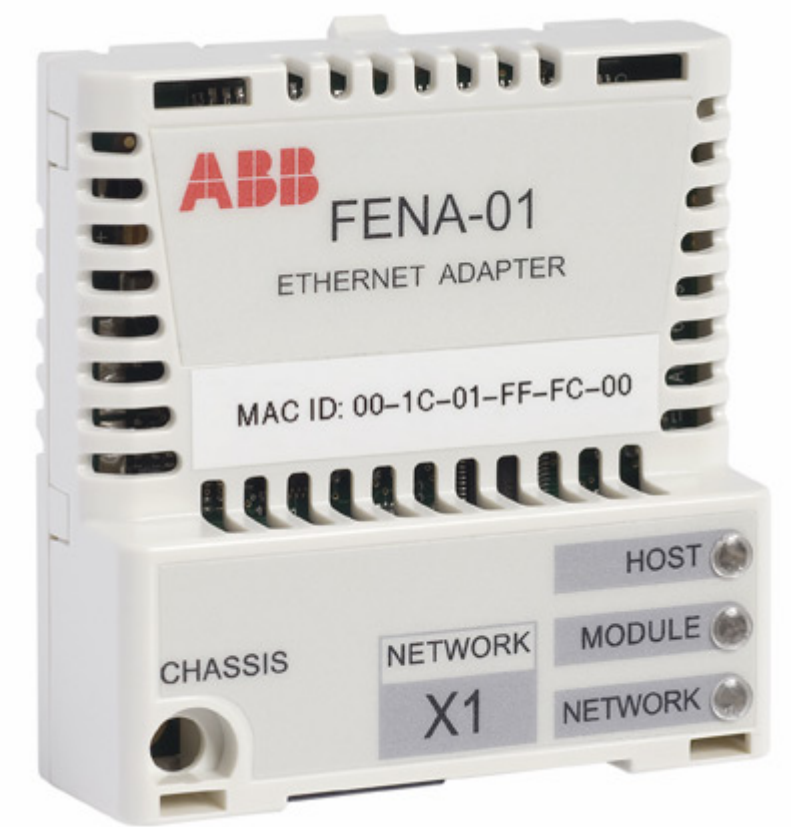
ABB Drives

Protocol Manual - Ethernet/IP
Ethernet Adapter Module
FENA-01



Ethernet Adapter Module - EtherNet/IP
FENA-01

Protocol Manual

3AUA0000033371 REV A
EN
EFFECTIVE: 31.3.2008

Safety instructions

Overview

This chapter states the general safety instructions that must be followed when installing and operating the FENA-01 Ethernet Adapter module.

The material in this chapter must be studied before attempting any work on, or with, the unit.

In addition to the safety instructions given below, read the complete safety instructions of the specific drive you are working on.

General safety instructions

WARNING! All electrical installation and maintenance work on the drive should be carried out by qualified electricians.

The drive and adjoining equipment must be properly earthed.

Do not attempt any work on a powered drive. After switching off the mains, always allow the intermediate circuit capacitors 5 minutes to discharge before working on the frequency converter, the motor or the motor cable. It is good practice to check (with a voltage indicating instrument) that the drive is in fact discharged before beginning work.

The motor cable terminals of the drive are at a dangerously high voltage when mains power is applied, regardless of motor operation.

There can be dangerous voltages inside the drive from external control circuits even when the drive mains power is shut off. Exercise appropriate care when working on the unit. Neglecting these instructions can cause physical injury or death.

Table of contents

Introduction

Intended audience

The manual is intended for people responsible for installing, commissioning and using an FENA-01 Ethernet Adapter module for EtherNet/IP communication. The reader is expected to have a basic knowledge of electrical fundamentals, electrical wiring practices and how to operate the drive.

Before you start

It is assumed that the drive is installed and ready to operate before starting the installation of the extension module.

In addition to conventional installation tools, have the drive manuals available during the installation as they contain important information not included in this manual. The drive manuals are referred to at various points of this document.

What this manual contains

This manual contains information on the configuration and use of the FENA-01 Ethernet Adapter module with the EtherNet/IP protocol.

Safety instructions are featured in the first few pages of this manual.

Overview contains short descriptions of the EtherNet/IP protocol and the FENA-01 Ethernet Adapter module, and a delivery checklist.

Drive configuration explains how to program the drive before the communication through the adapter module can be started.

Client configuration explains how to program the EtherNet/IP client before communication through the adapter module can be started.

Communication profiles describes the communication profiles used in the communication between the EtherNet/IP client, the FENA-01 module and the drive.

Communication contains a description of the EtherNet/IP functionality supported by the FENA-01.

Diagnostics explains how to trace faults with the status LEDs on the FENA-01 module.

Definitions and abbreviations explains definitions and abbreviations concerning EtherNet/IP on the FENA-01.

Further Information

Further information on the EtherNet/IP protocol is available on the world wide web from www.odva.org.

Overview

Overview

The FENA-01 Ethernet Adapter module supports the EtherNet/IP network protocol. This chapter contains a short description of EtherNet/IP and the FENA-01 Ethernet Adapter module.

For information on Ethernet standards, including media and topologies, see the Hardware Manual for the FENA-01 Ethernet 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 implementation of the EtherNet/IP server in the FENA-01 module is done according to

- 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

The supported EtherNet/IP features are listed in chapter *"Communication"*. Two simultaneous EtherNet/IP connections are supported.

EtherNet/IP on FENA-01 Ethernet Adapter Module

The FENA-01 Ethernet Adapter module is an optional device for ABB drives which enables the connection of the drive to an

Ethernet network. The module supports a variety of higher-level communications protocols, including EtherNet/IP. Through the FENA-01 Ethernet Adapter module it is possible to:

- give control commands to the drive (Start, Stop, Run enable, etc.)
- 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
- change drive parameter values
- reset a drive fault.

The FENA-01 acts as a EtherNet/IP server with support for ODVA AC/DC Drive, ABB Drives and Transparent profiles. The EtherNet/IP features supported by the FENA-01 Ethernet Adapter module are discussed in chapter *"Communication"*.

The adapter module is mounted into an option slot on the motor control board of the drive. See the drive documentation for module placement options.

Compatibility

The FENA-01 is compatible with 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

Drive configuration

Overview

This chapter gives information on configuring the FENA-01 Ethernet Adapter module for use with EtherNet/IP.

Ethernet connection configuration

After the FENA-01 Ethernet Adapter module has been mechanically and electrically installed according to the FENA-01 Hardware Manual, the drive must be prepared for communication with the module.

The detailed procedure of activating the module for Ethernet communication with the drive is dependent on the drive type. Normally, a parameter must be adjusted to activate the communication (see the drive documentation).

As communication between the drive and the FENA-01 is established, several configuration parameters are copied to the drive. These parameters (Tables 1, 5 and 6) must be checked first and adjusted where necessary. The alternative selections for these parameters are discussed in more detail below the tables.

Note: The new settings take effect only when the module is powered up the next time or when a ‘Fieldbus Adapter parameter refresh’ is given (see the drive documentation).

