T2017EN Technical Manual Rev 06

MV3000e

CAN Port DeviceNet

DOCUMENT HISTORY

Revision Number	Date Of Revision	Details
Issue 0001	July 2001	Initial Release.
Issue 0002	September 2003	Minor modifications.
Issue 0003	July 2006	Company name change.
Issue 0004	June 2012	Company name change, etc.
Rev 0005	April 2013	Company name change, etc.
Rev 0006	March 2021	Company name change, etc.

DEFINITIONS

NOTE: Notes separate important information from the text and give additional information.

CAUTION "An instruction that draws attention to the risk of damage to the product, process or surroundings" (BS 4884-1: 1992).

WARNING "An instruction that draws attention to the risk of injury or death" (BS 4884-1: 1992).

SAFETY INSTRUCTIONS

Care has been taken with the design of this product to ensure that it is safe. However, in common with all products of this type, misuse can result in injury or death. Therefore, it is very important that the instructions in this manual and on the product are observed during transportation, commissioning, operation, maintenance and disposal.

This technical manual must be regarded as part of the product. It should be stored with the product and must be passed on to any subsequent owner or user. Local safety laws and regulations must always be observed.

Persons working on the product must be suitably skilled and should have been trained in that work for these products.

The product is a component designed for incorporation in installations, apparatus and machines. The product must not be used as a single item safety system. In applications where maloperation of the product could cause danger, additional means must be used to prevent danger to persons.

Product approvals and certifications will be invalidated if the product is transported, used or stored outside its ratings or if the instructions in this manual are not observed.

Third party approvals to safety standards UL508C and CSA C22.2 No 14 are marked on the product.

In The European Union:

Products within the scope of the Low Voltage Directive, 2006/95/EC are CE marked.

The product complies with the essential protection requirements of the EMC directive 2004/108/EC, when installed and used as described in this manual.

The requirements of the EMC Directive should be established before any installation, apparatus or machine, which incorporates the product, is taken into service. A machine must not be taken into service until the machine has been declared in conformity with the provisions of the Machinery (Safety) Directive, 2006/42/EC.

DISPOSAL

This equipment or any part of the equipment should be disposed of in accordance with the laws of the country of use.

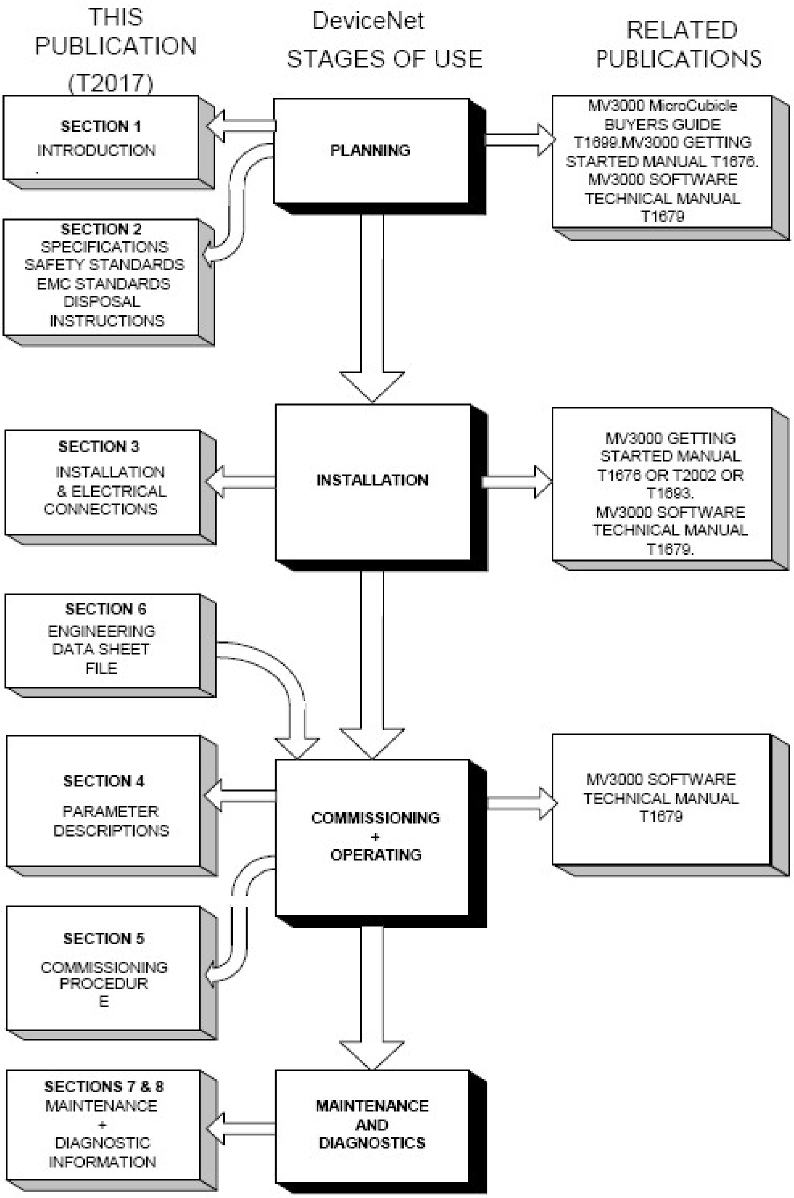
Modern high technology materials have been used in the manufacture of the equipment to ensure optimum performance. Care has been taken with the selection of these materials to minimise risks to health and safety. However, some materials require special consideration during disposal.

SCOPE

This publication should be read in conjunction with the appropriate Standard Product and/or MV3000e Manuals. This publication should be regarded as part of the product; it should be retained for the life of the product and passed on to any subsequent owner or user.

This manual describes the DeviceNet facility supported by the MV3000e Drive and provides detailed descriptions of the related drive parameters used to configure, monitor and operate the CANopen Interface.

OVERVIEW


THIS PUBLICATION (T2017)
DeviceNet STAGES OF USE
RELATED PUBLICATIONS
SECTION 1
INTRODUCTION
SECTION 2
SPECIFICATIONS
SAFETY STANDARDS
EMC STANDARDS
DISPOSAL
INSTRUCTIONS
PLANNING
MV3000 MicroCubicle BUYERS GUIDE T1699.MV3000 GETTING STARTED MANUAL T1676. MV3000 SOFTWARE TECHNICAL MANUAL T1679
SECTION 3
INSTALLATION & ELECTRICAL CONNECTIONS
INSTALLATION
MV3000 GETTING STARTED MANUAL T1676 OR T2002 OR T1693.
MV3000 SOFTWARE TECHNICAL MANUAL T1679.
SECTION 6
ENGINEERING DATA SHEET FILE
SECTION 4
PARAMETER DESCRIPTIONS
COMMISSIONING + OPERATING
MV3000 SOFTWARE TECHNICAL MANUAL T1679
SECTION 5
COMMISSIONING PROCEDUR E
SECTIONS 7 & 8
MAINTENANCE + DIAGNOSTIC INFORMATION
MAINTENANCE AND DIAGNOSTICS

TERMINOLOGY

Acronyms & Definitions

The following definitions, acronyms and terms are used within this manual:

CAN	Controller Area Network
CAN-ID (CAN Identifier)	All CAN packets require a CAN Identifier (CAN-ID). DeviceNet creates the CAN-ID from the node address, the service being implemented and sometimes the master node address, depending on the service being implemented.
CDC	Common Drive Controller the control board for the MV3000e range of drives.
CFSRC	A Control Flag Source
DeviceNet	CAN Protocol defined initially by Allen Bradley and now controlled by the ODVA (Open DeviceNet Vendor Association).
EDS	Electronic Data Sheet. The EDS file is used to provide a profile of the slave device e.g. the MV3000e Drive, to the CANopen Master. The file is supplied in PC readable format.
EPR	Expected Packet Rate attribute
GND	Ground
H	Used in this manual to indicate a number is hexadecimal
MAC-ID	Media Access Controller-Identifier, this is the Ethernet term for a node address
ODVA	Open DeviceNet Vendor Association
PDO	Process Data Object, the data telegram defined by CANopen to transfer data between CANopen nodes.

Terms & Definitions

Assembly Object	Used to collect various pieces of data together for transfer at the same time.
Class Descriptor	A DeviceNet node is comprised of a number objects. Objects of the same type belong to a class of objects. Examples of classes of objects are the Parameter, Assembly and Connection objects. Each object can exist more than once with an associated instance number. It is usual that instance zero of each object is the class descriptor.
Class Of Objects	Objects of the same type belong to a class of objects. Examples of classes of objects are Parameter, Assembly and Connection objects.
Connection Object	This object is used to manage the characteristics of a communication connection.
Consumer	A device that is a receptor and user of data sent on a network by a producing device.
DeviceNet Master	The device that gathers and distributes I/O data for the process controller.
DeviceNet Node	A DeviceNet node is comprised of a number of objects.
DeviceNet Object	The DeviceNet Object provides the configuration and status of a DeviceNet port. Each DeviceNet product must support one (and only one) DeviceNet object per physical connection to the DeviceNet communication link.
Explicit Messaging	Allows any Assembly Object data set to be read or written to by an Explicit Message to the data attribute of the Assembly Object Instance.
Fragmentation	DeviceNet allows data of more than 8 bytes (8 bytes is the maximum number of data bytes that can fit in a single CAN message) to be transmitted by using fragmentation.

Instance	A specific and real (physical) occurrence of an object.
I/O Message	An I/O Message consists of a Connection ID and associated I/O data. The meaning of the data within an I/O Message is implied by the associated Connection ID.
Master Node	A node that, as a minimum, can send 'Start' and 'Stop' commands to other nodes and controls the communication with a Slave Node.
Node	In a communications network it is a device connected to the network capable of communicating with other network devices. In a more generic sense, it is a connection point on a bus or a network.
Polled I/O	Allows the transfer of real time process data, via poll commands – a message is directed towards a single, specific slave – point-to-point. The master transmits a separate poll command for each of its slaves to be polled and poll responses (the polled slave returns its data if it is an input device).
Producer	A device that is a source and transmitter of data on a network.

CONTENTS

1. INTRODUCTION

1.1 GENERAL DESCRIPTION

The MV3000e Drive has a CAN (Controller Area Network) communication port which is used to provide high-speed communications with other CAN devices.

This manual describes the DeviceNet CAN protocol supported by the MV3000e Drive with Firmware Version 6.00 or later installed (Parameter P99.04 states the Firmware version for the drive).

The industry standard DeviceNet protocol provides a method of connecting DeviceNet compliant devices from any manufacturer onto a common DeviceNet Fieldbus network to achieve a desired function. The DeviceNet standard defines a number of 'Profiles' which a device can support. All devices have to support at least the Generic device Profile and can optionally support one other device function profile.

The MV3000e Drive supports the Generic Device Profile.

The DeviceNet Facility is parameterised and controlled using the same parameter interface as the rest of the drive. The board may also be configured over the drive's serial links, including the DeviceNet link itself.

1.2 EDS (ELECTRONIC DATA SHEET) FILE

An EDS file is used to provide a profile of a slave device, i.e. the MV3000e Drive, to the DeviceNet Master. The file is supplied in a PC readable format.

The Electronic Data Sheet File is described in more detail in Section 6.

1.3 CUSTOMER SUPPORT & TRAINING

GE Power Conversion provides comprehensive telephone technical support, application planning and service for customers. At the Academy, experienced engineers provide customers with first class training for all GE Power Conversion drive and control products.

Contact GE Power Conversion at the appropriate customer support telephone number shown at the end of this manual.

1.4 ASSOCIATED PUBLICATIONS

T1679 MV3000e Software Technical Manual

T1689 MV DELTA (Air Cooled) Technical Manual

T1692 Drive Coach MVS3004-4001 User's Guide

T1693 MV DELTA (Liquid Cooled) Technical Manual

T1968 MV3000e 2nd CAN port CANopen & DeviceNet Fieldbus Facility Technical Manual

T2013 MV3000e CANopen Technical Manual.

2. SPECIFICATION

2.1 DEVICENET PROTOCOL

The DeviceNet Fieldbus Facility implementation acts as a Group 2 DeviceNet Server Only, supporting Polled I/O and Explicit Messaging.

Polled I/O allows the transfer of real time process data, via Poll Commands (a message is directed towards a single, specific slave (point-to-point). The master transmits a separate poll command for each of its slaves to be polled and Poll Responses (the polled slave returns its data (if it is an input device)).

Explicit Messaging allows any Assembly Object data set to be read or written to by an explicit message to the data attribute of the Assembly Object Instance.

2.2 DEVICENET OBJECT CONCEPT

This is fully explained in the ODVA standard, but is briefly outlined here to ease the understanding of the manual.

A DeviceNet node comprises a number of objects. Objects of the same type belong to a class of objects. Examples of a class of objects are the Parameter, Assembly and Connection Objects. Each object can exist more than once with an associated instance number. It is usual that instance zero of each object is the class descriptor.

Each Instance of a Class has the same set of attributes, but has its own particular set of attribute values. Each Class instance has a number of attributes that can be accessed by the appropriate service code. Examples of attributes are: data attribute; status; number of attributes. Examples of service codes are: ‘get attribute single’, i.e. read the attribute value; and ‘set attribute single’, i.e. write to the attribute value.

2.3 DEVICENET OBJECT CLASSES SUPPORTED

The following object classes are supported:

- Identity Object – class code 1.
- Message Router Object – class code 2.
- DeviceNet Object – class code 3.
- DeviceNet Assembly Object – class code 4.
- DeviceNet Connection Object – class code 5.
- Parameter Object – class code 15.

These classes are described in the following sections. The reader is referred to the ODVA specification for a full description of these object classes.

2.3.1 Identity Object

Class Code 1.

This Object provides identification of, and general information about, the device. The Identity Object is required to be present in all DeviceNet products.

Class Attributes:

Instance 0 Attributes: Class Descriptor – not supported

Instance 1 Attributes: Drive Instance

Table 2-1 includes a description of the layout of the Identity Class.

Attribute ID	Access Rule	Name	Data Type
1	Get	Vendor	WORD
2	Get	Product Type	WORD
3	Get	Product Code	WORD
4	Get	Revision Major Revision Minor Revision	Structure BYTE BYTE
5	Get	Status	WORD
6	Get	Serial Number	LONG
7	Get	Product Name String Length ASCII String	Structure BYTE STRING

Table 2-1. – The Identity Object

2.3.2 Message Router Object

Class Code 2.

This object class is not accessible via the network.

2.3.3 DeviceNet Object

Class Code 3.

The Object provides the configuration and status of the DeviceNet port. Each DeviceNet product must support one (and only one) DeviceNet Object per physical connection to the DeviceNet communication link. This is defined within the DeviceNet Specification.

Instance 0 Attributes: Class Descriptor – see Table 2-2.

Attribute ID	Access Rule	Name	Data Type
1	Get	Revision	UINT

Table 2-2. — Instance 0 Attributes: Class Descriptor

Attribute ID	Access Rule	Name	Data Type
1	Get	Node Address	CHAR
2	Get	Data Rate	CHAR
3	Get	BOI	BOOL
4	Get/Set	Bus Off Counter	BYTE
5	Get	Allocation Info Allocation Choice Master Node Address	Structure Of CHAR BYTE

Table 2-3. – Instance 1 Attributes: DeviceNet Object Instance

Service Code	Service Name
0x0E	Get_Attribute_Single
0x10	Set_Attribute_Single
0x4B	Allocate_Master/Slave_Connection_Set
0x4C	Release_Master/Slave_Connection_Set

Table 2-4. – Supported Service Codes

2.3.4 DeviceNet Assembly Object

Class Code 4.

