COPRA Products
Modbus Specification

Revision History

Revision Level	Originator	Description of Change	Date
0.0	Satya Akkina	Initial Release	8/5/2022
0.1	Satya Akkina	Added Fan Control registers	9/13/2022
0.2	Satya Akkina	Fixed R and R/W registers Added default demand registers	12/2/2022
0.3	Satya Akkina		
0.4	Satya Akkina	Added new changes Fixed some errors	
0.5	Satya Akkina	Added new registers Added detailed descriptions of all features and registers	8/16/2023
0.6	Satya Akkina	Corrected scale factor for digital input discrete demands Corrected address for input power Corrected descriptions for some of the registers	11/16/2023

While this document is under informal control in Engineering Revision numbering will be used. This numbering will be ALPHA in nature. (i.e. A.A for the first revision) Once the document has been fully reviewed and approved the numbering shall change to the standard. (i.e. 1.0 will be initial release) This will include resetting the revision history back to a single entry in the table.

Table of Contents

1 Introduction

1.1 Abstract

This document specifies the Modbus serial communications interface for the COPRA products and includes details of the register set and application notes.

1.2 Overview

Typically, user interaction with the COPRA products is over serial communications through a personal computer or hardware panel. The standard user interface is implemented using the Modbus message protocol (see Ref. [1]) via a RS585 half-duplex asynchronous serial communications link. The standard Modbus user interface shall support the following features:

- Metering of motor/fan variables and status
- Monitoring of user inputs
- Configuration of the user input settings

1.3 Scope

This is a technical document aimed primarily at systems developers and diagnostic level personnel. Additionally, it may be provided under third party agreement for restricted use by external systems developers, and qualified service and field application engineers. The information contained in this document is likely to be used as the source material to prepare user documentation for various user interfaces and control programs for customers, distributors and sales staff, as well as service, field application engineers and production staff.

1.4 Related Documents

This document relates to the following documents:

1.4.1 Modbus Standard Documentation

- MODBUS Application Protocol Specification V1.1b at http://www.Modbus-IDA.org
- MODBUS Messaging Implementation Guide V1.0a at http://www.Modbus-IDA.org
- MODBUS over Serial Line Specification and Implementation Guide V1.02 at http://www.modbus.org/

1.5 Abbreviations and Definitions

Table 1: Glossary of terms and acronyms

Abbreviation/Term	Definition
Bit	A binary digit. Represented by a “0” or a “1”
Byte	A measure of data size associated with a Modbus data element. A byte is equivalent to 8 bits.
Word	A measure of data size associated with a Modbus register number. A word is equivalent to 2 bytes or 16 bits.
LSB	Least Significant Byte/Bit – when referring to a Byte within a larger word, the LSB is the right-most byte of the word.
MSB	Most Significant Byte/Bit – when referring to a Byte within a larger word, the MSB is the left-most byte of the word.
ASCII	American Standard Code for Information Interchange - refers to the standard binary representation of text characters used by most computers.
CRC	Cyclic Redundancy Check – this value is calculated from a string of data values and is used by the recipient of the data string to check the string integrity.
RS485	Serial communications standard.
R	Read
RW	Read and Write

FC	Function Code

2 Physical and Data Specifications

2.1 RS485 Interface

Serial communications with the COPRA products are available via a RS485 link with the following physical

- Serial Communications Interface Type: RS485
- Electrical Characteristics: 3-wire, half duplex, single drop
- Baud Rate: 115200
- Data Format: 8 data bits, 1 stop bits, no parity
- Flow Control: None

Data is transferred as 8-bit characters, with no parity checking, each framed with 1 start and 1 stop bits, via a 3-wire line (A, B and GND), conforming to the RS485 standard. COPRA products are configured as clients and waiting for a server to request from (or) write information to COPRA products.

2.2 Modbus Communications Protocol

The Modbus protocol governs all communications over the RS485 link and employs packet-based messaging with the following characteristics.

- Communications Protocol: Modbus
- Message Format: Binary RTU
- Inter-packet Gap: 10.0 mS minimum
- Minimum Response Time: 50 mS
- Maximum Response Time: 500 mS
- Retry Transmission Attempts: 20
- Address Format: Zero base addressing

The COPRA products act as a client on Modbus and therefore never initiates unsolicited transmissions. A client only transmits a response to a valid message from the server containing its configured unit. The COPRA products use the binary RTU message, where each byte of data is transmitted in binary, rather than two ASCII characters. Each packet is terminated by a gap in transmission of at least 8**.**7 characters.

When responding to a Modbus request, the COPRA products will respond between the minimum and maximum response time (above) after receipt of the last byte of a packet from the server. The minimum response time gives the server time to switch its transceiver from transmit back to receive ready for the response, as well as ensuring that the slowest unit on the link recognizes the gap in transmission as a packet terminator. If the server does not receive a response after the maximum response time, then it should assume that the request message was invalid and attempt a retry. A communications failure is only reported after the number of consecutives retries.

2.3 General Modbus RTU Packet Format

All Modbus packets, when using the RTU message-framing scheme, have the following basic format, for both requests (sometimes called queries) from server to client and responses from client to server.

Table 2: General Modbus Binary RTU packet format

Byte 1	Client Address (1-247d, 0x01-0xF7)
Byte 2	Function Code (3, 4, 6, 16)
Byte 3	Variable length data section, B bytes in length (B = 0 to 255)
: :	
Byte 3+B	CRC – LSB
Byte 4+B	CRC – MSB

A variable length data section is framed by a header and footer. The header section contains the client address and a function code. The footer contains a 16-bit cyclic redundancy check in a pair of bytes.

The client address (byte 1) is the first byte in the message to be transmitted and received, followed by the function code and the variable length data section. The last pair of bytes is always the CRC; the low byte is always transmitted and received before the high byte. For each byte of the message, the least significant bit is transmitted and received first and the most significant bit last.

Except for the CRC, all 16-bit quantities (such as starting register, number of registers and register data) are always transmitted and received most significant byte first and least significant byte last. This is independent of the endianness of the processor data storage model.

2.4 Client Address

The client address allows the server to address a specific client on a RS485 link and is obligatory when using a multi-drop configuration

- Default Client Address: 247d (0xF7)

2.4.1 Global Address

The global also called broadcast address below is reserved for issuing global requests from the server to the client on a RS485 bus.

- Global Address: 0d (0x00)

On receiving a global request, the client must validate and take action regarding the request as if it is specifically addressed to that client. However, the client shall not transmit a response to a global request.

2.5 Function Codes

The COPRA PRODUCTS supports the following function codes:

Table 3: Valid COPRA PRODUCTS Function Codes

Function Codes	Description/Register Type	RW
3d (0x03)	Read Holding Registers	R
4d (0x04)	Read Input Registers	R
6d (0x06)	Write Single Holding Register	R/W
16d (0x10)	Write Single or Multiple Holding Registers	R/W

Any other function code will result in an illegal function exception

2.6 Modbus Cyclic Redundancy Check (CRC)

For all Modbus messages the footer section contains a 16-bit cyclic redundancy check in a pair of bytes (low byte first).

2.6.1 Message Transmission

During message transmission the CRC is computed for all the bytes in the message, starting with the client address (byte 1) up to but excluding the CRC bytes themselves. The result is placed in the last two bytes in the message (least significant byte first followed by most significant byte, which is the last byte of the packet).

2.6.2 Message Reception

During message reception the CRC is again computed for all the bytes in the message, starting with the client address (byte 1) up to but excluding the CRC bytes themselves. The result is compared with the CRC

bytes received in the last two bytes of the message. If the CRC matches, then the message is valid and is decoded and actioned. However, if the CRC is invalid then the message is discarded, and no further action takes place.

2.6.3 Computing the CRC

The standard method of CRC computation is to use a regressive polynomial:

- CRC Polynomial value: 0xA001 (see below)

Table 4: Pseudo Code for CRC calculation

Step 1	The 16-bit CRC register is first pre-loaded with all bits set (0xFFFF)
Step 2	The 8-bit byte from the message is logically XOR-ed with the lower order byte of the 16-bit CRC register, putting the result back in the CRC register
Step 3	The CRC register is shifted one bit to the right (toward the LSB), zero-filling the MSB. The previous LS Bit is extracted and examined.
Step 4	If the extracted LS Bit was 0b, then do nothing, else if the LS Bit was 1b, logically XOR the CRC register with the CRC polynomial
Step 5	Repeat steps 3-4 until eight shifts have been performed, one for each of the bits in the message byte
Step 6	Repeat steps 2-5 for each successive byte in the message, excluding the CRC bytes themselves.
Step 7	When the CRC is placed into or compared with the message, its upper and lower bytes must be swapped. The low order byte is transmitted and received first, followed by the high order byte which is always last.

2.7 Modbus Registers

The concept of Modbus registers allows items of data to be individually or collectively addressable by their assigned register number and is independent of the storage type, size and access method. A *register* is the identity of a *data entity* (a monolithic data item) that cannot be sub-divided or partially accessed. Typically, an entity is stored in a physical *location* in the client unit and has a physical storage *size*. A data entity may typically be a simple data type (such as a bit, byte or integer), but it may also be a complex data type (such as an array or data structure). The collection of all registers in a client unit is called a *register map* and can be visualized as a ragged array, where each element may be a different size, but any element may be indexed using its unique Modbus register number. The register number is effectively an index to the physical storage of a data entity and its properties such as size, interpretation, meaning, type of storage, physical location and access method).

2.7.1 Modbus Register Sizes

Strictly speaking the size of a Modbus register is equal to a 16-bit word. However, the specification implies that a register may be any multiple of 16-bits, from 1 to 125 words. In practice no register may be larger since the maximum number of data bytes allowed in a single Modbus message is 250 bytes. The COPRA products implement Modbus registers of a single 16-bit word.

2.7.2 Transporting Different Sized Data Entities through Modbus Registers

The storage size of a location is unrestricted and may be smaller or larger than 16 bits. For transportation over Modbus, a data entity is transformed from its storage size, into an appropriately sized register, by extending it to the next multiple of 16-bits and vice versa. When a location is less than 16-bits in length, then it is extended to occupy a single 16-bit register upon transmission and truncated on reception to the appropriate storage size of the location. If the data is signed, then it is sign extended to 16-bits, otherwise it is zero padded to 16-bits. However, if a location is larger than 16-bits in length, then it is extended to the next multiple of 16-bits upon transmission and truncated on reception to the appropriate storage size of the location. If the data is signed, then it is sign extended as appropriate, otherwise it is zero padded as appropriate.

2.7.3 The 'Number of Registers' Field

There are two schools of thought as to the interpretation of this field in the header of a Modbus message:

- The number of 16-bit words to be transferred. (This is the interpretation in the COPRA products).
- The number of data entities to be transferred.

The two interpretations are synonymous when dealing with uniformly sized registers of 16-bits. However, if registers may on occasion contain entities that are larger than a single 16-bit word, confusion can occur.

2.7.4 The 'Number of Data Bytes' Field

At first sight this field appears to be a duplication of information contained in the 'Number of Registers' field. However, when the 'Number of Registers' field is interpreted as the number of data entities to transfer then it becomes apparent that the addition of this field clarifies the confusion (mentioned above). This field in both the request and response message always equals the total number of data bytes contained in the message, excluding the Modbus header and CRC bytes.

2.7.5 The 'Starting Register' Field

The 'Starting Register' field in both the request and response message identifies the first register location to be accessed. Please note that this field contains the register number decreased by one. Thus the 'Starting Register' field contains zero (0) for register one (1), etc.

2.8 Read Multiple Registers (Function Code 0x03)

The following request and response messages are used to read one or more sequential registers from the.COPRA PRODUCTS.

