

## MR-DI16

Modulo espansione 16 ingressi digitali

PROTOCOLLO DI COMUNICAZIONE  
MODBUS

## MR-DI16

Expansion unit 16 digital inputs

MODBUS COMMUNICATION PROTOCOL

## PROTOCOLLO MODBUS

Il modulo ingressi MR-DI16 supporta il protocollo di comunicazione Modbus RTU sulla porta seriale RS-485.

Quando si utilizza il protocollo Modbus RTU, la struttura del messaggio di comunicazione è così costituita:

Start frame	Indirizzo (8 bit)	Funzione (8 bit)	Dati (N x 8 bit)	CRC (16bit)	End frame
-------------	-------------------	------------------	------------------	-------------	-----------

- Il campo Indirizzo contiene l'indirizzo dello strumento slave cui il messaggio viene inviato.
- Il campo Funzione contiene il codice della funzione che deve essere eseguita dallo slave.
- Il campo Dati contiene i dati inviati allo slave o quelli inviati dallo slave come risposta ad una domanda.
- Il campo CRC consente sia al master che allo slave di verificare se ci sono errori di trasmissione.

## FUNZIONI MODBUS

Le funzioni disponibili sono:

03H = Read input register	Consente la lettura delle misure disponibili nell'MR-DI16
10H = Preset multiple register	Permette la scrittura di più parametri
11H = Report slave ID	Permette di leggere informazioni relative all'apparecchio

Tempo di risposta MR-DI16:

- Caso tipico: 150ms
- Caso peggiore: 300ms

## MODBUS PROTOCOL

The MR-DI16 digital inputs board supports the communication protocol Modbus RTU on the RS-485 serial port.

If one selects the Modbus RTU protocol, the structure communication message has the following structure:

Start frame	Address (8 bit)	Function (8 bit)	Data (N x 8 bit)	CRC (16bit)	End frame
-------------	-----------------	------------------	------------------	-------------	-----------

- The Address field holds the serial address of the slave destination device.
- The Function field holds the code of the function that must be executed by the slave.
- The Data field contains data sent to the slave or data received from the slave in response to a query.
- The CRC field allows the master and slave devices to check the message integrity.

## MODBUS FUNCTIONS

The available functions are:

03H = Read input register	Allows to read the MR-DI16 measures
10H = Preset multiple register	Allows writing several parameters
11H = Report slave ID	Allows to read information about the device

MR-DI16 response time:

- Typical case: 150ms
- Worst case: 300 ms

## CALCOLO DEL CRC (CHECKSUM per RTU)

*Esempio di calcolo:*

```
static unsigned char auchCRCHi [] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x40
};
```

```
static unsigned char auchCRCLo [] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4,
0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD,
0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7,
0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2,
0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB,
0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91,
0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88,
0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80,
0x40
}
```

```
unsigned short CRC16 (ptMsg, usDataLen)
unsigned char *ptMsg; /* message to calculate CRC upon */
unsigned short usDataLen; /* number of bytes in message */
{
    unsigned char uchCRCHi = 0xFF; /* CRC high byte */
    unsigned char uchCRCLo = 0xFF; /* CRC low byte */
    unsigned ulIndex;

    while (usDataLen--) /* pass through message buffer */
    {
        ulIndex = uchCRCHi ^ *ptMsg++; /* calculate the CRC */
        uchCRCHi = uchCRCLo ^ auchCRCHi [ ulIndex ];
        uchCRCLo = auchCRCLo [ ulIndex ];
    }
    return (uchCRCHi << 8 | uchCRCLo);
}
```

## CRC CALCULATION (CHECKSUM for RTU)

*Example of the CRC calculation:*

## FUNZIONE 03H: READ INPUT REGISTER

La funzione 03 permette di leggere una o più grandezze consecutive in memoria. L'indirizzo di ciascuna grandezza è indicato nelle tabelle riportate nelle ultime pagine del presente manuale. Se l'indirizzo richiesto non è compreso nella tabella o il numero di registri richiesti è maggiore del numero consentito, l'MR-DI16 ritorna un messaggio di errore (vedi tabella errori).

### Richiesta Master:

Indirizzo slave	01h
Funzione	03h
MSB indirizzo registro	01h
LSB indirizzo registro	04h
MSB numero registri	00h
LSB numero registri	20h
MSB CRC	04h
LSB CRC	2Fh

Nell'esempio vengono richiesti, allo slave numero 1, 32 registri consecutivi a partire dall'indirizzo 0104h. Quindi vengono letti i registri dal 0104h al 0123h. Il comando termina sempre con il valore checksum CRC.