Assembly Objects are used to collect various pieces of data together for transfer at the same time. The faster transfer mechanism is to use the polled I/O method as the (up to) 4 items of receive data and the (up to) 4 items of transmit data are transferred in one 5 ms slot.

An Assembly Object Instance can also be accessed via explicit messaging. Because fragmentation is required to read or write to the Assembly Object instance's data attribute, it takes longer to complete the transaction. In addition, explicit messaging is only processed at the 10 ms scan rate. This means it will take at least 30 ms to transfer the 4 parameters contained in a fully populated Assembly Object.

Only the data attribute (3) is supported for the object class instances.

Supported services are: -

- get_attribute_single;
- set_attribute_single.

It is important to remember that as many as 4 completely unrelated drive parameters can be transferred per assembly object transaction. This is considerably faster than using the parameter object mechanism to transfer the data.

Because there are 5 transmit and 5 receive assembly objects, a maximum of 20 parameters can be received and a further 20 parameters can be transmitted by this mechanism.

The drive allows the user to specify the instance number of each of the 10 assembly object definitions provided via Menu 63. This allows the user the flexibility to configure class instances which are similar to other DeviceNet nodes on the network.

Using Assembly Object Instances to access parameter values requires that the DeviceNet Master has prior knowledge of the Parameter to Assembly Object data mapping.

2.3.5 DeviceNet Connection Object

Class Code 5.

This Object is used to manage the characteristics of a communication connection – see table 2-5.

Instance 0 Attributes: Class Descriptor – not supported

Instance 1 Attributes: Explicit Message connection instance

Instance 2 Attributes: Polled I/O connection instance

Attribute ID	Access Rule	Name	Data Type
1	Get	State	BYTE
2	Get	Instance Type	BYTE
3	Get	Transport Class Trigger	BYTE
4	Get	Produced Connection ID	BYTE
5	Get	Consumed Connection ID	BYTE
6	Get	Initial Comm Characteristics.	BYTE
7	Get	Produced Connection Size	BYTE
8	Get	Consumed Connection Size	BYTE
9	Get/Set	Expected Packet Rate	UINT
12	Get/Set	Watchdog Action	BYTE
13	Get	Produced Connection Path Length	BYTE
14	Get/Set	Produced Connection Path	
15	Get	Consumed Connection Path Length	BYTE
16	Get/Set	Consumed Connection Path	

Table 2-5. – The Connection Object

2.3.6 DeviceNet Parameter Object

Class Code 15.

DeviceNet requires that instances of the Parameter Object Class begin at 1 and continue uninterrupted to the maximum instance supported. In order to retain an intuitive association between drive parameters and DeviceNet parameters, there is a one-to-one relationship between the parameters.

To illustrate this, drive parameter P9.01 (speed feedback) has an equivalent DeviceNet parameter instance number of 901. Thus, multiply the drive ‘P’ number by 100 to get the DeviceNet instance number.

NOTE: Instance 0 is the class descriptor.

A consequence of this one-to-one relationship is that there are many undefined drive parameters which result in a null DeviceNet parameter instance. Thus an attempt to access a non-existent drive parameter will result in an access error.

The value of the drive parameter can be accessed using attribute 1 of the Parameter Object Instance. A complete enquiry of all the parameter attributes will give the whole definition of that parameter. This information is contained in the EDS file, but it may also be scanned directly from the drive.

Because there are more than 256 parameters in the drive, a 16 bit instance number has to be used to access Parameter Objects. Thus the drive requires that an 8 bit class and 16 bit instance number format is used to identify object class instances. This format is declared when the master establishes a connection to the drive. All access to any object class instance, not just parameter objects, is required to use the same format.

Instance 0 Attributes: Class Descriptor – see Table 2-6.

Attribute ID	Access Rule	Name	Data Type
2	Get	Max. Instance	UINT
3	Get	Parameter Class Descriptor	WORD
4	Get	Configuration Assembly Instance	UINT

Table 2-6. – Instance 0 Attributes: Class Descriptor

Instance 1-9999 Attributes: Parameter – see Table 2-7.

Attribute ID	Access Rule	Name	Data Type
1	Get / Set	Parameter Value	Data type specified in Descriptor, Data Type and Data Size.
2	Get	Link Path Size	USINT
3	Get	Link Path	ARRAY of DeviceNet path:
		Segment type/port	BYTE
		Segment Address	Path (format depends on data contained in segment type/port)
4	Get	Descriptor	WORD
5	Get	Data Type	USINT
6	Get	Data Size	USINT
7	Get	Parameter Name	SHORT_ STRING
8	Get	Units String	SHORT_ STRING
9	Get	Help String	SHORT_ STRING
10	Get	Minimum Value	data type
11	Get	Maximum Value	data type
12	Get	Default Value	data type
13	Get	Scaling Multiplier	UINT
14	Get	Scaling Divisor	UINT
15	Get	Scaling Base	UINT
16	Get	Scaling Offset	INT
17	Get	Multiplier Link	UINT
18	Get	Divisor Link	UINT
19	Get	Base Link	UINT
20	Get	Offset Link	UINT
21	Get	Decimal Precision	USINT

Table 2-7. – Instance 1-9999 Attributes: Parameters

Any attempt to access an instance that does not map to a drive parameter will return the service not supported error code.

2.4 DEVICENET I/O DATA

Polled data that is to be transferred is specified by a receive and transmit "Assembly Object".

When a connection is established for polled I/O, various attributes of the polled I/O connection instance specify the length and location (path) of the Assembly Object's instance number and data attribute.

The default receive path is Object class 4 (Assembly Object), instance number 120, attribute 3 (data). The length of the data is 4 bytes, i.e. 2 parameters.

The default transmit path is Object class 4 (Assembly Object), instance number 170, attribute 3 (data). The length of the data is 4 bytes, i.e. 2 parameters.

The maximum number of bytes a CAN data packet can have is eight, the minimum number of data bytes is zero. The I/O packet contains data that is transmitted to achieve control of the process being implemented over the DeviceNet network. After node power-on and following I/O packet configuration, which includes the CAN-ID allocation described in Section 2.9, the I/O packet transmissions have to be started by the DeviceNet Master. It does this in the following order.

- Establish an explicit message connection through the use of the 'Unconnected Explicit Request Message'.
- Establish an I/O message connection through the use of the 'Unconnected Explicit Request Message'. At the moment, the drive only supports the polled I/O message connection.
- Activate the message connections by writing to each connections 'expected packet rate' (EPR) attribute.
- Transmit data to the node using the I/O Poll Command Message CAN-ID. Receive Poll responses using the Slave Poll Response CAN-ID.

The instance number and length can be changed by the master using the appropriate explicit messages, before the connection is activated.

In order for a polled connection to function correctly, i.e. be polled and respond, the receive and transmit Assembly Instances must exist. Selecting the standard slave configuration (P61.42) will ensure that the instance number and parameter definitions are correct.

Instances are allocated to I/O assemblies in a similar way to I/O data being 'mapped' to an I/O assembly. In the Drive, the smallest unit of data is a 16-bit word. Therefore 4 words of data can be mapped to a single I/O assembly. Through the use of parameter pointers (Menu 63), the Drive I/O mapping can be configured.

Thoughtful allocation of data to I/O assemblies can result in shorter length packets and thus optimise the CAN network bandwidth.

Only one poll connection is allowed per node, and as only 4 receive and transmit parameters can be transferred this way, there is a need to transfer other values using some other mechanism. This can be achieved in two ways.

- Use explicit messaging to access any of the other Assembly Objects.
- Use explicit messaging to access any drive parameter through the Parameter Object Class.

2.5 DEVICENET PROFILE FOR VARIABLE SPEED DRIVES

The DeviceNet Profile for Variable Speed Drives contains the drive specific definition of a minimum set of Assembly Objects as well as a number of additional object class definitions, such as the motor data object class.

The drive does not support these extra object classes, so the DeviceNet type, found by accessing the identity object class, device type attribute, is reported as a generic device rather than the AC drive type.

Because the Assembly Object Class Instance Number is configurable through the use of Menu 63, it is possible to configure Assembly Object Instances which closely match the AC drives profile instance definitions. Indeed, close examination of the AC drives profile will reveal that the default instance numbers used in the drive are offset from the AC drives profile by a value of 100.

It is possible to configure the drive parameters contained in the Assembly Objects to behave in the same way as the AC drives profile Assembly Instances – see Appendix A.

2.5.1 I/O Assemblies for the AC Drive Profile

The I/O Assembly Instance definitions in this section define the format of the “data” attribute (attribute 3) for the AC Drive Profile I/O Assembly Instances. They are included here for reference. Consult the ODVA Specification for a more complete definition.

Table 2-8 defines the I/O Assembly Instances for AC drives.

Number dec	hex	Required /Optional	Type	Name
20	14	Required	Output	Basic Speed Control Output
21	15	Optional	Output	Extended Speed Control Output
22	16	Optional	Output	Speed and Torque Control Output
23	17	Optional	Output	Extended Speed and Torque Control Output
24	18	Optional	Output	Process Control Output
25	19	Optional	Output	Extended Process Control Output
70	46	Required	Input	Basic Speed Control Input
71	47	Optional	Input	Extended Speed Control Input
72	48	Optional	Input	Speed and Torque Control Input
73	49	Optional	Input	Extended Speed and Torque Control Input
74	4A	Optional	Input	Process Control Input
75	4B	Optional	Input	Extended Process Control Input

Table 2-8. – I/O Assembly Instances

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. Thus, instance 20, which is described as an output (from the DeviceNet network) is actually an input to the drive as it contains control bits and the speed reference.

2.5.2 AC Drive Profile I/O Assembly Data Attribute Format

The I/O Assembly Data Attributes have the format shown at Table 2-9 and Table 2-10. If a bit is not used in an I/O Assembly, it is reserved for use in other Assemblies. Reserved bits in Output Assemblies are ignored by the consuming device. Reserved bits in Input Assemblies are set (i.e. updated) to zero by the producing device.

2.6 CONNECTION PATHS TO I/O ASSEMBLY INSTANCES

The I/O Assembly Instances are chosen for I/O Connections by setting the "produced_connection_path" (attribute 14) and "consumed_connection_path" (attribute 16) attributes in the appropriate connection object.

2.7 DEVICENET DATA TYPES

The following standard data types are supported:

- Integer16
- Integer32
- Unsigned16
- Unsigned 32
- Simple String

Table 2-9. – I/O Assembly Data Attribute Format

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
20	0						Fault Reset		Run Fwd
	1								
	2	Speed Reference (Low Byte)							
	3	Speed Reference (High Byte)							
21	0		NetRef	NetCtrl			Fault Reset	Run Rev	Run Fwd
	1								
	2	Speed Reference (Low Byte)							
	3	Speed Reference (High Byte)							
22	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)							
23	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)							
24	0						Fault Reset		Run Fwd
	1								
	2	Speed Reference (Low Byte)							
	3	Speed Reference (High Byte)							
	4	Process Reference (Low Byte)							
	5	Process Reference (High Byte)							
25	0	Net Proc	NetRef	NetCtrl			Fault Reset	Run Rev	Run Fwd
	1	Mode							
	2	Speed Reference (Low Byte)							
	3	Speed Reference (High Byte)							
	4	Process Reference (Low Byte)							
	5	Process Reference (High Byte)							

NOTE: **Reserved bits in the I/O Assembly Data Attribute Format in table 2-9 are shaded.**

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
70	0						Running 1		Faulted
	1								
	2	Speed Actual (Low Byte)							
	3	Speed Actual (High Byte)							
71	0	At Ref.	Reference from net	Control from net	Ready	Running 2 (Rev)	Running 1 (Fwd)	Warning	Faulted
	1	Drive State							
	2	Speed Actual (Low Byte)							
	3	Speed Actual (High Byte)							
72	0						Running1		Faulted
	1								
	2	Speed Actual (Low Byte)							
	3	Speed Actual (High Byte)							
	4	Torque Actual (Low Byte)							
	5	Torque Actual (High Byte)							
73	0	At Ref.	Reference from net	Control from net	Ready	Running 2 (Rev)	Running 1 (Fwd)	Warning	Faulted
	1	Drive State							
	2	Speed Actual (Low Byte)							
	3	Speed Actual (High Byte)							
	4	Torque Actual (Low Byte)							
	5	Torque Actual (High Byte)							
74	0						Running 1		Faulted
	1								
	2	Speed Actual (Low Byte)							
	3	Speed Actual (High Byte)							
	4	Process Actual (Low Byte)							
	5	Process Actual (High Byte)							
75	0	At Ref.	Reference from net	Control from net	Ready	Running 2 (Rev)	Running 1 (Fwd)	Warning	Faulted
	1	Drive State							
	2	Speed Actual (Low Byte)							
	3	Speed Actual (High Byte)							
	4	Process Actual (Low Byte)							
	5	Process Actual (High Byte)							

Table 2-10. – I/O Assembly Data Attribute Format

NOTE: Reserved bits in the I/O Assembly Data Attribute Format in Table 2-10 are shaded.

2.8 COMMUNICATION ASSEMBLY OBJECTS

The MV3000e DeviceNet Fieldbus Facility provides 5 transmit 'Assembly Objects' and 5 receive 'Assembly Objects'. Each Assembly Object can be configured to transfer up to 4 drive parameters each. To reference an Assembly Object each one is given an 'instance' number.

One transmit and one receive Assembly Object can be used for the Polled I/O data transfer. The other 4 transmit and 4 receive Assembly Objects can then be used for Explicit Messaging.

2.8.1 Communication Between Drives

To communicate between two or more MV3000e Drives over DeviceNet all messages have to be routed through a DeviceNet Scanner (Master). Information from one drive is taken to the master in one polled connection and then passed to the other drive from another polled connection.

2.9 CAN-ID ALLOCATION

All CAN packets require a CAN Identifier (CAN-ID). DeviceNet creates the CAN-ID from the node address, the service being implemented and sometimes the master node address, depending on the service.

The CAN-ID is normally hidden from the user but it is useful to understand how it is calculated as the CAN-ID can be viewed by use of the P89.00 Data Spy feature (see section 2.9.1).

The CAN-ID is an 11-bit field. The top two bits identify the message group. For Group 2 messages, the lower 3 bits specify the message function. Table 2-11 identifies the types of message supported. You will note that the I/O Poll response requires a Group 1 message and the format is different to the Group 2 message.

CAN-ID Field	Group 2 Message Type
10xxxxxx111	Duplicate MAC ID Check Messages
10xxxxxx110	Unconnected Explicit Request Messages
10xxxxxx101	Master I/O Poll Command Messages
10xxxxxx100	Master Explicit Request Messages
10xxxxxx011	Slave Explicit Response Messages
01111xxxxxx	Slave Poll Response Messages

Table 2-11. – Identification of CAN-ID Message Types

2.9.1 Calculation of CAN-ID

In Table 2-11:

xxxxxx = Communication Module Node Address (MAC ID).

To calculate a CAN-ID in decimal for a Group 2 message:

CAN-ID = Type (0 to 7) + (MAC ID x 8) + 1024

For example, a poll command to node 5 gives a CAN-ID of:

5 + (5 x 8) + 1024 = 1069.

To calculate a CAN-ID in decimal for a Group 1 message:

CAN-ID = (MAC ID) + 512 + 448

For example, a poll response from node 5 gives a CAN-ID of:

5 + 512 + 448 = 965.

