Modbus™ RTU

VLT®6000 Adjustable Frequency Drive

Instruction Manual

175R5467
Revision C 06/04

⚠WARNING

Equipment Hazard!

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. Failure to follow the NEC or local regulations could result in death or serious injury.

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

Unintended Start!

In Auto Mode, a start signal via digital inputs may cause drive to start at any time. The drive, motor, and any driven equipment must be in operational readiness. Failure to be in operational readiness in Auto Mode could result in death, serious injury, or equipment or property damage.

CAUTION

Electrostatic Discharge!

Electronic circuit card components 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.

⚠WARNING

Hazardous Voltage!

Touching electrical parts may be fatal, even after equipment has been disconnected from the AC input line. To be sure that capacitors have fully discharged, wait 14 minutes for 208 V and 480 V units and 30 minutes for 600 V units over 25 hp after power has been removed before touching any internal component. Failure to wait for capacitors to fully discharge before touching internal components could result in death or serious injury.

Table of Contents

Overview

Introduction 4
About This Manual 4
Assumptions 4
What You Should Already Know 4
References 4
Modbus RTU Overview 5
Overview: VLT 6000 with Modbus RTU 5

Network Connection

Network Connection 6
Drive DIP Switch Settings for Network Connection 6
Drive Parameter Settings for Modbus RTU Communication 6

Network Configuration

Remote Terminal Unit 7
Modbus RTU Message Structure 7
Memory Mapping: Coils 9
Memory Mapping: Holding Registers 11
Memory Mapping: Numbering 12
Conversion Index and Data Type 12

Communication Examples

Standard Modbus Functions 13
Example 1 Start the Drive and Give It a Reference via Modbus RTU ... 14
Example 2 Give the Drive a Reference via Modbus RTU 15
Example 3 Read Basic Drive Status Information 16
Example 4 Give the Drive a Ramp to Stop Command 18
Example 5 Give the Drive a Coast to Stop Command 19
Example 6 Set How the Drive Stores Parameter Values 20
Example 7 Write a Parameter Value from a List of Choices 21
Example 8 Write a 16 Bit Parameter 22
Example 9 Write a 32 Bit Parameter 23
Example 10 Read a 32 Bit Parameter 24
Example 11 Read Indexed Parameter and Read 8 Bit Parameter 25
Example 12 Write a Text String 27
Example 13 Read a Text String 28

Exception Codes

Exception Codes 29

Introduction

This manual provides comprehensive instructions on 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 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 protocol 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 Maintenance Manual, Danfoss Graham document number 175R5469.
(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, or a timeout will occur.

Overview: VLT 6000 with Modbus RTU

The VLT 6000 Adjustable Frequency Drive communicates in Modbus RTU format over an EIA-485 (formerly RS-485) network. Modbus RTU 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.

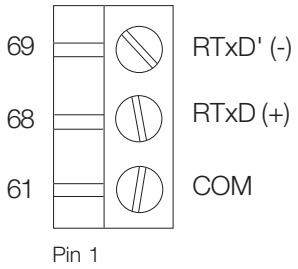
It is also possible to access the drive parameters, read their values, and, where possible, write values to them. This permits a range of control possibilities, including controlling the drive's setpoint when its internal PID controller is used.

Network Connection

Connect the Modbus RTU control wiring to drive terminals (+)68 and (-)69 on control card as illustrated.

Use 18 - 24 AWG (1.5 mm) shielded, twisted pair control wiring. Tighten to 4.5 in-lbs (0.5 Nm).

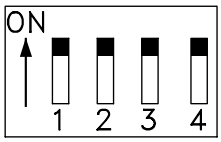
Shielded wiring must be used. Ground shield to drive terminal 61, drive casing, or as appropriate for the network application.



EIA-485 Connector

Drive DIP Switch Settings for Network Connection

DIP switches 2 and 3, located on the main control board of the drive directly above the EIA-485 connector, can be used to terminate drives on the Modbus network (see figure). In addition to serial communication, switch 4 is used to isolate a low voltage, external DC power supply. The factory default switch positions are ON (see table).



Drive DIP Switches

SWITCH	SETTING
Switch 1	No function
Switches 2 & 3	Used for terminating an EIA-485 interface. When the drive is the first, last, or only device on a network, switches 2 and 3 must be ON. When the drive is in any other location on the network, switches 2 and 3 must be OFF.
Switch 4	Separates the common potential of the internal 24 VDC supply from the common potential of external 24 VDC supply. When switch 4 is in the OFF position, the external 24 VDC supply is isolated from the drive. See the drive operation manual for more information on switch 4.