Table 5: Function 0x03 Query Message Format from Server

Byte	Contents
1	Client Address (0x01 – 0xF7)
2	Function Code (0x03)
3	Starting Modbus Register (S) - MSB
4	Starting Modbus Register (S) - LSB
5	Number of Modbus Registers (r) - MSB
6	Number of Modbus Registers (r) - LSB
7	CRC - LSB
8	CRC - MSB

Table 6: Function 0x03 Response Message Format from COPRA PRODUCTS

Byte	Contents
1	Client Address (0x01 – 0xF7)
2	Function Code (0x03)
3	Number of data bytes (N) following in message (0-100d)
4	Starting Modbus Register (S) - MSB
5	Starting Modbus Register (S) - LSB
6 to (5+N)	Repeat register data MSB then LSB for number of registers, r
6+N	CRC - LSB
7+N	CRC - MSB

2.9 Read Input Registers (Function Code 0x04)

The following request and response messages are used to read one or more sequential registers from the COPRA products.

Table 7: Function 0x04 Query Message Format from Server

Byte	Contents
1	Client Address (0x01 – 0xF7)
2	Function Code (0x04)
3	Starting Modbus Register Address (S) - MSB
4	Starting Modbus Register Address (S) - LSB
5	Number of Modbus Registers (r) - MSB
6	Number of Modbus Registers (r) - LSB
7	CRC - LSB
8	CRC - MSB

Table 8: Function 0x04 Response Message Format from COPRA PRODUCTS

Byte	Contents
1	Client Address (0x01 – 0xF7)
2	Function Code (0x04)
3	Number of data bytes (N) following in message (0-100d)
4 to (3+N)	Repeat register data MSB then LSB for number of registers, r
4+N	CRC - LSB
5+N	CRC - MSB

2.10 Write Single Holding Register (Function Code 0x06)

The following request and response messages are used to write one holding registers to the COPRA products:

Table 9: Function 0x06 Query Message Format from Server

Byte	Contents
1	Client Address (0x01 – 0xF7)
2	Function Code (0x06)
3	Modbus Holding Register Address (S) – MSB
4	Modbus Holding Register Address (S) – LSB
5	Register data – LSB
6	Register data – MSB
7	CRC - LSB
8	CRC - MSB

Table 10: Function 0x06 Response Message Format from COPRA PRODUCTS

Byte	Contents
1	Client Address (0x01 – 0xF7)
2	Function Code (0x06)
3	Modbus Holding Register Address (S) - MSB
4	Modbus Holding Register Address (S) - LSB
5	Register Value – LSB
6	Register Value – MSB
7	CRC - LSB
8	CRC - MSB

2.11 Write Multiple Registers (Function Code 0x10)

The following request and response messages are used to write one or more sequential registers to the COPRA PRODUCTS:

Table 11: Function 0x10 Query Message Format from Server

Byte	Contents
1	Client Address (0x01 – 0xF7)
2	Function Code (0x10)
3	Starting Modbus Register Address (S) – MSB
4	Starting Modbus Register Address (S) – LSB
5	Number of Modbus Registers (r) – MSB
6	Number of Modbus Registers (r) – LSB
7	Number of data bytes following (N)
8	Register data – LSB
9	Register data – MSB
10-(7+N)	Repeat register data LSB then MSB for number of registers, r
8+N	CRC - LSB
9+N	CRC - MSB

Table 12: Function 0x10 Response Message Format from COPRA PRODUCTS

Byte	Contents
1	Client Address (0x01 – 0xF7)
2	Function Code (0x10)
3	Starting Modbus Register Address (S) - MSB
4	Starting Modbus Register Address (S) - LSB
5	Number of Modbus Registers (r) - MSB
6	Number of Modbus Registers (r) - LSB
7	CRC - LSB
8	CRC - MSB

2.12 Modbus Exception Response

The COPRA PRODUCTS respond with an exception response message in place of the normal response if any of the following conditions is detected for one or more locations accessed.

Table 13: Modbus Exception Response Message Format

Byte	Contents
1	Client Address (0x01 – 0xF7)
2	Function Code + 128d (0x83 or 0x90)
3	Exception Code (0x01-0x05)
4	CRC – LSB
5	CRC – MSB

An exception response is only sent provided:

- The client address in the request directly matches the unit's configured address (i.e.: not addressed to another unit or a global request)
- and the request message is valid (i.e.: the CRC check did not fail).

2.13 Exception Code

The COPRA PRODUCTS use the following exception codes:

Table 14: User Classes

Code	Response Type	Condition for Exception

None	No response	• Request contained global address • Unrecognized client address • Failed CRC
0x01	Illegal Function	• Requested function code is not supported
0x02	Illegal Data Address	• An invalid or unknown register address • Invalid number of registers • Invalid number of data words for specified registers in request • Currently unable to perform request on a register (e.g. invalid user access rights, invalid mode of operation) • Request is invalid for a register (e.g. write to read-only location) • Location access failure, invalid type or physical address
0x03	Illegal Data Value	• Invalid value or data type
0x04	Client Device Failure	• Gross failure of client unit
0x05	Acknowledge	• An operation is deferred or in progress

3 Application Notes

3.1 Register Groups

Registers are grouped into User Interface registers and Drive Interface registers. Within these groups, registers are sub divided into modules. Each module represents a specific hardware/firmware feature and their associated Modbus registers.

3.2 Register Properties

The properties of a register are defined by its:

- Address
- Type and size of physical storage
- Type and interpretation of data
- Valid data types
- Read and/or write access permissions

All register accesses are validated to ensure they conform to the properties of the register. If the access violates any of the properties of the register, then the operation cannot be performed, and an exception response is returned.

3.3 User Classes and Roles

Two classes of user have been identified as follows:

Table 15 User Classes

Class	Function
Customer	Basic end-user. May monitor metered levels, operational status and inspect the configuration. May alter basic operational states of the unit, change its behavior, accuracy or configuration. Default user type.
Admin	Service or diagnostic engineer. Allows on-site diagnostics and installation configurations, enabling of test modes.

Thus, each class of user is granted a set of access rights

3.4 Saving setting to FLASH

Any modification made to “Holding Registers” are temporary and will be lost once power is turned off. To preserve the changes the user must save them to the memory. To save settings follow the below procedure

a) Verify that “Flash Status” register is not in “FLASH_ERROR” state. If the status is “FLASH_ERROR”, settings may not be saved to memory and are using default factory settings. If the status is “FLASH_OK” or “READ_COMPLETE” everything is working as expected.
b) Set “Flash commands” register to “FLASH_WRITE_RAM2FLASH_USER_SETTINGS_CMD”
c) Wait until the “Flash status” register shows “FLASH_SETTINGS_WRITE_COMPLETE”. If the status register shows “FLASH_ERROR”, something went wrong during the saving process. Try again and see if the error goes away. If not, something has gone wrong with the firmware.

Note: If you have changed both app and drive registers, this process must be repeated for both App and Drive. Refer to “App Flash” and “Drive Flash” registers.

3.5 Retrieving Setting From FLASH

Any modification made to “Holding Registers” are temporary and will be lost once power is turned off. To preserve the changes the user must save them to the memory. To retrieve settings saved to the memory, follow the below procedure

a) Verify that “Flash Status” register is not in “FLASH_ERROR” state. If the status is “FLASH_ERROR”, settings may not be saved to memory and are using default factory settings. If the status is “FLASH_OK” or “READ_COMPLETE” everything is working as expected.
b) Set “Flash commands” register to “FLASH_READ_FLASH2RAM_USER_SETTINGS_CMD” if “Flash Status” is not in “HLASH_ERROR” state.
c) Wait until the “Flash status” register shows “FLASH_READ_COMPLETE”. If the status register shows “FLASH_ERROR”, something went wrong during the retrieval process. Try again and see if the error goes away. If not, something has gone wrong with the firmware.

Note that if you have changed both app and drive registers, this process must be repeated for both App and Drive. Refer to “App Flash” and “Drive Flash” registers.

3.6 Recovery Factory Setting from Flash

Any modification made to “Holding Registers” are temporary and will be lost once power is turned off. To preserve the changes the user must save them to the memory. To retrieve factory settings saved to the memory, follow the below procedure

a) Verify that “Flash Status” register is not in “FLASH_ERROR” state
b) Set “Flash commands” register to “FLASH_READ_FLASH2RAM_DEFAULT_SETTINGS” if “Flash Status” is not in “HLASH_ERROR” state.
c) Wait until the “Flash status” register shows “FLASH_READ_COMPLETE”. If the status register shows “FLASH_ERROR”, something went wrong during the saving process. Try again and see if the error goes away. If not, something has gone wrong with the firmware.

Note that if you have changed both app and drive registers, this process must be repeated for both App and Drive. Refer to “App Flash” and “Drive Flash” registers

App/User Interface FLASH Register

MODBUS Address (decimal)	Register Name	Description	Units	Scale Factor	Min	Max	Default Settings (dec)	Data Type	Register Type
35624	App Flash User Discretes 01	Bit 0: is_copyUserFlash2RamSuccess, Bit 1: is_copyDefaultFlash2RamSuccess, Bit 2: is_copyRam2USerFlashSucess	Bit Field	1	0	7	NA	u16	Input

35625	Flash status	FLASH_OK (0), FLASH_ERROR (2), FLASH_BUSY (3), FLASH_TIMEOUT (4), FLASH_EMPTY (5), FLASH_NOT_EMPTY (6), FLASH_SETTINGS_INIT_COMPLETE (7), FLASH_SETTINGS_WRITE_COMPLETE (8), FLASH_DEFAULT_SETTINGS_INIT_COMPLETE (9), FLASH_READ_IN_PROCESS (9), FLASH_ERASE_IN_PROCESS (9), FLASH_ERASE_ADD_ERROR (10), FLASH_WRITE_IN_PROCESS (11), FLASH_READ_COMPLETE (12), FLASH_ERASE_COMPLETE (13), FLASH_WRITE_COMPLETE (14), FLASH_WRITE_CRC_COMPLETE (15), FLASH_CRC_VALID (16), FLASH_CRC_EMPTY (17), FLASH_CRC_ERROR (18), FLASH_READ_ERROR (19), ~~FLASH_OUT_OF_RANGE_ERROR (20),~~ FLASH_VERSION_ERROR (21), ~~FLASH_WRITE_DATA_LENGTH_ERROR (22), FLASH_BUF_FULL (23),~~ ~~FLASH_COPY_TO_BUF_COMPLETE (24),~~ FLASH_UNKNOWN_STATE = 0xFF	enum	1	0	255	NA	u16	Input
35689	Flash Commands	USER Commands: FLASH_WAITING_FOR_CMD = 0, FLASH_WRITE_RAM2FLASH_USER_SETTINGS_CMD(1), FLASH_READ_FLASH2RAM_USER_SETTINGS_CMD(2); FLASH_READ_FLASH2RAM_DEFAULT_SETTINGS(3) ADMIN Commands: FLASH_WRITE_RAM2FLASH_DEFAULT_SETTINGS_CMD(4), FLASH_COPY_USER_SETTINGS_TO_DEFAULTS_CMD(5), FLASH_CRC_UPDATE(6), FLASH_ERASE_CMD(6), FLASH_WRITE_CMD(7), FLASH_UNKNOWN_CMD = 0xFF	enum	1	0	255	0	u16	Holding