2.10 DATA

2.10.1 Data Transmission Rate

The following Baud rates are supported:

125 k Baud

250 k Baud

500 k Baud

2.10.2 Data Refresh Rate

Polled I/O data is processed every 5 ms. This means that a received I/O data packet is processed within 5 ms from reception of the packet.

Explicit messages are processed at a 10 ms rate.

2.10.3 Fragmentation

DeviceNet allows data of more than 8 bytes (8 bytes is the maximum number of data bytes that can fit in a single CAN message) to be transmitted by using fragmentation.

The MV3000e Drive does not support fragmentation of I/O data, thus I/O data can only occupy 8 bytes.

The drive does support fragmentation of explicit messages. However, the explicit message has to be split up into fragments that fit in the basic 8 byte CAN packet.

It will take 10 ms multiplied by the number of fragments to receive the explicit message. It will also take 10 ms multiplied by the number of fragments in the response.

For example, a 3 fragment message will take 30 ms to process.

2.10.4 Data Consistency

CAUTION

- **When using the DeviceNet interface to transfer values that span more than one 16-bit word (e.g. a 32-bit value) the interface does not guarantee that all members of the data set will be sent at the same time. The user must therefore ensure that the received value is valid.**

This is best explained by an example:

The position Feedback is contained in two parameters, P38.02 and P38.03. If the drive is configured to transmit these parameters to the DeviceNet network and the value is changing, the sequence of values shown in Table 2-12 could be transmitted on consecutive DeviceNet transmissions.

DeviceNet Transmission Number	Actual Value	P38.03	P38. 02	Value Received over DeviceNet
1	00009999	0000	9999	00009999
2	00010000	0001	0000	00000000[1]
3	00010000	0001	0000	00010000

Table 2-12. – Example of Problems with Changing Values During Transmission

It is safe to assume that the double word items will be correct if they are received or transmitted at the same scan rate as the drive produces or consumes the double word items.

2.11 CABLE SPECIFICATION

2.11.1 Electrical Specification

The connection between two or more MV3000e Drives is made via a 2-wire screened cable. It is recommended that a two-core cable, screened and twisted be used. The cable electrical specification is detailed in Table 2-13.

Function	Specification
Impedance	100 to 120 ohm at frequency > 100 kHz
Capacitance	< 60 pF per metre
Minimum Cross Section	0.22 mm^2 (24 AWG)
Velocity of Propagation	75%
Voltage Rating	300 V

Table 2-13. – DeviceNet Cable Electrical Specification

NOTE: The length of cable is determined by the network Baud rate – see Table 2-15.

[1] i.e. the value of P38.03 from transmission 1 and P38.02 from transmission 2

2.11.2 Recommended Cable Types

The recommended DeviceNet Thin and Thick cable types are listed in Table 2-14. Normally the DeviceNet Thick Cables are used for trunk cables and the Thin Cables for drop cables. Both types of cable comprise colour coded cores. Each cable has a red and black pair for carrying 24 V d.c. power and a blue and white pair for carrying data. These two pairs, together with a drain wire, are enclosed in a tinned copper braided screen.

The MV3000e Drive's CAN port takes its power from the Control Module Assembly and thus only requires three of the wires for connection, the blue and white data pair and the drain wire used as the ground connection. The 24 V d.c. power must not be connected.

A third type of DeviceNet cable, a flat cable, is not supported.

Cable Type	Insulation Type & Colour
Thin Cable	
Thin Cable – Belden Type 3084A	Grey PVC
Thin Cable – Belden Type 3085A	Yellow CPE
Thick Cable – Alternative Types	
Thick Cable – Belden Type 3082A	Grey PVC
Thick Cable – Belden Type 3083A	Yellow CPE

Table 2-14. –Recommended DeviceNet Cable Types

2.11.3 Cable Length

The length of a DeviceNet Network is determined by the Baud rate and type of cable used. The cable is specified in Table 2-15.

	125 kBaud	250 kBaud	500 kBaud
Thick Wire Trunk			
Trunk Length	500 m	250 m	100 m
Maximum Drop Length	6 m	6 m	6 m
Maximum Cumulative Drop	156 m	78 m	39 m
Number of Nodes	64	64	64
Thin Wire Trunk			
Trunk Length	100 m	100 m	100 m
Maximum Drop Length	6 m	6 m	6 m
Maximum Cumulative Drop	156 m	78 m	39 m
Number of Nodes	64	64	64

Table 2-15. – Maximum Cable Lengths for each Baud Rate

2.12 CONNECTOR SPECIFICATION

2.12.1 Connectors & Connections

The DeviceNet connection is achieved via the CAN_HI and CAN_LO and COMMS_GND terminals provided on Terminal Block TB4 as detailed in Table 3-1.

The connections at TB4 are made by screw-type terminations for each separate flexible cable.

All cables should have strain relief to prevent damage to the CANopen connection at TB4.

2.13 BUS CABLE TERMINATION

2.13.1 Terminating Resistors

Figure 3-1 shows a typical connection diagram for DeviceNet network cabling (using either thin or thick cables), including the terminating resistors, between an MV3000e Drive and a DeviceNet Master.

Each end of the DeviceNet network must have a 121 Ω ± 10%, ¼ W resistor fitted between the CAN_HI and CAN_LO wires. It is therefore necessary for the resistors to be supplied and fitted when the network is being installed.

NOTE: **Some devices have the terminating resistor built-in. If the MV3000e is at the end of a DeviceNet network, the 121 Ω resistor will be required to be fitted.**

The cable screen (drain wire) should be continuous and connected to a d.c. earth/GND at each MV3000e AC Drive and at all other nodes in accordance with the recommendations of the node equipment manufacturer.

2.14 STANDARDS

2.14.1 Safety Standards

EN50178 Electronic equipment for use in power installations.

2.14.2 EMC Standards

EN61800-3 / IEC61800-3 – Adjustable speed electrical power drive systems: EMC product standard including specific test methods.

2.14.3 DeviceNet Standards

DeviceNet Specification for Group 2 Server only V2.0
30 June 1998 - Open DeviceNet Vendor Association (ODVA).

NOTE: **This implementation of DeviceNet is for a slave device only.**

2.15 DISPOSAL INSTRUCTIONS

This equipment or any part of the equipment should be disposed of in accordance with the laws of the country of use.

3. INSTALLATION

WARNING

- This equipment may be connected to more than one live circuit. Disconnect all supplies before working on the equipment.
- Wait at least 5 minutes after isolating supplies and check that the voltage between DC+ and DC- has reduced to a safe level before working on this equipment.

3.1 INTRODUCTION

The MV3000e Drive, with Firmware Version 6.00 or later, has the DeviceNet Fieldbus Facility installed. The installation requirements therefore only apply to the wiring between drives or between a drive and a DeviceNet Master. These wiring details are included in this section.

3.2 DEVICENET WIRING TO THE DRIVE

The connection between two or more MV3000e Drives, or between a drive and a DeviceNet Master, is made via a 2 pair screened cable. The cable type and electrical specification is detailed in Section 2. Allowable network topologies are indicated in Figure 1 - Network Topologies at the rear of the manual.

It is important to strain relieve the DeviceNet cable before it is connected to the drive. Strain relief helps to prevent damage to the DeviceNet connection or the cable being unexpectedly unplugged.

These DeviceNet connections should also include terminating resistors – see 2.13.1 for specification. The connections are shown in Figure 3-1 and connection details listed in Table 3-1 and Table 3-2.[2]

[2] Early models of MV3000e MicroCubicle™ have TB4, later models have two connectors, TB4A (CAN Port) and TB4B (HSIO). Check the Drive for the appropriate connectors fitted.
For DELTA Controllers MVC3001-4001 use TB4 on user I/O termination Panel MVC3002-4001.
For DELTA Controllers MVC3001-4002 use TB4A on the Controller itself.

The connections at TB4 and TB4A are made by screw-type terminations.

The cable screen (drain wire) should be continuous and connected to a d.c. earth/GND at each MV3000e Drive and at all other nodes in accordance with the recommendations of the node equipment manufacturer.

The length of the DeviceNet cable is determined by the network Baud rate – see Table 2-15.

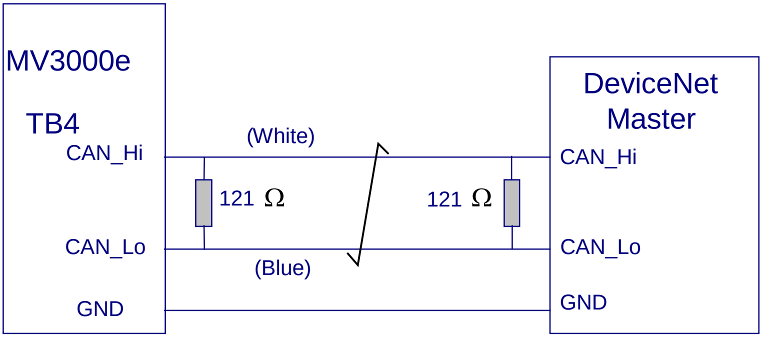


Figure 3-1. – Network Cabling Between an MV3000e Drive & a DeviceNet Master

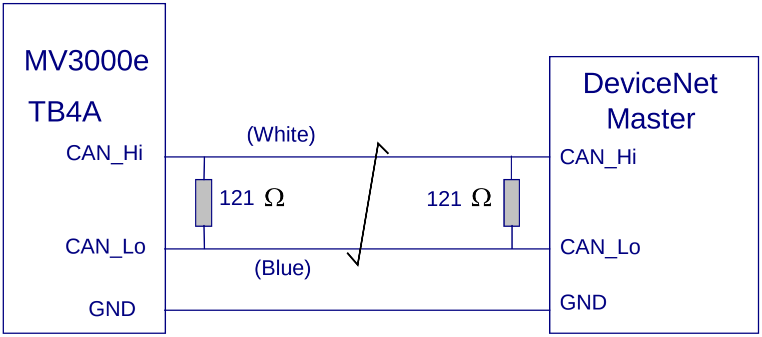


Figure 3-2. – Network Cabling Between an MV3000e Drive & a DeviceNet Master

TB Number	Pin Number	Designation	CAN Cable
TB4	5	Comms Ground	CAN_GND
	6	CAN_HI	CAN_HI
	7	CAN_LO	CAN_LO

Table 3-1. – MV3000e TB4 DeviceNet/CAN Connections

TB Number	Pin Number	Designation	CAN Cable
TB4A	1	CAN OV	CAN_GND
	2	CAN LO	CAN_LO
	3	SCN	
	4	CAN HI	CAN_HI
	5	Not Connected	

Table 3-2. – MV3000e TB4A DeviceNet/CAN Connections

3.3 CABLE SEGREGATION

Electrical noise and electromagnetic interference can be introduced into a microelectronics system via the cables and wires connected to it. To avoid this, wiring which could carry noise, that is 'dirty' cables, should be kept away from cables that are to be kept free from electrical noise or 'clean'. Wiring that falls into the same group can be run together, while wiring from different groups should be kept apart, though paths may cross at right angles. All connections to DeviceNet are considered to be clean.

3.4 DEVICENET EARTHING/GROUNDING

To ensure an EMC (Electromagnetic Compatibility) compliant installation, the network cable screens must be continuous and connected to a d.c. earth/GND at each MV3000e Drive and at all other nodes in accordance with the recommendations of the node equipment manufacturer.

Full details for drive earthing/grounding are included in the appropriate manuals T1676, T1689 or T1693 – see Section 1-4.

4. PARAMETERS

4.1 INTRODUCTION

This section describes the method of configuring the DeviceNet connection using the drive parameters. The section begins with a brief outline of the DeviceNet protocol when applied to Variable Speed Drives, followed by a list of the drive menus used by the DeviceNet.

4.2 DRIVE PARAMETERS

4.2.1 Drive Parameter Attributes

Each drive parameter has a number of attributes, which define the security classification, how the value is used and how the drive acts when a value is changed. The attributes for each drive parameter are given in the parameter list at the end of this section - see table 4-5 to Table 4-7.

The description for each parameter attribute C, E, L, N, O, R, and S is now described.

C = Confirm Edit, ‘Enter to confirm’, ‘Esc to abort’.

E = Engineer Accessible, these are drive configuration parameters which are set as part of the system design or during commissioning, and which should not be changed during normal operation. Refer to the drive manual for information on access control.

L = List Parameter, when stepping through the parameter values using the Drive Data Manager™ (Keypad) up and down arrow buttons, the values are chosen from a pre-set list, and no intermediate values are available.

N = Enter Parameter, some parameters may cause undesirable results if intermediate values are passed to the drive immediately while stepping through the values. To activate a changed parameter value with an N attribute press the Drive Data Manager™ (Keypad) ENTER button after selecting the correct value.

O = Operator Accessible, these are parameters which are used to control the drive within normal operating limits, and may need to be changed during normal operation. Refer to the drive manual for information on access control.

R = Read Only, these are either set during manufacture or are parameters that are used to monitor the state of various functions of the drive while it is running.

S = Stop To Edit, the drive must be stopped before this parameter can be updated.

4.2.2 Drive Parameter Access Authority

Updating any of the user parameters is only allowed if the communications device attempting to read or write to the parameter has the correct access authority. The DeviceNet Fieldbus has an 'engineer' status with regard to updating/accessing drive parameters.

The DeviceNet Fieldbus has the same access authority as the other Fieldbus links detailed in the drive manual.

4.2.3 List Parameters

When trying to access list parameters (see Section 4.2.1it is important to note that the value required is not necessarily the value indicated by the Drive Data Manager™ (Keypad). The first element in a list parameter has a value of 1, the second parameter has a value of 2, and so on. The value displayed by the Drive Data Manager™ may be different to this value. For example when configuring the RS232 serial link Baud rate, P32.10, the Drive Data Manager™ options are 9.6, 19.2 and 38.4 K Baud, which correspond to values of 1, 2 and 3 when configured via a DeviceNet network.

4.3 DEVICENET MENU

The configuration of the DeviceNet is implemented via parameter Menus 61 and 63. The menus are fully described in Sections 4.6 and 4.7 respectively.

4.4 BASIC SETTINGS EXPLAINED

The basic settings are those required to get the DeviceNet connection operating and to define the basic operating modes of the board. A block diagram is provided at the end of the manual to help with configuration of the DeviceNet connection.

The stages in configuring a DeviceNet connection are as follows:

a) Select the DeviceNet protocol in Parameter P61.00.

b) Select the DeviceNet Baud rate in Parameter P61.01.

c) Set up the DeviceNet network node address in Parameter P61.02

d) Configure the DeviceNet I/Os in Menu 63, making use of any of the CAN references and Control Words as required.

e) Set P61.03 to 1 to configure the link.

Appendix A gives a worked example.

4.5 SCALE MENU 60 DESCRIPTION

This menu allows the scaling between drive and Fieldbus to be specified for various different data types.

Each scale is specified by a pair of parameters per scale. The first parameter specifies the drive value equivalent to the external Fieldbus value contained in the second parameter of the pair. For example, if the drive represents a value of 100% by the value 10000 and the Fieldbus represent 100% by the value 16383, the first parameter will contain 10000 and the second parameter will contain 16383. When a parameter is to be output to the Fieldbus using the scale detailed in the above example the formula is:

$$Output = ParameterValue \times \frac{ExternalEquivalentValue}{DriveEquivalentValue}$$

When receiving a parameter from the Fieldbus the formula is:

$$ParameterValue = Input \times \frac{DriveEquivalentValue}{ExternalEquivalentValue}$$

Table 4-1 lists the data types that have been provided.

