Options for ABB drives, converters and inverters

User's manual
FCNA-01 ControlNet adapter module

Power and productivity
for a better world™

ABB

List of related manuals

	Code (EN/Multilingual)
Drive manuals and guides	
ACS355 drives (0.37…22 kW, 0.5…30 hp) user's manual	*3AUA0000066143*
ACSM1 manuals	*00578051*
ACS850-04 (0.37…45 kW) manuals	*00592009*
ACS880-01 manuals	*9AKK105408A7004*
ACS880-04 manuals	*9AKK105713A4819*
ACS880-07 manuals	*9AKK105408A8149*
Option manuals and guides	
FCNA-01 ControlNet adapter module user's manual	*3AUA0000141650*

You can find manuals and other product documents in PDF format on the Internet. See section *Document library on the Internet* on the inside of the back cover. For manuals not available in the Document library, contact your local ABB representative.

User's manual

FCNA-01 ControlNet adapter module

Table of contents	
Safety	
Mechanical installation	
Electrical installation	
Start-up	

Table of contents

7. Communication profiles

8. Communication protocol

9. Diagnostics

10. Technical data

Further information

1

Safety

Contents of this chapter

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

Use of warnings

Warnings tell you about conditions which can cause injury or death, or damage to the equipment. They also tell you how to prevent the danger. The manual uses these warning symbols:

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

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

Safety in installation

These instructions are for all who install or connect an optional module to a drive, converter or inverter, and need to open its front cover or door to do the work.

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

- If you are not a qualified electrician, do not do installation or maintenance work.
- Disconnect the drive, converter or inverter from all possible power sources. After you have disconnected the drive, converter or inverter, always wait for 5 minutes to let the intermediate circuit capacitors discharge before you continue.
- Disconnect all dangerous voltages connected to any control signal connectors in reach. For example, it is possible that 230 V AC is connected from outside to a relay output of the drive, converter or inverter.
- Always use a multimeter to make sure that there are no parts under voltage in reach. The impedance of the multimeter must be at least 1 Mohm.

2

Introduction to the manual

Contents of this chapter

This chapter introduces this manual.

Applicability

This manual applies to the FCNA-01 ControlNet adapter module, SW version V1.05 or later.

Compatibility

The FCNA-01 ControlNet adapter module is compatible with the following drives:

- ACS355
- ACSM1
- ACS850
- ACS880.

The adapter module is compatible with all master stations that support the ControlNet protocol.

Target audience

This manual is intended for people who plan the installation, install, start up, use and service the adapter module. Before you do work on the module, read this manual and the applicable drive/converter /inverter manual that contains the hardware and safety instructions for the product in question.

You are expected to know the fundamentals of electricity, wiring, electrical components and electrical schematic symbols.

The manual is written for readers worldwide. Both SI and imperial units are shown.

Purpose of the manual

The manual provides information on installing, commissioning and using an FCNA-01 ControlNet adapter module.

Contents

The manual consists of the following chapters:

- *Safety* presents the safety instructions which you must follow when installing a fieldbus adapter module.

- *Introduction to the manual* introduces this manual.

- *Overview of the ControlNet network and the FCNA-01 adapter module* contains a short description of the ControlNet network and the FCNA-01 ControlNet adapter module.

- *Mechanical installation* contains a delivery checklist and instructions to install the adapter module.

- *Electrical installation* contains general cabling instructions and instructions on connecting the module to the ControlNet network.

- *Start-up* presents the steps to take during the start-up of the drive with the adapter module and gives examples of configuring the master system.

- *Communication profiles* describes the communication profiles used in the communication between the ControlNet network, the adapter module and the drive.

- *Communication protocol* describes the communication on an ControlNet network.
- *Diagnostics* explains how to trace faults with the status LEDs on the adapter module.
- *Technical data* contains the technical data of the adapter module and the ControlNet link.

Terms and abbreviations

Note: Later in this manual, term *drive* substitutes for string drive/converter/inverter.

■ Terms

Term	Explanation
Application object	Reference to multiple Object classes that implement product-specific features.
Attribute	Description of an externally visible characteristic or feature of an object. Typically, attributes provide status information or govern the operation of an Object. For example: the ASCII name of an object; and the repetition rate of a cyclic object.
Behavior	Specification of how an object acts. Actions result from different events the object detects, such as receiving service requests, detecting internal faults or elapsing timers.
Class	Set of objects that all represent the same kind of system component. A class is a generalization of an object. All objects in a class are identical in form and behavior, but may contain different attribute values.
Command word	See Control word.
Communication module	Communication module is a name for a device (eg, a fieldbus adapter) through which the drive is connected to an external communication network (eg, a fieldbus). The communication with the module is activated with a drive parameter.

Term	Explanation
Communication object	Reference to the Object classes that manage and provide the run-time exchange of implicit (I/O) and explicit messages.
Control word	16-bit or 32-bit word from master to slave with bit-coded control signals (sometimes called the Command word)
EDS file	Electronic data sheet (EDS) file identifies the properties of the device to the EtherNet/IP client. Each type of drive and application program requires its own EDS file.
FCNA-01 ControlNet adapter module	One of the optional fieldbus adapter modules available for ABB drives. FCNA-01 is a device through which an ABB drive is connected to an ControlNet network.
Input	In the ODVA ControlNet specification the word 'input' is used to describe data flow from a device (such as the adapter module) to the network.
Instance	Specific and real (physical) occurrence of an object. For example: New Zealand is an instance of the object class Country. Terms Object, Instance, and Object Instance all refer to a specific Instance.
Instantiate	To create an instance of an object with all instance attributes initialized to zero unless default values are specified in the object definition.
Media access control identifier	Identification value assigned to each node on the CIP network. This value distinguishes a node among all other nodes on the same link. MAC ID format is network-specific.
Object	Abstract representation of a particular component within a product.
Output	In the ODVA ControlNet specification the word 'output' is used to describe data flow from the network into a device (such as the adapter module).

Term	Explanation
Profile	Adaptation of the protocol for certain application field, for example, drives. In this manual, drive-internal profiles (eg, DCU or FBA) are called native profiles.
Service	Function supported by an object and/or object class. CIP defines a set of common services and provides for the definition of Object class and/or vendor-specific services. CIP common services are those whose parameters and required behaviors are defined in Appendix A.
Status word	16-bit or 32-bit word from slave to master with bit-coded status messages.

■ Abbreviations

Abbreviation	Explanation
MAC ID	See *Media access control identifier*.
NAP	Network access port for a temporary connection of configuration tools
NUI	Network update interval
NUT	Network update time
ODVA™	ODVA stands for Open DeviceNet Vendor Association. ODVA is an independent organization that promotes interoperativity between different manufacturers' ControlNet products. ABB is an Associate Member at ODVA.

3

Overview of the ControlNet network and the FCNA-01 adapter module

Contents of this chapter

This chapter contains a short description of the ControlNet network and the FCNA-01 ControlNet adapter module.

ControlNet network

ControlNet is a scheduled communication network designed for cyclic data exchange. The protocol operates in cycles called Network update intervals (NUIs).

Each NUI has three phases:

- The first phase is dedicated to scheduled traffic, where all nodes with scheduled data are guaranteed a transmission opportunity.

- The second phase is dedicated to unscheduled traffic.

  There is no guarantee that every node will get an opportunity to transmit in every unscheduled phase.

- The third phase is network maintenance or "guardband".

  It includes synchronization and a means of determining the starting node on the next unscheduled data transfer.

Both the scheduled and unscheduled phase use an implicit token ring media access method. The amount of time each NUI consists of is called the Network Update Time (NUT). It is configurable from 2 to 100 ms. The default NUT on an unscheduled network is 5 ms.

The maximum size of a scheduled or unscheduled ControlNet data frame is 510 Bytes.

The network topology is a bus structure with short taps. ControlNet supports a star topology if used with the appropriate hardware.

ControlNet cables consist of a RG-6 coaxial cable with BNC connectors. Optical fiber is sometimes used for long distances. ControlNet can operate with a single RG-6 coaxial cable bus, or a dual RG-6 coaxial cable bus for cable redundancy. In all cases, the RG-6 should be of quad-shield variety.

The maximum cable length without repeaters is 1000 m and the maximum number of nodes on the bus is 99. The network can support 5 repeaters (10 when used for redundant networks).

The physical layer signaling uses Manchester code at 5 Mbit/s. For more information, see chapter *Technical data*.

■ **Example topology**

An example of an allowable topology is shown below.
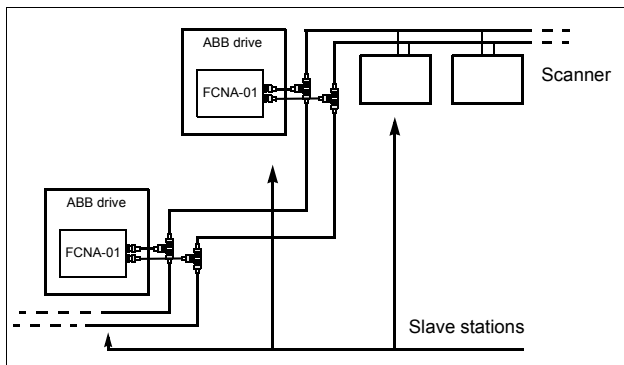
FCNA-01 adapter module

The FCNA-01 ControlNet adapter module is an optional device for ABB drives. It enables the connection of the drive to a ControlNet network.

Through the adapter module you can:

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

■ Layout of the adapter module

No.	Description	See chapter
1	Lock	*Mechanical installation*
2	Mounting screw	*Mechanical installation*
3	Bus connector Line A	*Electrical installation*
4	Network Access Port (NAP) for a temporary connection of configuration tools	*Communication protocol*
5	Bus connector Line B	*Electrical installation*
6	Diagnostic LEDs	*Diagnostics*
7	Node ID address selection switch	*Start-up*

4

Mechanical installation

Contents of this chapter

This chapter contains a delivery checklist and instructions to install the adapter module.

Necessary tools and instructions

See the applicable drive hardware manual.

Unpacking and examining the delivery

1. Open the option package.

2. Make sure that the package contains:
   • ControlNet adapter module, type FCNA-01
   • this manual.

3. Make sure that there are no signs of damage.

Installing the adapter module

⚠ **WARNING!** Obey the safety instructions. See chapter *Safety* on page *11*. If you ignore the safety instructions injury or death can occur.

The adapter module has a specific position in the drive. Plastic pins, a lock and one screw hold the adapter module in place. The screw also makes an electrical connection between the module and drive frame for cable shield termination.

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

When you install or remove the adapter module from the control unit:

1. Pull out the lock.
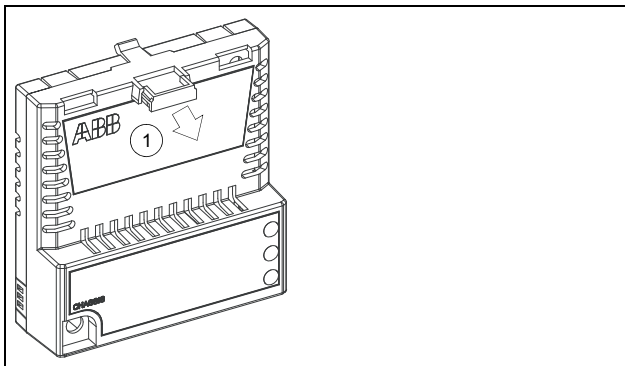
2. Insert the adapter module carefully into its position on the drive.

3. Push in the lock.
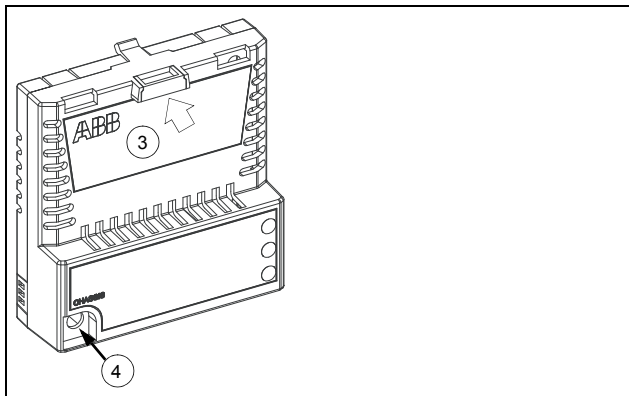
4. Tighten the screw (0.8 N·m).

Note: It is essential to install the screw properly to fulfill the EMC requirements and to ensure the proper operation of the adapter module.

See the appropriate drive manual for instructions on how to install the adapter module to the drive.

5

Electrical installation

Contents of this chapter

This chapter contains:
- general cabling instructions
- instructions on connecting the module to the ControlNet network.

Warnings

WARNING! Obey the safety instructions. See chapter *Safety* on page *11*. If you ignore the safety instructions, injury or death can occur. If you are not a qualified electrician, do not do electrical work.

Necessary tools and instructions

See the applicable drive hardware manual.

General cabling instructions

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

When you connect the network cables, carefully insert the cable so that the plug enters the jack straightly without any misalignment and without applying any twisting or bending moments to the cable or the plug. Do not use excessive force. Make sure that the plug latches into place and finally check that the plug has entered all the way into the jack.

Route the cables so that they do not transmit bending stress to the connector.

The drive must be mounted so that there is enough room (15 cm) for the network cables so that the cables can be easily connected and disconnected and the cables need not go through an unreasonably small bending radius.

Connecting the adapter module to the ControlNet network

The network cable can be RG6/U - Quad Shielded Belden 3092A, 3092F & YR28890, 3093A or CommScope 5060 & 5060IS, 5060F & 5740F, 5061 or equivalent.

■ Connection procedure

1. Connect the bus cable to the BNC connectors A and/or B on the adapter module.

   Use both connectors in redundant operation, otherwise use either connector A or B.

2. Terminate the ControlNet bus line with a 75-ohm resistor. See chapter *Technical data*.

6

Start-up

Contents of this chapter

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

Warnings

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

Drive configuration

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

■ ControlNet connection configuration

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

The detailed procedure of activating the adapter module for ControlNet communication with the drive depends on the drive type. Normally, you must adjust a parameter to activate the communication. See the drive-specific start-up sections starting on page *46*.

Once communication between the drive and the adapter module has been established, several configuration parameters are copied to the drive. These parameters are shown in the tables below and must be checked first and adjusted where necessary.

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

Note: The new settings take effect only when the adapter module is powered up the next time or when the fieldbus adapter refresh parameter is activated.

FCNA-01 configuration parameters – group A (group 1)

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

- parameter group 51 in ACS355, ACSM1 and ACS850
- parameter group 51 in ACS880 if the adapter is installed as fieldbus adapter A or group 54 if the adapter is installed as fieldbus adapter B.

No.	Name/Value	Description	Default
01	FBA TYPE	**Read-only.** Shows the fieldbus adapter type as detected by the drive. Value cannot be adjusted by the user. If the value is **0** = None, the communication between the drive and the module has not been established.	**0x65** = Control-Net
02	PROTOCOL/ PROFILE **ACS355:** FB PAR 2 **ACSM1:** FBA PAR2 **ACS850:** FBA par2 **ACS880:** Profile	Selects the application protocol and communication profile for the network connection. Selections available for ControlNet communication are listed below.	**0** = ODVA
	0 = ODVA	ODVA AC/DC drive profile	
	1 = ABB Drives profile	ABB Drives profile	
	2 = Transparent 16	Transparent 16-bit profile	
	3 = Transparent 32	Transparent 32-bit profile	
03	MODULE MACID **ACS355:** FB PAR 3 **ACSM1:** FBA PAR3 **ACS850:** FBA par3 **ACS880:** MacID	Selects the MAC ID for the node.	**2**
	0…99	MAC ID	

No.	Name/Value	Description	Default
04	MODULE BAUD RATE	**Read-only.** Fixed to 5 Mbit/s.	**5** = 5 Mbit/s
	ACS355: FB PAR 4		
	ACSM1: FBA PAR4		
	ACS850: FBA par4		
	ACS880: Baud rate		
	5 = 5 Mbit/s	Communication speed is 5 Mbit/s.	
05	HW/SW OPTION	Defines the selection source of the module MAC ID.	**0** = HW
	ACS355: FB PAR 5		
	ACSM1: FBA PAR5		
	ACS850: FBA par5		
	ACS880: HW/SW Option		
	0 = HW	HW option selected	
	1 = SW	SW option selected	
06	ODVA STOP FUNCTION	Applies only when the ODVA AC7DC Drive profile is used. Determines how the motor will be stopped when a stop command is received.	**0** = RAMP
	ACS355: FB PAR 6		
	ACSM1: FBA PAR6		
	ACS850: FBA par6		
	ACS880: Stop function		
	0 = RAMP	Motor decelerates along the active deceleration ramp.	
	1 = COAST	Motor comes to a stop by coasting.	