Drive FLASH Register

MODBUS Address (decimal)	Register Name	Description	Units	Scale Factor	Min	Max	Default Settings (dec)	Data Type	Register Type
4904	Drive Flash User Discretes 01	Bit 0: is_copyUserFlash2RamSuccess, Bit 1: is_copyDefaultFlash2RamSuccess, Bit 2: is_copyRam2USerFlashSucess	Bit Field	1	0	7	NA	u16	Input
4905	Flash status	FLASH_OK = 0, FLASH_ERROR, FLASH_BUSY, FLASH_TIMEOUT, FLASH_EMPTY, FLASH_NOT_EMPTY, FLASH_SETTINGS_INIT_COMPLETE, FLASH_SETTINGS_WRITE_COMPLETE, FLASH_DEFAULT_SETTINGS_INIT_COMPLETE, FLASH_READ_IN_PROCESS, FLASH_ERASE_IN_PROCESS, FLASH_ERASE_ADD_ERROR, FLASH_WRITE_IN_PROCESS, FLASH_READ_COMPLETE, FLASH_ERASE_COMPLETE, FLASH_WRITE_COMPLETE, FLASH_WRITE_CRC_COMPLETE, FLASH_CRC_VALID, FLASH_CRC_EMPTY, FLASH_CRC_ERROR, FLASH_READ_ERROR, FLASH_OUT_OF_RANGE_ERROR, FLASH_VERSION_ERROR, FLASH_WRITE_DATA_LENGTH_ERROR, FLASH_BUF_FULL, FLASH_COPY_TO_BUF_COMPLETE, FLASH_UNKNOWN_STATE = 0xFF	enum	1	0	255	NA	u16	Input
4969	Flash Commands	USER Commands: FLASH_WAITING_FOR_CMD = 0, FLASH_WRITE_RAM2FLASH_USER_SETTINGS_CMD(1), FLASH_READ_FLASH2RAM_USER_SETTINGS_CMD(2); FLASH_READ_FLASH2RAM_DEFAULT_SETTINGS(3) ADMIN Commands: FLASH_WRITE_RAM2FLASH_DEFAULT_SETTINGS_CMD(4), FLASH_COPY_USER_SETTINGS_TO_DEFAULTS_CMD(5), FLASH_CRC_UPDATE(6), FLASH_ERASE_CMD(6), FLASH_WRITE_CMD(7), FLASH_UNKNOWN_CMD = 0xFF	enum	1	0	255	0	u16	Holding

4 COPRA PRODUCTS Modbus Register Map

This section provides the COPRA PRODUCTS Modbus registers.

Table 16: Register Table Nomenclature

Column Name	Description
MODBUS Address (decimal)	Modbus address of the register in decimal
Register Name Description	Register name
Units	Units of the register value
Scale Factor	To get the actual value of the parameter multiply register value with he scale factor. When writing to the register, take the value you want to write and divide it by scale factor and convert the result into the appropriate data type before writing to the register.
Min	Minimum value allowed by the register
Max	Maximum value allowed by the register
Default Settings (dec)	Default value set by factory
Data Type	Data type of the register
Register Type	Identifies whether the register is input or holding register

Table 17: Data Type

Data Type	Description	Size (words)	Minimum value	Maximum value
s16	Signed 16-bit integer with no decimal point	1	-32768	32767
s16 1DP	Signed 16-bit integer with one decimal point	1	-3276.8	3276.7
s16 2DP	Signed 16-bit integer with two decimal points	1	-327.68	327.67
s16 3DP	Signed 16-bit integer with three decimal points	1	-32.768	32.767
u16	Unsigned 16-bit integer with no decimal point	1	0	65535
u16 1DP	Unsigned 16-bit integer with one decimal point	1	0.0	6553.5
u16 2DP	Unsigned 16-bit integer with two decimal points	1	0.00	655.35
u16 3DP	Unsigned 16-bit integer with three decimal points	1	0.000	65.535
ls16	Signed 32-bit integer with no decimal point	2	-2147483648	2147483647
ls16 1DP	Signed 32-bit integer with one decimal point	2	-214748364.8	214748364.7
ls16 2DP	Signed 32-bit integer with two decimal points	2	-21474836.48	21474836.47
ls16 3DP	Signed 32-bit integer with three decimal points	2	-2147483.648	2147483.647
SIGNED_FIXED_POINT	Signed fixed-point representation, (a, b), of a floating-point number, x, of the format x = a*2^(b) where “a” and “b” are SIGNED_0DP ranging from -15 to +15	2	-32768	2147483647
BIT_FIELD	A subtype of an unsigned 16-bit integer represented by X bits. Multiple bit fields can be contained in a single Modbus register.	No more than 1 bit	0	1
ASCII	ASCII characters represented in hexadecimal format. Each byte representing a character	1	ASCII (0)	ASCII (255)
ENUMERATED	An unsigned integer defined by an enumerated list. The list shall be defined in a corresponding table.	1	0	65536

The units of a value or parameter accessed through a register are indicated as follows:

Table 18: Units of Modbus Registers

Key	Unit
%	Percentage
V	Volts
mV	Millivolts
A	Amps
mA	Milliamps
W	Watts
kW	Kilowatts
kVA	Kilo volt amps
kVAR	Kilo volt amps reactive
^{o}C	Degree Celsius
S or Sec	Seconds
mS or mSec	Milli Seconds
M on Min	Minutes
H or Hr	Hours
Hz	Hertz or cycles/second
RPM	Revolutions Per Minute
pu	Per unit or unitless
enum	For a register of the enumerated data type, the description column will contain the possible valid values of that enumerated type.

4.1 Register Group 1: User Interface

4.1.1 Analog 0-10V Input

a) User can read measured input voltage at 0-10V terminal by reading input register "*Analog Volts*"
b) The demand reference from analog input can be read by reading input register "*Analog Volts Demand Percent*"
c) User can enable or disable analog input by setting *Bit0* of "*Analog Input (0 -10) User Flags 01*" holding register
d) User can invert analog input (when invert is active 0V corresponds to 100% demand) by setting *Bit1* of "*Analog Input (0 -10) User Flags 01*" holding register
e) Minimum and maximum analog voltage can be adjusted by using holding registers *"Min Volts"* and "*Max Volts"* registers respectively
    I. For example, if hardware can read up to 11.5V, but the application only requires a range of 2 -8V, these settings can be modified to match these. Min volts correspond to min demand and max volts correspond to max demand. This will control the slope of the demand.
f) Allowable min and max demand correspond to min and max analog voltage can be adjusted by *"Min Demand*" and *"Max Demand"* holding registers respectively.
g) User can enable analog input low alarm by setting "*Bit2*" of "*Analog Input (0 -10) User Flags 01*" holding register
    I. Low analog input is detected if the measured analog voltage is below the set threshold or if the voltage is below threshold on power up. This threshold can be adjusted by "*Analog Low Volts*" holding register.
    II. When low analog voltage is detected an alarm will be triggered. This alarm can be read by reading "*Bit0"* of "*Analog Input (0 -10) User Discretes 01*" input register
    III. Low analog is enabled after preset delay on power up
    IV. User can select if the motor has to shut down or keep running (alarm, fail safe function activates)
        i. Fail safe can be enabled by setting "*Bit3*" of "*Analog Input (0 -10) User Flags 01*" holding register
        ii. If user selected enables fail safe function motor activates failsafe function when the analog voltage is below or equal to low analog threshold.
        iii. Fail safe function is not active on power up when analog voltage is zero.
        iv. If the analog voltage falls below low analog voltage threshold (when fail safe function is active) the demand output is set to fail safe demand, the demand can be adjusted by "*Fail Safe Demand*" holding register. This allows the fan/motor to keep spinning in critical application.
h) Analog input can be set as a digital input by settings "*Analog Volts Input Mode*" holding register to "*DIGITAL_MODE*"
    I. Digital input status can be read by reading "*Bit1*" of "*Analog Input (0 -10) User Discretes 01*"
    II. Note that this feature does not tie into digital inputs features

Analog Input (0-10V) Registers

MODBUS Address (decimal)	Register Name	Description	Units	Scale Factor	Min	Max	Default Settings (dec)	Data Type	Register Type

32808	Analog Input (0 - 10) User Discretes 01	Bit 0: analogLow (TRUE(1) analog input voltage is below "Analog Low Volts") Bit 1: analogVoltsDigitalInputON (TRUE (1) when analog input voltage is above 3.8V and "Analog Volts Input Mode" is set to "DIGITAL_MODE")	bool	1	0	65536	NA	u16	Input
32809	Analog Volts	Measured Analog 0-10V input volts	Volts	0.01	0.00	15.00	NA	u16	Input
32810	Analog Volts Percent	Analog Volts scaled to 0.00% - 100.00%. Where 100% corresponds to maximum voltage drive can measure.	%	0.01	0 (%)	10000	NA	u16	Input
32811	Analog Volts Demand Percent	Calculated demand % based on analog 0-10V input.	%	0.01	0 (%)	10000	NA	u16	Input
32872	Analog Input (0 -10) User Flags 01	Bit 0: enable_Analog, Bit 1: invertAnalog. Set this to TRUE if max volts corresponds to minimum demand and min volts corresponds to max demand. Bit 2: AnalogLowEnable. When set to TRUE(1), "analogVoltsLow" is set to TRUE when analog input is low. Bit 3: analogFailSafeEnable. When TRUE (1), command demand is set to "Failsafe Demand" when analog input is low Bit 4: is_analogMinDemandEnable. When TRUE (1), command demand is set to "Min Demand" when analog input is low Bit 5: is_minDemandAdjustableEnable. When TRUE(1), min demand can be adjusted by user. When FALSE(0) min demand will be set to the min commendable speed	bool	1	0	65536	1	u16	Holding
32873	Analog Volts Input Mode	Input mode for 0-10V Analog input, DISABLE_INPUT = 0, ANALOG_MODE = 1, DIGITAL_MODE = 2	pu	1	0	2	1	u16	Holding

32874	Min Volts	Min analog volts corresponding to Min Demand. This should be above ON_VOLTS. ON_VOLTS (0.77V) is minimum volts below which the demand is 0%.	volts	0.002583	ON Volts (300)	4095	767	u16	Holding
32875	Max Volts	Max acceptable analog volts corresponding to Max Demand. This is the max voltage customer can provide.	volts	0.002583	ON Volts + 387 (1V)	4095	3650	u16	Holding
32876	Min Demand	Min allowed demand	%	0.001525	0	65535	3676	u16	Holding
32877	Max Demand	Maximum allowed demand	%	0.001525	Min Demand	65535	65535	u16	Holding
32878	Analog Low Volts	Volts below which the analogLow discrete flag is set to TRUE	volts	0.002583	ON Volts	4095	250	u16	Holding
32879	Fail Safe Demand	Fail safe demand when analogLow flag is set to TRUE	%	0.001525	Min Demand	65535	32767	u16	Holding

4.1.2 Analog 4-20mA Input

a) User can read measured input amps at 4-20mA terminal by reading input register "*Analog Amps*"
b) The demand reference from analog input can be read by reading input register "*Analog Amps Demand Percent*"
c) User can enable or disable analog input by setting *Bit0* of "*Analog Input (4-20mA) User Flags 01*" holding register
d) User can invert analog input (when invert is active 4mA corresponds to 100% demand) by setting *Bit1* of "*Analog Input (4-20mA) User Flags 01*" holding register
e) Minimum and maximum analog mAmps can be adjusted by using holding registers *"Minimum mA"* and "*Max mA"* registers respectively
    I. For example, if hardware can read up to 21.5V, but the application only requires a range of 5 – 15mA, these settings can be modified to match these. Minimum mA correspond to min demand and maximum mA correspond to max demand. This will control the slope of the demand.
f) Allowable min and max demand correspond to min and max analog mA can be adjusted by *"Min Demand*" and *"Max Demand"* holding registers respectively.
g) User can enable analog input low alarm by setting "*Bit2*" of "*Analog Input (4-20mA) User Flags 01*" holding register
    V. Low analog input is detected if the measured analog mA is below the set threshold or if the mA is below threshold on power up. This threshold can be adjusted by "*Analog Low mA*" holding register.
    VI. When low analog mA is detected an alarm will be triggered. This alarm can be read by reading "*Bit0"* of "*Analog Input (4-20mA) User Discretes 01*" input register
    VII. Low analog is enabled after preset delay on power up
    VIII. User can select if the motor has to shut down or keep running (alarm, fail safe function activates)
        i. Fail safe can be enabled by setting "*Bit3*" of "*Analog Input (4-10mA) User Flags 01*" holding register
        ii. If user selected enables fail safe function motor activates failsafe function when the analog mA is below or equal to low analog threshold.
        iii. Fail safe function is not active on power up when analog mA is zero.
        iv. If the analog mA falls below low analog mA threshold (when fail safe function is active) the demand output is set to fail safe demand, the demand can be adjusted by "*Fail Safe Demand*" holding register. This allows the fan/motor to keep spinning in critical application.

