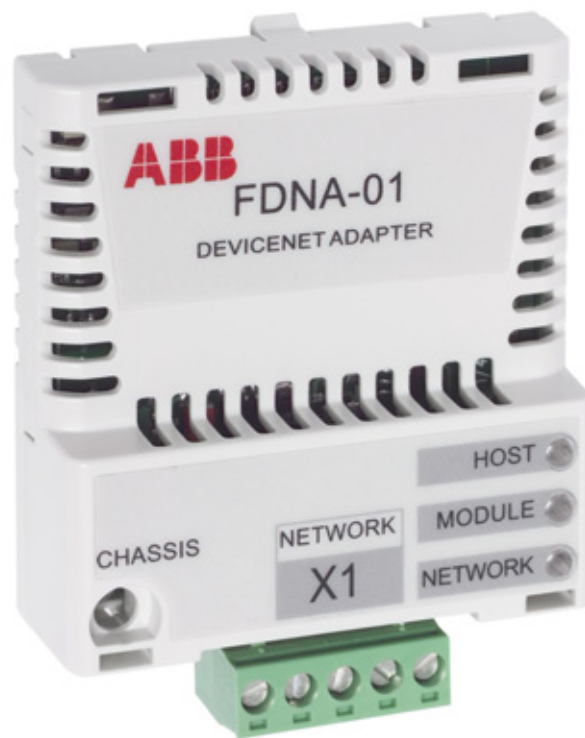


ABB Drives

User's Manual DeviceNet Adapter Module FDNA-01



DeviceNet Adapter Module FDNA-01

User's Manual

3AFE68573360 Rev D
EN
EFFECTIVE: 2009-12-01

Safety instructions

What this chapter contains

This chapter states the general safety instructions that must be followed when installing and operating the FDNA-01 DeviceNet 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

Safety instructions

What this chapter contains	5
General safety instructions	5

Table of contents

Introduction

What this chapter contains	11
Intended audience	11
Before you start	11
What this manual contains	11

Overview

What this chapter contains	13
DeviceNet standard	13
FDNA-01 DeviceNet Adapter module	14
Compatibility	15
Delivery check	16

Quick start-up guide

What this chapter contains	17
PLC configuration	17
Mechanical and electrical installation	22
Drive configuration	23
Example configuration: ODVA Basic Speed Control in ACS350 and ACS355	24
Example configuration: ABB Drives Profile w/ Set Speed and Set Torque plus Drive Parameters in ACS350 and ACS355	25
Example configuration: ODVA Extended Speed and Torque Control in ACSM1	27
Example configuration: ABB Drives profile with Speed control in ACSM1	28
Example configuration: ODVA Extended Speed and Torque Control in ACS850	29
Example configuration: ABB Drives profile with Speed control in ACS850 and ACQ810	30

Mechanical installation

What this chapter contains	31
Mounting	31

Electrical installation

What this chapter contains	33
----------------------------------	----

General cabling instructions	33
DeviceNet connection	34
DeviceNet bus termination	35

Drive configuration

What this chapter contains	37
DeviceNet connection configuration	37
FDNA-01 configuration parameters - group A (group 1)	38
FDNA-01 configuration parameters - group B (group 2)	46
FDNA-01 configuration parameters - group C (group 3)	48
Control locations	49

Master configuration

What this chapter contains	51
Configuring the system	51
EDS files	51

Communication profiles

What this chapter contains	53
Communication profiles	53
The ODVA AC/DC Drive Profiles	55
ODVA output attributes	56
ODVA input attributes	59
The ABB Drives communication profile	64
Control word and Status word	64
References	64
Actual values	66

Communication

What this chapter contains	73
Introduction to DeviceNet	73
Object modelling and functional profiles	73
Assembly objects	74
BASIC SPEED CONTROL assembly	75
BASIC SPEED CONTROL PLUS DRIVE PARAMETERS assembly	76
EXTENDED SPEED CONTROL assembly	78
EXTENDED SPEED CONTROL PLUS DRIVE PARAMETERS assembly	79
BASIC SPEED AND TORQUE CONTROL assembly	81
BASIC SPEED AND TORQUE CONTROL PLUS DRIVE PARAMETERS assembly	82
EXTENDED SPEED AND TORQUE CONTROL assembly	84
EXTENDED SPEED AND TORQUE CONTROL PLUS DRIVE PARAMETERS assembly	85
ABB DRIVES PROFILE SET SPEED assembly	87
ABB DRIVES PROFILE WITH SET SPEED PLUS DRIVE PARAMETERS assembly	88
ABB DRIVES PROFILE WITH SET SPEED AND SET TORQUE assembly	90
ABB DRIVES PROFILE WITH SET SPEED AND SET TORQUE PLUS DRIVE PARAMETERS	

assembly	91
TRANSPARENT16 WITH ONE assembly	94
TRANSPARENT16 WITH ONE assembly PLUS DRIVE PARAMETERS	95
TRANSPARENT16 WITH TWO assembly	97
TRANSPARENT16 WITH TWO assembly PLUS DRIVE PARAMETERS	97
TRANSPARENT32 WITH ONE assembly	100
TRANSPARENT32 WITH ONE assembly PLUS DRIVE PARAMETERS	100
TRANSPARENT32 WITH TWO assembly	103
TRANSPARENT32 WITH TWO assembly PLUS DRIVE PARAMETERS	104
Class objects	107
Identity object, class 01h	108
DeviceNet object, class 03h	111
Connection object, class 05h	112
Acknowledge handler object, class 2Bh	114
Motor data object, class 28h	115
Control supervisor object, class 29h	117
AC/DC-drive object, class 2Ah	119
Drive parameter object, class 90h	120
Fieldbus configuration object 91h	121

Diagnostics

What this chapter contains	127
LED indications	127

Definitions and abbreviations

What this chapter contains	131
----------------------------	-----

Technical data

What this chapter contains	135
FDNA-01	135
Fieldbus link	136

Appendix - Varying number of drive parameters

Further information

Product and service inquiries	141
Product training	141
Providing feedback on ABB Drives manuals	141
Document library on the Internet	141

Introduction

What this chapter contains

This chapter includes a description of the contents of the manual.

Intended audience

The manual is intended for people responsible for installing, commissioning and using an adapter module. The reader is expected to have 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 wiring, configuration and use of the adapter module.

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

Overview contains short descriptions of the DeviceNet protocol and the adapter module, and a delivery checklist.

Quick start-up guide contains a short description of how to set up the adapter module.

Mechanical installation contains placing and mounting instructions for the module.

Electrical installation contains wiring and bus termination instructions.

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

Master configuration explains how to program the DeviceNet master before communication through the adapter module can be started.

Communication profiles describes the communication profiles used in communication between the DeviceNet network, the FDNA-01 module and the drive.

Communication contains a description of the DeviceNet functionality supported by the FDNA-01.

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

Definitions and abbreviations explains definitions and abbreviations concerning the DeviceNet protocol family.

Technical data contains information on physical dimensions, configurable settings and connectors of the module and a specification of the DeviceNet link.

Overview

What this chapter contains

This chapter contains a short description of the DeviceNet standard and the FDNA-01 Adapter module, and a delivery checklist.

DeviceNet standard

The DeviceNet network has a linear bus topology. Terminating resistors are required on each end of the trunk line. Drop lines as long as 6 metres (20 feet) each are permitted, allowing one or more nodes to be attached. DeviceNet allows branching structures only on drop lines. An example of an allowable topology is shown in [Figure 1](#).

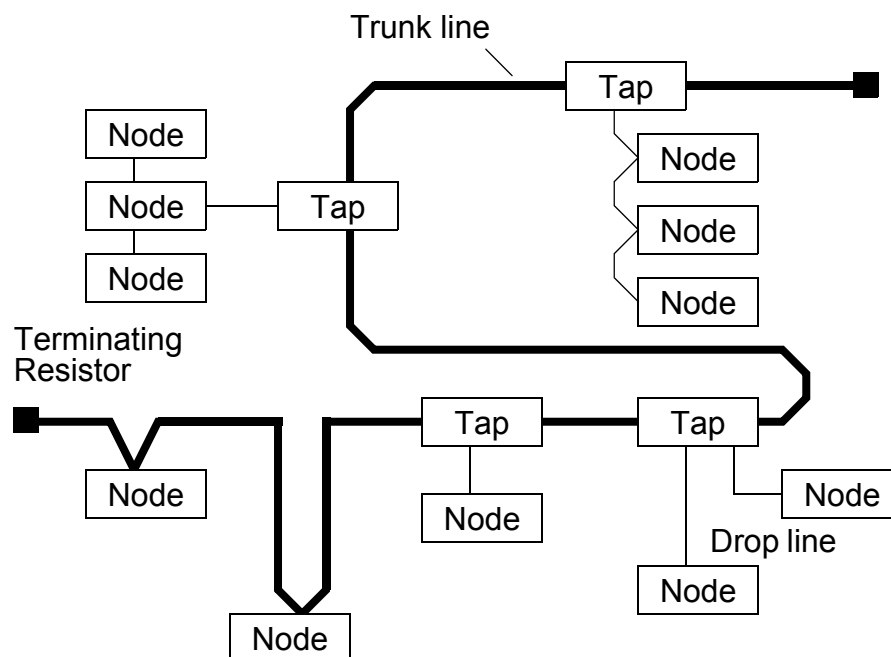


Figure 1. DeviceNet bus topology

The maximum length of the trunk cable depends on the data rate and on the type of the cable used (see chapter [Technical data](#) on page [135](#)).

Further information on the DeviceNet protocol is available on the world wide web from www.odva.org

Terminating resistor

The DeviceNet network should be terminated at both ends of the trunk cable with a 121 Ohm, ¼ W, 1% metal film resistor. Connect the resistor between the two signal wires (CAN_H, CAN_L) on the DeviceNet cable.

FDNA-01 DeviceNet Adapter module

The FDNA-01 DeviceNet Adapter module is an optional device for ABB drives. It enables the connection of the drive to a DeviceNet network. The drive is considered as a slave in the DeviceNet network. With the adapter module you can:

- give control commands to the drive
(for example, Start, Stop, Run enable)
- feed a motor Speed or Torque Reference to the drive
- give the Actual value or Reference of the process to the PID controller of the drive
- read status information and Actual values from the drive
- change the drive parameter values
- reset a drive fault.

The FDNA-01 acts as a class 2 slave only with predefined master-slave connection set services. These include the explicit messaging, the poll-response service and the change of state/ cyclic service. Chapter [Communication](#) describes the DeviceNet commands and services supported by the adapter module.

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

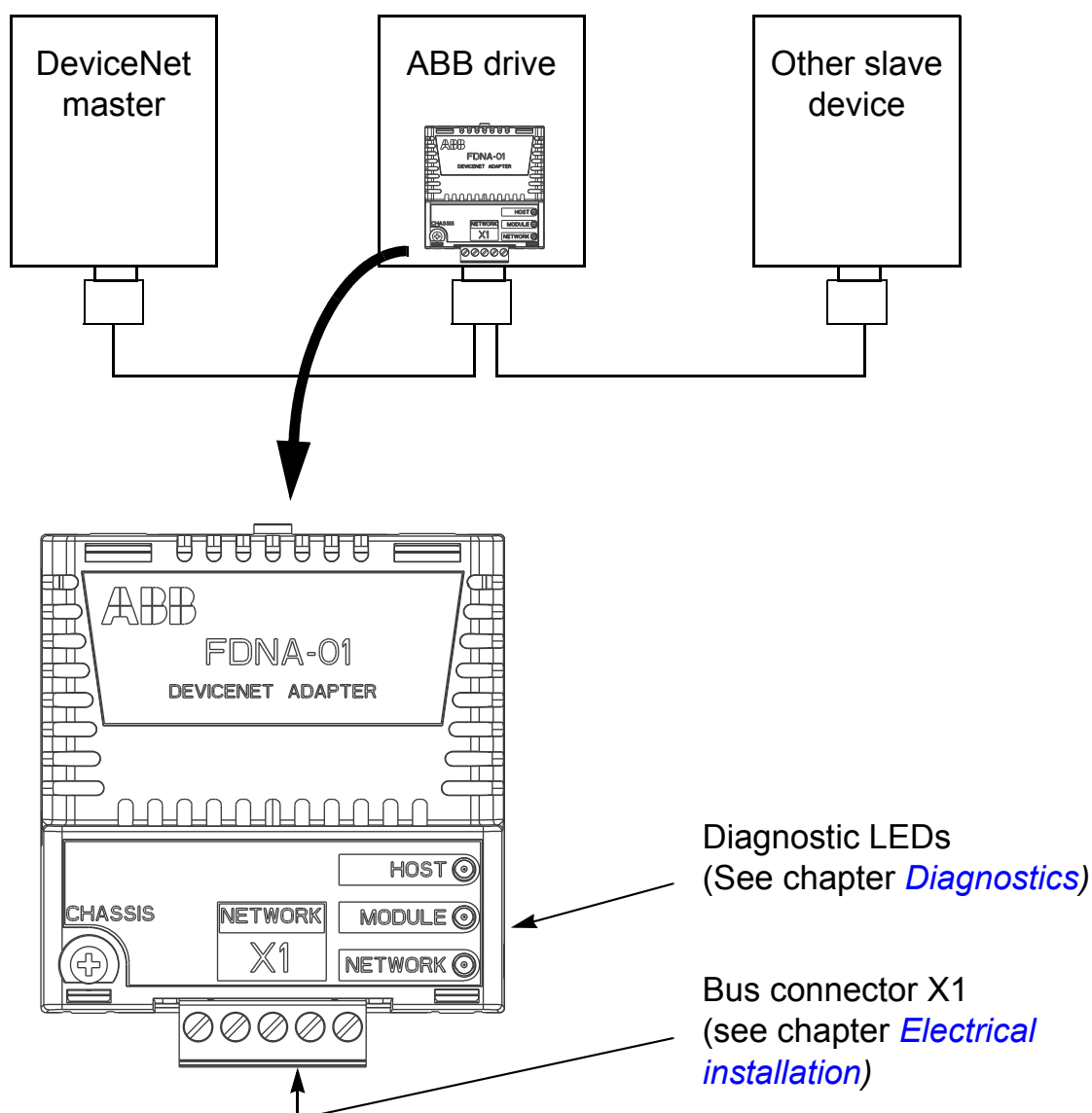


Figure 2. The construction of the DeviceNet link and the FDNA-01 Adapter module.