Data Type	Parameter
Speed (%)	P60.00 and P60.01
Speed (rpm)	P60.02 and P60.03
Torque (%)	P60.04 and P60.05
Torque (Nm)	P60.06 and P60.07
Frequency	P60.08 and P60.09
Percent	P60.10 and P60.11
Speed Ramp Rate	P60.12 and P60.13
Torque Slew Rate	P60.14 and P60.15
Current	P60.16 and P60.17
Volts	P60.18 and P60.19
Resistance	P60.20 and P60.21
Inductance	P60.22 and P60.23
User scaler 1	P60.24 and P60.25
User scaler 2	P60.26 and P60.27
User scaler 3	P60.28 and P60.29

Table 4-1. – Data Types

A full list of Menu 60 parameters is included in Table 4-5.

4.6 CDC CAN PORT MENU 61 DESCRIPTION

A full list of Menu 61 parameters is included in table 4-6.

4.6.1 CAN Protocol P61.00

Option 2 must be selected to activate the DeviceNet protocol.

4.6.2 CAN Baud Rate P61.01

On selecting the DeviceNet protocol in P61.01, the following options are available:

2=500 k Baud

3=250 k Baud

4=125 k Baud

Select the Baud rate according to: the application requirements; the DeviceNet Fieldbus cable length; the existing network Baud rate; the available Baud rates on other nodes.

The maximum number of CAN packets per second for a selected Baud rate can be approximated by dividing the Baud rate by 100. For example, the 500 k Baud rate would allow approximately 5000 packets per second on the DeviceNet bus.

4.6.3 Node ID P61.02

WARNING

- Avoidance of Same Node ID
  Ensure that no DeviceNet has the same node ID as any other device connected to the same network.

The DeviceNet requires that each node attached to a DeviceNet network has a unique node address, between 0 and 63. The node address is needed to calculate the unique CAN identifier.

(CAN-ID) for each CAN message. Reference to Section 2.9 shows that:

6 bits of the 11 bit CAN-ID are taken from the Node ID (also known as the MAC ID - Media Access Controller Identifier in DeviceNet).

This limits the MAC-ID to 63.

If nodes are configured with the same node address they will all try to respond at the same time and give unpredictable results.

4.6.4 Configure CAN Port P61.03

Once the node address has been entered, together with the Baud rate; node ID and any Menu 63 parameters, it is necessary to instruct the firmware to update the DeviceNet configuration. This is done by setting P61.03 to a 1. The firmware automatically rewrites this value to a zero when configuration is complete. Only after this occurs will any changes to the board's configuration take effect.

4.6.5 Run On Power Up P61.04

WARNING

- Avoidance of Automatic Configuring on Power-up
  The user should be aware that allowing the DeviceNet link to automatically configure and run on power-up could represent a possible safety hazard as the data from the network will be acted upon by the drive.

Many users may require the DeviceNet link to be automatically configured and run as the drive powers up. This option is available by setting P61.04 to a 1. This would normally be the last step during commissioning of the link.

For safety reasons the value of P61.04 is set to a default of zero.

4.6.6 Use Received Data P61.05

Disabling this parameter allows data to be sent to the CAN link, and data from the CAN link to appear in the 'spy' menu but not acted on, in any way, by the drive. Once it has been confirmed that the correct data is being received, the Drive can then be configured to 'go-live' be enabling this parameter.

Data can be sent to the link from the Drive without any disabling.

Disabling this parameter has no effect on the CAN 1 healthy status flag and it does not cause a freeze/fallback event. Because parameters are no longer updated, in effect, a freeze situation has occurred.

4.6.7 CAN Communications State P61.06

This parameter displays one of the values shown in Table 4-2, depending upon the state of the CAN interface.

Value	State	Meaning
0	Off-line	The CAN link is not communicating.
1	Bus Off	The CAN interface has gone into the Bus Off state due to many errors on the CAN bus. Incorrect Baud rate or CAN lines held low are typical causes. It may also be due to faulty or rogue devices.
2	Tx/Rx Errors Limit	Usually caused by being the only active node on the CAN network.
8	Disconnected	The DeviceNet node is configured but no connection with a master node has been established.
9	Connected	An explicit messaging connection with a DeviceNet master node is established

Table 4-2. – Communication States

4.6.8 Tx Per Second P61.07

This is a diagnostic display of the number of transmissions per second that this Drive is producing.

4.6.9 Rx Per Second P61.08

This is a diagnostic display of the total number of receptions per second that this Drive is receiving.

NOTE: The CAN total packets per second on the network is the sum of P61.07 and P61.08.

4.6.10 CAN Error Word P61.09

When transferring data between the CAN link and the Drive parameters, several possible errors could occur. The error format is as follows:

Bit	Error Format
Bit 0	Reserved.
Bit 1	Parameter write access denied, parameter has 'R' attribute set.
Bit 2	Data clamped to lower limit before being written.
Bit 3	Data clamped to upper limit before being written.
Bit 4	Data below bit parameter range and not written.
Bit 5	Data below bit parameter range and not written.
Bit 6	Reserved.
Bit 7	Attempt to write to a stop to edit parameter when running, data ignored.
Bit 8	Attempt to read from a non-existent parameter.
Bit 9	Parameter read access denied.
Bit 10-15	Reserved.

Table 4-3. – Error Format for P61.09

The errors apply because one or more of the data words in the active polled I/O assemblies have some form of access fault. Use of Menu 89 - Data Spy - will further identify the error cause.

The error word is the logical OR of any error occurring since the last power-on or CAN Port Configuration (P61.03). Thus it is possible to catch the occurrence of infrequent errors, such as an attempt to write outside a parameter's range.

The error word will be reset on enabling Configure CAN Port P61.03.

4.6.11 CAN Error Count P61.10

If the error is persistent, the count will increment rapidly. If it increments slowly, it may be that occasional over range errors have occurred.

The count will be reset on enabling Configure CAN Port P61.03.

4.6.12 Bus Off Action P61.11

If a bus off event occurs, it can be automatically cleared by selecting 1=Reset. Entering a value of 0=Hold means that the CAN peripheral is disconnected from the network until a power cycle or configuration event occurs.

Be careful in selecting 1= Reset as a persistent error will jam the CAN network.

4.6.13 CAN Loss Action P61.12

A loss of reference may be declared if the DeviceNet link detects a loss of communications with the master.

When a loss of reference is declared, one of three things can happen, depending upon the state of P61.12. Valid values of P61.12 are:

0	Ignore loss of reference errors and carry on running.
1	Upon a loss of reference set a warning and carry on running.
2	Upon a loss of reference trip the drive.

The warning code for loss of communications is 130, and the trip code is 200.

Status flag 111 indicates whether Fieldbus communications are healthy or not.

4.6.14 Freeze/Fallback Action P61.13

If a loss of DeviceNet communications is detected, the Fieldbus References and Control Words will no longer be updated by the DeviceNet link. P61.13 determines the action to be taken by the drive under these circumstances i.e. to Freeze or Fallback. The user has the choice of using the last values obtained from DeviceNet "Freeze" or using the values specified in the associated parameter "Fallback".

This fallback mechanism only applies to values in the odd numbered parameters between P61.15 and P61.33; values written directly to other parameters will always freeze.

4.6.15 CAN References P61.14 to P61.28
CAN Fallbacks P61.15 to P61.29

The CAN module can provide up to 8 references for use within the drive. These references can be used to control such values as the speed (if pointed to by one of the speed reference pointers) and torque. The CAN reference module needs to be used in conjunction with Menu 42 (the Pointer Menu), which is shown on the Control Block Diagrams at the end of this manual and also described in Section 6 of the Software Technical Manual T1679.

To place data from Rx 1 Word 1 (P63.52) into CAN Reference 1, for example:

Set Rx 1 Word 1 to point at the CAN Reference 1 parameter, i.e. P63.52 = 61.14

The data received at this position will then be copied to P61.14.

WARNING

- Avoidance of Duplicated Mapping
  Ensure that no two Fieldbus parameters are mapped to the same drive parameter as this will lead to unpredictable results.

If a loss of DeviceNet reference is declared, the reference value will no longer be updated by the DeviceNet link. In this situation the user has the choice of using the last value obtained from the DeviceNet link or using the fallback value.

These parameters will be written to by the DeviceNet. The references can be used to control such values as the speed reference (if pointed to by one of the speed reference pointers). The DeviceNet needs to be used in conjunction with Menu 42 (the Pointer Menu), which is described in the T1679 MV3000e Software Technical Manual.

4.6.16 CAN Control Words 1 & 2 P61.30 & P61.32 Control Fallback P61.31 & P61.33

These parameters will be written to by the DeviceNet. They each provide a source of 16 Control Flags that in turn allow control of functions such as starting and stopping.

Control flags in the CAN Control Words are selected by specifying a control flag source (CFSRC) value in the range 7.000 to 7.031 which corresponds to bits 0 to 31 respectively.

The description of how to map the Fieldbus Control Word parameters to the drive control flags is explained in Menus 33 and 34 of the T1679 MV3000e Software Technical Manual.

4.6.17 Parameters P61.34 to P61.39

These parameters are specifically for CANopen and have no function when DeviceNet is selected.

4.6.18 Life Time Factor P61.40

With DeviceNet, the Life Time Factor is the number of 100 ms re-try periods plus 1 that the Drive will wait before deciding that it has not received a I/O packet.

4.6.19 I/O Inhibit Time P61.41

Reserved for use with CANopen only.

4.6.20 Standard Slave P61.42

The default configuration of the first 2 of each of the Tx and Rx assembly objects is achieved by setting P61.42 to a 1. This causes the standard slave configuration to be invoked. The standard slave configuration corresponds with the EDS file.

This is a momentary enable parameter, which, when selected, causes all the following parameters in Menu 63, i.e. P63.0x, P63.1x, P63.5x and P63.6x, x = 1 to 9, to follow the table definition in table 4-4.

Parameter	Standard Slave Value
P63.01 = Tx 1 Instance	170
P63.02 = Tx 1 Word 1 Pointer	**P41.32 Prog Status Word 0**
P63.03 = Tx 1 Word 1 Scale	0 = Unity
P63.04 = Tx 1 Word 2 Pointer	P9.01 = Speed Feedback
P63.05 = Tx 1 Word 2 Scale	1 =Speed (%)
P63.06 = Tx 1 Word 3 Pointer	P0.00 = Unknown Parameter
P63.07 = Tx 1 Word 3 Scale	0 = Unity
P63.08 = Tx 1 Word 4 Pointer	P0.00 = Unknown Parameter
P63.09 = Tx 1 Word 4 Scale	0 = Unity
P63.11 = Tx 2 Instance	172
P63.12 = Tx 2 Word 1 Pointer	**P41.32 Prog Status Word 0**
P63.13 = Tx 2 Word 1 Scale	0 = Unity
P63.14 = Tx 2 Word 2 Pointer	P9.01 = Speed Feedback
P63.15 = Tx 2 Word 2 Scale	1 =Speed (%)
P63.16 = Tx 2 Word 3 Pointer	P9.04 = Torque Demand
P63.17 = Tx 2 Word 3 Scale	3 = Torque(%)
P63.18 = Tx 2 Word 4 Pointer	P0.00 = Unknown Parameter
P63.19 = Tx 2 Word 4 Scale	0 = Unity
P63.51 = Rx 1 Instance	120
P63.52 = Rx 1 Word 1 Pointer	P61.30 CAN Control Word 1
P63.53 = Rx 1 Word 1 Scale	0 = Unity
P63.54 = Rx 1 Word 2 Pointer	P61.14 = CAN Reference 1
P63.55 = Rx 1 Word 2 Scale	1 =Speed (%)
P63.56 = Rx 1 Word 3 Pointer	P0.00 = Unknown Parameter
P63.57 = Rx 1 Word 3 Scale	0 = Unity
P63.58 = Rx 1 Word 4 Pointer	P0.00 = Unknown Parameter
P63.59 = Rx 1 Word 4 Scale	0 = Unity
P63.61 = Rx 2 Instance	122
P63.62 = Rx 2 Word 1 Pointer	P61.30 CAN Control Word 1
P63.63 = Rx 2 Word 1 Scale	0 = Unity
P63.64 = Rx 2 Word 2 Pointer	P61.14 = CAN Reference 1
P63.65 = Rx 2 Word 2 Scale	1 =Speed (%)
P63.66 = Rx 2 Word 3 Pointer	P61.16 = CAN Reference 2
P63.67 = Rx 2 Word 3 Scale	3 = Torque(%)
P63.68 = Rx 2 Word 4 Pointer	P0.00 = Unknown Parameter
P63.69 = Rx 2 Word 4 Scale	0 = Unity

Table 4-4. – Standard Slave Parameter Definition

NOTE: **All previous values of these parameters (in Table 4-4) are overwritten, so use this parameter with care.**

4.6.21 Warn/Trip Source P61.43

If the drive generates a warning or trip that refers to this parameter for more details, the last cause is given here.

NOTE: Only the last cause of the fault will be displayed even though there may be multiple causes.

The causes are as follows:

0	None
14	1 to 14 Reserved for CANopen
15	Network Fault
16	No Process Data
17	Tx Poll loss
18	Rx Poll loss
19	Tx Strobe loss
20	Rx Strobe loss
21	Tx Cyclic/COS (Change of State) loss
22	Rx Cyclic/COS (Change of State) loss

States 17, 19 to 22 are reserved for future use and will not appear.

State 15 - 'Network Fault' occurs if the DeviceNet cable is disconnected or there are no other active nodes on the network. Refer to P61.06 to determine the type of network fault.

State 16 will be present until an I/O connection is established.

State 18 will exist if polled I/O commands are not received within the period defined by P61.40.

4.6.22 Unmonitored PDO Mask P61.44

By the use of P61.44, it is possible to exclude other instances from the presence checking. In this way, only instances essential to the particular process application will cause a fault if they are missing.

Any consumed instance is by default checked for presence at an interval of 100 ms. P61.40 specifies the number of checks before a fault is declared. P61.44 can be used to exclude instances from the check. Bits 0 and 1 set exclude the polled I/O from the check, bits 2 and 3 apply to the strobed I/O when supported and bits 4 and 5 will apply to the change-of-state/cyclic I/O when supported – see Figure 4-1.

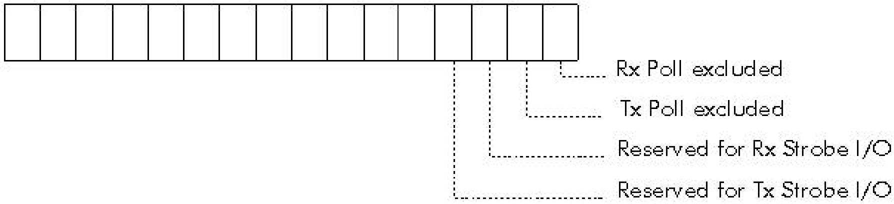


Figure 4-1. – Word Bits Set for Exclusion of Polled I/O & strobed I/O

4.6.23 Bus Off Count P61.50

This is a read only display of the number of Bus Off events that have occurred since power on. It is specific to DeviceNet and is used for diagnostic purposes only.

4.7 CDC DEVICENET MENU 63 DESCRIPTION

This menu will be used to configure the 10 DeviceNet assembly object instances.

Each active I/O packet can contain up to four 16-bit words. These words are defined in P63.x2 to P63.x9[3], where x is 0 to 9. See Figure 4-2.