No.	Name/Value	Description	Default
07	ODVA SPEED SCALE **ACS355:** FB PAR 7 **ACSM1:** FBA PAR7 **ACS850:** FBA par7 **ACS880:** Speed scale	Defines the speed scale in the ODVA AC/DC drive profile. Units of reference and actual speeds for the ODVA AC/DC drive profile are given by the formula below. No effect on the ABB Drives profiles. **Note:** While a wide range of resolutions may be configured, the actual performance is limited to the performance capabilities of the drive. Speed unit = RPM × $2^{(-1 \times \text{ODVA speed scale value})}$ Table below shows how the values of drive parameter ODVA SPEED SCALE correspond to the ODVA Speed Scale units.	128[1]

ODVA speed scale value[1]	Speed scale value of drive parameter[2]	Unit
-5	123	32 RPM
-4	124	16 RPM
-3	125	8 RPM
-2	126	4 RPM
-1	127	2 RPM
0 (default)	128	1 RPM
1	129	0,5 RPM
2	130	0,25 RPM
3	131	0,125 RPM
4	132	0,0625 RPM
5	133	0,03125 RPM

[1] Use the ODVA speed scale value when reading/writing parameter *07 ODVA SPEED SCALE* via *AC/DC drive, class 2Ah*. When written via the AC/DC drive object, the new value takes effect immediately.

[2] Use the speed scale value of the drive parameter when reading/writing parameter *07 ODVA SPEED SCALE* via the drive control panel, *Drive parameter, class 90h* and *Fieldbus configuration, class 91h*. When written via these methods, the new value takes effect after the drive is repowered or a "Fieldbus Adapter Parameter refresh" is given.

| 0…255 | Speed scale value of the drive parameter | |

No.	Name/Value	Description	Default
08	ODVA TORQUE SCALE **ACS355:** FB PAR 8 **ACSM1:** FBA PAR8 **ACS850:** FBA par8 **ACS880:** Torque scale	Defines the torque scale in the ODVA AC/DC drive profile. Units of reference and actual torques for the ODVA AC/DC drive profile are given by the formula below. No effect on the ABB Drives profiles. **Note:** 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 $2^{(-1 \times \text{ODVA torque scale})}$	128
		Table below shows how the values of drive parameter ODVA TORQUE SCALE correspond to the ODVA Torque Scale units.	

ODVA torque scale value[1]	Torque scale value of drive parameter[2]	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

[1] Use the ODVA torque scale value when reading/writing parameter *08 ODVA TORQUE SCALE* via *AC/DC drive, class 2Ah*. When written via the AC/DC drive object, the new value takes effect immediately.

[2] Use the torque scale value of the drive parameter when reading/writing parameter *08 ODVA TORQUE SCALE* via the drive control panel, *Drive parameter, class 90h* and *Fieldbus configuration, class 91h*. When written via these methods, the new value takes effect after the drive is repowered or a "Fieldbus Adapter Parameter refresh" is given.

	0…255	Torque scale value of the drive parameter	

No.	Name/Value	Description	Default
09	T16 SCALE **ACS355:** FB PAR 9 **ACSM1:** FBA PAR9 **ACS850:** FBA par9 **ACS880:** T16 scale	Defines the reference multiplier/actual value divisor for the adapter module. Effective only when the Transparent 16 profile is selected AND the drive is using the native communication profile (eg, DCU or FBA) and 16-bit transparent Reference 1/Actual value 1. With an ACS355 drive, the speed reference from the PLC is multiplied by the value of this parameter plus one. For example, if the parameter has a value of 99 and a reference of 1000 given by the master, the reference will be multiplied by 99 +1 = 100 and forwarded to the drive as 100000. According to the DCU profile, this value is interpreted as a reference of 100 rpm in the drive. With ACSM1, ACS850 and ACS880, setting this parameter to 65535 provides the approximation of 1 = 1 rpm.	99
	0…65535	Reference multiplier/actual value divisor	
10	CONTROL TIMEOUT **ACS355:** FB PAR 10 **ACSM1:** FBA PAR10 **ACS850:** FBA par10 **ACS880:** Control Timeout	The ControlNet protocol specifies connection timeout for I/O messaging (Class 1) and Connected explicit messaging (Class 3), but not Unconnected explicit messaging. This parameter provides a timeout for Unconnected explicit messaging and for instances of Connected explicit messaging (Class 3), where the client breaks the connection in between requests.	99

No.	Name/Value	Description			Default
	Connection type	**Control timeout**	**Timeout source**		
	I/O messaging (Class 1)	0…65535	(Requested Packet Interval) X (Connection Timeout Multiplier) **Note:** Timeout behavior may be modified by Watchdog Timeout Action attribute of Connection object.		
	Connected explicit messaging (Class 3)	0	(Requested Packet Interval) X (Connection Timeout Multiplier) **Note:** Timeout behavior may be modified by Watchdog Timeout Action attribute of Connection object.		
		1…65534	100ms X (Control Timeout Value) since last Control Event		
		65535	Never Timeout		
	Unconnected explicit messaging	0	Always Timeout **Note:** Control Timeout must be greater than zero to control drive with Unconnected Explicit Messaging.		
		1…65534	100ms X (Control Timeout Value) since last Control Event		
		65535	Never Timeout		
		Control timeout events: • Write of an output assembly object instance • Write of control bits (Run1, Run2, NetCtrl, NetRef and FaultReset) • Write Speed Reference • Write Torque Reference • Reset Control Supervisor object • Write Force Fault via Control Supervisor object In the event of a timeout, the adapter module will signal the drive that communication with the client has been lost. The drive configuration will determine how it will respond. **Example:** If the timeout is configured for 250 ms and the drive is configured to fault on a communication failure with a delay of 500 ms, then the drive will fault 750 ms after communications is lost.			
0…65535		Control timeout value			

No.	Name/Value	Description	Default
11	IDLE ACTION **ACS355:** FB PAR 11 **ACSM1:** FBA PAR11 **ACS850:** FBA par11 **ACS880:** Idle Action	I/O connections may include a Run/Idle notification. Determines the action the drive takes in response to an Idle notification.	**1** = ONLINE
	0 = OFFLINE	In the event of an Idle notification, the adapter module will signal the drive that communication with the client has been lost. Drive configuration will determine how it will respond. **Example:** If the time-out is configured for 250 ms and the drive is configured to fault on a communication failure with a delay of 500 ms, then the drive will fault 750 ms after communications is lost.	
	1 = ONLINE	In the event of an Idle notification, the drive will continue to operate using the last command and references received.	
12	T-> O Real Time Format **ACS355:** FB PAR 12 **ACSM1:** FBA PAR12 **ACS850:** FBA par12 **ACS880:** Real Time Format	Defines the length of the header of class 0 message sent from drive to PLC.	**0** = No Run/Idle notification included
	0 = No Run/Idle notification included	Message does not contain a Run/Idle header. Default value.	
	1 = 32 bit Run/Idle header	Message contains the run/idle header. Value of header is always 0. Adapter is compatible with in this mode.	
…	…	…	…

No.	Name/Value	Description	Default
26	RESTORE FACTORY SETTINGS **ACS355:** FB PAR 26 **ACSM1:** FBA PAR26 **ACS850:** FBA par26 **ACS880:** Restore Def Conf	Clears the non-volatile storage in the adapter module. Saved objects are Id, Keeper and COCO.	0
	0 = No	Non-volatile storage will not be cleared	
	1 = Yes	Non-volatile storage will be cleared	
27	FBA PAR REFRESH **ACS355/ACSM1:** FBA PAR REFRESH **ACS850:** FBA par refresh **ACS880:** FBA A/B par refresh	Validates any changed adapter module configuration parameter settings. After refreshing, the value reverts automatically to **0** = Done. **Note:** This parameter cannot be changed while the drive is running.	**0** = Done
	0 = Done	Refreshing done	
	1 = Refresh / Configure	Refreshing	
28	PAR TABLE VER **ACS355:** FILE CPI FW REV **ACSM1:** PAR TABLE VER **ACS850:** Par table ver **ACS880:** FBA A/B par table ver	**Read-only.** Displays the parameter table revision of the fieldbus adapter module mapping file stored in the memory of the drive. In format **xyz**, where **x** = major revision number **y** = minor revision number **z** = correction number OR in format **axyz**, where **a** = major revision number **xy** = minor revision numbers **z** = correction number or letter.	N/A
		Parameter table revision	

No.	Name/Value	Description	Default
29	DRIVE TYPE CODE **ACS355:** FILE CONFIG ID **ACSM1:** DRIVE TYPE CODE **ACS850:** Drive type code **ACS880:** FBA A/B drive type code	**Read-only.** Displays the drive type code of the fieldbus adapter module mapping file stored in the memory of the drive.	N/A
		Drive type code of the fieldbus adapter module mapping file	
30	MAPPING FILE VER **ACS355:** FILE CONFIG REV **ACSM1:** MAPPING FILE VER **ACS850:** Mapping file ver **ACS880:** FBA A/B mapping file ver	**Read-only.** Displays the fieldbus adapter module mapping file revision stored in the memory of the drive in decimal format.	N/A
		Mapping file revision	
31	D2FBA COMM STA **ACS355:** FBA STATUS **ACSM1:** D2FBA COMM STA **ACS850:** D2FBA comm sta **ACS880:** D2FBA A/B comm status	**Read-only.** Displays the status of the fieldbus adapter module communication. **Note:** The value names may vary by drive.	**0** = Idle OR **4** = Off-line
	0 = Idle	Adapter is not configured.	
	1 = Exec.init	Adapter is initializing.	

No.	Name/Value	Description	Default
	2 = Time out	A timeout has occurred in the communication between the adapter and the drive.	
	3 = Conf.err	Adapter configuration error: The major or minor revision code of the common program revision in the fieldbus adapter module is not the revision required by the module or mapping file upload has failed more than three times.	
	4 = Off-line	Adapter is off-line.	
	5 = On-line	Adapter is on-line.	
	6 = Reset	Adapter is performing a hardware reset.	
32	FBA COMM SW VER **ACS355:** FBA CPI FW REV **ACSM1:** FBA COMM SW VER **ACS850:** FBA comm sw ver **ACS880:** FBA A/B comm SW ver	**Read-only.** Displays the common program revision of the adapter module in format **axyz**, where: **a** = major revision number **xy** = minor revision numbers **z** = correction number or letter.	N/A
		Common program version of the adapter module	
33	FBA APPL SW VER **ACS355:** FBA APPL FW REV **ACSM1:** FBA APPL SW VER **ACS850:** FBA appl sw ver **ACS880:** FBA A/B appl SW ver	**Read-only.** Displays the application program revision of the adapter module in format **axyz**, where: **a** = major revision number **xy** = minor revision numbers **z** = correction number or letter.	N/A
		Application program revision of the adapter module	

FCNA-01 configuration parameters – group B (group 2)

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

• parameter group 55 in ACS355

• parameter group 53 in ACSM1 and ACS850

• parameter group 53 in ACS880 if the adapter is installed as fieldbus adapter A or group 56 if the adapter is installed as fieldbus adapter B.

No.[1]	Name/Value	Description		Default
01	DATA OUT 1 (client to drive) **ACS355:** FBA DATA OUT 1 **ACSM1:** FBA DATA OUT1 **ACS850:** FBA data out1 **ACS880:** FBA A/B data out1	In output assembly instances that include drive parameters, this parameter specifies which parameter's value will be placed in location DATA OUT 1 value received by the drive from the ControlNet client. Content is defined by a decimal number in the range of 0 to 9999 as follows:		**0** = None
		0	Not used	
		1…99	Virtual address area of drive control. Not used when the ControlNet protocol is used.	
		101…9999	Parameter area of the drive	
	0 = None	Not used		
	101…9999	Parameter index with format **xxyy**, where • **xx** is the parameter group number (1…99) • **yy** is the parameter number index within that group (01…99). **Note:** In ACS880, choose Other to display a list of mappable drive parameters.		
02… 10	DATA OUT 2 … DATA OUT 10	See parameter *01 DATA OUT 1*.		**0** = None

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

FCNA-01 configuration parameters – group C (group 3)

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

- parameter group 54 in ACS355
- parameter group 52 in ACSM1 and ACS850
- parameter group 52 in ACS880 if the adapter is installed as fieldbus adapter A or group 55 if the adapter is installed as fieldbus adapter B.

No.[1]	Name/Value	Description	Default
01	DATA IN 1 (drive to client)	In input assembly instances that include drive parameters, this parameter specifies which parameter's value will be placed in location DATA IN 1 value sent by the drive to the ControlNet client. Content is defined by a decimal number in the range of 0 to 9999 as follows:	**0** = None
	ACS355: FBA DATA IN 1 **ACSM1:** FBA DATA IN1 **ACS850:** FBA data in1 **ACS880:** FBA A/B data in1	<table><tr><td>0</td><td>Not used</td></tr><tr><td>1…99</td><td>Virtual address area of drive control. Not used when the ControlNet protocol is used.</td></tr><tr><td>101… 9999</td><td>Parameter area of the drive</td></tr></table>	
	0 = None	Not used	
	101…9999	Parameter index with format **xxyy**, where • **xx** is the parameter group number (1…99) • **yy** is the parameter number index within that group (01…99). **Note:** In ACS880, choose Other to display a list of mappable drive parameters.	
02… 10	DATA IN 2 … DATA IN 10	See parameter *01 DATA IN 1*.	**0** = None

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

■ **Control locations**

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

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

Starting up ACS355 drives

1. Power up the drive.

2. Enable the communication between the adapter module and the drive with parameter 9802 COMM PROT SEL.

3. Set the FCNA-01 configuration parameters in group 51.

   At the minimum, select the communication protocol and profile with parameter 5102 and set the MAC ID address either with parameters 5103 and 5105 or with the switches.

4. With parameter 3018 COMM FAULT FUNC, select how the drive reacts to a fieldbus communication break.

5. With parameter 3019 COMM FAULT TIME, define the time between communication break detection and the selected action.

6. Define the process data transferred to and from the drive in parameter groups 54 and 55.

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

7. Validate the settings made in parameter groups 51, 54 and 55 with parameter 5127 FBA PAR REFRESH.

8. Set the relevant drive control parameters to control the drive according to the application.

   Examples of appropriate values are shown in the tables below.

■ Parameter setting examples – ACS355

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

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

The start/stop commands and reference scaling are according to the ODVA AC/DC drive profile. For more information, see section *ODVA AC/DC drive profile* on page *77*.

When Reference 1 (REF1) is used for speed control and the parameter 5107 value is 128, an ODVA speed reference value of ±30000 (decimal) corresponds to an equal amount of rpm in the drive. The reference value sent from the PLC is limited by parameter 1105 REF1 MAX in the forward and reverse directions.