Compatibility

The FDNA-01 is compatible with all master stations that support the DeviceNet protocol.

Delivery check

The option package of the adapter module contains:

- DeviceNet Adapter module, type FDNA-01
- this manual.

Quick start-up guide

What this chapter contains

This chapter presents the steps to take during the start-up of the adapter module.



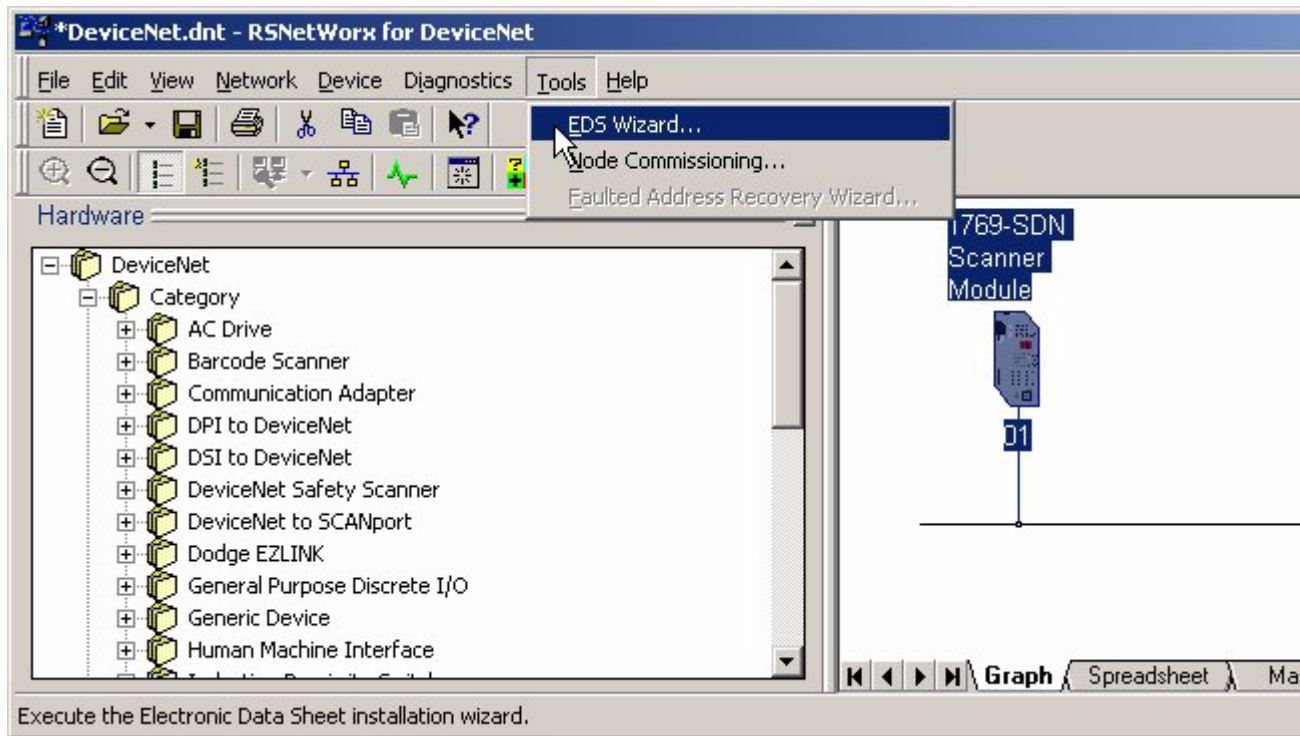
WARNING! Follow the safety instructions given in this manual and the hardware manual of the drive.

PLC configuration

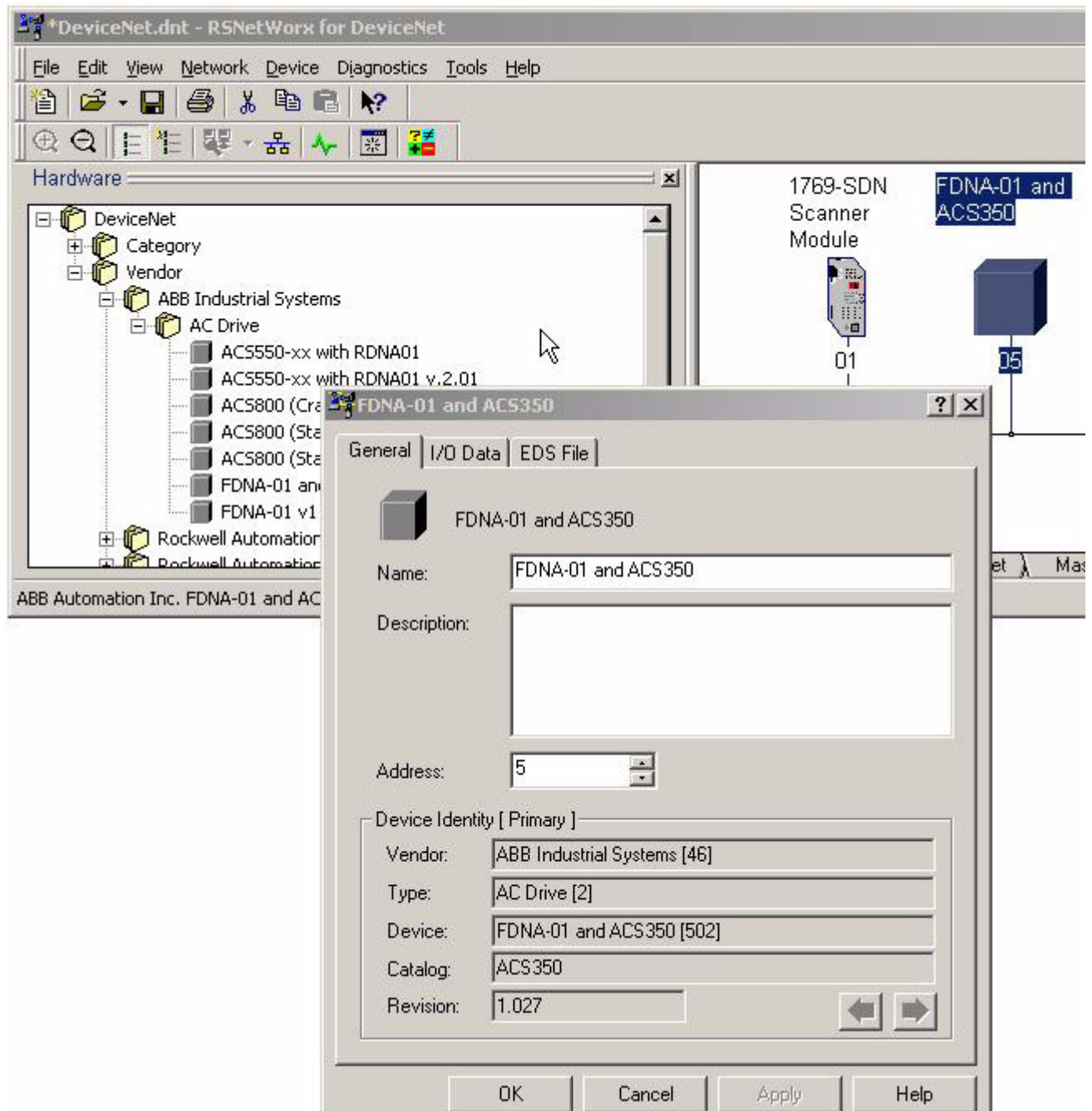
This example shows how to set up an Allen-Bradley CompactLogix PLC with a 1769-SDN DeviceNet Scanner to use a drive equipped with a FDNA-01 fieldbus adapter, using RSNetWorx and RSLogix 5000 to configure and control the network. Slight vendor specific differences may exist when using other programs or master devices. In this example, ACS350 is used as the drive, but the same example can be applied to ACS355, too.

- Select and import the EDS file for the drive, for example ACS350_FDNA01_v1.27_eu.EDS. For more help on choosing the correct file, refer to *ABB EDS Selection Guide* delivered with the EDS files.

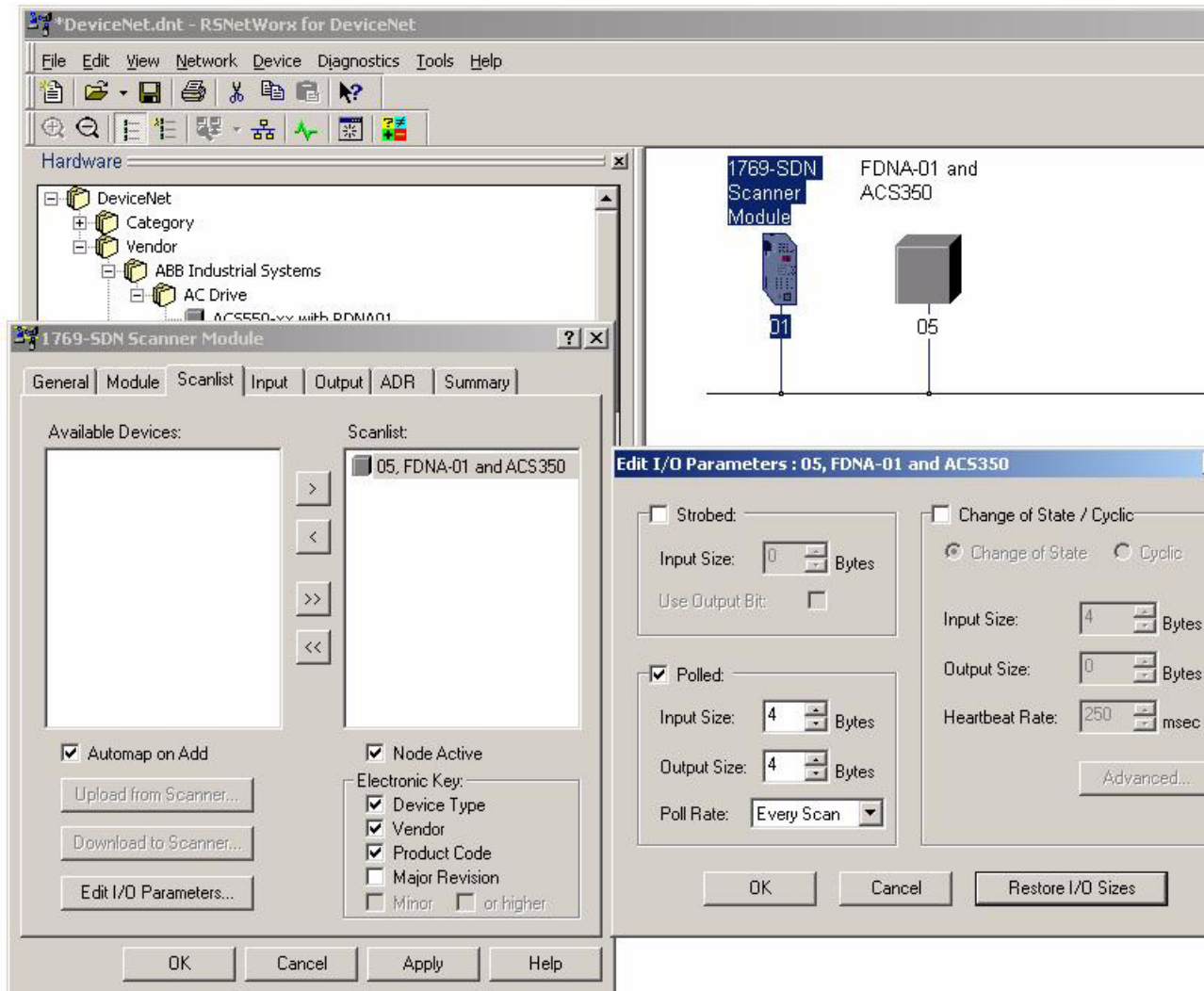
Note: Only one EDS file with the same Product ID can be installed in the PLC at a time.



