Instruction Manual

Modbus RTU

VLT® 5000

VLT® 6000 HVAC

Danfoss

Danfoss Graham

Modbus RTU Option Card

VLT®6000 Adjustable Frequency Drive

Instruction Manual

23-6130-00
Revision B 02/03

DANGER

Rotating shafts and electrical equipment can be hazardous. Therefore, it is strongly recommended that all electrical work conform to National Electrical Code (NEC) and all local regulations. Installation, start-up and maintenance should be performed only by qualified personnel.

Motor control equipment and electronic controls are connected to hazardous line voltages. When servicing drives and electronic controls, there will be exposed components at or above line potential. Extreme care should be taken to protect against shock. Stand on an insulating pad and make it a habit to use only one hand when checking components. Always work with another person in case of an emergency. Disconnect power whenever possible to check controls or to perform maintenance. Be sure equipment is properly grounded. Wear safety glasses whenever working on electric control or rotating equipment.

WARNING

Warnings Against Unintended Start

1. While the drive is connected to the AC line, the motor can be brought to a stop by means of external switch closures, serial bus commands or references. If personal safety considerations make it necessary to ensure that no unintended start occurs, these stops are not sufficient.
2. During programming of parameters, the motor may start. Be certain that no one is in the area of the motor or driven equipment when changing parameters.
3. A motor that has been stopped may start unexpectedly if faults occur in the electronics of the drive, or if an overload, a fault in the supply AC line or a fault in the motor connection or other fault clears.
4. If the “Local/Hand” key is activated, the motor can only be brought to a stop by means of the “Stop/Off” key or an external safety interlock.

CAUTION

Electronic components of BACLink portal are sensitive to electrostatic discharge (ESD). ESD can reduce performance or destroy sensitive electronic components. Follow proper ESD procedures during installation or servicing to prevent damage.

DANGER

Touching electrical parts may be fatal, even after equipment has been disconnected from AC line. To be sure that capacitors have fully discharged, wait 14 minutes after power has been removed before touching any internal component.

Table of Contents

Introduction

This manual provides comprehensive instructions on the installation and set up of the Modbus RTU option card for the VLT 6000 Adjustable Frequency Drive to communicate over a Modbus network.

For specific information on installation and operation of the adjustable frequency drive, refer to the *VLT 6000 Operating Instructions*.

About This Manual

This manual is intended to be used for both instruction and reference. It only briefly touches on the basics of the Modbus protocol whenever necessary to gain an understanding of the Modbus RTU option card for the VLT 6000.

This manual is also intended to serve as a guideline when you specify and optimize your communication system. Even if you are an experienced Modbus programmer, it is suggested that you read this manual in its entirety before you start programming since important information can be found in all sections.

Assumptions

This manual assumes that you have a controller that supports the interfaces in this document and that all the requirements stipulated in the controller, as well as the VLT 6000 Adjustable Frequency Drive, are strictly observed, along with all limitations therein.

What You Should Already Know

The VLT 6000 Modbus RTU option card is designed to communicate with any controller that supports the interfaces defined in this document. It is assumed that you have full knowledge of the capabilities and limitations of the controller.

References

VLT® 6000 Installation, Operation and Instruction Manual, Danfoss Graham document number 23-6108-00.
(Referred to as the *VLT 6000 Operating Instructions* in this document.)

Modbus RTU Overview

The common language used by all Modicon controllers is the Modbus RTU (Remote Terminal Unit) protocol. This protocol defines a message structure that controllers will recognize and use, regardless of the type of networks over which they communicate. It describes the process a controller uses to request access to another device, how it will respond to requests from the other devices, and how errors will be detected and reported. It establishes a common format for the layout and contents of message fields.

During communications on a Modbus RTU network, the protocol determines how each controller will know its device address, recognize a message addressed to it, determine the kind of action to be taken, and extract any data or other information contained in the message. If a reply is required, the controller will construct the reply message and send it.

Controllers communicate using a master-slave technique in which only one device (the master) can initiate transactions (called *queries*). The other devices (slaves) respond by supplying the requested data to the master, or by taking the action requested in the query.

The master can address individual slaves, or can initiate a broadcast message to all slaves. Slaves return a message (called a *response*) to queries that are addressed to them individually. Responses are not returned to broadcast queries from the master.

The Modbus RTU protocol establishes the format for the master's query by placing into it the device (or broadcast) address, a function code defining the requested action, any data to be sent, and an error-checking field. The slave's response message is also constructed using Modbus protocol. It contains fields confirming the action taken, any data to be returned, and an error-checking field. If an error occurred in receipt of the message, or if the slave is unable to perform the requested action, the slave will construct an error message and send it in response.

