

For the fault and warning messages concerning the adapter module, see the drive firmware manual.

LEDs

The adapter module is equipped with three bicolor diagnostic LEDs. The LEDs are described below.



Name	Color	Function
HOST	Flashing green	Establishing communication to drive
	Green	Connection to drive OK
	Flashing red	Communication to drive lost temporarily
	Flashing orange, alternating with the MODULE flashing orange	Internal file system error. The error may be cleared by cycling drive power. If the error persists, contact your local ABB representative.
MODULE	Off	There is no PC tool connected to the device.
	Flashing orange	Device is attempting to obtain IP configuration from the DHCP server.
	Orange	Device is executing Duplicate Address Detection.
	Green	PC tool is connected to the device.
	Flashing red	Ethernet link is down.
	Red	Ethernet interface is disabled. Duplicate Address Detection may have detected a duplicate address. Check the IP configuration and either initiate a Fieldbus Adapter parameter refresh or cycle power to the drive.
	Flashing orange, alternating with the HOST flashing orange	Internal file system error. The error may be cleared by cycling drive power. If the error persists, contact your local ABB representative.
NETWORK / NET	Off	Ethernet link is down.
	Flashing green	Ethernet link is up at 100 Mbps. Flashing indicates activity on interface.
	Flashing orange	Ethernet link is up at 10 Mbps. Flashing indicates activity on interface.

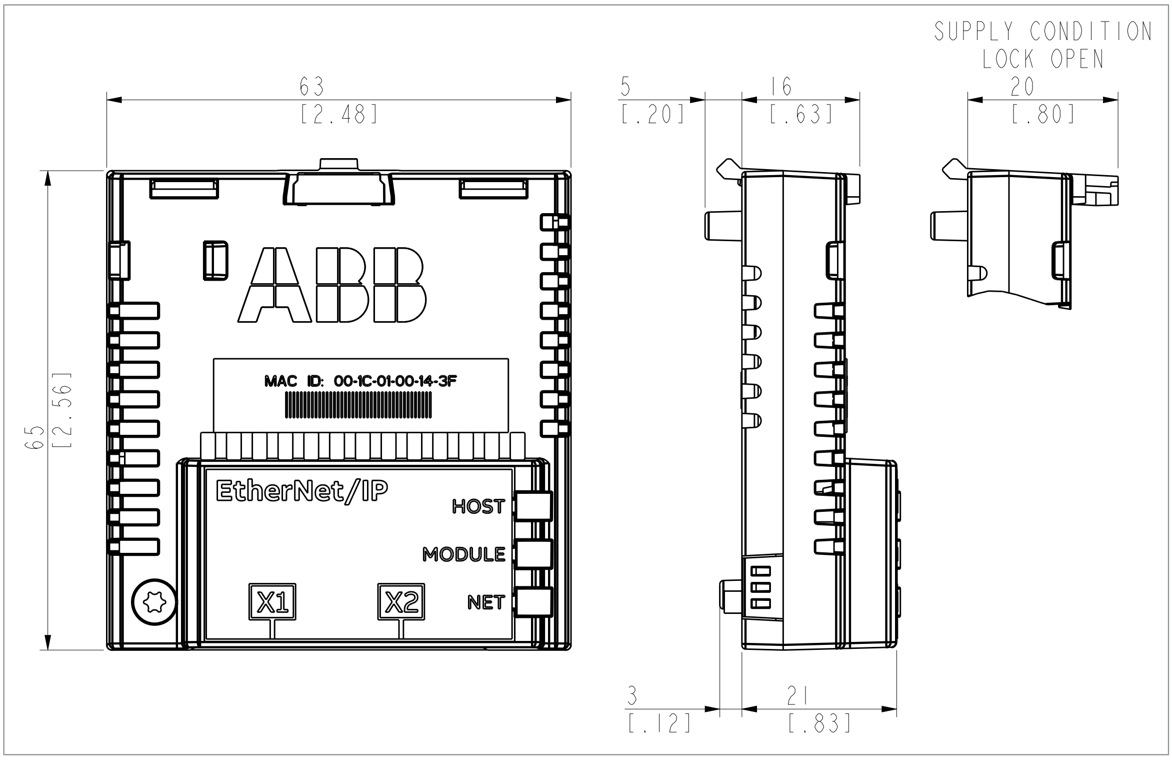
12

Technical data

Contents of this chapter

This chapter contains the technical specifications of the FEIP-21 module.

Dimension drawing



General data

Installation	Into an option slot on the drive control unit
Degree of protection	IP20
Ambient conditions	The applicable ambient conditions specified for the drive in its manuals are in effect.
Package	Cardboard. Plastic wrapping: Antistatic air bubble sheet (PE).
Indicators	Three bicolor LEDs (HOST, MODULE, NETWORK/NET)
Connectors	A 20-pin connector to the drive RJ-45 connector to Ethernet (X1) RJ-45 connector for chaining another adapter module (X2)
Power supply	+3.3 V ±5% max. 400 mA (supplied by the drive)
General	Complies with EMC standard EN 61800-3:2004 Printed circuit board conformal coated

Ethernet link

Compatible devices	Ethernet Standard IEEE 802.3 and IEEE 802.3u devices.
Medium	10BASE-TX or 100Base-TX with Auto-negotiation and Auto-MDIX (Auto-crossover) • Wiring: CAT5e/6 S/FTP, CAT5e/6 S/STP, CAT5e/6 SF/FTP • Connector: RJ-45 • Termination: Internal • Maximum segment length: 100 m / 328 ft
Topology	Bus, star or ring. Max. 50 nodes allowed for FEIP-21 module in a ring topology. In a chain maximum recommended amount of nodes is 50.
Transfer rate	10 Mbps or 100 Mbps
Serial communication type	Half or full duplex
Protocol	EtherNet/IP
Number of connections	Three

TCP and UDP service ports

There are multiple in-bound and out-bound network services running on the module. Some ports are protocol-specific and are not used when other protocols are selected.

Port	Service	Purpose
80 (TCP)	HTTP	HTTP protocol, used for Ethernet tool Network (like Drive composer pro) To disable, go to Service Configuration parameter 51.15.
68 (UDP)	DHCP	DHCP client **Note:** Used only when IP configuration method is selected as "Dyn IP DHCP".

Port	Service	Purpose
24576 (UDP)	ABB Netconfig	• Auto discovery protocol • Used by ControlBuilder plus (IP Configuration tool) and Drive composer pro tools • Discovers ABB-specific Ethernet devices in a local network segment, by listening to and responding to UDP broadcasts. To disable, go to **Service Configuration parameter 51.15** or to Service configuration web page.
44818 (TCP)	Ethernet/IP	Ethernet/IP, explicit messaging. **Note:** Used only when Ethernet/IP protocol is selected
2222 (UDP)	Ethernet/IP	Ethernet/IP, implicit messaging. **Note:** Used only when Ethernet/IP protocol is selected
123 (UDP)	SNTP	Simple Network Time Protocol. This service is disabled by default. To enable the service, go to Service configuration page.
443 (TCP)	HTTPS	HTTPS protocol, used for access to FEIP-21's web page and for Ethernet tool Network (like Drive Composer pro).