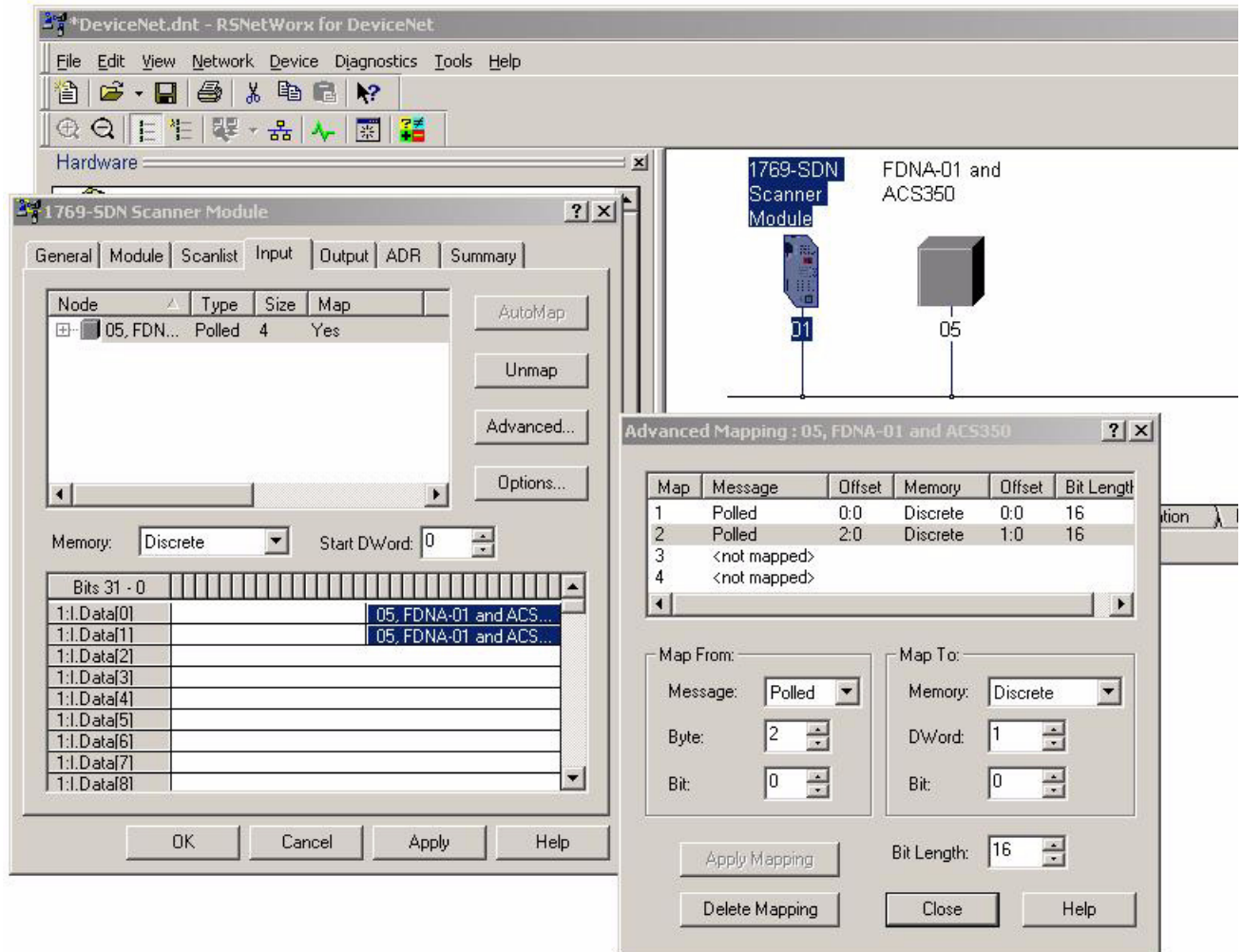
- Add a drive to the network in RSNetWorx for DeviceNet.
- Configure the device MAC address.



- Configure the DeviceNet Scanner. Add the drive to Scanlist and edit I/O parameters. Select Polled or Change of State/Cyclic. Then enter the input and output sizes of the I/O assemblies that are used (eg for Basic Speed Control Assemblies 20 and 70, each size is set to 4 bytes.).



- For convenience, you can also edit the Input and Output memory mappings. By default, the two 16-bit input words are mapped into a single 32-bit double word. It is more convenient to map them into separate double words. Output words can be mapped similarly.



- Download the scanner settings to the device in the online mode.

- Add your DeviceNet scanner to an RSLogix 5000 project. You should get new Controller Tags Local:<slot>:I and Local:<slot>:O. You can use them to access the data as mapped above:

Local:<slot>:O.Data[0] is the Control word

Local:<slot>:O.Data[1] is the Reference

Local:<slot>:I.Data[0] is the Status word

Local:<slot>:I.Data[1] is the Actual value

Note: You may need to change the value of

Local:<slot>:O.CommandRegister.Run or

Local:<slot>:I.CommandRegister.Run to 1.

Mechanical and electrical installation

- Insert the FDNA-01 into its specified slot in the drive.
- Fasten the screw.
- Plug the fieldbus connector into the module.

Note: It is recommended to make the module communication settings (for example, the baud rate) before connecting it to an active DeviceNet network (see [Drive configuration](#) on page 23).

Drive configuration

- Power up the drive.
- The detailed procedure of activating the drive for communication with the module is dependent on the drive type. Normally, a parameter must be adjusted to activate the communication. Refer to the following table and drive documentation for information on the communication settings.

Drive	Parameter	Setting
ACS350 ACS355	98.02 COMM PROT SEL	4 = EXT FBA
ACSM1 ACS850 ACQ810	50.01 FBA ENABLE	ENABLE

- In ACSM1, ACS850 and ACQ810, ensure that 50.04 FBA REF1 MODESEL is SPEED. In ACSM1 and ACS850, ensure that 50.05 FBA REF2 MODESEL is TORQUE.
- Verify that FBA TYPE is **37** = DEVICENET. In ACS350, ACS355, ACSM1, ACS850 and ACQ810, this is parameter 51.01.
- Configure the drive to accept Start/Stop, Direction, Reference, Run Enable and Fault Reset from the module. Examples of appropriate values are shown in the following tables.
- Configure the module for desired network characteristics with the fieldbus configuration groups. Examples of suggested values are shown in the following tables.
- Initiate a "Fieldbus adapter parameter refresh" by setting parameter 51.27 FBA PAR REFRESH.

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).

Example configuration: ODVA Basic Speed Control in ACS350 and ACS355

Drive parameter	Example setting for ACS350 and ACS355
10.01 EXT1 COMMANDS	COMM
11.03 REF1 SELECT	COMM
16.01 RUN ENABLE	NOT SEL*
16.04 FAULT RESET SEL	COMM
98.02 COMM PROT SEL	EXT FBA
51.01 FBA TYPE	DEVICENET
51.02 MAC ID	2
51.03 BAUD RATE	0 (125 kbit/s)
51.05 ODVA STOP FUNC	0 (Ramp Stop)
51.06 OUTPUT INSTANCE	20 (ODVA Basic Speed Control)
51.07 OUTPUT NUM PARS	10**
51.08 INPUT INSTANCE	70 (ODVA Basic Speed Control)
51.09 INPUT NUM PARS	10**
51.10 ODVA SPEED SCALE	128 (1 RPM)
51.11 ODVA TORQUE SCALE	128 (1 N·m)
51.27 FBA PAR REFRESH	REFRESH

* The ODVA AC/DC Drive Profile does not provide a means to supply a RUN ENABLE signal via DeviceNet. The RUN ENABLE signal for the drive must be set to either NOT SEL or another source (for example, a digital input).

** OUTPUT NUM PARS and INPUT NUM PARS must always be set to 10, except as provided for in the [Appendix - Varying number of drive parameters](#) on page 137.

Scanner: The input and output sizes are 4 bytes each.

Example configuration: ABB Drives Profile w/ Set Speed and Set Torque plus Drive Parameters in ACS350 and ACS355

In this example:

- The output drive parameters control analog output 1 and relay output 1.
- The input drive parameters read analog input 1 (01.20), CURRENT (01.04) and POWER (01.06).

Drive parameter	Example setting for ACS350 and ACS355
10.01 EXT1 COMMANDS	COMM
11.03 REF1 SELECT	COMM
11.06 REF2 SELECT	COMM
14.01 RELAY OUTPUT 1	COMM
15.01 AO1 CONTENT SEL	COMM VALUE 1
16.01 RUN ENABLE	COMM [*]
16.04 FAULT RESET SEL	COMM
98.02 COMM PROT SEL	EXT FBA
51.01 FBA TYPE	DEVICENET
51.02 MAC ID	53
51.03 BAUD RATE	2 (500 kbit/s)
51.05 ODVA STOP FUNC	0 (Ramp Stop) ^{**}
51.06 OUTPUT INSTANCE	902 (ABB Drives Profile w/ Speed and Torque plus Drive Parameters)
51.07 OUTPUT NUM PARS	10 ^{***}
51.08 INPUT INSTANCE	952 (ABB Drives Profile w/ Speed and Torque plus Drive Parameters)
51.09 INPUT NUM PARS	10 ^{***}
51.10 ODVA SPEED SCALE	128 (1 RPM) ^{**}
51.11 ODVA TORQUE SCALE	128 (1 N·m) ^{**}

54.01 FBA DATA IN 1	120 (01.20 Analog Input 1)
54.02 FBA DATA IN 2	104 (01.04 Current)
54.03 FBA DATA IN 3	106 (01.06. Power)
54.04...54.10 FBA DATA IN 4...10	0 (Unused)
55.01 FBA DATA OUT 1	135 (01.35 COMM VALUE 1)
55.02 FBA DATA OUT 2	134 (01.34 COMM RO WORD)
55.03...55.10 FBA DATA OUT 3...10	0 (Unused)
51.27 FBA PAR REFRESH	REFRESH

* The ABB Drives Profile does provide a means to supply a RUN ENABLE signal via DeviceNet. The RUN ENABLE for the drive may be set to COMM or another source (for example, a digital input).

** These parameters are ignored when using the ABB Drives profile.

*** OUTPUT NUM PARS and INPUT NUM PARS must always be set to 10, except as provided for in [Appendix - Varying number of drive parameters](#) on page 137.

Scanner: The input and output sizes are 26 bytes each.

Example configuration: ODVA Extended Speed and Torque Control in ACSM1

Drive parameter	Example setting for ACSM1
10.01 EXT1 START FUNC	FBA
24.01 SPEED REF1 SEL	FBA REF1
32.01 TORQ REF1 SEL	FBA REF2
34.03 EXT1 CTRL MODE1	SPEED
34.04 EXT1 CTRL MODE2	TORQUE
50.01 FBA ENABLE	ENABLE
50.04 FBA REF1 MODESEL	SPEED
50.05 FBA REF2 MODESEL	TORQUE
51.01 FBA TYPE	DEVICENET
51.02 MAC ID	10
51.03 BAUD RATE	0 (125 kbit/s)
51.05 ODVA STOP FUNC	0 (Ramp Stop)
51.06 OUTPUT INSTANCE	23 (ODVA Extended Speed and Torque Control)
51.07 OUTPUT NUM PARS	10*
51.08 INPUT INSTANCE	73 (ODVA Extended Speed and Torque Control)
51.09 INPUT NUM PARS	10*
51.10 ODVA SPEED SCALE	128 (1 RPM)
51.11 ODVA TORQUE SCALE	128 (1 Nm)
51.27 FBA PAR REFRESH	REFRESH

* OUTPUT NUM PARS and INPUT NUM PARS must always be set to 10, except as provided for in the [Appendix - Varying number of drive parameters](#) on page 137.

Scanner: The input and output sizes are 6 bytes each.

Example configuration: ABB Drives profile with Speed control in ACSM1

Drive parameter	Example setting for ACSM1
10.01 EXT1 START FUNC	FBA
24.01 SPEED REF1	FBA REF1
34.03 EXT1 CTRL MODE1	SPEED
50.01 FBA ENABLE	ENABLE
50.04 FBA REF1 MODESEL	SPEED
50.05 FBA REF2 MODESEL	TORQUE
51.01 FBA TYPE	DEVICENET
51.02 MAC ID	10
51.03 BAUD RATE	1 (250 kbit/s)
51.05 ODVA STOP FUNC	0 (Ramp Stop)
51.06 OUTPUT INSTANCE	801 (ABB Drives with Set speed)
51.07 OUTPUT NUM PARS	10 [*]
51.08 INPUT INSTANCE	851 (ABB Drives with Set speed)
51.09 INPUT NUM PARS	10 [*]
51.10 ODVA SPEED SCALE	128 (1 RPM)
51.11 ODVA TORQUE SCALE	128 (1 Nm)
51.27 FBA PAR REFRESH	REFRESH

* OUTPUT NUM PARS and INPUT NUM PARS should always be set to 10, except as provided for in [Appendix - Varying number of drive parameters](#) on page 137.

Example configuration: ODVA Extended Speed and Torque Control in ACS850

Drive parameter	Example setting for ACS850
16.15 MENU SET SEL	LOAD LONG
10.01 EXT1 START FUNC	FBA
21.01 SPEED REF 1 SEL	FBA REF1
24.01 TORQ REF1 SEL	FBA REF2
12.03 EXT1 CTRL MODE	SPEED
12.05 EXT2 CTRL MODE	TORQUE
50.01 FBA ENABLE	ENABLE
50.04 FBA REF1 MODESEL	SPEED
50.05 FBA REF2 MODESEL	TORQUE
51.01 FBA TYPE	DEVICENET
51.02 MAC ID	10
51.03 BAUD RATE	0 (125 kbit/s)
51.05 ODVA STOP FUNC	0 (Ramp Stop)
51.06 OUTPUT INSTANCE	23 (ODVA Extended Speed and Torque Control)
51.07 OUTPUT NUM PARS	10*
51.08 INPUT INSTANCE	73 (ODVA Extended Speed and Torque Control)
51.09 INPUT NUM PARS	10*
51.10 ODVA SPEED SCALE	128 (1 RPM)
51.11 ODVA TORQUE SCALE	128 (1 Nm)
51.27 FBA PAR REFRESH	REFRESH

* OUTPUT NUM PARS and INPUT NUM PARS must always be set to 10, except as provided for in [Appendix - Varying number of drive parameters](#) on page 137.