VLT 6000 with Modbus RTU Option Overview

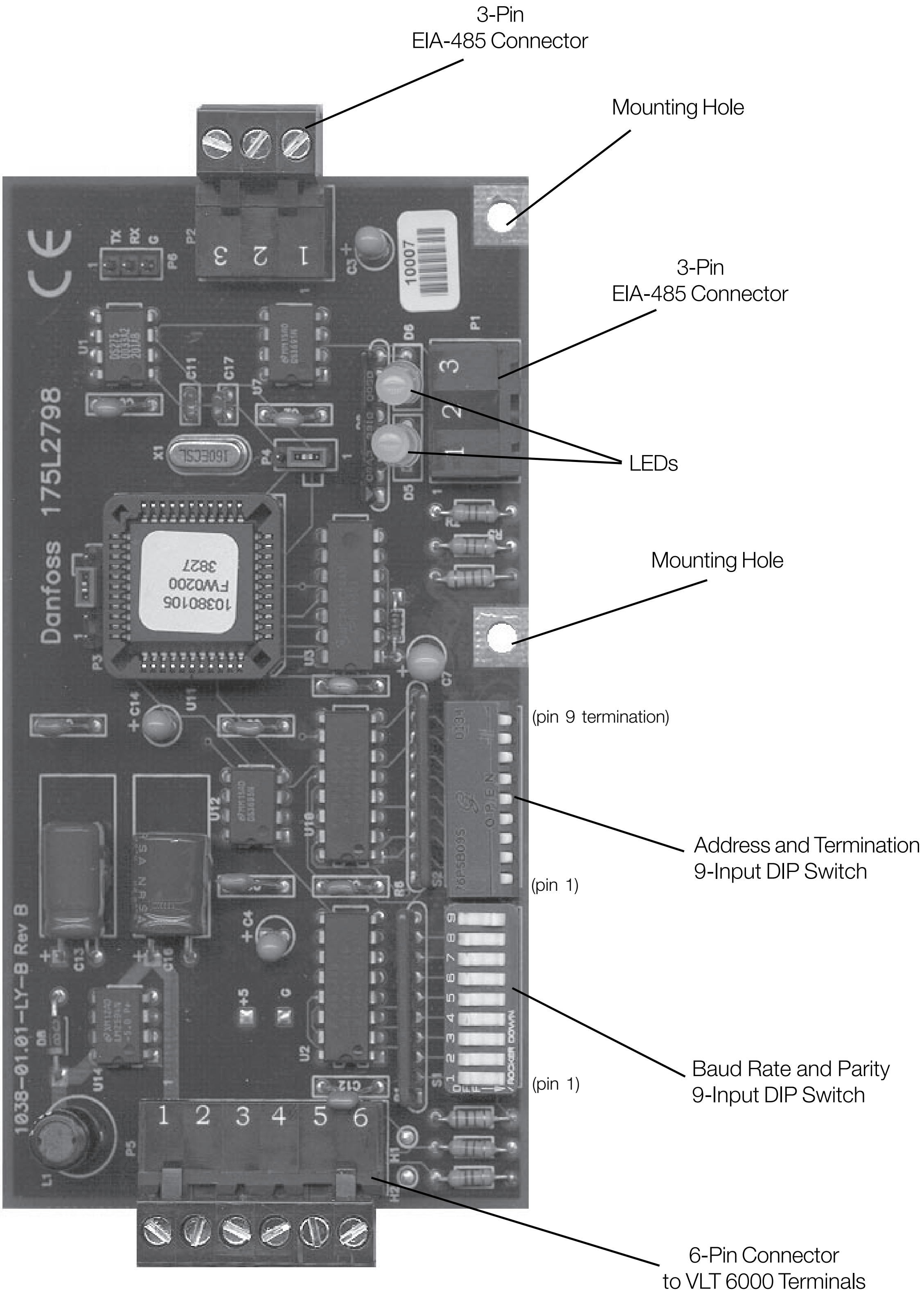
The VLT 6000 Adjustable Frequency Drive with the Modbus RTU option card installed communicates in Modbus RTU format over an EIA-485 (formerly RS-485) network. The option card acts as a translator between the drive's internal FC protocol and Modbus RTU. This allows access to the drive's Control Word and Bus Reference.

The Control Word allows Modbus to control several important functions of the drive:

- Start
- Stop the drive in several ways:
  - Coast stop
  - Quick stop
  - DC Brake stop
  - Normal (ramp) stop
- Reset after a fault trip
- Run at a variety of preset speeds
- Run in reverse
- Change the active setup
- Control the drive's two built-in relays

The Bus Reference is commonly used for speed control.

It is also possible to access the drive parameters, read their values, and, where possible, write values to them. This permits a range of control possibilities, including controlling the drive's setpoint when its internal PID controller is used.



VLT 6000 Modbus RTU Option Card

Modbus RTU Option Card Baud Rate and Parity Settings

The Modbus RTU communication protocol accesses the internal Danfoss FC protocol within the VLT 6000 to control the drive through serial communications. (The Modbus-to-FC interface uses 9600 Baud, 8 Bits, Even Parity, and 1 Stop Bit.)

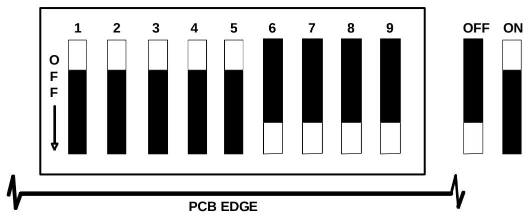
The Modbus RTU option card has a 9-input DIP switch for setting baud rate and parity (see figure below). The option card generally uses 9600 baud rate with no parity. Set the switch positions in accordance with the following instructions.

- Set input switches 1-3 to ON to select 9600 baud rate.
- Set input switches 4 and 5 to ON to select no parity.

Switches 6-9 are unassigned reserved switches. Their setting does not matter.

NOTE
Set baud rate and parity switch settings prior to installing Modbus RTU option card for ease of access.

For Modbus RTU networks operating at other than 9600 baud and no parity, determine switch positions from the tables provided below.



9600 Baud Rate and No Parity Switch Settings

Baud Rate	SW1	SW2	SW3
300	OFF	OFF	OFF
1200	OFF	OFF	ON
2400	OFF	ON	OFF
4800	OFF	ON	ON
9600	ON	ON	ON
19200	ON	OFF	ON

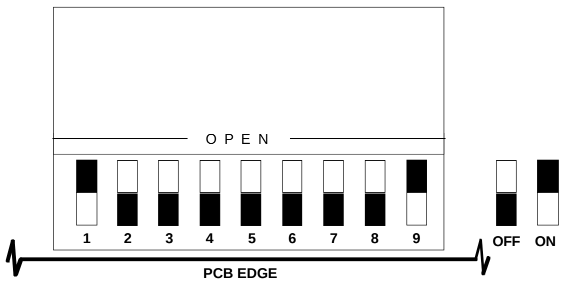
Parity	SW4	SW5
None	ON	ON
Odd	OFF	ON
Even	OFF	OFF

Optional Baud Rate and Parity Switch Settings

Modbus RTU Option Card Network Address Settings

The Modbus RTU option card has an address and termination 9-input DIP switch. The Modbus network address for the VLT 6000 is set by DIP switch positions on the switch. Pin 9 is an ON/OFF switch for network termination. DIP switch positions are read on power-up only, so position changes will not be recognized until the next power-up.

- Set the Modbus address for the VLT 6000 in accordance with the table below. The default input setting is for ADDRESS 1 and termination ON.



Address and Termination

Address (Hex)	SW1 2^0	SW2 2^1	SW3 2^2	SW4 2^3	SW5 2^4	SW6 2^5	SW7 2^6	SW8 2^7	TERM
Default 01	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
55	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON
AA	OFF	ON	OFF	ON	OFF	ON	OFF	ON	ON
F7	ON	ON	ON	OFF	ON	ON	ON	ON	ON

Address Input Selection

VLT 6000 Parameter Settings

The Modbus RTU option card interface to the VLT 6000 Adjustable Frequency Drive FC protocol requires drive parameter values selected as shown. They are the default settings for those parameters and probably require no change to operate the drive using Modbus. The Modbus RTU option card always transmits to the drive in which it resides as address one (001). See the *VLT 6000 Operating Instructions* for details on selecting and changing parameter values, if necessary.

- Parameter 500, *Protocol*: FC protocol
- Parameter 501, *Address*: 001
- Parameter 502, *Baud Rate*: 9600 baud

Modbus RTU Option Card Environmental Requirements

Environmental requirements for the Modbus RTU option card are listed below.