13

Appendix A - ABB IP configuration tool

Contents of this chapter

This chapter shows how to use the ABB IP configuration tool to:

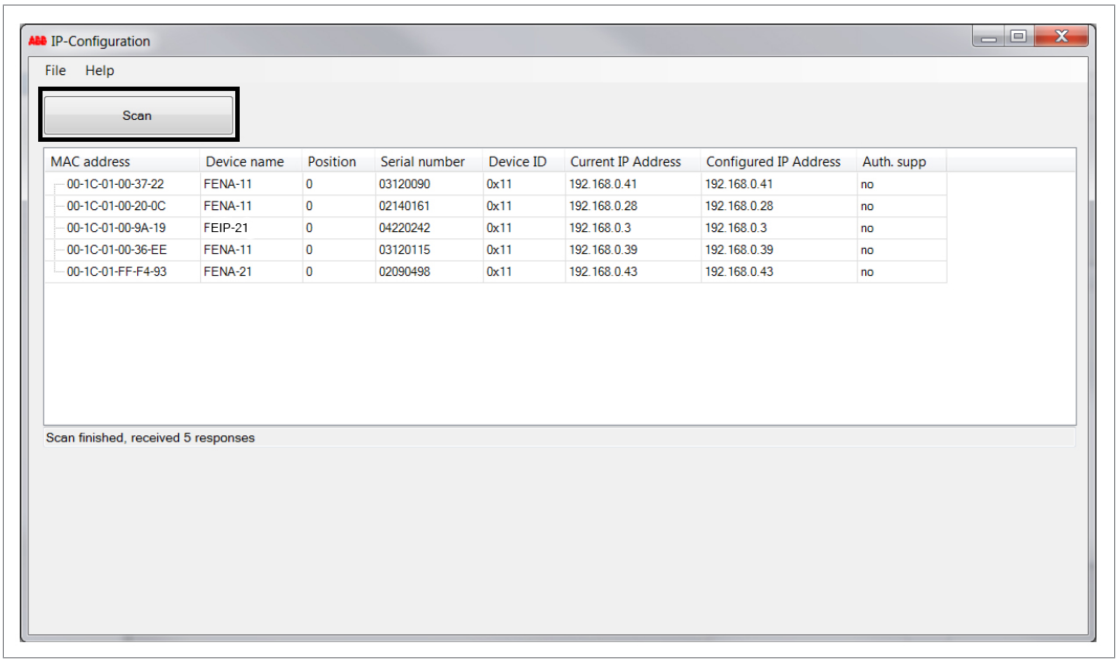
- find configured and unconfigured **FEIP-21** adapter modules in the network
- rewrite the IP configuration of the adapter modules.

Installation

The ABB IP configuration tool is part of the ABB Automation Builder software. No separate installation is needed.

Finding adapter modules in the network

1. Open the ABB IP configuration tool.
2. Click the **Scan** button.
   The FEIP-21 adapter modules present in the network appear on the results list.

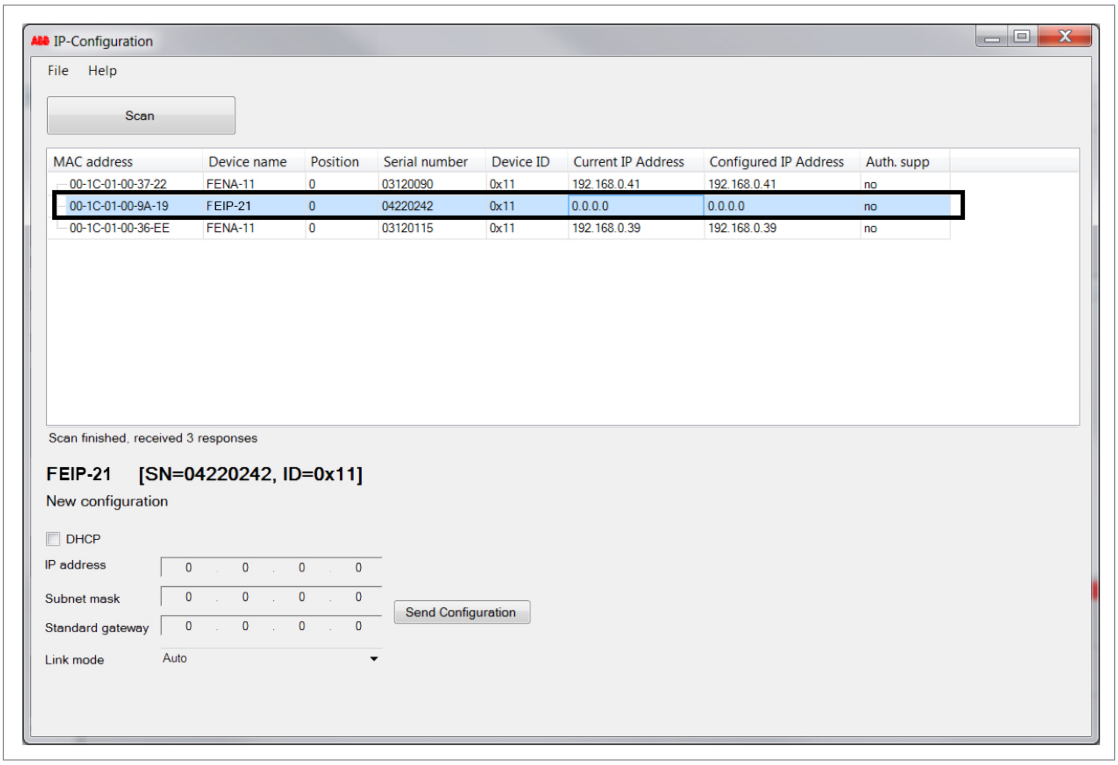


MAC address	Device name	Position	Serial number	Device ID	Current IP Address	Configured IP Address	Auth. supp
00-1C-01-00-37-22	FENA-11	0	03120090	0x11	192.168.0.41	192.168.0.41	no
00-1C-01-00-20-0C	FENA-11	0	02140161	0x11	192.168.0.28	192.168.0.28	no
00-1C-01-00-9A-19	FEIP-21	0	04220242	0x11	192.168.0.3	192.168.0.3	no
00-1C-01-00-36-EE	FENA-11	0	03120115	0x11	192.168.0.39	192.168.0.39	no
00-1C-01-FF-F4-93	FENA-21	0	02090498	0x11	192.168.0.43	192.168.0.43	no

Scan finished, received 5 responses

Rewriting the IP configuration of adapter modules

1. Scan the network for adapter modules.
   For instructions, see section Finding adapter modules in the network (page 127).
2. On the results list, click to select the adapter module whose IP configuration you want to modify.