Example configuration: ABB Drives profile with Speed control in ACS850 and ACQ810

Drive parameter	Example setting for ACS850 and ACQ810
16.15 MENU SET SEL (ACS850) 16.21 MENU SELECTION (ACQ810)	LOAD LONG (ACS850) LONG (ACQ810)
10.01 EXT1 START FUNC	FBA
21.01 SPEED REF 1 SEL (ACS850) 21.01 SPEED REF SEL (ACQ810)	FBA REF1
12.03 EXT1 CTRL MODE	SPEED (Only in ACS850)
50.01 FBA ENABLE	ENABLE
50.04 FBA REF1 MODESEL	SPEED
51.01 FBA TYPE	DEVICENET
51.02 MAC ID	10
51.03 BAUD RATE	1 (250 kbit/s)
51.05 ODVA STOP FUNC	0 (Ramp Stop)
51.06 OUTPUT INSTANCE	801 (ABB Drives with Set speed)
51.07 OUTPUT NUM PARS	10 [*]
51.08 INPUT INSTANCE	851 (ABB Drives with Set speed)
51.09 INPUT NUM PARS	10 [*]
51.10 ODVA SPEED SCALE	128 (1 RPM)
51.11 ODVA TORQUE SCALE	128 (1 Nm)
51.27 FBA PAR REFRESH	REFRESH

* OUTPUT NUM PARS and INPUT NUM PARS must always be set to 10, except as provided for in [Appendix - Varying number of drive parameters](#) on page 137.

Mechanical installation

What this chapter contains

This chapter contains placing and mounting instructions for the module.



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

Mounting

The FDNA-01 is to be inserted into its specific position in the drive. The module is held in place with plastic pins and one screw. The screw also provides the earthing of the fieldbus cable shield connected to the module.

On installation of the module, the signal and power connection to the drive is made through a 20-pin connector. (All drives do not use all the available signals so the connector on the drive may have fewer pins.)

Mounting procedure:

- Insert the module carefully into its position on the drive.
- Fasten the screw.

Note: Correct installation of the screw is essential for fulfilling the EMC requirements and for proper operation of the module.

Electrical installation

What this chapter contains

This chapter contains:

- general cabling instructions
- instructions for connecting the module to the DeviceNet bus.



WARNING! Before installation, switch off the drive power supply. Wait 5 minutes to ensure that the capacitor bank of the drive is discharged. Switch off all dangerous voltages connected from external control circuits to the inputs and outputs of the drive.

General cabling instructions

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

DeviceNet connection

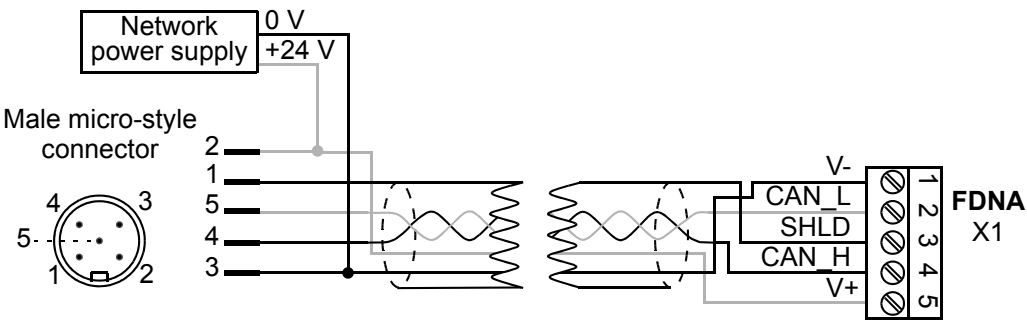
The bus cable is connected to terminal block X1 on the FDNA-01.

Terminal block description

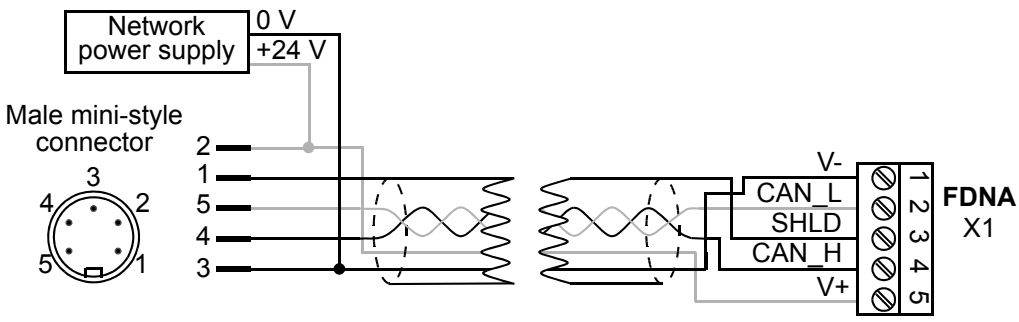
X1		Description
1	V-	Network power supply ground (0V DC)
2	CAN_L	CAN_L bus line
3	SHLD	Network cable shield
4	CAN_H	CAN_H bus line
5	V+	Network power supply source (24V DC)

Connection examples

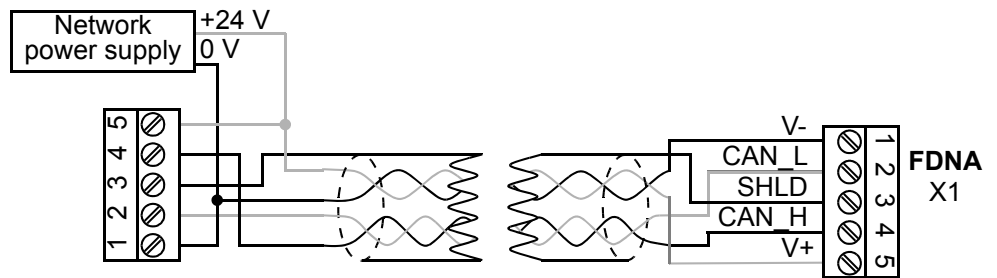
5-pin micro-style connector



5-pin mini-style connector

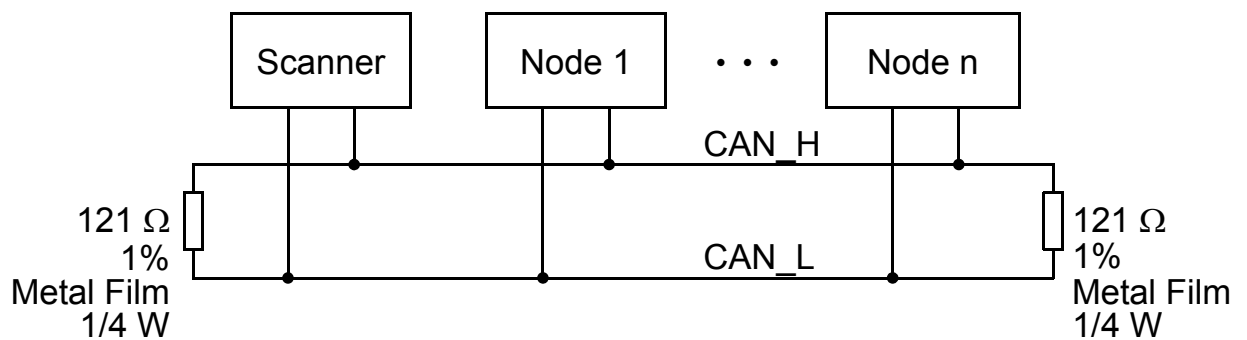


Standard open-style screw connector



DeviceNet bus termination

The FDNA-01 does not provide bus termination. The DeviceNet bus line must be terminated with 121 ohm resistors connected between the CAN_L and CAN_H wires at each end as shown below.



Drive configuration

What this chapter contains

This chapter gives information on configuring the FDNA-01 DeviceNet Adapter module.

DeviceNet connection configuration

After the adapter module has been mechanically and electrically installed according to the instructions in chapters [Mechanical installation](#) and [Electrical installation](#), the drive must be prepared for communication with the module.

The detailed procedure of activating the module for DeviceNet 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 FDNA-01 is established, several configuration parameters are copied to the drive. These parameters (Tables [1](#), [6](#) and [7](#)) 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).

Note: If communication between the FDNA-01 and DeviceNet master is established, changes to the configuration parameters can be done also through [Fieldbus configuration object 91h](#).

FDNA-01 configuration parameters - group A (group 1)

Table 1. FDNA-01 configuration parameters - group A (1) *

Par. no.	Parameter name	Alternative settings	Default setting
1	FBA TYPE	(Read-only)	DEVICENET
2	MAC ID	0 to 63	63
3	BAUD RATE	0 = 125 kbit/s 1 = 250 kbit/s 2 = 500 kbit/s	0
4	DRIVE PROFILE	0 = DCU Profile	0
5	ODVA STOP FUNC	0 = Ramp Stop 1 = Coast Stop	0
6	OUTPUT INSTANCE	See Table 2.	20
7	OUTPUT NUM PARS	1 to 10 **, ***	10
8	INPUT INSTANCE	See Table 2.	70
9	INPUT NUM PARS	1 to 10 **, ****	10
10	ODVA SPEED SCALE	See Table 4.	128
11	ODVA TORQUE SCALE	See Table 5.	128
26	UNRECOVERABLE ERROR	0 = no errors See section 26 UNRECOVERABLE ERROR.	0

* Actual parameter group number depends on the drive type. Eg, group A (group 1) equals to parameter group 51 in ACS350, ACS355, ACSM1, ACS850 and ACQ810.

** See [Appendix - Varying number of drive parameters](#) on page [137](#).

*** This parameter is only used when the output assembly instance is 120, 121, 122, 123, 901, 902, 911, 912, 921, 922. It must always be set to the default, 10, except as described in [Appendix - Varying number of drive parameters](#) on page [137](#).

**** This parameter is only used when the input assembly instance is 170, 171, 172, 173, 951, 952, 961, 962, 971, 972. It must always be set to the default, 10, except as described in the [Appendix - Varying number of drive parameters](#) on page [137](#).

[parameters](#) on page 137.

1 FBA TYPE

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

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

2 MAC ID

Each device on the DeviceNet network must have a unique MAC ID number. This parameter is used to define the MAC ID number for the drive it is connected to.

3 BAUD RATE

Sets the baud rate for the DeviceNet interface. You can freely select the parameter value, but it must be the same on every node of the DeviceNet network.

4 DRIVE PROFILE

The “drive profile” defines the communication used between the module and the drive (not fieldbus and module). If a drive supports more than one drive profile, this parameter is used to select the preferred profile. Presently to use the ODVA and ABB Drives profiles the drive must support the DCU/FBA profile.

Transparent16 and Transparent32 profiles may be used with any drive profile.

5 ODVA STOP FUNC

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

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

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

6 OUTPUT INSTANCE

8 INPUT INSTANCE

These parameters configure the output and input assembly instances used by the DeviceNet Adapter Module. The following tables list supported assemblies and allowed combinations. For descriptions of assembly instances, see [Assembly objects](#) on page 74.

Table 2. Assembly instances

Name	Output instance	Input instance	Default input 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	801	851	4	ABB Drives Profile
ABB Drives Profile w/ Set Speed and Set Torque	802	852	6	ABB Drives Profile
ABB Drives Profile w/ Set Speed plus Drive Parameters	901	951	24	ABB Drives Profile
ABB Drives Profile w/ Set Speed and Set Torque plus Drive Parameters	902	952	26	ABB Drives Profile
Transparent16 w/One	811	861	4	Transparent16 Profile
Transparent16 w/Two	812	862	6	Transparent16 Profile
Transparent16 w/One plus Drive Parameters	911	961	24	Transparent16 Profile
Transparent16 w/Two plus Drive Parameters	912	962	26	Transparent16 Profile
Transparent32 w/One	821	871	8	Transparent32 Profile

Name	Output instance	Input instance	Default input size (bytes)	Profile
Transparent32 w/Two	822	872	12	Transparent32 Profile
Transparent32 w/One plus Drive Parameters	921	971	28	Transparent32 Profile
Transparent32 w/Two plus Drive Parameters	922	972	32	Transparent32 Profile

Note: With ACSM1, ACQ810 and ACS850, when using ODVA AC/DC Drive or ABB Drives profile, make sure that the drive parameter 50.4 FBA REF MODESEL is set to SPEED. With ACSM1 and ACS850, make sure that 50.5 FBA REF2 MODESEL is set to TORQUE.

Table 3. Allowed output/input assembly instance combinations

Output/Input	ODVA (70-73; 170-173)	ABB DRIVES (851-852; 951-952)	TRANSPARENT16 (861-862; 961-962)	TRANSPARENT32 (871-872; 971-972)
ODVA (20-23; 120-123)	X			
ABB DRIVES (801-802; 901-902)		X		
TRANSPARENT16 (811-812; 911-912)			X	
TRANSPARENT32 (821-822; 921-922)				X

7 OUTPUT NUM PARS

9 INPUT NUM PARS

Some assembly instances support transferring drive parameter values between the I/O scanner and drive. These parameters specify how many drive parameter values should be included in the respective assembly instance. Changing the numbers of the drive parameters to values other than the defaults also changes the lengths of the associated assembly instances. This requires manual changes to either the EDS file or I/O scanner configuration. Please consult the [Appendix - Varying number of drive parameters](#) on page 137 before changing these parameters.