Description	Requirement
Operating temperature	-5° F to +140° F (-20° C to +60° C)
Storage temperature	-40° F to +176° F (-40° C to + 80° C)
Humidity	5% to 95% relative, non-condensing

Power Supply

The Modbus RTU option card is powered by 24 VDC and draws 38 mA of current at 24V.

Installation

The following section describes the installation procedures for the Modbus RTU option card. For additional information on installation and operation of the VLT 6000, refer to the *VLT 6000 Operating Instructions*.

⚠DANGER

VLT adjustable frequency drive contains dangerous voltages when connected to line voltage. After disconnecting from power line, wait at least 14 minutes before touching any electrical components.

⚠WARNING

Only a competent electrician should carry out electrical installation. Improper installation of motor or drive can cause equipment failure, serious injury or death. Follow this manual, National Electrical Codes and local safety codes.

⚠CAUTION

Electronic components of adjustable frequency drive and Modbus RTU option card are sensitive to electrostatic discharge (ESD). ESD can reduce performance or destroy sensitive electronic components. Follow proper ESD procedures during installation or servicing to prevent damage.

⚠CAUTION

It is responsibility of user or installer of VLT adjustable frequency drive to provide proper grounding and motor overload and branch protection according to National Electrical Codes and local codes.

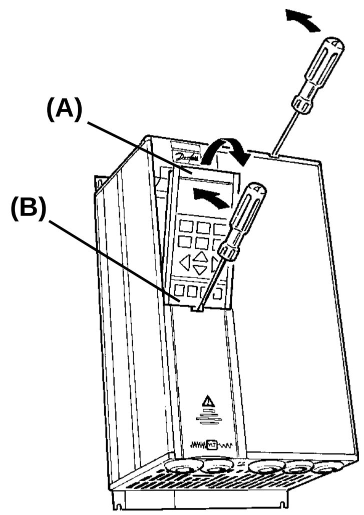
1. Access to Control Card Cassette

NEMA 1 Drives:

- Remove Local Control Panel (LCP) by pulling out from top of display (A) by hand. LCP connector on panel back will disconnect.
- Remove protective cover by gently prying with a screw driver at notch (B) and lift cover out of guide pin fittings.

NEMA 12 Drives:

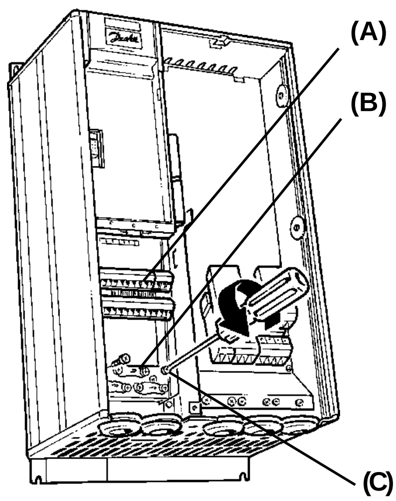
- Open front panel of drive by loosening captive screws and swing open.
- Disconnect Local Control Panel (LCP) cable.



---

2. Disconnect Control Card Cassette

- Remove control wiring by unplugging connector terminals (A).
- Remove grounding clamps (B) by removing two screws holding each in place. Save screws for reassembly.
- Loosen two captive screws (C) securing cassette to chassis.

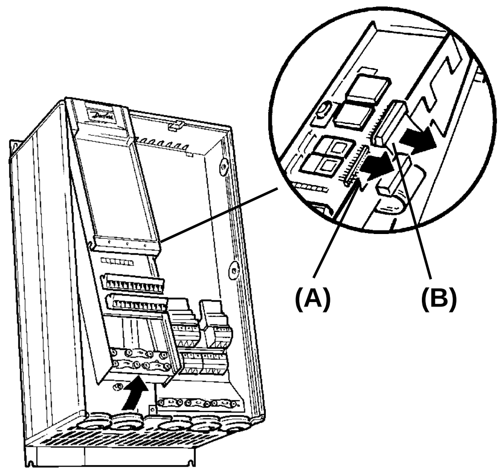


---

3. Remove Cassette and Ribbon Cables

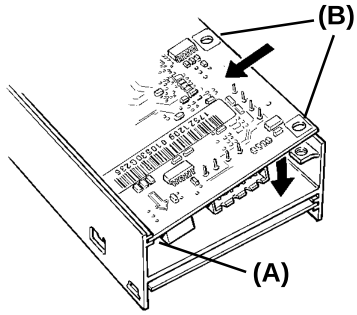
- Lift control card cassette from bottom.
- Unplug two ribbon cables (A) and (B) from VLT 6000 control board.
- Unhinge cassette at top to remove.

NOTE
Ribbon cables will need to be reconnected to same connections from which removed.



4. Secure Modbus RTU Option Card

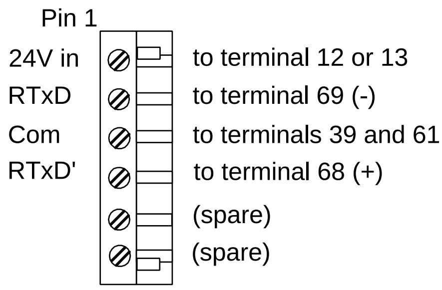
- On back of cassette, insert edge of Modbus RTU option card into slot at side of cassette (A).
- Secure opposite side of card with 2 self-tapping screws and washers provided (B). Using a Torx T-10 screw driver, tighten to 8 in-lbs.



5. Wire Modbus RTU Option Card Connector to VLT 6000 Terminals

NOTE
Use 18 to 22 gauge wire. Torque terminals to 4.5 in-lbs. Modbus interface connector terminals 5 and 6 are spares.

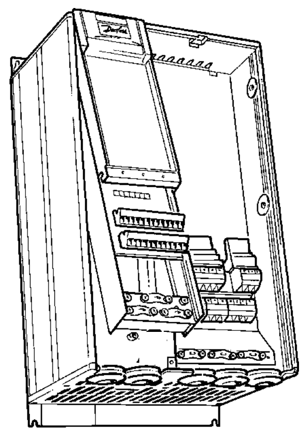
- Wire Modbus interface connector (24 V power) pin 1 to VLT 6000 terminal 12 or 13.
- Wire Modbus interface connector (RTxD) pin 2 to VLT 6000 terminal 69.
- Wire Modbus interface connector (com) pin 3 to VLT 6000 terminals 39 and 61.
- Wire Modbus interface connector (RTxD') pin 4 to VLT 6000 terminal 68.
- Plug Modbus interface connector into bottom of Modbus RTU option card.



