KOSMOS SERIE

CODE: 30727405 EDITION: 04.01.2012

INSTRUCTIONS MANUAL
PROTOCOL MODBUS-RTU

MICRA-E

DITEL

INDEX

1. MODBUS RTU PROTOCOL

The modbus RTU protocol is a serial data transmission format widely used in communications with PLC's but easy adaptable to other types of remote units thanks to its particular message structure (it doesn't operate with variables but with memory addresses).

The implementation of a universal standard protocol such as modbus allows the instrument be connected to existing systems without need for creating specific communication logic programs.
Moreover, the quantity and variety of processable data can be infinite since it is no more necessary to specify the desired parameter or parameters but only the first memory address and the quantity to transmit.

The following definitions of modbus protocol are given adapted to the KOSMOS SERIES instruments.

The MODBUS RTU protocol has no delimiter character at the beginning and the end of a message.
Each frame must be preceded by a silent interval of at least 3, 5 character times and must be finished with a silent of the same duration.

The first character of a frame is the slave address, followed by the function number, the information field and two bytes of error check code (CRC).

Ad	Fn	information	CRC

One character is composed of 10 bits: 1 start bit, 8 data bits and 1 stop bit.

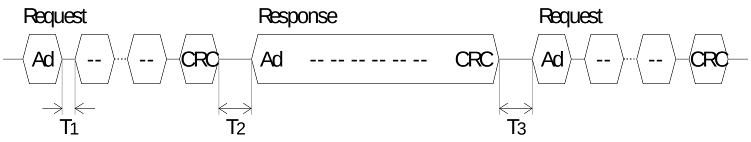
2. CRC GENERATION *(according to modbus RTU)*

1. Load a 16-bit register with H'FFFF (all '1'). Call it CRC.
2. Exclusive OR the first message byte with the low-order byte of the CRC register and put the result in CRC.
3. Shift the CRC register 1 bit towards the right (to the LSB) writing a '0' in the MSB.
4. If the new LSB is '0' repeat from point 3. If the LSB is '1', exclusive OR the CRC register with the 16-bit value H'A001 (1010 0000 0000 0001).
5. Repeat points 3 and 4 until completing a total of 8 shifts, after what the first byte has been processed.
6. Repeat points 2 to 5 with the second byte and continue these operations until all bytes on the message are processed.
7. Append the obtained CRC to the message so that the low order byte is transmitted first.

CRC Polynome: $2^{15} + 2^{13} + 2^{0}$
CRC Initial value: H'FFF

3. TIMING

The instrument detects the start of a message when receives a valid character (containing its address or the address 00) after a silent interval of at least 3,5 character times.
The end of a message is acknowledged by a silent interval of the same duration.



T1: time between two characters (min 0, max 3,5CT)
T2: time between query and response (min 3,5CT)
T3: time between response and next query (min 3,5CT)

CT = 1 character time

baud (bits/s)	3,5CT
1200	30ms
2400	15ms
4800	8ms
9600	4ms
19200	2ms

4. MODBUS FUNCTIONS

The modbus functions supported by the instrument are the following:

CODE	FUNCTION
03 (03H)	read n words
05 (05H)	force state
16 (10H)	write n words

Function 03 Used to read, in floating point format, dynamic variables such as display value, peak, valley, tare…, depending on model.
Function 05 Used to force the instrument to take an action such as 'make tare', 'reset tare', 'reset peak', 'reset latched setpoints'…, depending on model.
Function 16 Used to write in the instrument memory, programming variable in word format.

5. DATA TYPE AND MEMORY LOCATIONS

The instrument manages different types of data which are accessible to the user by programming, display or through the serial channel RS232C or RS485.
The data is located depending on type, in specific memory areas with unique addresses incrementing in 1 per byte from the zero address.
The figure below shows the user accessible memory areas, with the data types and modbus functions to be used each zone.