### Risposta Slave:

Indirizzo slave	01h
Funzione	03h
Numero di byte	40h
MSB dato 0104h	00h
LSB dato 0104h	00h
-----	----
MSB dato 0103h	00h
LSB dato 0103h	01h
MSB CRC	83h
LSB CRC	F6h

La risposta è composta sempre dall'indirizzo dello slave, dalla funzione richiesta dal Master e dai dati dei registri richiesti. La risposta termina sempre con il valore di checksum CRC.

## FUNCTION 03H: READ INPUT REGISTER

The Modbus function 03 allows to read one or more consecutive registers from the slave. The address of each measures is given in the tables on the final page of this manual. If the measure address is not included in the table or the number of requested registers exceeds the acceptable max number, the MR-DI16 will return an error code (see error table).

### Master query:

Slave address	01h
Function	03h
MSB register address	01h
LSB register address	04h
MSB register number	00h
LSB register number	20h
MSB CRC	04h
LSB CRC	2Fh

In the above example, slave 1 is requested for 32 consecutive registers beginning with address 0104h. Thus, registers from 0104h to 0123h will be returned. As usual, the message ends with CRC checksum.

### Slave response:

Slave address	01h
Function	03h
Byte number	40h
MSB data register 0104h	00h
LSB data register 0104h	00h
-----	----
MSB data register 0103h	00h
LSB data register 0103h	01h
MSB CRC	83h
LSB CRC	F6h

The response is always composed of the slave address, the function code requested by the master and the contents of the requested registers. The answer ends with the CRC.

## FUNZIONE 10H: PRESET MULTIPLE REGISTER

Questa funzione permette di modificare più parametri consecutivamente o parametri composti da più di 2 byte. È permessa la scrittura di massimo 32 registri consecutivi.

### Richiesta Master:

Indirizzo slave	01h
Funzione	10h
MSB indirizzo registro	14h
LSB indirizzo registro	00h
MSB numero registri	00h
LSB numero registri	0Ah
Numero byte	14h
MSB Dato 1400h	0000h
LSB Dato 1400h	0001h
-----	----
MSB Dato 1408h	0000h
LSB Dato 1408h	0001h
MSB CRC	A5h
LSB CRC	1Ah

### Risposta Slave:

Indirizzo slave	01h
Funzione	10h
MSB indirizzo registro	14h
LSB indirizzo registro	00h
MSB numero byte	00h
LSB numero byte	0Ah
MSB CRC	45h
LSB CRC	FEh

## FUNZIONE 11H: REPORT SLAVE ID

Questa funzione permette di identificare il tipo di multimetro.

### Richiesta Master:

Indirizzo slave	01h
Funzione	11h
MSB CRC	C0h
LSB CRC	2Ch

### Risposta Slave:

Indirizzo slave	01h
Funzione	11h
Numero di byte	02h
Dato 1 (Tipo) ❶	74h
Dato 2 (Run indicator status)	FFh
MSB CRC	DAh
LSB CRC	7Ch

❶ 74h = MR-DI16

## ERRORI

Nel caso lo slave riceva un messaggio errato, segnala la condizione al master rispondendo con un messaggio composto dalla funzione richiesta in OR con 80h, seguita da un codice di errore.

Nella seguente tabella vengono riportati i codici di errore inviati dallo slave al master.

CODE	ERRORE
01	Funzione non valida
02	Indirizzo registro illegale
03	Valore del parametro fuori range

## FUNCTION 10H: PRESET MULTIPLE REGISTER

This function allows to modify multiple parameters with a single message, or to preset a value longer than one register. It has allowed the writing of a maximum of 32 consecutive registers.

### Master query:

Slave address	01h
Function	10h
MSB register address	14h
LSB register address	00h
MSB register number	00h
LSB register number	0Ah
Byte number	14h
MSB data	0000h
LSB data	0001h
-----	----
MSB data	0000h
LSB data	0001h
MSB CRC	A5h
LSB CRC	1Ah

### Slave response:

Slave address	01h
Function	10h
MSB register address	14h
LSB register address	00h
MSB byte number	00h
LSB byte number	0Ah
MSB CRC	45h
LSB CRC	FEh

## FUNCTION 11H: REPORT SLAVE ID

This function allows to identify the multimeter type.