6. Install Ribbon Cables

NOTE
Ribbon cables must be reconnected to same connections from which removed.

- Connect control card cassette to hinge at top of drive.
- Connect ribbon cables.



7. Install Control Card Cassette

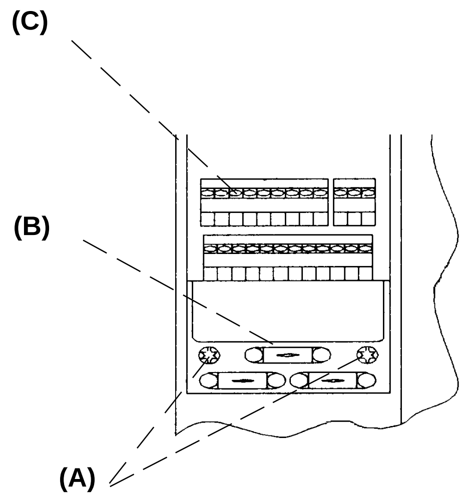
- Fasten control card cassette by alternately tightening two captive screws (A). Tighten to 8 in-lbs.
- Route control wires through clamp fasteners (B) and secure clamps with two screws.
- Connect control terminals (C) by firmly pressing them into connector receptacles.

NEMA 1 Drives

- Install LCP by sliding bottom into guide slots on cradle, then press into place ensuring that connector on back of LCP is engaged.
- Replace protective cover by positioning guide pins at bottom of cover into holes in bottom of chassis and snap top of cover into place.

NEMA 12 Drives

- Plug cable from LCP into connector on main control card.

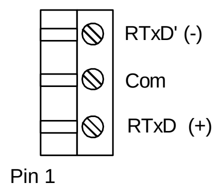


8. Plug in Terminal Connector

- Connect Modbus signal wire (RTxD+) to pin 1 of EIA-485 (formerly RS-485) connector.
- Connect Modbus signal wire (Com) to pin 2 of EIA-485 connector.
- Connect Modbus signal wire (RTxD-) to pin 3 of EIA-485 connector.
- Plug EIA-485 connector into terminal port at side of Modbus RTU option card.

NEMA 12 Drives

- Close front cover panel and fasten with captive screws. Tighten to 7 in-lbs.

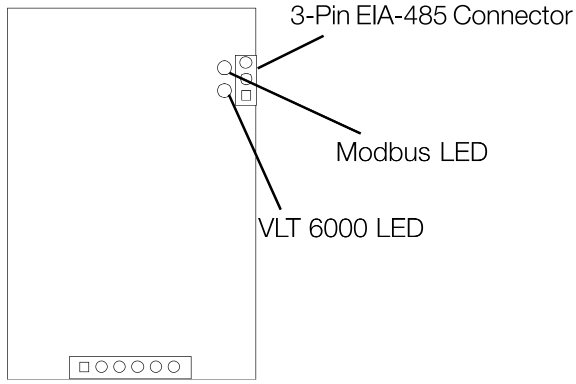


EIA-485 Connector

Status LEDs

The Modbus RTU option card has two LEDs. One LED is used as a status for Modbus RTU communications and the other as a status for VLT 6000 drive communications. Both LEDs use the same communications patterns. On power up, each LED state is flashed on for 250 milliseconds (Red, Green, Orange, Off). The VLT 6000 LED powers up first, then the Modbus LED. After power up, the following are the only valid states:

- Flashing Green (1 Hz): Communications online (VLT 6000 LED) or receiving data (Modbus LED)
- Flashing Red (1 Hz): Communications time out
- Solid Red: Fault, communications halted



Option Card Operability Loop Back Test

A loop back test to confirm Modbus RTU option card operability can be performed. The option card must be removed from the adjustable frequency drive to gain access to the 9-input dip switch for baud rate and parity and to rewire the option card connectors.

⚠DANGER

Ensure that power has been removed from adjustable frequency drive for a minimum of 14 minutes to allow voltage to dissipate.

- Remove the option card in accordance with the procedures described in the installation section of this manual.
- Set the dip switch positions in accordance with the table below.
- Remove all wiring from both the 6-pin option card connector and the 3-pin EIA-485 connector.
- Wire the 6-pin option card connector to the 3-pin EIA-485 connector as described below.

6-Pin Connector		EIA-485 Connector
Pin 2	to	Pin 3
Pin 3	to	Pin 2
Pin 4	to	Pin 1

- Apply power to the unit.

After the normal status LED check at power-up (see Status LEDs), the loop back test sets both LEDs to orange for a successful test or red if the test fails. The orange or red indicator lasts around 10-15 seconds. The LEDs will then flash red or green communication indications.

SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8	SW9
ON	ON	ON	OFF	OFF	ON	ON	ON	ON

Loop Back Test Switch Positions

Remote Terminal Unit

The controllers are setup to communicate on the Modbus network using RTU (Remote Terminal Unit) mode, with each 8-bit byte in a message contains two 4-bit hexadecimal characters. The format for each byte is shown below.

Coding System: 8-bit binary, hexadecimal 0-9, A-F
Two hexadecimal characters contained in each 8-bit field of the message

Bits Per Byte: 1 start bit
8 data bits, least significant bit sent first
1 bit for even/odd parity; no bit for no parity
1 stop bit if parity is used; 2 bits if no parity

Error Check Field: Cyclical Redundancy Check (CRC)

Modbus RTU Message Framing Structure

A Modbus RTU message is placed by the transmitting device into a frame with a known beginning and ending point. This allows receiving devices to begin at the start of the message, read the address portion, determine which device is addressed (or all devices, if the message is broadcast), and to know when the message is completed. Partial messages are detected and errors set as a result.

The allowable characters transmitted for all fields are hexadecimal 0-9, A-F. The adjustable frequency drives monitor the network bus continuously, including 'silent' intervals. When the first field (the address field) is received, each drive or device decodes it to determine whether it is the addressed device.

Modbus RTU messages addressed to zero are converted to broadcast messages using the FC protocol. No response is needed on broadcast messages.

To ensure the attribute data returned is the most current, each attribute access must include one attribute only.

A typical message frame is shown below.

Start	Address	Function	Data	CRC Check	End
T1-T2-T3-T4	8 Bits	8 Bits	*n x* 8 Bits	16 Bits	T1-T2-T3-T4

Typical Modbus RTU Message Structure

Modbus RTU Message Framing Structure (continued)

Start/Stop Field

Messages start with a silent interval of at least 3.5 character times. This is implemented as a multiple of character times at the 9600 network baud rate (shown as Start T1-T2-T3-T4). The first field then transmitted is the device address. Following the last transmitted character, a similar interval of at least 3.5 character times marks the end of the message. A new message can begin after this interval.