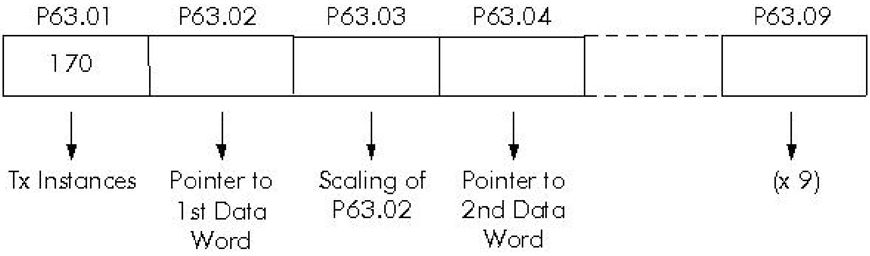


Figure 4-2. – I/O Packet Word Structure

4.7.1 P63.x0

This parameter is unused and not displayed.

4.7.2 Instance P63.x1

This parameter is used to specify the instance number that the assembly object will be allocated. Flexibility in specifying the instance number allows the user to configure instance numbers applicable to the particular application. For example, the AC Drive Profile instance numbers could be used to provide AC Drive Profile compatible polled I/O.

[3] These parameters are shared in the equivalent positions within the CDC CANopen Menu 62.

4.7.3 Tx/Rx Word 1 Pointer P63.x2

Depending on whether the assembly object is a producer or consumer, this is the pointer to the source or destination parameter of the first assembly object's data word.

Although it is possible to select any valid parameter number, the DeviceNet Fieldbus may not have the necessary write access rights to the parameter. While some parameters can be directly written to, others, such as references, cannot. For this reason a group of reference parameters are available to hold the received reference (P61.14 to P61.28). These references are then pointed to by Menu 42 for inclusion into the relevant reference selector – see Control Block Diagrams at the end of this manual.

4.7.4 Tx/Rx 1 Word Scaler P63.x3

This is the Scaler for the Word 1 Pointer. The default of zero is unity scaling. One of the scale types set up in Menu 60 can be selected to convert the Drive scaling for a parameter into DeviceNet scaling and vice-versa.

For example, if the parameter selected is speed, where a value of 10000 represents 100.00% within the drive, but the DeviceNet network requires 16383 to represent 100%, option 1 (Speed %) would be selected for the parameter scaler. In menu 60, P60.00 would be set to 10000 and P60.01 would be set to 16383.

NOTE: In the above example, the drive can represent ±300% using its scaling, but the network can only represent ±200%.

4.7.5 Tx/Rx 1 Word 2 Pointer P63.x4

The description for this is the same as for the Tx/Rx 1 Word 1 Pointer except that it is the data that is in the second two bytes of the assembly object's data.

NOTE: If this and the subsequent Tx/Rx Word pointers contain 00.00, less than 8 bytes of I/O data will be sent, thus optimising the CAN bandwidth usage.

4.7.6 Tx/Rx 1 Word 2 Scaler P63.x5

Description as for Tx/Rx 1 Word 1 Scaler.

4.7.7 Tx/Rx 1 Word 3 Pointer P63.x6

The description for this is the same as for the Tx/Rx 1 Word 1 Pointer except that it is the data that is in the third two bytes of the assembly object's data.

NOTE: If this and the subsequent I/O Word pointers contain 00.00, less than 8 bytes of I/O data will be sent, thus optimising the CAN bandwidth usage.

4.7.8 Tx/Rx 1 Word 3 Scaler P63.x7

Description as for Tx/Rx 1 Word 1 Scaler.

4.7.9 Tx/Rx 1 Word 4 Pointer P63.x8

The description for this is the same as for the Tx/Rx 1 Word 1 Pointer except that it is the data that is in the last two bytes of the assembly object's data.

NOTE: If this pointer is 00.00, no more than 6 bytes of I/O data will be sent, thus optimising the CAN bandwidth usage.

4.7.10 Tx/Rx 1 Word 4 Scaler P63.x9

Description as for Tx/Rx 1 Word 1 Scaler.

4.8 DATA SPY PARAMETERS P89.00 TO P89.06

The Data Spy parameters can be used to help debug and commission the DeviceNet network as well as monitoring data transactions during normal operation. Using the spy area of Menu 89 it is possible to monitor data received by the module, in its unmodified form. The data is available in parameters P89.01 to P89.05 and corresponds to the CAN-ID (see Section 2.9) of the assembly object of interest followed by up to four 16 bit words of data.

The values are displayed in decimal with no scaling and display exactly what is appearing on the DeviceNet network.

To select an assembly to be spied upon, enter a parameter number in P89.00 that is one of the assembly parameters. For example, P89.00 = 63.0x for Tx assembly 1, 63.5x for Rx assembly 1.

The values in P89.01 onwards will display uncollated until the newly selected assembly object has been accessed via DeviceNet.

If the assembly being spied upon is one of the polled I/O assemblies, the values in P89.01 onward will update on reception of the next poll and continue to do so while polls continue.

If the assembly being spied upon is not one of the polled I/O assemblies, the values in P89.01 onward will update the next time an explicit message access to that assembly is made.

The Data Spy Menu 89 Parameters are listed in table 4-8.

4.9 PARAMETER LISTING

4.9.1 Menus Listed

This section includes a listing of all the parameters, grouped in menus, for ease of reference. The listing includes:

Menu 60	Scaling	- see Table 4-5
Menu 61	CDC CAN Port	- see Table 4-6
Menu 63	CDC DeviceNet	- see table 4-7
Menu 89	Data Spy	- see Table 4-8.

4.9.2 Attributes

All parameters have attributes that specify how they may be accessed. The types of attributes used in this manual are:

C	=	Confirm Edit
E	=	Engineer Accessible
L	=	List Parameter
N	=	Enter Parameter
O	=	Operator Accessible
R	=	Read only
S	=	Stop to edit

4.9.3 Scaling Parameter Menu 60

Par No	Name	Default	Range	Attrib.	Comment
P60.00	Drive Speed (%)	10000	±32,767 pu %	E	
P60.01	Extern Speed (%)	10000	±32,767 pu %	E	
P60.02	Drive Speed (rpm)	1000	±32,767 pu rpm	E	
P60.03	Extern Speed (rpm)	1000	±32,767 pu rpm	E	
P60.04	Drive Torque (%)	10000	±32,767 pu %	E	
P60.05	Extern Torque (%)	10000	±32,767 pu %	E	
P60.06	Drive Torque (Nm)	10000	±32,767 pu Nm	E	
P60.07	Extern Torque (Nm)	10000	±32,767 pu Nm	E	
P60.08	Drive Frequency	100	±32,767 pu Hz	E	
P60.09	Extern Frequency	1	±32,767 pu Hz	E	
P60.10	Drive Percent	10000	±32,767 pu %	E	
P60.11	Extern Percent	10000	±32,767 pu %	E	
P60.12	Drive Ramp Rate	10000	±32,767 pu/s	E	
P60.13	Extern Ramp Rate	10000	±32,767 pu/s	E	
P60.14	Drive Torque Rate	10000	±32,767 pu/s	E	
P60.15	Extern Torque rate	10000	±32,767 pu/s	E	
P60.16	Drive Current	10	±32,767 pu A	E	
P60.17	Extern Current	1	±32,767 pu A	E	
P60.18	Drive Volts	1	±32,767 pu V	E	
P60.19	Extern Volts	1	±32,767 pu V	E	
P60.20	Drive Resistance	1	±32,767 pu Ω	E	
P60.21	Extern Resistance	1	±32,767 pu Ω	E	
P60.22	Drive Inductance	1	±32,767 pu H	E	
P60.23	Extern Inductance	1	±32,767 pu H	E	
P60.24	Drive Scalar 1	1	±32,767 pu	E	
P60.24	Extern Scalar 1	1	±32,767 pu	E	
P60.26	Drive Scalar 2	1	±32,767 pu	E	
P60.27	Extern Scalar 2	1	±32,767 pu	E	
P60.28	Drive Scalar 3	1	±32,767 pu	E	
P60.29	Extern Scalar 3	1	±32,767 pu	E	

Table 4-5. – Scaling Parameter Menu 60

4.9.4 CDC CAN Port Menu 61

Table 4-6. – CDC CAN Port Menu 61

Par No	Name	Default	Range	Attrib.	Comment
P61.00	CAN Protocol	0	0=None 1= Standard CANopen 2=DeviceNet 3= MV1000 CANopen	E, L	
P61.01	CAN Baud Rate	2	0= 1MBaud* 1= 800kBaud* 2= 500k Baud 3= 250k Baud 4= 125k Baud 5= 100k Baud* 6= 50k Baud* 7= 20k Baud* 8= 10k Baud*	E, L	* Not supported
P61.02	Node ID	0	0..255 for DeviceNet 0..63 for DeviceNet	E	
P61.03	Config CAN Port	0	0 or 1	E, N L	
P61.04	Config. on Power Up	0	0 or 1	E, L,N	
P61.05	Use Received Data	1	0 or 1	E, L	
P61.06	CAN Comms. State		0=Off-line 1=Bus Off 2=Tx/Rx Errors Limit 8=Disconnected 9=Connected	R	
P61.07	TX Per Second		Packets/sec	R	
P61.08	RX Per Second		Packets/sec	R	
P61.09	CAN Error Word		0000 to FFFFh	R	
P61.10	CAN Error Count	0	0-65535	R	
P61.11	Bus Off Action	1	0=Hold 1=Reset Port	E,N,L	
P61.12	CAN Loss Action	1	0=Ignore 1=Set Warning Bit 2=Trip Drive	E, L	
P61.13	Freeze/Fallback	0	0= Freeze 1= Fallback	E, L, S	
P61.14	CAN Reference 1	0	±200.00 %	E	
P61.15	CAN Ref. 1 Fallback	0	±200.00 %	E	
P61.16	CAN Reference 2	0	±200.00 %	E	
P61.17	CAN Ref. 2 Fallback	0	±200.00 %	E	
P61.18	CAN Reference 3	0	±200.00 %	E	
P61.19	CAN Ref. 3 Fallback	0	±200.00 %	E	
P61.20	CAN Reference 4	0	±200.00 %	E	
P61.21	CAN Ref. 4 Fallback	0	±200.00 %	E	

Table 4-6. – CDC CAN Port Menu 61					
Par No	**Name**	**Default**	**Range**	**Attrib.**	**Comment**
P61.22	CAN Reference 5	0	±200.00 %	E	
P61.23	CAN Ref. 5 Fallback	0	±200.00 %	E	
P61.24	CAN Reference 6	0	±200.00 %	E	
P61.25	CAN Ref. 6 Fallback	0	±200.00 %	E	
P61.26	CAN Reference 7	0	±200.00 %	E	
P61.27	CAN Ref. 7 Fallback	0	±200.00 %	E	
P61.28	CAN Reference 8	0	±200.00 %	E	
P61.29	CAN Ref. 8 Fallback	0	±200.00 %	E	
P61.30	CAN Control Word 1	0	0-FFFF	E	
P61.31	Control Fallback 1	0	0-FFFF	E	
P61.32	CAN Control Word 2	0	0-FFFF	E	
P61.33	Control Fallback 2	0	0-FFFF	E	
P61.34	CANopen Version	3.0	3.0 – 3.0	E,N	CANopen only
P61.35	CANopen Master Func	2	0=Disable 1=NMT Master 2=I/O Master	E, L,N	CANopen only
P61.36	CANopen Sync Master	0	0=Disable 1=Enable	E, L,N	CANopen only
P61.37	Sync Master Period	0	0..1000ms	E	CANopen only
P61.38	Node Guard Method	0	0=None 1=PDO Presence 2=Node Guarding 3=Heartbeat	E, L	Only option 1 supported CANopen only
P61.39	Node Guard Period	0	0..1000ms	E	CANopen only
P61.40	Life Time Factor	1	0 to 60	E	CANopen only
P61.41	IO Inhibit Time			E,N	CANopen only
P61.42	Standard Slave	0	0=Disable 1=Enable	E, L,N	
P61.43	Warn/Trip Source	None		R	
P61.44	Unmonitored PDO Mask	0000h	0-FFFFh	E	
P61.50	Bus Off Count	0	0-65536	R	

4.9.5 CDC DeviceNet Menu 63

Table 4-7. – CDC DeviceNet Menu 63

Par No	Description	Default	Range	Attrib.	Comment
P63.00	Unused	0			
P63.01	Tx 1 Instance *	0	0-200	E	
P63.02	Tx 1 Word 1 Pointer *	P0.00	P0.00 to P99.99	E,N	
P63.03	Tx 1 Word 1 Scale*	0	0=Unity 1=Speed (%) 2=Speed (rpm) 3=Torque (%) 4=Torque (Nm) 5=Frequency 6=Percent 7=Speed Ramp Rate 8=Torque Slew Rate 9=Current 10=Volts 11=Resistance 12=Inductance 13=Scalar 1 14=Scalar 2 15=Scalar 3	E, L	
P63.04	Tx 1 Word 2 Pointer *	P0.00	P0.00 to P99.99	E,N	
P63.05	Tx 1 Word 2 Scale*	0	See P63.03	E, L	
P63.06	Tx 1 Word 3 Pointer *	P0.00	P0.00 to P99.99	E,N	
P63.07	Tx 1 Word 3 Scale*	0	See P63.03	E, L	
P63.08	Tx 1 Word 4 Pointer *	P0.00	P0.00 to P99.99	E,N	
P63.09	Tx 1 Word 4 Scale*	0	See P63.03	E, L	
P63.11	Tx 2 Instance	0	0-200		
P63.12	Tx 2 Word 1 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.13	Tx 2 Word 1 Scale *	0	See P63.03	E, L	
P63.14	Tx 2 Word 2 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.15	Tx 2 Word 2 Scale *	0	See P63.03	E, L	
P63.16	Tx 2 Word 3 Pointer *	P0.00	P0.00 to P99.99		
P63.17	Tx 2 Word 3 Scale *		See P63.03	E, L	
P63.18	Tx 2 Word 4 Pointer *	P0.00	P0.00 to P99.99		

Table 4-7. – CDC DeviceNet Menu 63

Par No	Description	Default	Range	Attrib.	Comment
P63.19	Tx 2 Word 4 Scale *		See P63.03	E, L	
P63.20	Unused				
P63.21	Tx 3 Instance	0	0-200		
P63.22	Tx 3 Word 1 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.23	Tx 3 Word 1 Scale *		See P63.03	E, L	
P63.24	Tx 3 Word 2 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.25	Tx 3 Word 2 Scale *	0	See P63.03	E, L	
P63.26	Tx 3 Word 3 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.27	Tx 3 Word 3 Scale *	0	See P63.03	E, L	
P63.28	Tx 3 Word 4 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.29	Tx 3 Word 4 Scale *	0	See P63.03	E, L	
P63.30	Unused				
P63.31	Tx 4 Instance	0	0-200		
P63.32	Tx 4 Word 1 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.33	Tx 4 Word 1 Scale *	0	See P63.03	E, L	
P63.34	Tx 4 Word 2 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.35	Tx 4 Word 2 Scale *	0	See P63.03	E, L	
P63.36	Tx 4 Word 3 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.37	Tx 4 Word 3 Scale *		See P63.03	E, L	
P63.38	Tx 4 Word 4 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.39	Tx 4 Word 4 Scale *	0	See P63.03	E, L	
P63.40	Unused				
P63.41	Tx 5 Instance	0	0-200		
P63.42	Tx 5 Word 1 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.43	Tx 5 Word 1 Scale *	0	See P63.03	E, L	
P63.44	Tx 5 Word 2 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.45	Tx 5 Word 2 Scale *	0	See P63.03	E, L	
P63.46	Tx 5 Word 3 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.47	Tx 5 Word 3 Scale *	0	See P63.03	E, L	
P63.48	Tx 5 Word 4 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.49	Tx 5 Word 4 Scale *	0	See P63.03	E, L	
P63.50	Unused	0			
P63.51	Rx 1 Instance *	0	0-200	E	
P63.52	Rx 1 Word 1 Pointer *	P0.00	P0.00 to P99.99	E,N	
P63.53	Rx 1 Word 1 Scale*	0	See P63.03	E, L	
P63.54	Rx 1 Word 2 Pointer *	P0.00	P0.00 to P99.99	E,N	
P63.55	Rx 1 Word 2 Scale*	0	See P63.03	E, L	

