



ELR-51AS

Earth leakage relay 1 toroidal input

Modbus RTU Communication protocol

Introduction

Modbus is a master-slave communication protocol able to support up to 247 slaves organized as a bus or as a star network. The physical link layer is RS485.

The communication is half-duplex. The network messages can be Query-Response or Broadcast type. The Query-Response command is transmitted from the Master to an established Slave and generally it is followed by an answering message.

The Broadcast command is transmitted from the Master to all Slaves and is never followed by an answer.

Generic RTU message structure:

START OF FRAME	=	Starting message marker. (silence on line for time >= 4 characters)
ADDRESS FIELD [1 CHAR]	=	Includes device address in which you need to communicate in Query-Response mode. In case the message is a Broadcast type it includes 00.
FUNCTION CODE [1 CHAR]	=	The operation code that you need to perform.
DATA FIELD [N CHAR]	=	Includes the data field.
ERROR CHECK [2 CHARS]	=	Field for the error correction code.
END OF FRAME	=	End message marker. (silence on line for time >= 4 characters)

Wait time for response:

Request of 16 register (64 bytes) → typical 15ms / worst 30 ms

Request of 64 register (128 bytes) → typical 15ms / worst 50 ms

Scan rate max recommended: 250 ms

Reading multiple registers [function code 03h]

Reads the binary contents of holding registers (2X references) in the slave.

Broadcast is not supported. The Query message specified the starting register and quantity of register to be read.

QUERY:

	0° Byte	1° Byte	2-3° Byte	4-5° Byte	6-7° Byte	
Start of Frame	Address Field	Function Code	Start Address	Number of Registers	Check Sum	End of Frame
START OF FRAME	=	Starting message marker.				
ADDRESS FIELD	=	Device address (0x01... 0xF7)		(1 byte).		
FUNCTION CODE	=	Operation code (0x03)		(1 byte).		
START ADDRESS	=	First register address to be read		(2 byte).		
No. OF REGISTERS	=	Number of registers (max 252 bytes) to read		(4 bytes [1 long] for 1 measure value).		
CHECK SUM	=	Check sum.				
END OF FRAME	=	End message marker.				

WARNING:

It is possible to read more than one variable at the same time (**max 252 bytes**) only if their addresses are consecutive and the variables on the same line cannot be divided.

The register data in the response message are packet as two bytes per register, with the binary contents right justified within each byte.

For each register, the first byte contains the high order bits and the second contains the low order bits.

RESPONSE:

	0° Byte	1° Byte	2° Byte	n° Byte	n+1 - n+2° Byte	
Start of Frame	Address Field	Function Code	Number of Bytes	Data	Check Sum	End of Frame
START OF FRAME	=	Starting message marker.				
ADDRESS FIELD	=	Device address (0x01... 0xF7)		(1byte).		
FUNCTION CODE	=	Operation code (0x03)		(1 Byte).		
No. OF SEND BYTES	=	Number of data bytes (0x00...??)		(1 byte). 1 register requires 2 data bytes.		
D0, D1, ..., Dn	=	data bytes (0x00...??)		(Nr. of register x 2 = n. byte).		
CHECK SUM	=	Check sum.				
END OF FRAME	=	End message marker.				

Write multiple registers [function code 10h]

Write values into a sequence of holding registers (2X references).

WARNING: It is possible to write more than one variable at the same time only if their addresses are consecutive and the variables on the same line cannot be divided. (max 64 bytes).

QUERY:

	0° Byte	1° Byte	2-3° Byte	4-5° Byte	6° Byte	n° Byte	n+1 - n+2° Byte	
Start of Frame	Address Field	Function Code	Start Address	Number of Registers	Number of Bytes	Data	Check Sum	End of Frame

START OF FRAME	=	Starting message marker.						
ADDRESS FIELD	=	Device address (0x01... 0xF7)			(1 byte).			
FUNCTION CODE	=	Operation code (0x10)			(1 byte).			
START ADDRESS	=	First register address to be written			(2 byte).			
No. OF REGISTER	=	Number of registers to be written (1 to 4, ...)			(2 byte).			
No. OF BYTES	=	Number of data bytes (HEX)			(1 byte): 1register requires 2 data bytes.			
D0,D1,...,Dn	=	Data bytes (0x00...?)			(1 byte) (Nr. of register x 2 = n. byte).			
CHECK SUM	=	Check sum.						
END OF FRAME	=	End message marker.						

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

RESPONSE:

	0° Byte	1° Byte	2-3° Byte	4-5° Byte	6-7° Byte	
Start of Frame	Address Field	Function Code	Start Address	Number of Registers	Check Sum	End of Frame

START OF FRAME	=	Starting message marker.				
ADDRESS FIELD	=	Device address (0x01... 0xF7)			(1 byte).	
FUNCTION CODE	=	Operation code (0x10)			(1 byte).	
START ADDRESS	=	First register address to be written			(2 byte).	
No. OF REGISTER	=	Number of registers to be written			(2 byte).	
ERROR CHECK	=	Check sum.				
END OF FRAME	=	End message marker.				