*Table 1. FENA-01 Configuration Parameters - Group #1 **

Par. no.	Parameter name	Alternative settings	Default setting
1	FBA TYPE	(Read-only)	ETHERNET
2	PROTOCOL/ PROFILE	Modbus/TCP: (0) ABB Drives Classic (1) ABB Drives Enhanced (2) Transparent 16-bit (3) Transparent 32-bit EtherNet/IP (100) ODVA AC/DC Drive (101) ABB Drives Profile (102) Transparent 16-bit (103) Transparent 32-bit	(0) Modbus/TCP - ABB Drives Classic
3	COMMRATE	(0) Auto-negotiate; (1) 100 Mbps, Full Duplex (2) 100 Mbps, Half Duplex (3) 10 Mbps, Full Duplex (4) 10 Mbps, Half Duplex	(0) Auto-negotiate
4	IP CONFIGURATION	(0) Static IP (1) Dynamic IP (DHCP)	(1) Dynamic IP (DHCP)
5	IP ADDRESS 1	0…255	0
6	IP ADDRESS 2	0…255	0
7	IP ADDRESS 3	0…255	0
8	IP ADDRESS 4	0…255	0
9	SUBNET CIDR	1…31	1
10	GW ADDRESS 1	0…255	0
11	GW ADDRESS 2	0…255	0
12	GW ADDRESS 3	0…255	0
13	GW ADDRESS 4	0…255	0

14 - 19	Reserved	N/A	N/A
20	Control Timeout	0…65535	0
21	Idle Action	(0) Off-line (1) On-line	0
22	ODVA Stop Function	(0) Ramp (1) Coast	0
23	ODVA Speed Scale	0…255	128
24	ODVA Torque Scale	0…255	128
25-26	Reserved	N/A	N/A

*Actual parameter group number depends on the drive type. Eg, Group#1 equals to parameter Group 51 in ACS350.

1 FBA TYPE

This parameter shows the fieldbus adapter type as detected by the drive. The value should not be adjusted by the user.

If this parameter is undefined, the communication between the drive and the module has not been established.

2 PROTOCOL/PROFILE

Selects the application protocol and communication profile for the network connection.

0 = Modbus/TCP protocol with ABB Drives Classic profile.

1 = Modbus/TCP protocol with ABB Drives Enhanced profile.

2 = Modbus/TCP protocol with Transparent 16-bit profile.

3 = Modbus/TCP protocol with Transparent 32-bit profile.

100 = EtherNet/IP protocol with ODVA AC/DC Drive profile.

101 = EtherNet/IP protocol with ABB Drives profile.

102 = EtherNet/IP protocol with Transparent 16-bit profile.

103 = EtherNet/IP protocol with Transparent 32-bit profile.

3 COMMRATE

Sets the bit rate for the Ethernet interface.

0 = Auto-negotiate

1 = 100 Mbit/s, full duplex

2 = 100 Mbit/s, half duplex

3 = 10 Mbit/s, full duplex

4 = 10 Mbit/s, half duplex

4 IP CONFIGURATION

Sets the method for configuring the IP address, subnet mask and gateway address for the module.

0 = Static IP: Configuration will be obtained from configuration parameters 5-13

1 = Dynamic IP: Configuration will be obtained via DHCP

DHCP, Dynamic Host Configuration Protocol, is a protocol for automating the configuration of IP devices. DHCP can be used to automatically assign IP addresses and related network information.

5 IP ADDRESS 1

6 IP ADDRESS 2

7 IP ADDRESS 3

8 IP ADDRESS 4

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. These parameters define the four octets of the IP address.

9 SUBNET CIDR

Subnet masks are used for splitting networks into smaller networks called subnets. A subnet mask is a 32-bit binary number that is used to split 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.

Dotted Decimal	CIDR	Dotted Decimal	CIDR
255.255.255.254	31	255.254.0.0	15
255.255.255.252	30	255.252.0.0	14
255.255.255.248	29	255.248.0.0	13
255.255.255.240	28	255.240.0.0	12
255.255.255.224	27	255.224.0.0	11
255.255.255.192	26	255.224.0.0	10
255.255.255.128	25	255.128.0.0	9
255.255.255.0	24	255.0.0.0	8
255.255.254.0	23	254.0.0.0	7
255.255.252.0	22	252.0.0.0	6
255.255.248.0	21	248.0.0.0	5
255.255.240.0	20	240.0.0.0	4
255.255.224.0	19	224.0.0.0	3
255.255.192.0	18	192.0.0.0	2
255.255.128.0	17	128.0.0.0	1
255.255.0.0	16		

10 GW ADDRESS 1

11 GW ADDRESS 2

12 GW ADDRESS 3

13 GW ADDRESS 4

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. These parameters define the four octets of the gateway address.

14-19 Reserved

These parameters are unused by the FENA-01 when configured for EtherNet/IP.

20 CONTROL TIMEOUT

The EtherNet/IP protocol specifies connection timeout for I/O Messaging (Class 1) and Connected Explicit Messaging (Class3), 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.

Table 2. Timeout Source

Connection Type	**Control Timeout**	**Timeout Source**
I/O Messaging (Class 1)	0…65535	(Requested Packet Interval) X (Connection Timeout Multiplier) Note: Timeout behavior may be modified by Watchdog Timeout Action attribute of Connection object.
Connected Explicit Messaging (Class 3)	0	(Requested Packet Interval) X (Connection Timeout Multiplier) Note: Timeout behavior may be modified by Watchdog Timeout Action attribute of Connection object.
	1…65534	100ms X (Control Timeout Value) since last Control Event
	65535	Never Timeout
Unconnected Explicit Messaging	0	Always Timeout Note: Control Timeout must be greater than zero to control drive with Unconnected Explicit Messaging.
	1…65534	100ms X (Control Timeout Value) since last Control Event
	65535	Never Timeout

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.

In the event of a timeout, the FENA-01 will signal the drive that communication with the client has been lost. The drive configuration will determine how it will respond. For example, if the timeout is configured for 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.

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 = In the event of a Idle notification, the FENA-01 will signal the drive that communication with the client has been lost. The drive configuration will determine how it will respond. For example, if the timeout is configured for 250 ms and the drive is configured to fault on a communication failure with a delay of 500ms, then the drive will fault 750ms after communications is lost.

1 = In the event of an Idle notification, the drive will continue to operate using the last command and references received.

22 ODVA STOP FUNC

This parameter only applies when using the ODVA AC/DC Drive Profile. It determines how the motor will be stopped when a stop command is received via EtherNet/IP

0 = Ramp stop: the motor decelerates along the active deceleration ramp.

1 = Coast stop: the motor comes to a stop by coasting.

23 ODVA SPEED SCALE

This parameter only applies when using the ODVA AC/DC Drive Profile. Units of reference and actual speeds for ODVA AC/DC Drive profiles are given by the formula below. This parameter does not affect ABB Drives profiles. Note that while a wide range of resolutions may be configured, actual performance will be limited to the performance capabilities of the drive.

Speed Unit = RPM X $2^{(-1 \text{ X ODVA Speed Scale Value})}$

Table 3. ODVA Speed Scaling

ODVA Speed Scale Value[1]	**Drive Parameter Speed Scale Value[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 “ODVA Speed Scale Value” when reading/writing ODVA Speed Scale via the AC/DC Drive Object (2Ah). When written via AC/DC Drive Object, the new value takes effect immediately.

[2] Use “Drive Parameter Speed Scale Value” when reading/writing ODVA Speed Scale via Drive Panel, Drive Parameter Object (90h) and Drive Configuration Object (91h). When written via these methods, the new value takes effect after the drive is repowered or a “Fieldbus Adapter Parameter refresh” is given.

24 ODVA TORQUE SCALE

This parameter only applies when using the ODVA AC/DC Drive Profile. Units of reference and actual torques for ODVA AC/DC Drive profiles are given by the formula below. This parameter does not affect ABB Drives profiles. Note that while a wide range of resolutions may be configured, actual performance will be limited to the performance capabilities of the drive. (Nm = Newton x Meter)

Torque Unit = Nm x $2^{(-1 \text{ X ODVA Torque Scale})}$

Table 4. ODVA Torque Scaling

ODVA Torque Scale Value[1]	**Drive Parameter Torque Scale Value[2]**	**Unit**
-5	123	32 Nm
-4	124	16 Nm
-3	125	8 Nm
-2	126	4 Nm
-1	127	2 Nm
0 (default)	128	1 Nm
1	129	0.5 Nm
2	130	0.25 Nm
3	131	0.125 Nm
4	132	0.0625 Nm
5	133	0.03125 Nm

[1]Use “ODVA Torque Scale Value” when reading/writing ODVA Torque Scale via the AC/DC Drive Object (2Ah). When written via AC/DC Drive Object, the new value takes effect immediately.