### Master query:

Slave address	01h
Function	11h
MSB CRC	C0h
LSB CRC	2Ch

### Slave response:

Slave address	01h
Function	11h
Byte number	02h
Data 1 (Type) ❶	74h
Data 2 (Run indicator status)	FFh
MSB CRC	DAh
LSB CRC	7Ch

❶ 74h = MR-DI16

## ERRORS

In case the slave receives an incorrect message, it answers with a message composed by the queried function OR with 80h, followed by an error code byte.

In the following table are reported the error codes sent by the slave to the master.

CODE	ERROR
01	Invalid function
02	Invalid address
03	Parameter out of range

**MISURE FORNITE DAL PROTOCOLLO DI COM.**  
(Utilizzabili con funzione 03H)

**MEASURES SUPPLIED BY COMM. PROTOCOL**  
(To be used with function 03H)

INDIRIZZO / ADDRESS	WORDS	MISURA	MEASURE	UNITÀ/UNIT	FORMATO / FORMAT
0100H	2	Stato fisico ingressi ❶	Inputs status ❶	bool	Unsigned long
0102H	2	Stato logico ingressi ❶	Logical inputs status ❶	bool	Unsigned long
0104H	2	Stato ingresso n°1 ❷	Input n°1 status ❷	bool	Unsigned long
0106H	2	Stato ingresso n°2	Input n°2 status	bool	Unsigned long
0108H	2	Stato ingresso n°3	Input n°3 status	bool	Unsigned long
010AH	2	Stato ingresso n°4	Input n°4 status	bool	Unsigned long
010CH	2	Stato ingresso n°5	Input n°5 status	bool	Unsigned long
010EH	2	Stato ingresso n°6	Input n°6 status	bool	Unsigned long
0110H	2	Stato ingresso n°7	Input n°7 status	bool	Unsigned long
0112H	2	Stato ingresso n°8	Input n°8 status	bool	Unsigned long
0114H	2	Stato ingresso n°9	Input n°9 status	bool	Unsigned long
0116H	2	Stato ingresso n°10	Input n°10 status	bool	Unsigned long
0118H	2	Stato ingresso n°11	Input n°11 status	bool	Unsigned long
011AH	2	Stato ingresso n°12	Input n°12 status	bool	Unsigned long
011CH	2	Stato ingresso n°13	Input n°13 status	bool	Unsigned long
011EH	2	Stato ingresso n°14	Input n°14 status	bool	Unsigned long
0120H	2	Stato ingresso n°15	Input n°15 status	bool	Unsigned long
0122H	2	Stato ingresso n°16	Input n°16 status	bool	Unsigned long
0124H	2	Stato logico ingresso n°1 ❷	Logical input n°1 status ❷	bool	Unsigned long
0126H	2	Stato logico ingresso n°2	Logical input n°2 status	bool	Unsigned long
0128H	2	Stato logico ingresso n°3	Logical input n°3 status	bool	Unsigned long
012AH	2	Stato logico ingresso n°4	Logical input n°4 status	bool	Unsigned long
012CH	2	Stato logico ingresso n°5	Logical input n°5 status	bool	Unsigned long
012EH	2	Stato logico ingresso n°6	Logical input n°6 status	bool	Unsigned long
0130H	2	Stato logico ingresso n°7	Logical input n°7 status	bool	Unsigned long
0132H	2	Stato logico ingresso n°8	Logical input n°8 status	bool	Unsigned long
0134H	2	Stato logico ingresso n°9	Logical input n°9 status	bool	Unsigned long
0136H	2	Stato logico ingresso n°10	Logical input n°10 status	bool	Unsigned long
0138H	2	Stato logico ingresso n°11	Logical input n°11 status	bool	Unsigned long
013AH	2	Stato logico ingresso n°12	Logical input n°12 status	bool	Unsigned long
013CH	2	Stato logico ingresso n°13	Logical input n°13 status	bool	Unsigned long
013EH	2	Stato logico ingresso n°14	Logical input n°14 status	bool	Unsigned long
0140H	2	Stato logico ingresso n°15	Logical input n°15 status	bool	Unsigned long
0142H	2	Stato logico ingresso n°16	Logical input n°16 status	bool	Unsigned long

❶ Esempio: Il valore all'indirizzo 0100H è 0xFFFF0005 (solo LSB), =0x0000000000000101 vuol dire che gli ingressi 1 e 3 sono attivi.