PROGRAMMING DATA **IN TABLE 4** *(READ AND WRITE)*	These are the data saved, in binary format in the instrument e2prom memory. Modbus function to used: • Read: 03 (03H). • Write: 16(10H).
RESERVED	
FLOATING POINT VARIABLES **IN TABLE 1** *(READ ONLY)*	These are measurement variables such as display, peak…, in floating point format (IEEE single precision). Modbus function to used: • Read : 03 (03H).
RESERVED	

CONTROL VARIABLES are not located to specific memory addresses but they consist of internal flags that the instrument interprets to take an action, and then deactivate the flag (see chapter 5.4).

5.1. Reading programming variables

FUNCTION 03 *(READ N WORDS))*

Request frame

1 byte	1 byte	2 bytes	2 bytes	2 bytes
slave address	MODBUS function	1st byte address according to table	number of words (nº bytes / 2)	CRC

Response frame

1 byte	1 byte	1 byte	n bytes	2 bytes
address	function	nº bytes	information read	CRC

Examples (see addresses in chapter 8)

Request input type of instrument at address 01

H'01	H'03	H'00	H'00	H'00	H'01	H'84	H'0A
Slave address	modbus function	1er byte address (Dec. 0 according to table)		number of words =1		CRC	

Response

H'01	H'03	H'02	H'01	H'00	H'B9	H'D4
address	function	nº bytes	Voltmeter AC		CRC	

5.2 Write programming variables

Data programming can be modified writing the wished direction using the function 10.

FUNCTION 10 *(WRITE N WORDS))*

Request frame

1 byte	1 byte	2 bytes	2 bytes	1 byte	n bytes	2 bytes
Slave address	MODBUS function	1st word address according to table	number of words (nº bytes / 2)	nº of bytes to write	Data to write	CRC

Response frame

1 byte	1 byte	2 bytes	2 bytes	2 bytes
address	function	1st address	number of words written	CRC

Examples (see addresses in chapter 8)

Programming of high level of brightness and round to 10 points

H’01	H’10	H’00	H’0C	H’00	H’01	H’02	H’00	H’02
Slave address	function MODBUS	1st byte address (dec.12 according to table)		number of words =1		number of bytes=2	Bright Hi	Round=10

H’27	H’5D
CRC	

Response

H’01	H’10	H’00	H’0C	H’00	H’01	H’C1	H’CA
address	function	1st byte address		number of words =1		CRC	

5.3 Reading dynamic variables

Dynamic variables are those that can vary based on the process without the user has access to modify them directly.
The dynamic variables are normally the values of display, peak, valley.... Their positions in memory specify in the tables of chapter 8.
These variables are asked for by means of the function MODBUS like type variables integer of 2 bytes (1 word).

FUNCTION 03 *(WRITE N WORDS))*

Request frame

1 byte	1 byte	2 bytes	2 bytes	2 bytes
Slave address	function MODBUS	1st word address according to the tables 1a and 1b	number of words (nº bytes / 2)	CRC

Response frame

1 byte	1 byte	1 byte	n bytes	2 bytes
address	function	nº bytes	Read Information	CRC

NOTE: In general these variables are transmitted without the decimal point of the display which is in other register.

Examples (see addresses in chapter 8)

Request of the value of display of instrument at address 01

H'01	H'03	H'00	H'3E	H'00	H'01	H'E5	H'C6
Slave address	function MODBUS	1st byte address (dec.62 according to table)		number of words =1		CRC	

Response (supposing display = +992)

H'01	H'03	H'02	H'03	H'E0	H'B9	H'3C
address	function	nº bytes	data (display value whole format)		CRC	

Request of the values of peak and valley to the instrument to address 01

H'01	H'03	H'00	H'43	H'00	H'02	H'35	H'DF
Slave address	function MODBUS	1st byte address (dec.67 according to table)		number of words =2		CRC	

Response (supposing peak=+1520, valley=-968)

H'01	H'03	H'04	H'5	H'F0	H'FC	H'38	H'BA	H'1E
address	function	nº bytes	data (peak value)		data (valley value)		CRC	

5.4 Commands (Control Variables)

It implies the execution of an order by the instrument. The address of the variable is replaced by the command indicated in the table below.