10 ODVA SPEED SCALE

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

$$\text{Speed unit} = \text{RPM} \times 2^{(-1 \times \text{ODVA speed scale value})}$$

Table 4. ODVA speed scaling

ODVA speed scale value *	Speed scale value of drive parameter **	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

* Use the ODVA speed scale value when reading/writing parameter ODVA SPEED SCALE via [AC/DC-drive object, class 2Ah](#). When written via the AC/DC drive object, the new value takes effect immediately.

** Use the speed scale value of the drive parameter when reading/writing parameter ODVA SPEED SCALE via the drive control panel, [Drive parameter object, class 90h](#) and [Fieldbus 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.

11 ODVA TORQUE SCALE

This parameter only applies when using the ODVA AC/DC Drive Profile. The units of Reference and actual torques for ODVA AC/

DC Drive profiles are given by the formula below. This parameter does not have effect on the ABB Drives profiles. Note that while a wide range of resolutions may be configured, the actual performance is limited to the performance capabilities of the drive. (N·m = Newton x Meter)

Torque unit = N·m x $2^{(-1 \times \text{ODVA torque scale})}$

Table 5. ODVA torque scaling

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

^{*} Use the ODVA torque scale value when reading/writing parameter ODVA TORQUE SCALE via [AC/DC-drive object, class 2Ah](#). When written via the AC/DC drive object, the new value takes effect immediately.

^{**} Use the torque scale value of the drive parameter when reading/writing parameter ODVA TORQUE SCALE via the drive control panel, [Drive parameter object, class 90h](#) and [Fieldbus 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.

26 UNRECOVERABLE ERROR

This parameter is a bit field parameter, that is, several status bits can be selected at a time. Value 0 indicates that there are no errors.

Bit	Name	Value	Description
1	DUP_MAC_ERROR	0x0001	Duplicate MAC ID error
2	RX_QUEUE_OVERRUN	0x0002	The message receive queue is full.
3	TX_QUEUE_OVERRUN	0x0004	The message transmit queue is full.
4	IO_SEND_ERROR	0x0008	Transmitting I/O data has failed.
5	CAN_BUS_OFF	0x0010	Bus-off is detected.
6	CAN_OVERRUN	0x0020	A CAN message was lost.
7	DNS_RESET	0x0040	DeviceNet driver of the module is reset.
8	DNS_BUS_SENSE_ERROR	0x0080	No voltage detected in the network.
9	DNS_SWITCH_ERROR	0x0100	DeviceNet driver reset failed due to an invalid MAC ID or baud rate.
13	SYS_FILE_ERR	0x1000	Initialization with the drive failed.

FDNA-01 configuration parameters - group B (group 2)

Table 6. FDNA-01 configuration parameters - group B (group 2)*

Par. no.	Parameter name	Alternative settings	Default setting
1	DATA OUT 1 (master to drive)	0 to 9999 Format: xyyy , 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 B (group 2) equals to parameter group 55 in ACS350, ACS355 and group 53 in ACSM1, ACS850 and ACQ810.

1 DATA OUT 1

In the output assembly instances that include drive parameters, this parameter specifies which parameter's value **is** placed in location DATA OUT 1 value received by the drive from the DeviceNet network. The content is specified by a decimal number as follows:

0	Unused
1...99	Virtual address area of drive
101...9999	Parameter area of drive

The parameter numbers are formatted as xxyy, where xx is the parameter group number (1...99) and yy is the parameter index within that group (01...99).

Note: The FDNA-01 configuration parameters are 16-bit parameters. If the mapped parameter is a 32-bit parameter, it automatically reserves two consecutive parameters. For example, mapping a 32-bit parameter to parameter no. 1 reserves also parameter no. 2.

2-10 DATA OUT 2 to DATA OUT 10

See DATA OUT 1 above.

FDNA-01 configuration parameters - group C (group 3)

Table 7. FDNA-01 configuration parameters - group C (group 3)*

Par. no.	Parameter name	Alternative settings	Default setting
1	DATA IN 1 (drive to master)	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 C (group 3) equals to parameter group 54 in ACS350 and ACS355, or group 52 in ACSM1, ACS850 and ACQ810.

1 DATA IN 1

In input assembly instances that include drive parameters, this parameter specifies which parameter's value is placed in location DATA IN 1 value sent by the drive to the DeviceNet network. The content is specified by a decimal number as follows.

- 0** Unused
- 1...99** Virtual address area of drive
- 101...9999** Parameter area of drive

The parameter numbers are formatted as xxyy, where xx is the parameter group number (1...99) and yy is the parameter index within that group (01...99).

Note: The FDNA-01 configuration parameters are 16-bit parameters. If the mapped parameter is a 32-bit parameter, it automatically reserves two consecutive parameters. For example, mapping a 32-bit parameter to parameter no. 1 reserves also parameter no. 2.

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 (for example, FDNA-01). ABB drives allow the user to separately determine the source for each type of control information (Start, Stop, Direction, Reference, Fault Reset, and so on). In order to give the fieldbus master station 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.

Master configuration

What this chapter contains

This chapter gives information on configuring the DeviceNet master station for communication through the FDNA-01 DeviceNet Adapter module.

Configuring the system

After the adapter module has been mechanically and electrically installed according to the instructions in the previous chapters, and has been initialized by the drive, the master station must be prepared for communication with the module.

Refer to the scanner documentation for information on configuring the system for communication with the FDNA-01.

EDS files

The Electronic Data Sheet (EDS) files specify the properties of the device for the DeviceNet scanner. The DeviceNet scanner identifies the device by means of the product code, device type, and major revision attributes. For more information, see [Identity object, class 01h](#) on page 108.

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

EDS files are available from your ABB representative.

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

Communication profiles

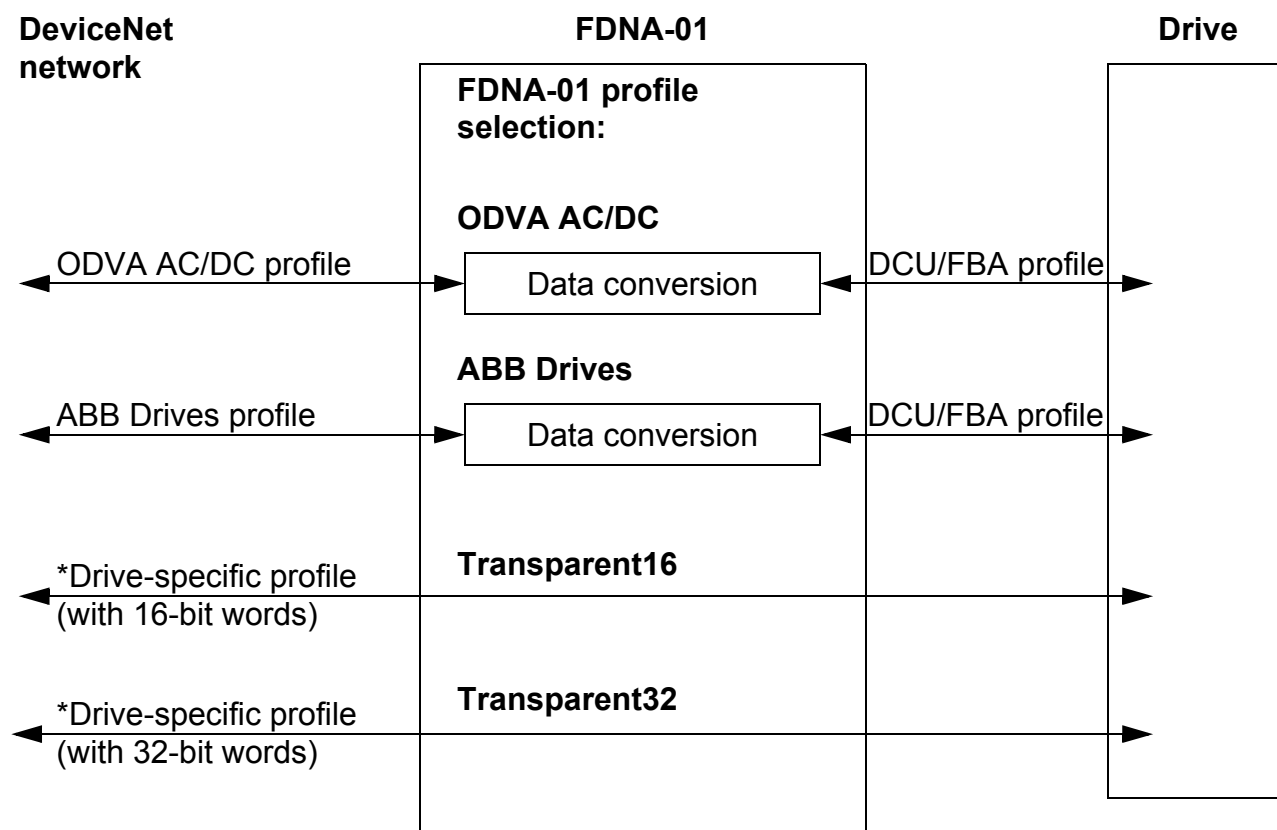
What this chapter contains

This chapter describes the communication profiles used in communication between the DeviceNet network, the FDNA-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 FDNA-01 module, the DeviceNet network may employ either the ODVA AC/DC Drive profile or the ABB Drives profile. Both are converted to the DCU/FBA profile (detailed in drive documentation) by the FDNA-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.



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/FBA 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.

A DeviceNet node is modelled 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 the AC and DC drives. The objects supported by the FDNA-01 DeviceNet Adapter are listed in [Class objects](#) on page 107.

The objects are defined by:

- Service
- Class
- Instance
- Attribute
- Behavior

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

This is an example of explicit messaging where each attribute of a class is set individually. While this is allowed, it is inefficient. Instead implicit messaging using input and output assembly instances is recommended. Implicit messaging allows the DeviceNet Master to set or get predefined groups of attributes in a single message exchange. The assembly instances supported by the FDNA-01 are listed and defined in [Assembly objects](#) on page 74.

ODVA output attributes

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

Run Forward & Run Reverse (Control supervisor object)

These attributes are used to assert run and stop commands to the Control supervisor object state machine according to the following table. See [State \(Control supervisor object\)](#) on page 60.

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 is 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 are 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 4](#) for details.

Scalar mode

When the drive is operating in the scalar mode, the FDNA-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 4.](#))
- Mf Motor nominal frequency in Hz
- Mss Motor synchronous speed in rpm (not motor nominal speed).

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

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

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

Vector mode

When the drive is operating in the vector mode, the FDNA-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 4.](#))

$$Dsr = Osr \times Us$$

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

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

Torque Reference (AC/DC drive object)

This attribute is the Torque Reference for the drive. The units are scaled by the Torque Scale attribute of the AC/DC drive object. See [Table 5.](#) for details.

The FDNA-01 provides the drive with a Torque Reference in percent of the motor nominal torque. The ODVA AC/DC Drive profiles use Newton-meter (N·m) 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 5.)
Mt	Motor nominal torque in N·m

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

For a 1000 N·m motor nominal torque with a unit of 1 N·m and an ODVA Torque Reference of 500:

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

ODVA input attributes

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

Faulted (Control supervisor object)

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

Warning (Control supervisor object)

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

Running Forward (Control supervisor object)

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

Running Reverse (Control supervisor object)

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

Ready (Control supervisor object)

This attribute indicates that the Control supervisor object state machine is in the Ready, Running or Stopping state. See [State \(Control supervisor object\)](#) on page 60.

Ctrl From Net (Control supervisor object)

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

Ref From Net (AC/DC drive object)

This attribute indicates if the Speed and Torque References are being supplied locally (Ref From Net = 0) or by the network (Ref From Net = 1).

At Reference (AC/DC drive object)

This attribute indicates that the drive is operating at the specified Speed or Torque Reference.

State (Control supervisor object)

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

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

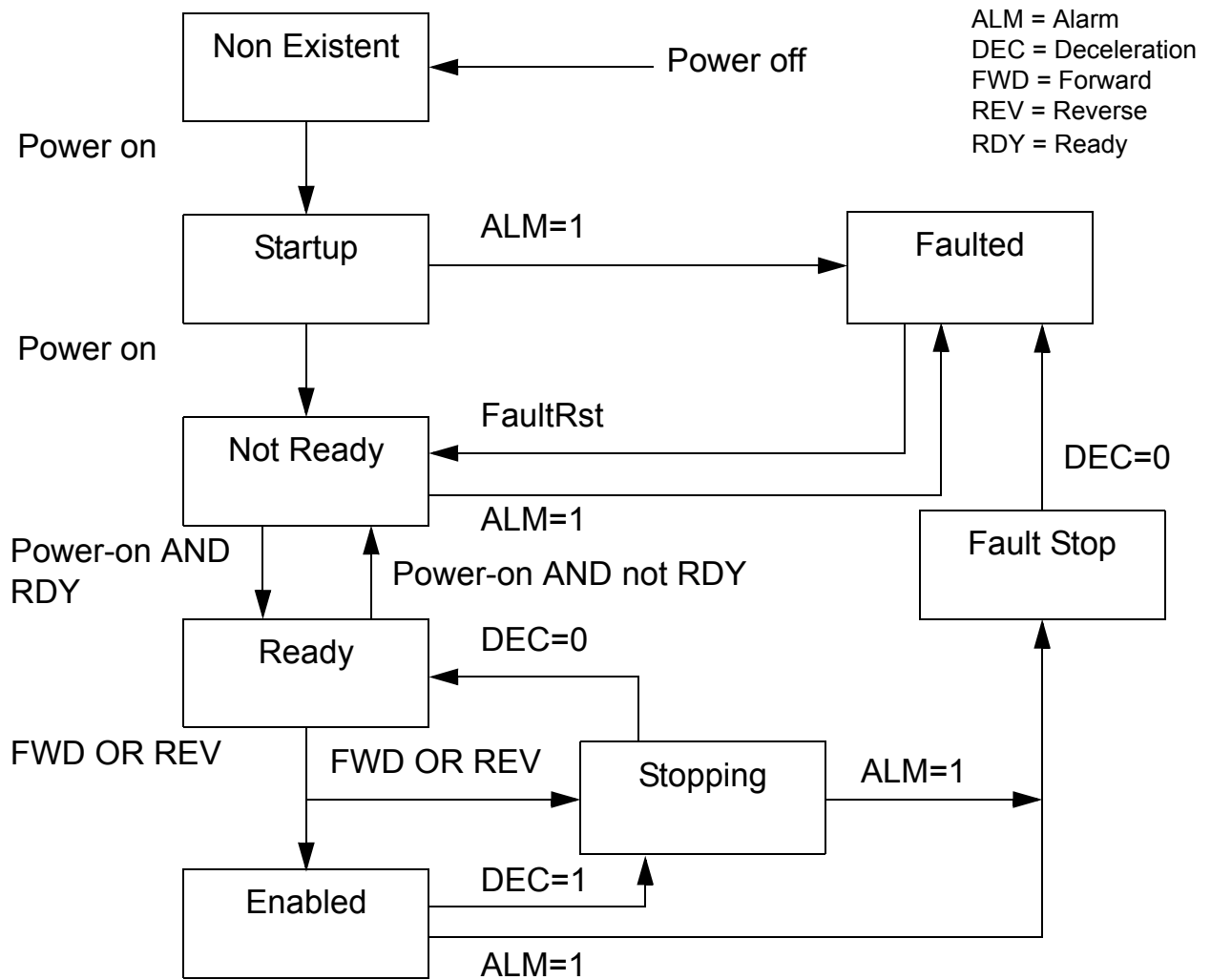


Figure 3. ODVA state transition diagram

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 4.](#) for details.