The entire message frame must be transmitted as a continuous stream. If a silent interval of more than 1.5 character times occurs before completion of the frame, the receiving device flushes the incomplete message and assumes that the next byte will be the address field of a new message.

Similarly, if a new message begins earlier that 3.5 character times following a previous message, the receiving device will consider it a continuation of the previous message. This will set an error, since the value in the final CRC field is not valid for the combined messages.

Modbus RTU Message Framing Structure (continued)

Address Field

The address field of a message frame contains 8 bits. Valid slave device addresses are in the range of 0 – 247 decimal. The individual slave devices are assigned addresses in the range of 1 – 247. (0 is reserved for broadcast mode, which all slaves recognize.) A master addresses a slave by placing the slave address in the address field of the message. When the slave sends its response, it places its own address in this address field to let the master know which slave is responding.

Function Field

The function field of a message frame contains 8 bits. Valid codes are in the range of 1 – 255 decimal. (See Appendix A for a description of supported Modbus functions.) When a message is sent from a master to a slave device, the function code field tells the slave what kind of action to perform.

When the slave responds to the master, it uses the function code field to indicate either a normal (error-free) response, or that some kind of error occurred (called an exception response). For a normal response, the slave simply echoes the original function code. For an exception response, the slave returns a code that is equivalent to the original function code with its most-significant bit set to a logic 1. In addition, the slave places a unique code into the data field of the response message. This tells the master what kind of error occurred, or the reason for the exception. See the *Exception Codes* section in this manual for definitions.

Data Field

The data field is constructed using sets of two hexadecimal digits, in the range of 00 to FF hexadecimal. These are made from one RTU character. The data field of messages sent from a master to slave device contains additional information which the slave must use to take the action defined by the function code. This can include items like discrete and register addresses, the quantity of items to be handled, and the count of actual data bytes in the field. The data field can have a length of zero.

CRC Check Field

Messages include an error-checking field that is based on a cyclical redundancy check (CRC) method. The CRC field checks the contents of the entire message. It is applied regardless of any parity check method used for the individual characters of the message. The CRC value is calculated by the transmitting device, which appends the CRC as the last field in the message. The receiving device recalculates a CRC during receipt of the message and compares the calculated value to the actual value received in the CRC field. If the two values are not equal, an error results.

The error checking field contains a 16-bit binary value implemented as two 8-bit bytes. When this is done, the low-order byte of the field is appended first, followed by the high-order byte. The CRC high-order byte is the last byte sent in the message.

Coil/Register Addressing

All data addresses in Modbus messages are referenced to zero. The first occurrence of a data item is addressed as item number zero. For example:

The coil known as 'coil 1' in a programmable controller is addressed as coil 0000 in the data address field of a Modbus message. Coil 127 decimal is addressed as coil $007E_{HEX}$ (126 decimal).

Holding register 40001 is addressed as register 0000 in the data address field of the message. The function code field already specifies a 'holding register' operation. Therefore, the '4XXXX' reference is implicit. Holding register 40108 is addressed as register $006B_{HEX}$ (107 decimal).

Translation from Modbus RTU Protocol to FC Protocol

Refer to *Serial Communication for FC Protocol* in the *VLT 6000 Operating Instructions* for details on the Danfoss FC protocol used for Modbus RTU serial communication within the VLT 6000 Adjustable Frequency Drive.

Parameter Block

PKE

PKE contains AK with the parameter commands and replies, and PNU with the parameter number. The AK value is determined by the Modbus function code. Coil 65 decimal determines whether data written to the drive are stored in EEPROM and RAM (coil 65 = 1) or just RAM (coil 65 = 0). PNU is translated from the register address contained in the Modbus read/write message. The parameter number is translated to Modbus as $(10 \times \text{parameter number})_{DECIMAL}$.

IND

IND contains the index. The index is used, together with the parameter number, for read/write access. Index has 2 bytes – a low byte and a high byte. However, only the low byte is used for indexing. The high byte is used for reading and writing text. IND is set by a register in Modbus (40001_{HEX}). IND must be cleared by the Modbus master after reading/writing text.

PWE_{HIGh}/PWE_{LOW}

PWE contains the parameter value. The parameter value block consists of 2 words (4 bytes). The value depends on the command given (AK). PWE is zero filled on reads. On writes, PWE is filled with the data field of the Modbus write message.

PCD_1/PCD_2

PCD contains the process word block. The parameter value block consists of 2 words (4 bytes). The process word block is divided into two blocks of 16 bits and is stored in Modbus as status coils. The mapping of the PCD is shown below.

Process Block Updates

Upon every write to the PCD coils, the process block is written to the drive and returned from the drive. On parameter reads and writes, the PCD is deactivated on messages from the Modbus RTU option card to the drive. The PCD coils are updated on response messages from the drive to the Modbus RTU option card.

Text Blocks

Parameters stored as text strings are accessed the same as the other parameters except PWE is replaced with the text block. The maximum text block size is 20 characters. If a read request for a parameter is for more characters than the parameter stores, the response is space filled. If the read request for a parameter is for less characters than the parameter stores, the response is truncated.

	PCD_1	PCD_2
Control packet (master → slave)	Control word (Coils 1 – 16)$_{DEC}$	Reference value (Coils 17 – 32)$_{DEC}$
Reply packet (slave → master)	Status word (Coils 33 – 48)$_{DEC}$	Given output frequency (Coils 49 – 64)$_{DEC}$

PCD Mapping

FC Protocol Control Word Bit Descriptions

Control Word Bit Descriptions

Bit	0	1
00	Preset Ref. LSB	
01	Preset Ref. MSB	
02	DC Brake	no DC Brake
03	Coast Stop	no Coast Stop
04	"Quick" Stop	no "Quick" Stop
05	Freeze Freq.	no Freeze Freq.
06	Ramp Stop	Start
07	no Reset	Reset
08	no Jog	Jog
09	no function	
10	Data Not Valid	Data Valid
11	Relay 1 OFF	Relay 1 ON
12	Relay 2 OFF	Relay 2 ON
13	Setup LSB	
14	Setup MSB	
15	no Reversing	Reversing

Stop Commands

The precedence of the stop commands is as follows:

1. Coast stop
2. Quick stop
3. DC Brake
4. Normal (Ramp) stop

Conversion Factor

Conversion

The different attributes for each parameter can be seen in the section on factory settings. Since a parameter value can only be transferred as a whole number, a conversion factor must be used to transfer decimals.