h) Analog input can be set as a digital input by settings "*Analog mAmps Input Mode*" holding register to "*DIGITAL_MODE*"
   I. Digital input status can be read by reading "*Bit1*" of "*Analog Input (4-20mA) User Discretes 01*"
   II. Note that this feature does not tie into digital inputs features

Analog Input (4-20mA) Registers

MODBUS Address (decimal)	Register Name	Description	Units	Scale Factor	Min	Max	Default Settings (dec)	Data Type	Register Type
33064	Analog input (4- 20mA) User Discretes 01	Bit 0: analogAmpsLow, Bit 1: analogAmpsDigitalInputON	bool	1	0	65535	NA	u16	Input
33065	Analog Amps	Measure Analog 4-20 mAmps	mAmps	0.01	0	2200	NA	u16	Input
33066	Analog Volts	Analog mAmps converted to analog volts	volts	0.01	0	1500	NA	u16	Input
33067	Analog Volts Percent	Analog input mAmps scalled 0 to 100% (corresponds to max measurable mAmps)	%	0.01	0	10000	NA	u16	Input
33068	Analog Amps Demand Percent	Calculated demand % based on analog 4-20mA input. Scaled to 0 to 100% (corresponds to max speed or torque)	%	0.01	0	10000	NA	u16	Input
33128	Analog input (4- 20) User Flags 01	Bit 0: enable_Analog, Bit 1: invertAnalog. Set this to TRUE if max mAmps corresponds to minimum demand and min mAmps correspond sto max demand. Bit 2: AnalogLowEnable. When set to TRUE(1), "analogAmpsLow" is set to TRUE when analog input is low. Bit 3: analogFailSafeEnable. When TRUE (1), command demand is set to "Failsafe Demand" when analog input is low Bit 4: is_analogMinDemandEnable. When TRUE (1), command demand is set to "Min Demand" when analog input is low Bit 5: is_minDemandAdjustableEnable. When TRUE(1), min demand can be adjusted by user. When FALSE(0) min	bool	1	0	65535	1	u16	Holding

		demand will be set to the min commendable speed							
33129	Analog mAmps Input Mode	Input mode for 4-20mA analog input DISABLE_INPUT = 0, ANALOG_MODE = 1, DIGITAL_MODE = 2	enum	1	0	2	1	u16	Holding
33130	Minimum mA (in Counts)	Input mAmps that corresponds to minimum demand.	pu	0.005372	0	4095	740	u16	Holding
33131	Maximum mA (in Counts)	Input mAmps that corresponds to maximum demand	pu	0.005372	0	4095	3680	u16	Holding
33132	Minimum Demand	Minimum allowed demand (scaled to rated speed or torque)	pu	0.001526	0	65535	13107	u16	Holding
33133	Maximum Demand	Maximum allowed demand (scaled to rated speed or torque)	pu	0.001526	Min Demand	65535	65535	u16	Holding
33134	Analog Low mA	milli Amps threshold below which analog input is considered as low	pu	0.005372	0	4095	820	u16	Holding
33135	Failsafe Demand	Fail safe demand used when low analog input is detected	pu	0.001526	0	65535	32767	u16	Holding

4.1.3 Digital Inputs

a) Logical low (0) is when the input pin is open and logical high (1) is when the input is connected to voltage greater than 1.6Vdc.
b) Each of digital inputs can be configured to perform a specific function by setting "*Input Function1*"(DIN1), "*InputFunction2*"(DIN2) and "*InputFunction3*"(DIN3) holding registers
c) Each of the inputs can be configured to one of the following functions
    I. Input disable (no function)
    II. Motor Enable
    III. Discrete Speed Input (1-3)
d) Each individual input can be configured as 'normally open' or 'normally closed' by settings corresponding bit in "*Digital Input Polarity*" register
    I. Normally Open(0): When voltage is lower than 1.4Vdc function is active.
    II. Normally Closed(1): When voltage is higher than 1.6Vdc, function is active.
    III. For example, if DIN1 is set to "Motor Enable" and the polarity of the input is set to "Normally Closes (1)" and if the input is connected to 10V and a speed command is issued, motor will start spinning. If the DIN1 is opened (0V) motor will be disabled and will stop spinning.

4.1.3.1 Function Specification – Motor Enable:

a) Assume polarity is set to "Normally Closed"
b) Enable the motor when the input is connected to Logic High.
c) Keep motor in stop state when the input is Logic Low.

4.1.3.2 Function Specification - Direction of Rotation:

a) CCW is default direction of rotation when the input is Logic Low
b) Clockwise when the input is Logic High.
c) Clockwise and CCW are defined as observed from the Drive End.
d) Change of direction will only occur after the motor is in stop state.

Note: COPRA only supports CCW (observed from drive end)

4.1.3.3 Function Specification - Discrete Speed Input:

a) This feature will accept a minimum of 2 digital input and a max of 3 inputs.
b) Below tables defines the speeds for a 2 and 3 digital inputs respectively. The 2 digital input case actually matches the 3 digital input cases 0 through 3.

D2	D1	Demand Percentage
0	0	0
0	1	Demand Percentage 1
1	0	Demand Percentage 2
1	1	Demand Percentage 3

D3	D2	D1	Demand Percentage
0	0	0	0
0	0	1	Demand Percentage 1
0	1	0	Demand Percentage 2
0	1	1	Demand Percentage 3
1	0	0	Demand Percentage 4
1	0	1	Demand Percentage 5
1	1	0	Demand Percentage 6
1	1	1	Demand Percentage 7

Note: *Polarity set for the digital input will affect the combinates above*

c) "0" represents when the input is Logic Low. "1", when the input is Logic High.
d) The demand percentage for each of the cases can be adjusted by updating "*Demand Percent 1 – 8*" holding registers

Digital Inputs Registers

MODBUS Address (decimal)	Register Name	Description	Units	Scale Factor	Min	Max	Default Settings (dec)	Data Type	Register Type
33321	Discrete Demand Percentage	Demand when input mode is selected as a discrete demand input	pu	0.01	0	10000	NA	u16	Input
33322	Digital Inputs Filtered Status Bits	Reflects the value of each digital input; Bit0 – DIN1; Bit1 – DIN2; Bit2 - DIN3; Bit3 – PWM_IN; Bit4 – AUX_GPIO1; Bit5 - AUX_GPIO2; BIT6 - AUX_GPIO3; BIT7 - AUX_GPIO4;	Bit field	1	0	65535	NA	u16	Input
33385	Digital Input Enable (bits)	Each bit indicates the enable for its corresponding digital input (0 = DISABLED; 1 = ENABLED) Bit 0 = DIN1 Bit 1 = DIN2 Bit 2 = DIN3 Bit 3 to Bit 15 = UNUSED	Bit Field	1	0	65535	65535	u16	Holding
33386	Digital Input Polarity (bits)	Each bit indicates the polarity for its corresponding digital input (0 = Normally Open; 1 = Normally closed) Bit 0 = DIN1; Bit 1 = DIN2; Bit 2 = DIN3; Bit 3 = PWM_IN; Bit 4 -7 = RESERVED; Bit 8 to Bit 15 = UNUSED;	Bit Field	1	0	65535	8	u16	Holding
33387	Demand Percentage 1	Demand percentage corresponding to discrete demand digital inputs combination 1	pu	0.01	0	10000	0	u16	Holding
33388	Demand Percentage 2	Demand percentage corresponding to discrete demand digital inputs combination 2	pu	0.01	0	10000	4000	u16	Holding
33389	Demand Percentage 3	Demand percentage corresponding to discrete demand digital inputs combination 3	pu	0.01	0	10000	6000	u16	Holding

33390	Demand Percentage 4	Demand percentage corresponding to discrete demand digital inputs combination 4	pu	0.01	0	10000	8000	u16	Holding
33391	Demand Percentage 5	Demand percentage corresponding to discrete demand digital inputs combination 5	pu	0.01	0	10000	10000	u16	Holding
33392	Demand Percentage 6	Demand percentage corresponding to discrete demand digital inputs combination 6	pu	0.01	0	10000	10000	u16	Holding
33393	Demand Percentage 7	Demand percentage corresponding to discrete demand digital inputs combination 7	pu	0.01	0	10000	10000	u16	Holding
33394	Demand Percentage 8	Demand percentage corresponding to discrete demand digital inputs combination 8	pu	0.01	0	10000	10000	u16	Holding
33395	*InputFunction 1	Function for DIN1 (0 = input disabled, 1 = motor enable, 2 = RESERVED, 3 = RESERVED 4 = RESERVED, 5 = alarm reset, 6-8 = set demand input 0-2, 9 = RESERVED	enum	1	0	9	3	u16	Holding
33396	*InputFunction 2	Function for DIN2 (0 = input disabled, 1 = motor enable, 2 = RESERVED, 3 = RESERVED, 4 = RESERVED, 5 = alarm reset, 6-8 = set demand input 0-2, 9 = RESERVED	enum	1	0	9	6	u16	Holding
33397	*InputFunction 3	Function for DIN3 (0 = input disabled, 1 = motor enable, 2 = RESERVED, 3 = RESERVED, 4 = RESERVED, 5 = alarm reset, 6-8 = set demand input 0-2, 9 = RESERVED	enum	1	0	9	7	u16	Holding

Note: * Each digital input should be set to unique input functions

4.1.4 Digital Outputs

a) COPRA support the following digital outputs
   I. DOUT1 (output 1)
   II. LED output (output 2)
   III. Relay (output 3)
b) DOUT1 can be configurable to perform a specific function by settings "*Output Function 1*" holding register
   I. DOUT can be set to one of the following functions
      i. Fault Output
      ii. LED Output
      iii. PWM duty cycle output

iv. PWM frequency output

c) Each individual output can be configured as 'normally open' or 'normally closed' by setting corresponding bit in "*Output Polarity*" holding register
   I. Normally Open: Logical 1 is 'High Voltage'
   II. Normally Closed: Logical 1 is 'Low Voltage'
d) Each feature can be enabled or disabled in by setting corresponding bits in "*Output Enable*" holding register
e) Each feature can be set to a specific mode of operation. This mode of operation can be set by "*Output Function Mode x*" holding register. Where "x" corresponds to the output number identified in "a". See below for more information about the various modes each output supports

4.1.4.1 Function Specification – Fault/LED Output:

a) When the control is faulted this output provides feedback to user
b) The output can be set to one of three modes by updating
   I. Mode 1
      i. Output is low during normal operation
      ii. Output is high when controller is faulted
   II. Mode 2
      i. Output is high during normal operation
      ii. Output toggle between high and low when controller is faulted
   III. Mode 3
      i. Output is high during normal operation
      ii. Output toggles when controller is faulted
      iii. Output toggle sequence depends on the fault
      iv. If multiple faults happen one after other, finish current fault sequence and then move to the highest priority fault sequence.
      v. Only the highest priority fault sequence is shown in case of multiple faults.
c) Delay between toggle can be configured by adjusting "*Output High Time*" (amount of time the output stays high) and "*Output Low Time*" (amount of time the output stays low) holding registers
d) Delay between two consecutive toggle sequence in mode 3 can be configured by adjusting "*Output Sequence Delay"* holding register
e) If multiple faults happen one after other, finish current fault sequence and then move to the highest priority fault sequence.
f) Only the highest priority fault sequence is shown in case of multiple faults.