FUNCTION 05 *(TO FORCE STATE)*

Request format

1 byte	1 byte	2 bytes	2 bytes	2 bytes
Slave address	function MODBUS	address word (command)	put bit to '1' (fijo H'FF H'00)	CRC

Response format

1 byte	1 byte	2 bytes	2 bytes	2 bytes
address	function	command	bit to '1' (H'FF H'00)	CRC

Control Variables MICRA-E

Command	Command to execute	Request Format
110	RAZ PIC	01 05 00 6E FF 00 ED E7
111	RAZ VAL	01 05 00 6F FF 00 BC 27

6. FORMAT OF ERROR MESAGES

Error Codes

CODE	ERROR TYPE
01	Illegal function or incompatible with data
02	Illegal data or invalid CRC

Error 01 :

- Error 01 is produced when the referenced address is outside the valid range or the allowable range for the requested function.
- Error 01 is produced when the function is not supported by the instrument (03H, 05H o 10H).

Error 02 :

- Error 02 is produced when the number of bytes in a ‘write n bits’ request frame exceeds from the maximum number of writable bytes or from the limit of 250.
- Error 02 is produced when the calculated CRC doesn’t match the received message CRC.

Response Format

1 byte	1 byte	1 byte	2 bytes
Slave address	function +H'80	Code error	CRC

Example

Data Error (code 02) message from slave 01 as response to a function 03.

H'01	H'83	H'02	H'00	H'2C
address	H'03+H'80	code	CRC	

Example:

Request input type of instrument address 01

H'01	H'03	H'00	H'00	H'00	H'01	H'84	H'0B
Slave address	function MODBUS	1st byte address (d'0 According table)		number of words =1		CRC erroneous	

Message Error

H'01	H'83	H'02	H'C0	H'F1
address	H'03+H'80	code	CRC	

7. USING STANDARD COMMUNICATIONS PROGRAMS

Introduction

It is found in the market a wide variety of standard programs that allow collecting data through the serial port of a computer to be used in a virtual control panel on the screen of the PC. The modbus communications protocol is used as a universal tool for connecting these programs with all types of remote instrumentation capable of managing this protocol.

In general, such programs continuously scan, at a fixed rate, the setup addresses according to the type of data inside. The type of the data depends on the modbus function used to request it.
The information is continuously updated in the data bus according to the frame format described in section 5. The program user only has to extract each variable from the frame to be represented on the PC screen in the desired format.
Due to that not always the standard modbus data formats match with those of most instrumentation units, in the case of KOSMOS instruments, the data extraction must be made under the following considerations.

3.2. COLLECTING FLOATING POINT VARIABLES USING FUNCTION 03

In modbus format, the variables by function 03 are represented by words (1 word=2 bytes) and they are contained in addresses that increment in 1 per word, that is, each address contains 2 bytes.
In the instrument, the addresses are incremented in 1 per byte that means that each address contains one byte.
The result of this is that a floating point variable takes 4 addresses in the instrument and 2 addresses in modbus representation.

8. ADRESS OF VARIABLES MEMORY

Programming Data (Reading / Writing)

BYTE	MODBUS	Variable	Description
0	0	(char) Input	0= Voltmeter DC 1= Voltmeter AC 2= Ampmeter DC 3= Ampmeter AC
1		(char) Range	Volverte (Input = 0 ó 1) 0= 600V 1= 200V 2= 20V 3= 2V Ampmeter (Input = 2 ó 3) 0= 5A 1= 1A 2= 0.2A 3= Shunt / 100mV 4= Shunt / 60mV 5= Shunt / 50mV
2	1	(char) Input 1 [0]	Digit 4 (MSB)= 0, 1, -1 (= 10), " - " (= 11)
3		(char) Input 1 [1]	Digit 3= 0 a 9
4	2	(char) Input 1 [2]	Digit 2= 0 a 9
5		(char) Input 1 [3]	Digit 1= 0 a 9
6	3	(char) Input 1 [4]	Digit 0 (LSB)= 0 a 9
7		(char) Input 2 [0]	Digit 4 (MSB)= 0, 1, -1 (= 10), " - " (= 11)
8	4	(char) Input 2 [1]	Digit 3= 0 a 9
9		(char) Input 2 [2]	Digit 2= 0 a 9
10	5	(char) Input 2 [3]	Digit 1= 0 a 9
11		(char) Input 2 [4]	Digit 0 (LSB)= 0 a 9