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

Bytes	Instance 121	Instance 171
0…1	Control word	Status word
2…3	Speed reference	Speed actual value
4…5	Acceleration time[1]	Power[1]
6…7	Deceleration time[1]	DC bus voltage[1]

[1] Example

The table below gives the recommended drive parameter settings.

Drive parameter	Setting for ACS355 drives	Description
9802 COMM PROT SEL	**4** = EXT FBA	Enables communication between the drive and the fieldbus adapter module.
5101 FBA TYPE	ControlNet[1]	Displays the type of the fieldbus adapter module.
5102 FB PAR 2 (PROTOCOL/PROFILE)	**0** (= ODVA)	Selects the ControlNet protocol and the ODVA AC/DC drive profile.

Drive parameter	Setting for ACS355 drives	Description
5103 FB PAR 3 (MODULE MACID)	2	Selects the MAC ID for the node. Used when the HW switch (5105) is not in use.
5104 FB PAR 4 (MODULE BAUD RATE)	**5** (= 5 Mbit/s)	Baud rate is fixed to 5 Mbit/s.
5105 FB PAR 5 (HW/SW OPTION)	**0** (= HW)[2]	Sets the MAC ID. Set the rotary switch, eg, to 2.
5106 FB PAR 6 (ODVA STOP FUNCTION)	**0** (= RAMP)	Ramp stop
5107 FB PAR 7 (ODVA SPEED SCALE)	128	Sets the scaling for the ODVA speed reference. Speed reference given in rpm.
3018 COMM FAULT FUNC	**1** = FAULT[2]	Enables fieldbus communication fault monitoring.
3019 COMM FAULT TIME	3.0 s[2]	Defines the fieldbus communication break supervision time.
5401 FBA DATA IN 1	106[2]	Power
5402 FBA DATA IN 2	107[2]	DC bus voltage
5501 FBA DATA OUT 1	2202[2]	Acceleration time
5502 FBA DATA OUT 2	2203[2]	Deceleration time
5127 FBA PAR REFRESH	**1** = REFRESH	Validates the FCNA-01 configuration parameter settings.
9904 MOTOR CTRL MODE	**1** = VECTOR: SPEED	Selects the speed control mode as the motor control mode.
1001 EXT1 COMMANDS	**10** = COMM	Selects the fieldbus interface as the source of the start and stop commands for external control location 1.
1103 REF1 SELECT	**8** = COMM	Selects the fieldbus reference 1 as the source for speed reference 1.
1601 RUN ENABLE	**7** = COMM	Selects the fieldbus interface as the source for the inverted Run enable signal (Run disable).

Drive parameter	Setting for ACS355 drives	Description
1604 FAULT RESET SEL	**8** = COMM	Selects the fieldbus interface as the source for the fault reset signal.

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

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

Control word:

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

  or

  2h (2 decimal) –> ENABLED (Running reverse).

Starting up ACSM1 drives

1. Power up the drive.

2. Enable the communication between the adapter module and the drive with parameter 50.01 FBA ENABLE.

3. With parameter 50.02 COMM LOSS FUNC, select how the drive reacts to a fieldbus communication break.

   Note that this function monitors both communication between the fieldbus master and the adapter module and communication between the adapter module and the drive.

4. With parameter 50.03 COMM LOSS T OUT, define the time between communication break detection and the selected action.

5. Select application-specific values for parameters 50.04…50.11.

   Examples of appropriate values are shown in the tables below.

6. Select the profile with parameter 51.02.

7. Set the MacID address either with parameters 51.03 and 51.05 or with the switches.

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

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

9. Validate the settings made in parameter groups 51, 52 and 53 with parameter 51.27 FBA PAR REFRESH.

10. Set the relevant drive control parameters to control the drive according to the application.

    Examples of appropriate values are shown in the tables below.

■ **Parameter setting examples – ACSM1**

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

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

The start/stop commands and reference scaling are according to the ODVA AC/DC drive profile. For more information, see section *ODVA AC/DC drive profile* on page *77*.

When Reference 1 (REF1) is used for speed control and the value of parameter 51.07 is 128, an ODVA speed reference value of ±30000 (decimal) corresponds to an equal amount of rpm in the drive. The speed reference value sent by the PLC is limited by parameter 20.01 MAXIMUM SPEED in the forward direction and 20.02 MINIMUM SPEED in the reverse direction.

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

Bytes	Instance 121	Instance 171
0…1	Control word	Status word
2…3	Speed reference	Speed actual value
4…7	Acceleration time[1]	Power[1]
8…11	Deceleration time[1]	DC bus voltage[1]

[1] Example

The table below gives the recommended drive parameter settings.

Drive parameter	Setting for ACSM1 drives	Description
50.01 FBA ENABLE	Enable	Enables communication between the drive and the fieldbus adapter module.
50.02 COMM LOSS FUNC	Fault[2)]	Enables fieldbus communication fault monitoring.
50.03 COMM LOSS T OUT	3.0 s[2)]	Defines the fieldbus communication break supervision time.
50.04 FBA REF1 MODESEL	Speed	Selects the fieldbus reference 1 scaling.
51.03 FBA PAR 3 (MODULE MACID)	2	Select MAC ID for the node. Used if the HW switch (51.05) is not in use.
5104 FB PAR 4 (MODULE BAUD RATE)	**5** (= 5 Mbit/s)	Baud rate is fixed to 5 Mbit/s.
51.05 FBA PAR 5 (HW/SW OPTION)	**0** (= HW)[2)]	Sets the MAC ID. Set the rotary switch, eg, to 2.
51.06 FB par 6 (ODVA STOP FUNC-TION)	**0** (= RAMP)	Ramp stop
51.07 FB par 7 (ODVA SPEED SCALE)	128	Sets the scaling for the ODVA speed reference. Speed reference is given in rpm.
52.01 FBA DATA IN1	122[2)]	Power
52.03 FBA DATA IN3	107[2)]	DC bus voltage
53.01 FBA DATA OUT1	2503[2)]	Acceleration time
53.03 FBA DATA OUT3	2504[2)]	Deceleration time
51.27 FBA PAR REFRESH	REFRESH	Validates the FCNA-01 configuration parameter settings.
10.01 EXT1 START FUNC	FBA	Selects the fieldbus interface as the source of the start and stop commands for external control location 1.
24.01 SPEED REF1 SEL	FBA REF1	Selects the fieldbus reference 1 as the source for speed reference 1.

Drive parameter	Setting for ACSM1 drives	Description
34.01 EXT1/EXT2 SEL	C.FALSE	Selects that the external control location is always EXT1.
34.03 EXT1 CTRL MODE1	Speed	Selects speed control as the control mode 1 for external control location 1.

[1] Read-only or automatically detected/set

[2] Example

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

Control word:

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

  or

  2h (2 decimal) –> ENABLED (Running reverse).

Starting up ACS850 drives

1. Power up the drive.

2. Enable the communication between the adapter module and the drive with parameter 50.01 FBA enable.

3. With parameter 50.02 Comm loss func, select how the drive reacts to a fieldbus communication break.

   Note that this function monitors both communication between the fieldbus master and the adapter module and communication between the adapter module and the drive.

4. With parameter 50.03 Comm loss t out, define the time between communication break detection and the selected action.

5. Select application-specific values for parameters 50.04…50.11.

   Examples of appropriate values are shown in the tables below.

6. Select the profile with parameter 51.02.

7. Set the Mac ID address either with parameters 51.03 and 51.05 or with the switches.

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

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

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

10. Set the relevant drive control parameters to control the drive according to the application.

    Examples of appropriate values are shown in the tables below.

◼ Parameter setting examples – ACS850

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

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

The start/stop commands and reference scaling are according to the ODVA AC/DC drive profile. For more information, see section *ODVA AC/DC drive profile* on page *77*.

When Reference 1 (REF1) is used for speed control and the value of parameter 51.07 is 128, an ODVA speed reference value of ±30000 (decimal) corresponds to an equal amount of rpm in the drive. The speed reference value sent from the PLC is limited by parameter 20.01 Maximum speed in the forward direction and 20.02 Minimum speed in the reverse direction.

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

Bytes	Instance 121	Instance 171
0…1	Control word	Status word
2…3	Speed reference	Speed actual value
4…7	Acceleration time[1]	Power[1]
8…11	Deceleration time[1]	DC bus voltage[1]

[1] Example

The table below gives the recommended drive parameter settings.

Drive parameter	Setting for ACS850 drives	Description
50.01 Fba enable	Enable	Enables communication between the drive and the fieldbus adapter module.
50.02 Comm loss func	Fault[2]	Enables fieldbus communication fault monitoring.
50.03 Comm loss t out	3.0 s[2]	Defines the fieldbus communication break supervision time.

Drive parameter	Setting for ACS850 drives	Description
50.04 Fb ref1 modesel	Speed	Selects the fieldbus reference 1 scaling.
51.03 FBA PAR 3 (MODULE MACID)	2	Selects the MAC ID for the node. Used if the HW switch (51.05) is not in use.
51.04 FB PAR 4 (MODULE BAUD RATE	**5** (= 5 Mbit/s)	Baud rate is fixed to 5 Mbit/s.
51.05 FBA PAR 5 (HW/SW OPTION)	**0** (= HW)[2]	Sets the MAC ID. Set the rotary switch, eg, to 2
51.06 FB par 6 (ODVA STOP FUNC-TION)	**0** (= RAMP)	Ramp stop.
51.07 FB par 7 (ODVA SPEED SCALE)	128	Sets the scaling for the ODVA speed reference. Speed reference given in rpm.
52.01 FBA data in1	122[2]	Power
52.03 FBA data in3	107[2]	DC bus voltage
53.01 FBA data out1	2202[2]	Acceleration time
53.03 FBA data out3	2203[2]	Deceleration time
51.27 FBA par refresh	Refresh	Validates the FCNA-01 configuration parameter settings.
10.01 Ext1 start func	FB	Selects the fieldbus interface as the source of the start and stop commands for external control location 1.
21.01 Speed ref1 sel (ACS850)	FBA ref1	Selects the fieldbus reference 1 as the source for speed reference 1.

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

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

Control word:

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

  or

  2h (2 decimal) –> ENABLED (Running reverse).

Starting up ACS880 drives

1. Power up the drive.

2. Enable the communication between the adapter module and the drive with parameter 50.01 FBA A Enable.

    The selection must correspond to the slot where the adapter module is installed. For example, if the adapter module is installed in slot 1, select slot 1.

3. With parameter 50.02 FBA A comm loss func, select how the drive reacts to a fieldbus communication break.

    Note that this function monitors both communication between the fieldbus master and the adapter module and communication between the adapter module and the drive.

4. With parameter 50.03 FBA A comm loss t out, define the time between communication break detection and the selected action.

5. Select application-specific values for the rest of the parameters in group 50, starting from 50.04.

    Examples of appropriate values are shown in the tables below.

6. Select the profile with parameter 51.02.

7. Set the Mac ID address either with parameters 51.03 and 51.05 or with the switches.

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

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

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

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

11. Set the relevant drive control parameters to control the drive according to the application.

    Examples of appropriate values are shown in the tables below.

■ Parameter setting examples – ACS880

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

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

The start/stop commands and reference scaling are according to the ODVA AC/DC drive profile. For more information, see section *ODVA AC/DC drive profile* on page *77*.

When Reference 1 (REF1) is used for speed control and the value of parameter 51.07 is 128, an ODVA speed reference value of ±30000 (decimal) corresponds to an equal amount of rpm in the drive. The speed reference value sent from the PLC is limited by parameter 30.12 Maximum speed in the forward direction and 30.11 Minimum speed in the reverse direction.

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

Bytes	Instance 121	Instance 171
0…1	Control word	Status word
2…3	Speed reference	Speed actual value
4…7	Acceleration time[1)	Power[1)
8…11	Deceleration time[1)	DC bus voltage[1)

[1) Example

The table below gives the recommended drive parameter settings.

Drive parameter	Setting for ACS880 drives	Description
50.01 FBA A enable	**1** = Option slot1t[2]	Enables communication between the drive and the fieldbus adapter module.
50.02 FBA A comm loss func	**1** = Fault[2]	Enables fieldbus A communication fault monitoring.
50.03 FBA A comm loss t out	3.0 s[2]	Defines the fieldbus A communication break supervision time.
50.04 FBA A ref1 type	**4** = Speed	Selects the fieldbus A reference 1 type and scaling.
51.03 FBA PAR 3 (MODULE MACID)	2	Selects the MAC ID for the node. Used if the HW switch (51.05) is not in use.
5104 FB PAR 4 (MODULE BAUD RATE)	**5** (= 5 Mbit/s)	Baud rate is fixed to 5 Mbit/s.
51.05 FBA PAR 5 (HW/SW OPTION)	**0** (= HW)[2]	Sets the MAC ID. Set the rotary switch, eg, to 2.
51.06 FB par 6 (ODVA STOP FUNC-TION)	**0** (= RAMP)	Ramp stop.
51.07 FB par 7 (ODVA SPEED SCALE)	128	Sets the scaling for the ODVA speed reference. Speed reference given in rpm.
52.01 FBA A data in1	P.1.14[2]	Output power
52.03 FBA A data in3	P.1.11[2]	DC voltage
53.01 FBA A data out1	P.23.12[2]	Acc time 1
53.03 FBA A data out3	P.23.13[2]	Dec time 1
51.27 FBA A par refresh	**1** = Refresh	Validates the FCNA-01 configuration parameter settings.
20.01 Ext1 commands	**12** = Fieldbus A	Selects the fieldbus A interface as the source of the start and stop commands for external control location 1.
22.11 Speed ref1 source	**4** = FB A ref1	Selects the fieldbus A reference 1 as the source for speed reference 1.

Drive parameter	Setting for ACS880 drives	Description

[1] Read-only or automatically detected/set

[2] Example

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

Control word:

- Reset the fieldbus communication fault (if active).

- Enter 0h (0 decimal) –> READY.

- Enter 1h (1 decimal) –> ENABLED (Running forward)

    or

    2h (2 decimal) –> ENABLED (Running reverse).

Configuring the client

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

The example can be applied to all drive types compatible with the module.

▪ Before you start

Decide on the following points before starting the client configuration.

Select protocol/profile

During the configuration of the drive and the client, it is necessary to select a communication protocol, in this case ControlNet, and a communication profile. The communication profile determines what I/O assemblies and objects are available. See chapter *Communication profiles* for more information.

Select output and input assembly instances

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

Name	Output instance	Input instance	Size (bytes)	Profile
Basic speed control	20	70	4	ODVA AC/DC drive
Enhanced speed control	21	71	4	ODVA AC/DC drive
Basic speed and torque control	22	72	6	ODVA AC/DC drive
Enhanced speed and torque control	23	73	6	ODVA AC/DC drive

Name	Output instance	Input instance	Size (bytes)	Profile
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
User specific drive parameters assembly up to 32 words	124	153	20	ODVA AC/DC drive
ABB Drives profile w/ set speed	1	51	4	ABB Drives profile
ABB Drives profile w/ set speed and set torque	2	52	6	ABB Drives profile
ABB Drives profile w/ set speed plus drive parameters	101	151	24	ABB Drives profile
ABB Drives profile w/ set speed and set torque plus drive parameters	102	152	26	ABB Drives profile
User specific drive parameters assembly up to 32 words	103	174	20	ABB Drives profile
Transparent 16 with one assembly	11	61	4	Transparent16 profile
Transparent 16 with two assembly	12	62	6	Transparent16 profile
Transparent 16 with one plus drive parameters	111	161	24	Transparent16 profile
Transparent 16 with two plus drive parameters	112	162	26	Transparent16 profile
Transparent 32 with one assembly	21	71	8	Transparent32 profile
Transparent 32 with two assembly	22	72	12	Transparent32 profile

Name	Output instance	Input instance	Size (bytes)	Profile
Transparent 32 with one plus drive parameters	121	171	28	Transparent32 profile
Transparent 32 with two plus drive parameters	122	172	32	Transparent32 profile

Select connection method

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

Note: The choice of the connection method has a significant impact on the time-out behavior. Refer to configuration parameters *10 CONTROL TIMEOUT* and *11 IDLE ACTION* for more information.