[2] Use “Drive Parameter Torque Scale Value” when reading/writing ODVA Torque Scale via Drive Panel, Drive Parameter Object (90h) and Drive Configuration Object (91h). When written via these methods, the new value takes effect after the drive is repowered or a “Fieldbus Adapter Parameter refresh” is given.

25-26 Reserved

These parameters are unused by the FENA-01 when configured for EtherNet/IP.

*Table 5. FENA-01 Configuration Parameters - Group 2**

Par. no. **	**Parameter name**	**Alternative settings**	**Default setting**
1	DATA OUT 1 (client to drive)	0…9999 Format: **xxyy**, where **xx** = parameter group and **yy** = parameter index.	0
2	DATA OUT 2	See DATA OUT 1 above.	0
3	DATA OUT 3	See DATA OUT 1 above.	0
4	DATA OUT 4	See DATA OUT 1 above.	0
5	DATA OUT 5	See DATA OUT 1 above.	0
6	DATA OUT 6	See DATA OUT 1 above.	0
7	DATA OUT 7	See DATA OUT 1 above.	0
8	DATA OUT 8	See DATA OUT 1 above.	0
9	DATA OUT 9	See DATA OUT 1 above.	0
10	DATA OUT 10	See DATA OUT 1 above.	0

*Actual parameter group number depends on the drive type. Eg, Group#2 equals to parameter Group 55 in ACS350.

** Number of parameters in this group may vary by drive.

1 DATA OUT 1

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. Content is specified by a decimal number as follows.

0	Not Used
1 - 99	Virtual Address Area of Drive
101 - 9999	Parameter Area of Drive

Parameter numbers are formatted as xxyy, where xx is the parameter group number (1 to 99) and yy is the parameter index within that group (01 to 99). For example, parameter 99.02 would be entered as 9902.

2-10 DATA OUT 2 to DATA OUT 10

See DATA OUT 1 above.

*Table 6. FENA-01 Configuration Parameters - Group 3**

Par. no. **	**Parameter name**	**Alternative settings**	**Default setting**
1	DATA IN 1 (drive to client)	0 to 9999 Format: **xxyy**, where **xx** = parameter group and **yy** = parameter index.	0
2	DATA IN 2	See DATA IN 1 above.	0
3	DATA IN 3	See DATA IN 1 above.	0
4	DATA IN 4	See DATA IN 1 above.	0
5	DATA IN 5	See DATA IN 1 above.	0
6	DATA IN 6	See DATA IN 1 above.	0
7	DATA IN 7	See DATA IN 1 above.	0
8	DATA IN 8	See DATA IN 1 above.	0
9	DATA IN 9	See DATA IN 1 above.	0
10	DATA IN 10	See DATA IN 1 above.	0

*Actual parameter group number depends on the drive type. Eg, Group#3 equals to parameter Group 54 in ACS350.

** Number of parameters in this group may vary by drive.

1 DATA IN 1

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. Content is specified by a decimal number as follows.

0	Not used
1 - 99	Virtual Address Area of Drive
101 - 9999	Parameter Area of Drive

Parameter numbers are formatted as xxyy, where xx is the parameter group number (1 to 99) and yy is the parameter index within that group (01 to 99). For example, parameter 99.02 would be entered as 9902.

2-10 DATA IN 2 to DATA IN 10

See DATA IN 1 above.

Control locations

ABB drives can receive control information from multiple sources including digital inputs, analogue inputs, the drive control panel and a communication module (eg, FENA-01). ABB drives allow the user to separately determine the source for each type of control information (Start, Stop, Direction, Reference, Fault Reset, etc.). In order to give the fieldbus client the most complete control over the drive, the communication module must be selected as source for this information. See the user documentation of the drive for information on the selection parameters.

Client configuration

Overview

This chapter gives information on configuring the EtherNet/IP client for communication through the FENA-01 Ethernet Adapter module.

Configuring the client

After the FENA-01 Ethernet Adapter module has been mechanically and electrically installed according to the instructions in the FENA-01 Hardware Manual, and has been initialized by the drive, the client must be prepared for communication with the module.

Select Protocol/Profile

During configuration of the drive it is necessary to select a communication protocol, in this case EtherNet/IP, and a communication profile. The communication profile determines what I/O Assemblies and Objects are available. See Communication Profiles 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 communication profile. Table 7. provides a listing of output and input assemblies

Table 7. Assembly Instances

Name	Output Instance	Input Instance	Size (bytes)	Profile
Basic Speed Control	20	70	4	ODVA AC/DC Drive
Enhanced Speed Control	21	71	4	ODVA AC/DC Drive
Basic Speed and Torque Control	22	72	6	ODVA AC/DC Drive
Enhanced Speed and Torque Control	23	73	6	ODVA AC/DC Drive
Basic Speed Control plus Drive Parameters	120	170	24	ODVA AC/DC Drive
Enhanced Speed Control plus Drive Parameters	121	171	24	ODVA AC/DC Drive
Basic Speed and Torque Control plus Drive Parameters	122	172	26	ODVA AC/DC Drive
Enhanced Speed and Torque Control plus Drive Parameters	123	173	26	ODVA AC/DC Drive
ABB Drives Profile w/ Set Speed	1	51	4	ABB Drives Profile
ABB Drives Profile w/ Set Speed and Set Torque	2	52	6	ABB Drives Profile
ABB Drives Profile w/ Set Speed plus Drive Parameters	101	151	24	ABB Drives Profile
ABB Drives Profile w/ Set Speed and Set Torque plus Drive Parameters	102	152	26	ABB Drives Profile
Transparent16 w/One	11	61	4	Transparent16 Profile
Transparent16 w/Two	12	62	6	Transparent16 Profile
Transparent16 w/One plus Drive Parameters	111	161	24	Transparent16 Profile
Transparent16 w/Two plus Drive Parameters	112	162	26	Transparent16 Profile
Transparent32 w/One	21	71	8	Transparent32 Profile

Name	Output Instance	Input Instance	Size (bytes)	Profile
Transparent32 w/Two	22	72	12	Transparent32 Profile
Transparent32 w/One plus Drive Parameters	121	171	28	Transparent32 Profile
Transparent32 w/Two plus Drive Parameters	122	172	32	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: Timeout behavior is significantly impacted by choice of connection method. Refer to Control Timeout and Idle Action configuration parameters in Drive Configuration for more information on timeout behavior.

The FENA-01 Ethernet Adapter supports:

I/O Connections

The FENA-01 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 FENA-01 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 is being controlled via unconnected explicit messaging.

Unconnected Explicit Messaging

The FENA-01 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 Control Timeout configuration parameter in Drive Configuration

---

Client configuration

Please refer to the client documentation for information on configuring the system for communication with the FENA-01.

EDS files

Electronic Data Sheet (EDS) files specify the properties of the device for the EtherNet/IP client. The device is identified by the EtherNet/IP client by means of the Product Code, Device Type, and Major Revision (See Identity Object 01h).

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 ABB website or your ABB representative.