Scalar mode

When the drive is operating in the scalar mode, the drive provides the FDNA-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 4.)
Mf	Motor nominal frequency in Hz
Mss	Motor synchronous speed in rpm (not motor nominal speed).

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

For a 4-pole 60 Hz motor (Mss = 1800 rpm) with a unit of 1 rpm and a drive frequency actual of 30 Hz:

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

Vector mode

When the drive is operating in the vector mode, the drive provides the FDNA-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 4.)

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

For a drive speed actual of 900 rpm with a unit of 0.5 rpm:

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

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 5.](#) for details.

The drive provides the FDNA-01 with a torque actual in percent of the motor nominal torque. The ODVA AC/DC Drive profiles use Newton-meter (N·m) 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 5.)
Mt	Motor nominal torque in N·m

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

For a 1000 N·m motor nominal torque with a unit of 1 N·m and a drive torque actual of 50%:

$$Ota = \frac{Dta \times Mt}{100 \times Ut} = \frac{50 \times 1000 \text{ N}\cdot\text{m}}{100 \times 1 \text{ N}\cdot\text{m}} = 500$$

The ABB Drives communication profile

Control word and Status word

The Control word is the principal means for controlling the drive from a fieldbus system. It is sent by the fieldbus 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 10 and 11 respectively. The drive states are presented in the ABB Drives profile state machine (Figure 4).

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.

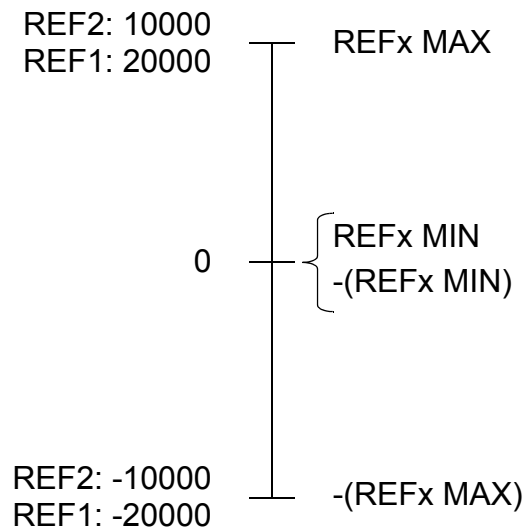
ABB drives can receive control information from multiple sources including analogue and digital inputs, the drive control panel and a communication module (for example, FDNA-01). In order to have the drive controlled through the fieldbus, the module must be defined as the source for control information,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

Scaling

References are scaled as shown below.

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

Fieldbus \longrightarrow Drive



Actual values

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

Scaling

The Actual values are scaled as shown below.

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

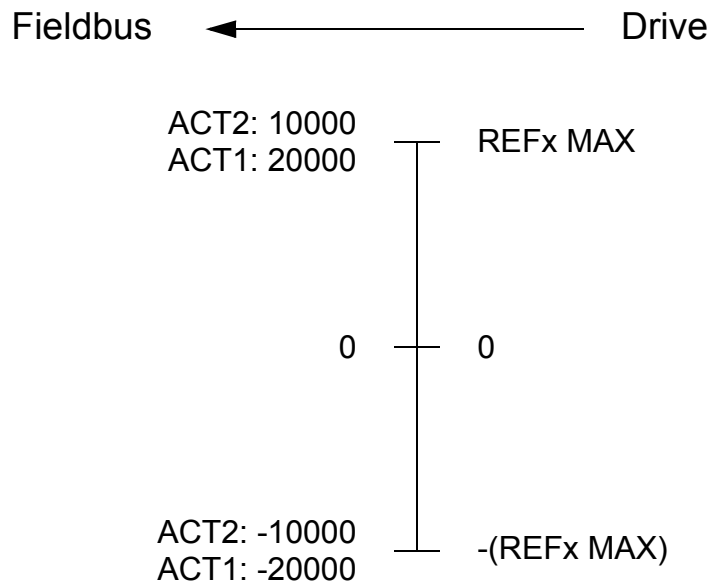


Table 10. The Control word for the ABB Drives communication profile.

The upper case text in boldface refers to the states shown in [Figure 4](#).

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 with drive parameter. Proceed to OFF3 ACTIVE ; proceed to SWITCH-ON INHIBITED . Warning: Ensure that motor and driven machine can be stopped using this stop mode.
3	INHIBIT_ OPERATION	1	Proceed to OPERATION ENABLED . Note: Run enable signal must be active; see drive documentation. If the drive is set to receive the Run enable signal from the fieldbus, this bit activates the signal.
		0	Inhibit operation. Proceed to OPERATION INHIBITED .
4	RAMP_OUT_ ZERO	1	Normal operation. Proceed to RAMP FUNCTION GENERATOR: OUTPUT ENABLED .
		0	Force Ramp Function Generator output to zero. Drive ramps to stop (current and DC voltage limits in force).

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: Effective only if the fieldbus interface is set as the source for this signal with 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: Effective only if the fieldbus interface is set as the source for this signal with 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 11. The Status word for the ABB Drives communication profile.

The upper case text in boldface refers to the states shown in [Figure 4](#).

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 with 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.

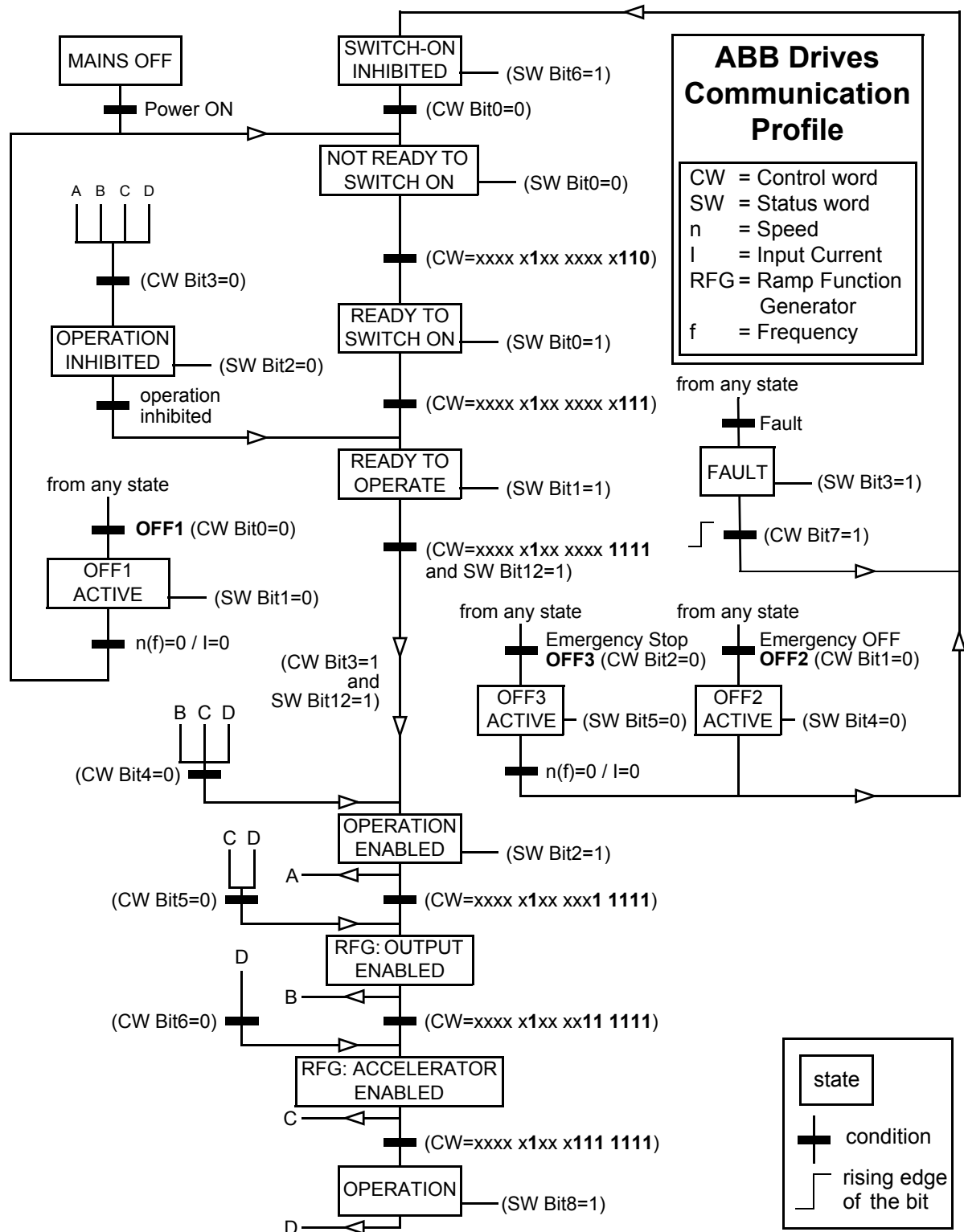


Figure 4. ABB State Transition Diagram

Communication

What this chapter contains

This chapter describes the DeviceNet communication protocol for the FDNA-01 and the configuration of the scanner. For detailed information on DeviceNet communication, refer to ODVA DeviceNet Specifications Release 2.0.

Introduction to DeviceNet

DeviceNet is a protocol based on the CAN technology. CAN specifies the physical layer interface. DeviceNet specifies the wiring, and the data transfer through CAN.

The FDNA-01 is a device acting as a Group 2 only Server realizing the Predefined Master Slave Connection Set functionality. The Off-line Connection Set functionality and Unconnected Message Manager (UCMM) are not supported.

Object modelling and functional profiles

One of the main features of DeviceNet is object modelling. A group of objects can be described with a Functional Profile. The FDNA-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 FDNA-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 (for example a polled 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 FDNA-01 uses static assemblies (in other words, fixed groupings of different object data only).