Drive DIP Switche Settings

Drive Parameter Settings for Modbus RTU Communication

The Modbus RTU interface to the VLT 6000 Series drive requires drive parameter values selected as shown. See the *VLT 6000 Operating Instructions* for details on selecting and changing parameter values, if necessary.

- Parameter 500, *Protocol*: Modbus RTU
- Parameter 501, *Address*: 001 - 247
- Parameter 502, *Baud Rate*: 300 - 9600 baud. (Factory default is 9600 baud.)
- Parameter 570, *Modbus Parity/Frame*: select from the following, based on application.

  Even Parity/1 Stop

  Odd Parity/1 Stop

  No Parilty/1 Stop (factory default)
- Parameter 571, *Modbus Communication Timeout*: 10 - 2000 ms. (Factory default setting is 100 ms.)

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 broadcast messages. 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 selected 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 cause a timeout (no response from the slave) 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. 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 coil or 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, a bus timeout 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 40150 is addressed as register 0095_{HEX} (149 decimal).

Memory Mapping: Coils

This is primarily used to control drive operation and to read out basic status information.

Coil 65 determines whether parameter changes made via Modbus RTU are held in the temporary memory (RAM) of the drive or are, in addition, written to permanent memory (EEPROM).

Coil Number	Description	Signal Direction
1 – 16	Drive Control Word (see table)	Master to Drive
17 - 32	Drive Speed or Setpoint Reference 0_{HEX} = 0% Reference, the value set in parameter 204. 4000_{HEX} = 100% Reference, the value set in parameter 205. 8000_{HEX} = 200% Reference, the maximum value that can be transmitted via Modbus RTU. In most cases, if references from multiple sources are supplied to the drive, the drive will respond to the sum of the references	Master to Drive
33 – 48	Drive Status Word (see table)	Drive to Master
49 – 64	**Open Loop Mode**: Drive Output Frequency 0_{HEX} = 0% of the reference value set in parameter 204. 4000_{HEX} = 100% of the reference value set in parameter 205. **Closed Loop Mode**: Drive Feedback Signal 0_{HEX} = 0% of the reference value set in parameter 204. 4000_{HEX} = 100% of the Reference value set in parameter 205.	Drive to Master
65	Parameter Write Control (Master to Drive) 0 ➔ Parameter changes are written to the drive's RAM and are lost when power is removed from the drive. 1 ➔ Parameter changes are written to the drive RAM and to its EEPROM and are maintained when power is removed from the drive.	Master to Drive
66 - 65536	Reserved	

Memory Mapping: Coils (continued)

Control Word Coil Description

Coil	0	1
01	Preset Ref. LSB	
02	Preset Ref. MSB	
03	DC Brake	No DC Brake
04	Coast stop	No coast stop
05	Quick stop	No quick stop
06	Freeze freq.	No freeze freq.
07	Ramp stop	Start
08	No reset	Reset
09	No jog	Jog
10	no function	
11	Data not valid	Data valid
12	Relay 1 OFF	Relay 1 ON
13	Relay 2 OFF	Relay 2 ON
14	Setup LSB	
15	Setup MSB	
16	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

Status Word Coil Description

Coil	0	1
33	Control not ready	Control ready
34	Drive not ready	Drive ready
35	Coasting Stop	Safety closed
36	No alarm	Alarm
37	Not in use	
38	Not in use	
39	Not in use	
40	No warning	Warning
41	Not at reference	At reference
42	Hand mode	Auto mode
43	Out of frequency range	In frequency range
44	Stopped	Running
45	No function	No function
46	No voltage warning	Voltage warning
47	Not in current limit	Current limit
48	No thermal warning	Thermal warning

Memory Mapping: Holding Registers

Holding registers are used to read and write drive parameters. They are standard Modbus 4x registers, 40001 through 4000F (see table below). A number of read-out parameters are provided in the 500 group of parameters to allow detailed operational information read through serial communications. A parameter value is accessed by using the register number that is 10 times its parameter number. For example, parameter 201 is accessed through register 2010. See the drive instruction manual for valid parameter numbers and their meanings.

For parameters with numeric data:

Refer to the Data Type table (on the following page) for each parameter to determine if one register (for 8 and 16-bit values) or two registers (for 32-bit values) must be transferred.

Refer to the Conversion Index table (on the following page) for each parameter, as shown in the drive instruction manual, to properly scale numeric values that are sent via Modbus RTU.

For parameters that involve selecting from a list, refer to the individual parameter description in the drive instruction manual to determine the numeric value associated with each selection. This value is shown in brackets [ ] in each parameter selection.

Parameters that involve alphanumeric data can store up to 20 ASCII characters. Two characters can be transferred in each register.