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

Communication profiles

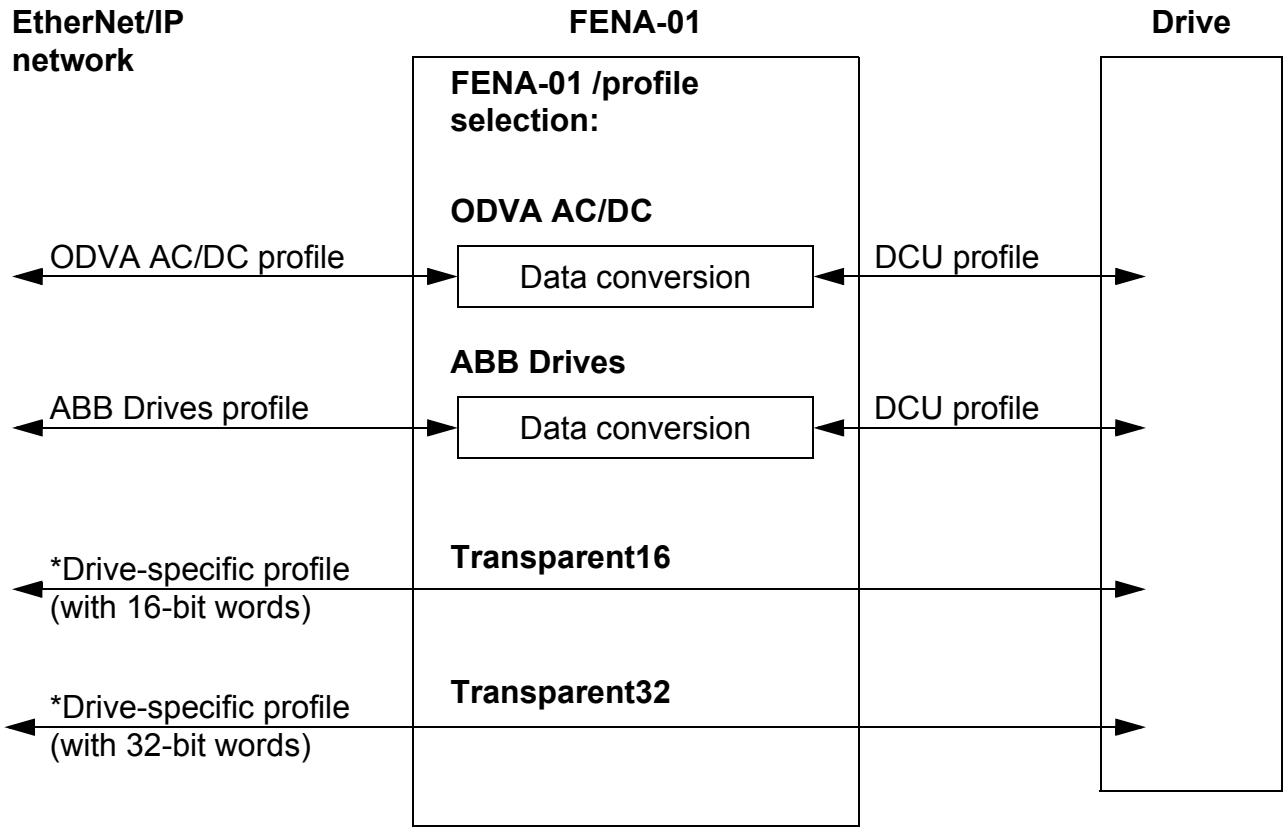
Overview

This chapter describes the communication profiles used in the communication between the EtherNet/IP client, the FENA-01 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 FENA-01 module, the EtherNet/IP network may employ either the ODVA AC/DC Drive profile or the ABB Drives profile. Both are converted to the DCU profile (detailed in drive documentation) by the FENA-01 module. In addition, two Transparent modes for 16 and 32 bit words respectively are available. With the Transparent modes, no data conversion takes place.



*To be used if the drive does not support the DCU communication profile.

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 DCU communication profile.

The ODVA AC/DC Drive Profiles

This section briefly describes the ODVA AC/DC Drive Profiles. Additional information can be obtained from www.odva.org.

An EtherNet/IP node is modeled as a collection of abstract objects. Each object represents the interface to and behaviour of a component within the product. The ODVA AC/DC Drive Profiles define a collection of objects suitable for the control of AC and DC drives. Objects supported by the FENA-01 EtherNet/IP Adapter are listed in Communications – Class Objects.

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 Attribute SpeedRef of the Class AC/DC Drive Object. 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. Assembly Instances supported by the FENA-01 are listed and defined in Communications – Assembly Objects.

ODVA Output Attributes

This section briefly describes the instances found in the ODVA AC/DC Drive Profiles output assemblies. Not all attributes listed here will be supported by all output assembly instances.

Run Forward & Run Reverse (Control Supervisor Object)

These attributes are used to assert run and stop commands to Control Supervisor Object state machine (see “State” below) according to the following table.

Table 8. Run/Stop event matrix

RunFwd	RunRev	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 Table 3. for details.

Scalar Mode

When the drive is operating in scalar mode, the FENA-01 provides the drive with a frequency reference. The ODVA AC/DC Drive Profiles use rpm units for the speed reference. The drive frequency reference is calculated according to

Dfr Drive Frequency Reference in Hz

Osr ODVA Speed Reference

Us ODVA Speed Unit (from Table 6)

Mf Motor Nominal Frequency in Hz

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

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

For a 4 pole 60 Hz motor (Mss = 1800 rpm) with a unit 1 rpm and an ODVA Speed Reference of 900.

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

Vector Mode

When the drive is operating in vector mode, the FENA-01 provides the drive with a speed reference. The ODVA AC/DC Drive Profiles use rpm units for the speed reference. The drive speed reference is calculated according to

Dsr Drive Speed Reference in rpm

Osr ODVA Speed Reference

Us ODVA Speed Unit (from Table 6).

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

For an ODVA Speed Reference of 900 rpm with a unit of 0.5 rpm.

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

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. SeeTable 4. for details.

The FENA-01 provides the drive with a torque reference in percent of motor nominal torque. The ODVA AC/DC Drive Profiles use Newton-meter (Nm) units for the torque reference. The drive torque reference is calculated according to

Dtr Drive Torque Reference in Percent of Motor Nominal Torque

Otr ODVA Torque Reference

Ut ODVA Torque Unit (from Table 7)

Mt Motor Nominal Torque in Nm.

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

For a 1000 Nm Motor Nominal Torque with a unit of 1 Nm and an ODVA Torque Reference of 500.

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

ODVA Input Attributes

This section briefly describes the instances found in the ODVA AC/DC Drive Profiles input assemblies. Not all attributes listed here will be supported by all input assembly instances.

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 (see “State” below) is in the Ready, Running or Stopping state.

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 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.

Table 9. Control Supervisor States.

State	Description	State	Description
0	Vendor Specific	4	Enabled
1	Startup	5	Stopping
2	Not Ready	6	Fault Stop
3	Ready	7	Faulted

