Index

REVISION INDEX						
REV.	**PAR.**	**DESCRIPTION**	**DATA**	**R**	**V**	**A**
1.0	-	First draft	19/04/2018	rz	fb	fb
1.1		General update	03/02/2023	ag	mb	mb

R = Editing

V = Verify

A = Approval

1. SAFETY PRECAUTIONS

Read carefully the following items so that you can safely use the drive avoiding causing injury to the operators, damaging the mechanic components driven by the drive or other objects in the area.

Make sure you that all warnings are correctly observed.

Safety marking legend:

MARKING	Meaning of the marking
⚠ WARNING	Indicates that errors that may lead to death or serious injury.
⚠ CAUTION	Indicates that errors that may lead to injury to people or damage objects.

MARKING	Meaning of the marking
⊘	Prohibition. Do not do it.
❗	Obligation. Follow the instruction.
⚠	Warning.

2. ABOUT THIS MANUAL

The present manual contains the information for the use of the AZ3 drive through the Modbus RTU protocol. Use this manual together with the drive’s user manual and the PC interface manual to understand the meaning of the parameters accessible via the Modbus communication.

⚠	Before trying to operate the drive using the Modbus protocol, the user must read the safety precautions reported in the drive’s user manual.

3. IMPORTANT PREFACE

3.1. Modbus VS PC Software Interface

The software PC interface is used for the tuning and the parametrization of the device and is in fact a Modbus master application. The software PC interface can also be used provided with additional components and features that can help the development of customer application based on Modbus.

The software application provides masking for drive parameters, the user enters the numeric value associated to a certain variable and the application does all the numeric and protocol conversion to send the data to the device.

3.2. Numerical representation

In this manual, decimal numbers are represented using digits without suffix while hexadecimal numbers are represented with the letter “h” after the number.

4. MODBUS INTRODUCTION

4.1. Modbus RTU

Modbus is a communication protocol for the exchange of data between a master (i.e. PLC, HMI, PC) and one or more slaves (i.e. motor drives, sensors, actuators): only the master can start the communication, asking for data from the slaves or setting the slaves parameters; no communication between slaves is possible. Both telegrams, request and answer, are composed by the same fields: address, function code, data and checksum. The only field with variable length is the data part of the telegram:

Address	Function code	Data	Checksum
8 bits	8 bits	N x 8 bits	16 bits

4.1.1. Address

The Modbus standard can manage the communication between one master and up to 247 slaves. In the request telegram, the master writes in the address field (1 byte) the address of the slave to which the message is destined while the slave, in the answer telegram, put in this field its own address.

The master can also send a message to all the slaves (broadcast telegram) writing zero in the address field. This special telegram does not generate any answer by the slaves.

4.1.2. Data

This is the only variable-size field in the telegram. Its content depends on the function code.

It is important to remember that Modbus does not specify how to represent the data exchanged in the communication. The device manufacturer defines the conversion rules between the raw data and their meaning in the real world.

4.1.3. Checksum (CRC)

The last field of the telegram is used for the transmission error check. The algorithm that is used for the computation of this two-byte field is the Cyclic Redundancy Check (CRC-16) deeply described in the Modbus standard. It is applied to the data bytes of the telegram (start, parity and stop bits are excluded) that is sent over the bus and added to the message. The receiver of the message recalculates the CRC-16 and compare the result with the content of the two-byte field: If the result is the same the telegram is considered valid, otherwise it is discarded.

4.1.4. Example of function code 03 (03h), read holding register

Read 3 holding register starting from word 15 of slave number 12.

Request:

Address	Function Code	index of 1st word		Number of words		CRC16	
0C	03	00	0F	00	03	34	D5
1 byte	1 byte	2 bytes		2 bytes		2 bytes	

Answer:

Address	Function code	Number of bytes sent	Word 15		Word 16		Word 17		CRC16	
0C	03	06	xx (hi)	xx (lo)	xx (hi)	xx (lo)	xx (hi)	xx (lo)	xx	xx
1 byte	1 byte	1 byte	2 bytes		2 bytes		2 bytes		2 bytes	

4.2. Communication with the drive

The PC-Interface can communicate with the drive through USB serial adapters from both RS485 channel and the TEM UART converter. These two channels cannot be used at the same time because they are access for the master communication and as the protocol states there can be only a single master unit.

5. COMMUNICATION SETUP

In order to setup a functional communication serial layer has to be configured.

5.1. Serial communication parameters

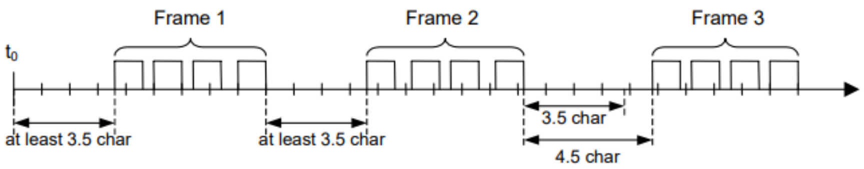
The serial parameters need to be set as follow:

- **BAUDRATE 19200** (bits per second)
- **PARITY** bit with **EVEN** parity check
- **DATABITS 8** (some conventions consider the parity bit inside the DATABITS in that case choose 9 if is the options)
- **STOP BITS ONE** stop bit

5.2. Modbus parameters

The default device address is 1, if the communication is with a master unit only and no other slaves are needed on the bus is suggested to keep this address number. In case of need of device address TEM will supply the necessary tools (software and firmware) to handle the addressing.