Register Number	Description
00001– 00008	Reserved
00009	Used to specify the index number to be used when an indexed parameter is accessed.
00010 – 00990	000 Parameter Group (parameters 001 through 099, as available)
01000 – 01990	100 Parameter Group (parameters 100 through 199, as available)
02000 – 02990	200 Parameter Group (parameters 200 through 299, as available)
03000 – 03990	300 Parameter Group (parameters 300 through 399, as available)
04000 – 04990	400 Parameter Group (parameters 400 through 499, as available)
05000 – 05990	500 Parameter Group (parameters 500 through 599, as available)
06000 – 06990	600 Parameter Group (parameters 600 through 699, as available)
07000 – 07990	700 Parameter Group (parameters 700 through 799, as available)
08000 – 08990	800 Parameter Group (parameters 800 through 899, as available)
09000 – 09990	900 Parameter Group (parameters 900 through 999, as available)

Holding Register Parameter Groups

Memory Mapping: Numbering

A coil or register number is offset by one from the coil or register address. Coil and register numbers start at 1 while the coil and register addresses for them start at 0. Therefore, to find a coil or register address from the number, subtract 1.

Conversion Index and Data Type

When accessing a parameter value in the drive, note the Conversion Index and Data Type of the parameter. These are shown in the parameter table in the back of the drive's instruction manual.

The conversion index is used to properly scale the value that is transmitted over the serial communications. This is necessary because only integers are passed over the serial communications. To represent decimal numbers, a conversion factor is associated with each conversion index. To convert a value passed over the serial communications to the actual value, simply multiply it by the index.

> Example: Parameter 201, *Minimum Frequency*, has a conversion index of -1. The conversion factor is therefore 0.1. A value of 150 that is transmitted via serial communications, therefore, represents 150 x 0.1 = 15.0.

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

Conversion Index

The data type indicates the number of bits needed to transmit the number via serial communications. 8 or 16 bit data occupy one Modbus RTU register while 32 bit data requires two registers.

> Example: Parameter 515, *Data Read-out: Power kW*, has a data type of 7. It is a 32-bit unsigned number. In order to read the output power from the drive, it is necessary to read two Modbus RTU registers.

Data Type	Description
3	Integer 16
4	Integer 32
5	Unsigned 8
6	Unsigned 16
7	Unsigned 32
9	Text string

Data Type

Standard Modbus Functions

The drive's Modbus RTU interface supports the following standard Modbus RTU functions.

01_{HEX}	Read Coil Status (Example 3)	03_{HEX}	Read Holding Registers (Example 10 & 11)
05_{HEX}	Force Single Coil (Example 6)	06_{HEX}	Preset Single Register (Example 7, 8 & 11)
$0F_{HEX}$	Force Multiple Coils (Examples 1, 2, 4 & 5)	10_{HEX}	Preset Multiple Registers (Example 9 & 12)

Communication Examples

The examples on the following pages demonstrate how to use these functions to control and monitor drive operation. All communication examples assume that the drive's Modbus RTU address is 1.

Example 1 **Start the Drive and Give It a Reference via Modbus RTU**
Setting the Control Word and Serial Reference Coils
Modbus Function $0F_{HEX}$ – Force Multiple Coils

Example 2 **Give the Drive a Reference via Modbus RTU**
Setting the Control Word and Serial Reference Coils
Modbus Function $0F_{HEX}$ – Force Multiple Coils

Example 3 **Read Basic Drive Status Information**
Reading the Status Word and Output Frequency/Feedback Coils
Modbus Function 01_{HEX} – Read Coil Status

Example 4 **Give the Drive a Ramp to Stop Command**
Setting the Control Word Coils
Modbus Function $0F_{HEX}$ – Force Multiple Coils

Example 5 **Give the Drive a Coast to Stop Command**
Setting the Control Word Coils
Modbus Function $0F_{HEX}$ – Force Multiple Coils

Example 6 **Set How the Drive Stores Parameter Values**
Setting Coil 65
Modbus Function 05_{HEX} – Force Single Coil

Example 7 **Write a Parameter Value from a List of Choices**
Writing a single register
Modbus Function 06_{HEX} – Preset Single Register

Example 8 **Write a 16 Bit Parameter**
Writing a single register
Modbus Function 06_{HEX} – Preset Single Register

Example 9 **Write a 32 Bit Parameter**
Writing two registers
Modbus Function 10_{HEX} – Preset Multiple Registers

Example 10 **Read a 32 Bit Parameter**
Reading two registers
Modbus Function 03_{HEX} – Read Holding Registers

Example 11 **Read an Indexed Parameter and Read an 8 Bit Parameter**
Setting the value of register 1 and reading a single register
Modbus Function 06_{HEX} – Preset Single Register
Modbus Function 03_{HEX} – Read Holding Registers

Example 12 **Write a Text String**
Writing multiple registers
Modbus Function 10_{HEX} – Preset Multiple Registers

Example 13 **Read a Text String**
Reading multiple registers
Modbus Function 03_{HEX} – Read Holding Registers

Example 1 — Start the Drive and Give It a 40% Reference via Modbus RTU

While this example shows both starting the drive and giving it a reference via Modbus RTU, it is possible to simply give it a start command. To do this, force only coils 1 through 16.