BROADCAST COMMAND:

It is possible to send a broadcast command (Address Field equal 0x00) for all write command.

QUERY:

	0° Byte	1° Byte	2-3° Byte	4-5° Byte	6° Byte	n° Byte	n+1 - n+2° Byte	
Start of Frame	0x00	Function Code	Start Address	Number of Registers	Number of Bytes	Data	Check Sum	End of Frame

RESPONSE: No Response.

Diagnostic [function code 08h]

This function provides a test for checking the communication system. Broadcast is not supported.

The instrument's protocol has only the sub-function 0 of the diagnostics sub-functions set of the standard modbus protocol.

The Query and the Response messages are the following:

QUERY:

	0° Byte	1° Byte	2-3° Byte	n° Byte	n+1 - n+2° Byte	
Start of Frame	Address Field	Function Code	Sub Function	Data	Check Sum	End of Frame

START OF FRAME	=	Starting message marker.				
ADDRESS FIELD	=	Device address (0x01... 0xF7)			(1 byte).	
FUNCTION CODE	=	Operation code (0x08 HEX)			(1 byte).	
SUB FUNCTION	=	Sub-function 0 (0x00 0x00)			(2 byte).	
DATA	=	Max 10 data bytes.				
CHECK SUM	=	Check sum.				
END OF FRAME	=	End message marker.				

RESPONSE:

The response must be the loopback of the same data.

	0° Byte	1° Byte	2-3° Byte	n° Byte	n+1 - n+2° Byte	
Start of Frame	Address Field	Function Code	Sub Function	Data	Check Sum	End of Frame

START OF FRAME	=	Starting message marker.				
ADDRESS FIELD	=	Device address (0x01... 0xF7)			(1 byte).	
FUNCTION CODE	=	Operation code (0x08 HEX)			(1 byte).	
SUB FUNCTION	=	Sub-function 0 (0x00 0x00)			(2 byte).	
DATA	=	Data bytes.				
CHECK SUM	=	Check sum.				
END OF FRAME	=	End message marker.				

DIAGNOSTIC EXAMPLE:

QUERY		RESPONSE	
Field Name	Example (Hex)	Field Name	Example (Hex)
Slave Address	0x01	Slave Address	0x01
Function Code	0x08	Function Code	0x08
Sub-function Hi	0x00	Sub-function Hi	0x00
Sub-function Lo	0x00	Sub-function Lo	0x00
Data Hi	0xF1	Data Hi	0xF1
Data Lo	0xA7	Data Lo	0xA7
Error Check (CRC)	0x??	Error Check (CRC)	0x??
	0x??		0x??

Report slave ID [function code 11h]

This function returns the type of the instrument and the current status of the slave run indicator. Broadcast is not supported.

The Query and the Response messages are the following:

QUERY:

Start of Frame	0° Byte Address Field	1° Byte Function Code	2 - 3° Byte Check Sum	End of Frame
START OF FRAME	= Starting message marker.			
ADDRESS FIELD	= Device address (0x01... 0xF7)		(1 byte).	
FUNCTION CODE	= Operation code (0x11)		(1 byte).	
CHECK SUM	= Check sum.			
END OF FRAME	= End message marker.			

RESPONSE:

Start of Frame	0° Byte Address Field	1° Byte Function Code	2° Byte Byte Count	3° Byte Slave ID	4° Byte Run Indicator Status	5° - 6° Byte Check Sum	End of Frame
START OF FRAME	= Starting message marker.						
ADDRESS FIELD	= Device address (0x01... 0xF7)			(1 byte).			
FUNCTION CODE	= Operation code (0x11)			(1 byte).			
BYTE COUNT	= Number of data bytes (0x02)			(1 byte).			
SLAVE ID	= Slave ID identifier (0x81)			(1 byte).			
RUN INDICATOR STATUS	= Run indicator status (0xFF)			(1 byte).			
DATA	= Data bytes						
CHECK SUM	= Check sum.						
END OF FRAME	= End message marker.						

The normal response has the slave ID identifier (0x81) and the run indicator Status (0xFF).

REPORT SLAVE ID EXAMPLE:

QUERY		RESPONSE	
Field Name	Example (Hex)	Field Name	Example (Hex)
Slave Address	0xXX	Slave Address	0x01
Function Code	0x11	Function Code	0x11
Error Check (CRC)	0x??	Byte count	0x02
	0x??	Slave ID	0x81
		Run indicator status	0xFF
		Error Check (CRC)	0x??
			0x??

Error message from slave to master

When a slave device receives a not valid query, it does transmit an error message.