As states the Modbus RTU standard the reception of the frames shall be time-based.



The recommended time between frames shall be at least 1.750 ms, for the time between characters there shall be no more than 750 us.

The time between characters is usually used to detect cross talking of more devices that responds when they should not.

Is suggested for PC based master application to not check the inter character time because CRC control is enough to detect message integrity and also system timers often does not have the ability to do real time monitoring on such fast events.

Modern serial peripheral can grant transmission timings with very precise temporizations, and they transmit character with respecting the baudrate with very high precision so especially for one to one communication is not necessary to monitor the inter character time.

5.2.1. Implementation suggestion of PC based master application

Usually in PC application the serial port is accessible through the virtual object handled by the operative system (COMx in windows/dos system, dev/ttySx in Unix system). Using virtual objects, so communication timings could suffer of delay due to the buffering of data. In that case, if the drive user has to implement the Modbus reception could be in the need of doing some "stretching" on the timings. A possible reception can be also made implementing the byte-count, that means for master application that is possible to know the byte length of the answer that depends on the request, then after reception of the last byte counting, a silent interval of t-interframe shall be respected.

5.3. Supported modbus functions

5.3.1. MODBUS DATA TYPES

Name	Access type	Values	Description
COILS	Read / Write	0-1 OFF-ON	Single bit variable with read write access
HOLDING REGISTER	Read / Write	WORD (16 bits)	16 bits variable with read write access
INPUT REGISTERS	Read only	WORD (16 bits)	16 bits variable with read only access

Every Modbus data has its own logical addressing from 1 to 65536, that is mapped into a physical array address from 0 to 65535.

In the composition of functions, the byte-stream shall use the physical addressing, while the presentation state shall use the logical representation.

Example: HOLDING REGISTER N°1 is physically addressed at position zero.

Refer to the example reported below: it is a request of read holding registers for device 1. The index of the first word is 0x0fh, which means position 15 referred to physical addressing while it is requesting address from 16 in the logical representation. Logical representation is only used to avoid the notation of zero index so the front-end presentation con refer to first, second, third, etc. register, coil etc.

Address	Function Code	index of 1st word		Number of words		CRC16	
0C	03	00	0F	00	03	34	D5
1 byte	1 byte	2 bytes		2 bytes		2 bytes	

<u>N.B:</u> MODBUS DATA TYPES ARE RAW BYTES AND BIT THAT ARE TRANSFERRED BEETWEEN THE MASTER UNIT AND SLAVES UNIT ON THE BUS, THE CODING USED TO INTERPRET THE DATA IS MANUFACTURER SPECIFIC.

5.3.2. Modbus functions

The AZ3 drives parametrization and control rely mainly on the listed below functions of the Modbus standard.

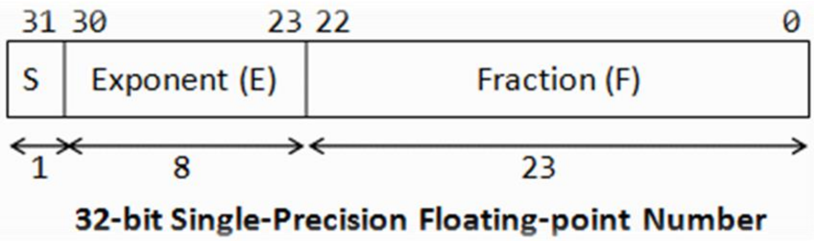
Function name	Function code	description
Read Coils	01 (0x01h)	Read one or more coils on the device
Read holding register	03(0x03h)	Read one or mere holding registers, max 124 in a single transaction
Read input registers	04(0x04h)	Read one or more input register, max 124 per transaction
Write single coil	05(0x05h)	Write the state (ON or OFF) of a single coil
Write single register	06(0x06h)	Write a single holding register (16 bits value)
Write multiple coils	15(0x0F)	Write one or more coils state with a single transaction
Write multiple registers	16(0x10)	Write one or more register up to 124 maximum in a single transaction
ADDITIONAL FUNCTION		
Read device identification	43(0x2Bh)	Supported only MEI type 0x0E with base identification

6. DEVICE DATA REPRESENTATION

6.1. Internal data type

The device internally used primitive data types listed as follow:

- **BIT**: Single bit (status 0-1 ON/OFF)
- **S8**: byte integer data with sign from -128 to 127
- **UNS8**: byte integer from 0 to 255
- **S16**: word data integer value with sign from -32768 to 32767
- **UNS16**: word data integer value from 0 to 65535
- **S32**: double word data integer from 0 to 4294967296
- **UNS32**: double word data integer with sign from -2147483648 to 2147483647
- **FLOATING POINT SINGLE PRECISION**: 32 bits standard representation IEEE 754



6.2. Modbus raw data handling

6.2.1. MODBUS BIT OPERATIONS

Coils and discrete inputs are physically store inside array of BYTE. The write and read access has to pass though block of coils or discrete inputs inside the transferred BYTE.

Example write multiple coils:

Address	Function Code	Starting index		Number of coils (bits)		Byte count	Byte 1	Byte 2	CRC16	
0C	0F	00	00	00	0A	02	01110010	10------	crcH	crcL
1 byte	1 byte	2 bytes		2 bytes		1 byte	1 byte	1 byte	2 bytes	

------ means that the byte is padded because only ten bits are requested in the write transaction.