4.1.4.2 Function Specification - Relay Output:

a) The output default state can be configurable as Normally Open or Normally Closed by setting "*Bit2*" of "*Output Polarities*" holding register
b) Operating mode of the relay can be set by setting "*Output Function Mode 3*" holding register. It can be set to only of the following modes
   I. Mode 1: Fault Mode
      i. When the control is faulted the output state changes
      ii. Output state latches until the fault is cleared
   II. Mode 2: User enable mode
      i. User can enable the relay using Modbus command by setting "*Bit0*" of "*Digital Output User Flags 01*" holding register to "1"
      ii. Set this bit to a value of "0" to flip the state of the relay
   III. Mode 3: Speed threshold mode
      i. In this mode the relay is enabled once the output demand exceeds user set threshold
      ii. User can configure the threshold by adjusting "*Relay Enable Demand Threshold*" holding register

4.1.4.3 Function Specification - Duty Cycle Output:

a) Duty cycle of the PWM output is proportional to the demand output of motor and frequency is kept constant
b) User can configure minimum and maximum duty cycle corresponding to minimum and maximum demand by adjusting "*Minimum PWM Output Duty Cycle*" and "*Minimum PWM Output Duty Cycle*" holding registers
c) The default PWM output frequency can be set by adjusting "*PWM Output Frequency - Duty Cycle mode*" holding register

4.1.4.4 Function Specification - Frequency Output:

a) PWM output frequency is proportional to the output demand of the motor and duty cycle is kept constant
b) Two modes
   I. Output frequency is proportional to measured speed output (tach output)
   II. Output frequency is proportional to commanded demand
c) Min and max frequency corresponding to min and max demand are configurable by adjusting "*Minimum PWM Output Frequency*" and "*Maximum PWM Output Frequency*" holding registers
d) Frequency scale factor is configurable by adjusting "*PWM Frequency Output Scale Factor*" holding register. Output frequency will be equal to demand * Scale Factor.

Digital Outputs Registers

MODBUS Address (decimal)	Register Name	Description	Units	Scale Factor	Min	Max	Default Settings (dec)	Data Type	Register Type
33577	Digital Outputs Status	Each bit indicates status of digital output. Bit 0 = Digital output 0 status Bit0 - DOUT1; Bit1 - LED_ONBOARD; Bit2 - RELAY_OUT; Bit3 - MODBUS_EN; Bit4 - ISC1_SCL; Bit5 - ISC1_SDA; BIT6 - SPI1_SCK; BIT7 - SPI1_MISO; BIT8 - SPI1_MOSI; Bit9 - SPI1_CS	Bit Field	1	0 (LOW)	1023	NA	u16	Input
33578	PWM Output Duty Cycle	Actual PWM output duty cycle in percentage	%	0.01	0 (0%)	10000	NA	u16	Input
33579	PWM Duty Cycle Output Frequency	Actual PWM frequency output when in PWM duty cycle mode , max 1000Hz	Hz	1	0	1100	NA	u16	Input
33580	PWM Output Frequency	Actual PWM frequency output in PWM frequency output mode, max 1000Hz	Hz	1	0	1100	NA	u16	Input
33640	Digital Output User Flags 01	Bit 0: is_relayoutUserEnable – User can set the state of the relay by setting this bit when the mode of the relay is set to "Mode 2: User enable mode"	Bit Field	1	0	1	0	u16	Holding

		Bit 1 - Bit 15: UNUSED							
33641	Output Enable	Each bit enables an individual digital output. Bit0 - DOUT1; Bit1 - LED_ONBOARD; Bit2 - RELAY_OUT; Bit3 - MODBUS_EN; Bit4 to Bin9 - RESERVED; Bit10 to Bit15 - UNUSED	Bit Field	1	0 (LOW)	1023	65535	u16	Holding
33642	Output Polarity	Each bit indicates the polarity of the output. 0= Normally Open, 1 = Normally closed. Bit0 - DOUT1; Bit1 - LED_ONBOARD; Bit2 - RELAY_OUT; Bit3 - MODBUS_EN; Bit4 - ISC1_SCL; Bit5 - ISC1_SDA; BIT6 - SPI1_SCK; BIT7 - SPI1_MISO; BIT8 - SPI1_MOSI; Bit9 - SPI1_CS	Bit Field	1	0 (LOW)	1023	6	u16	Holding
33643	Minimum PWM Output Frequency	Minimum PWM output frequency corresponding to minimum demand in frequency mode	Hz	1	0	1000	50	u16	Holding
33644	Maximum PWM Output Frequency	Maximum PWM output frequency corresponding to maximum demand in frequency mode	Hz	1	Min Output Freq	1000	1000	u16	Holding
33645	Minimum PWM Output Duty Cycle	Minimum PWM output duty cycle corresponding to minimum demand, in duty cycle mode	%	0.01	0	9500	500	u16	Holding
33646	Maximum PWM Output Duty Cycle	Maximum PWM output duty cycle corresponding to maximum demand, in duty cycle mode	%	0.01	Min output DC	9500	9500	u16	Holding
33647	PWM Output Frequency - Duty Cycle mode	Set PWM output frequency in duty cycle mode	Hz	1	0	1000	100	u16	Holding
33648	PWM Frequency Output Scale Factor	Scale factor for tachometer: output frequency = speed * tachometer Scale Factor	pu	1	0	20	5	u16	Holding
33649	Output High Time	Time between high to low (ON time) for fault sequence	mSec	1	0	65535	500	u16	Holding
33650	Output Low Time	Delay between output toggle in mSec(from ON to OFF)	mSec	1	0	65535	500	u16	Holding
33651	Output Sequence Delay	Delay between two consecutive sequences (used for fault sequences)	mSec	1	0	65535	2000	u16	Holding

33652	Relay Enable Demand Threshold	Demand value that will change the state of the relay (if demand exceeds this value)	%	0.01	0	10000	5000	u16	Holding
33653	Output 1 (DOUT) - Function	Function of digital output 1 0 = disabled, 1 = Fault, 2 = RESERVED, 3 = LED, 4 = relay, 5 = PWM duty cycle, 6 = PWM freq	enum	1	0	6	1	u16	Holding
33662	Output 1 (DOUT) - Function Mode	• LED/Fault function modes: 1 = LOW when ok, HIGH when fault; 2 = HIGH during normal operation, blinking when faulted, 3 = HIGH during normal operation, blink sequence when faulted • PWM duty cycle modes: NA • PWM frequency output modes: 1 = Measured Speed Output; 2 = Commanded Demand output;	enum	1	1	3	3	u16	Holding
33663	Output 2 (LED) - Function Mode 2	Function mode if fault output 2 (1 = LOW when ok, HIGH when fault; 2 = HIGH during normal operation, blinking when faulted, 3 = HIGH during normal operation, blink sequence when faulted)	enum	1	1	3	3	u16	Holding
33664	Output 3(RELAY) - Function Mode 3	Function mode if for Relay (1 = INACTIVE when ok, ACTIVE when fault; 2 = User enable through Modbus by setting is_relayoutUserEnable to "1", 3 = Relay enable when Demand is above "Relay Enable Demand Threshold")	enum	1	1	3	1	u16	Holding
33671	PWM output min demand	Min demand % corresponds to min PWM duty cycle (min PWM frequency in frequency mode)	%	0.01	0	10000	5000	u16	Holding
33672	PWM output max demand	Max demand % corresponds to max PWM duty cycle (max PWM frequency in frequency mode)	%	0.01	PWM Output min demand	10000	10000	u16	Holding

4.1.5 Modbus

a) User can set the baud rate by updating "*Modbus Baud Rate*" holding register. While operating multiple units on a bus, it is recommended to use lower baud rate like 9600 baud.
b) User can choose data bits by updating "*Modbus Data Bits*" holding register
c) User can select the parity by updating "*Modbus Parity*" holding register

d) User can choose a global address that all unit in the daisy chain network has to respond to by updating "*Modbus Global Address*" holding register
e) User can set unique Modbus address by updating "*Modbus Address*" holding register
f) Loss of Modbus communication alarm can be enabled by setting "*Bit1*" of "*Modbus RTU User Flags 01*" holding register
   I. When loss of Modbus is enabled, if any Modbus message is not received within the timeout, is_modbusLoss bit is enabled to indicate loss of Modbus communication
   II. Timeout for loss of Modbus communication can be configured by updating "*Modbus Heart Beat Timeout*" holding register.
   III. Alarm for the Modbus loss status can be read at "*Bit0*" of "*Modbus Discrete Flags*" input register
   IV. Loss of Modbus demand can be enabled by setting "*Bit2*" of "*Modbus RTU User Flags 01*" holding register
   V. To run fan at a specific demand when loss of Modbus is detected, user can set demand under "*Modbus Loss Demand*" holding register. If this register is set to "0", fan will stop when loss of Modbus is detected (assuming loss of Modbus is enabled)

Note: Updated baud rate, data bits, parity, global address and Modbus address are used only after power cycle.

Modbus Module Registers

MODBUS Address (decimal)	Register Name	Description	Units	Scale Factor	Min	Max	Default Settings (dec)	Data Type	Register Type
33832	Modbus discrete flags	Bit 0: is_modbusLoss	Bit Field	1	0	1	NA	u16	Input
33833	Current Client Modbus Address	Current client address based	pu	1	0	247	NA	u16	Input
33834	Message Received	number of Modbus messages received	pu	1	0	65535	NA	u16	Input
33896	Modbus RTU User Flags 01	Bit 0: is_autoAddressingEnable; Bit 1: is_modbusLossEnable; Bit2: is_modbusLossDemandEnable	Bit Field	1	0	8	0	u16	Holding
33897	MODBUS Baud Rate	User set baud rate values accepted are: 48, 96, 72, 144, 192, 384, 576, 1152 Default Baud rate is 1152 (115200). Any value outside above will not be reflected & defaults to 115200 baud. Note : Baud rate change is only applied after power cycle.	enum	100	48	1152	1152	u16	Holding
33898	MODBUS Data Bits	Default is 8; Only 8 or 9 are selectable; Any value outside this range will revert to the previous value	enum	1	8	9	8	u16	Holding
33899	MODBUS Stop Bits	Default is 1 stop bit; only 1 or 2 are allowable; Any value outside this range will revert to the previous value or default of 1	enum	1	0	2	1	u16	Holding

		stop bit Note : Change is only applied after power cycle.							
33900	MODBUS Parity	Default is NO parity; Accepted values are 0 = NONE; 1 = ODD; 2 = EVEN; Any value outside this range will revert to the previous value or default value "NONE" Note : Change is only applied after power cycle.	enum	1	0	2	0	u16	Holding
33901	MODBUS Global Address	Server can send a broadcast message to all units on the bus using this address. All controllers will responds to this address irrespective of what the unique Modbus Address is. Allowable follower address range : 0-247 0 = default, 1-247 allowable. Note: 1. Default Follower Address is 247. Global address should not be used for Modbus address 2. Address change is only applied after power cycle.	pu	1	0	247	0	u16	Holding
33902	Modbus Address	User entered Modbus address Note : Address change is only applied after power cycle.	pu	1	0	247	247	u16	Holding
33905	Modbus Heart Beat Timeout	If no Modbus communication is received before this timeout is_modbusLoss flag is set to true.	sec	1	1	65535	20	u16	Holding
33906	modbusLossDemand	Demand % used when loss of Modbus is detected. is_modbusLossEnable and is_modbusLossDemandEnable need to be set to TRUE to use this demand %.	%	0.01	0	10000	0	u16	Holding

4.1.6 Demand Multiplexer

The module access demands from multiple sources and provides determines which source should be used to run the motor.