BYTE	MODBUS	Variable	Description
12	6	(char) Display 1 [0]	Digit 4 (MSB)= 0, 1, -1 (= 10), " - " (= 11)
13		(char) Display 1 [1]	Digit 3= 0 a 9
14	7	(char) Display 1 [2]	Digit 2= 0 a 9
15		(char) Display 1 [3]	Digit 1= 0 a 9
16	8	(char) Display 1 [4]	Digit 0 (LSB)= 0 a 9
17		(char) Display 2 [0]	Digit 4 (MSB)= 0, 1, -1 (= 10), " - " (= 11)
18	9	(char) Display 2 [1]	Digit 3= 0 a 9
19		(char) Display 2 [2]	Digit 2= 0 a 9
20	10	(char) Display 2 [3]	Digit 1= 0 a 9
21		(char) Display 2 [4]	Digit 0 (LSB)= 0 a 9
22	11	(char) Decimal Point	0= 19999 1= 1999.9 2= 199.99 3= 19.999 4= 1.9999
23		(char) Filter P	0 (Without filter) a 9 (Filter fuerte)
24	12	(char) Brightness	0= Brightness high 1= Brightness low
25		(char) Round	0= Without Round 1= 5 points 2= 10 points
26	13	(char) Minutes Eco [0]	Digit 1 (MSB)= 0 a 9
27		(char) Minutes Eco [1]	Digit 0 (LSB)= 0 a 9
28	14	(char) Modo Eco	0= OFF 1= ON
29		(char) Logic Function 1	0 a 16
30	15	(char) Logic Function 2	0 a 16
31		(char) Logic Function 3	0 a 16

BYTE	MODBUS	Variable	Description
32	16	(char) Setpoint FLog	0= Set1 1= Set2 2= Set3 3= Set4
33		(char) Print Date	0= No, 1= Yes
34	17	(char) Color PROG	0= Red 1= Green 2= Orange
35		(char) Color RUN	0= Red 1= Green 2= Orange
36	18	(char) Locks [1]	Bit 7= - Bit 6= - Bit 5= Lock Display Bit 4= Lock Input Bit 3= Lock Setpoint 4 Bit 2= Lock Setpoint 3 Bit 1= Lock Setpoint 2 Bit 0= Lock Setpoint 1
37		(char) Locks [0]	Bit 7= - Bit 6= - Bit 5= - Bit 4= Lock TOTAL Bit 3= Lock Analog Output Bit 2= Lock Logics Functions Bit 1= Lock Output RS2 / RS4 Bit 0= Lock Prog. Direct Setpoints

BYTE	MODBUS	Variable	Description
38	19	(char) Code [0]	Digit 3 (MSB)= 0 a 9
39		(char) Code [1]	Digit 2= 0 a 9
40	20	(char) Code [2]	Digit 1= 0 a 9
41		(char) Code [3]	Digit 0 (LSB)= 0 a 9
42	21	(char) ON / OFF Setpoint 1	0= OFF 1= ON
43		(char) ON / OFF Setpoint 2	0= OFF 1= ON
44	22	(char) ON / OFF Setpoint 3	0= OFF 1= ON
45		(char) ON / OFF Setpoint 4	0= OFF 1= ON
46	23	(char) HI / LO Setpoint 1	0= HI 1= LO
47		(char) HI / LO Setpoint 2	0= HI 1= LO
48	24	(char) HI / LO Setpoint 3	0= HI 1= LO
49		(char) HI / LO Setpoint 4	0= HI 1= LO
50	25	(c2har) Dly / Hys Setpoint 1	0= Hys 1= Dly
51		(char) Dly / Hys Setpoint 2	0= Hys 1= Dly
52	26	(char) Dly / Hys Setpoint 3	0= Hys 1= Dly
53		(char) Dly / Hys Setpoint 4	0= Hys 1= Dly