The adapter module supports the following connection methods:

I/O connections

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

Connected explicit messaging

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

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

Unconnected explicit messaging

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

Note: ControlNet does not provide a time-out means for unconnected explicit messaging. To use unconnected explicit messaging for control, refer to configuration parameter *10 CONTROL TIMEOUT*.

■ EDS files

Electronic data sheet (EDS) files specify the properties of the device for the ControlNet client. The client 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 *134*.

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

EDS files are available from the Document library (www.abb.com/drives).

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

■ **Standard ABB drive on ControlNet^{TM} (FCNA-01) with RSLogix^{TM} 5000 and RSNetWorx^{TM}**

The following procedure shows how to set up a standard ABB drive with a FCNA-01 adapter module in RSLogix 5000 and RSNetWorx. The drive can be configured to a vendor-specific assembly (ABB Profile) or an ODVA-specific assembly (AC/DC Profile). The FCNA-01 and the PLC will transmit/receive up to thirteen words.

Configuring the FCNA-01

1. Open RSLogix 5000 and open or create a RSLogix 5000 program.

2. Right-click on **1756-CNB ControlNet** and select **New Module**.
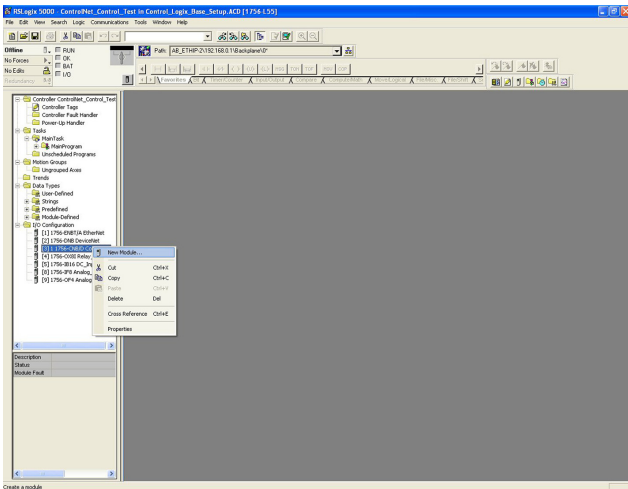
3.   Select **Generic ControlNet Module**.

Select Module Type

Type: CONTROLNET-MODULE

Type	Description
2364F RGU-CN1	2364F Regen Bus Supply via 1203-CN1
2711P-RN15S/A	ControlNet Module for PanelView Plus 700-1500/VersaView CE
CONTROLNET-MODULE	Generic ControlNet Module
FlexPak 3000	FlexPak 3000 DC Drive
GV3000	GV3000 AC Drive
PanelView	2711 PanelView Operator Terminal
PowerFlex 700 Vector-2l ...	PowerFlex 700 Vector Drive (208/240V) via 20-COMM-C
PowerFlex 700 Vector-4l ...	PowerFlex 700 Vector Drive (400/480V) via 20-COMM-C
PowerFlex 700 Vector-6l ...	PowerFlex 700 Vector Drive (600V) via 20-COMM-C
PowerFlex 700-200V-C	PowerFlex 700 (200V) Drive via 20-COMM-C
PowerFlex 700-400V-C	PowerFlex 700 Drive (400/480V) via 20-COMM-C
PowerFlex 700-600V-C	PowerFlex 700 Drive (600V) via 20-COMM-C
PowerFlex 700S-200V-C	PowerFlex 700S Drive (208/240V) via 20-COMM-C
PowerFlex 700S-400V-C	PowerFlex 700S Drive (400/480V) via 20-COMM-C

PowerFlex 700S Drive (208/240V) via 20-COMM-C

Show

Vendor: All

☑ Other ☑ Specialty I/O   Select All

☑ Analog ☑ Digital ☑ Communication ☑ Motion ☑ Controller   Clear All

OK   Cancel   Help

4. Do the tasks listed in the table below to provide the New Module window with the information required by the 1756-CNB ConrtolNet scanner.

The example below is using the ABB Profile 102 and 152.

1	Enter the name for the FCNA-01.
2	As the FCNA-01 uses 16 Bit words, change **Comm Format** to **Data-INT**.
3	Enter the node address of the FCNA-01.
4	Enter the input/output assembly instances.
5	See the next step on page *69* for the information on the size settings.
6	Set Configuration to 1 and size to 0.

5. Select the input and output word sizes from the table below.

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

For more information on the input/output assembly instances, see chapter *Communication protocol*.

6. Click **OK**.

The FCNA-01 is now added to the 1756-CNB ConrtolNet scanner.

7. Download the program to the PLC and make sure the PLC is in the program mode.
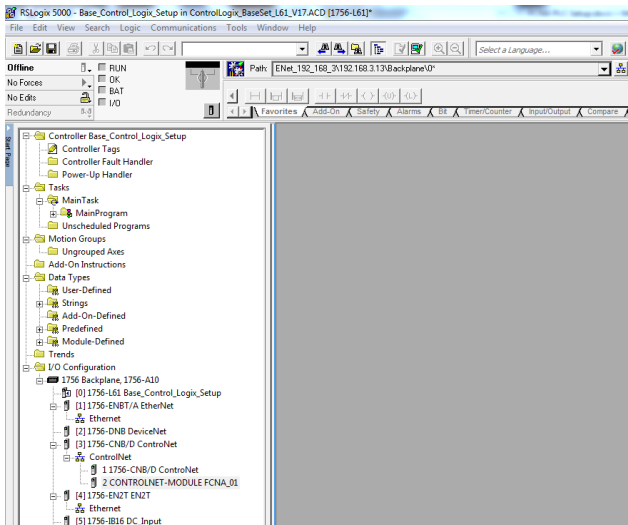
8. Click **OK**.

9. Open RSNetWorx for ControlNet.

10. Go online and scan the ControlNet network that needs to be configured.
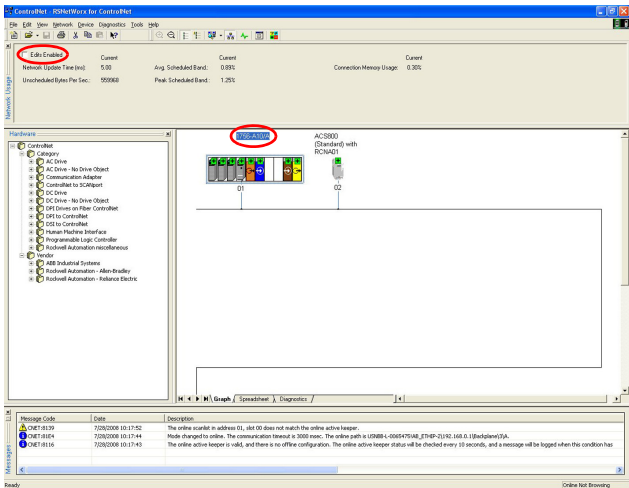
11. Click **1756-xxxx** and select **Edits Enabled**.

See the figure above.

12. Click **OK**.
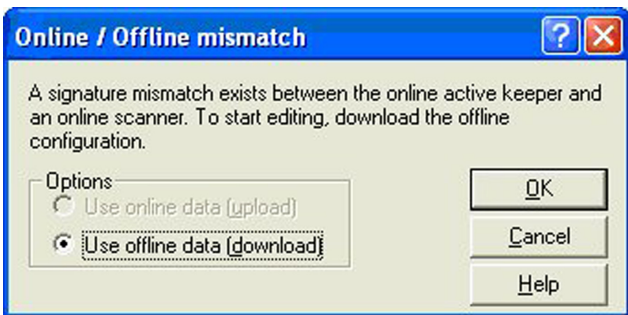
13. Click **Network –> Properties**.

14. Check that he Max Scheduled Address is equal to or greater than the last address node on the ControlNet network.

Click **OK**.
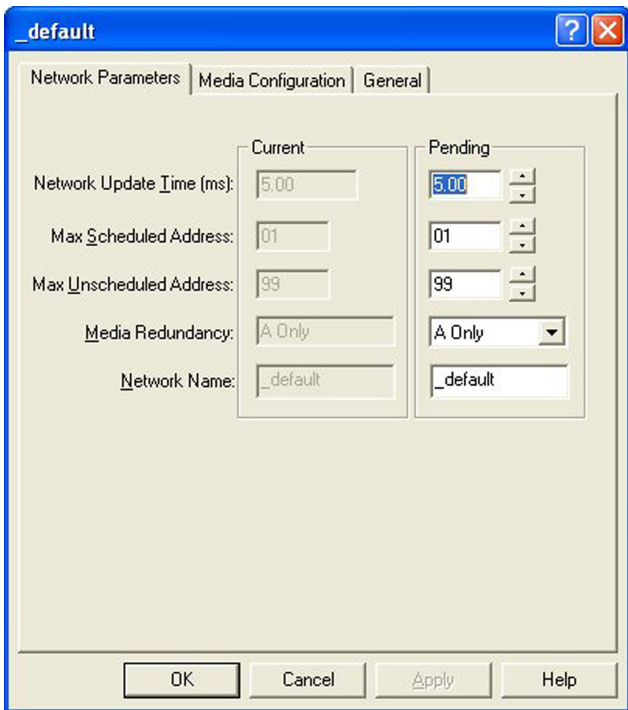
15. Click **File –> Save As** and name the file.

RSNetWorx asks if the changes should be downloaded to the keeper.

16. Click **OK**.

The FCNA-01 is now a scheduled connection in the ControlNet network.

7

Communication profiles

Contents of this chapter

This chapter describes the communication profiles used in the communication between the ControlNet network, the adapter module and the drive.

Communication profiles

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

With the FCNA-01 adapter module, the ControlNet network may employ either the ODVA AC/DC drive profile or the ABB Drives profile. Both are converted to the native profile (eg, DCU or FBA) by the adapter module. In addition, two Transparent modes – for 16-bit and 32-bit words respectively – are available. With the Transparent modes, no data conversion takes place.

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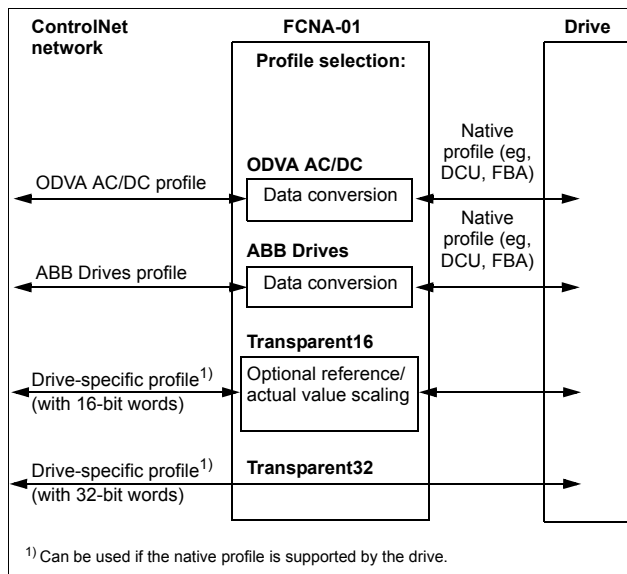
The following sections describe the Control word, the Status word, references and actual values for the ODVA AC/DC drive and ABB Drives communication profiles. Refer to the drive manuals for details on the native profiles.

ODVA AC/DC drive profile

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

A ControlNet node is modeled as a collection of abstract objects. Each object represents the interface to and behavior of a component within the product. The ODVA AC/DC drive profile defines a collection of objects suitable for the control of AC and DC drives. The objects supported by the adapter module are listed in section *Class objects* on page *133*.

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 ControlNet client to set or get predefined groups of attributes in a single message exchange. The assembly instances supported by the adapter module are listed and defined in section *Assembly objects* on page *98*.

■ **ODVA output attributes**

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

**Run Forward & Run Reverse
(Control supervisor object)**

These attributes are used to assert run and stop commands to the Control supervisor object state machine according to the following Run/Stop event matrix. See *State (Control supervisor object)* on page *83*.

RunFwd	RunRev	Trigger event	Run type
0	0	Stop	N/A
0 → 1	0	Run	RunFwd
0	0 → 1	Run	RunRev
0 → 1	0 → 1	No Action	N/A
1	1	No Action	N/A
0 → 1	1	Run	RunRev
1	1 → 0	Run	RunFwd

Fault Reset (Control supervisor object)

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

Net Ctrl (Control supervisor object)

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

Net Ref (AC/DC drive object)

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

Speed Reference (AC/DC drive object)

This attribute is the speed reference for the drive. The units are scaled by the Speed Scale attribute of the AC/DC drive object. See parameter *07 ODVA SPEED SCALE* for details.

Scalar mode

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

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

where

Dfr = Drive Frequency Reference in Hz
Osr = ODVA Speed Reference
Us   = ODVA Speed Unit (see *07 ODVA SPEED SCALE* on page *35*)
Mf   = Motor Nominal Frequency in Hz
Mss = Motor Synchronous Speed in rpm (not Motor Nominal Speed).

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

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

Vector mode

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

$$Dsr = Osr \times Us$$

where

Dsr = Drive Speed Reference in rpm
Osr = ODVA Speed Reference
Us   = ODVA Speed Unit (see *07 ODVA SPEED SCALE* on page *35*).

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

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

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 parameter *08 ODVA TORQUE SCALE* for details.

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

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

where

Dtr   = Drive Torque Reference in Percent of Motor Nominal Torque
Otr   = ODVA Torque Reference
Ut   = ODVA Torque Unit (see *08 ODVA TORQUE SCALE* on page *36*)
Mt   = Motor Nominal Torque in N·m.

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

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

Energy objects

The FCNA-01 supports the following energy objects:
- Base energy object
- Electrical energy object
- Non-electrical energy object.

The Electrical energy object can be measured and reported as produced and consumed energy types, in real energy generated and consumed in kilowatt hours, average current, phase to phase voltages, and phase rotation. The Non-electrical energy object can be measured and reported as net energy, in specified types and specified units, using a normalization multiplier or divisor.

■ **ODVA input attributes**

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

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 *83*.

Ctrl From Net (Control supervisor object)

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

Ref From Net (AC/DC drive object)

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

At Reference (AC/DC drive object)

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

State (Control supervisor object)

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

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

The ODVA state transition diagram is shown below:
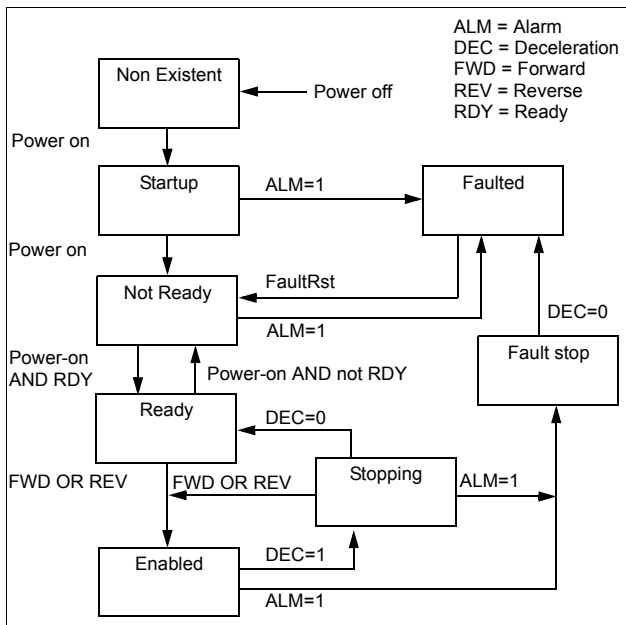
Speed Actual (AC/DC drive object)

This attribute indicates the actual speed at which the drive is operating. The units are scaled by the SpeedScale attribute of the AC/DC drive object. See parameter *07 ODVA SPEED SCALE* for details.