The following tables describe the assembly instances supported by the FDNA-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								
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								
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								
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)							
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								
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								
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 Forward
1								
2	Speed Reference (Low Byte)							
3	Speed Reference (High Byte)							

The format of the input assembly is:

Instance 71								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	At Reference	Ref From Net	Ctrl From Net	Ready	Running Reverse	Running 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								
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)							

Instance 121								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
23	DATA OUT 10 Value (High Byte)							

The format of the input assembly is:

Instance 171								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	At Reference	Ref From Net	Ctrl From Net	Ready	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								
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)							
4	Torque Reference (Low Byte)							
5	Torque Reference (High Byte)							

The format of the input assembly is:

Instance 72								
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	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								
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)							
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)							

Instance 122								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
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								
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)							

Instance 172								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
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								
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	Torque Reference (Low Byte)							
5	Torque Reference (High Byte)							

The format of the input assembly is:

Instance 73								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	At Reference	Ref From Net	Ctrl From Net	Ready	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								
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)							

Instance 123								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
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								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	At Reference	Ref From Net	Ctrl From Net	Ready	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)							
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)							

Instance 173								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
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 801								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Reset	Ramp in Zero	Ramp Hold	Ramp Out Zero	Inhibit Operation	Off 3 Control	Off 2 Control	Off 1 Control
1					Ext Ctrl Loc	Remote Cmd		
2	Set Speed (Low Byte)							
3	Set Speed (High Byte)							

The format of the input assembly is:

Instance 851								
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 901								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Reset	Ramp in Zero	Ramp Hold	Ramp Out Zero	Inhibit Operation	Off 3 Control	Off 2 Control	Off 1 Control
1					Ext Ctrl Loc	Remote Cmd		
2	Set Speed (Low Byte)							
3	Set Speed (High Byte)							
4	DATA OUT 1 Value (Low Byte)							
5	DATA OUT 1 Value (High Byte)							
6	DATA OUT 2 Value (Low Byte)							
7	DATA OUT 2 Value (High Byte)							
8	DATA OUT 3 Value (Low Byte)							
9	DATA OUT 3 Value (High Byte)							
10	DATA OUT 4 Value (Low Byte)							
11	DATA OUT 4 Value (High Byte)							
12	DATA OUT 5 Value (Low Byte)							
13	DATA OUT 5 Value (High Byte)							
14	DATA OUT 6 Value (Low Byte)							
15	DATA OUT 6 Value (High Byte)							
16	DATA OUT 7 Value (Low Byte)							
17	DATA OUT 7 Value (High Byte)							
18	DATA OUT 8 Value (Low Byte)							
19	DATA OUT 8 Value (High Byte)							
20	DATA OUT 9 Value (Low Byte)							

Instance 901								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
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 951								
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 (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)							

Instance 951								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
20	DATA IN 9 Value (Low Byte)							
21	DATA IN 9 Value (High Byte)							
22	DATA IN 10 Value (Low Byte)							
23	DATA IN 10 Value (High Byte)							

ABB DRIVES PROFILE WITH SET SPEED AND SET TORQUE assembly

The ABB DRIVES PROFILE WITH SET SPEED AND SET TORQUE assembly is defined by ABB. The format of the output assembly is:

Instance 802								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Reset	Ramp in Zero	Ramp Hold	Ramp Out Zero	Inhibit Operation	Off 3 Control	Off 2 Control	Off 1 Control
1					Ext Ctrl Loc	Remote Cmd		
2	Set Speed (Low Byte)							
3	Set Speed (High Byte)							
4	Set Torque (Low Byte)							
5	Set Torque (High Byte)							

The format of the input assembly is:

Instance 852								
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)							

Instance 852								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
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 902								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Reset	Ramp in Zero	Ramp Hold	Ramp Out Zero	Inhibit Operation	Off 3 Control	Off 2 Control	Off 1 Control
1					Ext Ctrl Loc	Remote Cmd		
2	Set Speed (Low Byte)							
3	Set Speed (High Byte)							
4	Set Torque (Low Byte)							
5	Set Torque (High Byte)							
6	DATA OUT 1 Value (Low Byte)							
7	DATA OUT 1 Value (High Byte)							
8	DATA OUT 2 Value (Low Byte)							
9	DATA OUT 2 Value (High Byte)							
10	DATA OUT 3 Value (Low Byte)							
11	DATA OUT 3 Value (High Byte)							
12	DATA OUT 4 Value (Low Byte)							

Instance 902								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
13	DATA OUT 4 Value (High Byte)							
14	DATA OUT 5 Value (Low Byte)							
15	DATA OUT 5 Value (High Byte)							
16	DATA OUT 6 Value (Low Byte)							
17	DATA OUT 6 Value (High Byte)							
18	DATA OUT 7 Value (Low Byte)							
19	DATA OUT 7 Value (High Byte)							
20	DATA OUT 8 Value (Low Byte)							
21	DATA OUT 8 Value (High Byte)							
22	DATA OUT 9 Value (Low Byte)							
23	DATA OUT 9 Value (High Byte)							
24	DATA OUT 10 Value (Low Byte)							
25	DATA OUT 10 Value (High Byte)							

The format of the input assembly is:

Instance 952								
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)							

Instance 952								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
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)							

TRANSPARENT16 WITH ONE assembly

The TRANSPARENT16 WITH ONE assembly, defined by ABB, provides unaltered 16-bit access to the configured drive profile.

The format of the output assembly is:

Instance 811								
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 861								
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)							

TRANSPARENT16 WITH ONE assembly PLUS DRIVE PARAMETERS

The TRANSPARENT16 WITH ONE assembly PLUS DRIVE PARAMETERS, defined by ABB, adds configurable drive parameters to the TRANSPARENT16 WITH ONE assembly.

The format of the output assembly is:

Instance 911								
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 961								
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)							

TRANSPARENT16 WITH TWO assembly

The TRANSPARENT16 WITH TWO assembly, defined by ABB, provides unaltered 16-bit access to the configured drive profile.

The format of the output assembly is:

Instance 812								
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)							

The format of the input assembly is:

Instance 862								
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)							

TRANSPARENT16 WITH TWO assembly PLUS DRIVE PARAMETERS

The TRANSPARENT16 WITH TWO assembly PLUS DRIVE PARAMETERS, defined by ABB, adds configurable drive parameters to the TRANSPARENT16 WITH TWO assembly.

The format of the output assembly is:

Instance 912								
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 962								
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)							

TRANSPARENT32 WITH ONE assembly

The TRANSPARENT32 WITH ONE assembly, defined by ABB, provides unaltered 32-bit access to the configured drive profile.

The format of the output assembly is:

Instance 821								
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 871								
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)							

TRANSPARENT32 WITH ONE assembly PLUS DRIVE PARAMETERS

The TRANSPARENT32 WITH ONE assembly PLUS DRIVE PARAMETERS, defined by ABB, adds configurable drive parameters to the TRANSPARENT32 WITH ONE assembly.

The format of the output assembly is:

Instance 921								
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 971								
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)							

TRANSPARENT32 WITH TWO assembly

The TRANSPARENT32 WITH TWO assembly, defined by ABB, provides unaltered 32-bit access to the configured drive profile.

The format of the output assembly is:

Instance 822								
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 872								
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)							

TRANSPARENT32 WITH TWO assembly PLUS DRIVE PARAMETERS

The TRANSPARENT32 WITH TWO assembly PLUS DRIVE PARAMETERS, defined by ABB, adds configurable drive parameters to the TRANSPARENT32 WITH TWO assembly.

The format of the output assembly is:

Instance 922								
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 922								
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 972								
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 972								
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 FDNA-01 DeviceNet Adapter Module is designed to provide DeviceNet communications for a variety of drives with different capabilities. Default, minimum and maximum values for the attributes vary based upon the capabilities of the drive to which the module is attached and are not documented herein. The default, minimum and maximum values for the attributes may be found in the:

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

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

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 describing the device	UINT16
4	Revision	Get	Revision of the item that 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 DeviceNet module	UINT32
7	Product Name	Get	Product identification; max 32 characters	Short String
8	State	Get	Present state of the 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.

For example, 2 = AC drive, 13 = DC drive

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 that the identity object represents.

Status

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

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

Bit(s)	Type/Name	Definition
11	Major Unrecoverable Fault	TRUE indicates that the device detected a problem which caused the device to transfer into the “Major Unrecoverable Fault” state.
12,13,14,15		Reserved, set to 0.

ODVA Serial Number:

This attribute is a number used in conjunction with the Vendor ID to form a unique identifier for each device on DeviceNet. 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 the current state of the identity object.

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

DeviceNet object, class 03h

The DeviceNet object provides the configuration and status of a DeviceNet port. Each DeviceNet product must support one (and only one) DeviceNet object per physical connection to the DeviceNet communication link.

Class attributes

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

Instance attributes

#	Attribute name	Services	Description	Data type
1	MAC ID	Get, Set	Node address	UINT8
2	Baud Rate	Get, Set	The baud rate of the device	UINT8
5	Allocation information	Get, Set	Allocation Choice Master's Mac ID	Struct UINT8 UINT8

The Allocation Information attribute consists of the following:

Allocation Choice Byte

The Allocation Choice byte indicates which of the Predefined Master/Slave Connections are active (in the Configuring, or Established state).

The Allocation Choice byte is initialized to 00 at device power-up or reset.

Master's MAC ID

The range of values is 0...63 and 255 decimal. A value in the range of 0...63 indicates that the Predefined Master/Slave Connection Set is currently allocated and denotes the MAC ID of the device that performed the allocation. The value 255 means that the Predefined Master/Slave Connection set has not been allocated. The Master's MAC ID attribute is initialized to 255 (FF hex) at device power-up/reset.

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 12. Connection object states

State	Description	State	Description
00	Nonexistent	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 DeviceNet 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 12.)	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 the CAN Identifier Field when the connection transmits	UINT16
5	Consumed Cnxn Id	Get	CAN Identifier Field value that denotes a message to be received	UINT16
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

#	Attribute name	Services	Description	Data type
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 DeviceNet 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 acknowledgement 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 is notified of the 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 DeviceNet 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
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 8 and Figure 3.

Class attributes

#	Attribute name	Services	Description	Data type
1	Revision	Get	Revision of the DeviceNet 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 the drive manual for further information on the DRIVECOMM codes.	UINT16

#	Attribute name	Services	Description	Data type
14	Warning Code	Get	Code word indicating a warning. If multiple warnings are present, the lowest code value is displayed. DRIVECOMM codes are reported. See the drive manual for further information on the 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 DeviceNet 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 4.	SINT16
8	SpeedRef	Get, Set	Units = See Table 4.	SINT16
11	Torque Actual	Get	Units = See Table 5.	SINT16
12	TorqueRef	Get, Set	Units = See Table 5.	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 4.	UINT8
24	Torque Scale	Get, Set	Torque scaling factor. See Table 5.	UINT8
29	Ref From Net	Get	Reflecting attribute 4	BOOL

Drive parameter object, class 90h

With the FDNA-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 fieldbus 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 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: FDNA-01 configuration parameters group A (group 1) **

#	Attribute name	Services	Description	Data type
1	Configuration group A (group 1) - Parameter 1	Get, Set	See Drive configuration on page 37.	UINT16
2	Configuration group A (group 1) - Parameter 2	Get, Set	See Drive configuration on page 37.	UINT16
3	Configuration group A (group 1) - Parameter 3	Get, Set	See Drive configuration on page 37.	UINT16
4	Configuration group A (group 1) - Parameter 4	Get, Set	See Drive configuration on page 37.	UINT16
5	Configuration group A (group 1) - Parameter 5	Get, Set	See Drive configuration on page 37.	UINT16
6	Configuration group A (group 1) - Parameter 6	Get, Set	See Drive configuration on page 37.	UINT16
7	Configuration group A (group 1) - Parameter 7	Get, Set	See Drive configuration on page 37.	UINT16
8	Configuration group A (group 1) - Parameter 8	Get, Set	See Drive configuration on page 37.	UINT16
9	Configuration group A (group 1) - Parameter 9	Get, Set	See Drive configuration on page 37.	UINT16
10	Configuration group A (group 1) - Parameter 10	Get, Set	See Drive configuration on page 37.	UINT16
11	Configuration group A (group 1) - Parameter 11	Get, Set	See Drive configuration on page 37.	UINT16
12	Configuration group A (group 1) - Parameter 12	Get, Set	See Drive configuration on page 37.	UINT16
13	Configuration group A (group 1) - Parameter 13	Get, Set	See Drive configuration on page 37.	UINT16
14	Configuration group A (group 1) - Parameter 14	Get, Set	See Drive configuration on page 37.	UINT16
15	Configuration group A (group 1) - Parameter 15	Get, Set	See Drive configuration on page 37.	UINT16
16	Configuration group A (group 1) - Parameter 16	Get, Set	See Drive configuration on page 37.	UINT16

#	Attribute name	Services	Description	Data type
17	Configuration group A (group 1) - Parameter 17	Get, Set	See Drive configuration on page 37.	UINT16
18	Configuration group A (group 1) - Parameter 18	Get, Set	See Drive configuration on page 37.	UINT16
19	Configuration group A (group 1) - Parameter 19	Get, Set	See Drive configuration on page 37.	UINT16
20	Configuration group A (group 1) - Parameter 20	Get, Set	See Drive configuration on page 37.	UINT16
21	Configuration group A (group 1) - Parameter 21	Get, Set	See Drive configuration on page 37.	UINT16
22	Configuration group A (group 1) - Parameter 22	Get, Set	See Drive configuration on page 37.	UINT16
23	Configuration group A (group 1) - Parameter 23	Get, Set	See Drive configuration on page 37.	UINT16
24	Configuration group A (group 1) - Parameter 24	Get, Set	See Drive configuration on page 37.	UINT16
25	Configuration group A (group 1) - Parameter 25	Get, Set	See Drive configuration on page 37.	UINT16
26	Configuration group A (group 1) - Parameter 26	Get, Set	See Drive configuration on page 37.	UINT16
27	Configuration group A (group 1) - Parameter 27	Get, Set	See Drive configuration on page 37.	UINT16

* Eg in ACS350, ACS355, ACSM1, ACS850 and ACQ810 parameter group 51.

*Instance #2: FDNA-01 configuration parameters group B (group 2) **

#	Attribute name	Services	Description	Data type
1	Configuration group B (group 2) - Parameter 1	Get, Set	See Drive configuration on page 37.	UINT16
2	Configuration group B (group 2) - Parameter 2	Get, Set	See Drive configuration on page 37.	UINT16
3	Configuration group B (group 2) - Parameter 3	Get, Set	See Drive configuration on page 37.	UINT16
4	Configuration group B (group 2) - Parameter 4	Get, Set	See Drive configuration on page 37.	UINT16
5	Configuration group B (group 2) - Parameter 5	Get, Set	See Drive configuration on page 37.	UINT16
6	Configuration group B (group 2) - Parameter 6	Get, Set	See Drive configuration on page 37.	UINT16
7	Configuration group B (group 2) - Parameter 7	Get, Set	See Drive configuration on page 37.	UINT16
8	Configuration group B (group 2) - Parameter 8	Get, Set	See Drive configuration on page 37.	UINT16
9	Configuration group B (group 2) - Parameter 9	Get, Set	See Drive configuration on page 37.	UINT16
10	Configuration group B (group 2) - Parameter 10	Get, Set	See Drive configuration on page 37.	UINT16

* Eg in parameter group 55 in ACS350, ACS355 or group 53 in ACSM1, ACS850 and ACQ810.