Example:
Parameter 201: *Minimum Frequency*, conversion factor 0.1. If parameter 201 is to be set to 10 Hz, a value of 100 must be transferred, since a conversion factor of 0.1 means that the transferred value will be multiplied by 0.1. A value of 100 will, therefore, be understood as 10.0.

Index	Factor
74	3.6
2	100.0
1	10.0
0	1.0
-1	0.1
-2	0.01
-3	0.001
-4	0.0001

Memory Mapping

Parameter Values

Standard Data Types
Standard data types are int16, int32, uint8, uint16 and uint32. They are stored as 4x registers (40001 – 4FFFF). The parameters are read using function 03_{HEX} "Read Holding Registers." Parameters are written using function 6_{HEX} "Preset Single Register" for 1 register (16 bits), and function 10_{HEX} "Preset Multiple Registers" for 2 registers (32 bits). Valid sizes to read are 1 register (16 bits) and 2 registers (32 bits).

Nonstandard Data Types
Nonstandard data types are text strings and are stored as 4x registers (40001 – 4FFFF). The parameters are read using function 03_{HEX} "Read Holding Registers" and written using function 10_{HEX} "Preset Multiple Registers." Valid sizes to read are 1 register (2 characters) through 10 registers (20 characters). See *Text Blocks* section in this manual for truncation/ space fill rules. IND (Modbus Register 1) must be written with a value of 0400_{HEX} (read) or 0500_{HEX} (write) prior to reading or writing a text string.

Memory Mapping: Status Coil Maps

Status Coils Map (128 coils total)

Address (Decimal)	Description
1 – 16	PCD_1 Control word (master → slave)
17 – 32	PCD_2 Reference value (master → slave)
33 – 48	PCD_1 Status word (slave → master)
49 – 64	PCD_2 Given output frequency (slave → master)
65	Write memory storage type bit (used with AK), see PKE
66 – 128	Reserved

Memory Mapping: Register Maps

Register Maps (65536 registers total)

Address (Decimal)	Description
00001	IND (index word)
00002	Modbus Communications Timeout Value (10 millisecond units)
00003	Drive Communications Timeout Value (10 millisecond units)
00004 – 00009	Reserved
00010 ↓ 00170	Parameter 001, *Language* ↓ Parameter 017, *Operating State at Power-up*
00180 – 09999	Reserved
01000 ↓ 01170	Parameter 100, *Configuration* ↓ Parameter 117, *Motor Thermal Protection*
01180 – 01999	Reserved
02000 ↓ 02280	Parameter 200, *Output Frequency Range* ↓ Parameter 228, *Warning: High Feedback*
02290 – 02999	Reserved
03000 ↓ 03280	Parameter 300, *Terminal 16 Digital Input* ↓ Parameter 328, *Pulse Feedback, Max. Freq.*
03290 – 03999	Reserved
04000 ↓ 04270	Parameter 400, *Reset Function* ↓ Parameter 427, *PID Lowpass Filter Time*
04280 – 04999	Reserved
05000 ↓ 05660	Parameter 500, *Protocol* ↓ Parameter 566, *FLN Time Function*
05670 – 05999	Reserved
06000 ↓ 06310	Parameter 600, *Operating Data: Operating Hours* ↓ Parameter 631, *Nameplate: Communication Option Ordering No.*
06320 – 06999	Reserved
07000 ↓ 07110	Parameter 700, *Relay 6, Output Function* ↓ Parameter 711, *Relay 9, Off Delay*
07120 – 65536	Reserved

Message Translation Examples

EXAMPLE 1: Start Motor, Run Speed 40%

Modbus function $0F_{HEX}$ (*Force Multiple Coils*).

Message sent to Modbus RTU option card from Modbus master

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Slave Address	Function	Coil Addr HI	Coil Addr LO	# of Coils HI	# of Coils LO	Byte Count	Force Data HI Coils (0-7)
01	0F	00	00	00	20	04	7C

Byte 8	Byte 9	Byte 10	Byte 11
Force Data LO Coils (8-15)	Force Data HI Coils (16-23)	Force Data LO Coils (24-31)	Error Check
04	99	19	[37] [43]

Modbus message string:
[01] [0F] [00] [00] [00] [20] [04] [7C] [04] [99] [19] [37] [43]

Start Command: 0000010001111100 = $047C_{HEX}$ (reversed)
(see *FC Protocol Control Word Bit Descriptions*)

Modbus message string:
[01] [0F] [00] [00] [00] [20] [04] [7C] [04] [99] [19] [37] [43]

NOTE: Speed Command: 4000_{HEX} = 100% speed
40% of 4000_{HEX} = 1999_{HEX} (reversed)

Message returned to Modbus master from Modbus RTU option card

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Slave Address	Function	Coil Addr HI	Coil Addr LO	No. of Coils HI	No. of Coils LO	Error Check
01	0F	00	00	00	20	[54] [13]

All values are in hexadecimal.

Message Translation Examples (continued)

EXAMPLE 2: Ramp Stop Motor

Message sent to Modbus RTU option card from Modbus master

Byte 0 Slave Address	Byte 1 Function	Byte 2 Coil Addr HI	Byte 3 Coil Addr LO	Byte 4 # of Coils HI	Byte 5 # of Coils LO	Byte 6 Byte Count	Byte 7 Force Data HI Coils (0-7)
01	0F	00	00	00	20	04	3C

Byte 8 Force Data LO Coils (8-15)	Byte 9 Force Data HI Coils (16-23)	Byte 10 Force Data LO Coils (24-31)	Byte 11 Error Check
04	00	00	[89] [19]

Modbus message string:
[01] [0F] [00] [00] [00] [20] [04] [3C] [04] [00] [00] [89] [19]

Stop Command: 0000010000111100 = $043C_{HEX}$ (reversed)
(see *FC Protocol Control Word Bit Descriptions*)

Speed Command: 0%

Message returned to Modbus master from Modbus RTU option card

Byte 0 Slave Address	Byte 1 Function	Byte 2 Coil Addr HI	Byte 3 Coil Addr LO	Byte 4 No. of Coils HI	Byte 5 No. of Coils LO	Byte 6 Error Check
01	0F	00	00	00	20	--

All values are in hexadecimal.