Scalar mode

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

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

where

Osa = ODVA Speed Actual
Dfa = Drive Frequency Actual in Hz
Us = ODVA Speed Unit (see *07 ODVA SPEED SCALE* on page *35*)
Mf = Motor Nominal Frequency in Hz
Mss = Motor Synchronous Speed in rpm (not Motor Nominal Speed).

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

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

Vector mode

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

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

where

Dsa = Drive Speed Actual in rpm
Osa = ODVA Speed Actual
Us   = ODVA Speed Unit (see *07 ODVA SPEED SCALE* on page *35*).

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

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

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 parameter *08 ODVA TORQUE SCALE* for details.

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

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

where

Dta = Drive Torque Actual in Percent of Motor Nominal Torque
Ota = ODVA Torque Actual
Ut = ODVA Torque Unit (see *08 ODVA TORQUE SCALE* on page *36*)
Mt = Motor Nominal Torque in N·m.

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

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

ABB Drives communication profile

■ Control word and Status word

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

The contents of the Control word and the Status word are detailed below. The drive states are presented on page *93*.

Control word contents

The table below shows the contents of the Control word for the ABB Drives communication profile. The upper case boldface text refers to the states shown in the state machine on page *93*.

Bit	Name	Value	STATE/Description
0	OFF1_CONTROL	1	Proceed to **READY TO OPERATE**.
		0	Stop along currently active deceleration ramp. Proceed to **OFF1 ACTIVE**; proceed to **READY TO SWITCH ON** unless other interlocks (OFF2, OFF3) are active.
1	OFF2_CONTROL	1	Continue operation (OFF2 inactive).
		0	Emergency OFF, coast to stop. Proceed to **OFF2 ACTIVE**, proceed to **SWITCH-ON INHIBITED**.
2	OFF3_CONTROL	1	Continue operation (OFF3 inactive).
		0	Emergency stop, stop within time defined by drive parameter. Proceed to **OFF3 ACTIVE**; proceed to **SWITCH-ON INHIBITED**. **Warning:** Ensure motor and driven machine can be stopped using this stop mode.

Bit	Name	Value	STATE/Description
3	INHIBIT_OPERATION	1	Proceed to **OPERATION ENABLED**. **Note:** Run enable signal must be active; see drive documentation. If the drive is set to receive the Run enable signal from the fieldbus, this bit activates the signal.
		0	Inhibit operation. Proceed to **OPERATION INHIBITED**.
4	RAMP_OUT_ZERO	1	Normal operation. Proceed to **RAMP FUNCTION GENERATOR: OUTPUT ENABLED**.
		0	Force Ramp Function Generator output to zero. Drive ramps to stop (current and DC voltage limits in force).
5	RAMP_HOLD	1	Enable ramp function. Proceed to **RAMP FUNCTION GENERATOR: ACCELERATOR ENABLED**.
		0	Halt ramping (Ramp Function Generator output held).
6	RAMP_IN_ZERO	1	Normal operation. Proceed to **OPERATION**. **Note:** This bit is effective only if the fieldbus interface is set as the source for this signal by drive parameters.
		0	Force Ramp Function Generator input to zero.
7	RESET	0 –> 1	Fault reset if an active fault exists. Proceed to **SWITCH-ON INHIBITED**. **Note:** This bit is effective only if the fieldbus interface is set as the source for this signal by drive parameters.
		0	Continue normal operation.
8…9	Reserved		

Bit	Name	Value	STATE/Description
10	REMOTE_ CMD	1	Fieldbus control enabled.
		0	Control word and reference not getting through to the drive, except for CW bits OFF1, OFF2 and OFF3.
11	EXT_CTRL_ LOC	1	Select External Control Location EXT2. Effective if control location parameterized to be selected from fieldbus.
		0	Select External Control Location EXT1. Effective if control location parameterized to be selected from fieldbus.
12… 15	Reserved		

Status word contents

The table below shows the contents of the Status word for the ABB Drives communication profile. The upper case boldface text refers to the states shown in the state machine on page *93*.

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	**OPERATION.** Actual value equals reference (= is within tolerance limits, ie, in speed control, speed error is 10% max. of nominal motor speed).
		0	Actual value differs from reference (= is outside tolerance limits).
9	REMOTE	1	Drive control location: REMOTE (EXT1 or EXT2)
		0	Drive control location: LOCAL

Bit	Name	Value	STATE/Description
10	ABOVE_ LIMIT	1	Actual frequency or speed equals or exceeds supervision limit (set by drive parameter). Valid in both directions of rotation.
		0	Actual frequency or speed within supervision limit
11	EXT_CTRL_ LOC	1	External Control Location EXT2 selected. **Note concerning ACS880:** This bit is effective only if the fieldbus interface is set as the target for this signal by drive parameters. User bit 0 selection (06.33).
		0	External Control Location EXT1 selected
12	EXT_RUN_ ENABLE	1	External Run Enable signal received. **Note concerning ACS880:** This bit is effective only if the fieldbus interface is set as the target for this signal by drive parameters. User bit 1 selection (06.34).
		0	No External Run Enable signal received
13… 14	Reserved		
15	FBA_ERROR	1	Communication error detected by fieldbus adapter module
		0	Fieldbus adapter communication OK

State machine

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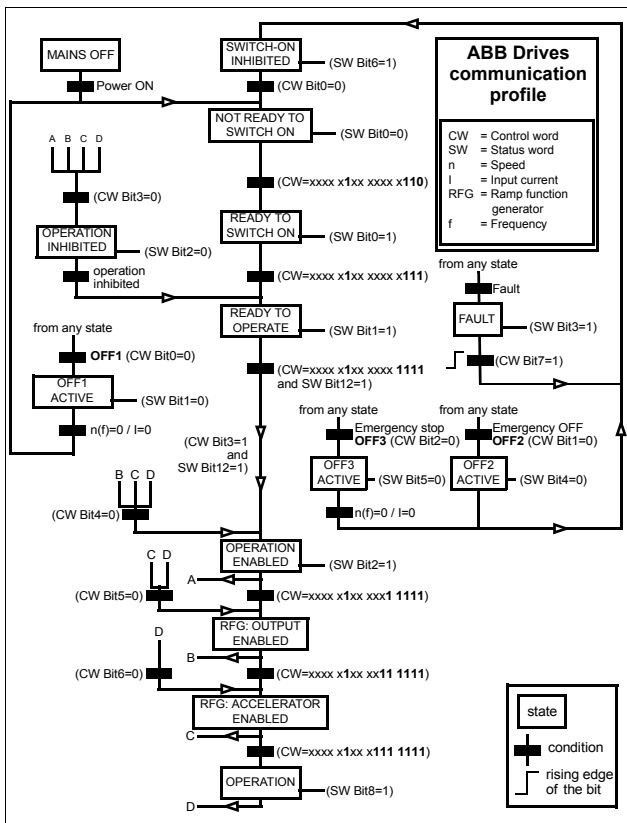
■ **References**

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

ABB drives can receive control information from multiple sources including analog and digital inputs, the drive control panel and a communication module (for example, the FCNA-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 manuals for further information.

In ACSM1, ACS850 and ACS880, the speed reference (REFx) in decimal (0…20000) corresponds to 0…100% of the speed scaling value (as defined with a drive parameter, eg, ACS880 parameter 46.10 Speed scaling.)

In ACS355, drive parameter REFx MIN may limit the actual minimum reference.
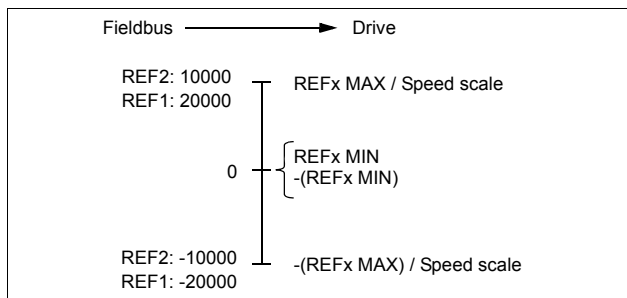
■ Actual values

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

Scaling

Actual values are scaled as shown below.

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

8
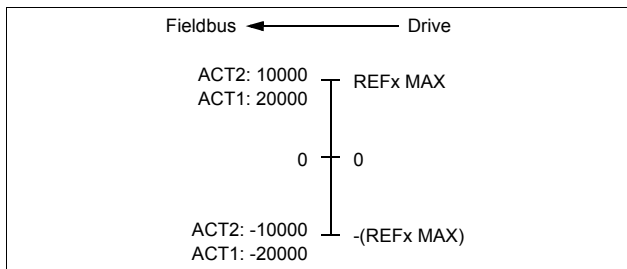
Communication protocol

Contents of this chapter

This chapter describes the ControlNet communication protocol for the adapter module.

ControlNet

The interface from the fieldbus towards the FCNA-01 is based on the standard ControlNet objects and vendor-specific objects.

The FCNA-01 has the following objects included:

Object name	Class
Identity	0x01
Message router	0x02
Assembly	0x04
Connection manager	0x06
Motor data	0x28
Control supervisor	0x29
AC/DC drive	0x2A
Drive parameter	0x90
Fieldbus configuration	0x91
ControlNet	0xF0

Object name	Class
Keeper	0xF1
Connection configuration	0xF3
Port	0xF4
Base energy	4E
Electrical energy	4F
Non-electrical energy	50

Assembly objects

I/O assembly instances may also be referred to as Block transfer of data. Intelligent devices realizing a Functional profile, such as FCNA-01, have several objects. Since it is not possible to transmit more than one object data through a single connection, it is practical and more efficient to group attributes from different objects into a single I/O connection using the assembly object. The assembly object acts as a tool for grouping these attributes.

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

The following tables describe the assembly instances supported by the adapter module.

■ Basic speed control assembly

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

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

The format of the input assembly is:

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

■ Basic speed control plus drive parameters assembly

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

The format of the output assembly is:

Instance 120 (ODVA AC/DC profile)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0						Fault Reset		Run Fwd
1								
2	Speed 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)							

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

The format of the input assembly is:

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

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

■ Extended speed control assembly

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

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

The format of the input assembly is:

Instance 71 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	At Refer-ence	Ref From Net	Ctrl From Net	Ready	Run-ning2 (Rev)	Run-ning1 (Fwd)	Warn-ing	Faulted
1	Drive State (See section *State (Control supervisor object)* on page *83*.)							
2	Speed Actual (Low Byte)							
3	Speed Actual (High Byte)							

■ Extended speed control plus drive parameters assembly

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

The format of the output assembly is:

Instance 121 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0		NetRef	NetCtrl			Fault Reset	Run Rev	Run Fwd
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)							

Instance 121 (ODVA AC/DC profile)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
11	DATA OUT 4 Value (High Byte)							
12	DATA OUT 5 Value (Low Byte)							
13	DATA OUT 5 Value (High Byte)							
14	DATA OUT 6 Value (Low Byte)							
15	DATA OUT 6 Value (High Byte)							
16	DATA OUT 7 Value (Low Byte)							
17	DATA OUT 7 Value (High Byte)							
18	DATA OUT 8 Value (Low Byte)							
19	DATA OUT 8 Value (High Byte)							
20	DATA OUT 9 Value (Low Byte)							
21	DATA OUT 9 Value (High Byte)							
22	DATA OUT 10 Value (Low Byte)							
23	DATA OUT 10 Value (High Byte)							

The format of the input assembly is:

Instance 171 (ODVA AC/DC profile)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	At Refer-ence	Ref From Net	Ctrl From Net	Ready	Run-ning2 (Rev)	Run-ning1 (Fwd)	Warn-ing	Faulted
1	Drive State (See section *State (Control supervisor object)* on page *83*.)							
2	Speed Actual (Low Byte)							
3	Speed Actual (High Byte)							
4	DATA IN 1 Value (Low Byte)							
5	DATA IN 1 Value (High Byte)							
6	DATA IN 2 Value (Low Byte)							
7	DATA IN 2 Value (High Byte)							
8	DATA IN 3 Value (Low Byte)							

Instance 171 (ODVA AC/DC profile)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
9	DATA IN 3 Value (High Byte)							
10	DATA IN 4 Value (Low Byte)							
11	DATA IN 4 Value (High Byte)							
12	DATA IN 5 Value (Low Byte)							
13	DATA IN 5 Value (High Byte)							
14	DATA IN 6 Value (Low Byte)							
15	DATA IN 6 Value (High Byte)							
16	DATA IN 7 Value (Low Byte)							
17	DATA IN 7 Value (High Byte)							
18	DATA IN 8 Value (Low Byte)							
19	DATA IN 8 Value (High Byte)							
20	DATA IN 9 Value (Low Byte)							
21	DATA IN 9 Value (High Byte)							
22	DATA IN 10 Value (Low Byte)							
23	DATA IN 10 Value (High Byte)							

■ **Basic speed and torque control assembly**

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

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

The format of the input assembly is:

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

■ **Basic speed and torque control plus drive parameters assembly**

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

The format of the output assembly is:

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

Instance 122 (ODVA AC/DC profile)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
21	DATA OUT 8 Value (High Byte)							
22	DATA OUT 9 Value (Low Byte)							
23	DATA OUT 9 Value (High Byte)							
24	DATA OUT 10 Value (Low Byte)							
25	DATA OUT 10 Value (High Byte)							

The format of the input assembly is:

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

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

■ Extended speed and torque control assembly

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

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

The format of the input assembly is:

Instance 73 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	At Reference	Ref From Net	Ctrl From Net	Ready	Running2 (Rev)	Running1 (Fwd)	Warning	Faulted
1	Drive State (See section *State (Control supervisor object)* on page *83*.)							
2	Speed Actual (Low Byte)							
3	Speed Actual (High Byte)							
4	Torque Actual (Low Byte)							
5	Torque Actual (High Byte)							

■ Extended speed and torque control plus drive parameters assembly

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

The format of the output assembly is:

Instance 123 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0		NetRef	NetCtrl			Fault Reset	Run Rev	Run Fwd
1								
2	Speed Reference (Low Byte)							
3	Speed Reference (High Byte)							
4	Torque Reference (Low Byte)							
5	Torque Reference (High Byte)							
6	DATA OUT 1 Value (Low Byte)							
7	DATA OUT 1 Value (High Byte)							
8	DATA OUT 2 Value (Low Byte)							

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

The format of the input assembly is:

Instance 173 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	At Refer- ence	Ref From Net	Ctrl From Net	Ready	Run- ning2 (Rev)	Run- ning1 (Fwd)	Warn- ing	Faulted
1	Drive State (See section *State (Control supervisor object)* on page *83*.)							
2	Speed Actual (Low Byte)							
3	Speed Actual (High Byte)							
4	Torque Actual (Low Byte)							

Instance 173 (ODVA AC/DC profile)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
5	Torque Actual (High Byte)							
6	DATA IN 1 Value (Low Byte)							
7	DATA IN 1 Value (High Byte)							
8	DATA IN 2 Value (Low Byte)							
9	DATA IN 2 Value (High Byte)							
10	DATA IN 3 Value (Low Byte)							
11	DATA IN 3 Value (High Byte)							
12	DATA IN 4 Value (Low Byte)							
13	DATA IN 4 Value (High Byte)							
14	DATA IN 5 Value (Low Byte)							
15	DATA IN 5 Value (High Byte)							
16	DATA IN 6 Value (Low Byte)							
17	DATA IN 6 Value (High Byte)							
18	DATA IN 7 Value (Low Byte)							
19	DATA IN 7 Value (High Byte)							
20	DATA IN 8 Value (Low Byte)							
21	DATA IN 8 Value (High Byte)							
22	DATA IN 9 Value (Low Byte)							
23	DATA IN 9 Value (High Byte)							
24	DATA IN 10 Value (Low Byte)							
25	DATA IN 10 Value (High Byte)							