*Instance #3: FDNA-01 configuration parameters group C (group 3) **

#	Attribute name	Services	Description	Data type
1	Configuration group C (group 3) - Parameter 1	Get, Set	See Drive configuration on page 37.	UINT16
2	Configuration group C (group 3) - Parameter 2	Get, Set	See Drive configuration on page 37.	UINT16
3	Configuration group C (group 3) - Parameter 3	Get, Set	See Drive configuration on page 37.	UINT16
4	Configuration group C (group 3) - Parameter 4	Get, Set	See Drive configuration on page 37.	UINT16
5	Configuration group C (group 3) - Parameter 5	Get, Set	See Drive configuration on page 37.	UINT16
6	Configuration group C (group 3) - Parameter 6	Get, Set	See Drive configuration on page 37.	UINT16
7	Configuration group C (group 3) - Parameter 7	Get, Set	See Drive configuration on page 37.	UINT16
8	Configuration group C (group 3) - Parameter 8	Get, Set	See Drive configuration on page 37.	UINT16
9	Configuration group C (group 3) - Parameter 9	Get, Set	See Drive configuration on page 37.	UINT16
10	Configuration group C (group 3) - Parameter 10	Get, Set	See Drive configuration on page 37.	UINT16

* Eg parameter group 54 in ACS350, ACS355 or group 52 in ACSM1, ACS850 and ACQ810.

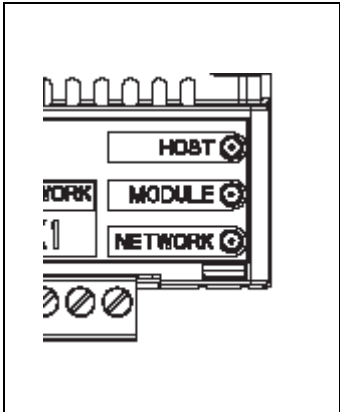
Diagnostics

What this chapter contains

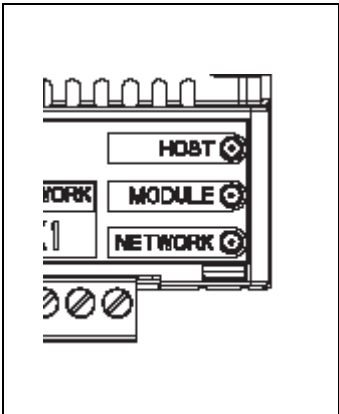
This chapter explains how to trace faults with the status LEDs on the FDNA-01 module.

LED indications

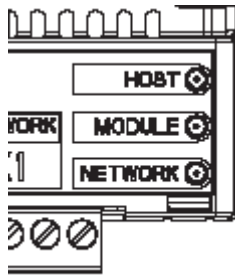
The FDNA-01 module is equipped with three bicolor diagnostic LEDs. The LEDs are described below.



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.
	Flashing red	Recoverable fault
	Red	The device has an unrecoverable fault. This may be cleared by: a Fieldbus Adapter parameter refresh, cycling DeviceNet network power or cycling drive power. This may have been caused by: a duplicate MAC error, a bus-off condition or buffer overflow.
	Flashing red-green	The device is in self test.



Name	Color	Function/State
NETWORK	Off	Device is not on-line. <ul style="list-style-type: none"> – The device has not completed the Dup_MAC_ID test yet. – The device may not be powered. Look at the Module Status LED.
	Flashing green	Device is on-line but has not connections in the established state. <ul style="list-style-type: none"> – The device has passed the Dup_MAC_ID test, is on-line, but has no established connections to other nodes. – For a Group 2 Only device it means that this device is not allocated to a master.
	Green	The device is on-line and has connections in the established state. <ul style="list-style-type: none"> – For a Group 2 Only device it means that the device is allocated to a master.
	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 Bus-off).

Definitions and abbreviations

What this chapter contains

This chapter contains abbreviations and definitions concerning the DeviceNet protocol family and FDNA-01 module.

Change of State/Cyclic Message

The Change of State/Cyclic Message is transmitted by either the master or the slave. A Change of State/Cyclic Message is directed towards a single specific node (point-to-point). An Acknowledge Message may be returned in response to this message.

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). Communication with the communication module is activated with a drive parameter.

EDS File

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

Input

In the ODVA DeviceNet specification the word 'input' is used to describe data flow from a device (such as the FDNA-01) to the network.

I/O assembly selection

Smart networked devices (like the FDNA-01) can produce and/or consume more than one I/O value. Typically, they 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 DeviceNet network has to have a unique identifier. This node number is called MAC ID (Media Access Control ID).

FDNA-01 DeviceNet Adapter module

The FDNA-01 Adapter module is one of the optional fieldbus adapter modules available for ABB drives. The FDNA-01 is a device through which an ABB drive is connected to a DeviceNet serial communication bus.

ODVA

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

Output

In the ODVA DeviceNet specification the word 'output' is used to describe data flow from the network into a device (such as the FDNA-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 FDNA-01 module.

Poll Message

Most DeviceNet Scanners as well as the FDNA-01 support two different data services. These are Poll and Change of State/Cyclic messages.

The Poll Command is an I/O Message that is transmitted by the master. A Poll Command is directed towards a single, specific slave (point-to-point, FDNA-01 always acts as a slave). A master must transmit a separate Poll Command Message for each of its slaves that is to be polled. The Poll Response is an I/O Message

that a slave transmits back to the master when the Poll Command is received.

Scanlist

The DeviceNet Scanner communicates with the DeviceNet slaves in a user-defined order. This order of communication is the scanlist. The scanlist contains a complete list of the slave nodes and the order in which the slaves are accessed.

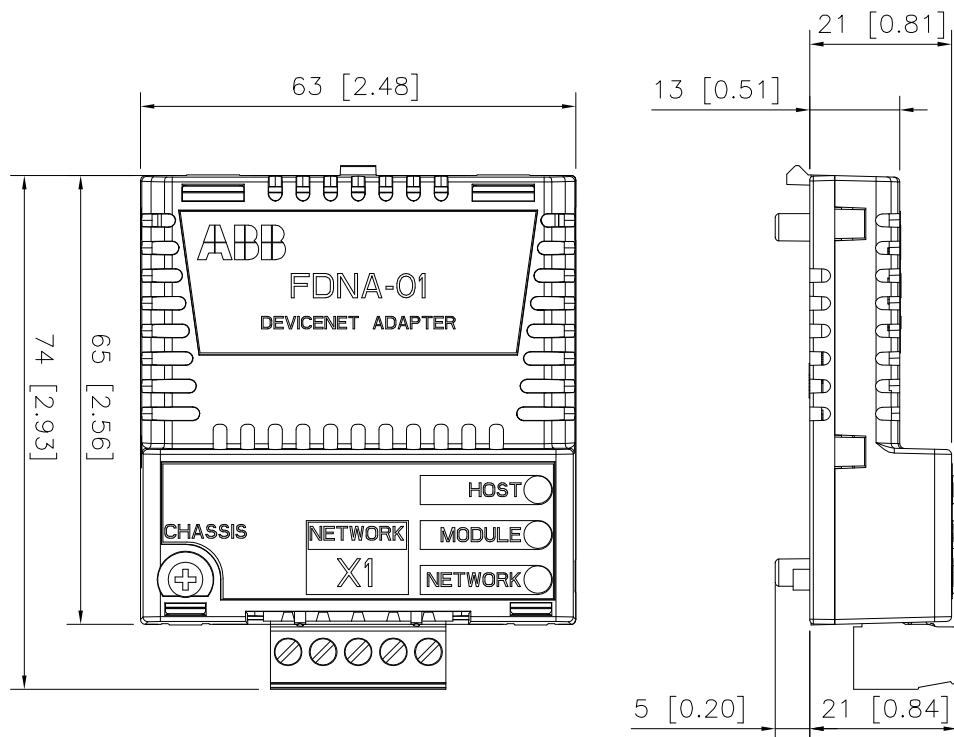
Technical data

What this chapter contains

This chapter contains technical specifications of the FDNA-01 module.

FDNA-01

Enclosure:



Mounting: Into the option slot on the drive.

Degree of protection: IP20

Ambient conditions: The applicable ambient conditions specified for the drive in its manuals are in effect.

Indicators: Three bicolor LEDs (HOST, MODULE, NETWORK)

Connectors:

- 20-pin connector to drive

- Removable 5-pole screw type connector for bus connection (max. 2.5 mm² cross section)

Power supply:

Input voltage

- From Drive: +3,3V DC (+/- 5%)
- From Network: +24V DC nominal (+11V to +30V)

Input Current

- From Drive: 180 mA typical, 300 mA maximum (+3.3V)
- From Network: 20 mA typical, 50 mA maximum (+24V)

General:

- Estimated min. lifetime: 100 000 h
- All materials UL/CSA-approved
- Complies with EMC standards EN 50081-2 and EN 50082-2
- Bus interface functionally isolated from drive

Fieldbus link

Compatible Devices: Any ODVA-compliant DeviceNet scanner supporting Poll - Response commands to Group 2 only Slaves

Medium:

- **Termination:** 121 Ω, 1%, Metal Film, 1/4 W
- **DeviceNet Cables:**
YR-29790 (Thick DeviceNet Cable)
YR-29832 (Thin DeviceNet Cable)
- **Maximum Bus Length:** 1200 m

Topology: Multi-drop

Serial Communication Type: Asynchronous, half Duplex

Transfer Rate: 125, 250 or 500 kBit/s

Protocol: DeviceNet

Appendix - Varying number of drive parameters

Assembly object instances can either be dynamic or static. The FDNA-01 only supports static assembly objects. For each supported assembly, the member list (and implicitly the length in bytes) has been fixed either by the ODVA AC/DC Drive Profiles or ABB.

FDNA-01 assemblies that include “plus Drive Parameters” have ten members each reserved for the output or input of drive parameters. In some cases, it may be desirable for these assemblies to have fewer drive parameter members. The FDNA-01 provides the flexibility to accomplish this.

However, this behaviour is non-standard and usually requires additional configuration of the DeviceNet master as well. In particular the length in bytes of the assembly must be changed to match the drive configuration. This can usually be accomplished by editing the EDS file for the drive or via manual configuration of the DeviceNet master. A description of how to change the EDS file is provided below. For manual configuration, see the documentation for the DeviceNet master.

Note: Only one EDS file with the same Product ID can be installed at a time. Changing the EDS file changes it for all drives with that Product ID. In situations where it is necessary to have different sizes of the same assembly on devices with the same Product ID, manual configuration is required.


The number of drive parameters may be configured between one and ten for output and input by changing the configuration value of "Number of Output Drive Parameters" or "Number of Input Drive Parameters" respectively. To use zero drive parameters, use the assembly upon which the "plus Drive Parameters" assembly is based.

To change the EDS file, copy it to a new file name and open the copy with a text editor. Search for the [IO_Info] section of the file

for the appropriate assembly. For example the entry for the Basic Speed Control Input plus Drive Parameters is below.

Input7=
 24, 0, 0x0005,
 "Basic Speed Control Input + Drive Parameters",
 7, "20 04 24 AA 30 03",
 "Basic Speed Control Input + Drive Parameters";

Length in bytes



The length in bytes needs to be updated to match the value from the table below. If more than one assembly "plus Drive Parameters" is being used, both assembly entries in the file must be changed. Save the file. Use the new EDS file to configure the DeviceNet master.

Length in bytes of assemblies with drive parameters

	Number of drive parameters									
Assembly	1	2	3	4	5	6	7	8	9	10
120	6	8	10	12	14	16	18	20	22	24
121	6	8	10	12	14	16	18	20	22	24
122	8	10	12	14	16	18	20	22	24	26
123	8	10	12	14	16	18	20	22	24	26
170	6	8	10	12	14	16	18	20	22	24
171	6	8	10	12	14	16	18	20	22	24
172	8	10	12	14	16	18	20	22	24	26
173	8	10	12	14	16	18	20	22	24	26
901	6	8	10	12	14	16	18	20	22	24
902	8	10	12	14	16	18	20	22	24	26
951	6	8	10	12	14	16	18	20	22	24
952	8	10	12	14	16	18	20	22	24	26
961	6	8	10	12	14	16	18	20	22	24
962	8	10	12	14	16	18	20	22	24	26
971	10	12	14	16	18	20	22	24	26	28
972	14	16	18	20	22	24	26	28	30	32

Further information

Product and service inquiries

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

Product training

For information on ABB product training, navigate to www.abb.com/drives and select *Training courses*.

Providing feedback on ABB Drives manuals

Your comments on our manuals are welcome. Go to www.abb.com/drives and select *Document Library – Manuals feedback form (LV AC drives)*.

Document library on the Internet

You can find manuals and other product documents in PDF format on the Internet. Go to www.abb.com/drives and select *Document Library*. You can browse the library or enter selection criteria, for example a document code, in the search field.



3AFE68573360 REV D EN
EFFECTIVE: 2009-12-01

ABB Oy

AC Drives

P.O. Box 184

FI-00381 HELSINKI

FINLAND

Telephone +358 10 22 11

Fax +358 10 22 22681

Internet 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

Internet www.abb.com

ABB Beijing Drive Systems Co. Ltd.

No. 1, Block D, A-10 Jiuxianqiao Beilu

Chaoyang District

Beijing, P.R. China, 100015

Telephone +86 10 5821 7788

Fax +86 10 5821 7618

Internet www.abb.com