Modbus RTU Function: $0F_{HEX}$ *(Force Multiple Coils)*

Overview:

- **Force coils 1_{10} through 16_{10} to $047C_{HEX}$.**
  This is a run command. Refer to the description of the Control Word. Other Control Words could also give a run command, depending on the other functions desired.
- **Force coils 17_{10} through 32_{10} to 1999_{HEX}.**
  Because a value of 4000_{HEX} represents a 100% reference, the value for a 40% reference is 1999_{HEX}.

Message from Modbus RTU Master to Drive

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Slave Address	Function	Coil Addr., High	Coil Addr. Low	No. of Coils, High	No. of Coils, Low	Byte Count
01	0F	00	00	00	20	04

Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12
Force Coils Addr. 0-7	Force Coils Addr. 8-15	Force Coils Addr. 16-23	Force Coils Addr, 24-31	CRC Error Check	
7C	04	99	19	37	43

Modbus RTU message string:

[01] [0F] [00] [00] [00] [20] [04] [7C] [04] [99] [19] [37] [43]

Run Command: 0000 0100 0111 1100 = $047C_{HEX}$
(See the Control Word Coil Description)

[01] [0F] [00] [00] [00] [20] [04] [7C] [04] [99] [19] [37] [43]

Commanded Reference: 40% of 4000_{HEX} = 1999_{HEX}

Message returned from Drive to Modbus RTU Master

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Slave Address	Function	Coil Addr., High	Coil Addr. Low	No. of Coils, High	No. of Coils, Low
01	0F	00	00	00	20

Byte 6	Byte 7
CRC Error Check	
54	13

Example 2 Give the Drive a 75% Reference via Modbus RTU

Modbus RTU Function: $0F_{HEX}$ *(Force Multiple Coils)*

Overview:

- **Force coils 17_{10} through 32_{10} to 3000_{HEX}.**
  Because a value of 4000_{HEX} represents a 100% reference, the value for a 75% reference is 3000_{HEX}.
  The value of coil 11 must have previously been set to 1 for the drive to respond to the changed speed reference.

Message from Modbus RTU Master to Drive

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Slave Address	Function	Coil Addr., High	Coil Addr. Low	# of Coils, High	# of Coils, Low	Byte Count
01	0F	00	10	00	10	02

Byte 7	Byte 8	Byte 09	Byte 10
Force Coils Addr. 16-23	Force Coils Addr, 24-31	CRC Error Check	
00	30	E0	A4

Modbus RTU message string:
[01] [0F] [00] [10] [00] [10] [02] [00] [30] [E0] [A4]

Commanded Reference: 75% of 4000_{HEX} = 3000_{HEX}

Message returned from Drive to Modbus RTU Master

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Slave Address	Function	Coil Addr., High	Coil Addr. Low	No. of Coils, High	No. of Coils, Low
01	0F	00	10	00	10

Byte 6	Byte 7
CRC Error Check	
55	C2

Example 3

Read Basic Drive Status Information

(Read the status and the speed of the drive — or the value of its feedback signal.)

While detailed status information can be read by accessing the Data Read-out parameters in the 500 group of parameters and some other locations, it is also possible to read general drive status by reading coils 33 through 64.

- Coils 33 through 48 contain the drive Status Word.
- Coils 49 through 64 contain the drive output speed during open loop operation and the drive feedback value during closed loop operation.

Modbus RTU Function: 01_{HEX} *(Read Coil Status)*

Overview:

- Read the value of coils 33_{10} through 64_{10}.