6.2.2. MODBUS REGISTER OPERATIONS

Example of request for read 3 registers from address 0x0F (physical address 15).

Request:

Address	Function Code	Number of 1st word		Number of words		CRC16	
0C	03	00	0F	00	03	34	D5
1 byte	1 byte	2 bytes		2 bytes		2 bytes	

Answer:

Address	Function code	Number of bytes sent	Word 15		Word 16		Word 17		CRC16	
0C	03	06	xx (hi)	xx (lo)	xx (hi)	xx (lo)	xx (hi)	xx (lo)	xx	xx
1 byte	1 byte	1 byte	2 bytes		2 bytes		2 bytes		2 bytes	

In the answer is possible to se how register byte are packed into the frame message. A sequence of WORD with most significant bytes that are left aligned.

6.3. How to convert Modbus data into device data representation

COILS: are address as single bits so every coil corresponds to a single ON/OFF state variable accessible through read and write operations, every bit represent a binary variable.

HOLDING REGISTERS AND INPUT REGISTERS: holding register are accessible through read and write operations while input registers are read only. The access to the different type of data is done through different functions.

HOLDING REGISTER ARRAY			
LOGIC ADDRESS	PHYSICAL ADDRESS	DATA	RESULTING LABEL
1	0	16 bits	HR1
2	1	16 bits	HR2
3	2	16 bits	HR3
…	…	…	…

INPUT REGISTER ARRAY			
LOGIC ADDRESS	PHYSICAL ADDRESS	DATA	RESULTING LABEL
1	0	16 bits	IR1
2	1	16 bits	IR2
3	2	16 bits	IR3
…	…	…	…

COILS ARRAY			
LOGIC ADDRESS	PHYSICAL ADDRESS	DATA	RESULTING LABEL
1	0	1 bit	CL1
2	1	1 bit	CL2
3	2	1 bit	CL3
…	…	…	…

To convert WORD Modbus raw data registers into application data the following rules are to be used for the different primitive data types. WORD definition can be a 16bit holding register or a 16bit input register.

- **S8: the S8 variables**

WORD 1	WORD 2									WORD 3	WORD 4
	0	1	2	3	4	5	6	7			
2Byte	1Byte								1Byte	2Byte	2Byte

WORD 1	WORD 1									WORD 1	WORD 1
		0	1	2	3	4	5	6	7		
2Byte	1Byte	1byte								2Byte	2Byte

The darker slot represents the bit sign.
S8 and UNS8 variable are stored in half of a Modbus register that can be the most significant part or the less significant half of the register.

- **UNS8: byte integer from 0 to 255**

WORD 1	WORD 2									WORD 3	WORD 4
	0	1	2	3	4	5	6	7			
2Byte	1Byte								1Byte	2Byte	2Byte

WORD 1	WORD 2									WORD 3	WORD 4
		0	1	2	3	4	5	6	7		
2Byte	1Byte	1Byte								2Byte	2Byte

Conversion example (pseudo-code):

For 8 bits data in the lower part of the register:

Uint8_t [int8_t] dataEightBit = (WORD2 & 0x00FF);

For 8bits data in the upper part of the register:

Uint8_t [int8_t] dataEightBit = (WORD2 & 0xFF00);

- **S16: word data integer value with sign from -32768 to 32767**

WORD 1	WORD 2						WORD 3	WORD 4
	0	1	2	3	…	15		
2Byte	2Byte						2Byte	2Byte

- **UNS16: word data integer value from 0 to 65535**

WORD 1	WORD 2						WORD 3	WORD 4
	0	1	2	3	…	15		
2Byte	2Byte						2Byte	2Byte

Signed and unsigned 16bit data type are stored into the complete length of a Modbus register, in case of signed data the most significant bit is the sign bit (darker slot).

Conversion example (pseudo-code):

int16_t dataWithSign = WORD2;

Uint16_t dataUnsigned = WORD2;

- **S32: double word data integer from 0 to 4294967296**

WORD 1	WORD 2				WORD 3			WORD 4
	0	1		15	16		32	
2Byte	2Byte				2Byte			2Byte

- **UNS32: double word data integer with sign from -2147483648 to 2147483647**

WORD 1	WORD 2				WORD 3			WORD 4
	0	1		15	16		32	
2Byte	2Byte				2Byte			2Byte

Signed and unsigned 32 occupy two WORDS (two registers) the most significant 16bit of the 32bit variables is inside the higher index register. The darker slot in the signed representation is the sign bit.

Conversion example (pseudo-code):

Long datal = (WORD2 | (WORD3 << 16));

Or unsigned long in case of unsigned variable

- **FLOATING POINT SINGLE PRECISION: 32 bits standard representation IEEE 754**

WORD 1		WORD 2			WORD 3			WORD 4
	0	1		15	16		32	
2Byte		2Byte				2Byte		2Byte

The floating point is stored in binary format inside two registers, the most significant part is in the higher index register.

Conversion example (pseudo-code):

Long datal = (WORD2 | (WORD3 << 16));

Float_t dataf = rawBitsToFloat(datal);

6.4. Data tables

The Modbus master can access in read only mode to the device internal control variables through four configurable monitor variables. The configuration is made through a write operation on four selectors holding registers HR1…4 corresponding to monitor variables var1...4. At every number is associated a control variable that the user may be interested in monitoring.