MAC address	Device name	Position	Serial number	Device ID	Current IP Address	Configured IP Address	Auth. supp
00-1C-01-00-37-22	FENA-11	0	03120090	0x11	192.168.0.41	192.168.0.41	no
00-1C-01-00-9A-19	FEIP-21	0	04220242	0x11	0.0.0.0	0.0.0.0	no
00-1C-01-00-36-EE	FENA-11	0	03120115	0x11	192.168.0.39	192.168.0.39	no

Scan finished, received 3 responses

FEIP-21 [SN=04220242, ID=0x11]

New configuration

☐ DHCP

IP address 0 . 0 . 0 . 0

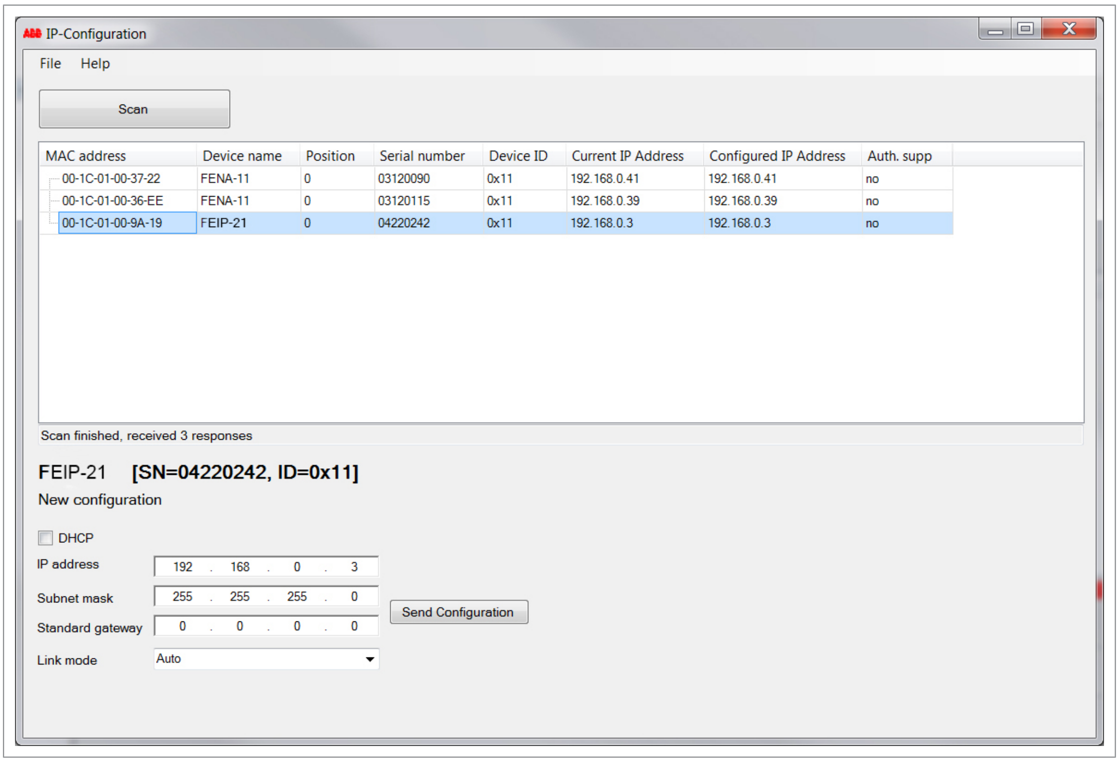
Subnet mask 0 . 0 . 0 . 0

Standard gateway 0 . 0 . 0 . 0

Link mode Auto

3. Below **New configuration**, define the IP configuration settings according to your network configuration.

4. If you want the adapter module to use a static IP address instead of **DHCP**, clear the DHCP check box.
5. To apply the new settings, click the **Send Configuration** button.
   The new current IP address and configured IP address appear on the results list.



MAC address	Device name	Position	Serial number	Device ID	Current IP Address	Configured IP Address	Auth. supp
00-1C-01-00-37-22	FENA-11	0	03120090	0x11	192.168.0.41	192.168.0.41	no
00-1C-01-00-36-EE	FENA-11	0	03120115	0x11	192.168.0.39	192.168.0.39	no
00-1C-01-00-9A-19	FEIP-21	0	04220242	0x11	192.168.0.3	192.168.0.3	no

Scan finished, received 3 responses

FEIP-21 **[SN=04220242, ID=0x11]**

New configuration

□ DHCP

IP address 192 . 168 . 0 . 3

Subnet mask 255 . 255 . 255 . 0

Standard gateway 0 . 0 . 0 . 0

Link mode Auto

14

Appendix B - Module configuration backup

Contents of this chapter

This chapter presents the settings for FEIP-21 configuration backup.

Compatibility

FEIP-21 settings are stored in the drive parameters and also in the configuration files. FEIP-21 adapter module supports backup of all settings to the drive. These settings are now also included in any backups made of the drive using the Drive composer PC tool or the control panel.

Settings for backup

Consider the following points:

- Backup is not slot-specific. For example, backup of FEIP-21 in FBA A, slot 1 can be restored to FEIP-21 FBA A, slot 2.
- Backup depends on the fieldbus channel. For example, backup of FEIP-21 in FBA A is not restored to FEIP-21 in FBA B.
- FEIP-21 configuration parameters are included in the backup when drive parameters are saved.

Configuration backup for all protocols in FEIP-21

The settings are saved to the drive after 10 seconds. If a Refresh command is given to FEIP-21 using parameter 51.27, the pending backup is transferred to drive immediately and FEIP-21 is rebooted after the transfer is completed.

Note: The new setting is not saved to drive if the drive was powered off or the adapter was disconnected from the drive within 10 seconds of changing a setting.

15

Appendix C - Replacing FENA-xx module with FEIP-21

Contents of this chapter

This chapter shows the configurations to replace FENA-xx with FEIP-21.

Compatibility

FEIP-21 supports automatic configuration of fieldbus settings and service configuration. The automatic configuration can be performed when FENA-xx is replaced with FEIP-21 and it is configured to use an Ethernet/IP profile. In case of other profiles (for example, Modbus), a manual configuration is needed.

Automatic configuration process

FEIP-21 reads the fieldbus configuration parameters from the drive (group A) during the first initialization. When FEIP-21 detects a FENA-xx or Ethernet usage, FEIP-21 accepts the configuration settings read from the drive and overwrites the configuration back to the drive.

The configuration for FENA-xx (v3.10 or later) is stored in a backup file on the drive, for example, ACS880, v2.6 or later, ACS380/ACS580, v2.04 or later.

During the first initialization, FEIP-21 checks the backup configuration of module type and protocol from the drive. When FEIP-21 detects that a FENA-xx or Ethernet/IP was used, FEIP-21 reads the backup file from the drive and extracts the service configuration parameters configured through FENA-xx web page. The parameters recognized by FEIP-21 (for example, ping response, IP config tool, ETH tool network) are applied and the unrecognized parameters are ignored.

After replacing FENA-xx with FEIP-21, FEIP-21 automatically enables the configuration lock to prevent unwanted modifications to the service configuration.

16

Appendix D - FEIP-21 configuration web pages

Contents of this chapter

This chapter presents the FEIP-21 configuration web pages.

Note: ABB recommends to disable the web pages after the commissioning to reduce cyber security risks. See the Service configuration page (page 140).

Browser requirements

Any web browser can be used.