■ **ABB Drives profile with set speed assembly**

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

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

The format of the input assembly is:

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

■ **ABB Drives profile with set speed plus drive parameters assembly**

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

The format of the output assembly is:

Instance 101 (ABB Drives profile)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Reset	Ramp in Zero	Ramp Hold	Ramp Out Zero	Inhibit Oper- ation	Off 3 Control	Off 2 Con- trol	Off 1 Con- trol
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)							

Instance 101 (ABB Drives profile)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
19	DATA OUT 8 Value (High Byte)							
20	DATA OUT 9 Value (Low Byte)							
21	DATA OUT 9 Value (High Byte)							
22	DATA OUT 10 Value (Low Byte)							
23	DATA OUT 10 Value (High Byte)							

The format of the input assembly is:

Instance 151 (ABB Drives profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Alarm	Swc On Inhib	Off 3 Sta	Off 2 Sta	Tripped	Rdy Ref	Rdy Run	Rdy On
1	Field-bus Error			Ext Run Enable	Ext Ctrl Loc	Above Limit	Remote	At Set-point
2	Actual Speed (Low Byte)							
3	Actual Speed (High Byte)							
4	DATA IN 1 Value (Low Byte)							
5	DATA IN 1 Value (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)							

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

■ ABB Drives profile with set speed and set torque assembly

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

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

The format of the input assembly is:

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

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

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

The format of the output assembly is:

Instance 102 (ABB Drives profile)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Reset	Ramp in Zero	Ramp Hold	Ramp Out Zero	Inhibit Oper-ation	Off 3 Con-trol	Off 2 Con-trol	Off 1 Con-trol
1					Ext Ctrl Loc	Re-mote Cmd		
2	Set Speed (Low Byte)							
3	Set Speed (High Byte)							
4	Set Torque (Low Byte)							
5	Set Torque (High Byte)							

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

The format of the input assembly is:

Instance 152 (ABB Drives profile)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Alarm	Swc On Inhib	Off 3 Sta	Off 2 Sta	Tripped	Rdy Ref	Rdy Run	Rdy On
1	Field-bus Error			Ext Run Enable	Ext Ctrl Loc	Above Limit	Remote	At Set-point
2	Actual Speed (Low Byte)							
3	Actual Speed (High Byte)							
4	Actual Torque (Low Byte)							
5	Actual Torque (High Byte)							
6	DATA IN 1 Value (Low Byte)							
7	DATA IN 1 Value (High Byte)							
8	DATA IN 2 Value (Low Byte)							
9	DATA IN 2 Value (High Byte)							
10	DATA IN 3 Value (Low Byte)							
11	DATA IN 3 Value (High Byte)							
12	DATA IN 4 Value (Low Byte)							
13	DATA IN 4 Value (High Byte)							
14	DATA IN 5 Value (Low Byte)							
15	DATA IN 5 Value (High Byte)							
16	DATA IN 6 Value (Low Byte)							
17	DATA IN 6 Value (High Byte)							
18	DATA IN 7 Value (Low Byte)							
19	DATA IN 7 Value (High Byte)							
20	DATA IN 8 Value (Low Byte)							
21	DATA IN 8 Value (High Byte)							
22	DATA IN 9 Value (Low Byte)							
23	DATA IN 9 Value (High Byte)							

Instance 152 (ABB Drives profile)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
24	DATA IN 10 Value (Low Byte)						
25	DATA IN 10 Value (High Byte)						

■ **Transparent 16 with one assembly**

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

The format of the output assembly is:

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

The format of the input assembly is:

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

■ **Transparent 16 with one assembly plus drive parameters**

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

The format of the output assembly is:

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

Instance 111 (Transparent 16 profile)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
22	DATA OUT 10 Value (Low Byte)							
23	DATA OUT 10 Value (High Byte)							

The format of the input assembly is:

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

Instance 161 (Transparent 16 profile)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
22	DATA IN 10 Value (Low Byte)							
23	DATA IN 10 Value (High Byte)							

■ Transparent 16 with two assembly

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

The format of the output assembly is:

Instance 12 (Transparent 16 profile)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Drive Profile 16-bit Control Word (Low Byte)							
1	Drive Profile 16-bit Control Word (High Byte)							
2	Drive Profile 16-bit Reference 1 Word (Low Byte)							
3	Drive Profile 16-bit Reference 1 Word (High Byte)							
4	Drive Profile16-bit Reference 2 Word (Low Byte)							
5	Drive Profile 16-bit Reference 2 Word (High Byte)							

The format of the input assembly is:

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

■ **Transparent 16 with two assembly plus drive parameters**

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

The format of the output assembly is:

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

Instance 112 (Transparent 16 profile)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
22	DATA OUT 9 Value (Low Byte)							
23	DATA OUT 9 Value (High Byte)							
24	DATA OUT 10 Value (Low Byte)							
25	DATA OUT 10 Value (High Byte)							

The format of the input assembly is:

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

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

■ Transparent 32 with one assembly

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

The format of the output assembly is:

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

The format of the input assembly is:

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

■ Transparent 32 with one assembly plus drive parameters

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

The format of the output assembly is:

Instance 121 (Transparent 32 profile)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Drive Profile 32-bit Control Word (Low Byte)							
1	Drive Profile 32-bit Control Word							
2	Drive Profile 32-bit Control Word							
3	Drive Profile 32-bit Control Word (High Byte)							
4	Drive Profile 32-bit Reference 1 Word (Low Byte)							
5	Drive Profile 32-bit Reference 1 Word							
6	Drive Profile 32-bit Reference 1 Word							
7	Drive Profile 32-bit Reference 1 Word (High Byte)							
8	DATA OUT 1 Value (Low Byte)							
9	DATA OUT 1 Value (High Byte)							

Instance 121 (Transparent 32 profile)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
10	DATA OUT 2 Value (Low Byte)							
11	DATA OUT 2 Value (High Byte)							
12	DATA OUT 3 Value (Low Byte)							
13	DATA OUT 3 Value (High Byte)							
14	DATA OUT 4 Value (Low Byte)							
15	DATA OUT 4 Value (High Byte)							
16	DATA OUT 5 Value (Low Byte)							
17	DATA OUT 5 Value (High Byte)							
18	DATA OUT 6 Value (Low Byte)							
19	DATA OUT 6 Value (High Byte)							
20	DATA OUT 7 Value (Low Byte)							
21	DATA OUT 7 Value (High Byte)							
22	DATA OUT 8 Value (Low Byte)							
23	DATA OUT 8 Value (High Byte)							
24	DATA OUT 9 Value (Low Byte)							
25	DATA OUT 9 Value (High Byte)							
26	DATA OUT 10 Value (Low Byte)							
27	DATA OUT 10 Value (High Byte)							

The format of the input assembly is:

Instance 171 (Transparent 32 profile)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Drive Profile 32-bit Status Word (Low Byte)							
1	Drive Profile 32-bit Status Word							
2	Drive Profile 32-bit Status Word							
3	Drive Profile 32-bit Status Word (High Byte)							
4	Drive Profile 32-bit Actual 1 Word (Low Byte)							
5	Drive Profile 32-bit Actual 1 Word (High Byte)							

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

■ **Transparent 32 with two assembly**

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

The format of the output assembly is:

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

The format of the input assembly is:

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

Instance 72 (Transparent 32 profile)	
8	Drive Profile 32-bit Actual 2 Word (Low Byte)
9	Drive Profile 32-bit Actual 2 Word
10	Drive Profile 32-bit Actual 2 Word
11	Drive Profile 32-bit Actual 2 Word (High Byte)

■ Transparent 32 with two assembly plus drive parameters

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

The format of the output assembly is:

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

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

The format of the input assembly is:

Instance 172 (Transparent 32 profile)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Drive Profile 32-bit Status Word (Low Byte)							
1	Drive Profile 32-bit Status Word							
2	Drive Profile 32-bit Status Word							
3	Drive Profile 32-bit Status Word (High Byte)							
4	Drive Profile 32-bit Actual 1 Word (Low Byte)							
5	Drive Profile 32-bit Actual 1 Word							
6	Drive Profile 32-bit Actual 1 Word							
7	Drive Profile 32-bit Actual 1 Word (High Byte)							

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

Class objects

The following ControlNet and CIP objects are supported.

Object name	Object
Identity	0x01
Message router	0x02
Assembly	0x04
Connection manager	0x06
Motor data	0x28
Control supervisor	0x29
AC/DC drive	0x2A
Drive parameter	0x90
Fieldbus configuration	0x91
ControlNet	0xF0
Keeper	0xF1
Connection configuration	0xF3
Port	0xF4
Base energy	0x4E
Electrical energy	0x4F
Non-electrical energy	0x50

Legend	Data type
BOOL	Boolean value
INT8	8-bit signed integer
Padded EPATH	Array of bytes, exact type depends on the place of use
SHORT_STRING	Short N byte character string
SINT16	16-bit signed integer
Struct of xx	Structure of an xx type
UINT8	8-bit unsigned integer
UINT16	16-bit unsigned integer

Legend	Data type
UINT32	32-bit unsigned integer

■ Identity object, class 01h

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

Class attributes

Supported services:
- 0x01 – Get attribute all
- 0x0e – Get attribute single

#	Attribute name	Services	Range, default	Description	Type
1	Revision	Get	1…1, 1	Revision of the identity object	UINT16
2	Max instance	Get	1…1, 1	Max instance	UINT16
6	Max class attribute	Get	7…7, 7	Max ID number of class attribute	UINT16
7	Max instance attribute	Get	10…10, 10	Max ID number of instance attribute	UINT16

Instance attributes

Supported services:
- 0x01 – Get attribute all
- 0x05 – Reset
- 0x0e – Get attribute single

#	Attribute name	Services	Range, default	Description	Type
1	Vendor Id	Get	46…46, 46	Identification of the device vendor	UINT16
			46	ABB Drives Oy	

#	Attribute name	Services	Range, default	Description	Type
2	Device Type	Get	xx…xx, 0x02	Identification of the general product type	UINT16
			0x02	ODVA AC Drive	
			0x13	ODVA DC Drive	
			0x64	Vendor Specific AC Drive	
				ABB Drives profile	
				Transparent 16	
				Transparent 32	
			0x65	Vendor Specific AC Drive	
				ABB Drives profile	
				Transparent 16	
				Transparent 32	
3	Product Code	Get	1000 +	Drive product code + 1000	UINT16
4	Revision	Get		Revision of the FCNA software	Struct of
				Major revision	UINT8
				Minor revision	UINT8

#	Attribute name	Services	Range, default	Description	Type
5	Status			Summary of device status	UINT16
		0		Self Test	
			Bit0	Owned	
			Bit1	0	
			Bit2	Configured	
			Bit3	0	
			Bit4… Bit7	Extended device status.	
				<table><tr><td>Bit4… Bit7</td><td>Extended status</td></tr><tr><td>0000</td><td>Self test</td></tr><tr><td>0001</td><td>Firmware update in progress</td></tr><tr><td>0010</td><td>Comm. fault. Can happen if a connection from another node is refused.</td></tr><tr><td>0011</td><td>Unkeyed, waiting for a connection</td></tr><tr><td>0100</td><td>Non-volatile configuration bad</td></tr><tr><td>0101</td><td>Major fault</td></tr><tr><td>0110</td><td>Connected, active</td></tr><tr><td>0111</td><td>Idle</td></tr><tr><td>1000… 1001</td><td>0</td></tr><tr><td>1010… 1111</td><td>Reserved for product specific states</td></tr></table>	
			Bit8	Minor Recoverable fault	
			Bit9	Minor Unrecoverable fault	
			Bit10	Major Recoverable fault	
			Bit11	Major Unrecoverable fault	
				0	
6	Serial Number	Get		Serial number of FCNA	UINT32
7	Product name	Get		Product identification. Max 32 characters, ie, "ACS880"	SHORT_ STRING
				Number of characters in array	UINT8
				Product identification	Array of UINT8

■ **Assembly object, class 04h**

The Assembly object binds attributes of multiple objects, which allows data to or from each object to be sent or received over a single connection. Assembly objects can be used to bind input data or output data. The terms "input" and "output" are defined from the network's point of view. An input will produce data on the network and an output will consume data from the network.

Class attributes

Supported services: 0x0e – Get attribute single

#	Attribute name	Services	Range, default	Description	Type
1	Revision	Get	1…1, 1	Revision of the assembly object	UINT16

Instance attributes

Supported services:
- 0x0e – Get attribute single
- 0x10 – Set attribute single

#	Attribute name	Services	Range, default	Description	Type
3	Assembly Data	Get, Set		Data of instance assembly	Array of UINT8

Output assemblies

#	Profile	Assembly name
20	ODVA AC/DC	Basic speed control
21	ODVA AC/DC	Extended speed control
22	ODVA AC/DC	Basic speed and torque control
23	ODVA AC/DC	Extended speed and torque control
121	ODVA AC/DC	Basic speed control plus drive parameters
122	ODVA AC/DC	Basic speed and torque control plus drive parameters
123	ODVA AC/DC	Extended speed and torque control plus drive parameters
124	ODVA AC/DC	User specific drive parameters assembly up to 32 words

#	Profile	Assembly name
1	ABB Drives	Speed
2	ABB Drives	Speed and torque
101	ABB Drives	Speed plus drive parameters
102	ABB Drives	Speed and torque plus drive parameters
103	ABB Drives	User specific drive parameters assembly up to 32 words
11	Transparent T16	Transparent 16 with one assembly
12	Transparent T16	Transparent 16 with two assembly
111	Transparent T16	Transparent 16 with one plus drive parameters
112	Transparent T16	Transparent 16 with two plus drive parameters
21	Transparent T32	Transparent 32 with one assembly
22	Transparent T32	Transparent 32 with two assembly
121	Transparent T32	Transparent 32 with one plus drive parameters
122	Transparent T32	Transparent 32 with two plus drive parameters

Input assemblies

#	Profile	Assembly name
70	ODVA AC/DC	Basic speed control
71	ODVA AC/DC	Extended speed control
72	ODVA AC/DC	Basic speed and torque control
73	ODVA AC/DC	Extended speed and torque control
171	ODVA AC/DC	Basic speed control plus drive parameters
172	ODVA AC/DC	Basic speed and torque control plus drive parameters
173	ODVA AC/DC	Extended speed and torque control plus drive parameters
174	ODVA AC/DC	User specific drive parameters assembly up to 32 words
51	ABB Drives	Speed
52	ABB Drives	Speed and torque
151	ABB Drives	Speed plus drive parameters
152	ABB Drives	Speed and torque plus drive parameters
153	ABB Drives	User specific drive parameters assembly up to 32 words
61	Transparent T16	Transparent 16 with one assembly
62	Transparent T16	Transparent 16 with two assembly

#	Profile	Assembly name
161	Transparent T16	Transparent 16 with one plus drive parameters
162	Transparent T16	Transparent 16 with two plus drive parameters
71	Transparent T32	Transparent 32 with one assembly
72	Transparent T32	Transparent 32 with two assembly
171	Transparent T32	Transparent 32 with one plus drive parameters
172	Transparent T32	Transparent 32 with two plus drive parameters

■ **Connection manager, class 06h**

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

Class attributes

Supported services:
- 0x0E – Get Attribute Single
- 0x01 – Get Attributes All

There are no class attributes.