IR x = input register with logical address x and physical address x-1

HR x = holding register with logical address x and physical address x-1

⚠	**Exceeding the variable range may cause incorrect behavior**

6.4.1. Monitor variables

IR 1	IR 2	IR 3	IR 4
Monitor var 1		Monitor var 2	
FLOAT 32 bit		FLOAT 32 bit	

IR 5	IR 6	IR 7	IR 8
Monitor var 2		Monitor var 3	
FLOAT 32 bit		FLOAT 32 bit	

HR 1		HR 2	
Channel_sel_1	Channel_sel_2	Channel_sel_3	Channel_sel_4
UNS8	UNS8	UNS8	UNS8
Range 0-80	Range 0-80	Range 0-80	Range 0-80

Channel_sel_1, Channel_sel_2, Channel_sel_3, Channel_sel_4:

Monitor var 1: the value of the control variable selected through Channel_sel_1

Monitor var 2: the value of the control variable selected through Channel_sel_2

Monitor var 3: the value of the control variable selected through Channel_sel_3

Monitor var 4: the value of the control variable selected through Channel_sel_4

7. PARAMETERS LIST

A list of all the parameters in object dictionary is below:

7.1. Holding registers

Variable	Type	Unit	Default	Range	Modbus logical address	Firmware version
SPIDAC CHANNEL SEL1	UNS8	-	0	0÷255	3L	1.50
SPIDAC CHANNEL SEL2	UNS8	-	0	0÷255	3H	1.50
SPIDAC CHANNEL SEL3	UNS8	-	0	0÷255	4L	1.50
SPIDAC CHANNEL SEL4	UNS8	-	0	0÷255	4H	1.50
SPIDAC CHANNEL 1 SCALE	FLOAT	-	1	-1.0E21÷1.0E21	5 6	1.50
SPIDAC CHANNEL 2 SCALE	FLOAT	-	1	-1.0E21÷1.0E21	7 8	1.50
SPIDAC CHANNEL 3 SCALE	FLOAT	-	1	-1.0E21÷1.0E21	9 10	1.50
SPIDAC CHANNEL 4 SCALE	FLOAT	-	1	-1.0E21÷1.0E21	11 12	1.50
ID KP	FLOAT	-	0.2	-1.0E21÷1.0E21	13 14	1.10
ID KI	FLOAT	-	200.0	-1.0E21÷1.0E21	15 16	1.10
ID KD	FLOAT	-	0	-1.0E21÷1.0E21	17 18	1.10
IQ KP	FLOAT	-	0.2	-1.0E21÷1.0E21	23 24	1.10
IQ KI	FLOAT	-	200.0	-1.0E21÷1.0E21	25 26	1.10
IQ KD	FLOAT	-	0	-1.0E21÷1.0E21	27 28	1.10
SPEED KP	FLOAT	-	0.001	-1.0E21÷1.0E21	33 34	1.10
SPEED KI	FLOAT	-	0.02	-1.0E21÷1.0E21	35 36	1.10
SPEED KD	FLOAT	-	0	-1.0E21÷1.0E21	37 38	1.10
PROFILE DECELERATION	UNS32	Speed user units/s	1000	0÷4.2E10	39 40	1.10

V REF SETPOINT	FLOAT	V	0	-1.0E21 ÷ 1.0E21	41 42	1.50
V REF POSITIVE RAMP	FLOAT	V	4	-1.0E21 ÷ 1.0E21	43 44	1.50
V REF NEGATIVE RAMP	FLOAT	V	-4	-1.0E21 ÷ 1.0E21	45 46	1.50
HEATSINK TEMPERATURE	FLOAT	°C	0	-1.0E21 ÷ 1.0E21	49 50	1.10
V BUS MAX	FLOAT	V	60	0÷100	59 60	1.10
ENCODER TYPE	UNS16	-	0	0÷10	61	1.10
SET POINT SELECTOR	S16	-	0	0÷10	62L	1.10
MOTOR POLES PAIR	UNS16	-	4	0÷16	64	1.10
ALIGNMENT CURRENT	FLOAT	A	1	-	65 66	1.50
NOMINAL CURRENT RMS	FLOAT	A	30	-100÷100	67 68	1.10
PID TUNING TYPE SELECTOR	S16	-	0	0÷2	73L	1.50
SQUARE WAVE SPEED PERIOD	UNS16	$\frac{1}{10}$ms	0	0÷65535	74	1.50
HIGH VALUE SPEED REF	FLOAT	Speed user units/s	0	-	75 76	1.50
LOW VALUE SPEED REF	FLOAT	Speed user units/s	0	-	77 78	1.50
SQUARE WAVE CURRENT PERIOD	UNS16	$\frac{1}{10}$ms	50	0÷65535	79	1.50
HIGH VALUE CURRENT REF	FLOAT	A	4	-	80 81	1.50
LOW VALUE CURRENT REF	FLOAT	A	-4	-	82 83	1.50
MAX MOTOR SPEED	UNS32	User units	3000	0÷4.2E10	84 85	1.10
PROFILE ACCELERATION	UNS32	Speed user units/s	1000	0÷4.2E10	86 87	1.10
GEAR RATIO MOTOR REVOLUTION	UNS32	rounds	1	0÷400000	88 89	1.10