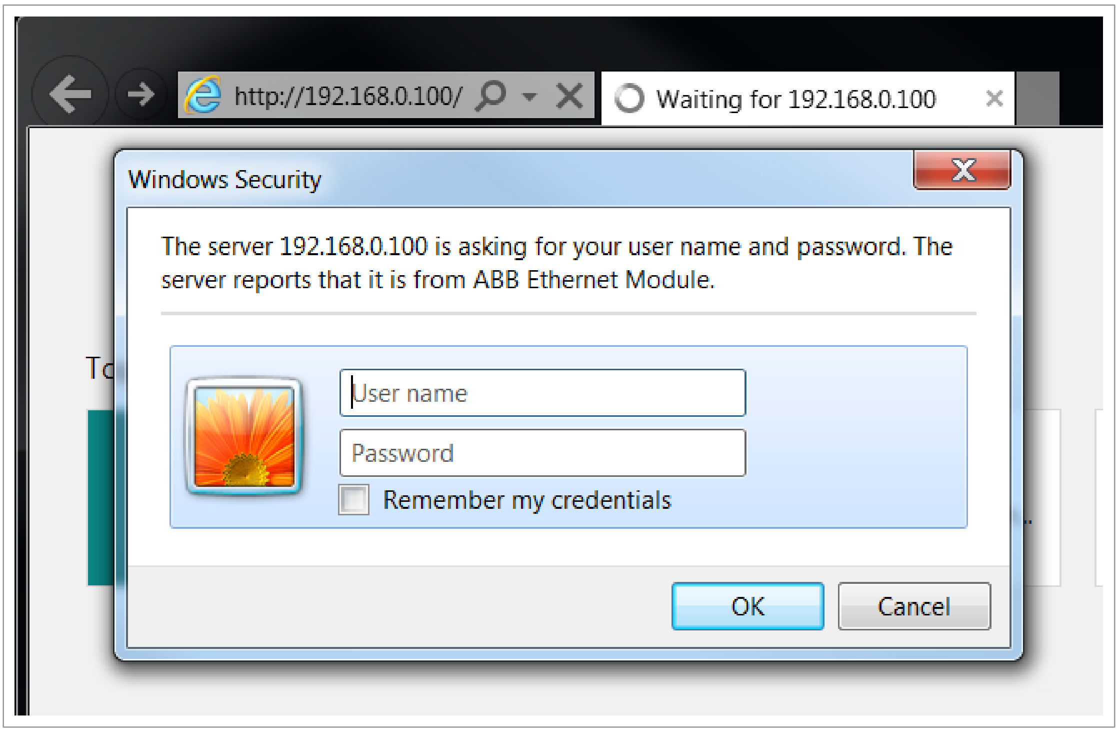
Compatibility

The web pages support all drives compatible with the FEIP-21 adapter module. For the compatibility table, see section Drives (page 13).

Logging in

1. Open a web browser and type the IP address of the adapter module in the address field. The IP address is visible in the FEIP-21 configuration parameters, group A, parameter 5...8.

   Example: http://192.168.0.100/



http://192.168.0.100/

Waiting for 192.168.0.100

Windows Security

The server 192.168.0.100 is asking for your user name and password. The server reports that it is from ABB Ethernet Module.

User name

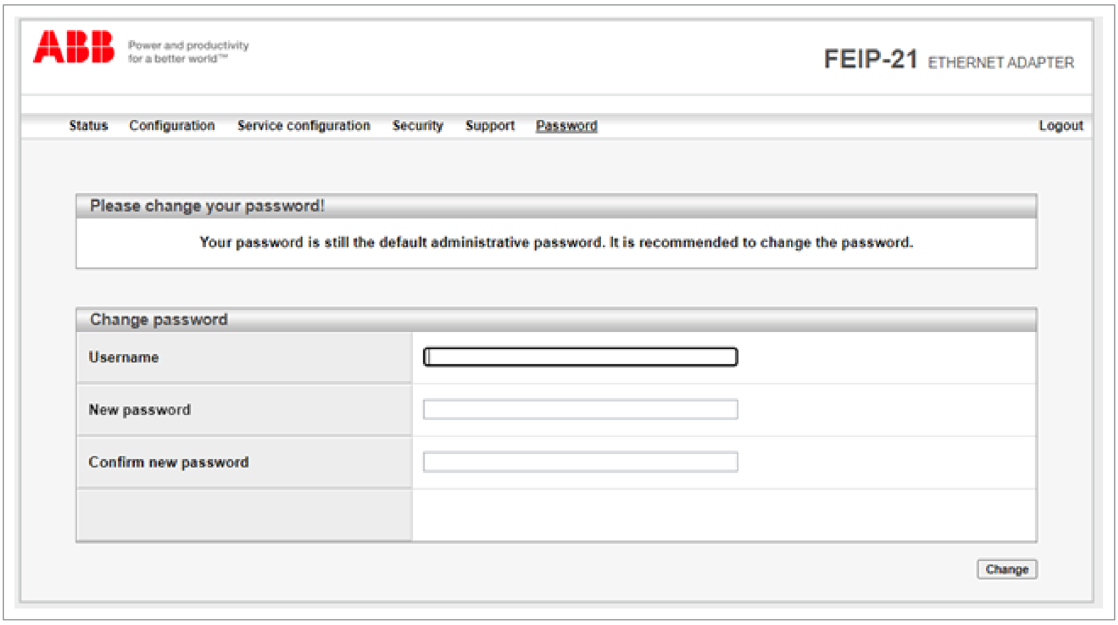
Password

□ Remember my credentials

2. Log in with a user name and a password.
   Default user name: admin
   Password: The last six digits of the MAC address of the adapter module, in upper case, without hyphens.
   The MAC ID is visible on the cover of the adapter module and in the ABB IP configuration tool, see Appendix A - ABB IP configuration tool (page 127).

   Example: If the MAC address of the adapter module is 00-1C-01-00-2F-73, the password is 002F73.
   The browser opens the user interface.

3. After successful login, you will be prompted to change the password for security reasons. It is recommended to change the default password.



Please change your password!

Your password is still the default administrative password. It is recommended to change the password.

Change password

Username	
New password	
Confirm new password	

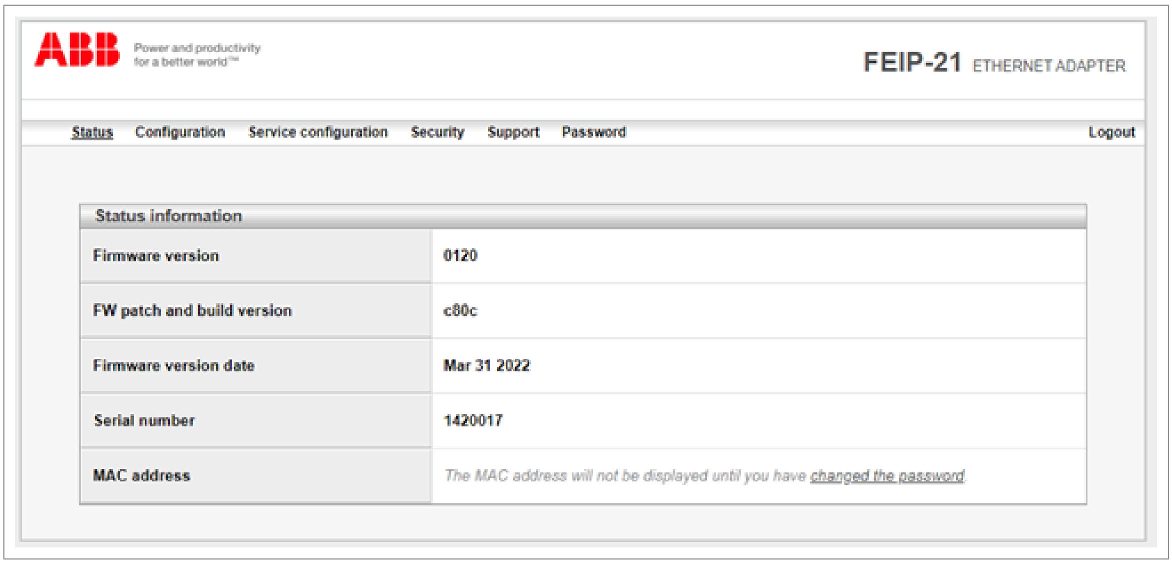
Menu overview

To navigate on the web pages, use the menu items available:

- Status
- Configuration
- Service configuration
- Security
- Support
- Password.