Message from Modbus RTU Master to Drive

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Slave Address	Function	Coil Addr., High	Coil Addr. Low	No. of Coils, High	No. of Coils, Low
01	01	00	20	00	20

Byte 6	Byte 7
CRC Error Check	
3C	18

Modbus RTU message string:
[01] [01] [00] [20] [00] [20] [3C] [18]

Starting Coil Address = Coil Number – 1 = 33 – 1 = 32_{10} = 20_{HEX}

[01] [01] [00] [20] [00] [20] [3C] [18]

Number of Coils to Read = 32_{10} = 20_{HEX}

Message returned from Drive to Modbus RTU Master

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Slave Address	Function	# of Bytes	Coil Data 33 - 40	Coil Data 41 - 48	Coil Data 49 - 56	Coil Data 57 - 64
01	01	04	07	0F	FF	1F

Byte 7	Byte 8
CRC Error Check	
CA	9E

CONTINUED ON NEXT PAGE

Example 3 (continued)

Modbus RTU message string:
[01] [01] [04] [07] [0F] [FF] [1F] [CA] [9E]

Status Word = 0F07 = 0000 1111 0000 0111

At Reference	Control Ready
Remote Mode	Drive Ready
In Frequency Range	Safety Closed
Running	

[01] [01] [04] [07] [0F] [FF] [1F] [CA] [9E]

Output Frequency (for Open Loop operation)
4000_{HEX} = 100%, so $1FFF_{HEX}$ = 50%
If Parameter 204, MIN. REFERENCE = 10 Hz
and Parameter 205, MAX. REFERENCE = 60 Hz,
then the output frequency is 50% of the way between 10 Hz and 60 Hz,
or 35 Hz.

Example 4 Give the Drive a Ramp to Stop Command

Modbus RTU Function: $0F_{HEX}$ *(Force Multiple Coils)*

Overview:

Note: *It isn't necessary to change the reference value to stop the drive. It also isn't necessary to give a reference vie Modbus RTU in order to start the drive.*

- **Force coils 1_{10} through 16_{10} to $043C_{HEX}$.**
  This is a ramp stop command. Refer to the description of the Control Word. Other Control Words could also give a ramp stop command, depending on the other functions desired.

Message from Modbus RTU Master to Drive

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Slave Address	Function	Coil Addr., High	Coil Addr. Low	No. of Coils, High	No. of Coils, Low	Byte Count
01	0F	00	00	00	10	02

Byte 7	Byte 8	Byte 9	Byte 10
Force Coils Addr. 0-7	Force Coils Addr. 8-15	CRC Error Check	
3C	04	F2	E3

Modbus RTU message string:

[01] [0F] [00] [00] [00] [10] [02] [3C] [04] [F2] [E3]

Ramp Stop Command: 0000 0100 0011 1100 = $043C_{HEX}$
(See the Control Word Coil Description)

Message returned from Drive to Modbus RTU Master

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Slave Address	Function	Coil Addr., High	Coil Addr. Low	No. of Coils, High	No. of Coils, Low
01	0F	00	00	00	10

Byte 6	Byte 7
CRC Error Check	
54	07

Example 5 Give the Drive a Coast to Stop Command

A Modbus RTU coasting stop command, when drive parameter 503 is set to allow coasting stop commands via the serial bus, acts much like a safety interlock. The drive cannot run in any mode until the coasting stop command is removed. The lower right corner of the drive's normal display will show UN. READY *when a coasting stop command is active.*

Modbus RTU Function: $0F_{HEX}$ *(Force Multiple Coils)*

Overview:

Note: It isn't necessary to change the reference value to stop the drive. It also isn't necessary to give a reference via Modbus RTU in order to start the drive.

- **Force coils 1_{10} through 16_{10} to 0474_{HEX}.**
  This is a coast stop command. Refer to the description of the Control Word. Other Control Words could also give a coast stop command, depending on the other functions desired.

Message sent from the Modbus RTU Master to the Drive

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Slave Address	Function	Coil Addr., High	Coil Addr. Low	No. of Coils, High	No. of Coils, Low	Byte Count
01	0F	00	00	00	10	02

Byte 7	Byte 8	Byte 9	Byte 10
Force Coils Addr. 0-7	Force Coils Addr. 8-15	CRC Error Check	
74	04	C4	E3

Modbus RTU message string:

[01] [0F] [00] [00] [00] [10] [02] [74] [04] [C4] [E3]

Coast Stop Command: 0000 0100 0111 0100 = 0474_{HEX}

(See the Control Word Coil Description)

Message returned from Drive to Modbus RTU Master

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Slave Address	Function	Coil Addr., High	Coil Addr. Low	No. of Coils, High	No. of Coils, Low
01	0F	00	00	00	10

Byte 6	Byte 7
CRC Error Check	
54	07

Example 6

Set How the Drive Stores Parameter Values

(Set the drive to store parameter changes in EEPROM)

By default, parameter changes made via Modbus RTU are only written to Random Access Memory (RAM). When power to the drive is cycled, these changes are lost. Because it is possible to write to parameters a large number of times using Modbus RTU, writing values by default to RAM avoids subjecting the Electronically Erasable Programmable Read Only Memory (EEPROM) to an excessive number of write operations. To write a parameter change to the drive's EEPROM, first write the value "1" to coil number 65. Then make the parameter change. Afterwards, to keep future parameter changes from being permanent, write the value "0" to coil number65.