BYTE	MODBUS	Variable	Description
54	27	(char) Value Setpoint 1 [4]	Digit 4 (MSB)= 0, 1, -1 (= 10), " - " (= 11)
55		(char) Value Setpoint 1 [3]	Digit 3= 0 a 9
56	28	(char) Value Setpoint 1 [2]	Digit 2= 0 a 9
57		(char) Value Setpoint 1 [1]	Digit 1= 0 a 9
58	29	(char) Value Setpoint 1 [0]	Digit 0 (LSB)= 0 a 9
59		(char) Value Setpoint 2 [4]	Digit 4 (MSB)= 0, 1, -1 (= 10), " - " (= 11)
60	30	(char) Value Setpoint 2 [3]	Digit 3= 0 a 9
61		(char) Value Setpoint 2 [2]	Digit 2= 0 a 9
62	31	(char) Value Setpoint 2 [1]	Digit 1= 0 a 9
63		(char) Value Setpoint 2 [0]	Digit 0 (LSB)= 0 a 9
64	32	(char) Value Setpoint 3 [4]	Digit 4 (MSB)= 0, 1, -1 (= 10), " - " (= 11)
65		(char) Value Setpoint 3 [3]	Digit 3= 0 a 9
66	33	(char) Value Setpoint 3 [2]	Digit 2= 0 a 9
67		(char) Value Setpoint 3 [1]	Digit 1= 0 a 9
68	34	(char) Value Setpoint 3 [0]	Digit 0 (LSB)= 0 a 9
69		(char) Value Setpoint 4 [4]	Digit 4 (MSB)= 0, 1, -1 (= 10), " - " (= 11)
70	35	(char) Value Setpoint 4 [3]	Digit 3= 0 a 9
71		(char) Value Setpoint 4 [2]	Digit 2= 0 a 9
72	36	(char) Value Setpoint 4 [1]	Digit 1= 0 a 9
73		(char) Value Setpoint 4 [0]	Digit 0 (LSB)= 0 a 9
74	37	(char) Dly / Hys Set 1 [4]	Digit 4 (MSB)= 0, 1, -1 (= 10), " - " (= 11)
75		(char) Dly / Hys Set 1 [3]	Digit 3= 0 a 9
76	38	(char) Dly / Hys Set 1 [2]	Digit 2= 0 a 9
77		(char) Dly / Hys Set 1 [1]	Digit 1= 0 a 9
78	39	(char) Dly / Hys Set 1 [0]	Digit 0 (LSB)= 0 a 9
79		(char) Dly / Hys Set 2 [4]	Digit 4 (MSB)= 0, 1, -1 (= 10), " - " (= 11)
80	40	(char) Dly / Hys Set 2 [3]	Digit 3= 0 a 9
81		(char) Dly / Hys Set 2 [2]	Digit 2= 0 a 9