Status page

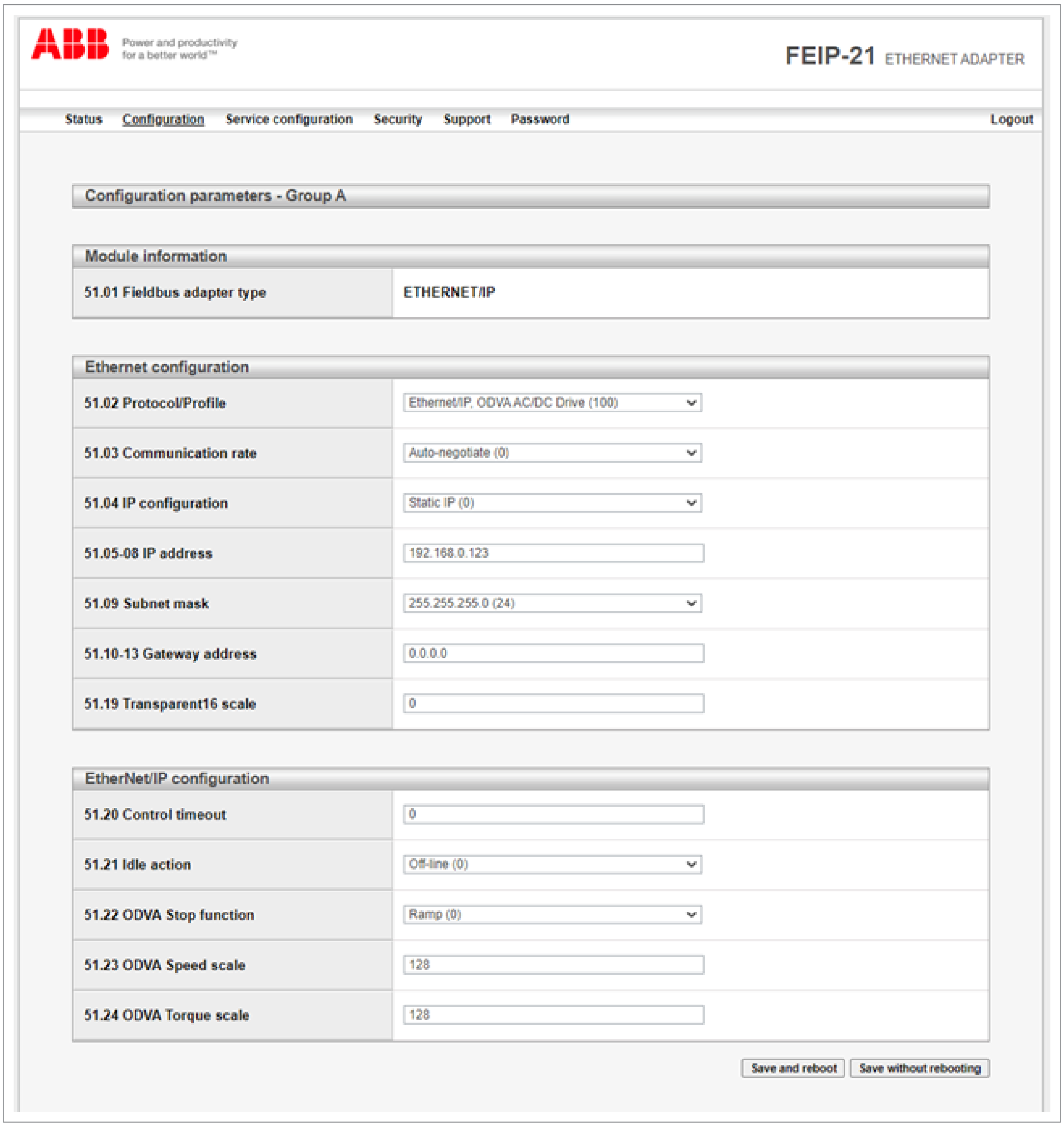
The Status page shows version information, as well as the serial number and MAC address (MAC ID) of the adapter module.



Status information	
Firmware version	0120
FW patch and build version	c80c
Firmware version date	Mar 31 2022
Serial number	1420017
MAC address	*The MAC address will not be displayed until you have changed the password*

Configuration page

On the Configuration page, you can modify parameter settings in the configuration parameter groups A (1), B (2) and C (3).

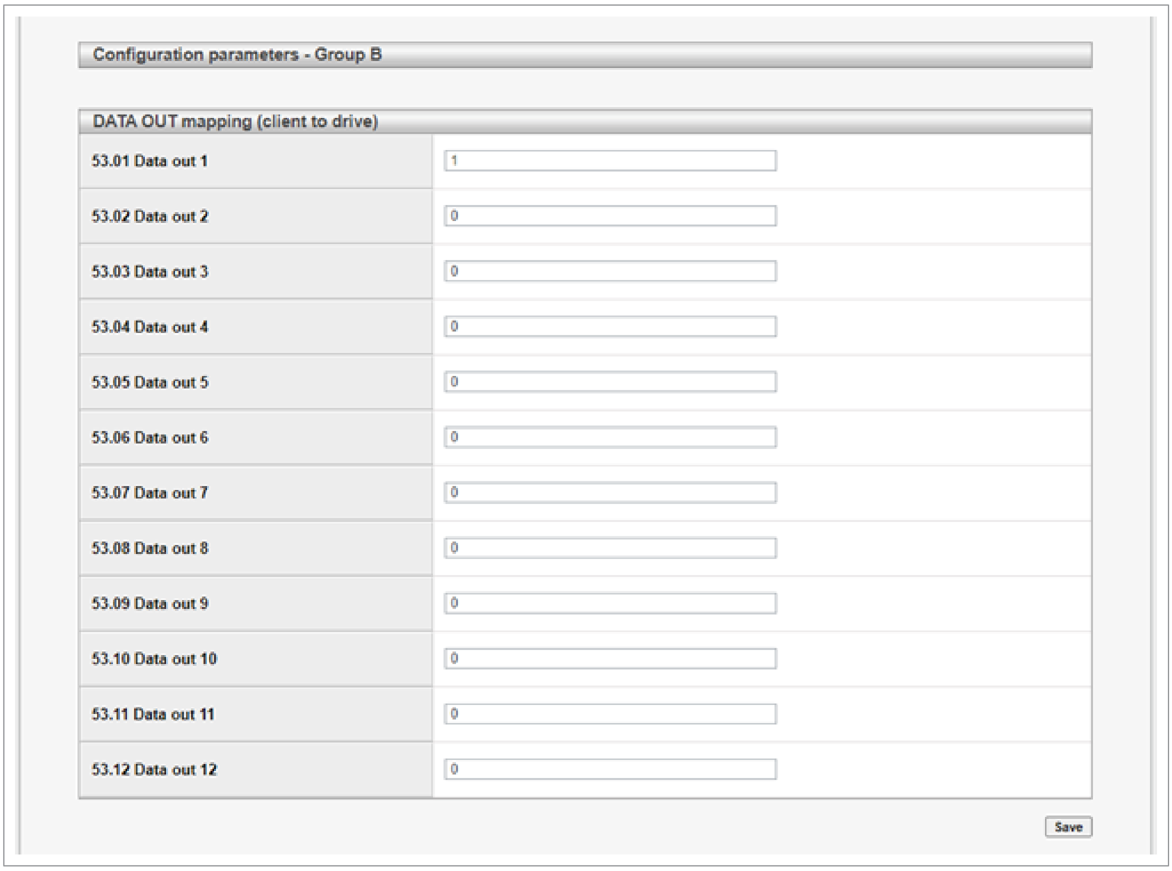


Configuration parameters - Group A

Module information	
51.01 Fieldbus adapter type	ETHERNET/IP

Ethernet configuration	
51.02 Protocol/Profile	EthernetIP, ODVA AC/DC Drive (100)
51.03 Communication rate	Auto-negotiate (0)
51.04 IP configuration	Static IP (0)
51.05-08 IP address	192.168.0.123
51.09 Subnet mask	255.255.255.0 (24)
51.10-13 Gateway address	0.0.0.0
51.19 Transparent16 scale	0