a) Module accepts demand commands from the following sources
   I. Modbus
   II. PWM Input
   III. Analog Input (0-10Vdc)
   IV. Analog Input (4-20mAdc)
   V. Digital inputs (discrete speeds)
   VI. Default Demand
b) Module allows user configurable unique priority for each of the above inputs.
   I. At any given point only one source is used to control the demand. This is based on the propriety and active input
   II. Each priority has to be unique and same value cannot be shared by two inputs
   III. Value of "1" is highest priority and a value of "6" is the lowest priority.
   IV. The priorities can be set by modifying the following holding registers
      i. *Analog 0-10V priority*
      ii. *Digital Inputs Priority*
      iii. *Modbus Priority*
      iv. *Analog 4-20mA Priority*
      v. *Default Demand Priority*
c) In case of applications that require running the motor when no demand source is present, user can use the "Default Demand" as a source.
   I. The default demand feature can be enabled by settings "*Bit0*" of "*Demand Multiplexer User Flags 01"* holding register
   II. The default demand percentage can be set by modifying "*Default Demand Percent*" holding register
   III. When other demand sources become active, default demand will be over taken by the active demand source
   IV. The default demand priority should be set to the lowest to allow other inputs to take over once they are present
d) User can read the current active demand source by reading register "*Current Demand Source*" input register
e) User can read the current demand value used to run the motor by reading "*Demand Value*" input register

Demand Multiplexer Registers

MODBUS Address (decimal)	Register Name	Description	Units	Scale Factor	Min	Max	Default Settings (dec)	Data Type	Register Type
34345	Demand Value	Demand value being applied to the motor	pu	0.01	0	10000	NA	u16	Input
34346	Current Demand Source	Source of the demand value (1 = 0-10V, 2 = 4-20mA, 3 = digital inputs, 4 = PWM, 5 = MODBUS, 6 = Default Demand)	enum	1	1	6	NA	u16	Input
34408	Demand Multiplexer User Flags 01	Bit 0: is_defaultDemandEnable	Bit Field	1	0	1	0	u16	Holding
34409	Demand Source	Demand source selected (0= Priority Mode 1 = 0-10V, 2 = 4-20mA, 3 = digital inputs, 4 = PWM, 5 = MODBUS, 6 = Default Demand)	enum	1	0	6	0	u16	Holding

34410	*Analog 0-10V Priority	Priority of analog 0-10V as the demand source (1 is highest, MAX is the lowest)	enum	1	1	6	2	u16	Holding
34411	*Digital Inputs Priority	Demand priority. 1 being highest 5 being lowest	enum	1	1	6	3	u16	Holding
34412	*MODBUS Priority	Demand priority. 1 being highest 5 being lowest	enum	1	1	6	1	u16	Holding
34413	*PWM Input Priority	Demand priority. 1 being highest 5 being lowest	enum	1	1	6	4	u16	Holding
34414	*Analog 4-20mA Priority	Demand priority. 1 being highest 5 being lowest	enum	1	1	6	5	u16	Holding
34415	*Default Demand Priority	Default Demand priority	enum	1	1	6	6	u16	Holding
34418	Default Demand Percent	Default demand value. If non-zero, this is used on power up	%	0.01	0	10000	0	u16	Holding

Note: * Each priority should be set to unique number between 1 and 6.

4.1.7 PWM Input

a) User can read measured PWM input duty cycle by reading "*PWM Stable Duty Cycle*" input register
b) User can read measured PWM input frequency by reading "*PWM Input Frequency*" input register
c) The demand reference from PWM input can be read by reading input register "*PWM Input Demand Percent*"
d) User can enable or disable PWM input by setting "*PWM Input Mode*" holding register to "*PWM_IN_PWM_MODE (1, enable)*" or "*PWM_IN_DISABLE_INPUT (0, disable)*"
e) User can invert PWM input (when invert is active 0% duty cycle corresponds to 100% demand) by setting "*Bit0*" of "*PWM Input User Flags 01*" holding register
f) Minimum allowed PWM frequency can be set by updating "*Minimum PWM Frequency*" holding register
g) Maximum allowed PWM frequency can be set by updating "*Maximum PWM Frequency*" holding register
h) Minimum and maximum duty cycle can be adjusted by using holding registers *"Minimum Duty Cycle"* and "*Max Duty Cycle"* registers respectively
    I. For example, if hardware can read up to 95%, but the application only requires a range of 10-80%, these settings can be modified to match these. Min duty cycle correspond to min demand and max duty cycle correspond to max demand. This will control the slope of the demand.
i) Allowable min and max demand correspond to min and max duty cycle can be adjusted by *"Minimum Demand*" and *"Maximum Demand"* holding registers respectively.
j) User can setup a low PWM frequency threshold by setting "PWM Loss of Frequency Threshold" holding register
    I. Low PWM frequency status can be read by reading "*Bit1"* of "*PWM Input User Discretes 01*" input register
k) User can enable PWM input low alarm by setting "*Bit1*" of "*PWM Input User Flags 01*" holding register
    I. Low PWM input is detected if the measured duty cycle is below the set threshold or if the voltage is below threshold on power up. This threshold can be adjusted by "*PWM Low Duty Cycle*" holding register.
    II. When low PWM unput is detected an alarm will be triggered. This alarm can be read by reading "*Bit0"* of "*PWM Input User Discretes 01*" input register

III. Low PWM input is enabled after preset delay on power up
IV. User can select if the motor has to shut down or keep running (alarm, fail safe function activates)
    i. Fail safe can be enabled by setting "*Bit3*" of "*PWM Input User Flags 01*" holding register
    ii. If user selected enables fail safe function motor activates failsafe function when the duty cycle is below or equal to low duty cycle threshold.
    iii. Fail safe function is not active on power up when PWM input duty cycle is zero
    iv. If the duty cycle falls below low duty cycle threshold (when fail safe function is active) the demand output is set to fail safe demand, the demand can be adjusted by "*PWM Fail Safe Demand*" holding register. This allows the fan/motor to keep spinning in critical application.

l) PWM input can be set as a digital input by settings "*PWM Input Mode*" holding register to "*PWM_IN_DIGITAL_MODE (2)*"
I. Digital input status can be read by reading "*Bit2*" of "*PWM Input User Discretes 01*"
II. Note that this feature does not tie into digital inputs features

PWM Input Registers

MODBUS Address (decimal)	Register Name	Description	Units	Scale Factor	Min	Max	Default Settings (dec)	Data Type	Register Type
34600	PWM Input User Discretes 01	Bit 0: is_pwmDutyCycleLow, Bit 1: is_pwmFrequencyLow, Bit 2: is_digitalInputOn	Bit Field	1	0	7	NA	u16	Input
34601	PWM Digital Input States	IF 1, PWM is in high state, if 2, PWM is in low state	enum	1	1 (HIGH)	2 (LOW)	NA	u16	Input
34602	PWM Input Frequency	Measured PWM input frequency	Hz	1	0	1000	NA	u16	Input
34603	PWM Stable Duty Cycle	Reading pf the previous duty cycle used to determine stability/bandwidth	pu	0.01	0	10000	NA	u16	Input
34604	PWM Input Demand Percentage	Output demand percentage given by PWM module	pu	0.01	0	10000	NA	u16	Input
34664	PWM Input User Flags 01	Bit 0: is_invertDigitalPWM, Bit 1: is_pwmLowEnable, Bit 2: is_pwmMinDemandEnable, Bit 3: is_pwmFailSafeEnable Bit 4: is_minDemandAdjustableEnable. When TRUE(1), min demand can be adjusted by user. When FALSE(0) min demand will be set to the min commendable speed	Bit Field	1	0	15	1	u16	Holding
34665	Input Mode	PWM_IN_DISABLE_INPUT = 0, PWM_IN_PWM_MODE = 1, PWM_IN_DIGITAL_MODE = 2	enum	1	0 (PWM)	1 (DIGITAL)	1	u16	Holding

34666	Minimum PWM Frequency	Minimum readable PWM frequency	Hz	1	0	1010	45	u16	Holding
34667	Maximum PWM Frequency	Maximum readable PWM frequency	Hz	1	0	1010	1010	u16	Holding
34668	Minimum Duty Cycle	Minimum acceptable duty cycle	%	0.01	0 (0.00%)	9500	500	u16	Holding
34669	Maximum Duty Cycle	Maximum acceptable duty cycle	%	0.01	0 (0.00%)	9500	9500	u16	Holding
34670	Minimum Demand	Minimum allowed demand, maps to minimum duty cycle	pu	0.001526	0	65535	3276	u16	Holding
34671	Maximum Demand	Maximum allowed demand, maps to maximum duty cycle	pu	0.001526	Min Demand	65535	65535	u16	Holding
34672	PWM Loss of Frequency Threshold	Frequency below which PWM input is considered lost	Hz	1	0	1000	0	u16	Holding
34673	PWM Low Duty Cycle	Minimum duty cycle below which the PWM input is considered lost	%	0.01	0	9500	300	u16	Holding
34674	PWM Failsafe Demand	Demand used when PWM input is lost	pu	0.001526	Min Demand	65535	5000	u16	Holding

4.1.8 Application Identification

a) User can access Application firmware version by reading "*Application Firmware Version*" input registers
b) User can set custom tag to identify the unit by updating "*Customer Tag01 – 08*" holding registers

Application Identification Registers

MODBUS Address (decimal)	Register Name	Description	Units	Scale Factor	Min	Max	Default Settings (dec)	Data Type	Register Type
35368	Application - ID User Discretes 01	Bit 0: is_enabled	pu	1	1	65535	NA	u16	Input
35369	Application Firmware Version minor - XX.YY.ZZ (LSB)	Application Firmware Revision - Bytes 1, 0 (LSB)	ASCII(2)	1	ASCII "00"	ASCII "99"	NA	ASCII	Input

35370	Application Firmware Version median - XX.YY.ZZ	Application Firmware Revision - Bytes 3, 2	ASCII(2)	1	ASCII "00"	ASCII "99"	NA	ASCII	Input
35371	Application Firmware Version major - XX.YY.ZZ (MSB)	Application Firmware Revision - Bytes 5, 4 (MSB)	ASCII(2)	1	ASCII "00"	ASCII "99"	NA	ASCII	Input
35372	Application firmware CRC minor	App code CRC minor	pu	1	0	65535	NA	u16	Input
35373	Application firmware CRC Major	App code CRC major	pu	1	0	65535	NA	u16	Input
35374	Application flash version minor - XXYY	Version of Modbus settings VV.XX.YY format. This register contains XX.YY. LSB is YY, MSB is XX in decimal format.	pu	1	0	65535	NA	u16	Input
35375	Application flash version major - 00VV	Version of Modbus settings VV.XX.YY format. This register contains VV. LSB is VV in decimal format, MSB is unused.	pu	1	0	65535	NA	u16	Input
35376	Application flash settings CRC minor	Flash Settings CRC	pu	1	0	65535	NA	u16	Input
35377	Application flash settings CRC major	Flash Settings CRC	pu	1	0	65535	NA	u16	Input
35378	Modbus Settings Rev - Minor (XXYY)	Version of Modbus settings VV.XX.YY format. This register contains XX.YY. LSB is YY, MSB is XX in decimal format.	pu	1	0	65535	NA	u16	Input
35379	Modbus Settings Rev - Major (00VV)	Version of Modbus settings VV.XX.YY format. This register contains VV. LSB is VV in decimal format, MSB is unused.	pu	1	0	65535	NA	u16	Input
35432	App ID - User Flags	User Settable Flags - Unused	pu	1	0	65535	0	u16	Holding
35433	Customer Tag 01	Customer set Tag 01	pu	1	0	65535	18241	ASCII	Holding
35434	Customer Tag 02	Customer set Tag 02	pu	1	0	65535	21536	ASCII	Holding
35435	Customer Tag 03	Customer set Tag 03	pu	1	0	65535	21573	ASCII	Holding
35436	Customer Tag 04	Customer set Tag 04	pu	1	0	65535	21280	ASCII	Holding
35437	Customer Tag 05	Customer set Tag 05	pu	1	0	65535	21061	ASCII	Holding
35438	Customer Tag 06	Customer set Tag 06	pu	1	0	65535	19791	ASCII	Holding
35439	Customer Tag 07	Customer set Tag 07	pu	1	0	65535	21587	ASCII	Holding
35440	Customer Tag 08	Customer set Tag 08	pu	1	0	65535	21827	ASCII	Holding