❷ Lo stato 0 del bit indica l'ingresso non impegnato.  
Lo stato 1 del bit indica l'ingresso attivo.

❶ Example: The value at address 0100H is 0xFFFF0005 (only LSB), =0x0000000000000101 means that inputs 1 and 3 are active.

❷ The status 0 of the bit indicates a not active input.  
The status 1 of the bit indicates an active input.

**MISURE FORNITE DAL PROTOCOLLO DI COM**  
(Utilizzabili con funzione 03H e 10H)

**MEASURES SUPPLIED BY COMM. PROTOCOL**  
(To be used with function 03H and 10H)

INDIRIZZO / ADDRESS	WORDS	MISURA	MEASURE	UNITÀ/UNIT	FORMATO / FORMAT
0200H	2	Ritardo ON ingresso n°1 ③	ON delay input n°1 ③	ms	Unsigned long
0202H	2	Ritardo ON ingresso n°2	ON delay input n°2	ms	Unsigned long
0204H	2	Ritardo ON ingresso n°3	ON delay input n°3	ms	Unsigned long
0206H	2	Ritardo ON ingresso n°4	ON delay input n°4	ms	Unsigned long
0208H	2	Ritardo ON ingresso n°5	ON delay input n°5	ms	Unsigned long
020AH	2	Ritardo ON ingresso n°6	ON delay input n°6	ms	Unsigned long
020CH	2	Ritardo ON ingresso n°7	ON delay input n°7	ms	Unsigned long
020EH	2	Ritardo ON ingresso n°8	ON delay input n°8	ms	Unsigned long
0210H	2	Ritardo ON ingresso n°9	ON delay input n°9	ms	Unsigned long
0212H	2	Ritardo ON ingresso n°10	ON delay input n°10	ms	Unsigned long
0214H	2	Ritardo ON ingresso n°11	ON delay input n°11	ms	Unsigned long
0216H	2	Ritardo ON ingresso n°12	ON delay input n°12	ms	Unsigned long
0218H	2	Ritardo ON ingresso n°13	ON delay input n°13	ms	Unsigned long
021AH	2	Ritardo ON ingresso n°14	ON delay input n°14	ms	Unsigned long
021CH	2	Ritardo ON ingresso n°15	ON delay input n°15	ms	Unsigned long
021EH	2	Ritardo ON ingresso n°16	ON delay input n°16	ms	Unsigned long

③ Ritardo attivazione. Permette di filtrare lo stato per evitare rimbalzi.

Range: 0-999999999

0: istantaneo

1: 10ms

2: 20ms

3: 30ms (DEFAULT)

...

10: 100ms

...

③ Activation delay. Allow to filter the input status to avoid bouncing.

Range: 0-999999999

0: instantaneous

1: 10ms

2: 20ms

3: 30ms (DEFAULT)

...

10: 100ms

...

**PARAMETRI SETUP**

(Utilizzabili con funzione 03H e 10H)

**SETUP PARAMETERS**

(To be used with function 03H and 10H)

INDIRIZZO / ADDRESS	WORDS	MENU	MENU	MIN	MAX	DEF
1400H	2	Indirizzo seriale nodo	Serial node address	1	247	1
1402H	2	Velocità seriale ①	Serial speed ①	0	5	3
1404H	2	Bit di stop ②	Stop bits ②	0	1	0
1046H	2	Formato dati ③	Data format ③	0	2	0
1408H	2	Ritardo risposta (ms)	Reply delay (ms)	5	100	10

① Velocità di trasmissione della porta di comunicazione.

0: 4800 bps

1: 9600 bps

2: 19200 bps

3: 38400 bps

4: 57600 bps

5: 115200 bps

② Numero bit di stop.

0: 1 bit di stop

1: 2 bit di stop

③ Formato dati

0: 8 bit, no parità

1: 8 bit, dispari

2: 8 bit, pari

① Serial communication speed.

0: 4800 bps

1: 9600 bps

2: 19200 bps

3: 38400 bps

4: 57600 bps

5: 115200 bps

② Number of stop bits.

0: 1 stop bit

1: 2 stop bit

③ Data format.

0: 8 bit, no parity

1: 8 bit, odd

2: 8 bit, even

Per ulteriori informazioni contattare:

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  
E-Mail: [contrel@contrel.eu](mailto:contrel@contrel.eu)  
[www.contrel.it](http://www.contrel.it)