EXAMPLE 3: Coast Stop Motor

Message sent to Modbus RTU option card from Modbus master

Byte 0 Slave Address	Byte 1 Function	Byte 2 Coil Addr HI	Byte 3 Coil Addr LO	Byte 4 # of Coils HI	Byte 5 # of Coils LO	Byte 6 Byte Count	Byte 7 Force Data HI Coils (0-7)
01	0F	00	00	00	20	04	20

Byte 8 Force Data LO Coils (8-15)	Byte 9 Force Data HI Coils (16-23)	Byte 10 Force Data LO Coils (24-31)	Byte 11 Error Check
2C	00	00	--

Modbus message string:
[01] [0F] [00] [00] [00] [20] [04] [20] [2C] [00] [00] [--]

Coast Command: 0010110000100000 = $2C20_{HEX}$ (reversed)
(see *FC Protocol Control Word Bit Descriptions*)

Speed Command: 0%

Message returned to Modbus master from Modbus RTU option card

Byte 0 Slave Address	Byte 1 Function	Byte 2 Coil Addr HI	Byte 3 Coil Addr LO	Byte 4 No. of Coils HI	Byte 5 No. of Coils LO	Byte 6 Error Check
01	0F	00	00	00	20	--

Message Translation Examples (continued)

EXAMPLE 4: Write Parameter 104,
***Motor Frequency*, with 60 Hz**
(Data Type 6: UINT16)
(Conversion factor = 0)

Modbus Function 06_{HEX} *Preset Single Register*

Message sent to Modbus RTU option card from Modbus master

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Slave Address	Function	Register Addr HI	Register Addr LO	Preset Data HI	Preset Data LO	Error Check
01	06	04	0F	00	3C	--

Modbus message string:
[01] 06] [04] [0F] [00] [3C] [error check]

Parameter 104 = $0F04_{HEX}$ (reversed)
Note that the starting address of a register is the parameter number x 10 -1 in HEX.
104 x 10 = 1040 -1 = 1039 = $0F04_{HEX}$ (reversed)

Modbus message string:
[01] 06] [04] [0F] [00] [3C] [error check]

Speed (60 Hz) = $3C_{HEX}$

Message returned to Modbus master from Modbus RTU option card

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Slave Address	Function	Register Addr HI	Register Addr LO	Preset Data HI	Preset Data LO	Error Check
01	06	04	0F	00	3C	--

All values are in hexadecimal.

EXAMPLE 5: Read Parameter 514,
***Motor Current* = 3 Amps**
(Data Type 7: UINT32)
(Conversion Factor = -2)

Modbus Function 03_{HEX} *Read Holding Registers*

Message sent to Modbus RTU option card from Modbus master

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Slave Address	Function	Start Addr HI	Start Addr LO	No. of Points HI	No. of Points LO	Error Check
01	03	14	13	00	02	--

Parameter 514 (5139) = 1413_{HEX}
Note that the starting address of a register is the parameter number x 10 -1 in HEX.

Message sent to Modbus master from Modbus RTU option card

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Slave Address	Function	Byte Count	Data HI (Reg 45140)	Data LO (Reg 45140)	Data HI (Reg 45141)	Data LO (Reg 45141)	Error Check
01	03	04	00	00	01	2C	--

All values are in hexadecimal.

Message Translation Examples (continued)

EXAMPLE 6: Write Parameter 533, *Display Text 1*, (VLT 6000 only) with "1234567890" (Data Type 9: Text String).

Modbus Function 06_{HEX} *Preset Single Register*

Write IND with "0500" to perform a text write.

Message sent to Modbus RTU option card from Modbus master

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Slave Address	Function	Register Addr HI	Register Addr LO	Preset Data HI	Preset Data LO	Error Check
01	06	00	00	05	00	--

Message sent to Modbus master from Modbus RTU option card

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Slave Address	Function	Register Addr HI	Register Addr LO	Preset Data HI	Preset Data LO	Error Check
01	06	00	00	05	00	--

Commands Modbus to text mode.

Modbus Function 10_{HEX} *Preset Multiple Registers*

Message sent to Modbus RTU option card from Modbus master

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Slave Address	Function	Start Addr HI	Start Addr LO	No. of Registers HI	No. of Registers LO	Byte Count	Data HI (Reg 414D2)
01	10	14	D1	00	05	0A	31

Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13	Byte 14	Byte 15
Data HI (Reg 414D2)	Data HI (Reg 414D3)	Data HI (Reg 414D3)	Data HI (Reg 414D4)	Data HI (Reg 414D4)	Data HI (Reg 414D5)	Data HI (Reg 414D5)	Data HI (Reg 414D6)
32	33	34	35	36	37	38	39

Byte 16	Byte 17
Data HI (Reg 414D6)	Error Check
30	--

Note that the starting address of a register is the parameter number x 10 -1 in HEX.

Message sent to Modbus master from Modbus RTU option card

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Slave Address	Function	Start Addr HI	Start Addr LO	No. Registers HI	No. Registers LO	Error Check
01	10	14	D1	00	05	--

All values are in hexadecimal.

Exception Codes

When the VLT 6000 responds to the master via the Modbus serial network, it uses the function code field to indicate either a normal (error-free) response or an error (called an exception response). In an error-free response, the drive simply echoes the original function code. For an exception response, the drive will return a code that is equivalent to the original function code with its most-significant bit set to a logic 1. In addition, the drive places a unique code into the data field of the response message. This tells the master what kind of error occurred, or the reason for the exception. The tables below identify the codes and describe their meaning.

Modbus Code (decimal)	Meaning
00	The parameter number does not exist
01	There is no write access to the parameter
02	The data value exceeds the parameter limits
03	The used sub-index does not exist
04	The parameter is not of the array type
05	The data type does not match the parameter called
17	Data change in the parameter called is not possible in the present mode of the drive. Some parameters can only be changed when the motor has stopped
130	There is no bus access to the parameter called
131	Data Change is not possible because factory setup is selected
255	Message Timeout

VLT 6000 Errors

Modbus Code (decimal)	Meaning
64	Invalid Data Address
65	Invalid Message Length
66	Invalid Data Length
67	Invalid Function Code

Modbus RTU Errors

APPENDIX A

Supported Modbus RTU Function Codes

Appendix A describes the following functions supported by the Modbus RTU option card.

Read Coil Status (01_{HEX})
Force Single Coil (05_{HEX})
Force Multiple Coils ($0F_{HEX}$)
Read Holding Registers (03_{HEX})
Preset Single Register (06_{HEX})
Preset Multiple Registers (10_{HEX})

Read Coil Status (01_{HEX})

Description
Reads the ON/OFF status of discrete outputs (0X references, coils) in the slave. Broadcast is never supported for reads.

Query
The query message specifies the starting coil and quantity of coils to be read. Coils are addressed starting at zero. Coils 1-16 are addressed as 0-15.

Example of a request to read coils 1-16 from slave device 01.