Instance attributes

Supported services:
- 0x54 – Forward Open
- 0x4E – Forward Close

#	Attribute name	Services	Range, default	Description	Type
1	Open requests	Set		Number of Forward Open service requests received.	UINT16
2	Open format rejects	Set		Number of Forward Open service requests which were rejected due to bad format.	UINT16

#	Attribute name	Services	Range, default	Description	Type
3	Open resource rejects	Set		Number of Forward Open service requests which were rejected due to lack of resources.	UINT16
4	Open other rejects	Set		Number of Forward Open service requests which were rejected for reasons other than bad format or lack of resources.	UINT16
5	Close requests	Set		Number of Forward Close service requests received.	UINT16
6	Close format requests	Set		Number of Forward Close service requests which were rejected due to bad format.	UINT16
7	Close other requests	Set		Number of Forward Close service requests which were rejected for reasons other than bad format.	UINT16
8	Connection timeouts	Set		Total number of connection timeouts that have occurred in connections controlled by this Connection Manager	UINT16

#	Attribute name	Services	Range, default	Description	Type
9	Connection entry list	Get		List of Connections	STRUCT of
	NumConnEntries			Number of connection entries. This attribute, divided by 8 and rounded up for any remainder, gives the length of the array (in bytes) of the ConnOpenBits field of this structure.	UINT
	ConnOpenBits			List of connection data which may be individually queried by the Get/Search Connection Data Services. Each bit represents a possible connection.	ARRAY of BOOL
11	CPU_Utilization	Get		CPU Utilization in tenths of a percent.	UINT16
12	MaxBuffSize	Get		Amount of buffer space originally available.	UINT32
13	BufSize Remaining	Get		Amount of buffer space available at this time.	UINT32

■ **Motor data, class 28h**

This object serves as a database for motor parameters.

Class attributes

Supported services: 0x0e – Get attribute single

#	Attribute name	Services	Range, default	Description	Type
1	Revision	Get	1…1, 1	Revision of the Motor Data object	UINT16
2	Max instance	Get	1…1, 1	Max Instance of the Motor Data object	UINT16

Instance attributes

Supported services:

- 0x0e – Get attribute single
- 0x10 – Set attribute single

#	Attribute name	Services	Range, default	Description	Type
3	Motor Type	Get	AC/DC	Motor type	UINT16
			1	PM DC Motor	
			2	FC DC Motor	
			3	PM Synchronous motor	
			6	Wound rotor induction motor	
			7	Squirrel cage induction motor	
6	Rated Current	Get, Set	AC/DC	Rated Stator Current from motor name plate Unit: [100mA]	UINT16
7	Rated Voltage	Get, Set	AC/DC	Rated Voltage from motor name plate Unit: [V]	UINT16
8	Rated Power	Get, Set	AC/DC	Rated Power at Rated Frequency Unit: [W]	UINT32
9	Rated Frequency	Get, Set	AC	Rated Electrical Frequency Unit: [Hz]	UINT16
12	Pole Count	Get, Set	AC	Number of poles in the motor	UINT16
15	Base Speed	Get, Set	AC/DC	Nominal speed at rated frequency from motor name plate Unit: [RPM]	UINT16

■ **Control supervisor, class 29h**

Note: The Control supervisor object class is supported only when the ODVA AC/DC drive profile is used.

Class attributes

Supported services: 0x0e – Get attribute single

#	Attribute name	Services	Range, default	Description	Type
1	Revision	Get	1…1, 1	Revision of the Control Supervisor object	UINT16
2	Max instance	Get	1…1, 1	Max Instance of the Control Supervisor object	UINT16

Instance attributes

Supported services:
- 0x05 – Reset
- 0x0e – Get attribute single
- 0x10 – Set attribute single

#	Attribute name	Services	Range, default	Description	Type
3	Run 1	Get, Set	0…1, 0		BOOL
			0	Stop	
			1	Run	
4	Run 2	Get, Set	0…1, 0		BOOL
			0	Stop	
			1	Run	
5	Net Control	Get, Set	0…1, 0		BOOL
			0	Local control	
			1	Network control	

#	Attribute name	Services	Range, default	Description	Type
6	State	Get	0…7, 0	Status of drive	UINT8
			2	Not_Ready	
			3	Ready	
			4	Enabled	
			5	Stopping	
			7	Faulted	
7	Running 1	Get	0…1, 0		BOOL
			0	Stopped	
			1	Running	
8	Running 2	Get	0…1, 0		BOOL
			0	Stopped	
			1	Running	
9	Ready	Get	0…1, 0		
			0	Other state	BOOL
			1	Ready or Enabled or Stopping	
10	Faulted	Get	0…1, 0		BOOL
			0	No fault	
			1	Fault occurred	
11	Warning	Get	0…1, 0		
			0	No warning	BOOL
			1	Warning	
12	FaultRst	Get, Set	0…1, 0		BOOL
			0 –> 1	Fault reset	
13	Fault Code	Get	0	Fault code that caused the last transition to the Faulted state. Fault code coming directly from the drive without conversion.	UINT16

#	Attribute name	Services	Range, default	Description	Type
14	Warning Code	Get	0	Code word indicating a warning. If there are multiple warnings, the lowest code value is displayed. Fault code coming directly from the drive without conversion.	UINT16
15	CtlFromNet	Get	0…1, 0	Status of Run/Stop control source.	BOOL
			0	Local control	
			1	Network control	
16	DNFaultMode	Get, Set	2	Vendor-specific	UINT8
17	ForceFault		0…1, 0		BOOL
			0 –> 1	Forces the drive to fault	
20	NetIdleMode	Get	2…2, 2		UINT8
			2	Vendor-specific	

■ **AC/DC drive, class 2Ah**

This object models the functions specific to an AC or a DC drive, for example, speed ramp and torque control.

Note: The AC/DC drive object class is supported only when the ODVA AC/DC drive profile is used.

Class attributes

Supported services: 0x0e – Get attribute single

#	Attribute name	Services	Range, default	Description	Type
1	Revision	Get	1…1, 1	Revision of the AC/DC drive object	UINT16
2	Max instance	Get	1…1, 1	Max Instance of the AC/DC drive object	UINT16

Instance attributes

Supported services:

- 0x0e – Get attribute single
- 0x10 – Set attribute single

#	Attribute name	Services	Range, default	Description	Type
3	At reference	Get	0…1, 0		BOOL
			0	Actual Speed or Torque is out of Reference window	
			1	Drive Actual At reference (speed or torque reference) based on mode	
4	NetRef	Get, Set	0…1, 1	Request torque or speed reference to be local or from the network	BOOL
			0	Set Reference not CN Control	
			1	Set Reference at CN Control	
5	Net Control	Get, Set	0…1, 0		BOOL
			0	Local control	
			1	Network control	
6	Drive Mode	Get, Set	0…0, 0		UINT8
			0	Vendor-specific	
7	Speed Actual	Get		Actual drive speed Units: RPM / $2^{SpeedScale}$ where SpeedScale is attribute 22	SINT16
8	SpeedRef	Get		Speed reference Units: RPM / $2^{SpeedScale}$ where SpeedScale is attribute 22	SINT16

#	Attribute name	Services	Range, default	Description	Type
11	Torque Actual	Get		Actual torque Units: Nm / $2^{TorqueScale}$ where TorqueScale is attribute 24	SINT16
12	TorqueRef	Get, Set		Torque reference Units: Nm / $2^{TorqueScale}$ where TorqueScale is attribute 24	SINT16
18	AccelTime	Get, Set		Acceleration time in milliseconds	SINT16
19	DecelTime	Get, Set		Deceleration time in milliseconds.	SINT16
22	SpeedScale	Get, Set	-128… 127, 0	Speed scaling factor. Scaling is accomplished as follows: ScaledSpeed = RPM / $2^{SpeedScale}$	INT8
24	TorqueScale	Get, Set	-128… 127, 0	Torque scaling factor. Scaling is accomplished as follows: ScaledTorque = Nm / $2^{TorqueScale}$	INT8
29	Ref From Net	Get		Status of torque/speed reference	BOOL
			0	Local torque/speed reference	
			1	Network torque/speed reference	

■ **Drive parameter, class 90h**

Class attributes

Supported services: 0x0e – Get attribute single

#	Attribute name	Services	Range, default	Description	Type
1	Revision	Get	1…1, 1	Revision of the Drive Parameter object	UINT16
2	Max instance	Get	254… 254, 254	Max Instance of the Drive Parameter object	UINT16

Instance attributes

Supported services:
- 0x0e – Get attribute single
- 0x10 – Set attribute single

The instance and attribute correspond to drive parameter Group and Index as follows:
- Instance = Parameter Group (0…255)
- Attribute = Parameter Index (0…255)

■ **Fieldbus configuration, class 91h**

Class attributes

Supported services: 0x0e – Get attribute single

#	Attribute name	Services	Range, default	Description	Type
1	Revision	Get	1…1, 1	Revision of the Fieldbus Configuration object	UINT16
2	Max instance	Get	254… 254, 254	Max Instance of the Fieldbus Configuration object	UINT16

Instance attributes

Supported services:

- 0x0e – Get attribute single
- 0x10 – Set attribute single

Instance 1					
#	Attribute name	Services	Range, default	Description	Type
1…27	Configuration Group #1 – Parameter 1…27	Get, Set		See section *Drive configuration*.	UINT16

Instance 2					
#	Attribute name	Services	Range, default	Description	Type
1…12	Configuration Group #2 – Parameter 1…12	Get, Set		DATA OUT	UINT16[1]

[1] In ACS880 UINT32

Instance 3					
#	Attribute name	Services	Range, default	Description	Type
1…12	Configuration Group #3 – Parameter 1…12	Get, Set		DATA IN	UINT16[1]

[1] In ACS880 UINT32

■ **ControlNet, class F0h**

The ControlNet object provide a consistent Station Management interface to the Physical and Data Link Layers.

Class attributes

Supported services:
- 0x0E – Get Attribute Single
- 0x01 – Get Attributes All

#	Attribute name	Services	Range, default	Description	Type
1	Revision	Get		Revision of this object	UINT
2	Max instance	Get		Maximum instance number	UDINT

Instance attributes

Supported services:
- 0x0E – Get Attribute Single
- 0x05 – Reset
- 0x01 – Get Attributes All
- 0x10 – Set Attribute Single
- 0x4C – Get And Clear
- 0x4D – Listen Only
- 0x4E – Where Am I

#	Attribute name	Services	Range, default	Description	Type
80	Pending_Link_ Config	Get		Pending link configuration parameters	STRUCT of: 34 bytes
	Link_Config			Pending link parameters	STRUCT of: 12 bytes
	NUT_length			DLL NUT_length	UINT
	smax			DLL smax	USINT
	umax			DLL umax	USINT
	slotTime			DLL slotTime	USINT
	blanking			DLL blanking	USINT
	gb_start			DLL gb_start	USINT
	gb_center			DLL gb_center	USINT
	reserved			reserved	UINT
	modulus			DLL modulus	USINT
	gb_prestart			DLL gb_prestarts	USINT
	TUI			Keeper TUI	STRUCT of: 22 bytes
	unique_ID			Keeper CRC	UDINT
	status_flag			TUI flag	UINT
	reserved			Reserved	USINT [16]
81	current_link_ config	Get		Current link configuration parameters	STRUCT of: 34 bytes

#	Attribute name	Services	Range, default	Description	Type
82	diagnostic_ counters	Get, Get and Clear		Diagnostic counters	STRUCT of: 42 bytes
	buffer_errors			Buffer event counter	UINT
	error_log			Bad MAC frame log	BYTE[8]
	event_counters			Diagnostic counters	STRUCT of: 32 bytes
	good_frames_ transmitted			Good MAC frames transmitted (LSB first)	BYTE[3]
	good_frames_ received			Good MAC frames received (LSB first)	BYTE[3]
	selected_ channel_ frame_errors			Framing errors detected on an active receive channel	USINT
	channel_A_ frame_errors			Framing errors detected on channel A	USINT
	channel_B_ frame_errors			Framing errors detected on channel B	USINT
	aborted_frames_ transmitted			MAC frames aborted during transmission (transmit underflows)	USINT
	highwaters			LLC transmit underflow and LLC receive overflow	USINT
	NUT_overloads			No unscheduled time in NUT (all time used for scheduled transmissions)	USINT
	slot_overloads			More scheduled data queued for one NUT than allowed by the sched_max_frame parameter	USINT

#	Attribute name	Services	Range, default	Description	Type
	blockages			Single Lpacket size exceeds the sched_max_frame parameter	USINT
	non_ concurrence			Two or more nodes could not agree whose turn it is to transmit	USINT
	aborted_frames_ received			Incomplete MAC frames received	USINT
	lonely_counter			Number of times nothing heard on a network for 8 or more NUTs	USINT
	duplicate_node			MAC frame received from a node with local node's MAC ID	USINT
	noise_hits			Noise detected that locked the modem rx PLL	USINT
	collisions			Rx data heard just as we were going to transmit	USINT
	mod_MAC_ID			MAC ID of the current moderator node	USINT
	non_lowman_ mods			Moderator frames heard from non-lowman nodes	USINT
	rogue_count			Rogue events detected	USINT
	unheard_ moderator			MAC frames being heard but no moderators being heard	USINT
	vendor_specific			Vendor specific	USINT
	reserved			Reserved	BYTE[4]

#	Attribute name	Services	Range, default	Description	Type
83	station_status	Get		Station status	STRUCT of: 6 bytes
	MAC_ver			MAC implementation and implementation revision	BYTE
	vendor_specific			Vendor specific	BYTE[4]
	channel_state			Channel state LEDs, redundancy warning, and active channel bits	BYTE
84	MAC_ID	Get, Set		MAC ID switch and current settings	STRUCT of: 4 bytes
	MAC_ID_current			Current MAC ID	USINT
	MAC_ID_ switches			MAC ID switch settings	USINT
	MAC_ID_ changed			MAC ID switches changed since reset	BOOL
	Reserved			Reserved	USINT
85	Sched_max_ frame	Get, Set		Scheduled data limit	STRUCT of: 2 bytes
	Sched_max_ frame			Scheduled maximum frame	USINT
	Reserved			Reserved	USINT
86	Error log	Get		Driver firmware buffer error counts and troublesome node list	STRUCT of: 10 bytes
	buffer_errors			Buffer event counter	UINT
	error_log			Bad MAC frame log	BYTE[8]

#	Attribute name	Services	Range, default	Description	Type
87	extended_ diagnostic_ counters	Get, Get and Clear		Additional diagnostic counters	STRUCT of: 264 bytes
	unsched_ transmitted			Number of unscheduled packets transmitted	UDINT
	sched_highwater			Max number of shared/unscheduled packets in a transmit queue	UDINT
	sched_data			Schedule information	128 STRUCT of: 2 bytes (256 bytes)
	words_in_use			Number of scheduled words in use by this node this NUT	USINT
	Lpkt_in_use			Number of scheduled Lpackets in use by this node this NUT	USINT
88	Active_node_ table	Get		One bit for each MAC ID 1 = node present 0 = node missing	ARRAY of: 32 bytes
89	New_node_table	Get		One bit for each MAC ID 1 = node has joined link recently 0 = node has not joined link recently	ARRAY of: 32 bytes

■ **Keeper, class F1h**

The Keeper object holds the link attributes for all ControlNet devices on a link and is responsible for distributing those attributes to those devices in an orderly fashion. It also holds (for link scheduling software) a copy of the connection originator data for all connection originator devices using a network. If there are multiple Keeper objects on a link, they perform negotiations to determine

which Keeper is the Master Keeper and which Keeper(s) perform backup Keeper responsibilities. The Master Keeper is the Keeper actively distributing attributes to the nodes on the network. A backup Keeper is one that monitors Keeper-related network activity and can transition into the role of Master Keeper should the Master Keeper fail to perform.

The Keeper object also contains support for obtaining, holding, and releasing the Network Resource, a network semaphore that is used to eliminate conflicts when modifying the attribute data held by the Keeper object(s) on a ControlNet link.