Modbus RTU Function: 05_{HEX} *(Force Single Coil)*

Overview:

- **Force coil number 65_{10} to 1.**

Message from Modbus RTU Master to Drive

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Slave Address	Function	Coil Addr., High	Coil Addr. Low	No. of Coils, High	No. of Coils, Low
01	05	00	40	FF	00

Byte 6	Byte 7
CRC Error Check	
8D	EE

Modbus RTU message string:

[01] [05] [00] [40] [FF] [00] [8D] [EE]

Coil Address: Coil Number – 1 = 65 – 1 = 64_{10} = 0040_{HEX}.

Message returned from Drive to Modbus RTU Master

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Slave Address	Function	Coil Addr., High	Coil Addr. Low	No. of Coils, High	No. of Coils, Low
01	05	00	40	FF	00

Byte 6	Byte 7
CRC Error Check	
8D	EE

Example 7

Write a Parameter Value from a List of Choices
(Set Parameter 301, DIGITAL INPUT 17, to "RUN PERMISSIVE")

From the tables in the back of the drive's instruction manual:

Data Type = 5 ➔ Unsigned 8 bit number

Conversion Index = 0 ➔ Conversion Factor = 1 (no conversion needed)

From the description of Parameter 301 in the drive's instruction manual:
"RUN PERMISSIVE" is choice number [8]

Modbus RTU Function: 06_{HEX} *(Preset Single Register)*
Overview:

- **Preset Register 3010_{10} to 08_{HEX}.**
  Because this is an 8 bit number, only a single register needs to be accessed.

Message from Modbus RTU Master to Drive

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Slave Address	Function	Register Addr., High	Register Addr. Low	Preset Data High	Preset Data Low
01	06	0B	C1	00	08

Byte 6	Byte 7
CRC Error Check	
DB	D4

Modbus RTU message string:
[01] [06] [0B] [C1] [00] [08] [DB] [D4]

Register Address = (Parameter Number x 10) – 1 = 3009_{10} = $0BC1_{HEX}$

[01] [06] [0B] [C1] [00] [08] [DB] [D4]

Set the parameter value to 8

Message returned from the Drive to the Modbus RTU Master:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Slave Address	Function	Register Addr., High	Register Addr. Low	Preset Data High	Preset Data Low
01	06	0B	C1	00	08

Byte 6	Byte 7
CRC Error Check	
DB	D4

Example 8

Write a 16 Bit Parameter

(Write the value 60 Hz to Parameter 104.)

From the tables in the back of the drive instruction manual:

Data Type = 6 ➔ Unsigned 16 bit number

Conversion Index = 0 ➔ Conversion Factor = 1 (no conversion needed)

Modbus RTU Function: 06_{HEX} *(Preset Single Register)*

Overview:

- **Preset Register 1040_{10} to $3C_{HEX}$.**
  Because this is a 16 bit number, only a single register needs to be accessed. There is no conversion needed to send the value, except to convert 60_{10} to $3C_{HEX}$.

Message from Modbus RTU Master to Drive

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Slave Address	Function	Register Addr., High	Register Addr. Low	Preset Data High	Preset Data Low
01	06	04	0F	00	3C

Byte 6	Byte 7
CRC Error Check	
B8	E8

Modbus RTU message string:

[01] [06] [04] [0F] [00] [3C] [B8] [E8]

Register Address = (Parameter Number x 10) – 1 = 1039_{10} = $040F_{HEX}$

[01] [06] [04] [0F] [00] [3C] [B8] [E8]

60_{10} = $003C_{HEX}$

Message returned from Drive to Modbus RTU Master

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Slave Address	Function	Register Addr., High	Register Addr. Low	Preset Data High	Preset Data Low
01	06	04	0F	00	3C

Byte 6	Byte 7
CRC Error Check	
B8	E8

Example 9

Write a 32 Bit Parameter

(Write the value 10.2 A to Parameter 105)

From the tables in the back of the drive's instruction manual:

Data Type = 7 ➔ Unsigned 32 bit number

Conversion Index = -2 ➔ Conversion Factor = 0.01

Modbus RTU Function: 10_{HEX} *(Preset Multiple Registers)*

Overview:

- **Preset two Registers, starting at 1050_{10}, $03FC_{HEX}$.**
  Because the Conversion Factor is 0.01, the number 1020 must be sent via Modbus RTU to represent 10.20 A. 1020_{10} = 0000 $03FC_{HEX}$.