Field Name	Example (HEX)
Slave Address	01
Function	01
Starting Address HI	00
Starting Address LO	00
No. of Points HI	00
No. of Points LO	10
Error Check (CRC)	–

Response
The coil status in the response message is packed as one coil per bit of the data field. Status is indicated as: 1 = ON; 0 = OFF. The LSB of the first data byte contains the coil addressed in the query. The other coils follow toward the high order end of this byte, and from ‘low order to high order’ in subsequent bytes.

If the returned coil quantity is not a multiple of eight, the remaining bits in the final data byte will be padded with zeros (toward the high order end of the byte). The Byte Count field specifies the quantity of complete bytes of data.

Field Name	Example (HEX)
Slave Address	01
Function	01
Byte Count	02
Data (Coils 8-1)	55
Data (Coils 16-9)	AA
Error Check (CRC)	–

Force Single Coil (05_{HEX})

Description

Forces a single coil (0X reference) to either ON or OFF. When broadcast, the function forces the same coil references in all attached slaves.

Query

The query message specifies the coil reference to be forced. Coils are addressed starting at zero. Coil 1 is addressed as 0. Force Data = 00 00_{HEX} (OFF) or FF 00_{HEX} (ON).

Example of a request to set coil 1 (addressed as 0) from slave device 01.

Field Name	Example (HEX)
Slave Address	01
Function	05
Coil Address HI	00
Coil Address LO	00
Force Data HI	FF
Force Data LO	00
Error Check (CRC)	–

Response

The normal response is an echo of the query, returned after the coil state has been forced.

Field Name	Example (HEX)
Slave Address	01
Function	05
Force Data HI	FF
Force Data LO	00
Quantity of Coils HI	00
Quantity of Coils LO	0A
Error Check (CRC)	–

Force Multiple Coils (0F$_{HEX}$)

Description
Forces each coil (0X reference) in a sequence of coils to either ON or OFF. When broadcast, the function forces the same coil references in all attached slaves.

Query
The query message specifies the coil references to be forced. Coils are addressed starting at zero. Coil 1 is addressed as 0.

Example of a request to set 10 coils starting at coil 1 (addressed as 0) from slave device 01.

Field Name	Example (HEX)
Slave Address	01
Function	0F
Coil Address HI	00
Coil Address LO	00
Quantity of Coils HI	00
Quantity of Coils LO	0A
Byte Count	02
Force Data HI (Coils 8-1)	FF
Force Data LO (Coils 10-9)	03
Error Check (CRC)	–

Response
The normal response returns the slave address, function code, starting address, and quantity of coils forced.

Field Name	Example (HEX)
Slave Address	01
Function	0F
Coil Address HI	00
Coil Address LO	00
Quantity of Coils HI	00
Quantity of Coils LO	0A
Error Check (CRC)	–

Read Holding Registers (03_{HEX})

Description
Reads the binary contents of holding registers (4x references) in the slave. Broadcast is never supported for reads.

Query
The query message specifies the starting register and quantity of registers to be read. Registers are addressed starting at zero. Registers 1-4 are addressed as 0-3.

Example of a request to read registers 40001-03 from slave device 01.

Field Name	Example (HEX)
Slave Address	01
Function	03
Starting Address HI	00
Starting Address LO	00
No. of Points HI	00
No. of Points LO	03
Error Check (CRC)	–

Response
The register data in the response message are packed as two bytes per register, with the binary contents right justified within each byte. For each register, the first byte contains the high order bits and the second contains the low order bits.

Field Name	Example (HEX)
Slave Address	01
Function	03
Byte Count	06
Data HI (Register 40001)	55
Data LO (Register 40001)	AA
Data HI (Register 40002)	55
Data LO (Register 40002)	AA
Data HI (Register 40003)	55
Data LO (Register 40003)	AA
Error Check (CRC)	–

Preset Single Register (06_{HEX})

Description
Presets a value into a single holding register (4x reference). When broadcast, the function presets the same register reference in all attached slaves.

Query
The query message specifies the register reference to be preset. Registers are addressed starting at zero. Register 1 is addressed as 0.

Example of a request to preset register 40002 to 00 03_{HEX} in slave device 01.

Field Name	Example (HEX)
Slave Address	01
Function	06
Register Address HI	00
Register Address LO	01
Preset Data HI	00
Preset Data LO	03
Error Check (CRC)	--

Response
The normal response is an echo of the query, returned after the register contents have been passed.

Field Name	Example (HEX)
Slave Address	01
Function	06
Register Address HI	00
Register Address LO	01
Preset Data HI	00
Preset Data LO	03
Error Check (CRC)	--

Preset Multiple Registers (10_{HEX})

Description
Presets values into a sequence of holding registers (4x references). When broadcast, the function presets the same register references in all attached slaves.

Query
The query message specifies the register references to be preset. Registers are addressed starting at zero. Register 1 is addressed as 0.

Example of a request to preset two registers starting at 40002 to 00 AA_{HEX} and 01 02_{HEX}, in slave device 1.

Field Name	Example (HEX)
Slave Address	01
Function	10
Starting Address HI	00
Starting Address LO	01
No. of Registers HI	00
No. of Registers LO	02
Byte Count	04
Write Data HI (Register 40001)	00
Write Data LO (Register 40001)	0A
Write Data HI (Register 40002)	01
Write Data LO (Register 40002)	02
Error Check (CRC)	–

Response
The normal response returns the slave address, function code, starting address, and quantity of registers preset.

Field Name	Example (HEX)
Slave Address	01
Function	10
Starting Address HI	00
Starting Address LO	01
No. of Registers HI	00
No. of Registers LO	02
Error Check (CRC)	–

Danfoss Graham
Division of Danfoss Inc.

8800 West Bradley Road
P.O. Box 245041
Milwaukee, Wisconsin 53224-9541
Phone: (414) 355-8800 Fax: (414) 355-6117
Toll free: (800) 621-8806
E-mail: graham@grahamdrives.com
http://www.namc.danfoss.com

XREF__BC
NOT LOADED ON RIP

www.danfoss.com/drives

For: Industrial Sales, Application Engineering

Danfoss Drives

Division of Danfoss Inc.
4401 N. Bell School Road
Loves Park, IL. 61111
Toll free: 800/432-6367
Telephone: 815/639-8600
Fax: 815/639-8000
www.namc.danfoss.com

For: HVAC Sales, Water Sales, Engineered Panels

Danfoss Graham and Danfoss US Water

Division of Danfoss Inc.
8800 W. Bradley Road
Milwaukee, WI. 53224
Toll free: 800/621-8806
Telephone: 414/355-8800
Fax: 414/355-6117
www.namc.danfoss.com

175R0504 MG10P222 *MG10P222* Rev. 2003-09-01