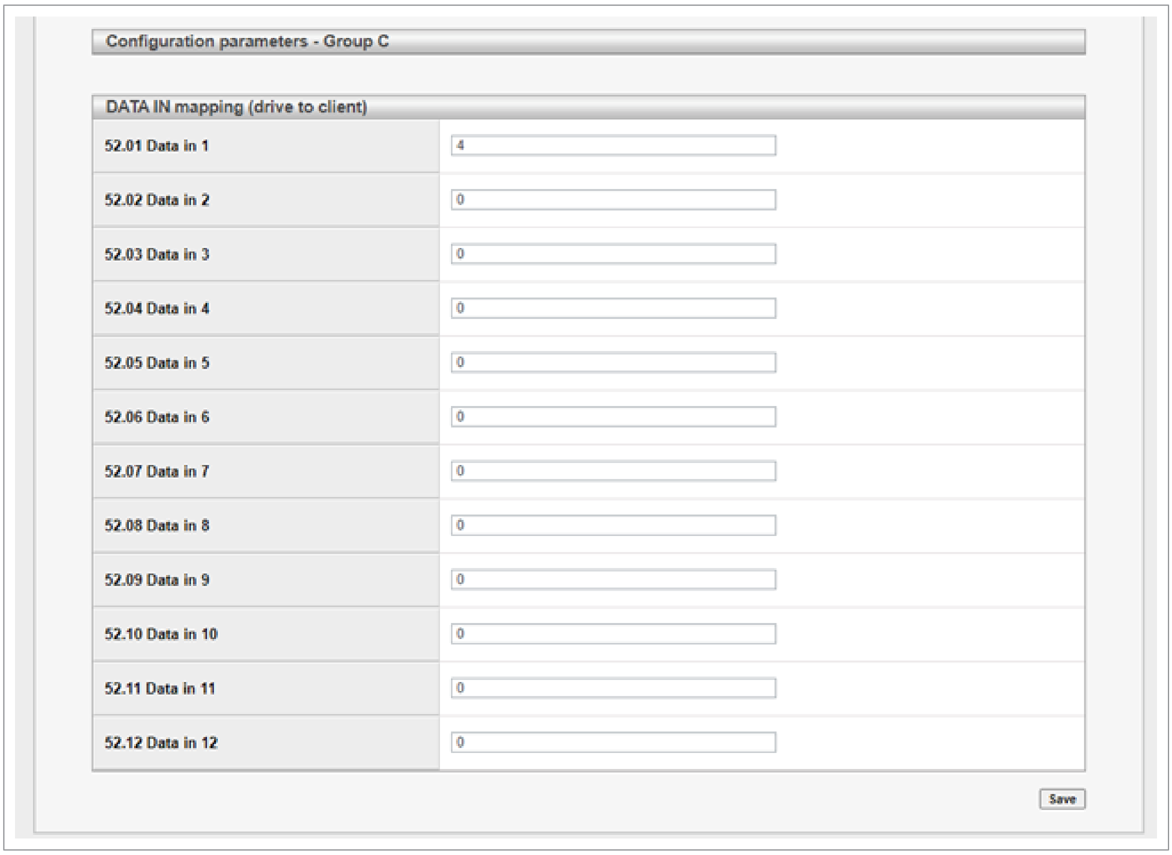
EtherNet/IP configuration	
51.20 Control timeout	0
51.21 Idle action	Off-line (0)
51.22 ODVA Stop function	Ramp (0)
51.23 ODVA Speed scale	128
51.24 ODVA Torque scale	128



Configuration parameters - Group B

DATA OUT mapping (client to drive)

53.01 Data out 1	1
53.02 Data out 2	0
53.03 Data out 3	0
53.04 Data out 4	0
53.05 Data out 5	0
53.06 Data out 6	0
53.07 Data out 7	0
53.08 Data out 8	0
53.09 Data out 9	0
53.10 Data out 10	0
53.11 Data out 11	0
53.12 Data out 12	0



Configuration parameters - Group C

DATA IN mapping (drive to client)

52.01 Data in 1	4
52.02 Data in 2	0
52.03 Data in 3	0
52.04 Data in 4	0
52.05 Data in 5	0
52.06 Data in 6	0
52.07 Data in 7	0
52.08 Data in 8	0
52.09 Data in 9	0
52.10 Data in 10	0
52.11 Data in 11	0
52.12 Data in 12	0

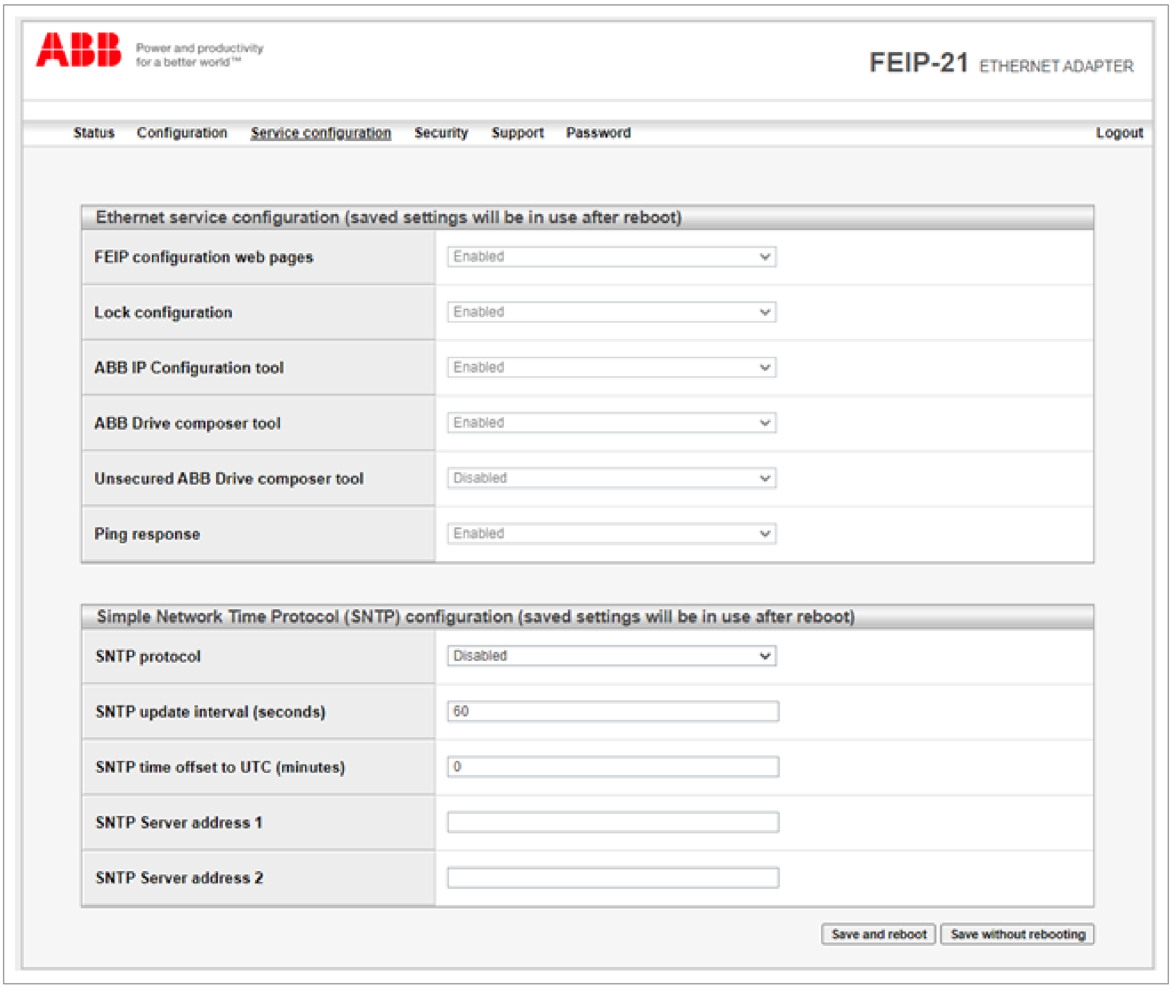
After you have changed any setting in any of the groups, you must click Save and reboot at the bottom of Group A to validate the settings.