Message from Modbus RTU Master to Drive

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Slave Address	Function	Start Register Addr., High	Start Register Addr. Low	No. of Registers High	No. of Registers Low	No. of Bytes
01	10	04	19	00	02	04

Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12
Register 1 Data, High	Register 1 Data, Low	Register 2 Data, High	Register 2 Data, Low	CRC Error Check	
00	00	03	FC	00	B8

Modbus RTU message string:

[01] [10] [04] [19] [00] [02] [04] [00] [00] [03] [FC] [00] [B8]

Register Address = (Parameter Number x 10) – 1 = 1049_{10} = 0419_{HEX}

[01] [10] [04] [19] [00] [02] [04] [00] [00] [03] [FC] [00] [B8]

10.2 x 100 = 1020_{10}= 0000 $03FC_{HEX}$

Message returned from Drive to Modbus RTU Master

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Slave Address	Function	Start Register Addr., High	Start Register Addr., Low	No. of Registers High	No. of Registers Low
01	10	04	19	00	02

Byte 6	Byte 7
CRC Error Check	
91	3F

Example 10

Read a 32 Bit Parameter

(Read the value of Parameter 514, Motor Current)

From the tables in the back of the drive instruction manual:

Data Type = 7 ➔ Unsigned 32 bit number

Conversion Index = -2 ➔ Conversion Factor = 0.01 (multiply the value from Modbus RTU by 0.01)

Modbus RTU Function: 03_{HEX} *(Read Holding Registers)*

Overview:

- **Read Two Registers, Starting at 5140_{HEX}.**
  Because this is a 32 bit number, two registers need to be read.
  It will be necessary to multiply the value returned by 0.01.

Message from Modbus RTU Master to Drive

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Slave Address	Function	Register Addr., High	Register Addr. Low	No. of Registers High	No. of Registers Low
01	03	14	13	00	02

Byte 6	Byte 7
CRC Error Check	
30	3E

Modbus RTU message string:

[01] [03] [14] [13] [00] [02] [30] [3E]

Register Address = (Parameter Number x 10) – 1 = 5139_{10} = 1413_{HEX}

Message returned from Drive to Modbus RTU Master

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Slave Address	Function	Byte Count	Data High (Register 5140)	Data Low (Register 5140)	Data High (Register 5141)	Data Low (Register 5141)
01	03	04	00	00	02	59

Byte 7	Byte 8
CRC Error Check	
3B	69

Modbus RTU message string:

[01] [03] [04] [00] [00] [02] [59] [3B] [69]

$00000259_{HEX} = 601_{10}$

601 x 0.01 = 6.01 A

Example 11

Read an Indexed Parameter and Read an 8 Bit Parameter
(Read the value of Parameter 615, Index 4, Fault Log: Error Code)

From the tables in the back of the drive instruction manual:

Data Type = 5 ➔ Unsigned 8 bit number

Conversion Index = 0 ➔ Conversion Factor = 0.01 (no conversion needed)

This is a 2-step process.

- First select the index value to use.
- Next, read the value of the parameter.

Step 1

Modbus RTU Function: 06_{HEX} *(Preset Single Register)*

Overview:

Set register 9_{10} to 4, the index value.

Message from Modbus RTU Master to Drive

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Slave Address	Function	Register Addr., High	Register Addr. Low	Preset Data High	Preset Data Low
01	06	00	08	00	04

Byte 6	Byte 7
CRC Error Check	
09	CB

Modbus RTU message string:

[01] [06] [00] [08] [00] [04] [09] [CB]

Register Address = Register Number – 1 = 9 – 1 = 8

[01] [06] [00] [08] [00] [04] [09] [CB]

Index Number = 4

Step 2

Modbus RTU Function: 03_{HEX} *(Read Holding Registers)*

Overview:

- **Read Register 6150_{10}**
  Because the value is an 8 bit number, only one register needs to be read.
  No conversion factor will be needed.

Message from Modbus RTU Master to Drive

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Slave Address	Function	Register Addr., High	Register Addr. Low	No. of Registers High	No. of Registers Low
01	03	18	05	00	01

Byte 6	Byte 7
CRC Error Check	
92	AB

CONTINUED ON NEXT PAGE

Example 11 (continued)

Modbus RTU message string:
[01] [03] [18] [05] [00] [01] [92] [AB]

Register Address = (Parameter Number x 10) – 1 = 6149_{10} = 1805_{HEX}

Recall that in Modbus RTU Register Address = Register Number -1

Message returned from Drive to Modbus RTU Master

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 7	Byte 8
Slave Address	Function	Byte Count	Data High	Data Low	CRC Error Check	
01	03	02	00	3C	B8	55

Modbus RTU message string:
[01] [03] [02] [00] [3C] [B8] [55]

$003C_{HEX} = 60_{10}$
The Alarm Table in the drive Instruction Manual shows that this is a Safety Interlock alarm.