GEAR RATIO SHAFT REVOLUTIONS	UNS32	rounds	1	0÷400000	90 91	1.10
VELOCITY ENCODER RESOLUTION MOTOR REVOLUTIONS	UNS32	round/s	1	0÷400000	92 93	1.10
VELOCITY ENCODER RESOLUTION INCREMENTS PER SECOND	UNS32	Incr/s	8192	0÷400000	94 95	1.10
FEED CONSTANT FEED	UNS32	-	1	0÷400000	96 97	1.10
FEED CONSTANT SHAFT REVOLUTIONS	UNS32	rounds	1	0÷400000	98 99	1.10
POSITION ENCODER RESOLUTION MOTOR REVOLUTIONS	UNS32	rounds	1	0÷400000	104 105	1.10
POSITION ENCODER RESOLUTIONS INCREMENTS	UNS32	-	8192	0÷400000	106 107	1.10
VELOCITY FACTOR NUMERATOR	S16	-	60	0÷32767	108	1.10
VELOCITY FACTOR DENOMINATOR	S16	-	1	0÷32767	109	1.10
POLARITY	UNS16	-	0	0÷65536	110	1.10
CONTROL WORD	UNS16	-	0	0÷65535	155	1.10
MODES OF OPERATION	S8	-	3	-128÷127	156L	1.10
TARGET VELOCITY	S32	User units	0	-2.0E6÷2.0E6	157 158	1.10
TORQUE LIMIT	FLOAT	Nm	-	-	160 161	1.50
TORQUE LIMIT FACTOR	S32	Nm	-	-2.0E6÷2.0E6	162 163	1.50
TDI CW MODE	UNS8	-	0	0÷255	172L	1.10
CUSTOM POSITION ACTUAL VALUE	S16	-	-	-32768÷32767	173	1.50
DIGITAL OUTPUT BYTES	UNS16	-	32	0÷65535	174	1.50
MOTOR TYPE	UNS16	-	10	0÷65535	175	1.10
MOTOR PHASE DISPLACEMENT	FLOAT	-	0	-	176 177	1.50

ZERO MARK DISPLACEMENT FROM U ZERO	FLOAT	-	0	-	178 179	1.50
T ALIGN	UNS16	A	20000	0÷65535	181	1.50
DELTA PHASING ANGLE ERROR	FLOAT	[rad]	0.18	-	182 183	1.50
ENCODER PHASE DISPLACEMENT	FLOAT	[rad]	3.14	-6.28÷6.28	184 185	1.10
OUTPUT 1 SELECTOR	UNS8	-	0	0÷255	186L	1.10
OUTPUT 2 SELECTOR	UNS8	-	0	0÷255	186H	1.10
ENCODER FEEDBACK MODE	UNS8	-	0	0÷255	187L	1.10
ABSOLUTE ENCODER NUMBER OF BITS	UNS16	-	25	0÷65535	188	1.50
ABSOLUTE ENCODER MULTI TURN STARTING BIT	UNS16	-	14	0÷65535	189	1.50
ABSOLUTE ENCODER MULTI TURN NUMBER OF BITS	UNS16	-	12	0÷65535	190	1.50
ABSOLUTE ENCODER SINGLE TURN STARTING BIT	UNS16	-	1	0÷65535	191	1.50
ABSOLUTE ENCODER SINGLE TURN NUMBER OF BITS	UNS16	-	13	0÷65535	192	1.50
ABSOLUTE ENCODER ENCODING	UNS16	-	1	0÷65535	193	1.50
ABSOLUTE ENCODER ANGLE PHASING OFFSET	FLOAT	-	0	-1.0E21÷1.0E21	194 195	1.50
SENSORLESS MEASURED SPEED	FLOAT	[rpm]	0	-1.0E21÷1.0E21	196 197	1.50
MODBUS ADDRESS	UNS16	-	1	0÷65535	198	1.68
MODBUS BAUD RATE	UNS32	-	19200	0÷4.2E10	199 200	1.68
PARITY	UNS16	-	-	0÷65535	201	1.68
ADDRESS 1	UNS16	-	0	0÷65535	202	1.68
ADDRESS 2	UNS16	-	0	0÷65535	203	1.68

ADDRESS 3	UNS16	-	0	0÷65535	204	
ADDRESS 4	UNS16	-	0	0÷65535	205	
MODBUS FAIL THRESHOLD TIME	UNS32	-	0	0÷4.2E10	204 205	1.68 (Solo su specifico cliente)
MAX FOLLOWING ERROR	UNS32	-	-	0÷4.2E10	206 207	1.68
FOLLOWING ERROR TIMEOUT	UNS16	-	-	0÷65535	208	1.68
FILTERED SAMPLES 1	S16	-	0	-32768÷32767	209	1.70
FILTERED SAMPLES 2	S16	-	0	-32768÷32767	210	1.70
CAN BIT RATE	UNS8	-	5	0÷255	226	1.68
BOOT CAN NODE	UNS8	-	1	0÷255	227	1.68
USER RANGE 1	FLOAT	-	1000	-1.0E21÷1.0E21	248 249	1.10
OFFSET 1	FLOAT	-	0	-1.0E21÷1.0E21	250 251	1.10
ZERO THRESHOLD 1	UNS16	-	5	0÷65535	252	1.10
USER RANGE 2	FLOAT	-	1000	-1.0E21÷1.0E21	253 254	1.10
OFFSET 2	FLOAT	-	-500	-1.0E21÷1.0E21	255 256	1.10
ZERO THRESHOLD 2	UNS16	-	1	0÷65535	257	1.10
SELECTOR 1	UNS8	-	0	0÷6	258L	1.10
SELECTOR 2	UNS8	-	0	0÷6	258H	1.10
AUX PWM OUT 1 SEL	UNS8	-	0	0÷6	265L	1.10
AUX PWM OUT 2 SEL	UNS8	-	0	0÷6	265H	1.10
BRAKE 1 MODE SEL	UNS8	-	1	0÷2	266L	1.10
BRAKE 1 POW VALUE	UNS8	%	100	0÷5	266H	1.10