Table 4-7. – CDC DeviceNet Menu 63

Par No	Description	Default	Range	Attrib.	Comment
P63.56	Rx 1 Word 3 Pointer *	P0.00	P0.00 to P99.99	E,N	
P63.57	Rx 1 Word 3 Scale*	0	See P63.03	E, L	
P63.58	Rx 1 Word 4 Pointer *	P0.00	P0.00 to P99.99	E,N	
P63.59	Rx 1 Word 4 Scale*	0	See P63.03	E, L	
P63.60	Unused	0			
P63.61	Rx 2 Instance *	0	0-200		
P63.62	Rx 2 Word 1 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.63	Rx 2 Word 1 Scale*	0	See P63.03	E, L	
P63.64	Rx 2 Word 2 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.65	Rx 2 Word 2 Scale*	0	See P63.03	E, L	
P63.66	Rx 2 Word 3 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.67	Rx 2 Word 3 Scale*	0	See P63.03	E, L	
P63.68	Rx 2 Word 4 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.69	Rx 2 Word 4 Scale*	0	See P63.03	E, L	
P63.70	Unused				
P63.71	Rx 3 Instance *	0	0-200		
P63.72	Rx 3 Word 1 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.73	Rx 3 Word 1 Scale*	0	See P63.03	E, L	
P63.74	Rx 3 Word 2 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.75	Rx 3 Word 2 Scale*	0	See P63.03	E, L	
P63.76	Rx 3 Word 3 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.77	Rx 3 Word 3 Scale*	0	See P63.03	E, L	
P63.78	Rx 3 Word 4 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.79	Rx 3 Word 4 Scale*	0	See P63.03	E, L	
P63.80	Unused				
P63.81	Rx 4 Instance	0	0-200		
P63.82	Rx 4 Word 1 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.83	Rx 4 Word 1 Scale*	0	See P63.03	E, L	
P63.84	Rx 4 Word 2 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.85	Rx 4 Word 2 Scale*	0	See P63.03	E, L	
P63.86	Rx 4 Word 3 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.87	Rx 4 Word 3 Scale*	0	See P63.03	E, L	
P63.88	Rx 4 Word 4 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.89	Rx 4 Word 4 Scale*	0	See P63.03	E, L	
P63.90	Unused				
P63.91	Rx 5 Instance *	0	0-200		
P63.92	Rx 5 Word 1 Pointer *	P0.00	P0.00 to P99.99	C, E	

Table 4-7. – CDC DeviceNet Menu 63

Par No	Description	Default	Range	Attrib.	Comment
P63.93	Rx 5 Word 1 Scale*	0	See P63.03	E, L	
P63.94	Rx 5 Word 2 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.95	Rx 5 Word 2 Scale*	0	See P63.03	E, L	
P63.96	Rx 5 Word 3 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.97	Rx 5 Word 3 Scale*	0	See P63.03	E, L	
P63.98	Rx 5 Word 4 Pointer *	P0.00	P0.00 to P99.99	C, E	
P63.99	Rx 5 Word 4 Scale*	0	See P63.03	E, L	

* This parameter requires a DeviceNet reconfiguration before changes can take effect.

4.9.6 Data Spy Menu 89

Parameter	Description	Value	Meaning
P89.00	Spy Config.	63.50	Spy Rx 1
P89.01	Spy Data 1	XXXX	CAN-ID (see Section 2.9)
P89.02	Spy Data 2	xxxx	Rx 1 Word 1
P89.03	Spy Data 3	8192	Rx 1 Word 2
P89.04	Spy Data 4	0	Rx 1 Word 3
P89.05	Spy Data 5	0	Rx 1 Word 4
P89.07	Spy Data 6	Uncollated	Unused
:		:	Unused
P89.32		Uncollated	Unused

Table 4-8. – Data Spy Menu 89

5. COMMISSIONING

WARNING

- This equipment may be connected to more than one live circuit. Disconnect all supplies before working on the equipment.
- Wait at least 5 minutes after isolating supplies and check that the voltage between DC+ and DC- has reduced to a safe level before working on this equipment.

5.1 MECHANICAL CHECKS

Check that the DeviceNet connection has been installed in accordance with the instructions given at Section 3. Check particularly for the following:

a) correct connections and cable types;

b) suitable cable length;

c) no ‘T’ connections in the network;

d) segregation of wiring to minimise electromagnetic interference;

e) network termination resistors have been fitted as recommended.

5.2 TOOLS & EQUIPMENT

The only equipment required for commissioning DeviceNet is either an Drive Data Manager™ MVS3000-4001 or an Drive Coach MVS3004-4001 with an associated PC for loading the DeviceNet software into either the MV3000e Drive or the MV DELTA Drive.

No special tools are required to commission DeviceNet.

5.2.1 DeviceNet Configuration

In most instances DeviceNet networks are configured using a DeviceNet Network Manager Software running on a pc connected via an interface to the DeviceNet network. This network will include a DeviceNet master module or a DeviceNet scanner module located in a PLC. MV3000e AC Drives are always a slave to this master or scanner.

5.2.2 MV3000e EDS File

2 EDS files are provided, DeviceNet.EDS and AB-DevNet.Eds as described in Section 6.

Choose the appropriate file for the DeviceNet master in the network and register it using the DeviceNet Network Manager Software. This manager will now be able to recognise the MV3000e AC Drive when it scans the network.

5.2.3 Configuring the MV3000e AC Drive

a) Set P61.00 - ' CAN Protocol = 2 - 'DeviceNet'

b) Set P61.01 - 'CAN Baud Rate' = 2 (500kBaud), 3 (250kaud) or 4 (125kBaud) as appropriate for the network to which the drive is being connected.

c) Set P61.02 - 'Node Id' = to the required Node Id (node address) for the network.

d) Set P61.42 - 'Standard Slave' = 1.

NOTE: This will reset to zero after a short time when the configuration is complete.

e) Set P61.03 - 'Config CAN Port' = 1 - Note: this will reset to zero after a short time when the configuration is complete

5.2.4 Configuring the DeviceNet Master

Configure the DeviceNet master to read and transmit the Polled I/O instances of the MV3000e AC Drive.

The default settings for the MV3000e Standard Slaver instances are described in Section 4.6.20., and for convenience, are repeated here.

Table 5-1. – Standard Slave Parameter Definitions

Parameter	Standard Slave Value
P63.01 = Tx 1 Instance	170
P63.02 = Tx 1 Word 1 Pointer	**P41.32 Prog Status Word 0**
P63.03 = Tx 1 Word 1 Scale	0 = Unity
P63.04 = Tx 1 Word 2 Pointer	P9.01 = Speed Feedback
P63.05 = Tx 1 Word 2 Scale	1 =Speed (%)
P63.06 = Tx 1 Word 3 Pointer	P0.00 = Unknown Parameter
P63.07 = Tx 1 Word 3 Scale	0 = Unity
P63.08 = Tx 1 Word 4 Pointer	P0.00 = Unknown Parameter
P63.09 = Tx 1 Word 4 Scale	0 = Unity
P63.11 = Tx 2 Instance	172
P63.12 = Tx 2 Word 1 Pointer	**P41.32 Prog Status Word 0**
P63.13 = Tx 2 Word 1 Scale	0 = Unity
P63.14 = Tx 2 Word 2 Pointer	P9.01 = Speed Feedback
P63.15 = Tx 2 Word 2 Scale	1 =Speed (%)
P63.16 = Tx 2 Word 3 Pointer	P9.04 = Torque Demand
P63.17 = Tx 2 Word 3 Scale	3 = Torque(%)
P63.18 = Tx 2 Word 4 Pointer	P0.00 = Unknown Parameter
P63.19 = Tx 2 Word 4 Scale	0 = Unity
P63.51 = Rx 1 Instance	120
P63.52 = Rx 1 Word 1 Pointer	**P61.30 CAN Control Word 1**
P63.53 = Rx 1 Word 1 Scale	0 = Unity
P63.54 = Rx 1 Word 2 Pointer	P61.14 = CAN Reference 1
P63.55 = Rx 1 Word 2 Scale	1 =Speed (%)
P63.56 = Rx 1 Word 3 Pointer	P0.00 = Unknown Parameter
P63.57 = Rx 1 Word 3 Scale	0 = Unity
P63.58 = Rx 1 Word 4 Pointer	P0.00 = Unknown Parameter
P63.59 = Rx 1 Word 4 Scale	0 = Unity

Table 5-1. – Standard Slave Parameter Definitions

Parameter	Standard Slave Value
P63.61 = Rx 2 Instance	122
P63.62 = Rx 2 Word 1 Pointer	**P61.30 CAN Control Word 1**
P63.63 = Rx 2 Word 1 Scale	0 = Unity
P63.64 = Rx 2 Word 2 Pointer	P61.14 = CAN Reference 1
P63.65 = Rx 2 Word 2 Scale	1 =Speed (%)
P63.66 = Rx 2 Word 3 Pointer	P61.16 = CAN Reference 2
P63.67 = Rx 2 Word 3 Scale	3 = Torque(%)
P63.68 = Rx 2 Word 4 Pointer	P0.00 = Unknown Parameter
P63.69 = Rx 2 Word 4 Scale	0 = Unity

To aid configuration of the DeviceNet connection a worked example is provided in Appendix A.

6. ELECTRONIC DATA SHEET FILE

6.1 INTRODUCTION

In order for a DeviceNet Master device to communicate with other DeviceNet compliant units on a network it needs to know, from a DeviceNet point of view, the capabilities of those units. These capabilities are described in a file provided by the unit's manufacturer known as an EDS (Electronic Data Sheet) file, and is in a simple text format and has file name extension of ".eds".

6.2 EDS FILE CONTENTS

What should be in an EDS file and how the file is constructed is described in the ODVA Specifications. For our purposes we can consider it to be in two parts:

(a) Header

This part of the file contains the manufacturer's information – the unit name, its issue, the issue of the EDS File etc. It also contains the basic information for the unit, such as the DeviceNet connection paths supported and the default connections that can be made.

(b) Body

This part describes all of the parameters that a DeviceNet Master can access and how it can access them i.e. read only, read and write etc.

The ODVA requires that every parameter that can be accessed is described and that the descriptions must start at the lowest parameter possible and be contiguous through to the highest parameter possible. Where a manufacturer has 'gaps' in his numbering system dummy parameters from P0.00 through to P0.99 need to be added at the beginning of the file and then dummy parameters added for any other 'missing' parameters up to the highest current parameter P99.17. This makes the file just under 10,000 parameters long.

Obviously if this file had to be generated by hand then it would be prone to numerous errors. To overcome this problem the parameter P35.07 – 'Print Out Option' (See Table 6-1) has been extended to add options to allow the drive to generate the 'body' of the EDS file itself.

P35.07 - 'Print Out Option'	Value
Default - No Print Out	0
Print All Parameters	1
Print User Edits	2
Print User Page 1	3
Print History	4
Auto Print History	5
CANopen EDS Body	6
DeviceNet EDS Body	7
AB-DevNet EDS Body	8

Table 6-1. – Print Out Options

Options 7 and 8 are for DeviceNet.

Option 7 - 'DeviceNet EDS Body' creates a file that when appended to a DeviceNet header file passes the ODVA's 'EDS Checker DeviceNet' Software.

Option 8 - 'AB-DevNet EDS Body' creates a file that when appended to a DeviceNet header file is suitable for being used with an Allen Bradley 1747-SDN DeviceNet Scanner and Rockwell Automation's RSNetWorx for DeviceNet configuration Software.

The file created by Option 8 is limited to 500 parameters and is a selection of the parameters available. Should the user require to access parameters that are not in the 500 selected then the EDS file entries for the parameters from the 'ODVA EDS file' can be substituted for parameters that are not required. Before attempting this operation the advice of the Engineering Services Group should be sought.

The parameters in the following Menus are included in the AB-DevNet EDS Body.

Menu	Description
1	User Configured Menu
2	Basic Motor Settings
3	Frequency Control Settings
4	Start and Stop Control
5	Speed Reference Settings
6	Ramp Settings
7	Plant I/O Settings
8	Torque Limits Settings
9	Basic Drive monitoring
10	Trips and Warnings
11	Advanced Drive Monitoring
16	PID Controller
19	Trim references
20	High Speed Digital I/O Settings
22	Skip Speed Settings
23	Dynamic Brake Control
24	Speed Trim
26	History Log Settings
27	History Log Playback Settings
35	Miscellaneous Features
38	Position Controller Monitor (Encoder Only)
51	SFE Monitor Menu
61	CAN Port Menu
98	Menu Control Menu
99	Configuration Menu

Table 6-2. – Menu Parameters included in AB-DevNet EDS Body

7. MAINTENANCE

7.1 GENERAL GUIDANCE

When DeviceNet is enabled on an MV3000e Drive, its maintenance requirements are included with maintenance of the drive, described in the T1676 MV3000e Getting Started Manual. The drive maintenance consists generally of checking for ingress of dust and moisture, and checking for security of electrical connections. The latter checks should include all the cables and connectors used for DeviceNet connections.

7.2 FIRMWARE REVISIONS

Firmware revisions for the DeviceNet software used in the MV3000e Drive are available from GE Power Conversion. Contact GE Power Conversion for details at the Customer Support telephone number listed at the end of this manual.

8. DIAGNOSTICS

8.1 FAULTS

If a previously working link has stopped working, one of the two error codes may be displayed depending upon the setting of P61.12.

8.2 CAN WARNING FAULT CODE

Table 8-1 shows the warning fault code displayed in the drive warnings tables, if the CAN link is not healthy, and P61.12, action on loss of CAN control source has been set to "warning".

Fault Code	Name	Description
130	CAN 1 Loss	CAN is not communicating correctly

Table 8-1. – Warning Fault Code

8.3 CAN TRIP FAULT CODE

Table 8-2 shows the trip fault code displayed in the drive trip tables, if the CAN link is not healthy, and P61.12, action on loss of CAN control source has been set to "trip".

Fault Code	Name	Class	Description
200	CAN 1 Loss	R	CAN is not communicating correctly

Table 8-2. – Trip Fault Code

A= Auto re-settable trip

R = Manually re-settable trip

S = System trip

N = Non re-settable trip.

8.4 CAN STATUS P61.06

Parameter P61.06 displays the state of the communications, see section 4.6.7.

8.5 SPARES

The DeviceNet interface is an integral part of the drive CDC (Common Drive Controller) board. The board does not contain any user replaceable parts and spares are therefore not applicable to the DeviceNet product. In the event of a CDC failure refer to GE Power Conversion Customer Support at the telephone number listed at the end of this manual. Whole CDC boards should be kept for spares. Any spare CDC boards must be at the same level, or higher, firmware release number.