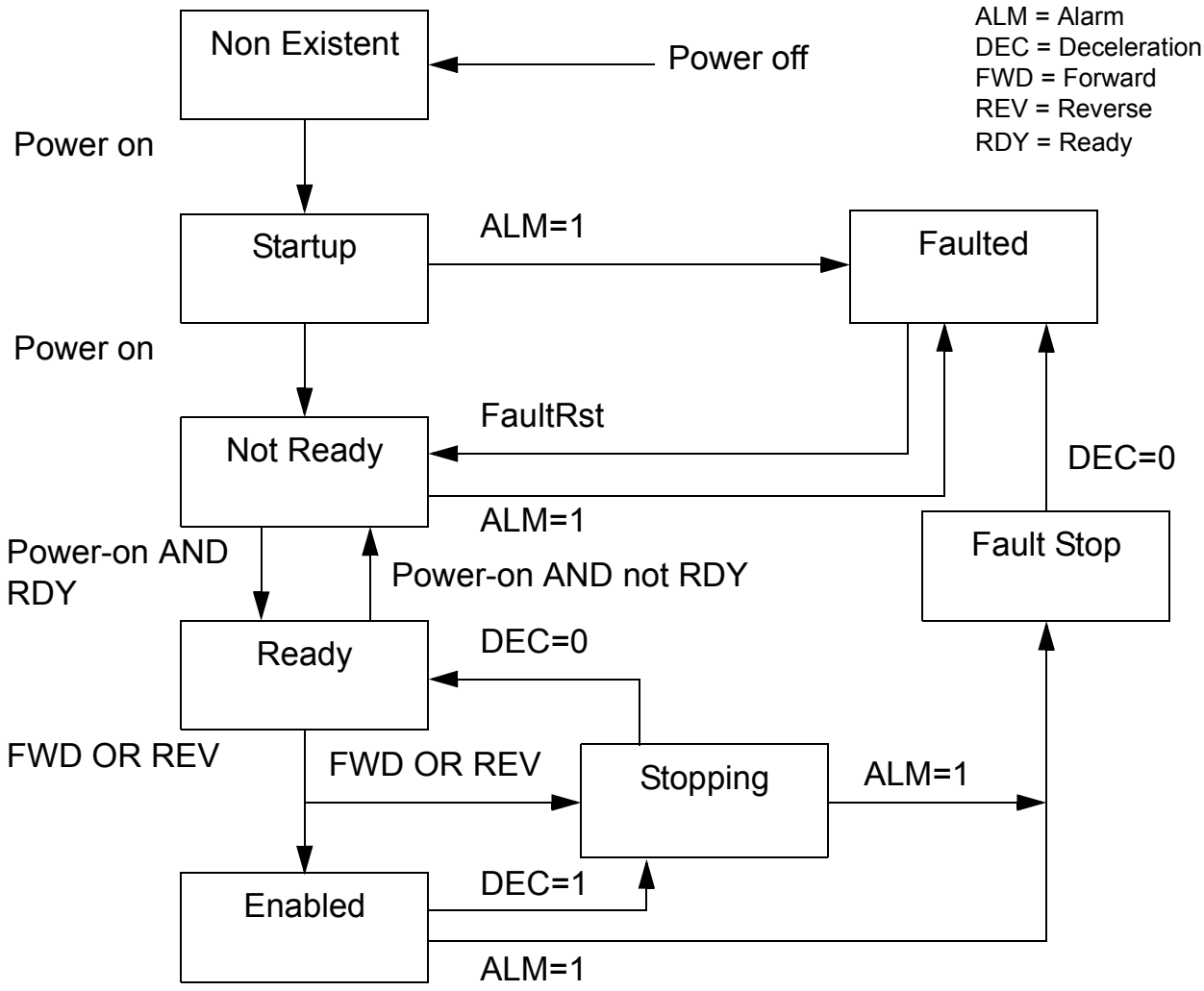


Figure 10. ODVA state transition diagram

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 Table 3. for details.

Scalar Mode

When the drive is operating in scalar mode, the drive provides the FENA-01 with a frequency actual. The ODVA AC/DC Drive Profiles use rpm units for the speed actual. The ODVA Speed Actual is calculated according to

Osa ODVA Speed Actual

Dfa Drive Frequency Actual in Hz

Us ODVA Speed Unit (from Table 6)

Mf Motor Nominal Frequency in Hz

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

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

For a 4 pole 60 Hz motor (Mss = 1800 rpm) with a unit of 1 rpm and a Drive Frequency Actual or 30 Hz.

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

Vector Mode

When the drive is operating in vector mode, the drive provides the FENA-01 with a speed actual. The ODVA AC/DC Drive Profiles use rpm units for the speed actual. The ODVA Speed Actual is calculated according to

Dsa Drive speed Actual in rpm

Osa ODVA Speed Actual

Us ODVA Speed Unit (from Table 6)

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

For a Drive Speed Actual of 900 rpm with a unit of 0.5 rpm.

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

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 Table 4. for details.

The drive provides the FENA-01 with a torque actual in percent of Motor Nominal Torque. The ODVA AC/DC Drive Profiles use Newton-meter (Nm) units for the torque actual. The ODVA Torque Actual is calculated according to

Dta Drive Torque Actual in Percent of Motor Nominal Torque

Ota ODVA Torque Actual

Ut ODVA Torque Unit (from Table 7)

Mt Motor Nominal Torque in Nm

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

For a 1000 Nm Motor Nominal Torque with a unit of 1 Nm and a drive torque actual of 50%.

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

The ABB Drives communication profile

The Control Word and the Status Word

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

The contents of the Control Word and the Status Word are detailed in Tables 11 and 12 respectively. The drive states are presented in the ABB Drives profile state machine (Figure 13).

References

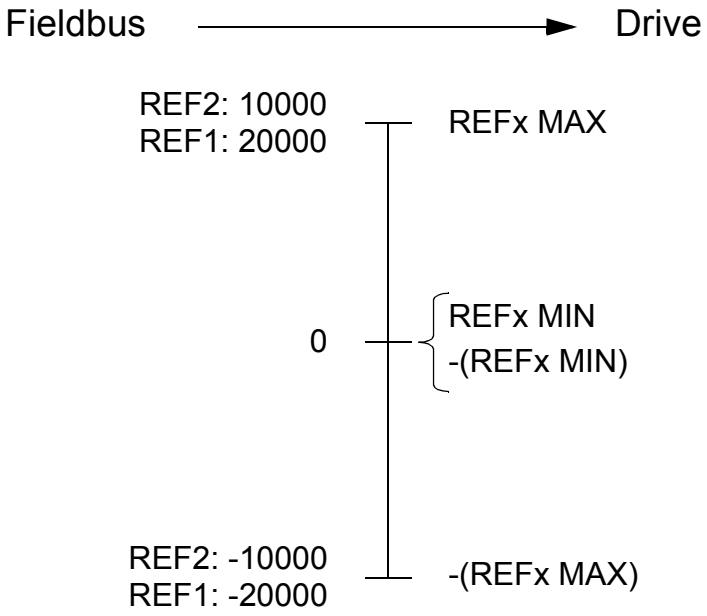
References are 16-bit words containing a sign bit and a 15-bit integer. A negative reference (indicating reversed direction of rotation) is formed by calculating the two's complement from the corresponding positive reference.

ABB drives can receive control information from multiple sources including analogue and digital inputs, the drive control panel and a communication module (eg, FENA-01). In order to have the drive controlled through the fieldbus, the module must be defined as the source for control information, eg, Reference.

Scaling

References are scaled as shown below.

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



Actual values

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

Scaling

Actual values are scaled as shown below.

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

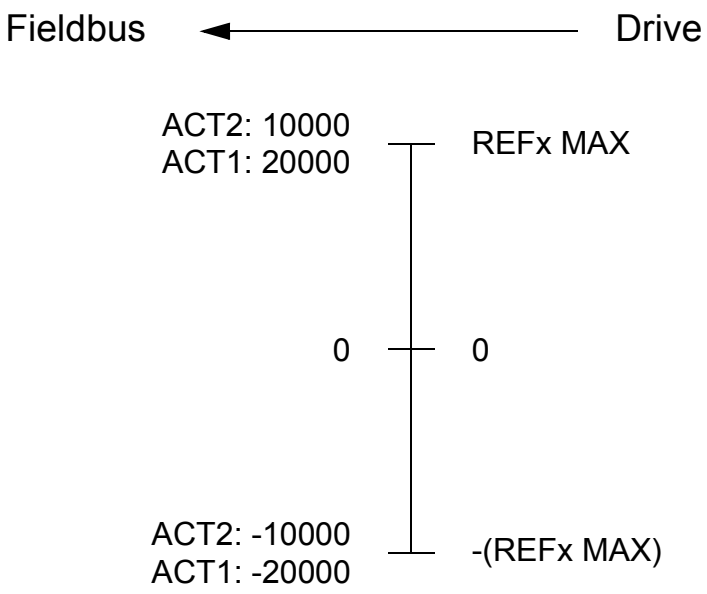


Table 11. The Control Word for the ABB Drives communication profile. The upper case boldface text refers to the states shown in Figure 13.

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 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).