Class attributes

Supported service: 0x0E – Get Attribute Single

#	Attribute name	Services	Range, default	Description	Type
1	Revision	Get		Revision of this object	UINT

Instance attributes

Supported services:
- 0x0E – Get Attribute Single
- 0x03 – Get Attribute List
- 0x04 – Set Attribute List
- 0x10 – Set Attribute Single
- 0x4B – Get Network Resource
- 0x4C – Hold Network Resource
- 0x4D – Release Network Resource
- 0x4E – Change Start
- 0x4F – Change Complete
- 0x50 – Change Abort
- 0x51 – Get Signature
- 0x52 – Get Attribute Fragment
- 0x53 – Set Attribute Fragment

#	Attribute name	Services	Range, default	Description	Type
0	status	Get		Keeper object status	STRUCT of: 2 bytes
	state		0…13	Current keeper operating state	USINT
			0	Power up - not on line	
			1	Power up - TUI wait	
			2	Power up - TUI poll	
			3	Backup	
			4	Master verify	
			5	Master	
			6	Faulted backup	
			7	Faulted master verify	
			8	Faulted master	
			9	Net change - backup	
			10	Net change - master	
			11	Net change - faulted Backup	
			12	Net change - faulted master	
	reserved			Reserved for data alignment	USINT

#	Attribute name	Services	Range, default	Description	Type
1	port_status	Get, Set		Port status for node 1 through node 99	STRUCT of: 10 bytes
	port_status			Port status	UINT
	ID			Node type identification	STRUCT of: 8 bytes
	vendor			Vendor-specific	UINT
	type			Product type	UINT
	code			Product code	UINT
	major			Major revision	USINT
	minor			Minor revision	USINT
FF	net_config	Get, Set		Current network parameters	STRUCT of: 12 bytes
	NUT			Network Update Time	UINT
	smax			Scheduled max node ID	USINT
	umax			Unscheduled max node ID	USINT
	Slot_Time			Slot time	USINT
	Blank_Time			Blanking time	USINT
	Gb_Start			Guard band start	USINT
	Gb_Center			Guard band centre	USINT
	Reserved			Reserved for data alignment	UINT
	Int_Cnt mod			Interval count modulus (macrocycle length)	USINT
	Gb_Prestart			Guard band prestart	USINT
100	name	Get, Set		Current name of this link	UINT [33]

#	Attribute name	Services	Range, default	Description	Type
101	RT_TUI	Get, Set		Table unique identifier	STRUCT of 22 bytes
	unique_ID			Attribute CRC	UDINT
	status_flag			TUI flag bits	UINT
	reserved			Reserved to allow common format with attribute 0x102	USINT [16]
102	Link_Tui	Get		Table unique identifier (current link only)	STRUCT of: 22 bytes
	unique_ID			Attribute CRC	UDINT
	status_flag			TUI flag bits	UINT
	Keeper_MAC_ID			MAC ID of node broadcasting the TUI	USINT
	reserved			Reserved for data alignment	USINT
	Net_Resource_ Vendor_Id			Vendor ID of object holding net resource	UINT
	Net_Resource_ Serial_Number			Serial number of object holding net resource	UDINT
	Net_Resource_ Class			Class of object holding net resource	UDINT
	Net_Resource_ Instance			Instance of object holding net resource	UDINT

#	Attribute name	Services	Range, default	Description	Type
103	Cable_Config	Get, Set		Current cable configuration	STRUCT of: up to 100 bytes
	Id			Id value	USINT
	Num_Elements			Number of cable configuration elements in network configuration	USINT
	Propagation_ time			Number of 100 nanosecond ticks	UINT
	Physical element				Phy element [24]
	Phy_element				STRUCT of: 4 bytes
	Vendor_id			Vendor code	UINT
	Product_code			Product code	USINT
	How_many				USINT
104	CO_summary	Get, Set		General information about the *co_data* attribute	STRUCT of 204 bytes
	Data_size			Size of the co_data attribute	UINT
	connection_info_ revision				USINT
	tool_keeper_ revision			Highest level co_data parse rule supported	USINT
	0ffsets			Array of word offsets into the co_data attribute, by MAC ID	UINT [100]

#	Attribute name	Services	Range, default	Description	Type
105	CO_data	Set/Get fragment		Connection originator information	STRUCT of: up to 14988 bytes
	Branch			Information relative to all routers, gateways, or connection originators on this link	STRUCT of: variable size
	num_devices			Number of devices on this link	UINT
	device			Details of this device	STRUCT of: variable size
	Type			Types of device	USINT
	Path_Size			Size of path to node	USINT
	Path			Path to device	array of UINT
	CO_data			Details of CO data	STRUCT of: variable size
	COP			CO password, as provided to the CO at the end of a scheduling session (see 242.3.10)	UDINT
	Size			Size of conn data	UINT
	Connection			Connection data	STRUCT of variable size

■ Connection configuration, class F3h

This object defines an interface used to create, configure and control CIP connections in a device.

Class attributes

Supported services:
- 0x01 – Get Attributes All
- 0x02 – Set Attributes All
- 0x08 – Create
- 0x09 – Delete
- 0x0e – Get Attribute Single
- 0x10 – Set Attribute Single
- 0x15 – Restore
- 0x4b – Kick Watchdog
- 0x4c – Open Connection
- 0x4d – Close Connection
- 0x4e – Stop Connection
- 0x4f – Change Start
- 0x50 – Get Status
- 0x51 – Change Complete (Write to flash)
- 0x52 – Audit change

#	Attribute name	Services	Range, default	Description	Type
1	Revision	Get		Revision	UINT
2	Max instance	Get		Maximum instant number	UDINT
3	Number of instances	Get		Number of connections currently instantiated	UDINT
8	Format Number	Get			UINT
9	Edit Signature	Get			UDINT

Instance attributes

Supported services:
- 0x01 – Get Attributes All
- 0x02 – Set Attributes All
- 0x08 – Create
- 0x09 – Delete
- 0x0e – Get Attribute Single
- 0x10 – Set Attribute Single
- 0x15 – Restore
- 0x4c – Open Connection
- 0x4d – Close Connection
- 0x4e – Stop Connection

#	Attribute name	Services	Range, default	Description	Type
6	Connection Path	Set			Struct of
	open_path_size			Open connection path size	USINT
	reserved			Reserved	USINT
	open_connection _path			Connection path	Padded EPATH
7	Proxy Config Data	Set			Struct of
	config_data_size			Length of config_data in bytes	UINT
	config_data			Proxy Config data	Array of octet
8	Connection Name	Set			Struct of
	name_size			Number of characters in the connection name	USINT
	reserved			Reserved	USINT
	connection_name			User-assigned connection name encoded in UNICODE	STRING2

#	Attribute name	Services	Range, default	Description	Type
9	I/O Mapping	Set			Struct of
	format_number			This number determines the format of this attribute	UINT
	mapping_data_size			Size in bytes of the mapping_data field that follows	UINT
	mapping_data			I/O mapping information associated with this instance	Array of BYTE
10	Target Config Data	Set			Struct of
	config_data_size			Length of config_data in bytes	UINT
	config_data			Target config data	Array of octet
11	Proxy Device ID	Set			Struct of
	vendor_id			Vendor ID	UINT
	product_type			Device type	UINT
	product_code			Product code	UINT
	major_rev			Major revision	USINT
	minor_rev			Minor revision	USINT

■ Port, class F4h

The Port object enumerates the CIP ports present on the device. One instance exists for each CIP port.

Class attributes

Supported services:
- 0x01 – Get attribute all
- 0x0e – Get attribute single

#	Attribute name	Services	Range, default	Description	Type
1	Revision	Get	1…1, 1	Revision of the Port object	UINT16

#	Attribute name	Services	Range, default	Description	Type
2	Max instance	Get	1…1, 1	Max Instance of the Port object	UINT16
3	Num Instances	Get	1…1, 1	Number of ports currently instantiated	UINT16
8	Entry Point	Get	1…1, 1	Returns the instance of the Port object that describes the port through which this request entered the device.	UINT16
9	Port Instance Info	Get		Array of structures containing instance attributes 1 and 2 from each instance.	ARRAY of STRUCT of
				Port type. Enumerates the type of port.	UINT16
				Port number. CIP port number associated with this port.	UINT16

Instance attributes

Supported services:
- 0x01 – Set attribute all
- 0x0e – Get attribute single

#	Attribute name	Services	Range, default	Description	Type
1	Port Type	Get	0…3, 2	Type of port	UINT16
			0	Connection terminates in this device	
			2	ControlNet	
			3	ControlNet redundant	
2	Port Number	Get	2…2, 2	CIP port number associated with this port	UINT16

#	Attribute name	Services	Range, default	Description	Type
3	Link Object	Get		Link object	Struct of
				Path length. Number of 16-bit words in the following path.	UINT16
				Link path. Logical path segments that identify the object for this port.	Padded EPATH
4	Port Name	Get		String which names the physical network port. **Note:** The connection through a NAP port cannot be identified. Port A connected through Port A. Port B connected through Port B.	SHORT_STRING
7	Node Address	Get		Node number of this device on a port. FCNA reports a zero length node address within the port segment.	

■ Base energy object, class 4Eh

Class attributes

Supported service: 0x0e – Get attribute single

#	Attribute name	Services	Range, default	Description	Type
1	Revision	Get	1…1, 1	Revision of the object	UINT16
2	Max Instance	Get	1…1, 1	Max instance of the object	UINT16

Instance attributes

Supported service: 0x0e – Get attribute single

#	Attribute name	Services	Range, default	Description	Type
1	Energy/ Resource type	Get	1…1, 1	Specifies the type of energy managed by this energy instance	UINT16
			1	Electrical	
2	Capabilities	Get	0…0, 0	Energy capabilities of the device. 0 - Energy measured	UINT16
3	Accuracy	Get	0…0, 0	Specifies the accuracy of power and energy metering results	UINT16
			0	Undefined	
7	Consumed Energy	Get		Consumed energy value in kWh	Array of UINT16
8	Produced Energy	Get		Produced energy value in kWh	Array of UINT16
12	Path			Array of Paths to Energy objects to be aggregated	EPATH

■ **Electrical energy object, class 4Fh**

Class attributes

Supported service: 0x0e – Get attribute single

#	Attribute name	Services	Range, default	Description	Type
1	Revision	Get	1…1, 1	Revision of the object	UINT16
2	Max Instance	Get	1…1, 1	Max Instance of the object	UINT16

Instance attributes

Supported service: 0x0e – Get attribute single

#	Attribute name	Services	Range, default	Description	Type
1	Real Energy Consumed	Get		Total real energy consumed	Array of UINT16
2	Real Energy Generated	Get		Total real energy generated	Array of UINT16
12	Average Current	Get	0…0, 0	Drive output current	INT32
19	L1-L2 voltage	Get	0	RMS L1 to L2 voltage	INT32
20	L2-L3 voltage	Get		RMS L2 to L3 voltage	INT32
21	L3-L1 voltage	Get		RMS L3 to L1 voltage	INT32
40	Phase Rotation	Get	0…2, 1	Phase rotation of a three-phase system 0 = None 1 = ABC 2= ACB	UINT16
41	Path			Path to associated Base energy object instance	EPATH

■ Non-electrical energy object, class 50h

Class attributes

Supported service: 0x0e – Get attribute single

#	Attribute name	Services	Range, default	Description	Type
1	Revision	Get	1…1, 1	Revision of the object	UINT16
2	Max Instance	Get	1…1, 1	Max Instance of the object	UINT16

Instance attributes

Supported service: 0x0e – Get attribute single

#	Attribute name	Services	Range, default	Description	Type
3	Energy Net	Get		Running total of energy consumed minus energy generated	Array of UINT16
4	Energy Type	Get	0x1001-0x1001, 0x1001	Specifies the type of energy managed by this energy instance. 0x1001 = Electricity	UINT16
5	Energy Unit	Get	0x1D04-0x1D04, 0x1D04	Engineering units used for the energy odometers. 0x1D04 = Kilowatt hours	UINT16
7	Normalization Multiplier	Get	1…1, 1	Multiplier value for the scaling formula to convert from engineering units to the normalized value expressed in Kilowatt-hours (kWh).	UINT16
8	Normalization Divisor	Get	1…1, 1	Divisor value for the scaling formula to convert from engineering units to the normalized value expressed in Kilowatt-hours (kWh).	UINT16
41	Path			Path to associated Base energy object instance	EPATH

9

Diagnostics

Contents of this chapter

This chapter explains how to trace faults with the status LEDs on the adapter module.

Faults and warning messages

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

LED indications

The adapter module is equipped with two bicolor LEDs and one tricolor diagnostic LED. The LEDs are described below.

Name	Color	Function
Module	Off	No power, offline, NAM not attached
	Orange	Device is online but no message has been sent or received in the last 5 seconds.
	Red	Error. Device has returned a status error or has not delivered a message or has received an error from the destination in the last 5 seconds. If there is an error, the LED stays red for 1 second.
	Green	Online. Device has been sending messages in the last 5 seconds.
	Red/Orange	Faulted master
	Red/Off	Faulted backup
	Green/Off	Net change
	Green/Orange	Fault net change
	Orange/Off	No master or backup

| Name	Color	Function
ControlNet; LEDs A and B	*Indications that use both ControlNet LEDs together*	
	Off	No power or reset
	Red	Failed unit
	Alternating red-green	Self test
	Alternating red-off	Incorrect node configuration (eg, duplicate node ID)
	Indications that use each ControlNet LED independently	
	Blinking red-green	Incorrect network configuration
	Blinking red	Cable fault or no other nodes present in the network (eg, broken cable, redundancy warning)
	Blinking green	Temporary network errors (eg, bad MAC frame, screeners not programmed)
	Green	Channel OK
	Off	Channel disabled

10

Technical data

Contents of this chapter

This chapter contains the technical data of the adapter module and the ControlNet link.

FCNA-01

The following figure describes the enclosure of the adapter module from the front and side.
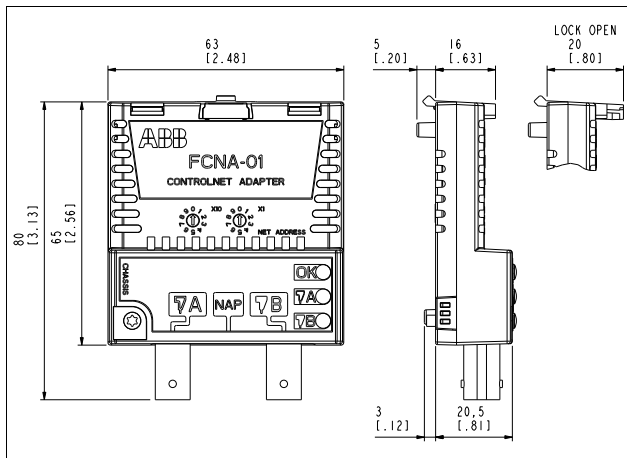
Installation	Into the option slot on the drive
Degree of protection	IP20
Ambient conditions	The applicable ambient conditions specified for the drive in its manuals are in effect.
Indicators	Two bicolor LEDs and one tricolor diagnostic LED: ControlNet, ControlNet and Module
Connectors	20-pin connector to the drive (X3) Two 8P8C modular jacks (X1 and X2)
Power supply	+3.3 V ±5% max. 450 mA (supplied by the drive)
General	Complies with EMC standard EN 61800-3:2004 Printed circuit board conformal coated

ControlNet link

Compatible devices	Any ControlNet scanner
Medium	RG-6 quad shielded cable (coax) • Termination: 75 Ohm, 1%, Metal Film, 1/4 W • Maximum Bus Length: 250 m (48 nodes), 1000 m (2 nodes) or 5000 m (with repeaters)
Topology	Bus, star or tree
Transfer rate	5 Mbit/s
Protocol	ControlNet

Further information

Product and service inquiries

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

Product training

For information on ABB product training, navigate to 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.

Contact us

www.abb.com/drives
www.abb.com/solar
www.abb.com/windpower
www.abb.com/drivespartners

Power and productivity
for a better world™

ABB