9. APPENDIX A: CONFIGURATION EXAMPLE

9.1 INTRODUCTION

Consider a drive on a DeviceNet network that requires a Run/Stop command, speed reference and minimum speed set via the DeviceNet network. The drive produces its speed feedback and torque demand onto the DeviceNet network. This drive is to have a DeviceNet node address of 10. The Baud rate is to be 500k Baud.

Realistically Run/Stop and speed reference would probably be a 'global' requirement for other drives on the network, whereas minimum speed may only be required individually.

The drive first needs to be commissioned to run the motor, so carry out the commissioning procedure in the MV3000e Drive technical manuals. Assuming the drive and DeviceNet are correctly installed, the DeviceNet network now needs to be configured.

Refer to the block diagram at the rear of this manual to visualise the parameters and their uses.

9.2 DECIDE WHAT INFORMATION NEEDS TO BE TRANSFERRED

For a more complex set of data to be transferred, the tables contained in Appendix B may help in the allocation of the parameters. In this example, the requirements are simple, so we will not use the configuration tables.

If we refer to section 4.6.20, we can see that the first receive assembly can be used for the run/stop and speed reference and the second transmit assembly contains the speed feedback and torque demand. In addition, this assembly contains the programmable status word as the first word in the assembly. The easiest method of achieving the requirement is to configure the drive as a standard DeviceNet slave.

The minimum speed can be set either by including it in the receive assembly, or by setting it once by the use of the explicit messaging connection to DeviceNet parameter object instance P5.17 attribute 1. The choice will be influenced by the number of times the minimum speed value needs to be changed and by the capabilities of the master software. We shall assume that there is no need to include the minimum speed in the assembly object, and that the explicit messaging method is sufficient.

9.3 CONFIGURE THE DEVICENET LINK

Before configuring P61.00, the CAN Comms. State, P61.06 will display 'Off-line'. To configure basic CAN information into the drive, the parameters at Table 9-1 will have to be set.

Parameter	Value	Meaning
P61.00	2	Set Protocol to DeviceNet
P61.01	2	Set the Baud rate to 500 k
P61.02	10	Set the node ID to 10
P61.42	1	Set standard slave configuration

Table 9-1. – Basic Set-up

P61.12 is set to warn until the link has been fully commissioned, at which time it can be set to the desired state.

To configure the drive set P61.03 to a 1. This will be rewritten to a zero when configuration has been completed. With the DeviceNet link disconnected, or if the DeviceNet master has not been configured to communicate with slave 10, the CAN Comms. State, P61.06 will display 'Disconnected'. As soon as the CAN link is made, the receive packet count, P61.08 will display the total number of packets on the DeviceNet network. As soon as the DeviceNet master has connected to node 10, P61.06 will display 'Connected'. Assuming the master cannot yet poll the drive, the warning LED will remain illuminated and P61.43 will display the reason for the warning.

NOTE: **Setting P61.04 to a 1 allows the DeviceNet link to be automatically configured as the drive powers up.**

9.4 ADDING DRIVE PARAMETERS

Having configured the DeviceNet link it is then necessary to configure where the information to be transmitted over the link comes from and received information goes to. There are three basic types of data parameters: simple parameters, references and control/status bits.

9.4.1 Status Word

Any drive status word can be sent to the CAN network, either a pre-defined one from Menu 11, or a user-configured one from Menu 41, or a user configured one from an unused CAN expanded digital output word from Menu 59, (see CANopen manual T2013). This example, shown at Table A-2, will send the 16 Status Flags selected by P41.00 to P41.15, as Word 1. If P41.00 is set to 2.000 (Drive Running) and P41.02 is set to 2.005 (Drive Tripped), this will correspond to the AC Drive Profile definition of assembly instance 72 word 1.

9.4.2 Simple Parameters

Simple parameters are parameters that the DeviceNet network can be configured to write to or read from directly. In this example the minimum speed, speed feedback and torque demand are examples of simple parameters.

To transmit simple parameters out onto the link, enter the parameter number into the appropriate Tx assembly Word Pointer parameter. The set-up for monitored values would be as in Table 9-2.

As it is, the standard definition of Tx 2 assembly not only meets this example’s requirements, but exactly matches the next two words of the AC Drive Profile instance 72 definition. Ensure that the correct values of scalar are selected to present the correctly scaled values to the DeviceNet network.

9.4.3 Example for Transmitted/Monitored Data

Parameter	Standard Slave Value
P63.11 = Tx 2 Instance	172
P63.12 = Tx 2 Word 1 Pointer	**P41.32 Prog Status Word 0**
P63.13 = Tx 2 Word 1 Scale	0 = Unity
P63.14 = Tx 2 Word 2 Pointer	P9.01 = Speed Feedback
P63.15 = Tx 2 Word 2 Scale	1 =Speed (%)
P63.16 = Tx 2 Word 3 Pointer	P9.04 = Torque Demand
P63.17 = Tx 2 Word 3 Scale	3 = Torque(%)
P63.18 = Tx 2 Word 4 Pointer	P10.10 Trip No.1
P63.19 = Tx 2 Word 4 Scale	0 = Unity

Table 9-2. – Tx 2 Assembly Parameter Set-up

9.4.4 Received Control Bits

Control bits are bits in the drive that cannot be written to directly, but may be accessed indirectly by placing the received Control Word into a CAN Control Word container. Control bits perform such tasks as starting and stopping the drive. This example requires access to 2 control bits, the start flag and the stop flag. The first stage to setting up a control bit is to produce a CAN Control Word, which is similar to a CAN reference. CAN Control Words are held in P61.30 and P61.32. CAN Control Word 1 (from Rx 1 Word 1) is used in this example. The start and stop bits are transferred using CAN Control Word 1. The CAN Control Word is set up in Table 9-3.

9.4.5 Received References

References are parameters within the drive that cannot be written to directly, speed reference is one such example. In order to be able to write to the speed reference, it is necessary to write the DeviceNet data to a CAN reference and then select this reference to be the speed reference.

The first stage to setting up a reference parameter is to produce a CAN reference. CAN references are held in P61.14 to P61.28. Even numbers contain the reference, odd numbers contain the fallback for the reference. In this example the standard slave configuration uses CAN reference 1, see Table 9-3.

Having configured the P63.52 pointer to transfer data to a CAN reference "container", the user has the option of declaring a ‘fallback’ value for that reference. This will be dealt with later.

9.4.6 Received Simple Parameters

This example also has a simple reference parameter, P5.17, minimum speed. As stated above, the parameter will be accessed using the parameter object instance 517 and not using assembly object mapping.

Parameter	Standard Slave Value
P63.51 = Rx 1 Instance	120
P63.52 = Rx 1 Word 1 Pointer	P61.30 CAN Control Word 1
P63.53 = Rx 1 Word 1 Scale	0 = Unity
P63.54 = Rx 1 Word 2 Pointer	P61.14 = CAN Reference 1
P63.55 = Rx 1 Word 2 Scale	1 =Speed (%)
P63.56 = Rx 1 Word 3 Pointer	P0.00 = Unknown Parameter
P63.57 = Rx 1 Word 3 Scale	0 = Unity
P63.58 = Rx 1 Word 4 Pointer	P0.00 = Unknown Parameter
P63.59 = Rx 1 Word 4 Scale	0 = Unity

Table 9-3. – Received Parameter Set-up

NOTE: The minimum speed parameter P5.17 and its units are in rpm, not percentages. The drive is connected to a 1500 r/min motor.

9.5 SCALING PARAMETERS

The ‘container’ for the received reference can only have a value of ±100.00%, which internally is represented as ±10000. We have already selected the speed reference scaler. All that is required is to ensure that the scaler settings in Menu 60 are correct. In this example 100% network speed and torque are represented by a value of 16383.

Parameter	Value	Meaning
P60.00	10000	Drive’s speed (%) equivalent
P60.01	16383	Network’s speed (%) equivalent
P60.02	1500	Drive’s speed (rpm) equivalent
P60.03	15000	Network’s speed (rpm) equivalent
P60.04	10000	Drive’s Torque (%) equivalent
P60.05	16383	Network’s Torque (%) equivalent

Table 9-4. – Scaler Parameter set-up

9.6 USING THE RECEIVED PARAMETERS

Once the CAN reference has been configured it is necessary to configure the speed reference to come from CAN reference 1. Each of the drive references has a parameter or parameters that set-up where the source of that reference is. For the speed references these parameter are P5.01 to P5.05. These parameters need to be configured to source the reference from the 'Pointer menu', Menu 42. Each of the pointers in Menu 42 is dedicated to one of the drive references. The speed reference has pointers 1 and 2 dedicated to it. We will use pointer 1, see Table 9-5.

Parameter	Value	Meaning
P5.01	21	Speed reference source 1 - use pointer 1
P5.02	0	Speed reference source 2 - not required
P5.03	0	Speed reference source 3 - not required
P5.04	0	Speed reference source 4 - not required
P5.05	0	Speed backup source – not required

Table 9-5. – speed Reference Set-up

Having configured the speed reference source to be Menu 42 pointer 1, the final stage is to configure pointer 1 to use CAN 1, set up previously. Pointer 1 is configured in P42.00 and P42.01 as shown in Table 9-6.

Parameter	Value	Meaning
P42.00	61.14	Pointer 1 source – P61.14 CAN reference 1
P42.01	100.00	Pointer 1 scale - 100.00%

Table 9-6. – Speed Reference Pointer 1 Set-up

The data transferred in the CAN Reference 1 word will now be used as the speed reference. Note that Menu 42 allows the use of a scaling function to scale the incoming data before it is written to the relevant reference. This scale will also apply to the fallback value if it used.

Once the CAN Control Word has been configured it is necessary to configure the start and stop control bits source to come from CAN Control Word 1. Each of the drive control bits has a parameter that sets up where the source of that control bit is. Referring to Table 9-7, for the start bit this parameter is P33.01 for the stop bit this parameter is P33.00. These parameters need to be configured to source their values from the CAN

Control Word. CAN Control Word 1 contains values 7.000 to 7.015 which are CAN Control Word 1 bits 0 to 15, values 7.016 to 7.031 are CAN Control Word 2 bits 0 to 15.

If we continue to use the AC Drive Profile model, bit 0 of the first word is the combined run/stop control. The drive has separate parameters for run and stop control. However, if we use the inverted state of bit zero as the stop flag, the combined run/stop bit can be used.

Parameter	Value	Meaning
P33.00	-7.000	Stop control flag source – 7.000 CAN Control Word 1, bit 0
P33.01	7.000	Start control flag source - 7.000 CAN Control Word 1, bit 0
P34.16	1	Pass Start/Stop control to the control flags

Table 9-7. – Control Flag Source Set-up

9.7 ACTIVATING THE CONFIGURATION

The Fieldbus parameters now contain a basic configuration. To configure the Drive, P61.03 needs to be momentarily enabled.

9.8 DATA FOR MASTER DEVICE CONFIGURATION

The DeviceNet master device producing the data for this drive should now be configured. The data required to get the drive running is shown in Table 9-8.

Word	Value	Meaning
0	XXXXH	Contains the run/stop bits
1	8192	Set the speed reference to 50%
2	0	Not used
3	0	Not used

Table 9-8. – DeviceNet Master’s Transmit I/O Data Allocation

The data produced by the drive is listed in Table 9-9.

Word	Units	Meaning
0	Binary flags	Contains the drive status bits
1	0.0061%	Speed feedback.
2	0.0061%	Torque demand.
3	0	Not used

Table 9-9. – DeviceNet Master's Received I/O data Allocation

NOTE: The DeviceNet master represents 100% by a value of 16383.

The drive will make the Rx 1 assembly instance of 120 the default consumed path for the polled I/O connection. However, it will also make the Tx 1 assembly instance of 170 the default produced connection path. The DeviceNet master must either change the default produced connection instance from 170 to 172 to connect to the Tx 2 assembly, OR, the instance value in the drive should be changed. This is achieved by altering the Tx 1 instance number to an unused value (e.g. 200) and Tx 2 instance number changed to 170.

If the latter option is taken, don't forget to re-configure the link by enabling P61.03.

9.9 USING THE DATA SPY

Having parameterised a basic configuration, the user may wish to know what data is actually being transferred over the DeviceNet network. This can be done using the spy module in Menu 89.

To view Rx 1, enter 63.50 in P89.00. Entering 63.10 in P89.00 will view Tx 2. The meaning of P89.01 to P89.05 is shown in Table 9-10.

Parameter	Description	Value	Meaning
P89.00		63.50	Spy Rx 1
P89.01		XXXX	CAN-ID (see Section 2.9)
P89.02		xxxx	Rx 1 Word 1
P89.03		8192	Rx 1 Word 2
P89.04		0	Rx 1 Word 3
P89.05		0	Rx 1 Word 4
P89.07		Uncollated	Unused
:		:	Unused
P89.32		Uncollated	Unused

Table 9-10. – Menu 89

The data, for Menu 89, will be displayed un-scaled and in decimal. The drive may limit data before writing it into a parameter if it exceeds that parameter's limits, see section 4.8.

9.10 CHECK FOR DATA ERRORS

Once the drive has been configured the user should check that no data errors are occurring. Data errors can occur when the DeviceNet module attempts to write the data from the polled I/O data into drive parameters. This may be because the parameter is not able to accept data in this manner, or because the data itself is outside accepted limits. For the receive and transmit I/O packets there is an error code of the last error that occurred (P61.09) and the number that have occurred since the last CAN configuration attempt (P61.10). The list of data error codes and their meaning is given in Table 4-3. An example for the source data is given in Table 9-11.

Parameter	Value	Meaning
P61.09	200h	Read access from a parameter was denied
P61.10	20	20 errors have occurred since the last CAN link configuration.

Table 9-11. – Data Error Examples

In this example read access to a parameter was denied, possibly because that parameter does not exist. This implies that one of the parameters set for source is incorrect and does not actually exist.

9.11 FALLBACK VALUES

The example declares a CAN reference and a CAN Control Word, both of which have the option of a fallback value. A fallback value is a value that will be used instead of the CAN value if the CAN link becomes unhealthy. As a simple example, if the CAN link is lost then the user may want to force the speed reference to zero. To use this feature the CAN link first needs to be configured to select fallback value on a loss of the CAN link. This is done by setting parameter P61.13 to a 1.

If P61.13's value is set to 0 then on a CAN loss the reference value will be frozen. This means that the last valid value received over the link will be used.

NOTE: ALL CAN reference/control values are affected by the freeze/fallback simultaneously.

The fallback values can now be entered for the relevant CAN reference, for the speed reference we will use a fallback value of 0.

If communication is lost then we would like the drive to stop; this means that the start control flag should be set to 0. This means the fallback value should be 0000h.

These settings are summarised in Table 9-12.

Param	Value	Meaning
P61.13	1	On CAN loss use the fallback not the freeze value
P61.15	0	CAN reference 1 (speed reference) fallback value
P61.31	0000h	CAN Control Word 1 (start/stop flags) fallback value. Bit 0 - Not set, stop flag active Bits 1 - 15 – not required

Table 9-12. – Fallback Value Examples

9.12 WARNING ON LOSS OF CAN LINK

This function is carried out by P61.12 (described in section 4.6.13). During the commissioning process P61.12 is set to 1 by default, and the drive generates a Warning. When commissioning has been successfully completed, P61.12 may be set to either:

0 = Ignore loss

1 = Warn of loss

2 = Trip the drive.