4.2 Register Group 2: Drive Interface

4.2.1 Drive Metering Data

a) User can read measured values by the drive through these registers

Drive Metering Registers

MODBUS Address (decimal)	Name	Description	Units	Scale Factor	Min	Max	Default Settings (dec)	Data Type	Register Type
41	mcState	Current Motor Control State IDEL – 0; START – 4; RUN – 6; STOP – 8; STOP_IDLE- 9; FAULT_NOW- 10; FAULT_OVER - 11	enum	1	0	255	NA	u16	Input
42	mcFaults 01	Fault 01 reported from the state machine bit0 - FOC duration fault; bit1 - Under voltage fault; bit2 - Over Voltage Fault; bit3 - Over Temperature Fault; bit4 - Speed Feedback Fault; bit 5 - Startup Fault; bit6 - Input Loss of Phase Fault; bit7 - Output Loss of Phase; bit 8 - Over Current Fault; bit 9 - Safety Core Fault; bit 10 - Internal Communication Loss Fault; bit 11 - Software Error Fault; bit 12 - bit 16 UNUSED	enum	1	0	255	NA	u16	Input
43	mcFaults 02	Fault 02 reported from the state machine which identifies the safety code faults. b0 - ADC_VREF_CHECK, b1 - ADC_STAGNANCY_CHECK, b2 - ADC_CURRENT_CHECK, b4 - ADC_MUX_CHECK, b5 - ADC_SHUNT_CURRENT_CHECK, b6 - ADC_LOCKED_ROTOR, b7 - ADC_OVER_SPEED_CHECK, b8 - ADC_MAX_STARTUP_CURRENT_CHECK, b9 - REG_CHECK_FAIL, b10 - CLOCK_SYSTIC_CHECK, b11 - CLOCK_LSI_CHECK, b12 -	enum	1	0	255	NA	u16	Input

		RAM_CHECK_FAIL, b13 - ROM_CHECK_MEERKAT_FAIL, b14 - ROM_CHECK_APPLICATION_FAIL							
44	mcAppState	State machine state 0 - MEM_INIT, 1- INIT, 2 - IDLE, 3 - PRE_START, 4 - OTF_STARTUP, 5 - MOTOR_RUNNING, 6 - STOP_MOTOR, 7 - MOTOR_STOPPING, 8 - MOTOR_BRAKING, 9 - CURRENT_DERATING, 10 - POWER_DERATING, 11 - TEMPERATURE_DERATING, 12 - MOTOR_START_RETRY, 13 - FAULT_REPORT, 14 - FAULT_PROCESS, 15 - FAULT_WAIT, 16 - TEMPERATURE_DERATING_LOW, 17 - TEMPERATURE_DERATING_TrimDown, 18 - TEMPERATURE_DERATING_Exit, 19 - TEMPERATURE_DERATING_TrimUp, 20 - TEMPERATURE_DERATING_MaintainTemp,	enum	1	0	255	NA	u16	Input
45	motorActualDirection	Motor direction, 9 = STD_ROTATION, 6 = REVERSE_ROTATION	enum	1	0	1	NA	u16	Input
46	dcBusVoltage	DC bus voltage	Volts DC	1	0	1000	NA	u16	Input
47	measuredSpeed	Measured speed feedback	RPM	1	-5000	5000	NA	s16	Input
48	measuredTorque	Measured torque feedback	pu	1	-32768	32768	NA	s16	Input
49	measuredInputPower	Calculated input power based on shaft power	Watts	1	0	10000	NA	s16	Input
50	ipmTemperature	Control IPM temperature. Note: some products do not have IPM temperature monitoring.	deg C	1	0	150	NA	s16	Input
51	phaseCurrentIa	Peak value of phase current Ia	mA	0.01	0	65535	NA	u16	Input
52	phaseCurrentIb	Peak value of phase current Ib	mA	0.01	0	65535	NA	u16	Input

53	mcFaults 01 history	saves last known Fault 01 reported by the state machine bit0 - FOC duration fault; bit1 - Under voltage fault; bit2 - Over Voltage Fault; bit3 - Over Temperature Fault; bit4 - Speed Feedback Fault; bit 5 - Startup Fault; bit6 - Input Loss of Phase Fault; bit7 - Output Loss of Phase; bit 8 - Over Current Fault; bit 9 - Safety Core Fault; bit 10 - Internal Communication Loss Fault; bit 11 - Software Error Fault; bit 12 - bit 16 UNUSED	Bit Field	1	0	65535	NA	u16	Input
54	mcFaults 02 History	saves last know Fault 02 reported by the safety core. b0 - ADC_VREF_CHECK, b1 - ADC_STAGNANCY_CHECK, b2 - ADC_CURRENT_CHECK, b4 - ADC_MUX_CHECK, b5 - ADC_SHUNT_CURRENT_CHECK, b6 - ADC_LOCKED_ROTOR, b7 - ADC_OVER_SPEED_CHECK, b8 - ADC_MAX_STARTUP_CURRENT_CHECK, b9 - REG_CHECK_FAIL, b10 - CLOCK_SYSTIC_CHECK, b11 - CLOCK_LSI_CHECK, b12 - RAM_CHECK_FAIL, b13 - ROM_CHECK_MEERKAT_FAIL, b14 - ROM_CHECK_APPLICATION_FAIL	Bit Field	1	0	65535	NA	u16	Input
65	measuredshaftPower	Calculated shaft power based on bus voltage and motor amps	Watts	1	0	10000	NA	s16	Input

4.2.2 Drive Identification

a) User can read firmware version of the drive by reading "*Firmware Rev*" input register

Drive Identification Registers

MODBUS Address (decimal)	Name	Description	Units	Scale Factor	Min	Max	Default Settings (dec)	Data Type	Register Type
554	Firmware Rev- Minor (YY)	Firmware version in in ASCII YY (VV.XX.YY)	pu	1	12336 00 ASCII	31097 zz ASCII	NA	ASCII	Input
555	Firmware Rev - Median (XX)	Firmware version in in ASCII YY (VV.XX.YY)	pu	1	12336 00 ASCII	31097 zz ASCII	NA	ASCII	Input
556	Firmware Rev - Major (VV)	Firmware version in in ASCII YY (VV.XX.YY)	pu	1	12336 00 ASCII	31097 zz ASCII	NA	ASCII	Input
557	Drive Firmware CRC - Minor(XXYY)	Motor firmware CRC	pu	1	0	65535	NA	u16	Input
558	Drive Firmware CRC - Major(00VV)	Motor firmware CRC	pu	1	0	65535	NA	u16	Input
559	Drive Flash Rev - Minor	Settings version In hex, but each nibble represents 0-9. VV.XX.YY format.	pu	1	0x0000	0x9999	NA	u16	Input
560	Drive Flash Rev - Major	Settings version In hex, but each nibble represents 0-9. VV.XX.YY format.	pu	1	0x0000	0x0099	NA	u16	Input
561	Drive Flash CRC - Minor	Motor flash settings CRC	pu	1	0	65535	NA	u16	Input
562	Drive Flash CRC - Major	Motor flash settings CRC	pu	1	0	65535	NA	u16	Input
563	Meerkat Firmware Rev - Minor	Firmware version in ASCII YY (VV.XX.YY)	pu	1	12336 00 ASCII	31097 zz ASCII	NA	ASCII	Input
564	Meerkat Firmware Rev - Median	Firmware version in ASCII XX (VV.XX.YY)	pu	1	12336 00 ASCII	31097 zz ASCII	NA	ASCII	Input
565	Meerkat Firmware Rev - Major	Firmware version in ASCII VV (VV.XX.YY)	pu	1	12336 00 ASCII	31097 zz ASCII	NA	ASCII	Input
566	Meerkat Firmware CRC - Minor	Motor Code CRC	pu	1	0	65535	NA	u16	Input
567	Meerkat Firmware CRC - Major	Motor Code CRC	pu	1	0	65535	NA	u16	Input
568	Meerkat Flash CRC - Minor	Motor flash settings CRC	pu	1	0	65535	NA	u16	Input
569	Meerkat Flash CRC - Major	Motor flash settings CRC	pu	1	0	65535	NA	u16	Input

570	Product Variant	GMI/P22 that represents product type and Motor variant (XXXYY) XXX = Product (GMI, P22 ….) (Max 654), Y = Variant (LS, MS, HS…) (Max 99)	pu	1	0	65535	NA	u16	Input
571	Load Variant	Fan/pump or any load coupled Version/Variant	pu	1	0	65535	NA	u16	Input
572	Hardware Variant	Defines if the drive is 230/460V, 1.3/4.5/8/12kW (XXXYY) XXXYY: XXX = kW/100 (Max 64kW); YY = Voltage/10 (Max 600V)	pu	1	0	65535	NA	u16	Input
573	Customer Variant	Firmware configuration for end user. Also used to identify different tester builds	pu	1	0	65535	NA	u16	Input
574	Modbus Settings Rev - Minor	Version of Modbus settings VV.XX.YY format. This register contains XX.YY. LSB is YY, MSB is XX in decimal format.	pu	1	0	65535	NA	u16	Input
575	Modbus Settings Rev - Major	Version of Modbus settings VV.XX.YY format. This register contains VV. LSB is VV in decimal format, MSB is unused.	pu	1	0	65535	NA	u16	Input

4.2.3 Drive Application Specific Registers

a) User can read minimum application speed of the drive by reading "*Minimum Application Demand*" input register
b) User can read minimum application speed of the drive by reading "*Maximum Application Demand*" input register

Application Specific Registers

MODBUS Address (decimal)	Name	Description	Units	Scale Factor	Min	Max	Default Settings (dec)	Data Type	Register Type
3882	Minimum Application Demand	Min application demand. Value in this register is depended on the control mode. If control mode is speed, the	rpm/ cfm/ torque	1	0	65535	NA	u16	Input

		value in this register in min application speed.							
3883	Maximum Application Demand	Max application demand. Value in this register is depended on the control mode. If control mode is speed, the value in this register in max application speed.	rpm/ cfm/ torque	1	0	65535	NA	u16	Input

4.2.4 Protections Parameters

a) User can read if any of the faults are latched by reading "*Bit0*" of "*Protections User Discretes01*" input register
b) User can read the current state of protection by reading "*Next Fault State*" input register
c) User can set number of restarts allowed per min by settings "*maxRestartsPerMin*" holding register
d) User can clear a latched fault by setting "*userFaultClear*" holding register to "1". This register is reset to "0" once the fault is cleared.