Bit	Name	Value	STATE/Description
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 **OPERATING**. **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.
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 to 9	Reserved.		
10	REMOTE_ CMD	1	Fieldbus control enabled.
		0	Control Word <> 0 or Reference <> 0: Retain last Control Word and Reference. Control Word = 0 and Reference = 0: Fieldbus control enabled. Reference and deceleration/ acceleration ramp are locked.
11	EXT_CTRL_ LOC	1	Select External Control Location EXT2. Effective if control location parameterised to be selected from fieldbus.
		0	Select External Control Location EXT1. Effective if control location parameterised to be selected from fieldbus.
12 to 15	Reserved.		

Table 12. The Status Word for the ABB Drives communication profile. The upper case boldface text refers to the states shown in Figure 13.

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.
8	AT_ SETPOINT	1	**OPERATING**. Actual value equals reference = is within tolerance limits, i.e. 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.

Bit	Name	Value	STATE/Description
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.
		0	External Control Location EXT1 selected.
12	EXT_RUN_E NABLE	1	External Run Enable signal received
		0	No External Run Enable signal received
13 to 14	Reserved.		
15		1	Communication error detected by fieldbus adapter module.
		0	Fieldbus adapter communication OK.

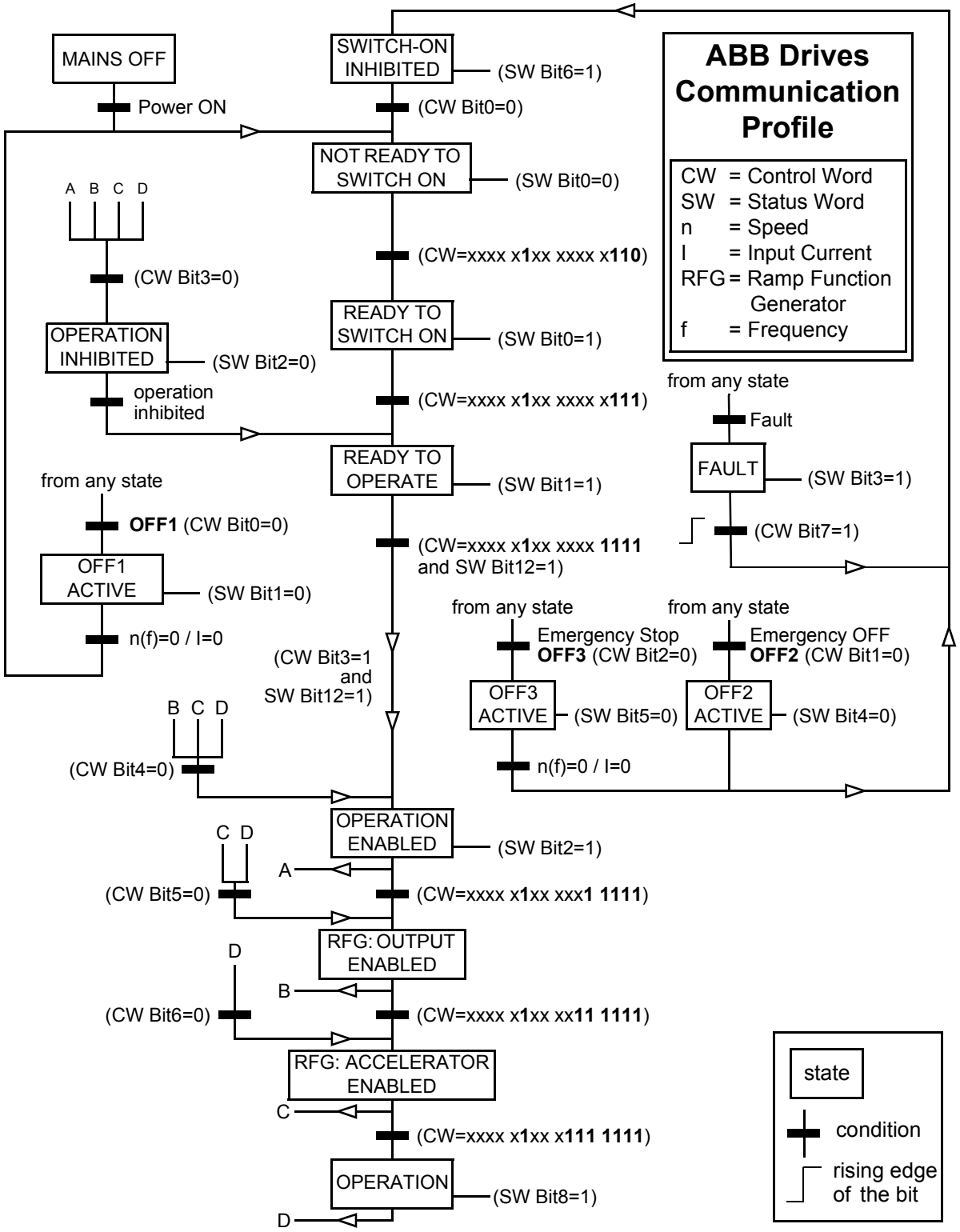


Figure 13. ABB State Transition Diagram

Communication

Overview

This chapter describes the EtherNet/IP communication protocol for the FENA-01. For detailed information on EtherNet/IP communication, refer to the ODVA EtherNet/IP Specifications

Introduction to EtherNet/IP

Ethernet/IP is a protocol based on Common Industrial Protocol (CIP) technology, which is also the framework for both DeviceNet and ControlNet. Ethernet/IP specifies the wiring, and the data transfer through the bus.

The FENA-01 module can act as a server on an EtherNet/IP network.

Object modelling and functional profiles

One of the main features of EtherNet/IP is object modelling. A group of objects can be described with a Functional Profile. The FENA-01 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 the FENA-01, 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 FENA-01 uses Static assemblies (in other words, fixed groupings of different object data only).

The following tables describe the assembly instances supported by the FENA-01.

BASIC SPEED CONTROL assembly

The BASIC SPEED CONTROL assembly is defined by 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 Forward
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						Running Forward		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 For-ward
1								
2	Speed Reference (Low Byte)							
3	Speed Reference (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 170 (ODVA AC/DC Profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0						Running Forward		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)							
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 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		Net Ref	Net Ctrl			Fault Reset	Run Reverse	Run For-ward
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 Refer-ence	Ref From Net	Ctrl From Net	Ready	Run-ning Reverse	Run-ning Forward	Warning	Faulted
1	Drive State (See Table 9.)							
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 Ctl			Fault Reset	Run Reverse	Run Forward
1								
2	Speed Reference (Low Byte)							
3	Speed Reference (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 171 (ODVA AC/DC Profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	At Refer-ence	Ref From Net	Ctrl From Net	Ready	Running Reverse	Running Forward	Warning	Faulted
1	Drive State (See Table 9.)							
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 Reset		Run For-ward
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						Run-ning Forward		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 For-ward
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)							
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						Run-ning Forward		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)							
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 Ctl			Fault Reset	Run Reverse	Run For-ward
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 Refer-ence	Ref From Net	Ctrl From Net	Ready	Running Reverse	Running Forward	Warning	Faulted
1	Drive State (See Table 9.)							
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 Reverse	Run Forward
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)							
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 Refer-ence	Ref From Net	Ctrl From Net	Ready	Running Reverse	Running Forward	Warn-ing	Faulted
1	Drive State (See Table 9.)							
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)							
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 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 Opera-tion	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	Field-bus Error			Ext Run Enable	Ext Ctrl Loc	Above Limit	Remote	At Set-point
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 Opera-tion	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	Field-bus Error			Ext Run Enable	Ext Ctrl Loc	Above Limit	Remote	At Set-point
2	Actual Speed (Low Byte)							
3	Actual Speed (High Byte)							
4	DATA IN 1 Value (Low Byte)							
5	DATA IN 1 Value (Low Byte)							
6	DATA IN 1 Value (High Byte)							
7	DATA IN 2 Value (Low Byte)							
8	DATA IN 2 Value (High Byte)							
9	DATA IN 3 Value (Low Byte)							
10	DATA IN 3 Value (High Byte)							
11	DATA IN 4 Value (Low Byte)							
12	DATA IN 4 Value (High Byte)							
13	DATA IN 5 Value (Low Byte)							
14	DATA IN 5 Value (High Byte)							
15	DATA IN 6 Value (Low Byte)							
16	DATA IN 6 Value (High Byte)							
17	DATA IN 7 Value (Low Byte)							
18	DATA IN 7 Value (High Byte)							
19	DATA IN 8 Value (Low Byte)							
20	DATA IN 8 Value (High Byte)							
21	DATA IN 9 Value (Low Byte)							
22	DATA IN 9 Value (High Byte)							
23	DATA IN 10 Value (Low 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 Opera-tion	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)							