10. APPENDIX B: CONFIGURATION TABLES

10.1 INTRODUCTION

This appendix includes a set of tables (Table 10-1 to Table 10-19) which enable a DeviceNet network to be configured. Each table includes an example of a typical configuration shown in the shaded rows of the table.

It is suggested that a user copies all the pages from this appendix and uses them to configure the required network.

It is also recommended that the completed tables are copied and retained safely as records of the network configuration.

No.	Param.	Mapped From	Fallback Value	Description
Example				
1	P61.14	63.64	0	Rx 2 Word 2 Speed ref.
1	**P61.14**			
2	P61.16			
3	P61.18			
4	P61.20			
5	P61.22			
6	P61.24			
7	P61.26			
8	P61.28			

Table 10-1. – CAN References

No.	Param.	Mapped From	Fallback Value	Description
Example				
1	**P61.30**	**63.62**	**0**	**Rx 2 Word 1**
1	P61.30			
2	P61.32			

Table 10-2. – CAN Control Words

Bit	Control Flag Number	Fallback Value	Description
Example			
0	7.000	0	Bit 0 mapped to Stop flag
0	7.000		
1	7.001		
2	7.002		
3	7.003		
4	7.004		
5	7.005		
6	7.006		
7	7.007		
8	7.008		
9	7.009		
10	7.010		
11	7.011		
12	7.012		
13	7.013		
14	7.014		
15	7.015		

Table 10-3. – Control Word 1, P61.30

Bit	Control Flag Number	Fallback Value	Description
Example			
0	**7.016**	**0**	**Bit 0 mapped to Stop flag**
0	7.016		
1	7.017		
2	7.018		
3	7.019		
4	7.020		
5	7.021		
6	7.022		
7	7.023		
8	7.024		
9	7.025		
10	7.026		
11	7.027		
12	7.028		
13	7.029		
14	7.030		
15	7.031		

Table 10-4. – Control Word 2, P61.32

No.	Param.	Destination Parameter	Scale Type	Description
1	P63.52			
2	P63.54			
3	P63.56			
4	P63.58			

Table 10-5. – DeviceNet Rx 1 Allocation

Example				
No.	**Param.**	**Destination Parameter**	**Scale Type**	**Description**
1	P63.52	61.30	Unity	Control word
2	P63.54	61.14	Speed(%)	Speed Reference
3	P63.56	0.00	Unity	Unused
4	P63.58	0.00	Unity	Unused

Table 10-6. – Example of DeviceNet Rx 1 Allocation

No.	Param.	Destination Parameter	Scale Type	Description
1	P63.62			
2	P63.64			
3	P63.66			
4	P63.68			

Table 10-7. – DeviceNet Rx 2 Allocation

No.	Param.	Destination Parameter	Scale Type	Description
1	P63.72			
2	P63.74			
3	P63.76			
4	P63.78			

Table 10-8. – DeviceNet Rx 3 Allocation

No.	Param.	Destination Parameter	Scale Type	Description
1	P63.82			
2	P63.84			
3	P63.86			
4	P63.88			

Table 10-9. – DeviceNet Rx 4 Allocation

No.	Param.	Destination Parameter	Scale Type	Description
1	P63.92			
2	P63.94			
3	P63.96			
4	P63.98			

Table 10-10. – DeviceNet Rx 5 Allocation

No.	Param.	Destination Parameter	Scale Type	Description
1	P63.02			
2	P63.04			
3	P63.06			
4	P63.08			

Table 10-11. – DeviceNet Tx 1 Allocation

No.	Param.	Destination Parameter	Scale Type	Description
1	P63.02			
2	P63.04			
3	P63.06			
4	P63.08			

Table 10-12. – DeviceNet Tx 1 Allocation

No.	Param.	Destination Parameter	Scale Type	Description
1	P63.12			
2	P63.14			
3	P63.16			
4	P63.18			

Table 10-13. – DeviceNet Tx 2 Allocation

No.	Param.	Destination Parameter	Scale Type	Description
1	P63.22			
2	P63.24			
3	P63.26			
4	P63.28			

Table 10-14. – DeviceNet Tx 3 Allocation

No.	Param.	Destination Parameter	Scale Type	Description
1	P63.32			
2	P63.34			
3	P63.36			
4	P63.38			

Table 10-15. – DeviceNet Tx 4 Allocation

No.	Param.	Destination Parameter	Scale Type	Description
1	P63.42			
2	P63.44			
3	P63.46			
4	P63.48			

Table 10-16. – DeviceNet Tx 5 Allocation

No.	Scaler	Drive Scale	External Scale	Description
Example				
1	**Speed(%)**	**10000**	**16383**	
1	Speed(%)			
2	Speed(rpm)			
3	Torque(%)			
4	Torque(Nm)			
5	Frequency			
6	Percent			
7	Speed Ramp Rate			
8	Torque Slew Rate			
9	Current			
10	Volts			
11	Resistance			
12	Inductance			
13	Scaler 1			
14	Scaler 2			
15	Scaler 3			

Table 10-17. – Scaler Allocation Menu 60

Bit	Param. No.	Control Flag No.	Description
Example			
0	**P41.00**	**2.001**	**Bit 0 mapped to Stopped flag**
0	P41.00		
1	P41.01		
2	P41.02		
3	P41.03		
4	P41.04		
5	P41.05		
6	P41.06		
7	P41.07		
8	P41.08		
9	P41.09		
10	P41.10		
11	P41.11		
12	P41.12		
13	P41.13		
14	P41.14		
15	P41.15		

Table 10-18. – Programmable Status Word 0

Bit	Param. No.	Control Flag No.	Description
Example			
0	**P41.16**	**2.001**	**Bit 0 mapped to Stopped flag**
0	P41.16		
1	P41.17		
2	P41.18		
3	P41.19		
4	P41.20		
5	P41.21		
6	P41.22		
7	P41.23		
8	P41.24		
9	P41.25		
10	P41.26		
11	P41.27		
12	P41.28		
13	P41.29		
14	P41.30		
15	P41.31		

Table 10-19. – Programmable Status Word 1

No.	Param.	Points To Param.	Scale	Function
Example				
1	**P42.00**	**61.14**	**10000**	**Speed Reference**
1	P42.00			Speed Reference
2	P42.02			Speed Reference
3	P42.04			Reference Sequencer
4	P42.06			PID Set-point
5	P42.08			PID Feedback
6	P42.10			Trim Reference
7	P42.12			Speed Trim Reference
8	P42.14			Torque Reference
9	P42.16			Torque Limits
10	P42.18			Torque Limits
11	P42.20			Temperature Compensation Scale
12	P42.22			Flux Limit
13	P42.24			Current Limit
14	P42.26			Torque/Magnet. Current
15	P42.28			Torque/Magnet. Current
16	P42.30			Position Reference
17	P42.32			Position Reference
18	P42.34			Tacho Feedback

Table 10-20. – Menu 42 – Reference Pointers

11. APPENDIX C

11.1 SUBJECT INDEX

This index applies to the English Edition of the T2013 Technical Manual for the MV3000e CANopen Fieldbus Facility for MV3000e Drives. The index does not include parameters. Terminology, which is listed separately in this manual, is referenced in the subject index.

The index is prepared with word-by-word alphabetisation and is presented with page numbers for subject location.

A

C

D

E

F

11.2 PARAMETER INDEX

F

I

L

N

R

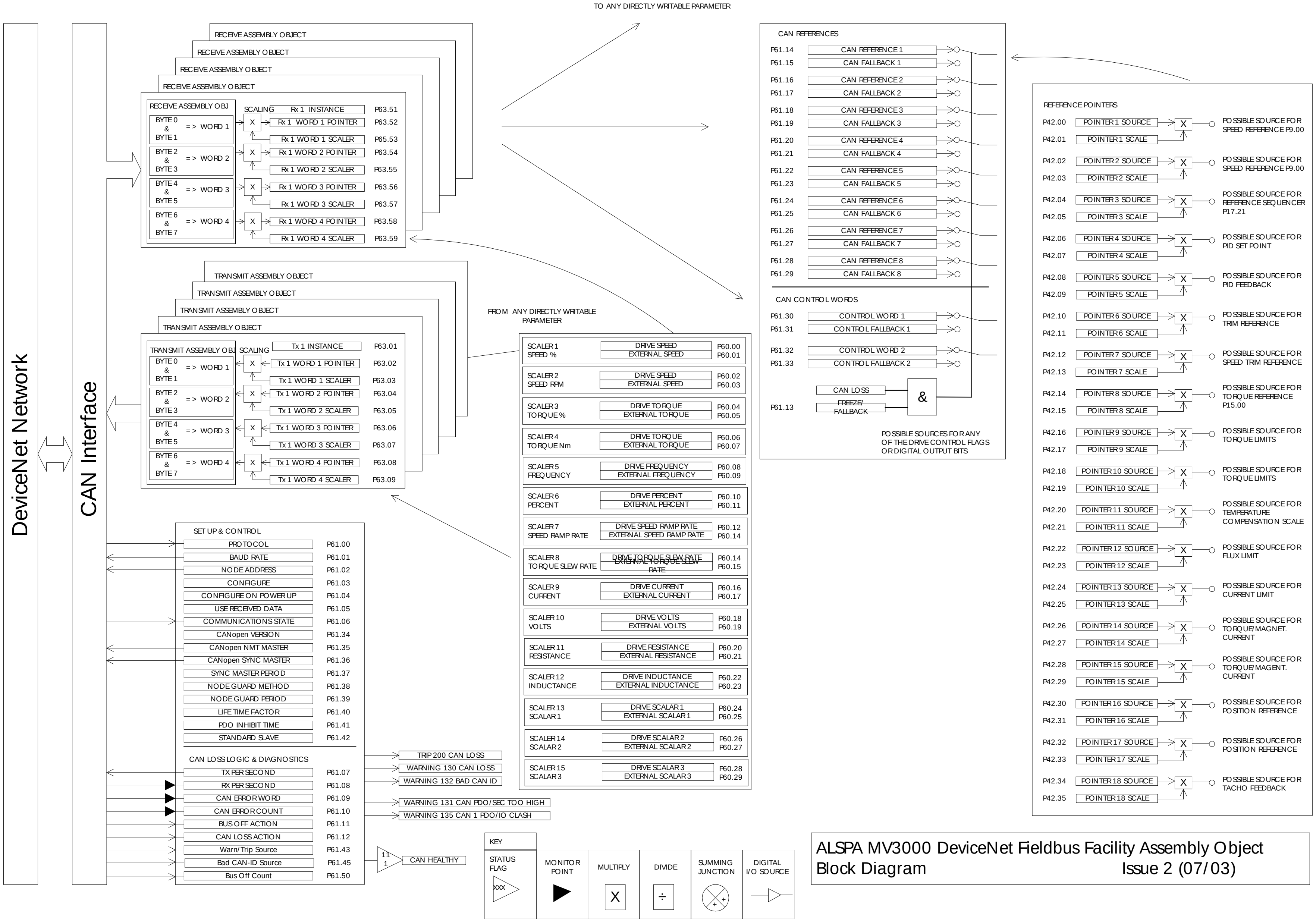
S

T

U

W

12. APPENDIX D - BLOCK DIAGRAM & TOPOLOGY



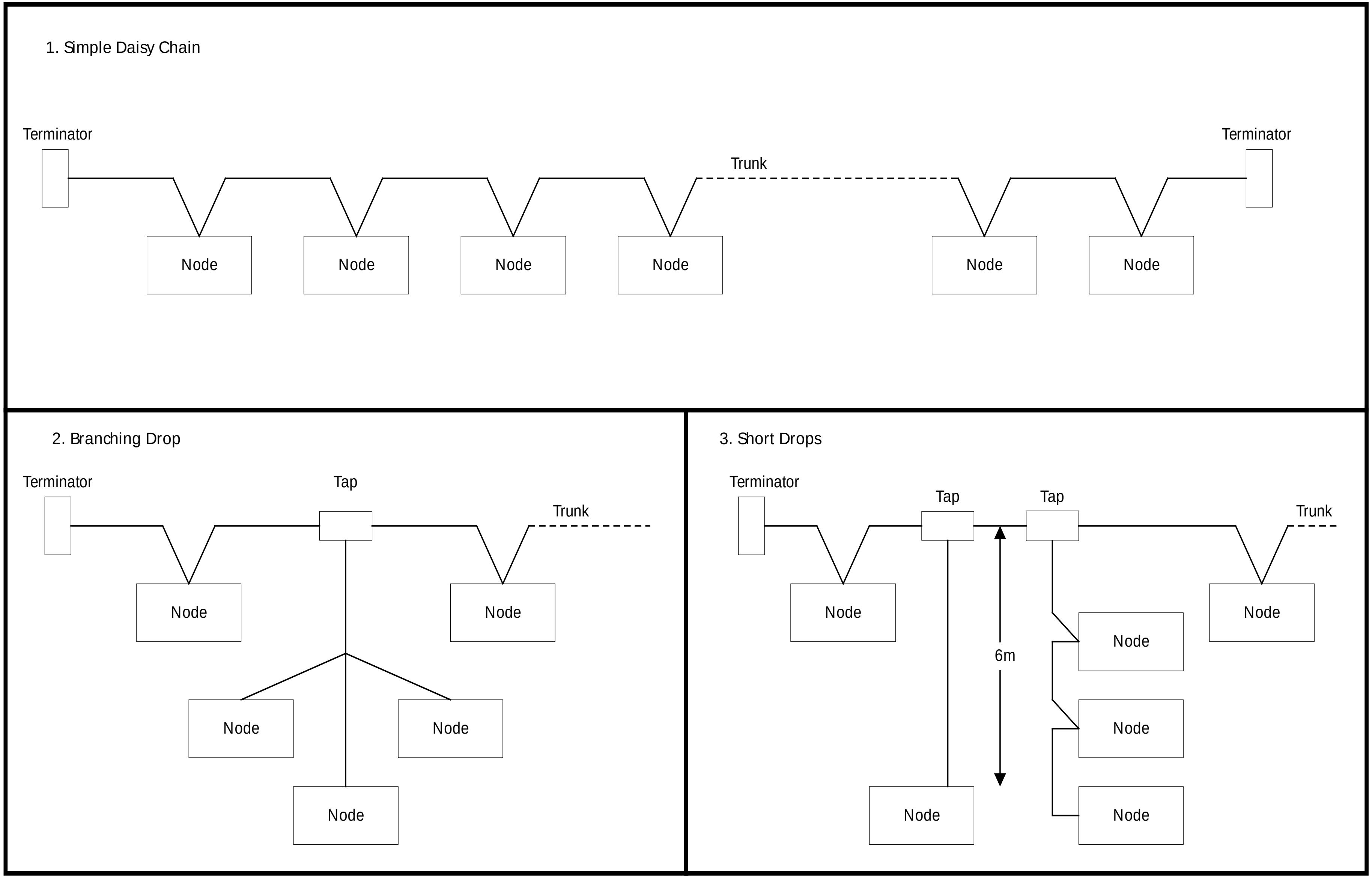


Figure 1. Network Topologies

13. CONTACT DETAILS FOR SALES, SERVICE & SUPPORT

www.avidcontrolsinc.com

Please refer to your local technical support centre if you have any queries about this product.

Technical Support Center

USA
Avid Controls, Inc.
41261 Park 290 Dr
Waller, TX 77484 USA
Tel: +1(281)640-8600
Fax: +1(281)640-8605

For all parts inquiries:
Email: info@avidcontrolsinc.com