BRAKE 2 MODE SEL	UNS8	-	1	0÷2	267L	1.50
BRAKE 2 POW VALUE	UNS8	%	100	0÷5	267H	1.50
TARGET TORQUE	S16	‰ of rated torque	0	-10000÷10000	269	1.10
MAX TORQUE	UNS16	‰ of rated torque	10000	0÷10000	270	1.10
MAX CURRENT	UNS16	‰ of rated current	10000	0÷10000	271	1.10
MOTOR RATED TORQUE	UNS32	mNm	2300	0÷4.2E10	272 273	1.10
MOTOR RATED CURRENT	UNS32	mA	16000	0÷4.2E10	274 275	1.10
TORQUE SLOPE	UNS32	‰ of rated torque/s	1000	0÷4.2E10	276 277	1.10
TARGET POSITION	S32	-	0	-2.0E6÷2.0E6	278 279	1.10
MIN POSITION RANGE LIMIT	S32	-	-100000	-2.0E6÷2.0E6	280 281	1.10
MAX POSITION RANGE LIMIT	S32	-	100000	-2.0E6÷2.0E6	282 283	1.10
SOFT MIN POSITION LIMIT	S32	-	0	-2.0E6÷2.0E6	284 285	1.10
SOFT MAX POSITION LIMIT	S32	-	0	-2.0E6÷2.0E6	286 287	1.10
MAX PROFILE VELOCITY	UNS32	User units	3000	0÷4.2E10	288 289	1.10
PROFILE VELOCITY	UNS32	-	1000	0÷4.2E10	290 291	1.10
POSITION KP	FLOAT	-	1.0	-1.0E21÷1.0E21	292 293	1.10
POSITION KI	FLOAT	-	0	-1.0E21÷1.0E21	294 295	1.10
POSITION KD	FLOAT	-	0	-1.0E21÷1.0E21	296 297	1.10
POSITION WINDOW	UNS32	-	-	0÷4.2E10	298 299	1.50
POSITION WINDOW TIME	UNS16	-	-	0÷65535	300	1.50
BRAKE DISENGAGE TIME	UNS16	ms	0	0÷65535	302	1.10

BRAKE ENGAGE TIME	UNS16	ms	0	0÷65535	303	1.10
SHUTDOWN OPTION CODE	S16	-	1	-32768÷32767	305	1.10
DISABLE OPERATION OPTION CODE	S16	-	1	-32768÷32767	306	1.10
BRAKING RESISTOR VALUE	FLOAT	Ω	0	0÷1.0E9	313 314	1.10
BRAKING RESISTOR MAX POWER	FLOAT	W	0	0÷1.0E9	315 316	1.10
ALIMENTATION POWER	FLOAT	W	3.36	0÷1.0E6	317 318	1.10
DI1 RISE SELECTOR	UNS8	-	0	0÷255	319L	1.10
DI1 FALL SELECTOR	UNS8	-	0	0÷255	319H	1.10
DI2 RISE SELECTOR	UNS8	-	0	0÷255	320L	1.10
DI2 FALL SELECTOR	UNS8	-	0	0÷255	320H	1.10
DI3 RISE SELECTOR	UNS8	-	0	0÷255	321L	1.10
DI3 FALL SELECTOR	UNS8	-	0	0÷255	321H	1.10
DI4 RISE SELECTOR	UNS8	-	0	0÷255	322L	1.10
DI4 FALL SELECTOR	UNS8	-	0	0÷255	322H	1.10
SPEED VALUE 1	S32	-	0	-2.0E6÷2.0E6	323 324	1.10
SPEED VALUE 2	S32	-	0	-2.0E6÷2.0E6	325 326	1.10
SPEED VALUE 3	S32	-	0	-2.0E6÷2.0E6	211 212	1.70
SPEED VALUE 4	S32	-	0	-2.0E6÷2.0E6	213 214	1.70
POSITION VALUE 1	S32	-	0	-2.0E6÷2.0E6	327 328	1.10
POSITION VALUE 2	S32	-	0	-2.0E6÷2.0E6	329 330	1.10
POSITION VALUE 3	S32	-	0	-2.0E6÷2.0E6	215 216	1.70