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	Field-bus Error			Ext Run Enable	Ext Ctrl Loc	Above Limit	Remote	At Set-point
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 Opera-tion	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)							
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	Field-bus Error			Ext Run Enable	Ext Ctrl Loc	Above Limit	Remote	At Set-point
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)							
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)							
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)							
3	Drive Profile 16-bit Reference 1 Word (High Byte)							
4	Drive Profile16-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)							
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 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)							
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)							
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)							
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)							
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							
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)							

Instance 122 (Transparent 32 Profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
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)							
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)							
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)							

Instance 172 (Transparent 32 Profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
29	DATA IN 9 Value (High Byte)							
30	DATA IN 10 Value (Low Byte)							
31	DATA IN 10 Value (High Byte)							

Class objects

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 FENA-01 Ethernet 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:
- User's Manual for the drive
- Electronic Data Sheet Files (EDS) for the drive.

Be aware that the units of attributes may differ from those of parameters documented elsewhere and those differences should be considered when interfacing to the drive via the module.

Identity Object, Class 01h

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

Class Attributes

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

Instance Attributes

#	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
8	State	Get	Present state of device.	USINT

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

Drive Type	Profile	Device Type	Value
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.

Revision
Revision attribute, which consists of Major and Minor Revisions, identifies the Revision of the item the Identity Object is representing.

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.
2	Configured	TRUE indicates 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.

Bit(s)	Type/Name	Definition
8	Minor Recoverable Fault	TRUE indicates the device detected a recoverable problem. The problem does not cause the device to go into a faulted state.
9	Minor Unrecoverable Fault	TRUE indicates the device detected a unrecoverable problem. The problem does not cause the device to go into a faulted state.
10	Major Recoverable Fault	TRUE indicates the device detected a problem which caused the device to go into the “Major Recoverable Fault” state.
11	Major Unrecoverable Fault	TRUE indicates the device detected a problem which caused the device to go 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.

State:
Represents current state of Identity Object**.**

Value	State
0	Nonexistent
1	Device Self Testing
2	Standby
3	Operational
4	Major Unrecoverable Fault
5	Minor Unrecoverable Fault

Connection Object, Class 05h

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.

Table 14. 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 Table 14.)	UINT8
2	Instance Type	Get	Indicates either I/O (1) or messaging connection (0).	UINT8
3	Transport Class Trigger	Get	Defines the behaviour 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 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 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

#	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

#	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

Motor Data Object, Class 28h

This 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

#	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

#	Attribute name	Services	Description	Motor type	Data type
3	Motor Type	Get	See table above.	AC	UNIT16
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

#	Attribute name	Services	Description	Motor type	Data type
15	Base Speed	Get, Set	Nominal speed at rated frequency from nameplate Units [RPM]	AC/DC	UINT16

Control Supervisor Object, Class 29h

The object models all the management functions for devices within the ‘Hierarchy of Motor Control Devices’. The behaviour of motor control devices is described by the *AC/DC-Drive Object, Class 2Ah* and the *Run/Stop event matrix*. See Table 9. and Figure 10.

Class Attributes

#	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

#	Attribute name	Services	Description	Data type
3	Run 1	Get, Set	0 = Stop, 1 = Run (See Table 8.)	BOOL
4	Run 2	Get, Set	0 = Stop, 1 = Run (See Table 8.)	BOOL
5	Net Control	Get, Set	0 = Local Control, 1 = Network Control	BOOL
6	State	Get	State of Object. (See Table 9.)	UINT8
7	Running 1	Get	0 = Stopped, 1 = Running	BOOL
8	Running 2	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 Drive Manual for further information on DRIVECOMM codes.	UINT16

#	Attribute name	Services	Description	Data type
14	Warning Code	Get	Code word indicating warning present. If multiple warnings are present, the lowest code value is displayed. DRIVECOMM codes are reported. See 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

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

Class Attributes

#	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

#	Attribute name	Services	Description	Data type
3	At Reference	Get	Frequency arrival	BOOL
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 Table 3.	SINT16
8	SpeedRef	Get, Set	Units = See Table 3.	SINT16
11	Torque Actual	Get	Units = See Table 4.	SINT16
12	TorqueRef	Get, Set	Units = See Table 4.	SINT16
18	AccelTime	Get, Set	Units = milliseconds	UINT16
19	DecelTime	Get, Set	Units = milliseconds	UINT16
22	Speed Scale	Get, Set	Speed scaling factor. See Table 3.	UINT8
24	Torque Scale	Get, Set	Torque scaling factor. See Table 3.	UINT8
29	Ref From Net	Get	Reflecting attribute 4	BOOL

Drive Parameter Object, Class 90h

With the FENA-01, 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 using the Drive Parameter Object to update the fieldbus configuration groups take effect only 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)
- *Attribute* = Parameter Index (01…99)

For example, Parameter 99.01 is accessed as follows:

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

Fieldbus Configuration Object 91h

The Fieldbus Configuration Object allows the user to configure the field bus configuration groups without needing to know the drive specific groups associated with the configuration groups.

Note: When using 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: FENA-01 Configuration Parameters Group #1 *

#	Attribute name	Services	Description	Data type
1	Configuration Group #1 - Parameter 1	Get, Set	See chapter *Drive configuration*	UINT16
2	Configuration Group #1 - Parameter 2	Get, Set	See chapter *Drive configuration*	UINT16
3	Configuration Group #1 - Parameter 3	Get, Set	See chapter *Drive configuration*	UINT16
4	Configuration Group #1 - Parameter 4	Get, Set	See chapter *Drive configuration*	UINT16
5	Configuration Group #1 - Parameter 5	Get, Set	See chapter *Drive configuration*	UINT16
6	Configuration Group #1 - Parameter 6	Get, Set	See chapter *Drive configuration*	UINT16
7	Configuration Group #1 - Parameter 7	Get, Set	See chapter *Drive configuration*	UINT16
8	Configuration Group #1 - Parameter 8	Get, Set	See chapter *Drive configuration*	UINT16
9	Configuration Group #1 - Parameter 9	Get, Set	See chapter *Drive configuration*	UINT16
10	Configuration Group #1 - Parameter 10	Get, Set	See chapter *Drive configuration*	UINT16
11	Configuration Group #1 - Parameter 11	Get, Set	See chapter *Drive configuration*	UINT16
12	Configuration Group #1 - Parameter 12	Get, Set	See chapter *Drive configuration*	UINT16
13	Configuration Group #1 - Parameter 13	Get, Set	See chapter *Drive configuration*	UINT16
14	Configuration Group #1 - Parameter 14	Get, Set	See chapter *Drive configuration*	UINT16
15	Configuration Group #1 - Parameter 15	Get, Set	See chapter *Drive configuration*	UINT16
16	Configuration Group #1 - Parameter 16	Get, Set	See chapter *Drive configuration*	UINT16

#	Attribute name	Services	Description	Data type
17	Configuration Group #1 - Parameter 17	Get, Set	See chapter *Drive configuration*	UINT16
18	Configuration Group #1 - Parameter 18	Get, Set	See chapter *Drive configuration*	UINT16
19	Configuration Group #1 - Parameter 19	Get, Set	See chapter *Drive configuration*	UINT16
20	Configuration Group #1 - Parameter 20	Get, Set	See chapter *Drive configuration*	UINT16
21	Configuration Group #1 - Parameter 21	Get, Set	See chapter *Drive configuration*	UINT16
22	Configuration Group #1 - Parameter 22	Get, Set	See chapter *Drive configuration*	UINT16
23	Configuration Group #1 - Parameter 23	Get, Set	See chapter *Drive configuration*	UINT16
24	Configuration Group #1 - Parameter 24	Get, Set	See chapter *Drive configuration*	UINT16
25	Configuration Group #1 - Parameter 25	Get, Set	See chapter *Drive configuration*	UINT16
26	Configuration Group #1 - Parameter 26	Get, Set	See chapter *Drive configuration*	UINT16
27	Configuration Group #1 - Parameter 27	Get, Set	See chapter *Drive configuration*	UINT16

* Eg, in ACS350 and ACSM1 parameter Group 51.