RESPONSE:

Start of Frame	0° Byte Address Field	1° Byte Function Code	2° Byte Error Code	3 - 4° Byte Check Sum	End of Frame
START OF FRAME	= Starting message marker.				
ADDRESS FIELD	= Device address (0x01... 0xF7)		(1 byte).		
FUNCTION CODE	= Operation code with bit 7 high		(1 byte).		
ERROR CODE	= Message containing communication failure		(1 byte).		
CHECK SUM	= Check sum.				
END OF FRAME	= End message marker.				

ERROR EXAMPLE:

QUERY		RESPONSE	
Field Name	Example (Hex)	Field Name	Example (Hex)
Slave Address	0x01	Slave Address	0x 01
Function Code	0x03	Function Code	0x83 (1)
Starting Address Hi	0x00	Error Code	0x02 (2)
Starting Address Lo	0x00	Error Check (CRC)	0x??
Number Of Word Hi	0x00		0x??
Number Of Word Lo	0x05		
Error Check (CRC)	0x??		
	0x??		

(1): Function Code transmitted by master with bit 7 high.
(2): Error type:
0x01 = Illegal Function
0x02 = Illegal data address
0x03 = Illegal data value
0x0F = Communication Protection Enabled
(password enabled)
Write PASSWORD parameter before retry.

Registers

The following tables shown all the device registers.

Measures – Long format

Register ^{HEX}	Register ^{DEC}	Word	Description	R/W	M.U.	Type
100	256	2	CURRENT	R	mA	Unsigned
102	258	2	CURRENT FILTERED	R	mA	Unsigned
104	260	2	MAX CURRENT	R	mA	Unsigned
106	262	2	MAX CURRENT FILTERED	R	mA	Unsigned
108	264	2	THD	R	% * 100	Unsigned
10A	266	2	CREST FACTOR	R	[thousandths]	Unsigned
10C	268	2	STATUS	R	bit 0 set: ALARM bit 1 set: TRIP bit 2 set: OPEN bit 3 set: DISABLE bit 4 set: OVER	-

Measures – Float format

Register ^{HEX}	Register ^{DEC}	Word	Description	R/W	M.U.	Type
200	512	2	CURRENT	R	mA	Float
202	514	2	CURRENT FILTERED	R	mA	Float
204	516	2	MAX CURRENT	R	mA	Float
206	518	2	MAX CURRENT FILTERED	R	mA	Float
208	520	2	THD	R	% * 100	Float
20A	522	2	CREST FACTOR	R	-	Float

Harmonics

Register ^{HEX}	Register ^{DEC}	Word	Description	R/W	Measure Unit	Type
1000	4096	2	1 ST HARMONIC (<i>Fundamental</i>)	R	% * 100	Unsigned
1002	4098	2	2 ND HARMONIC	R	% * 100	Unsigned
1004	4100	2	3 RD HARMONIC	R	% * 100	Unsigned
---	---	---	---	---	---	---
107C	4220	2	63 TH HARMONIC	R	% * 100	Unsigned

Setup Input

Warning: Must be send the entire parameter length (2 words or 1 word – see the long of each parameter) for a correct command setting.

Register ^{HEX}	Register ^{DEC}	Word	Description	R/W	Parameters	Default
2000	8192	2	FILTER	R/W	0: disable 1: enable	0
2002	8194	2	TRIP - mA	R/W	30 ÷ 30'000	30
2004	8196	2	TRIP DELAY - ms	R/W	20 ÷ 10'000 <i>Note: the value must be a multiple of 20 ms.</i>	20
2006	8198	2	WARNING RECOVERY MODE	R/W	0: OFF 1: manual 1: automatic	0
2008	8200	2	WARNING PERCENT	R/W	20 ÷ 90 percent of trip	50
200A	8202	2	WARNING DELAY - ms	R/W	20 ÷ 1'000'000 <i>Note: the value must be a multiple of 20 ms.</i>	20
200C	8204	2	HYSTERESIS ^[TRIP & WARNING]	R/W	95 ÷ 50% over the limit set	90
200E	8206	2	FAIL SAFE	R/W	0: disable 1: enable	0

Remote reset [trip/warning state]

Warning: Must be send the entire parameter length (2 words or 1 word – see the long of each parameter) for a correct command setting.

Register ^{HEX}	Register ^{DEC}	Word	Description	R/W	Parameters
2A00	10752	2	MANUAL RESET INPUT	W	Write 0x0A0A ^{hex} (2570 ^{dec}) to reset

Remote test

Warning: Must be send the entire parameter length (2 words or 1 word – see the long of each parameter) for a correct command setting.

Register ^{HEX}	Register ^{DEC}	Word	Description	R/W	Parameters
2A20	10784	2	MANUAL TEST INPUT	W	Write 0x5050 ^{hex} (20560 ^{dec}) to test

For further details please contact:

Contrel elettronica s.r.l.

Via San Fereolo, 9

I-26900 Lodi

Tel: +39 0371 30207 / 30761 / 35386

Fax: +39 0371 32819

www.contrel.it