Protections Registers

MODBUS Address (decimal)	Name	Description	Units	Scale Factor	Min	Max	Default Settings (dec)	Data Type	Register Type
5928	Protections User Discretes01	b0: is_global_Fault_Latched (True when fault latch is active); b1: RESERVED	pu	1	0	1	NA	bool	Input
5929	Next Fault State	Boot (0), INIT(1), RUN (2), FAULT_REPORT(3), FAULT_PROCESS(4), FAULT_WAIT(5), FAULT_LATCH(6), FAULT_CLEAR(7)	enum	1	0	7	NA	enum	Input
5930	Fault retry count per min	Number of fault retry attempted per minute	pu	1	0	65535	NA	u16	Input
5992	Protections05 User Flags01	b0: RESERVED, b1: is_maxRestartsPerMinEnable	bool	1	0	1	0	u16	Holding
5993	maxRestartsPerMin	Maximum number of restarts/retries allowed within one minute. If exceeds this value, automatic fault latch occurs. This feature is	pu	1	0	255	10	u16	Holding

		enabled when is_maxRestartsPerMinEnable is set to TRUE							
5994	UserFaultClear	Set this to "1" to clear latched faults or faults	pu	1	0	1	0	u16	Holding

4.3 Register Group 3: Fan control

a) User can set the speed of the motor by settings "*Command Speed*" or by setting percentage demand by setting "*Command Demand*" holding register
   I. If both "*Command Speed*" and "*Command Demand*" registers are set, "*Command Demand*" will take priority over "*Command Speed*"
b) To start the fan user must set "*Start Command*" holding register to "1 (START)" and to stop the fan set this register to "0 (STOP)".
c) "*Direction*" holding register should be left at default of "9 (STD_ROTATION)". COPRA only supports STD_ROTATION (CCW (counter close wise), observed from drive side).

Registers for fan control

MODBUS Address (decimal)	Register Name	Description	Units	Scale Factor	Min	Max	Default Settings (dec)	Data Type	Register Type
35945	Command Speed	Modbus Command speed	pu	1	0	6000	0	u16	Holding
35946	Command Demand	Modbus command Demand	%	0.01	0	10000	0	u16	Holding
35947	Start Command	Modbus Start/Stop motor command. 1 = START, 0 = STOP	pu	1	0	1	0	u16	Holding
35948	Direction	Modbus Set direction of rotation. 9 = STD_ROTATION, 6 = REVERSE_ROTATION	pu	1	6	9	9	u16	Holding

MODBUS Address (decimal)	Name	Description	Units	Scale Factor	Min	Max	Default Settings (dec)	Data Type	Register Type
3882	Minimum Application Demand	Min application demand. Value in this register is depended on the control mode. If control mode is speed, the value in this register in min application speed.	rpm/ cfm/ torque	1	0	65535	0	u16	Input

3883	Maximum Application Demand	Max application demand. Value in this register is depended on the control mode. If control mode is speed, the value in this register in max application speed.	rpm/ cfm/ torque	1	0	65535	0	u16	Input

5 Appendix

5.1 Instructions To Change Customer Modbus Setting

5.1.1 Modbus Registers Overview

a) User can set the baud rate by updating "*Modbus Baud Rate*" holding register. While operating multiple units on a bus, it is recommended to use lower baud rate like 9600 baud.
b) User can choose data bits by updating "*Modbus Data Bits*" holding register
c) User can select the parity by updating "*Modbus Parity*" holding register
d) User can choose a global address that all unit in the daisy chain network has to respond to by updating "*Modbus Global Address*" holding register
e) User can set unique Modbus address by updating "*Modbus Address*" holding register
f) Loss of Modbus communication alarm can be enabled by setting "*Bit1*" of "*Modbus RTU User Flags 01*" holding register
    I. Timeout for loss of Modbus communication can be configured by updating "*Modbus Heart Beat Timeout*" holding register.
    II. Alarm for the Modbus loss status can be read at "*Bit0*" of "*Modbus Discrete Flags*" input register
    III. Loss of Modbus demand can be enabled by setting "*Bit2*" of "*Modbus RTU User Flags 01*" holding register
    IV. Demand used when loss of Modbus is detected can be set by updating "*Modbus Loss Demand*" holding register.

Note: Updated baud rate, data bits, parity, global address and Modbus address are used only after power cycle.

5.1.2 Modbus Register Specification

The following registers are available for Modbus settings.

MODBUS Address (decimal)	Register Name	Description	Units	Scale Factor	Min	Max	Default Settings (dec)	Data Type	Register Type
33832	Modbus discrete flags	Bit 0: is_modbusLoss	Bit Field	1	0	1	NA	u16	Input
33833	Current Client Modbus Address	Current server address that firmware accepts messaged from	pu	1	0	247	NA	u16	Input
33834	Message Received	number of Modbus messages received	pu	1	0	65535	NA	u16	Input
33896	Modbus RTU User Flags 01	Bit 0: is_autoAddressingEnable; Bit 1: is_modbusLossEnable; Bit2: is_modbusLossDemandEnable	Bit Field	1	0	3	0	u16	Holding

33897	MODBUS Baud Rate	User set baud rate values accepted are: 48, 96, 144, 192, 384, 576, 1152 Default Baud rate is 1152 (115200). Any value outside above will not be reflected & defaults to 115200 baud. Note : Baud rate change is only applied after power cycle.	enum	100	48	1152	1152	u16	Holding
33898	MODBUS Data Bits	Default is 8; Only 8 or 9 are selectable; Any value outside this range will revert to the previous value	enum	1	8	9	8	u16	Holding
33899	MODBUS Stop Bits	Default is 1 stop bit; only 0, 1 or 2 are allowable; Any value outside this range will revert to the previous value or default of 1 stop bit Note : Change is only applied after power cycle.	enum	1	0	2	1	u16	Holding
33900	MODBUS Parity	Default is NO parity; Accepted values are 0 = NONE; 1 = ODD; 2 = EVEN; Any value outside this range will revert to the previous value or default value "NONE" Note : Change is only applied after power cycle.	enum	1	0	2	0	u16	Holding
33901	MODBUS Global Address	Server can send a broadcast message to all units on the bus using this address. All controllers will responds to this address irrespective of what the unique Modbus Address is. Allowable follower address range : 0-247 0 = default, 1-247 allowable. Note: 1. Default Follower Address is 247. Global address should not be used for Modbus address 2. Address change is only applied after power cycle.	pu	1	0	247	0	u16	Holding
33902	Modbus Address	User entered Modbus address Note : Address change is only applied after power cycle.	pu	1	0	247	247	u16	Holding
33904	Modbus Auto Address Command	Modbus address command	pu	1	0	65535	0	u16	Holding
33905	Modbus Heart Beat Timeout	If no Modbus communication is received before this timeout is_modbusLoss flag is set to true.	sec	1	0	65535	20	u16	Holding

33906	modbusLossDemand	Demand used when loss of Modbus is detected and when is_modbusLossEnable and is_modbusLossDemandEnable are set to TRUE.	%	0.01	0	10000	0	u16	Holding

5.1.3 Settings Change Procedure

a) Connect RS485 cable to A, B, GND terminals. Refer to the manual for the connections.
b) Apply power (minimum of 60Vac is required to turn on the motor. If 1phase is used connect phase to L1, neutral to L2)
c) Change necessary holding register settings shown in above table
d) Save the value to memory by following the below procedure.
   I. Set register 0x8B69 (35689 dec) to value “1”.
   II. Wait for few seconds and read register 0x8B29 (35625 dec) to make sure that the register value is either “0” (FLASH_OK) or “14” (FLASH_WRITE_COMPLETE)
   III. Power cycle
   IV. Read registers changed in step 3 to confirm they took the change after power cycle.
   V. Note that the Modbus settings do not take affect till after power cycle. After power cycle you need to use the new settings to talk to the drive unless that settings are not saved correctly.

5.2 Instructions To Change Relay Setting

5.2.1 Relay Registers Overview

a) The output default state can be configurable by setting the polarity of the output. “Output Polarity” register can be used to set polarity of the output.
b) Relay can be set to one of the following three modes of operation.
   I. Mode 1: Fault Mode
      i. When a fault is active the relay will be activated.
   II. Mode 2:User enable mode (through Modbus)
      i. User can enable the relay using a Modbus command.
      ii. Use “Output Enable” register and the corresponding bit to activate the output.
   III. Mode 3: Speed threshold mode
      i. User can set a threshold at which the relay will be activated.
      ii. Use register “Relay Enable Demand Threshold” to set % of speed at which relay need to be activated.

5.2.2 Modbus Register Specification

The following registers are available for relay output under digital outputs settings.

MODBUS Address (decimal)	Register Name	Description	Units	Scale Factor	Min	Max	Default Settings (dec)	Data Type	Register Type
33577	Digital Outputs Status	Each bit indicates status of digital output. Bit 0 = Digital output 0 status Bit0 - DOUT1; Bit1 - LED_ONBOARD; Bit2 - RELAY_OUT; Bit3 - MODBUS_EN; Bit4 - ISC1_SCL; Bit5 - ISC1_SDA; BIT6 - SPI1_SCK; BIT7 - SPI1_MISO; BIT8 - SPI1_MOSI; Bit9 - SPI1_CS	Bit Field	1	0 (LOW)	1023	NA	u16	Input
33640	Digital Output User Flags 01	Bit 0: is_relayoutUserEnable Bit 1 - Bit 15: UNUSED	Bit Field	1	0	1	0	u16	Holding
33641	Output Enable	Each bit enables an individual digital output. Bit0 - DOUT1; Bit1 - LED_ONBOARD; Bit2 - RELAY_OUT; Bit3 - MODBUS_EN; Bit4 - ISC1_SCL; Bit5 - ISC1_SDA; BIT6 - SPI1_SCK; BIT7 - SPI1_MISO; BIT8 - SPI1_MOSI; Bit9 - SPI1_CS; Bit10 to Bit15 - UNUSED	Bit Field	1	0 (LOW)	1023	65535	u16	Holding
33642	Output Polarity	Each bit indicates the polarity of the output. 0= Normally Open, 1 = Normally closed. Bit0 - DOUT1; Bit1 - LED_ONBOARD; Bit2 - RELAY_OUT; Bit3 - MODBUS_EN; Bit4 - ISC1_SCL; Bit5 - ISC1_SDA; BIT6 - SPI1_SCK; BIT7 - SPI1_MISO; BIT8 - SPI1_MOSI; Bit9 - SPI1_CS	Bit Field	1	0 (LOW)	1023	3	u16	Holding
33652	Relay Enable Demand Threshold	Demand value that will change the state of the relay (if demand exceeds this value)	%	0.01	0	10000	5000	u16	Holding
33664	Output Function Mode 3	Function mode if for Relay (1 = LOW when ok, HIGH when fault; 2 = User enable through Modbus by setting is_relayoutUserEnable to “1”, 3 = Relay enable when Demand is above “Relay Enable Demand Threshold”)	enum	1	1	3	1	u16	Holding

5.2.3 Settings Change Procedure

a) Connect RS485 cable to A, B, GND terminals. Refer to the manual for the connections.
b) Apply power (minimum of 60Vac is required to turn on the motor. If 1phase is used connect phase to L1, neutral to L2)
c) Relay output mode can be changed by using register 0x8380(33664 dec). Change the value for this register to either 1, 2 or 3. Refer to above table for the descriptions of the modes.
d) If the relay output mode is set to 3 (demand threshold), use register 0x8374 (33652 dec) to set the threshold at which the relay needs to activate. This is in percentage of output speed. For example, if the percentage speed needs to be 20%, set the register value to 2000 (20.00%)
e) The polarity of the relay can be set using the register 0x836A (33642 dec) bit 2. Set this bit to 0 for normally open or 1 for normally closed.
f) Save the value to memory by following the below procedure.
    I. Set register 0x8B69 (35698 dec) to value “1”.
    II. Wait for few seconds and read register 0x8B29 (35625 dec) to make sure that the register value is either 0 (FLASH_OK) or 14 (FLASH_WRITE_COMPLETE)
    III. Power cycle
    IV. Read register set in steps 1, 2 and 3 are still same as what you have set them to.

Nicotra Gebhardt GmbH
Gebhardtstraße 19-25
74638 Waldenburg
Germany
Telefon +49 (0)7942 1010
Telefax +49 (0)7942 101170
E-Mail info.ng.de@RegalRexnord.com
www.nicotra-gebhardt.com