POSITION VALUE 4	S32	-	0	-2.0E6÷2.0E6	217 218	1.70
TORQUE VALUE 1	S16	-	0	-10000÷10000	331	1.10
TORQUE VALUE 2	S16	-	0	-10000÷10000	332	1.10
TORQUE VALUE 3	S16	-	0	-10000÷10000	219	1.70
TORQUE VALUE 4	S16	-	0	-10000÷10000	220	1.70
RETENTION VOLTAGE 1	FLOAT	$\%_{\text{of rated voltage}}$	15	0÷100	333 334	1.50
RETENTION VOLTAGE 2	FLOAT	$\%_{\text{of rated voltage}}$	15	0÷100	335 336	1.50
HEART BEAT TIME	UNS16	ms	1000	0÷65535	337	1.50
CONSUMER HEART BEAT 1	UNS32	-	134072	0÷4.2E10	338 339	1.50
CONSUMER HEART BEAT 2	UNS32	-	68536	0÷4.2E10	340 341	1.50
MOTOR WINDINGS RESISTANCE	FLOAT	Ω	0,07	-	342 343	1.50
MOTOR WINDINGS INDUCTANCE	FLOAT	H	1.9E-4	-	344 345	1.50
MOTOR MAGNETS FLUX	FLOAT	Wb	1.93E-4	-	346 347	1.50
SENSORLESS SPEED MULTIPLIER FACTOR	FLOAT	-	9.3E-3	-	348 349	1.50
SENSORLESS SPEED CORRECTION ACCUMULATOR	FLOAT	-	2.5E-4	-	350 351	1.50
FLUX WEAKENING PARAMETER	FLOAT	-	0.03	-	352 353	1.50
BOOT TIMEOUT	UNS16	ms	1000	0÷65535	441	1.68
BOOT CAN RX BOX	UNS16	-	520	0÷65535	442	1.68
BOOT CAN TX BOX	UNS16	-	776	0÷65535	443	1.68
BOOT OPTIONS	UNS8	-	1	0÷255	445L	1.68
BOOT DIRECT BOOT	UNS8	-	0	0÷255	445H	1.68

RESERVED 1	UNS8	-	0	0÷255	447L	1.68
RESERVED 2	UNS8	-	0	0÷255	447H	1.68
STORE ALL PARAMETERS	UNS16	-	0	0÷65535	448 449	1.10
DIGITAL OUTPUT 1 DIRECT COMMAND	UNS8	-	0	0÷255	455L	1.68
DIGITAL OUTPUT 2 DIRECT COMMAND	UNS8	-	0	0÷255	455H	1.68

7.2. Input registers

Variable	Type	Unit	Default	Range	Modbus logical address	Firmware version
PHASE DATA INCREMENTAL CAPTURES 12	S16	-	-	-32768÷32767	9	1.50
PHASE DATA INCREMENTAL CAPTURES 0	S16	-	-	-32768÷32767	10	1.50
PHASE DATA INCREMENTAL CAPTURES 1	S16	-	-	-32768÷32767	11	1.50
PHASE DATA INCREMENTAL CAPTURES 2	S16	-	-	-32768÷32767	12	1.50
PHASE DATA INCREMENTAL CAPTURES 3	S16	-	-	-32768÷32767	13	1.50
FASATURA DATA HALL CAPTURES 4	S16	-	-	-32768÷32767	14	1.50
PHASE DATA INCREMENTAL CAPTURES 5	S16	-	-	-32768÷32767	15	1.50
PHASE DATA INCREMENTAL CAPTURES 6	S16	-	-	-32768÷32767	16	1.50
PHASE DATA INCREMENTAL SECTOR CAPTURES 0	UNS8	-	-	0÷255	17L	1.50
PHASE DATA INCREMENTAL SECTOR CAPTURES 1	UNS8	-	-	0÷255	17H	1.50
PHASE DATA INCREMENTAL SECTOR CAPTURES 2	UNS8	-	-	0÷255	18L	1.50
PHASE DATA INCREMENTAL SECTOR CAPTURES 3	UNS8	-	-	0÷255	18H	1.50
PHASE DATA INCREMENTAL SECTOR CAPTURES 4	UNS8	-	-	0÷255	19L	1.50
PHASE DATA INCREMENTAL SECTOR CAPTURES 5	UNS8	-	-	0÷255	19H	1.50
CAN BITRATE INDEX	UNS16	-	-	0÷65535	20	1.50
ERROR REGISTER	UNS16	-	-	0÷65535	21	1.50
NODE STATE	UNS16	-	127	0÷65535	23	1.68
STATUS WORD	UNS16	-	0	0÷65535	25	1.10

DIGITAL INPUT STATUS	UNS16	-	32	0÷65535	28	-
BOOT CAN NODE	UNS8	-	1	0÷255	31L	1.58
BOOT MODBUS NODE	UNS8	-	1	0÷255	31H	1.58
BOOT BIT RATE	UNS8	Baud/sec	0	0÷255	34L	1.58
RES	UNS8	-	-	0÷255	34H	1.58
RAM ERRORS COUNTER	UNS8	-	0	0÷255	37L*	-
RAM INDEX	UNS8	-	0	0÷255	37H*	-
ERROR CODE 1	UNS16	-	-	0÷65535	47	1.10
ERROR CODE 2	UNS16	-	-	0÷65535	54	1.10
ERROR CODE 3	UNS16	-	-	0÷65535	61	1.10
ERROR CODE 4	UNS16	-	-	0÷65535	68	1.10
ERROR CODE 5	UNS16	-	-	0÷65535	75	1.10
ERROR CODE 6	UNS16	-	-	0÷65535	82	1.10
ERROR CODE 7	UNS16	-	-	0÷65535	89	1.10
ERROR CODE 8	UNS16	-	-	0÷65535	96	1.10
TORQUE ACTUAL VALUE	S16	-?-	0	-10000÷10000	115	1.10
CURRENT ACTUAL VALUE	S16	-?-	0	-10000÷10000	116	1.50
TORQUE DEMAND	S16	-?-	0	-32768÷32767	117	1.50
DC LINK CIRCUIT VOLTAGE	UNS32	mV	-	0÷4.2E10	118 119	1.50
POSITION ACTUAL VALUE	S32	-	0	-2.0E6÷2.0E6	120 121	1.10
VL VELOCITY ACTUAL VALUE	S16	-	-	-32768÷32767	122	1.50