Service configuration page

On the Service configuration page, you can enable or disable certain Ethernet services. All services except Simple Network Time Protocol (SNTP) are enabled by default. You can disable the following services on this page:

- access to FEIP-21 configuration web page
- allow to change IP settings remotely via ABB IP configuration tool
- remote access drive with Drive composer tool via Ethernet tool network
- Ping response
- configure SNTP.

The new settings take effect after reboot of the module. You can click Save and reboot, to validate the new settings immediately or click Save without rebooting if you want to do other settings also and then reboot.



Ethernet service configuration (saved settings will be in use after reboot)	
FEIP configuration web pages	Enabled
Lock configuration	Enabled
ABB IP Configuration tool	Enabled
ABB Drive composer tool	Enabled
Unsecured ABB Drive composer tool	Disabled
Ping response	Enabled

Simple Network Time Protocol (SNTP) configuration (saved settings will be in use after reboot)	
SNTP protocol	Disabled
SNTP update interval (seconds)	60
SNTP time offset to UTC (minutes)	0
SNTP Server address 1	
SNTP Server address 2	

Note: These settings are available only through web pages. When you select to disable the web page, a warning appears to confirm before you can save the selection.

Or, you can use the web page password. See Enable web page access after it was disabled (page 143).

It is recommended to disable all services that are not used after commissioning.

Configuring SNTP

Simple Network Time Protocol (SNTP) can be used to synchronize drive time with a network time server. When SNTP is enabled, FEIP-21 will request the time from configured server at a given interval. To receive this time synchronization, you must configure the drive with parameter 96.20 time sync primary source. The table below shows the settings for SNTP:

Settings	Description	Value
SNTP update interval	Interval to request time from server.	Default: 30 seconds Minimum: 30 seconds
SNTP time offset to UTC	Time offset to the time received from SNTP. This value can also be set over Ethernet/IP Class 91h, Instance 10, attribute 1. **Note:** SNTP time offset change does not need a reboot.	-1440...1440 minutes
SNTP server address 1	Primary server address for requesting time. Format: IP address followed by optional port number, eg: 192.168.0.1:123 **Note:** If port number is missing, the default NTP port number "123" is used.	-
SNTP server address 2	Secondary server address used if the request to server 1 fails.	-

■ Security page

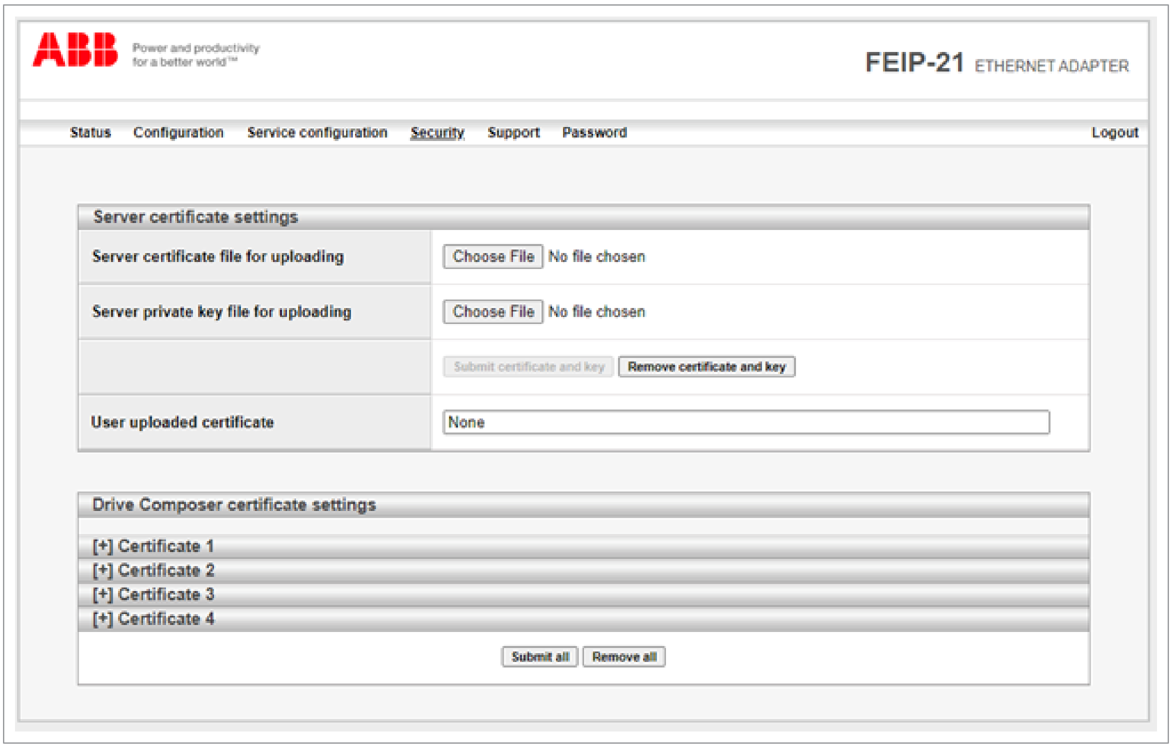
On the Security page, you can upload a private key and certificates for FEIP-21 to use instead of the self-signed certificate that FEIP-21 uses as a default. You can upload and manage Drive Composer certificate files on Security page under "Drive Composer certificate settings". Up to four certificates are supported at a time.

FEIP-21 uses secure HTTPS (TLS 1.2) communication for the web page server. By default, FEIP-21 does not require client authentication for the PC tool communication. You can use client authentication to improve the system security.

If unsecured communication is required, you can enable it from Service configuration page or by setting parameter 51.15 Service configuration bit 4. When unsecured communication is enabled, the PC tool communication is not encrypted or authenticated.

For more details, refer to Drive composer start-up and maintenance PC tool user's manual (3AUA0000094606 [English]).

Note: Secure PC tool communication is possible with Drive Composer Pro version 2.7 or later.