BYTE	MODBUS	Variable	Description
82	41	(char) Dly / Hys Set 2 [1]	Digit 1= 0 a 9
83		(char) Dly / Hys Set 2 [0]	Digit 0 (LSB)= 0 a 9
84	42	(char) Dly / Hys Set 3 [4]	Digit 4 (MSB)= 0, 1, -1 (= 10), " - " (= 11)
85		(char) Dly / Hys Set 3 [3]	Digit 3= 0 a 9
86	43	(char) Dly / Hys Set 3 [2]	Digit 2= 0 a 9
87		(char) Dly / Hys Set 3 [1]	Digit 1= 0 a 9
88	44	(char) Dly / Hys Set 3 [0]	Digit 0 (LSB)= 0 a 9
89		(char) Dly / Hys Set 4 [4]	Digit 4 (MSB)= 0, 1, -1 (= 10), " - " (= 11)
90	45	(char) Dly / Hys Set 4 [3]	Digit 3= 0 a 9
91		(char) Dly / Hys Set 4 [2]	Digit 2= 0 a 9
92	46	(char) Dly / Hys Set 4 [1]	Digit 1= 0 a 9
93		(char) Dly / Hys Set 4 [0]	Digit 0 (LSB)= 0 a 9
94	47	(char) Color Setpoint 1	0= No change 1= Red 2= Green 3= Orange
95		(char) Color Setpoint 2	0= No change 1= Red 2= Green 3= Orange
96	48	(char) Color Setpoint 3	0= No change 1= Red 2= Green 3= Orange
97		(char) Color Setpoint 4	0= No change 1= Red 2= Green 3= Orange
98	49	(char) Analog HI [0]	Digit 4 (MSB)= 0, 1, -1 (= 10), " - " (= 11)
99		(char) Analog HI [1]	Digit 3= 0 a 9

BYTE	MODBUS	Variable	Description
100	50	(char) Analog HI [2]	Digit 2= 0 a 9
101		(char) Analog HI [3]	Digit 1= 0 a 9
102	51	(char) Analog HI [4]	Digit 0 (LSB)= 0 a 9
103		(char) Analog LO [0]	Digit 4 (MSB)= 0, 1, -1 (= 10), " - " (= 11)
104	52	(char) Analog LO [1]	Digit 3= 0 a 9
105		(char) Analog LO [2]	Digit 2= 0 a 9
106	53	(char) Analog LO [3]	Digit 1= 0 a 9
107		(char) Analog LO [4]	Digit 0 (LSB)= 0 a 9
108 a 117	54 a 58	RESERVED	
118	59	(char) -	-
119		(char) RS Baud Rate	0= 1200 baud 1= 2400 baud 2= 4800 baud 3= 9600 baud 4= 19200 baud
120	60	(char) RS Address [0]	Digit 1 (MSB)= 0 a 9
121		(char) RS Address [1]	Digit 0 (LSB)= 0 a 9
122	61	(char) RS Protocol	0= ASCII 1= ISO 1745 2= MODBUS
123		(char) RS (RS4) Delay	0= 30ms 1= 60ms 2= 100ms

Dynamic Values (Only Reading)

BYTE	MODBUS	Variable	Description
124	62	(integer) Value Display	Integer 2 bytes with filter , Round y hold
125			
126	63	(integer) Value Setpoint 1	Integer 2 bytes, Programmed value
127			
128	64	(integer) Value Setpoint 2	Integer 2 bytes, Programmed value
129			
130	65	(integer) Value Setpoint 3	Integer 2 bytes, Programmed value
131			
132	66	(integer) Value Setpoint 4	Integer 2 bytes, Programmed value
133			
134	67	(integer) Value Pico	Integer 2 bytes Without filter
135			
136	68	(integer) Value Valle	Integer 2 bytes Without filter
137			
138	69	(char) State Relay/Opto 1	0= OFF 1= ON
139		(char) State Relay/Opto 2	0= OFF 1= ON
140	70	(char) State Relay/Opto 3	0= OFF 1= ON
141		(char) State Relay/Opto 4	0= OFF 1= ON
142	71	(integer) Ana.Output HI	Integer 2 bytes, Programmed value
143			
144	72	(integer) Ana.Output LO	Integer 2 bytes, Programmed value
145			
146	73	(char) Overflow sign	0= Positive 1= Negative
147		(char) Overflow	0= No 1= Yes
148	74	(char) Software Version	1 byte (min. 100, máx. 255)
149		(char) -	-

DISEÑOS Y TECNOLOGIA, S.A.
Polígono Industrial Les Guixeres
C/ Xarol 8 C
08915 BADALONA-SPAIN
Tel : +34 - 93 339 47 58
Fax : +34 - 93 490 31 45
E-mail : dtl@ditel.es