VELOCITY ACTUAL VALUE	S32	-	0	-2 E9÷2.0E9	123 124	1.10
RMS CURRENT	FLOAT	A	-	-1.0E21÷1.0E21	125 126	1.50
TORQUE ACTUAL VALUE NM	FLOAT	Nm	-	-1.0E21÷1.0E21	127 128	1.50
TORQUE SHAFT VALUE	FLOAT	Nm	-	-1.0E21÷1.0E21	129 130	1.50
LINEAR FORCE ACTUAL VALUE	FLOAT	-	-	-1.0E21÷1.0E21	131 132	1.50
VBUS MEAN	FLOAT	V	-	-1.0E21÷1.0E21	133 134	1.50
HALL TRANSITION 0	FLOAT	-	-	-1.0E21÷1.0E21	157 158	1.50
HALL TRANSITION 1	FLOAT	-	-	-1.0E21÷1.0E21	159 160	1.50
HALL TRANSITION 2	FLOAT	-	-	-1.0E21÷1.0E21	161 162	1.50
HALL TRANSITION 3	FLOAT	-	-	-1.0E21÷1.0E21	163 164	1.50
HALL TRANSITION 4	FLOAT	-	-	-1.0E21÷1.0E21	165 166	1.50
HALL TRANSITION 5	FLOAT	-	-	-1.0E21÷1.0E21	167 168	1.50
HALL TRANSITION SECTOR 0	UNS8	-	-	-1.0E21÷1.0E21	169L	1.50
HALL TRANSITION SECTOR 1	UNS8	-	-	-1.0E21÷1.0E21	169H	1.50
HALL TRANSITION SECTOR 2	UNS8	-	-	-1.0E21÷1.0E21	170L	1.50
HALL TRANSITION SECTOR 3	UNS8	-	-	-1.0E21÷1.0E21	170H	1.50
HALL TRANSITION SECTOR 4	UNS8	-	-	-1.0E21÷1.0E21	171L	1.50
HALL TRANSITION SECTOR 5	UNS8	-	-	-1.0E21÷1.0E21	171H	1.50

7.3. Coils

Variable	Type	Unit	Default	Range	Modbus logical address	Firmware version
PHASING IN PROGRESS	COIL	-	0	-	0	1.10
START PHASING	COIL	-	0	-	1	1.10
ALIGNMENT REQUEST	COIL	-	0	-	2	1.50
REVERT ENCODER DIRECTION	COIL	-	0	-	3	1.50
INVERT POSITION FEEDBACK	COIL	-	0	-	4	1.10
AUTO-PHASED MOTOR	COIL	-	0	-	5	1.10
INVERT SPEED FEEDBACK	COIL	-	0	-	6	1.10
SENSORLESS ACTIVATION	COIL	-	0	-	7	1.50
SENSORLESS DEFLUX ACTIVATION	COIL	-	0	-	8	1.50
ABSOLUTE ENCODER PHASING START	COIL	-	0	-	9	1.50
RESET POSITION	COIL	-	0	-	11	1.50
CUT OFF POWER ON 0 VELOCITY	COIL	-	0	-	12	1.50
RESET ERROR	COIL	-	0	-	13	1.50
TORQUE LIMIT ON	COIL	-	0	-	14	1.50
SOFTWARE RESET	COIL	-	0	-	15	1.50
STO CONTROLWORD	COIL	-	0	-	16	1.10
READY TO SWITCH ON AT STARTUP	COIL	-	0	-	17	1.10
OPERATION ENABLE AT STARTUP	COIL	-	0	-	18	1.10
EMCY-CODE	COIL	-	0	-	19	1.50

BRAKE HOLDING FUNCTION ACTIVATION	COIL	-	0	-	20	1.50
ENCODER CHECK	COIL	-	0	-	21	1.50
THERMISTOR CHECK ACTIVATION	COIL	-	0	-	22	1.50
BRAKE ACTIVATION	COIL	-	0	-	23	1.10
BRAKE VALUE	COIL	-	0	-	24	1.10
EEPROM DIAGNOSTIC LOG CLEANING	COIL	-	0	-	25	1.50
EEPROM AUTOPHASING DATA CLEANING	COIL	-	0	-	26	1.10
EEPROM PARAMETERS CLEANING	COIL	-	0	-	27	1.10
EEPROM TOTAL CLEANING	COIL	-	0	-	28	1.10
DIRECTION CHANGE LOCK	COIL	-	0	-	30	1.70
DIGITAL INPUT HANDLING STO ENABLED	COIL	-	0	-	31	1.50
START	COIL	-	0	-	80	1.10
NOT READY TO SWITCH ON	COIL	-	0	-	81	1.10
SWITCH ON DISABLED	COIL	-	0	-	82	1.10
READY TO SWITCH ON	COIL	-	0	-	83	1.10
SWITCHED ON	COIL	-	0	-	84	1.10
OPERATION ENABLED	COIL	-	1	-	85	1.10
QUICK STOP ACTIVE	COIL	-	0	-	86	1.10
FAULT	COIL	-	0	-	87	1.10
FAULT REACTION ACTIVE	COIL	-	0	-	88	1.10
POWER OFF RESET	COIL	-	0	-	89	1.50