*Instance #2: FENA-01 Configuration Parameters Group #2 **

#	Attribute name	Services	Description	Data type
1	Configuration Group #2 - Parameter 1	Get, Set	See chapter *Drive configuration*	UINT16
2	Configuration Group #2 - Parameter 2	Get, Set	See chapter *Drive configuration*	UINT16
3	Configuration Group #2 - Parameter 3	Get, Set	See chapter *Drive configuration*	UINT16
4	Configuration Group #2 - Parameter 4	Get, Set	See chapter *Drive configuration*	UINT16
5	Configuration Group #2 - Parameter 5	Get, Set	See chapter *Drive configuration*	UINT16
6	Configuration Group #2 - Parameter 6	Get, Set	See chapter *Drive configuration*	UINT16
7	Configuration Group #2 - Parameter 7	Get, Set	See chapter *Drive configuration*	UINT16
8	Configuration Group #2 - Parameter 8	Get, Set	See chapter *Drive configuration*	UINT16
9	Configuration Group #2 - Parameter 9	Get, Set	See chapter *Drive configuration*	UINT16
10	Configuration Group #2 - Parameter 10	Get, Set	See chapter *Drive configuration*	UINT16

Eg, in parameter Group 55 in ACS350 or group 53 in ACSM1.

*Instance #3: FENA-01 Configuration Parameters Group #3 **

#	Attribute name	Services	Description	Data type
1	Configuration Group #3 - Parameter 1	Get, Set	See chapter *Drive configuration*	UINT16
2	Configuration Group #3 - Parameter 2	Get, Set	See chapter *Drive configuration*	UINT16
3	Configuration Group #3 - Parameter 3	Get, Set	See chapter *Drive configuration*	UINT16
4	Configuration Group #3 - Parameter 4	Get, Set	See chapter *Drive configuration*	UINT16
5	Configuration Group #3 - Parameter 5	Get, Set	See chapter *Drive configuration*	UINT16
6	Configuration Group #3 - Parameter 6	Get, Set	See chapter *Drive configuration*	UINT16
7	Configuration Group #3 - Parameter 7	Get, Set	See chapter *Drive configuration*	UINT16
8	Configuration Group #3 - Parameter 8	Get, Set	See chapter *Drive configuration*	UINT16
9	Configuration Group #3 - Parameter 9	Get, Set	See chapter *Drive configuration*	UINT16
10	Configuration Group #3 - Parameter 10	Get, Set	See chapter *Drive configuration*	UINT16

* Eg, parameter Group 54 in ACS350 or group 53 in ACSM1.

TCP/IP Interface Object, Class F6h

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

Class Attributes

#	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

#	Attribute name	Services	Description	Data type
3	Configuration Control	Get, Set	0 = Configuration from FENA-01 configuration parameters; 2 = Configuration from DHCP	DWORD
5	Interface Configuration	Get, Set		ARRAY
	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

Ethernet Link Object, Class F7h

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

Class Attributes

#	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

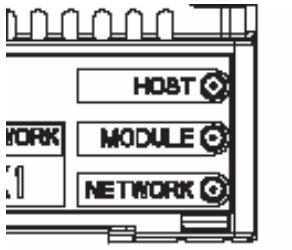
Instance Attributes

#	Attribute name	Services	Description	Data type
1	Interface Speed	Get, Set	10 or 100 Mbps	UDINT
2	Interface Flags	Get, Set	-	DWORD
3	Physical Address	Get	Ethernet MAC address of the module	ARRAY 6XUSINT

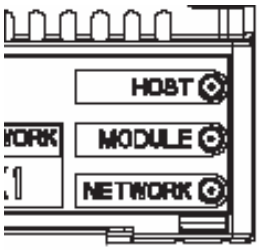
Diagnostics

LED indications

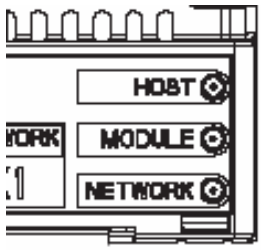
The FENA-01 module is equipped with three bicolor diagnostic LEDs. The LEDs are described below.



Name	Color	Function/State
HOST	Blinking green	Establishing communication to host.
	Green	Connection to host OK.
	Blinking red	Communication to host lost temporarily.



Name	Color	Function/State
MODULE	Off	There is no power applied to the device.
	Green	The device is operating in a normal condition.
	Flashing green	The 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 for Duplicate Address Detection to complete.
	Flashing red	Recoverable fault.
	Red	The device has an unrecoverable fault. This may be cleared by: a Fieldbus Adapter parameter refresh or cycling drive power. This may have been caused by the device detecting another device on the network with the same MAC ID or IP Address.
	Flashing red-green	The device is in Self Test.



Name	Color	Function/State
NETWORK	Off	Device is not on-line. – The device has not completed the Duplicate Address Detection yet. – The device may not be powered, look at 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	The 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).

Definitions and abbreviations

Communication Module

Communication Module is a name for a device (eg, a fieldbus adapter) through which the drive is connected to an external serial communication network (eg, a fieldbus). The communication with the communication module is activated by a drive parameter.

EDS File

The Electronic Data Sheet (EDS) file identifies the properties of the device to the EtherNet/IP Client. Each type of drive and application program requires its own EDS file.

Input

In the ODVA EtherNet/IP specification the word ‘input’ is used to describe data flow from a device (such as the FENA-01) to the network.

I/O Assembly selection

Smart networked devices (like the FENA-01) can produce and/or consume more than one I/O value. Typically, they will produce and/or consume one or more I/O value, as well as status and diagnostic information. Each piece of data communicated by a device is represented by an attribute of one of the device’s internal objects.

Communicating multiple pieces of data (attributes) across a single I/O connection requires that the attributes be grouped or assembled together into a single block.

MAC ID

Every node on an Ethernet network has to have a unique identifier. This node number is called MAC ID (Media Access Control ID).

FENA-01 Ethernet Adapter module

The FENA-01 Adapter module is one of the optional fieldbus adapter modules available for ABB drives. The FENA-01 is a

device through which an ABB drive is connected to an Ethernet network.

ODVA

ODVA stands for Open DeviceNet Vendor Association. ODVA is an independent organization that promotes interoperativity between different manufacturers EtherNet/IP products. ABB is an Associate Member at the ODVA.

Output

In the ODVA EtherNet/IP specification the word ‘output’ is used to describe data flow from the network into a device (such as the FENA-01).

Parameter

A parameter is an operating instruction for the drive. Parameters can be read and programmed using the drive control panel, or through the FENA-01 module.

3AUA0000033371 REV A EN
EFFECTIVE: 31.3.2008

ABB Oy
AC Drives
P.O. Box 184
FI-00381 HELSINKI
FINLAND
Telephone +358 10 22 11
Fax +358 10 22 22681
Internet http://www.abb.com

ABB Inc.
Automation Technologies
Drives & Motors
16250 West Glendale Drive
New Berlin, WI 53151
USA
Telephone 262 785-3200
800-HELP-365
Fax 262 780-5135