Example 12 Write a Text String

(Write the text string “Drive” to Parameter 534)

Some parameters are designed to hold text strings. The maximum length of a text string is 20 characters. Text is transferred in ASCII format. Each register can hold two ASCII characters.

From the tables in the back of the drive’s instruction manual:

Data Type = 9 ➔ Text string

Conversion Index = 0 ➔ Conversion Factor = 1 (no conversion needed)

Modbus RTU Function: 10_{HEX} *(Preset Multiple Registers)*

Overview:

- **Preset three Registers, starting as 5340_{10}, to the value 44 72 69 76 65 20_{HEX}.**

Message from Modbus RTU Master to Drive

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Slave Address	Function	Start Register Addr., High	Start Register Addr. Low	No. of Registers High	No. of Registers Low	No. of Bytes
01	10	14	DB	00	03	06

Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12
Register 1 Data, High	Register 1 Data, Low	Register 2 Data, High	Register 2 Data, Low	Register 2 Data, High	Register 2 Data, Low
44	72	69	76	65	20

Byte 13	Byte 14
CRC Error Check	
33	BD

Modbus RTU message string:

[01] [10] [14] [DB] [00] [03] [06] [44] [72] [69] [76] [65] [20] [33] [BD]

Register Address = (Parameter Number x 10) – 1 = 5339_{10} = $14DB_{HEX}$

[01] [10] [14] [DB] [00] [03] [06] [44] [72] [69] [76] [65] [20] [33] [BD]

D r i v e [space]

Message returned from Drive to Modbus RTU Master

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Slave Address	Function	Start Register Addr., High	Start Register Addr., Low	No. of Registers High	No. of Registers Low
01	10	14	DB	00	03

Byte 6	Byte 7
CRC Error Check	
F5	C3

Example 13 Read a Text String

(Read the value of Parameter 622, Power Section)

A text field in the drive can contain up to 20 characters. If too few characters are read, the string that is transferred will be truncated. Text is transferred in ASCII format. Each register can hold two ASCII characters.

From the tables in the back of the drive's instruction manual:

Data Type = 9 ➔ Text string

Conversion Index = 0 ➔ Conversion Factor = 1 (no conversion needed)

Modbus RTU Function: 03_{HEX} *(Read Holding Registers)*

Overview:

- **Read ten registers, starting at 6220_{10}.**

Message from Modbus RTU Master to Drive

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Slave Address	Function	Register Addr., High	Register Addr. Low	No. of Registers High	No. of Registers Low
01	03	18	4B	00	0A

Byte 6	Byte 7
CRC Error Check	
B3	7B

Modbus RTU message string:

[01] [03] [18] [4B] [00] [0A] [B3] [7B]

Register Address = (Parameter Number x 10) – 1 = 6219_{10} = $184B_{HEX}$

Message returned from Drive to Modbus RTU Master

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Slave Address	Function	Byte Count	Data High (Register 6220)	Data Low (Register 6220)	Data High (Register 6221)	Data Low (Register 6221)
01	03	14	53	54	41	4E

Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13	Byte 13
Data High (Register 6222)	Data Low (Register 6222)	Data High (Register 6223)	Data Low (Register 6223)	Data High (Register 6224)	Data Low (Register 6224)	Data High (Register 6225)	Data Low (Register 6225)
44	41	52	44	20	20	20	20

Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13	Byte 13
Data High (Register 6226)	Data Low (Register 6226)	Data High (Register 6227)	Data Low (Register 6227)	Data High (Register 6228)	Data Low (Register 6228)	Data High (Register 6229)	Data Low (Register 6229)
20	20	20	20	20	20	20	20

Byte 7	Byte 8
CRC Error Check	
0E	C5

Modbus RTU message string:

[01] [03] [14] [53] [54] [41] [4E] [44] [41] [52] [44] [20] [20] [20] [20] [20] [20] [20] [20] [20] [20] [20] [0E] [C5]

S T A N D A R D [filled with spaces]

Exception Codes

When the drive 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 (hexadecimal)	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.
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.
40	Invalid data address
41	Invalid message length
42	Invalid data length
43	Invalid function code
82	There is no bus access to the parameter called

Exception Codes

Danfoss

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

http://www.namc.danfoss.com

23-6130-00