FEIP-21 ETHERNET ADAPTER

Status Configuration Service configuration Security Support Password Logout

Server certificate settings

Server certificate file for uploading	Choose File No file chosen
Server private key file for uploading	Choose File No file chosen
	Submit certificate and key / Remove certificate and key
User uploaded certificate	None

Drive Composer certificate settings

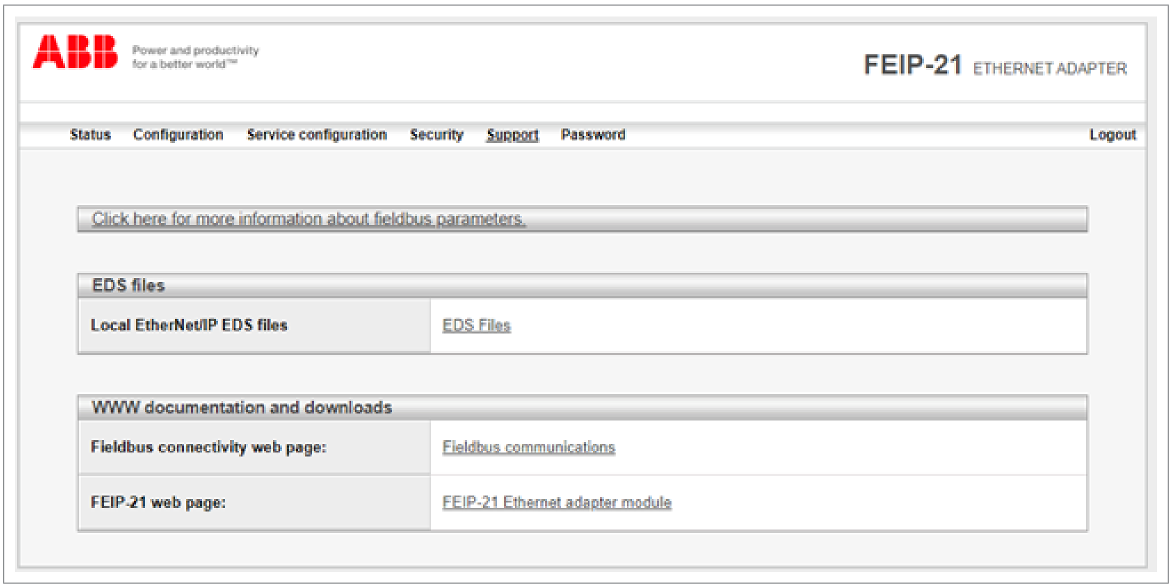
[+] Certificate 1
[+] Certificate 2
[+] Certificate 3
[+] Certificate 4

Submit all / Remove all

Support page

On the Support page, you can access documentation related to the adapter module and the EDS file.

The latest files corresponding to the drive firmware are available through the hyperlinks listed under "WWW documentation and downloads". You can find more information about each parameter under "Click here for more information about fieldbus parameters".



FEIP-21 ETHERNET ADAPTER

Status Configuration Service configuration Security Support Password Logout

Click here for more information about fieldbus parameters.

EDS files

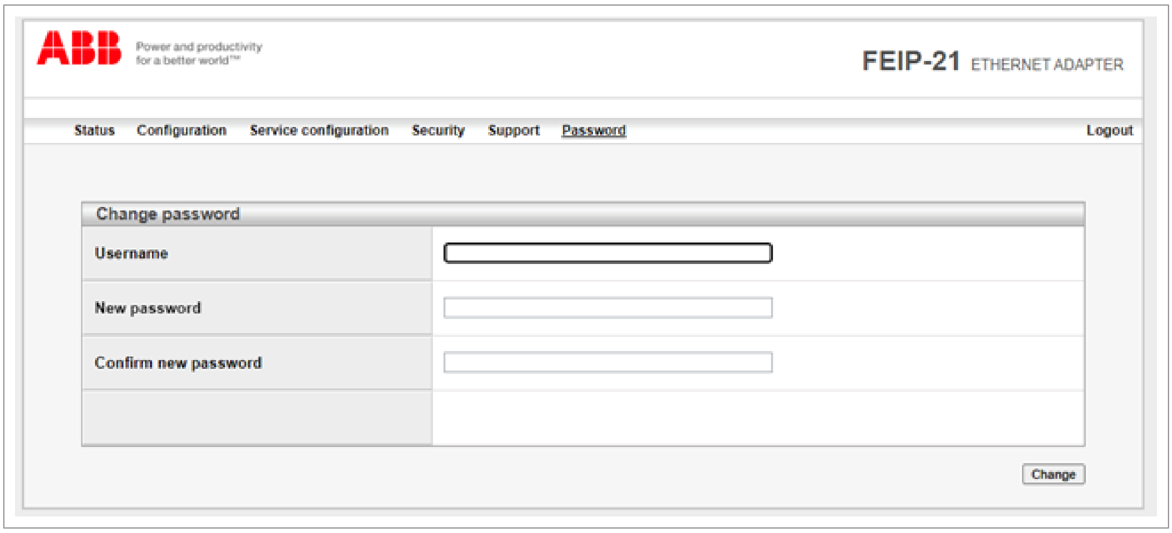
Local EtherNet/IP EDS files	EDS Files

WWW documentation and downloads

Fieldbus connectivity web page:	Fieldbus communications
FEIP-21 web page:	FEIP-21 Ethernet adapter module

Password page

On the Password page, you can change password. FEIP-21 supports only one user access level.



Reset FEIP-21 web page password to default

You can reset the FEIP-21 web page password to factory default.

Note: The password can be reset only with local access to the drive.

1. Disconnect all cable connections to FEIP-21.
   The NET led should switch off.
2. Write 0 (zero) to the parameter 26 under Group A (e.g. 51.26).
3. Refresh the settings by selecting Refresh in parameter 27 (e.g. 51.27).
4. Write 17989 to parameter 26 under Group A.
5. Refresh the settings by selecting Refresh in parameter 27.
6. Write 20033 to parameter 26 under Group A.
7. Refresh the settings by selecting Refresh in parameter 27.
8. Write 0 to parameter 26 under Group A.

FEIP-21 password is now reset to the default password. For information of default password, see section Logging in (page 135).

Enable web page access after it was disabled

You can enable the access to web pages with drive parameters.

1. Disconnect all cable connections to FEIP-21.
   The NET led should switch off.
2. Write 0 (zero) to the parameter 26 under Group A (e.g. 51.26).
3. Refresh the settings by selecting Refresh in parameter 27 (e.g. 51.27).
4. Write 87 to parameter 26 under Group A.
5. Refresh the settings by selecting Refresh in parameter 27.
6. Write 17730 to parameter 26 under Group A.
7. Refresh the settings by selecting Refresh in parameter 27.
8. Write 0 to parameter 26 under Group A.

Access to web pages is now enabled.

Further information

Product and service inquiries

Address any inquiries about the product to your local ABB representative, quoting the type designation and serial number of the unit in question. A listing of ABB sales, support and service contacts can be found by navigating to www.abb.com/searchchannels.

Product training

For information on ABB product training, navigate to new.abb.com/service/training.

Providing feedback on ABB manuals

Your comments on our manuals are welcome. Navigate to new.abb.com/drives/manuals-feedback-form.

Document library on the Internet

You can find manuals and other product documents in PDF format on the Internet at www.abb.com/drives/documents.

a3 (frozen)
PDF-A4
Created 2022-05-25, 10:38:34

www.abb.com/drives
www.abb.com/solarinverters
www.abb.com/windconverters

3AXD50000158621C

© Copyright 2022 ABB. All rights reserved.
Specifications subject to change without notice.

3AXD50000158621 Rev C (EN) 2022-05-25